NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DEVELOPMENT OF GRAPHICAL TIME RESPONSE USING THE OPTSYSX PROGRAM

by

Harry Allen Diel

September 1984

Thesis Advisor: D. J. Collins

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Development of Graphical Time Response Using the OPTSYSX Program	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1984
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Harry Allen Diel	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943	12. REPORT DATE September 1984
	13. NUMBER OF PAGES 241
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93943	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution 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)

Graphical Time Response
OPTSYSX
DISSPLA

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis discusses the modification of and additions to an existing Optimal Systems Control FORTRAN Program (OPTSYSX) originally obtained from Professor Arthur E. Bryson of Stanford University and subsequently redesigned to run interactively on the IBM 3033 VM/CMS by Lieutenant Commander John G. Hoden of the Naval Postgraduate School (NPS).

The modified FORTRAN Program (OPTSYSX) and the additional FORTRAN Programs (OPTCALC) and (OPTPLOT) are now designed to run interactively under VM/CMS on the IBM 3033 utilizing a library double precision numerical integration subroutine and highly accurate time response of a system which has been designed on the OPTSYSX Program. This series of programs permits the user to rapidly design, analyze and test all types of Optimal Systems Control problems. Examples of the various types of problems are worked through to illustrate all of the capabilities available.

Approved for public release; distribution unlimited.

Development of Graphical Time Response using the OPTSYSX Program

by

Harry A. Diel
Commander, United States Navy
B.S., University of Illinois, 1967

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1984

ABSTRACT

This thesis discusses the modification of and additions to an existing Optimal Systems Control FORTRAN Program (OPTSYS) originally obtained from Professor Arthur E. Bryson of Stanford University and subsequently redesigned to run interactively on the IBM 3033 VM/CMS by Lieutenant Commander John G. Hoden of the Naval Postgraduate School (NPS).

The modified FORTRAN program (OPTSYSX) and the additional FORTRAN Programs (OPTCALC) and (OPTPLOT) are now designed to run interactively under VM/CMS on the IBM 3033 utilizing a library double precision numerical integration subroutine and high resolution precision plotting software to provide the user with a highly accurate time response of a system which has been designed on the OPTSYSX Program. This series of programs permits the user to rapidly design, analyze and test all types of Optimal Systems Control problems. Examples of the various types of problems are worked through to illustrate all of the capabilities available.

TABLE OF CONTENTS

I. INTRODUCTION 10

II. THE OPTSYSX COMPUTER PROGRAM 12

A. GENERAL BACKGROUND 12

B. PROGRAM OVERVIEW 12

C. OPTSYS EXEC 13

D. OPTSYSX MODIFICATIONS 13

1. RDMATF Subroutine 14

2. RDMAT Subroutine 15

3. WRTMAT Subroutine 15

E. OPTCALC PROGRAM 16

1. System Integration 16

2. System Equation Representation 17

3. System Time Response 20

F. OPTPLOT PROGRAM 21

1. General Operation 22

2. VERSATEC Pen-plots 23

III. SYSTEM USE AND EXAMPLES 26

A. OPEN-LOOP SYSTEM TIME RESPONSE 26

B. CLOSED-LOOP SYSTEM TIME RESPONSE 42

C. FILTER CLOSED-LOOP SIMULATION 61

D. FILTER AND REGULATOR CLOSED LOOP TIME RESPONSE 83

IV. CONCLUSIONS AND RECOMMENDATIONS 103

A. CONCLUSIONS 103

B. RECOMMENDATIONS 108

1. Program Availability 108

2. Program Memory Requirements 108

3. Further Modifications 109

LIST OF REFERENCES 239

BIBLIOGRAPHY 240

INITIAL DISTRIBUTION LIST 241

LIST OF FIGURES

3.1 Open-loop Time Response 41

3.2 Closed-loop Time Response 60

3.3 Filter Closed-loop Time Response 82

3.4 Filter plus Regulator Closed-loop Time Response 102

4.1 X-29A Longitudinal Time Response 104

4.2 X-29A Longitudinal Time Response 105

4.3 X-29A Longitudinal Time Response 106

4.4 X-29A Longitudinal Time Response 107

SYMBOLS

A = state (Ns,Ns) or output (No,No) weighting matrix

B = control (Nc,Nc) weighting matrix

C = control gain matrix (Nc,Ns)

D = control (No,Nc) or noise (No,Ng) feedforward matrix

F = open-loop dynamics matrix (Ns,Ns)

G = control distribution matrix (Ns,Nc)

GAM = state disturbance distribution matrix (Ns,Ng)

H = measurement scaling matrix (No,Ns)

K = estimator gain matrix (Ns,No)

Nc = number of controls

Ng = number of process noise sources

Ns = number of states

No = number of observations or measurements

Q = white process noise covariance matrix (Ng,Ng)

R = white meas. noise covariance matrix (No,No)

S = steady-state covariance matrix of control (Nc,Nc)

u = control vector (Nc,1)

uc = control input (Nc,1)

x = state vector (Ns,1)

xdot = state vector derivative (Ns,1)

xe = estimate of state vector (Ns,1)

xedot = derivative of estimate of state vector (Ns,1)

$\tilde{x}$ = state reconstruction error (Ns,1)

y = output/measurement vector (No,1)

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to Professor D. J. Collins, whose assistance and encouragement contributed immeasurably to this research.

I would also like to express my thanks to Commander V. C. Gordon, USN, whose willing assistance and technical expertise were invaluable during a critical phase of this research.

I wish to dedicate this thesis to my wife, Gloria, and children, Stephanie, Gregory and Angela. Without their constant love, support, and understanding this work would not have been possible.

I. INTRODUCTION

The purpose of this thesis is to describe and demonstrate the modification and additions to the existing FORTRAN program (OPTSYSX) which is used in the study, design, and application of Optimal Systems Control theory.

The OPTimal SYStems control program (OPTSYS) was originated in 1971 by Hall [Ref. 1] to support his research in rotary-wing aircraft control systems. The most recent program modifications were made by Walker [Ref. 2] and Liu [Ref. 3] of Stanford University and are designated OPTSYS 4 and OPTSYS 5, respectively. The OPTSYS modifications made by Hoden [Ref. 4] were primarily devoted to creating a user-friendly interactive version (OPTSYSX) of the OPTSYS 4 Program.

The goal of this thesis work was to develop a program set which will operate in an interactive mode and plot the time response of a State Variable Control System which has been developed and/or analyzed using the OPTSYSX Program. Minor modifications to the OPTSYSX Program were necessary to allow the user to build a data file of the matrices required for the time response calculations plus additional matrices which could be used again upon reentry to the OPTSYSX Program without the laborious and time-consuming task of reentering each data element in all of the required matrices.

It is assumed that the reader/user is familiar with the basic concepts of Control Theory and Optimal Systems Design. The symbol/naming conventions of Bryson [Ref. 5] are used in the program operation discussion and in the examples of the problems solved using this system. A glossary of the symbols and abbreviations used in this discussion is provided on page 8.

An explanation of the OPTSYSX capabilities and a program set overview are presented first.

This work concludes with examples of various types of problems demonstrated in the interactive mode, including a copy of each terminal session with the final results. A set of complete program listings are included in Appendices A, B, C and D.

II. THE OPTSYSX COMPUTER PROGRAM

A. GENERAL BACKGROUND

OPTSYSX is an interactive, double-precision FORTRAN program employing modern control theory analysis techniques. Its extensive capabilities include the synthesis and analysis of filters and regulators as well as eigensystem analysis, modal distribution, transfer function analysis and power spectral density computations. The modifications to the OPTSYS Program introduced by this thesis work have not affected any of the program's original capabilities.

B. PROGRAM OVERVIEW

OPTSYSX is an extremely large and complex program with over 2800 lines of code. In order to use this program in its small (set up for 32X32 matrices) version, the user is required to extend the IBM 370/3033 virtual machine (VM) memory capacity beyond 720 kilobytes which is the default VM memory size. A significant increase in the size of the OPTSYSX program would make the program too large to operate on a one megabyte VM, the largest virtual memory available on a user's virtual machine. The high resolution plotting software is limited to single precision variables. Therefore double precision library routines cannot be called from the plotting program. For these reasons the task of obtaining the time response of a system was divided between three programs, OPTSYSX, OPTCALC and OPTPLOT. An Executive program (OPTSYS EXEC) was written to make the interfacing of the three programs transparent to the user.

Minor modifications were made to the OPTSYSX program including the addition of three subroutines to handle the input and output of matrix data to and from a data file on the user's disk. The OPTCALC program performs the double precision numerical integration of the system of equations over time and creates another disk data file of the state variable variation with time. OPTPLOT takes this time response data and presents it in a graphical format on the TEK 618 graphical display or as a VERSATEC pen plot.

C. OPTSYS EXEC

The OPTSYS EXEC is written in the EXEC 2 language. This language allows the EXEC to issue almost any command that can be entered in the direct mode at the terminal. Therefore an EXEC is the ideal controller for the "black box" type of system where the user is not aware of what is actually taking place within the program(s). The OPTSYS EXEC was written to complete all of the required interfacing between the three programs (OPTSYSX, OPTCALC and OPTPLOT), without the direct guidance or control of the user. By answering questions presented on the terminal screen, the user determines the logic flow through the EXEC while the EXEC establishes the appropriate FILEDEFs and loads the programs required by the user's desires.

D. OPTSYSX MODIFICATIONS

Three subroutines (RDMATF, RDMAT, and WRTMAT) were added to the OPTSYSX Program for data file read error check, matrix input from a data file and matrix output to a data file, respectively. These three subroutines provide the user with the opportunity to save the [F], [G], [H], [GAMMA], [A] and [B] matrices for use in a subsequent run of the OPTSYSX Program. The WRTMAT subroutine also saves the [C] and [K] matrices for system time response calculation and plotting by the OPTCALC and OPTPLOT programs.

1. RDMATF Subroutine

The RDMATF subroutine is used to check for the existence of a previously generated file containing matrix data. Seven flags may be set by this subroutine. Six of these flags correlate with the six matrices that the user may save for reuse later in the OPTSYSX Program. The remaining flag (IRDMAT) must be set to enable the RDMAT subroutine to read matrix information from the data file. A READ statement of the form

```
READ (9,111,ERR=222,END=333) A,B
```

(where "111", "222" and "333" designate line numbers for the FORMAT statement and branch on ERROR or branch on END-OF-FILE routines, respectively) is used for the data file check. The nonexistence of the file or premature END-OF-FILE are detected by the ERR and END checks which cause a branch to a routine that sets the IRDMAT flag to "0" and returns to the calling program.

If no error is detected during the initial read attempt, the variable B is checked for the sentinel "1". This second check is to help ensure that the file is actually a file which contains valid matrix elements. The user is then presented a message which asks if he/she wants to use the matrices which are available. The user may respond with on of three answers:

1. Use all of the matrices.
2. Use selected matrices.
3. Use none of the matrices.

If the answer is "1" or "2", the subroutine reads the matrix dimensions (Ns, Nc, No and Ng) from the data file and changes the IRDMAT flag to 1 to key the RDMAT subroutine to read the matrix elements from the data file. If the answer is "1" all of the matrix-save flags are set to "1".

If the answer is "2", the user is given the opportunity to select individual matrices for reuse while rejecting other matrix information. This is accomplished by setting individual matrix-save flags to "1" if the matrix is to be saved and "0" if new matrix data will be input from the terminal. If the answer is "3" (Use none of the matrices) the IRDMAT flag is set to 0 and the subroutine returns to the main program. When all actions have been completed, the flag information and the matrix dimensions are passed to the main program for later use.

2. RDMAT Subroutine

The RDMAT subroutine is used to read all of the matrix information in the data file and transfer the information to the appropriate variables. As previously discussed, The actions of this subroutine depend on the status of the IRDMAT flag. If this flag had been set to "0", no read operations are attempted and program flow immediately returns to the calling program.

When the IRDMAT flag is set to "1", the RDMAT subroutine reads the matrix dimensions from the data file, and uses these dimensions to transfer the matrix information from the file to the appropriate variables. The file matrix dimensions are used for the read operations and are not fed back to the calling program, since the dimensions of some of the matrices which are not being reused may have changed from the the previous run. Similarly, using the current matrix dimensions in the RDMAT subroutine would cause data read-in problems due to the changing number of elements in each matrix as the matrix dimensions vary.

3. WRTMAT Subroutine

The WRTMAT subroutine is used to write a data file of the data file flags, the matrix dimensions and selected matrices. When the user has completed the analysis/design of the system of interest, the WRTMAT subroutine asks the user if he/she wants to calculate the time response of the system which the user just designed. If the user answers YES, the WRTMAT subroutine generates a data file of appropriate matrix information and halts execution of the OPTSYSX program. Control then reverts to the OPTSYS EXEC. If the user answers NO, the WRTMAT subroutine returns control to the main program and normal OPTSYSX program operation continues.

The information written to the data file consists of 2 "1"s (which are used as a sentinel or flag by the RDMATF subroutine (as previously explained) and in a similar manner by the OPTSYS EXEC), followed by the matrix dimensions (Ns, Nc, Nc and Ng) and then by the [F], [G], [H], [GAMMA], [C], [K], [A] and [B] matrices. These matrix elements are written to the OPTMAT DATA file using a 4D20.13 format as a compromise between the maximum feasible accuracy of data exchange between the double-precision programs and the use of a moderate amount of the user's disk space.

E. OPTCALC PROGRAM

1. System Integration

The OPTCALC program is a FORTRAN interactive double-precision system integration routine. This program uses the International Mathematical & Statistical Library (IMSL) subroutine DGEAR to perform the numerical integration of the system under analysis. The stiff system mode of DGEAR is used in order to provide the capability to do time response calculations of the X-29A longitudinal axis back-up mode system which is an 98 X 98 stiff system.

2. System Equation Representation

The OPTCALC program uses the state variable format such as

$$xdot = [F]*x + [G]*uc \tag{2.1}$$

to define the system. In this system the [F] matrix is the open-loop dynamics matrix (system or plant) and the [G] matrix is the control matrix. The variable assignments are **x** as the state vector and **uc** as the control input vector. It follows that **xdot** is the time derivative of x.

Various forms of equation 2.1 are used for all the time response calculations. The [F] matrix is modified to [F+G*C][1] for closed-loop (regulator only) system calculations as in equations 2.2 and 2.3.

$$xdot = [F+G*C]*x + [G]*uc \tag{2.2}$$

$$u = [C]*x + uc \tag{2.3}$$

For this closed loop system, the [C] matrix is the control gain or regulator gain matrix and **u** is the total input vector.

The combined filter and regulator systems can be represented by equations 2.2, 2.4, 2.5, 2.6 and 2.7. The [H] matrix is the measurement scaling (observer output) matrix and the [K] matrix is the estimator or Kalman filter observer gain matrix. The variables **xe**, **xedot** and $\tilde{x}$ are the state estimate vector, the derivative of the state estimate vector and the state reconstruction error, respectively.

---

[1] The OPTSYSX sign convention for the C matrix is the negative of the standard normally used in controls. Therefore [F+G*C] has the correct sign for OPTSYSX matrices.

$$xdot = [F+G*C]*x + [G]*uc \qquad (2.2)$$

$$z = [H]*x \qquad (2.4)$$

$$xedot = [F]*xe + [G]*u + [K]*(z - [H]*xe) \qquad (2.5)$$

$$u = [C]*xe \qquad (2.6)$$

$$\tilde{x} = x - xe \qquad (2.7)$$

Equations 2.2, 2.4, 2.5, and 2.6 can be combined into the augmented matrix form of equation 2.8.

$$\begin{vmatrix} xdot \\ xedot \end{vmatrix} = \begin{vmatrix} F & G*C \\ K*H & F+G*C-K*H \end{vmatrix} * \begin{vmatrix} x \\ xe \end{vmatrix} + \begin{vmatrix} G \\ G \end{vmatrix} *uc \qquad (2.8)$$

Equation 2.8 is an augmented equation in which the Ns dimension has been doubled and the state and state estimate vectors have been combined into one vector of 2*Ns length. The [G] matrix is also augmented by repeating the first Ns rows again beginning at row Ns+1, making the new [G] matrix dimensions (2*Ns,Nc).

In a similar manner, a filter only system can represented by the same equations with the [C] matrix set to 0. These equations are:

$$xdot = [F]*x + [G]*uc \qquad (2.1)$$

$$z = [H]*x \qquad (2.4)$$

$$xedot = [F]*xe + [G]*u + [K]*(z - [H]*xe) \quad (2.5)$$

$$\tilde{x} = x - xe \quad (2.7)$$

The combination of equations 2.1, 2.4, 2.5 and 2.7 into an augmented [F] matrix is similar to equation 2.8 but with fewer terms and the upper right quadrant equal to zero. This filter-only augmented system equation is shown below.

$$\begin{vmatrix} xdot \\ xedot \end{vmatrix} = \begin{vmatrix} F & 0 \\ K*H & F-K*H \end{vmatrix} * \begin{vmatrix} x \\ xe \end{vmatrix} + \begin{vmatrix} G \\ G \end{vmatrix} *uc \quad (2.9)$$

a. System Selection

When the CPTCALC program is run, the |F|, |G| and |C| matrices (and |H| and |K| matrices, if available) are presented on the terminal as a check and as a reminder of the characteristics of the system that has been passed from OPTSYSX to OPTCALC. The user is requested to select the type of system response to be calculated.

```
1. CPEN LCOP TIME RESPONSE.
2. CIOSED LOOP TIME RESPONSE.
3. CIOSED LOOP FILTER ONLY TIME RESPONSE.
4. CICSED LOOP FILTER + REGULATOR TIME RESPONSE.
```

Selection of 2, 3 or 4 forms the appropriate system matrix equations 2.2, 2.9 or 2.8, respectively and doubles the length of the |G| matrix, if required.

b. Defining Calculation Limits and Inputs

After the user determines the type of system under study, the OPTCALC program prompts for the integration start and stop times and the number of data points desired.

The user has some control over the tradeoff between curve fidelity and computer time used by varying the number of data points calculated. Computer time use is normally a factor only on very large systems. If less than 200 points are calculated, the OPTPLOT program uses a curve smoothing function which may cause minor inaccuracies in the plotted curve but avoids the sharp peaks and irregular appearance generated by plotting straight lines between an insufficient number of data points. When 200 or more points are calculated no smoothing is done. The points are connected by very short straight lines which has the appearance of a smooth curve.

Step or ramp functions are available as control inputs. Only one type of function can be used for each control, but the magnitude and start and stop times can be set as desired.

The state and state estimate initial conditions can be set to any value by the user. The control inputs and non-zero initial conditions can be used simultaneously, if desired. Before the time response calculations begin, the user is given the opportunity to make changes in any area of the system integration initial conditions that have been previously selected.

3. System Time Response

Equation 2.1 is evaluated directly in the open loop system response calculations. The FCN subroutine was written to evaluate the system of equations for the DGEAR IMSL subroutine. Each time the FCN subroutine is called by DGEAR, it updates the control inputs (**uc**) and then evaluates each state derivative by summing all the terms across that row of the [F]*x and [G]*uc matrices. The same FCN subroutine is used for all system integrations. As explained in the previous section, the [F] matrix is replaced by the [F+G*C] matrix for closed loop system response problems.

a. Systems With Filters

Augmented equations 2.9 and 2.8 are used for the time response evaluation of systems with filters only and systems with filters plus regulators, respectively. The augmented matrix is developed as a dummy matrix and is then inserted as the [F] matrix with the dimensions doubled (2*Ns). The [G] matrix is also augmented by repeating the first Ns rows again beginning at row Ns+1, making the new [G] matrix dimensions (2*Ns,Nc).

The augmented system can be evaluated by simply doubling the old system row and column dimension (Ns) and calling the DGEAR integration subroutine. Using this method, the existing FCN subroutine requires no changes to evaluate the augmented system.

b. OPTCALC Output

The OPTCALC program uses FILEDEF 8 for the data file output as well as FILEDEF 5 to read and write to the terminal. The output data file contains the following discrete information: the matrix dimensions Ns and Nc, the augmented matrix dimensions, the number of data points calculated and a flag to indicate that an augmented matrix was calculated. The [C] matrix is passed to permit the calculation of **u** the total control input to the system. The final portion of the data file is individual data points of time, external control input (**uc**) and each state (**x**) and state estimate (**xe**). This data file provides all of the data required by the OPTPLOT program to make a smooth graphical response curve.

F. OPTPLOT PROGRAM

The OPTPLOT program is a FORTRAN interactive plotting program using the Display Integrated Software System and

Plotting Language (DISSPLA) by Integrated Software Systems Corporation. This program provides the user a high resolution graphical display of the system's time response and if desired will provide a VERSATEC pen-plot of the same graph.

1. General Operation

Plotting data is received from the program OPTCALC via a data file on FILEDEF 8. The types of data provided in the file are discussed in the previous section. The program presents the user with a series of questions to determine:

```
1. The number of curves to plot.
2. Select the type of variable for each curve.
3. Select the variable subscript for each curve.
4. Select the number of headings and contents of each.
```

The program then plots the selected variables and provides the user a graphical display on the Tektronics 618 (TEK 618) display.

The following Main Menu is then presented which provides the user with the major decision points of the program.

```
1. BEGIN NEW GRAPH OF OTHER CONTROLS ,STATES, OR ESTIMATES.
2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
3. EDIT THE CURRENT GRAPH.
4. PLOT REVISED GRAPH ON THE TEK618.
5. QUIT AND/OR MAKE METAFILE OF THE CURVES
PREVIOUSLY SAVED.
```

The purpose of each of the selections is self-explanatory, however the methods of their use may not be. If number 3 is selected, the user is then presented the following Edit Menu of items to make additions, deletions or corrections to the curves that are plotted on the TEK 618 screen:

1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
2. DELETE CURVE FROM CURRENT PLOT.
3. EDIT CURVE TITLE(S).
4. EDIT PAGE HEADING(S).
5. CHANGE THE Y-AXIS SCALE.
6. CHANGE THE TIME AXIS SCALE.
7. CHANGE THE PLOT SIZE.
8. CHANGE THE LETTERING HEIGHT.
9. CHANGE POSITION OF THE LEGEND.
10. EDITING COMPLETE.

This extensive list of modification capabilities provides the user with the tool to make almost any imaginable alteration to an existing plot.

Since the OPTPLOT program receives the time response data from the OPTCALC program, item 6 of the Edit Menu cannot be used to expand the time scale beyond the time span previously calculated. Therefore the time axis change feature can be used only to select a subset of the original data.

2. VERSATEC Pen-plots

The VERSATEC pen-plots are provided through the DISSPOP portion of DISSPLA. In order to use the DISSPOP feature, a device independent plot file called a metafile must be generated.

To view the graphical time response plot on the TEK 618 terminal, the TEK618 option of DISSPLA must be called within the OPTPLOT plotting program. The graphical image data is then sent to the TEK 618 display screen.

A metafile is created when the COMPRS option of DISSPLA is called by the OPTPLOT plotting program. When the plotting program is executed with the COMPRS option, the graphical image data is sent to a metafile on the user's disk. The TEK618 option and COMPRS option are mutually exclusive (only one can be active at a time), therefore graphical data cannot go to both the terminal screen and the metafile, concurrently. As a further complication, the

TEK618 and COMPRS options cannot be used in alternating pattern, first to originate and edit each graph and then to add this graph to a metafile possibly containing several other graphs. Therefore if more than one pen-plot per terminal session is desired, some type of capability must be provided to save the information required to reproduce a given graph.

When the user attempts to leave the current plot (ie. selecting items 1, 2 or 5 of the Main Menu) the program asks the user to save the current graph data for later use in generating a metafile. This feature provides the capability to save any desired graph data in order to later make a metafile and obtain a pen-plot. When the COMPRS option is used (by selecting item 5 of the Main Menu), any number of graphs may be added to the metafile up to the limit of available user disk space (provided graph data has been previously saved). After exiting the OPTPLOT program, the OPTSYS EXEC asks the user if he/she wants a hard copy of the metafile that had been generated during the session. If the user answers YES, the OPTSYS EXEC calls the DISSPOP EXEC with the VRSTEC option.

When the user exits the DISSPOP EXEC, the OPTSYS EXEC gives the user the options to:

1. RUN OPTSYS AGAIN.
2. RUN OPTCALC AGAIN.
3. QUIT.

The option to run OPTSYSX again allows the user to use all or part of the matrices that had been saved in the data file without manually reentering each element. The OPTCALC option could be exercised if the user wants to use the same system matrices again, but change the control input or initial conditions or change the type of system (open, closed, filter only or filter plus regulator) that was evaluated on the previous run.

III. SYSTEM USE AND EXAMPLES

This chapter contains several basic examples of the four types of problems which may be solved using OPTSYSX, OPTCALC and OPTPLOT under control of the OPTSYS EXEC. Included with these examples are copies of each recorded terminal session.

A. OPEN-LOOP SYSTEM TIME RESPONSE

The following open-loop system example was taken from [Ref. 6, pp 5.3 - 5.7].

The full terminal session is recorded below, with user input at the left margin in <u>lower case letters</u> or <u>numbers</u> below each "?".

```
record on
BEGIN RECORDING OF TERMINAL SESSION
R; T=0.01/0.02 19:58:26
optsys

     THE OPTSYS EXEC CONTROLS A TRIO OF PROGRAMS:

      1. OPTSYSX FORTRAN   (SYSTEM ANALYSIS)
      2. OPTCALC FORTRAN   (CALCULATE TIME RESPONSE)
      3. OPTPLOT FORTRAN   (DISSPLA PLOTTING ROUTINE)

   EACH PROGRAM PASSES INFORMATION TO THE NEXT PROGRAM
   THROUGH A DATA FILE WRITTEN TO THE USERS DISK.  IN THIS
   CASE, THESE FILES ARE "OPTMAT DATA" AND "OPTPLOT DATA".
   THE SIZE OF THESE FILES VARY WITH THE SYSTEM ORDER, AND
   CAN USE ABOUT 20% OF THE USERS DISK SPACE.  THEREFORE
   ENSURE THAT SUFFICIENT DISK SPACE IS AVAILABLE.

     - TYPE "E" TO EXIT, ANY OTHER ENTRY TO CONTINUE -

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

    YOU HAVE A DATA FILE NAMED 'OPTMAT DATA' ON YOUR A
    DISK  THAT WAS PREVIOUSLY GENERATED BY THE  OPTSYS
    PROGRAM  AND  CONTAINS THE F, G, H, GAMMA, A AND B
    MATRICES FROM THAT RUN.

    IF  YOU WOULD LIKE TO USE THESE SAME MATRICES  FOR
    THIS  RUN, THE  OPTSYS  PROGRAM  WILL  READ IN THE
    DESIRED DATA AT THE APPROPRIATE TIME,

                   IF YOU TYPE (Y) ES.

    ANY OTHER INPUT WILL RESULT IN THAT FILE BEING ERASED!
```

```
*******************************************************************
y

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

      DO YOU WANT THE NUMERICAL OUTPUT FROM OPTSYSX TO GO
        TO YOUR TERMINAL S(CREEN) OR TO A D(ISK) FILE?
                          (S OR D)

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

s

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

            OUTPUT WILL COME TO YOUR TERMINAL SCREEN.

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

LOADING OPTSYS....
EXECUTION BEGINS...
   OPTSYSX IS A COMPLETELY INTERACTIVE OPTIMAL SYSTEMS CONTROL
      PROGRAM. IT WILL SOLVE NUMEROUS CONTROL PROBLEMS ON THE
      FOLLOWING TYPES OF SYSTEMS CONTROL EQUATIONS:

              XDOT = {F}*X + {G}*U + {GAM}*(W+W0)

                   MEASUREMENT EQUATION--

              Z = {H}*X + {D}*W + V

                   REGULATOR PERFORMANCE INDEX--

              J = 1/2 * INTEGRAL (Y *{A}*Y + U *{B}*U) DT

                   STATE FEEDBACK GAIN DEFINITION--

                        U = -{C}*X

              DO YOU WISH TO CONTINUE?  TYPE "YES" OR "NO".
y
                        --DATA ENTRY--

     ALTHOUGH OPTSYSX IS SPECIFICALLY DESIGNED TO READ
     ALL MATRIX DATA INTERACTIVELY, SEVERAL ALTERNATE
     METHODS ARE AVAILABLE TO USERS:

        METHOD 1--THE "F","G", AND "GAMMA" MATRICES
           MAY BE READ FROM SEPARATE DATA FILES.

        METHOD 2--THE "F","G", AND "GAMMA" MATRICES MAY BE
           EXPLICITLY DEFINED WITHIN SUBROUTINE "SETUP".

      {NOTE: IN EITHER CASE, THE USER SHOULD OBTAIN A COPY
             OF THE PROGRAM LISTING AND EXAMINE
             THE EXAMPLES CONTAINED IN S/R "SETUP".}

          DO YOU WISH TO CONTINUE?  TYPE "YES" OR "NO".
y

     DO YOU WISH TO INPUT THE "F", "G", AND "GAMMA"
          MATRICES FROM SUBROUTINE "SETUP" IAW THE
          METHOD DESCRIBED ON THE PREVIOUS SCREEN?

              TYPE "YES" OR "NO".
n
```

```
                         GENERAL OPTSYSX OPTIONS:
          OPTION 1 -- SYSTEM ANALYSIS WITHOUT
                      OPEN-LOOP EIGENSYSTEM CALCULATIONS.
          OPTION 2 -- SYSTEM ANALYSIS WITH OPEN-LOOP
                      EIGENSYSTEM CALCULATIONS.
          OPTION 3 -- OPEN-LOOP EIGENSYSTEM FOUND
                      AND PROGRAM TERMINATES.
                      {"F"-MATRIX ENTRY FOLLOWS IMMEDIATELY.}
          OPTION 4 -- MODAL DISTRIBUTION MATRICES COMPUTED
                      WITHOUT FILTER OR REGULATOR SYNTHESIS
                      OR STEADY-STATE ANALYSIS.
               SELECT AN OPTION: 1,2,3, OR 4.
?
4

     DO YOU DESIRE RMS VALUES OF STATE AND CONTROL?
          TYPE "YES" OR "NO".
n

     OPEN-LOOP TRANSFER FUNCTION OPTIONS:
        OPTION 1 -- NO OPEN-LOOP TRANSFER FUNCTIONS COMPUTED.
        OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.
        OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.
        OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.
        SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

     NOISE TRANSFER FUNCTION OPTIONS:
        OPTION 1 -- NO NOISE TRANSFER FUNCTIONS COMPUTED.
        OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.
        OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.
        OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.
          SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

     COMPENSATOR TRANSFER FUNCTION OPTIONS:
        OPTION 1 -- NO COMP. TRANSFER FUNCTIONS COMPUTED.
        OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.
        OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.
        OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.
             {NOTE: A COMPENSATOR TRANSFER FUNCTION CAN BE
                    COMPUTED ONLY IF BOTH A REGULATOR
```

```
                    AND FILTER ARE SYNTHESIZED
                    AND/OR INPUT.}

            SELECT AN OPTION: 1, 2, 3, OR 4.
?
1


        WILL A FEED-FORWARD DISTRIBUTION MATRIX
        {"D" - MATRIX} BE INPUT ?

                TYPE "YES" OR "NO".
n

       THIS OPTION DETERMINES THE CRITERIA FOR DECIDING WHEN A
       MARKOV PARAMETER IS ZERO-THE MARKOV PARAMETER INDICATES
       THE ORDER OF THE NUMERATOR POLYNOMIAL OF EACH TRANSFER
       FUNCTION.

         ALL "N" ZEROS OF THIS POLYNOMIAL ARE PRINTED OUT AND
         THIS TEST TELLS HOW MANY EXTRA ROOTS EXIST AT Z = 0.
         LESS THAN 10.0**{-IE} IS CONSIDERED ZERO.

           THE DEFAULT VALUE OF THIS PARAMETER {IE} IS 6.
           IN OTHER WORDS, IE = 1.0E-6.

      IF YOU DESIRE A DIFFERENT MARKOV CRITERIA,
      TYPE THE INTEGER VALUE.

      IF YOU DESIRE THE DEFAULT VALUE, TYPE "0" {ZERO}
?
0


      POWER SPECTRAL DENSITY {PSD} OPTION 1 :

    OPTION 1 -- COMPUTE THE PSD OF THE OUTPUTS AND/OR THE
                CONTROLS OF THE CONTROLLED SYSTEM WHEN FORCED BY
                PROCESS AND MEASUREMENT NOISE.  {NOTE: BOTH A
                REGULATOR AND A FILTER MUST BE RESIDENT IN THE
                PROGRAM TO USE THIS OPTION.}

    OPTION 2 -- SAME AS OPTION 1 ABOVE BUT ONLY PRINT THE
                RESIDUES OF EACH TRANSFER FUNCTION
                USED IN THE PSD COMPUTATION.

    OPTION 3 -- NOT DESIRED.

            SELECT AN OPTION: 1, 2, OR 3.
?
3


            THE "F", "G", "H", "GAM", "A" AND "B" MATRICES
              FROM YOUR PREVIOUS OPTSYS RUN WERE SAVED.

            THE FOLLOWING OPTIONS ARE AVAILABLE:
                 1. USE ALL OF THE SAME MATRICES AGAIN.
                 2. USE SELECTED MATRICES AGAIN.
                 3. INPUT ALL NEW MATRICES.

            ENTER 1, 2, OR 3.

            NOTE: EACH SAVED MATRIX WILL BE REDISPLAYED AT
            THE PROPER INPUT SEQUENCE INTERVAL
            AND YOU WILL HAVE THE OPTION OF CHANGING
            INDIVIDUAL MATRIX ELEMENTS.
?
```

```
1
   FIAG/PARAMETER SETTINGS FOR THIS RUN ARE AS FOLLCWS:
IOL  IQ  IR  ISS  IM  ITF1  ITF2  ITF3  IFDFW  IE  IDEBUG
 3    0   0   0    1   0     0     0     0      0   0
ISET  IDSTAB  IPSD  IYU  INORM  IREG  NS  NC  NOB  NG
 0      0      0     0    0      0     4   0   0    0

 ORDER OF SYSTEM =  4
 NUMBER CF CONTROLS =  0
 NUMBER CF CBSERVATICNS =  0
 NUMBER CF PROCESS NCISE SOURCES =  0

              THE SYSTEM MATRIX {"F"-MATRIX}...

     0.0          1.00000     0.0          0.0
     0.0         -0.41500    -0.01110      0.0
     9.80000     -1.43000    -0.01980      0.0
     0.0          0.0         1.00000      0.0

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" CR "NO".
n

   OPEN LOOP DYNAMICS MATRIX....................F..

  0.0           0.1000D+01   0.0           0.0
  0.0          -0.4150D+00  -0.1110D-01    0.0
  0.9800D+01   -0.1430D+01  -0.1980D-01    0.0
  0.0           0.0          0.1000D+01    0.0

   OPEN LOOP EIGENVALUES..........DET(SI-F)..

 0.0          :-6.80767D-01: 1.22984D-01, 3.80349D-01:

   OPEN LOOP RIGHT EIGENVECTOR MATRIX.......T....

 0.0            -3.449493D-02 -1.375658D-02  9.725766D-03
 0.0             2.348301D-02 -5.391019D-03 -4.036193D-03
 0.0             5.622534D-01  1.229836D-01  3.803490D-01
 1.000000D+00   -8.259115D-01  1.000000D+00  0.0

   OPEN LOOP LEFT EIGENVECTOR MATRIX......T-INV..

 3.738739D+01  9.009009D+01 -4.260481D-15  1.000000D+00
-5.858605D+00  2.423391D+01  4.069740D-01  0.0
-4.222608D+01 -7.007502D+01  3.361245D-01  0.0
 2.231407D+01 -1.316561D+01  1.918868D+00  0.0

   MODAL MEASUREMENT SCALING MATRIX...H(BAR)*T..

   0.0            0.0            0.0            0.0
```

```
            DO YOU WISH TO OBTAIN A TIME RESPONSE
              OF THE SYSTEM YOU ARE EVALUATING?
                          (Y OR N)

       NOTE: YOU MUST BE LOGGED ON AT A DUAL SCREEN
             (TEK 618) TERMINAL TO UTILIZE THIS MODE.

   THE F (SYSTEM), G (CONTROL), H (OBSERVABLES), GAM (NOISE),
      A (OUTPUT COST) AND B (CONTROL COST) MATRICES WILL BE
         SAVED FOR REENTRY TO THE MAIN OPTSYS PROGRAM.
y

****************************************************************
           IF YOU ARE DISSATISFIED WITH THE RESULTS
           THUS FAR AND WOULD LIKE TO EXIT TO CMS,

                    -TYPE 'Y' TO EXIT-

               (ANY OTHER INPUT TO CONTINUE)
****************************************************************

LOADING OPTCALC....
EXECUTION BEGINS...

     DURING THIS SECTION OF THE PROGRAM YOU WILL:

  - SELECT THE TYPE OF SYSTEM RESPONSE TO PLOT
     (OPEN LOOP, CLOSED LOOP, OR FILTER/REGULATOR)
  - PROVIDE START AND STOP TIME FOR PLOTTING CALCULATIONS
  - SELECT THE TYPE OF DRIVING FUNCTION(S) (STEP OR RAMP)
  - PROVIDE START AND STOP TIMES FOR THE DRIVING FUNCTION(S)
  - PROVIDE DRIVING FUNCTION MAGNITUDE(S).

                 CLEAR THE SCREEN TO CONTINUE

                  THE F MATRIX

     0.0           1.00000      0.0          0.0
     0.0          -0.41500     -0.01110      0.0
     9.80000      -1.43000     -0.01980      0.0
     0.0           0.0          1.00000      0.0

                  THE G MATRIX

     0.0
     0.0
     0.0
     0.0

                  THE C MATRIX

     0.0           0.0          0.0          0.0

     THE FOLLOWING PLOTTING OPTIONS ARE AVAILABLE IF THE
     REQUIRED MATRICES WERE CALCULATED IN OPTSYSX:

  1.  OPEN LOOP TIME RESPONSE
       XDOT = {F}*X + {G}*UC

  2.  CLOSED LOOP TIME RESPONSE
       XDOT = {F-G*C}*X + {G}*UC,    U =  {C}*X

  3.  OPTIMIZED FILTER CLOSED LOOP SYSTEM RESPONSE.
```

```
       XDOT = {F}*X + {G}*UC,        Z = {H}*X
       XHDOT = {F}*XH +{G}U + {K}*{Z - H*XH}

  4.   FILTER + REGULATOR CLOSED LOOP SYSTEM RESPONSE.
       XDOT = {F+G*C}*X + {G}*UC,               Z = {H}*X
       XHDOT = {F}*XH +{G}U + {K}*{Z - H*XH},   U = {C}*XH

          SELECT 1, 2, 3 OR 4.
?
1


          AT WHAT TIME DO YOU WANT TO START
          THE TIME RESPONSE CALCULATIONS?

          INPUT START TIME IN SECONDS. (NORMALLY 0.0)
?
0


          AT WHAT TIME DO YOU WANT TO STOP
          THE TIME RESPONSE CALCULATIONS?

            INPUT STOP TIME IN SECONDS.
?
25


     THIS PROGRAM DIVIDES THE TIME INTERVAL YOU HAVE
     JUST SPECIFIED INTO UP TO 500 SMALL INTERVALS FOR
     THE INTEGRATION AND PLOTTING ROUTINES.  IN ORDER
     TO SAVE COMPUTER TIME, THE NUMBER OF POINTS CAN BE
     CAN BE REDUCED WITH SOME LOSS IN CURVE FIDELITY.

          HOW MANY POINTS DO YOU WANT TO CALCULATE?
?
500

 DOES THE SYSTEM UTILIZE A DRIVING FUNCTION (CONTROL INPUT)?
                              (Y)ES OR (N)O
n

 DOES THE SYSTEM START WITH ALL INITIAL CONDITIONS = 0.0 ?
                              (Y)ES OR (N)O?
n

          WHAT IS THE INITIAL CONDITION FOR X( 1) ?
?
0.02

          WHAT IS THE INITIAL CONDITION FOR X( 2) ?
?
0

          WHAT IS THE INITIAL CONDITION FOR X( 3) ?
?
0

          WHAT IS THE INITIAL CONDITION FOR X( 4) ?
?
0

          THIS IS YOUR LAST OPPORTUNITY TO
          MAKE CHANGES IN THE FOLLOWING AREAS.

            1.  SELECT ANOTHER TYPE OF SYSTEM TO PLOT
```

(OPEN, CLOSED, FILTER OR FILTER/REGULATOR)

2. START AND STOP TIMES
3. DRIVING FUNCTIONS
4. INITIAL CONDITIONS
5. CONTINUE

SELECT A NUMBER BETWEEN 1 AND 5.

?
5

THE FOLLOWING INFORMATION IS PROVIDED ONLY
FOR AN INDICATION OF PROPER PROGRAM OPERATION.

ALL CONTROLS, STATES AND STATE ESTIMATES CAN BE PLOTTED.

TIME	U(1)	X(1)	X(2)	X(3)
0.0	0.0	0.2000000D-01	0.0	0.0
0.50	0.0	0.1995704D-01	-0.2532513D-03	0.9752469D-01
1.00	0.0	0.1967379D-01	-0.9447647D-03	0.1937236D+00
1.50	0.0	0.1895459D-01	-0.1982623D-02	0.2872200D+00
2.00	0.0	0.1764746D-01	-0.3282819D-02	0.3757431D+00
2.50	0.0	0.1564162D-01	-0.4763866D-02	0.4563456D+00
3.00	0.0	0.1286715D-01	-0.6343471D-02	0.5256108D+00
3.50	0.0	0.9295838D-02	-0.7936863D-02	0.5798568D+00
4.00	0.0	0.4942574D-02	-0.9456467D-02	0.6153434D+00
4.50	0.0	-0.1333996D-03	-0.1081268D-01	0.6284849D+00
5.00	0.0	-0.5827877D-02	-0.1191559D-01	0.6160655D+00
5.50	0.0	-0.1199212D-01	-0.1267744D-01	0.5754551D+00
6.00	0.0	-0.1843467D-01	-0.1301571D-01	0.5048190D+00
6.50	0.0	-0.2492487D-01	-0.1285666D-01	0.4033160D+00
7.00	0.0	-0.3119815D-01	-0.1213910D-01	0.2712751D+00
7.50	0.0	-0.3696332D-01	-0.1081832D-01	0.1103423D+00
8.00	0.0	-0.4191173D-01	-0.8869902D-02	-0.7641071D-01
8.50	0.0	-0.4572836D-01	-0.6293238D-02	-0.2844264D+00
9.00	0.0	-0.4810447D-01	-0.3114546D-02	-0.5076571D+00
9.50	0.0	-0.4875163D-01	0.6108039D-03	-0.7386238D+00
10.00	0.0	-0.4741670D-01	0.4796890D-02	-0.9685450D+00
10.50	0.0	-0.4389725D-01	0.9326917D-02	-0.1187536D+01
11.00	0.0	-0.3805676D-01	0.1405416D-01	-0.1384885D+01
11.50	0.0	-0.2983914D-01	0.1880428D-01	-0.1549396D+01
12.00	0.0	-0.1928159D-01	0.2337913D-01	-0.1669798D+01
12.50	0.0	-0.6525273D-02	0.2756201D-01	-0.1735216D+01
13.00	0.0	0.8177035D-02	0.3112445D-01	-0.1735684D+01
13.50	0.0	0.2445706D-01	0.3383438D-01	-0.1662679D+01
14.00	0.0	0.4183151D-01	0.3546543D-01	-0.1509680D+01
14.50	0.0	0.5970749D-01	0.3580735D-01	-0.1272699D+01
15.00	0.0	0.7739326D-01	0.3467693D-01	-0.9507871D+00
15.50	0.0	0.9411463D-01	0.3192943D-01	-0.5464664D+00
16.00	0.0	0.1090370D+00	0.2746961D-01	-0.6608227D-01
16.50	0.0	0.1212926D+00	0.2126234D-01	0.4799671D+00
17.00	0.0	0.1300134D+00	0.1334194D-01	0.1077141D+01
17.50	0.0	0.1343671D+00	0.3819764D-02	0.1706829D+01
18.00	0.0	0.1335980D+00	-0.7110370D-02	0.2346610D+01
18.50	0.0	0.1270681D+00	-0.1916965D-01	0.2970698D+01
19.00	0.0	0.1143002D+00	-0.3199492D-01	0.3550599D+01
19.50	0.0	0.9501896D-01	-0.4514306D-01	0.4055963D+01
20.00	0.0	0.6918881D-01	-0.5809930D-01	0.4455624D+01
20.50	0.0	0.3704708D-01	-0.7028958D-01	0.4718817D+01
21.00	0.0	-0.8704588D-03	-0.8109723D-01	0.4816533D+01
21.50	0.0	-0.4371323D-01	-0.8988342D-01	0.4722976D+01
22.00	0.0	-0.9031068D-01	-0.9601136D-01	0.4417079D+01
22.50	0.0	-0.1391800D+00	-0.9887356D-01	0.3884014D+01
23.00	0.0	-0.1885480D+00	-0.9792139D-01	0.3116637D+01
23.50	0.0	-0.2363885D+00	-0.9269602D-01	0.2116795D+01

```
24.00   0.0    -0.2804747D+00-0.8285960D-01 0.8964261D+00
24.50   0.0    -0.3184475D+00-0.6822545D-01-0.5216259D+00
25.00   0.0    -0.3478981D+00-0.4878571D-01-0.2103153D+01

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

          IF YOU ARE DISSATISFIED WITH THE RESULTS
          THUS FAR AND WOULD LIKE TO EXIT TO CMS,

               -TYPE 'Y' TO EXIT-

            (ANY OTHER INPUT TO CONTINUE)

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

B (120) R/O
C (121) R/O
E (122) R/O


          ... Your Fortran program is now being loaded ...
                ... execution will soon follow ...
EXECUTION BEGINS...

          THIS PORTION OF THE PROGRAM PLOTS:
               - THE STATES,
               - EXTERNAL CONTROL INPUTS,
               - FEEDBACK CONTROL INPUTS,
               - STATE ESTIMATES AND
               - RECONSTRUCTION ERRORS
          FROM THE DATA THAT YOU JUST CALCULATED.

          THE CAPABILITY IS ALSO AVAILABLE TO REVIEW ANY
          GRAPHS THAT YOU HAD PREVIOUSLY SAVED AS DATA
          FILES ON YOUR DISK.

               CLEAR THE SCREEN TO CONTINUE.


          THE FOLLOWING OPTIONS ARE AVAILABLE:

               1. PLOT THE DATA YOU JUST CALCULATED.
               2. PLOT A CURVE THAT YOU PREVIOUSLY SAVED.

          ENTER 1 OR 2
?
1


          YOU MAY PLOT UP TO 4 SYSTEM VARIABLES VS TIME.
HOW MANY VARIABLES DO YOU WISH TO PLOT?
?
4


 WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 1?

     1. STATE VARIABLE (IE., X1, X2, ETC)
     2. FEEDBACK CONTROL (IE., U = -C*X)
     3. CONTROL INPUT (IE., U1, U2, ETC.)
     4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
     5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
        X2-XHAT2, ETC)

          ENTER 1,2,3,4 OR 5
?
```

```
1

          WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
          YOU WANT TO PLOT AS THE NUMBER  1 CURVE VS TIME?
?
1

          WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE. (A) => ALPHA
                   (B) => BETA
                   (F) => PHI
                   (Q) => THETA
state y1

 WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 2?

    1. STATE VARIABLE (IE., X1, X2, ETC)
    2. FEEDBACK CONTROL (IE., U = -C*X)
    3. CONTROL INPUT (IE., U1, U2, ETC.)
    4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
    5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
       X2-XHAT2, ETC)

          ENTER 1,2,3,4 OR 5
?
1

          WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
          YOU WANT TO PLOT AS THE NUMBER  2 CURVE VS TIME?
?
2

     WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE. (A) => ALPHA
                   (B) => BETA
                   (F) => PHI
                   (Q) => THETA
state y2

 WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 3?

    1. STATE VARIABLE (IE., X1, X2, ETC)
    2. FEEDBACK CONTROL (IE., U = -C*X)
    3. CONTROL INPUT (IE., U1, U2, ETC.)
    4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
    5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
       X2-XHAT2, ETC)

          ENTER 1,2,3,4 OR 5
?
1
```

```
        WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
        YOU WANT TO PLOT AS THE NUMBER  3 CURVE VS TIME?
?
3


        WHAT IS THE CURVE LABEL FOR THIS VARIABLE?


    NOTE:  1. 40 CHARACTERS MAX LENGTH
           2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
              ENCLOSED IN PARENTHESES.
              IE.  (A) => ALPHA
                   (B) => BETA
                   (F) => PHI
                   (Q) => THETA
state y3

 WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 4?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC)

        ENTER 1,2,3,4 OR 5
?
1


        WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
        YOU WANT TO PLOT AS THE NUMBER  4 CURVE VS TIME?
?
4


    WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

    NOTE:  1. 40 CHARACTERS MAX LENGTH
           2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
              ENCLOSED IN PARENTHESES.
              IE.  (A) => ALPHA
                   (B) => BETA
                   (F) => PHI
                   (Q) => THETA
state y4


        YOU MAY USE UP TO 3 HEADINGS.
        HOW MANY HEADINGS DO YOU DESIRE ON THIS GRAPH?

                  0, 1, 2 OR 3
?
3

        WHAT IS THE DESIRED HEADING NUMBER 1?


        NOTE:  1. 40 CHARACTERS MAX LENGTH
               2. GREEK SYMBOLS WILL BE PRINTED FOR ANY
                  LETTERS ENCLOSED IN PARENTHESES.
                  IE.  (A)  => ALPHA
                       (B)  => BETA
                       (F)  => PHI
                       (Q)  => THETA
open loop system
```

```
          WHAT IS THE DESIRED HEADING NUMBER 2?

          NOTE:  1. 40 CHARACTERS MAX LENGTH
                 2. GREEK SYMBOLS WILL BE PRINTED FOR ANY
                    LETTERS ENCLOSED IN PARENTHESES.
                    IE. (A) => ALPHA
                        (B) => BETA
                        (F) => PHI
                        (Q) => THETA
example 2

          WHAT IS THE DESIRED HEADING NUMBER 3?

          NOTE:  1. 40 CHARACTERS MAX LENGTH
                 2. GREEK SYMBOLS WILL BE PRINTED FOR ANY
                    LETTERS ENCLOSED IN PARENTHESES.
                    IE. (A) => ALPHA
                        (B) => BETA
                        (F) => PHI
                        (Q) => THETA
modern control theory
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.

    THE FOLLOWING OPTIONS ARE AVAILABLE.

 1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
 2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
 3. EDIT THE CURRENT GRAPH.
 4. PLOT REVISED GRAPH ON THE TEK618.
 5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
    PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
3

               THE GRAPH EDIT MENU

   1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
   2. DELETE CURVE FROM CURRENT PLOT.
   3. EDIT CURVE TITLE(S).
   4. EDIT PAGE HEADING(S).
   5. CHANGE THE Y-AXIS SCALE.
   6. CHANGE THE TIME AXIS SCALE.
   7. CHANGE PLOT SIZE. (DEFAULT IS 8.5 X 6.0)
   8. CHANGE THE LETTERING HEIGHT.
   9. CHANGE POSITION OF THE LEGEND.
  10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.
?
9

          HOW MANY INCHES IN THE X DIRECTION
          (LEFT OR RIGHT), DO YOU WANT TO MOVE
          MOVE THE LEGEND BOX FROM ITS PRESENT POSITION

          NOTE: 1. DEFAULT PLOT SIZE IS  8.5 X 6.0
                2. LEFT IS NEGATIVE
                3. RIGHT IS POSITIVE
?
-4

          HOW MANY INCHES IN THE Y DIRECTION
          (UP OR DOWN), DO YOU WANT TO MOVE
          MOVE THE LEGEND BOX FROM ITS PRESENT POSITION
```

```
          NOTE: 1. DEFAULT PAGE SIZE IS  8.5 X 6.0
                2. DOWN IS NEGATIVE
                3. UP IS POSITIVE
?
0

                THE GRAPH EDIT MENU

    1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
    2. DELETE CURVE FROM CURRENT PLOT.
    3. EDIT CURVE TITLE(S).
    4. EDIT PAGE HEADING(S).
    5. CHANGE THE Y-AXIS SCALE.
    6. CHANGE THE TIME AXIS SCALE.
    7. CHANGE PLOT SIZE. (DEFAULT IS 8.5 X 6.0)
    8. CHANGE THE LETTERING HEIGHT.
    9. CHANGE POSITION OF THE LEGEND.
   10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.
?
10

    THE FOLLOWING OPTIONS ARE AVAILABLE.

 1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
 2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
 3. EDIT THE CURRENT GRAPH.
 4. PLOT REVISED GRAPH ON THE TEK618.
 5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
        PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
4

    THE FOLLOWING OPTIONS ARE AVAILABLE.

 1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
 2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
 3. EDIT THE CURRENT GRAPH.
 4. PLOT REVISED GRAPH ON THE TEK618.
 5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
    PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
5

     DO YOU WANT TO SAVE THE CURRENT GRAPH DATA TO
     BE USED LATER TO GENERATE A METAFILE?

                   Y OR N

     NOTE: A METAFILE IS REQUIRED FOR SMOOTH VERSATEC PLOTS.
     THERE WILL BE AN OPPORTUNITY TO GENERATE A METAFILE
     JUST BEFORE EXITING THIS PROGRAM.
y

     WHAT FILE NAME DO YOU WANT THE CURVE DATA STORED UNDER?
                   (8 CHARACTERS MAX)
openloop

          THE CURVE DATA IS BEING FILED UNDER OPENLOOP DATA
END OF DISSPLA 9.0 -- 26506 VECTORS GENERATED IN 2 PLOT FRAMES
    PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CA.
3442 VIRTUAL STORAGE REFERENCES; 6 READS; 0 WRITES.
```

```
          THE FOLLOWING OPTIONS ARE AVAILABLE:

          1.  MAKE METAFILE OF PREVIOUSLY SAVED CURVE.
          2.  QUIT.

               ENTER 1 OR 2
?
1


          WHAT FILE NAME IS THE DATA STORED UNDER?
openloop

THE CURVE DATA IS BEING LOADED FROM FILE OPENLOOP DATA
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.

          THE FOLLOWING OPTIONS ARE AVAILABLE:

          1.  MAKE METAFILE OF PREVIOUSLY SAVED CURVE.
          2.  QUIT.

               ENTER 1 OR 2
?
2
END OF DISSPLA 9.0 -- 13197 VECTORS GENERATED IN 1 PLOT FRAMES
    PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CA.
1817 VIRTUAL STORAGE REFERENCES; 5 READS; 0 WRITES.
DASD 121 DETACHED
DASD 122 DETACHED
DASD 120 DETACHED

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

     DO YOU WANT A VRSTEC PLOTTER SMOOTH COPY OF THE
       THE DISSPLA METAFILE THAT YOU JUST CREATED?
                        (Y OR N)

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

y
B (120) R/O
DASD 001 LINKED R/O; R/W BY MVS
Z (001) R/O - OS
DASD 001 DETACHED
CREATING NEW FILE:
CREATING NEW FILE:
PUN FILE 6680  TO  MVS      COPY 001   NOHOLD
DASD 120 DETACHED

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

     YOUR GRAPH(S) CAN BE PICKED UP AT THE COMPUTER CENTER.

       THE GRAPH(S) WILL BE ADDRESSED TO "POP (USER ID)".

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


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

           DO YOU WANT TO

     1. RUN OPTSYSX AGAIN
     2. RUN THE PLOT PROGRAM USING THE SAME MATRICES?
```

```
     (TO PLOT ANOTHER TYPE OF SYSTEM (OPEN/CLOSED))
   3. QUIT

                    ENTER 1, 2 OR 3

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

3

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

                    HAVE A GOOD DAY!!

****************************************************************
R; T=11.05/17.26 20:08:15
record off
END RECORDING OF TERMINAL SESSION
```

The graphical output generated by this example follows as figure 3.1.

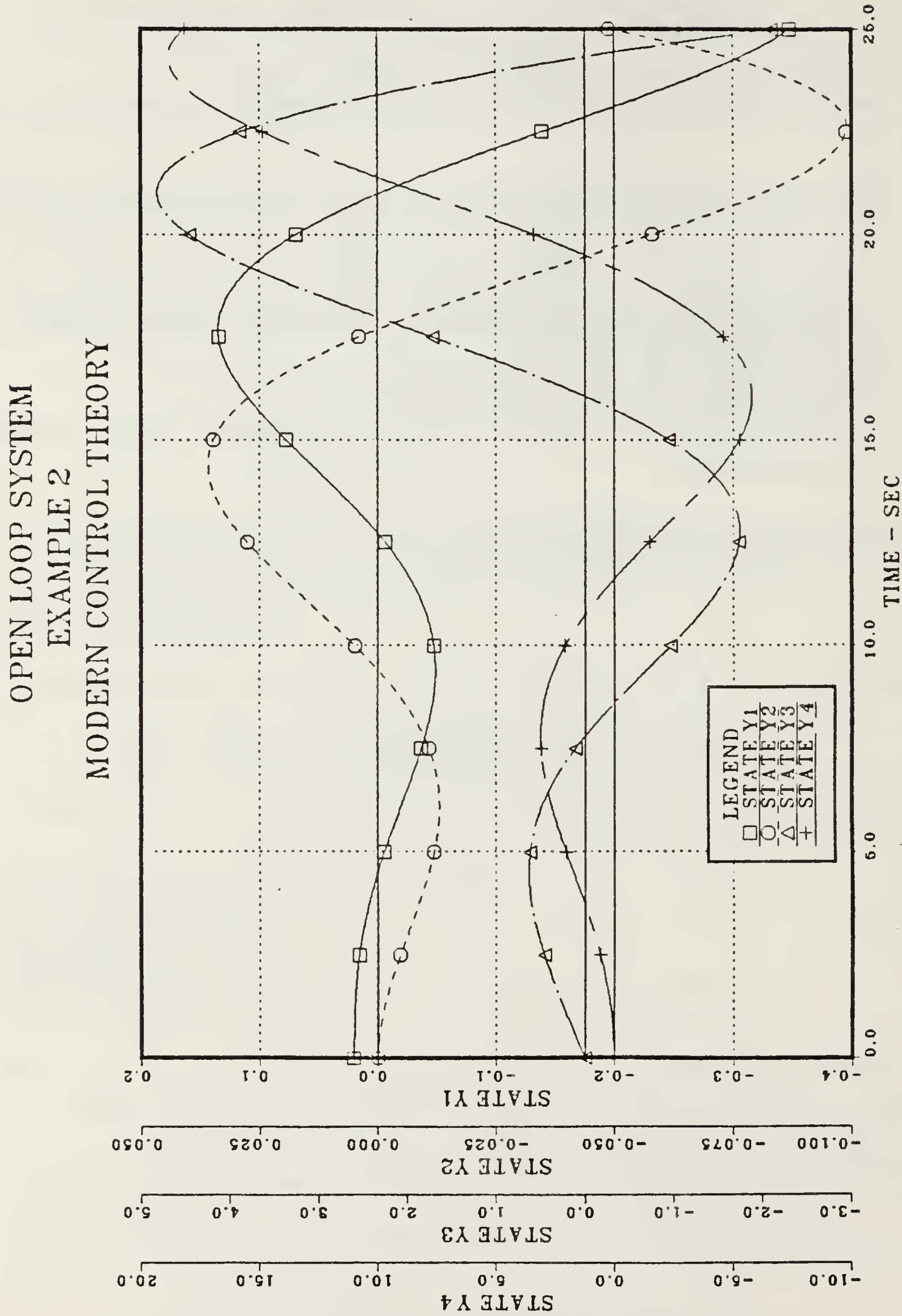

Figure 3.1 Open-loop Time Response

B. CLOSED-LOOP SYSTEM TIME RESPONSE

The following closed-loop system example was taken from [Ref. 6, pp 5.8 - 5.19].

The full terminal session is recorded below, with user input at the left margin in lower case letters or numbers below each "?".

```
record on
BEGIN RECORDING OF TERMINAL SESSION
R; T=0.01/0.02 20:19:44
optsys

      THE OPTSYS EXEC CONTROLS A TRIO OF PROGRAMS:

          1. OPTSYSX FORTRAN   (SYSTEM ANALYSIS)
          2. OPTCALC FORTRAN   (CALCULATE TIME RESPONSE)
          3. OPTPLOT FORTRAN   (DISSPLA PLOTTING ROUTINE)

    EACH PROGRAM PASSES INFORMATION TO THE NEXT PROGRAM
    THROUGH A DATA FILE WRITTEN TO THE USERS DISK.  IN THIS
    CASE, THESE FILES ARE "OPTMAT DATA" AND "OPTPLOT DATA".
    THE SIZE OF THESE FILES VARY WITH THE SYSTEM ORDER, AND
    CAN USE ABOUT 20% OF THE USERS DISK SPACE.  THEREFORE
    ENSURE THAT SUFFICIENT DISK SPACE IS AVAILABLE.

      - TYPE "E" TO EXIT, ANY OTHER ENTRY TO CONTINUE -

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

     YOU HAVE A DATA FILE NAMED 'OPTMAT DATA' ON YOUR A
     DISK  THAT WAS PREVIOUSLY GENERATED BY THE  OPTSYS
     PROGRAM  AND  CONTAINS THE F, G, H, GAMMA, A AND B
     MATRICES FROM THAT RUN.

     IF  YOU WOULD LIKE TO USE THESE SAME MATRICES  FOR
     THIS  RUN, THE  OPTSYS  PROGRAM  WILL  READ IN THE
     DESIRED DATA AT THE APPROPRIATE TIME,

                     IF YOU TYPE (Y)ES.

     ANY OTHER INPUT WILL RESULT IN THAT FILE BEING ERASED!

***************************************************************
y

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

    DO YOU WANT THE NUMERICAL OUTPUT FROM OPTSYSX TO GO
      TO YOUR TERMINAL S(CREEN) OR TO A D(ISK) FILE?
                         (S OR D)

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

s
***************************************************************

          OUTPUT WILL COME TO YOUR TERMINAL SCREEN.
```

```
************************************************************************
LOADING OPTSYS....
EXECUTION BEGINS...
 OPTSYSX IS A COMPLETELY INTERACTIVE OPTIMAL SYSTEMS CONTROL
    PROGRAM. IT WILL SOLVE NUMEROUS CONTROL PROBLEMS ON THE
    FOLLOWING TYPES OF SYSTEMS CONTROL EQUATIONS:

               XDOT = {F}*X + {G}*U + {GAM}*(W+WO)

                    MEASUREMENT EQUATION--

               Z = {H}*X + {D}*U + V

                    REGULATOR PERFORMANCE INDEX--

               J = 1/2 * INTEGRAL (Y *{A}*Y + U *{B}*U) DT

                    STATE FEEDBACK GAIN DEFINITION--

                         U = -{C}*X

               DO YOU WISH TO CONTINUE?  TYPE "YES" OR "NO".
y
                         --DATA ENTRY--

     ALTHOUGH OPTSYSX IS SPECIFICALLY DESIGNED TO READ
     ALL MATRIX DATA INTERACTIVELY, SEVERAL ALTERNATE
     METHODS ARE AVAILABLE TO USERS:

        METHOD 1--THE "F","G", AND "GAMMA" MATRICES
           MAY BE READ FROM SEPARATE DATA FILES.

        METHOD 2--THE "F","G", AND "GAMMA" MATRICES MAY BE
           EXPLICITLY DEFINED WITHIN SUBROUTINE "SETUP".

     {NOTE: IN EITHER CASE, THE USER SHOULD OBTAIN A COPY
            OF THE PROGRAM LISTING AND EXAMINE
            THE EXAMPLES CONTAINED IN S/R "SETUP".}

          DO YOU WISH TO CONTINUE?  TYPE "YES" OR "NO".
y

     DO YOU WISH TO INPUT THE "F", "G", AND "GAMMA"
          MATRICES FROM SUBROUTINE "SETUP" IAW THE
          METHOD DESCRIBED ON THE PREVIOUS SCREEN?

               TYPE "YES" OR "NO".
n
                    GENERAL OPTSYSX OPTIONS:

        OPTION 1 -- SYSTEM ANALYSIS WITHOUT
                    OPEN-LOOP EIGENSYSTEM CALCULATIONS.

        OPTION 2 -- SYSTEM ANALYSIS WITH OPEN-LOOP
                    EIGENSYSTEM CALCULATIONS.

        OPTION 3 -- OPEN-LOOP EIGENSYSTEM FOUND
                    AND PROGRAM TERMINATES.
                    {"F"-MATRIX ENTRY FOLLOWS IMMEDIATELY.}

        OPTION 4 -- MODAL DISTRIBUTION MATRICES COMPUTED
                    WITHOUT FILTER OR REGULATOR SYNTHESIS
                    OR STEADY-STATE ANALYSIS.

               SELECT AN OPTION: 1,2,3, OR 4.
?
```

```
1

     DO YOU DESIRE RMS VALUES OF STATE AND CONTROL?

          TYPE "YES" OR "NO".
n

                     OPTSYSX LQR/CLASSICAL OPTIONS:

         OPTION 1 -- OPTIMAL FILTER AND/OR REGULATOR
                     SYNTHESIS WITH NO EXTERNAL "C" OR "K"
                     MATRIX INPUT.

         OPTION 2 -- OPTIMAL FILTER AND/OR REGULATOR
                     SYNTHESIS WITH EXTERNAL "C"
                     MATRIX INPUT.

         OPTION 3 -- OPTIMAL FILTER AND/OR REGULATOR
                     SYNTHESIS WITH EXTERNAL "K"
                     MATRIX INPUT.

         OPTION 4 -- OPTIMAL FILTER AND/OR REGULATOR
                     SYNTHESIS WITH EXTERNAL "C" AND "K"
                     MATRIX INPUT.

          SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

     DO YOU WISH TO DETERMINE THE STEADY-STATE RESPONSE
        FOR A CONSTANT DISTURBANCE?

          TYPE "YES" OR "NO".
n
     DO YOU WISH TO DETERMINE THE MODAL DISTRIBUTION
        AND GAIN MATRICES?

          TYPE "YES" OR "NO".
n

     OPEN-LOOP TRANSFER FUNCTION OPTIONS:

         OPTION 1 -- NO OPEN-LOOP TRANSFER FUNCTIONS COMPUTED.

         OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.

         OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.

         OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.

         SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

     NOISE TRANSFER FUNCTION OPTIONS:

         OPTION 1 -- NO NOISE TRANSFER FUNCTIONS COMPUTED.

         OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.

         OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.

         OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.

         SELECT AN OPTION: 1, 2, 3, OR 4.
```

```
?
1

     COMPENSATOR TRANSFER FUNCTION OPTIONS:

          OPTION 1 -- NO COMP. TRANSFER FUNCTIONS COMPUTED.

          OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.

          OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.

          OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.

                 {NOTE: A COMPENSATOR TRANSFER FUNCTION CAN BE
                        COMPUTED ONLY IF BOTH A REGULATOR
                        AND FILTER ARE SYNTHESIZED
                        AND/OR INPUT.}

          SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

     WILL A FEED-FORWARD DISTRIBUTION MATRIX
     {"D" - MATRIX} BE INPUT ?

               TYPE "YES" OR "NO".
n

    THIS OPTION DETERMINES THE CRITERIA FOR DECIDING WHEN A
    MARKOV PARAMETER IS ZERO-THE MARKOV PARAMETER INDICATES
    THE ORDER OF THE NUMERATOR POLYNOMIAL OF EACH TRANSFER
    FUNCTION.

      ALL "N" ZEROS OF THIS POLYNOMIAL ARE PRINTED OUT AND
      THIS TEST TELLS HOW MANY EXTRA ROOTS EXIST AT Z = 0.
      LESS THAN 10.0**{-IE} IS CONSIDERED ZERO.

       THE DEFAULT VALUE OF THIS PARAMETER {IE} IS 6.
       IN OTHER WORDS, IE = 1.0E-6.

   IF YOU DESIRE A DIFFERENT MARKOV CRITERIA,
   TYPE THE INTEGER VALUE.

   IF YOU DESIRE THE DEFAULT VALUE, TYPE "0" {ZERO}
?
0

 DO YOU DESIRE TO SYNTHESIZE A STABLE FILTER {OR REGULATOR} BY
      DESTABILIZING THE ORIGINAL SYSTEM?

          {NOTE:WORKS FOR FILTER OR REGULATOR BUT NOT FOR BOTH
                     IN THE SAME RUN.}

           TYPE "YES" OR "NO".
n
     DO YOU DESIRE TO PRINT THE EULER-LAGRANGE EIGENSYSTEM
        PRIOR TO DECOMPOSITION {FOR CHECKING THE PROGRAM}?

           TYPE "YES" OR "NO".
n

  POWER SPECTRAL DENSITY {PSD} OPTION 1 :

  OPTION 1 -- COMPUTE THE PSD OF THE OUTPUTS AND/OR THE
              CONTROLS OF THE CONTROLLED SYSTEM WHEN FORCED BY
```

```
              PROCESS AND MEASUREMENT NOISE.  {NOTE: BOTH A
              REGULATOR AND A FILTER MUST BE RESIDENT IN THE
              PROGRAM TO USE THIS OPTION.}

    OPTION 2 -- SAME AS OPTION 1 ABOVE BUT ONLY PRINT THE
              RESIDUES OF EACH TRANSFER FUNCTION
              USED IN THE PSD COMPUTATION.

    OPTION 3 -- NOT DESIRED.

           SELECT AN OPTION: 1, 2, OR 3.
?
3


       DO YOU DESIRE REGULATOR SYNTHESIS ONLY?

           TYPE "YES" OR "NO".

y


           THE "F", "G", "H", "GAM", "A" AND "B" MATRICES
              FROM YOUR PREVIOUS OPTSYS RUN WERE SAVED.

           THE FOLLOWING OPTIONS ARE AVAILABLE:
                1. USE ALL OF THE SAME MATRICES AGAIN.
                2. USE SELECTED MATRICES AGAIN.
                3. INPUT ALL NEW MATRICES.

           ENTER 1, 2, OR 3.

           NOTE: EACH SAVED MATRIX WILL BE REDISPLAYED AT
           THE PROPER INPUT SEQUENCE INTERVAL
           AND YOU WILL HAVE THE OPTION OF CHANGING
           INDIVIDUAL MATRIX ELEMENTS.
?
2


       DO YOU WISH TO SAVE THE "F"-MATRIX FROM THE LAST
       RUN TO BE USED IN THIS RUN?

       NOTE: THE MATRIX WILL BE REDISPLAYED AT
       THE PROPER INPUT SEQUENCE INTERVAL
       AND YOU WILL HAVE THE OPTION OF CHANGING
       INDIVIDUAL MATRIX ELEMENTS.

                TYPE "YES" OR "NO".
y


       DO YOU WISH TO SAVE THE "A"-MATRIX FROM THE LAST
       RUN TO BE USED IN THIS RUN?

       NOTE: THE MATRIX WILL BE REDISPLAYED AT
       THE PROPER INPUT SEQUENCE INTERVAL
       AND YOU WILL HAVE THE OPTION OF CHANGING
       INDIVIDUAL MATRIX ELEMENTS.

                TYPE "YES" OR "NO".
n


       DO YOU WISH TO SAVE THE "B"-MATRIX FROM THE LAST
       RUN TO BE USED IN THIS RUN?
```

```
     NCIE: THE MATRIX WILL BE REDISPLAYED AT
     THE PROPER INPUT SEQUENCE INTERVAL
     AND YOU WILL HAVE THE OPTION OF CHANGING
     INDIVIDUAL MATRIX ELEMENTS.

              TYPE "YES" OR "NO".
n

   ENTER THE # OF CONTROLS {NC} OF THE CONTROL SYSTEM MODEL
   {"G"-MATRIX}.
?
1

     ENTER THE # OF MEASUREMENTS OR OBSERVATIONS {NO} OF THE
     {"H"-MATRIX}.
?
4

     ENTER THE # OF PROCESS NCISE SOURCES {NG} OF THE
     {"GAMMA"-MATRIX}.
?
0
    FLAG/PARAMETER SETTINGS FOR THIS RUN ARE AS FOLLOWS:
IOL   IQ   IR   ISS   IM   ITF1   ITF2   ITF3   IFDFW   IE   IDEBUG
 0     0    0    0     0    0      0      0      0       0     0
ISET   IDSTAB  IPSD   IYU   INORM   IREG   NS   NC   NOB   NG
  0      0      0      0      0       1     4    1    4    0

 ORDER OF SYSTEM =   4
 NUMBER OF CONTROLS =  1
 NUMBER OF CBSERVATICNS =  4
 NUMBER OF PROCESS NCISE SOURCES =  0

              THE SYSTEM MATRIX {"F"-MATRIX}...

     0.0            1.00000     0.0          0.0
     0.0           -0.41500    -0.01110      0.0
     9.80000       -1.43000    -0.01980      0.0
     0.0            0.0         1.00000      0.0

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" CR "NO".
n

    OPEN LOOP DYNAMICS MATRIX.................F..

  0.0           0.1000D+01   0.0           0.0
  0.0          -0.415CD+00  -0.1110D-01    0.0
  0.98C0D+01   -0.1430D+01  -0.1980D-01    0.0
  0.0           0.0          0.1000D+01    0.0
     ENTER THE MEASUREMENT SCALING MATRIX {"H"-MATRIX}.
          DIMENSION = # OBSERVATIONS {NO} X # STATES {NS}
     THE ELEMENT H( 1, 1)=
?
1
```

```
      THE ELEMENT H( 1, 2)=
?
0
      THE ELEMENT H( 1, 3)=
?
0
      THE ELEMENT H( 1, 4)=
?
0
      THE ELEMENT H( 2, 1)=
?
0
      THE ELEMENT H( 2, 2)=
?
1
      THE ELEMENT H( 2, 3)=
?
0
      THE ELEMENT H( 2, 4)=
?
0
      THE ELEMENT H( 3, 1)=
?
0
      THE ELEMENT H( 3, 2)=
?
0
      THE ELEMENT H( 3, 3)=
?
1
      THE ELEMENT H( 3, 4)=
?
0
      THE ELEMENT H( 4, 1)=
?
0
      THE ELEMENT H( 4, 2)=
?
0
      THE ELEMENT H( 4, 3)=
?
0
      THE ELEMENT H( 4, 4)=
?
1

          THE MEASUREMENT SCALING MATRIX {"H"-MATRIX}...

      1.00000      0.0          0.0          0.0
      0.0          1.00000      0.0          0.0
      0.0          0.0          1.00000      0.0
      0.0          0.0          0.0          1.00000

      DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" OR "NO".
n

     MEASUREMENT SCALING MATRIX.................H..

   0.1000D+01   0.0          0.0          0.0
   0.0          0.1000D+01   0.0          0.0
   0.0          0.0          0.1000D+01   0.0
   0.0          0.0          0.0          0.1000D+01
```

```
     ENTER THE OUTPUT MEASUREMENT COST MATRIX {"A"-MATRIX}.
     DIMENSION = # OBSERVATIONS {NO} X # OBSERVATIONS {NO}
     THE ELEMENT A( 1, 1)=
?
0
     THE ELEMENT A( 1, 2)=
?
0
     THE ELEMENT A( 1, 3)=
?
0
     THE ELEMENT A( 1, 4)=
?
0
     THE ELEMENT A( 2, 1)=
?
0
     THE ELEMENT A( 2, 2)=
?
0
     THE ELEMENT A( 2, 3)=
?
0
     THE ELEMENT A( 2, 4)=
?
0
     THE ELEMENT A( 3, 1)=
?
0
     THE ELEMENT A( 3, 2)=
?
0
     THE ELEMENT A( 3, 3)=
?
0
     THE ELEMENT A( 3, 4)=
?
0
     THE ELEMENT A( 4, 1)=
?
0
     THE ELEMENT A( 4, 2)=
?
0
     THE ELEMENT A( 4, 3)=
?
0
     THE ELEMENT A( 4, 4)=
?
0.25

     THE OUTPUT MEASUREMENT COST MATRIX {"A"-MATRIX}...

     0.0         0.0         0.0         0.0
     0.0         0.0         0.0         0.0
     0.0         0.0         0.0         0.0
     0.0         0.0         0.0         0.25000

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" OR "NO".
n

    OUTPUT COST MATRIX.........................A..
  0.0          0.0          0.0          0.0
```

```
  0.0           0.0           0.0           0.0
  0.0           0.0           0.0           0.0
  0.0           0.0           0.0           0.2500D+00

     ENTER THE CONTROL DISTRIBUTION MATRIX {"G"-MATRIX}.

          DIMENSION = # STATES {NS} X # CONTROLS {NC}
     THE ELEMENT G( 1, 1)=
?
0

     THE ELEMENT G( 2, 1)=
?
6.27
     THE ELEMENT G( 3, 1)=
?
9.8
     THE ELEMENT G( 4, 1)=
?
0


          THE CONTROL DISTRIBUTION MATRIX {"G"-MATRIX}...

     0.0
     6.27000
     9.80000
     0.0


     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?

          TYPE "YES" OR "NO".
n

     ENTER THE CONTROL COST WEIGHTING MATRIX {"B"-MATRIX}
          DIMENSION = # CONTROLS {NC} X # CONTROLS {NC}
     THE ELEMENT B( 1, 1)=
?
131.3

          THE CONTROL COST MATRIX..........B...

   131.30000

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?

          TYPE "YES" OR "NO".
n

    THE CONTROL DISTRIBUTION MATRIX...........G..

  0.0
  0.6270D+01
  0.9800D+01
  0.0

    THE CONTROL COST MATRIX...................B..

  0.1313D+03

    EIGENSYSTEM OF OPTIMAL REGULATOR.........

    C-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G*C)..
```

```
-1.23385D+00, 5.54546D-01:-4.19835D-01, 1.13532D+00:

    C-LOOP RIGHT EIGENVECTOR MATRIX..........M....

-1.019344D-01  2.308717D-02 -8.155484D-02  1.122264D-01
 1.129691D-01 -8.501340D-02 -9.317336D-02 -1.397074D-01
 1.000000D+00  0.0           1.000000D+00  0.0
-6.742684D-01 -3.030447D-01 -2.865351D-01 -7.748499D-01

    CONTROL EIGENVECTOR MATRIX..............C*M..

   -5.464314D-03  2.109409D-02  2.713925D-02 -1.676334D-02

    C-LOOP OPT. REG. LEFT E-VECTOR MATRIX..M-INV..

-3.764753D+00  2.578703D+00 -3.562309D-01 -1.010220D+00
-3.421605D+01 -9.486653D+00 -4.604269D+00 -3.245261D+00
 3.764753D+00 -2.578703D+00  1.356231D+00  1.010220D+00
 1.526581D+01  2.419863D+00  1.609198D+00  4.841548D-01

 THE OPTIMAL FEEDBACK GAIN CONTROL MATRIX...C=BINV*GT*S...

 -8.5492D-01  -3.2475D-01  -8.5345D-02  -4.3635D-02

    THE CLOSED LOOP DYNAMICS MATRIX ......F-G*C..

 0.0           1.000000D+00  0.0           0.0
-5.360337D+00 -2.451197D+00 -5.462116D-01 -2.735931D-01
 1.421803D+00 -4.612572D+00 -8.561786D-01 -4.276256D-01
 0.0           0.0           1.000000D+00  0.0

            DO YOU WISH TO OBTAIN A TIME RESPONSE
              OF THE SYSTEM YOU ARE EVALUATING?
                          (Y OR N)

      NOTE: YOU MUST BE LOGGED ON AT A DUAL SCREEN
           (TEK 618) TERMINAL TO UTILIZE THIS MODE.

   THE F (SYSTEM), G (CONTROL), H (OBSERVABLES), GAM (NOISE),
      A (OUTPUT COST) AND B (CONTROL COST) MATRICES WILL BE
         SAVED FOR REENTRY TO THE MAIN OPTSYS PROGRAM.
y

*************************************************************
          IF YOU ARE DISSATISFIED WITH THE RESULTS
          THUS FAR AND WOULD LIKE TO EXIT TO CMS,

                 -TYPE 'Y' TO EXIT-

              (ANY OTHER INPUT TO CONTINUE)
*************************************************************

LOADING OPTCALC....
EXECUTION BEGINS...

      DURING THIS SECTION OF THE PROGRAM YOU WILL:

   - SELECT THE TYPE OF SYSTEM RESPONSE TO PLOT
     (OPEN LOOP, CLOSED LOOP, OR FILTER/REGULATOR)
   - PROVIDE START AND STOP TIME FOR PLOTTING CALCULATIONS
```

```
- SELECT THE TYPE OF DRIVING FUNCTION(S) (STEP OR RAMP)
- PROVIDE START AND STOP TIMES FOR THE DRIVING FUNCTION(S)
- PROVIDE DRIVING FUNCTION MAGNITUDE(S).

          CLEAR THE SCREEN TO CONTINUE

               THE F MATRIX

 0.0          1.00000      0.0          0.0
 0.0         -0.41500     -0.01110      0.0
 9.80000     -1.43000     -0.01980      0.0
 0.0          0.0          1.00000      0.0

               THE G MATRIX

 0.0
 6.27000
 9.80000
 0.0

               THE C MATRIX

-0.85492     -0.32475     -0.08534     -0.04364

               THE H MATRIX

 1.00000      0.0          0.0          0.0
 0.0          1.00000      0.0          0.0
 0.0          0.0          1.00000      0.0
 0.0          0.0          0.0          1.00000

               THE K MATRIX

 0.0          0.0          0.0          0.0
 0.0          0.0          0.0          0.0
 0.0          0.0          0.0          0.0
 0.0          0.0          0.0          0.0

THE FOLLOWING PLOTTING OPTIONS ARE AVAILABLE IF THE
REQUIRED MATRICES WERE CALCULATED IN OPTSYSX:

1.  OPEN LOOP TIME RESPONSE
     XDOT = {F}*X + {G}*UC

2.  CLOSED LOOP TIME RESPONSE
     XDOT = {F-G*C}*X + {G}*UC,   U =  {C}*X

3.  OPTIMIZED FILTER CLOSED LOOP SYSTEM RESPONSE.
     XDOT = {F}*X + {G}*UC,        Z = {H}*X
     XHDOT =  {F}*XH +{G}U  +  {K}*{Z - H*XH}

4.  FILTER + REGULATOR CLOSED LOOP SYSTEM RESPONSE.
     XDOT = {F+G*C}*X + {G}*UC,              Z  =  {H}*X
     XHDOT =  {F}*XH +{G}U  +  {K}*{Z - H*XH},   U  =  {C}*XH

        SELECT 1, 2, 3 OR 4.
?
2

               THE AUGMENTED F MATRIX  (F+G*C)

 0.0          1.00000      0.0          0.0
-5.36034     -2.45120     -0.54621     -0.27359
```

```
     1.42180     -4.61257     -0.85618     -0.42763
     0.0          0.0          1.00000      0.0

          AT WHAT TIME DO YOU WANT TO START
          THE TIME RESPONSE CALCULATIONS?

          INPUT START TIME IN SECONDS. (NORMALLY 0.0)
?
0

          AT WHAT TIME DO YOU WANT TO STOP
          THE TIME RESPONSE CALCULATIONS?

            INPUT STOP TIME IN SECONDS.
?
25

     THIS PROGRAM DIVIDES THE TIME INTERVAL YOU HAVE
     JUST SPECIFIED INTO UP TO 500 SMALL INTERVALS FOR
     THE INTEGRATION AND PLOTTING ROUTINES.  IN ORDER
     TO SAVE COMPUTER TIME, THE NUMBER OF POINTS CAN BE
     CAN BE REDUCED WITH SOME LOSS IN CURVE FIDELITY.

          HOW MANY POINTS DO YOU WANT TO CALCULATE?
?
500
 DOES THE SYSTEM UTIIIZE A DRIVING FUNCTION (CONTROL INPUT)?
                    (Y)ES OR (N)O
n

   DOES THE SYSTEM START WITH ALL INITIAL CONDITIONS = 0.0 ?
                    (Y)ES OR (N)O?
n
          WHAT IS THE INITIAL CONDITION FOR X( 1)  ?
?
0.02
          WHAT IS THE INITIAL CONDITION FOR X( 2)  ?
?
0
          WHAT IS THE INITIAL CONDITION FOR X( 3)  ?
?
0
          WHAT IS THE INITIAL CONDITION FOR X( 4)  ?
?
0

          THIS IS YOUR LAST OPPORTUNITY TO
          MAKE CHANGES IN THE FOLLOWING AREAS.

            1.   SELECT ANOTHER TYPE OF SYSTEM TO PLOT
                 (OPEN, CLOSED, FILTER OR FILTER/REGULATOR)

            2.   START AND STOP TIMES

            3.   DRIVING FUNCTIONS

            4.   INITIAL CONDITIONS
```

5. CONTINUE

SELECT A NUMBER BETWEEN 1 AND 5.

?
5

THE FOLLOWING INFORMATION IS PROVIDED ONLY
FOR AN INDICATION OF PROPER PROGRAM OPERATION.

ALL CONTROLS, STATES AND STATE ESTIMATES CAN BE PLOTTED.

TIME	U(1)	X(1)	X(2)	X(3)
0.0	0.0	0.2000000D-01	0.0	0.0
0.50	0.0	0.1103108D-01	-0.2835004D-01	0.4438230D-01
1.00	0.0	-0.3124039D-02	-0.2499014D-01	0.7923124D-01
1.50	0.0	-0.1214295D-01	-0.1044492D-01	0.6850392D-01
2.00	0.0	-0.1372004D-01	0.3415974D-02	0.2513011D-01
2.50	0.0	-0.9736274D-02	0.1137285D-01	-0.2463973D-01
3.00	0.0	-0.3455204D-02	0.1276482D-01	-0.5991416D-01
3.50	0.0	0.2242363D-02	0.9465182D-02	-0.7172835D-01
4.00	0.0	0.5680882D-02	0.4176801D-02	-0.6209710D-01
4.50	0.0	0.6485095D-02	-0.7495987D-03	-0.3954511D-01
5.00	0.0	0.5231722D-02	-0.3909040D-02	-0.1411885D-01
5.50	0.0	0.2937493D-02	-0.4926279D-02	0.6308558D-02
6.00	0.0	0.5976676D-03	-0.4202851D-02	0.1776240D-01
6.50	0.0	-0.1106274D-02	-0.2524967D-02	0.2010080D-01
7.00	0.0	-0.1903554D-02	-0.6964301D-03	0.1580986D-01
7.50	0.0	-0.1875704D-02	0.7052577D-03	0.8417729D-02
8.00	0.0	-0.1314153D-02	0.1421627D-02	0.1131661D-02
8.50	0.0	-0.5646948D-03	0.1480370D-02	-0.3974507D-02
9.00	0.0	0.9105125D-04	0.1089964D-02	-0.6170345D-02
9.50	0.0	0.4951828D-03	0.5175573D-03	-0.5819287D-02
10.00	0.0	0.6168727D-03	-0.8506241D-05	-0.3898143D-02
10.50	0.0	0.5177499D-03	-0.3515887D-03	-0.1503636D-02
11.00	0.0	0.3023367D-03	-0.4749532D-03	0.5066495D-03
11.50	0.0	0.7303314D-04	-0.4181781D-03	0.1680906D-02
12.00	0.0	-0.9885395D-04	-0.2595192D-03	0.1964939D-02
12.50	0.0	-0.1831320D-03	-0.7995128D-04	0.1581498D-02
13.00	0.0	-0.1853826D-03	0.6125230D-04	0.8706565D-03
13.50	0.0	-0.1330746D-03	0.1362740D-03	0.1509012D-03
14.00	0.0	-0.6008964D-04	0.1460367D-03	-0.3658826D-03
14.50	0.0	0.5307624D-05	0.1100814D-03	-0.5993413D-03
15.00	0.0	0.4674427D-04	0.5451421D-04	-0.5788739D-03
15.50	0.0	0.6041264D-04	0.2181162D-05	-0.3974740D-03
16.00	0.0	0.5186717D-04	-0.3284278D-04	-0.1629461D-03
16.50	0.0	0.3120303D-04	-0.4633021D-04	0.3845110D-04
17.00	0.0	0.8570407D-05	-0.4175156D-04	0.1596441D-03
17.50	0.0	-0.8789315D-05	-0.2662910D-04	0.1929683D-03
18.00	0.0	-0.1764726D-04	-0.8965253D-05	0.1588805D-03
18.50	0.0	-0.1834821D-04	0.5243773D-05	0.9038270D-04
19.00	0.0	-0.1348538D-04	0.1305990D-04	0.1914584D-04
19.50	0.0	-0.6377176D-05	0.1440718D-04	-0.3322273D-04
20.00	0.0	0.1442643D-06	0.1111254D-04	-0.5799797D-04
20.50	0.0	0.4388011D-05	0.5723306D-05	-0.5745193D-04
21.00	0.0	0.5903699D-05	0.5212149D-06	-0.4042431D-04
21.50	0.0	0.5186879D-05	-0.3048585D-05	-0.1750608D-04
22.00	0.0	0.3210931D-05	-0.4510007D-05	0.2633876D-05
22.50	0.0	0.9808265D-06	-0.4161888D-05	0.1510481D-04
23.00	0.0	-0.7692717D-06	-0.2725618D-05	0.1891776D-04
23.50	0.0	-0.1696169D-05	-0.9909712D-06	0.1593480D-04
24.00	0.0	-0.1813170D-05	0.4364708D-06	0.9351335D-05
24.50	0.0	-0.1364003D-05	0.1247888D-05	0.2311875D-05
25.00	0.0	-0.6730920D-06	0.1419068D-05	-0.2984895D-05

****

```
          IF YOU ARE DISSATISFIED WITH THE RESULTS
          THUS FAR AND WOULD LIKE TO EXIT TO CMS,

                 -TYPE 'Y' TO EXIT-

             (ANY OTHER INPUT TO CONTINUE)

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

B (120) R/O
C (121) R/O
E (122) R/O


          ... Your Fortran program is now being loaded ...
                ... execution will soon follow ...
EXECUTION BEGINS...


          THIS PORTION OF THE PROGRAM PLOTS:
               - THE STATES,
               - EXTERNAL CONTROL INPUTS,
               - FEEDBACK CONTROL INPUTS,
               - STATE ESTIMATES AND
               - RECONSTRUCTION ERRORS
          FROM THE DATA THAT YOU JUST CALCULATED.

          THE CAPABILITY IS ALSO AVAILABLE TO REVIEW ANY
          GRAPHS THAT YOU HAD PREVIOUSLY SAVED AS DATA
          FILES ON YOUR DISK.

               CLEAR THE SCREEN TO CONTINUE.

          THE FOLLOWING OPTIONS ARE AVAILABLE:

               1. PLOT THE DATA YOU JUST CALCULATED.
               2. PLOT A CURVE THAT YOU PREVIOUSLY SAVED.

          ENTER 1 OR 2
?
1


          YOU MAY PLOT UP TO 4 SYSTEM VARIABLES VS TIME.
HOW MANY VARIABLES DO YOU WISH TO PLOT?
?
4


 WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 1?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC.)

          ENTER 1,2,3,4 OR 5
?
1


          WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
          YOU WANT TO PLOT AS THE NUMBER  1 CURVE VS TIME?
?
1
```

```
     WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

     NOTE:   1. 40 CHARACTERS MAX LENGTH
             2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
                ENCLOSED IN PARENTHESES.
                IE. (A) => ALPHA
                    (B) => BETA
                    (F) => PHI
                    (Q) => THETA
state y1

 WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 2?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC)

          ENTER 1,2,3,4 OR 5
?
1

          WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
          YOU WANT TO PLOT AS THE NUMBER  2 CURVE VS TIME?
?
2

     WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

     NOTE:   1. 40 CHARACTERS MAX LENGTH
             2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
                ENCLOSED IN PARENTHESES.
                IE. (A) => ALPHA
                    (B) => BETA
                    (F) => PHI
                    (Q) => THETA
state y2

 WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 3?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC)

          ENTER 1,2,3,4 OR 5
?
1

          WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
          YOU WANT TO PLOT AS THE NUMBER  3 CURVE VS TIME?
?
3

     WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

     NOTE:   1. 40 CHARACTERS MAX LENGTH
             2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
```

```
                    ENCLOSED IN PARENTHESES.
                    IE.  (A) => ALPHA
                         (B) => BETA
                         (F) => PHI
                         (Q) => THETA
state y3

 WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 4?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC)

          ENTER 1,2,3,4 OR 5
?
1

          WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
          YOU WANT TO PLOT AS THE NUMBER  4 CURVE VS TIME?
?
4

     WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE.  (A) => ALPHA
                    (B) => BETA
                    (F) => PHI
                    (Q) => THETA
state y4

          YOU MAY USE UP TO 3 HEADINGS.
          HOW MANY HEADINGS DO YOU DESIRE ON THIS GRAPH?

                    0, 1, 2 OR 3
?
3

     WHAT IS THE DESIRED HEADING NUMBER 1?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE.  (A) => ALPHA
                    (B) => BETA
                    (F) => PHI
                    (Q) => THETA
closed loop system

     WHAT IS THE DESIRED HEADING NUMBER 2?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE.  (A) => ALPHA
                    (B) => BETA
                    (F) => PHI
                    (Q) => THETA
example 3
```

```
      WHAT IS THE DESIRED HEADING NUMBER 3?

      NOTE:  1. 40 CHARACTERS MAX LENGTH
             2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
                ENCLOSED IN PARENTHESES.
                IE.  (A) => ALPHA
                     (B) => BETA
                     (F) => PHI
                     (Q) => THETA
modern control theory
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.

   THE FOLLOWING OPTIONS ARE AVAILABLE.

   1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
   2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
   3. EDIT THE CURRENT GRAPH.
   4. PLOT REVISED GRAPH ON THE TEK618.
   5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
      PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
5

      DO YOU WANT TO SAVE THE CURRENT GRAPH DATA TO
      BE USED LATER TO GENERATE A METAFILE?

                  Y OR N

      NOTE: A METAFILE IS REQUIRED FOR SMOOTH VERSATEC PLOTS.
      THERE WILL BE AN OPPORTUNITY TO GENERATE A METAFILE
      JUST BEFORE EXITING THIS PROGRAM.
y

   WHAT FILE NAME DO YOU WANT THE CURVE DATA STORED UNDER?
                       (8 CHARACTERS MAX)
closedlp

           THE CURVE DATA IS BEING FILED UNDER CLOSEDLP DATA
END OF DISSPLA 9.0 -- 16300 VECTORS GENERATED IN 1 PLOT FRAMES
    PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CA.
1888 VIRTUAL STORAGE REFERENCES; 6 READS; 0 WRITES.

           THE FOLLOWING OPTIONS ARE AVAILABLE:

           1.  MAKE METAFILE OF PREVIOUSLY SAVED CURVE.
           2.  QUIT.

                ENTER 1 OR 2
?
1

           WHAT FILE NAME IS THE DATA STORED UNDER?
closedlp

THE CURVE DATA IS BEING LOADED FROM FILE CLOSEDLP DATA
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.

           THE FOLLOWING OPTIONS ARE AVAILABLE:

           1.  MAKE METAFILE OF PREVIOUSLY SAVED CURVE.
           2.  QUIT.

                ENTER 1 OR 2
```

```
?
2
END CF DISSPLA 9.0 -- 16260 VECTORS GENERATED IN 1 PLOT FRAMES
    PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CA.
1886 VIRTUAL STORAGE REFERENCES; 5 READS; 0 WRITES.
DASD 121 DETACHED
DASD 122 DETACHED
DASD 120 DETACHED

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

       DO YOU WANT A VRSTEC PLOTTER SMOOTH COPY OF THE
         THE DISSPLA METAFILE THAT YOU JUST CREATED?
                          (Y OR N)

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

y
B (120) R/O
DASD 001 LINKED R/O; R/W BY MVS; R/O BY 0085P
Z (001) R/C - OS
DASD 001 DETACHED
CREATING NEW FILE:
CREATING NEW FILE:
PUN FILE 6749  TO  MVS      COPY 001   NOHOLD
DASD 120 DETACHED

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

    YOUR GRAPH(S) CAN BE PICKED UP AT THE COMPUTER CENTER.

      THE GRAPH(S) WILL BE ADDRESSED TO "POP (USER ID)".

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

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

             DO YOU WANT TO

        1. RUN OPTSYSX AGAIN
        2. RUN THE PLCT PROGRAM USING THE SAME MATRICES?
          (TO PLOT ANOTHER TYPE OF SYSTEM (OPEN/CLOSED))
        3. QUIT

                     ENTER 1, 2 OR 3

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

3

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

                   HAVE A GOOD DAY!!

****************************************************************
R; T=9.59/15.78 20:35:04
record off
END RECORDING OF TERMINAL SESSION
```

The graphical output generated by this example follcws as figure 3.2.

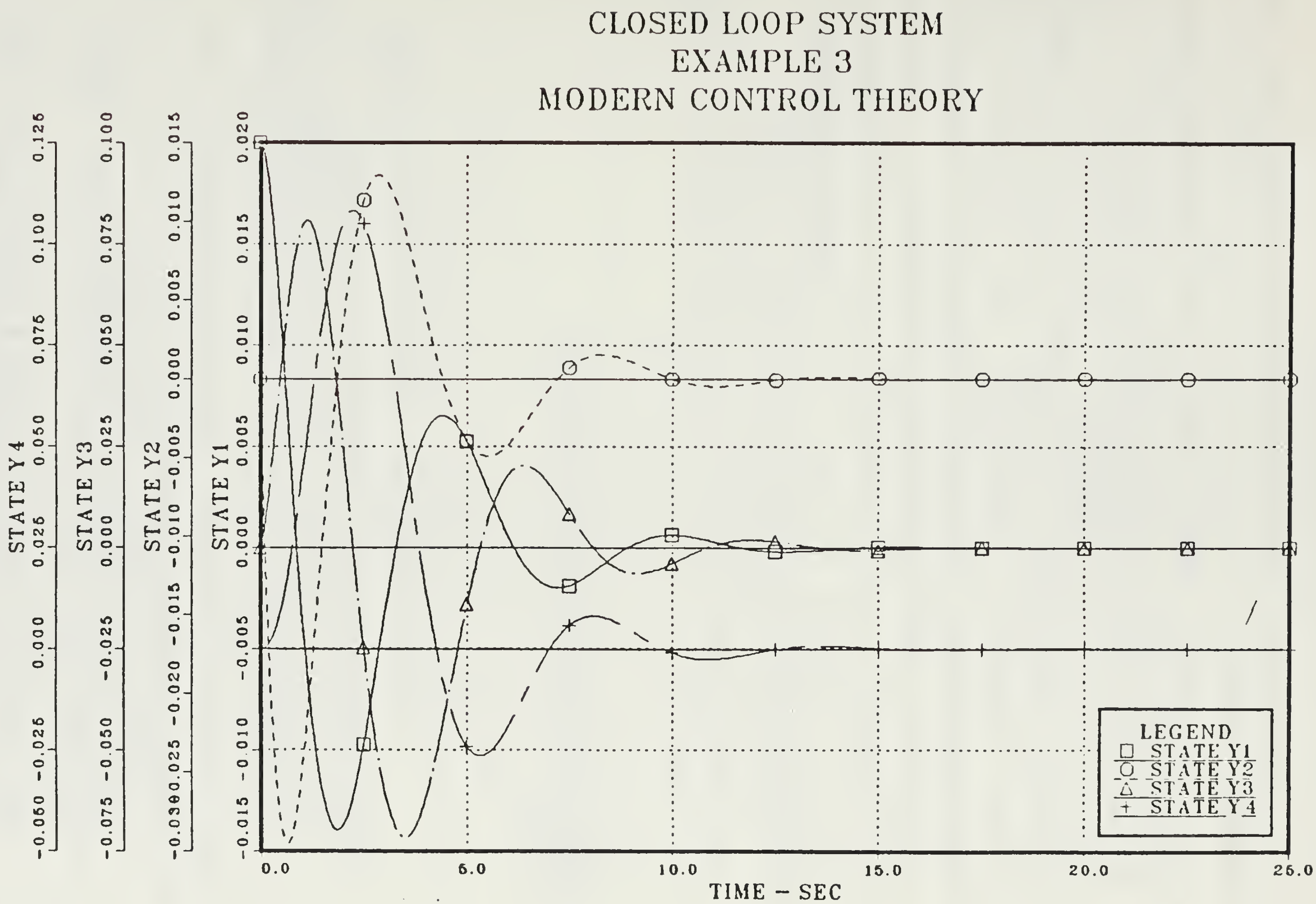

Figure 3.2 Closed-loop Time Response

C. FILTER CLOSED-LOOP SIMULATION

The following filter simulation was taken from [Ref. 7 pp. 332 - 334].

In its present configuration, OPTSYSX program sequencing requires the input of a [C] matrix or design of an optimal regulator (if a [G] matrix has been provided), prior to initiating the optimal estimator synthesis or user provided |K| matrix evaluation. In order to comply with built-in program sequencing conventions, and circumvent program difficulties which may not be specified in the particular system model, optimal filter synthesis may be accomplished by entering the identity matrix [I] in those program input sequences requiring the entry of an output cost (weighting) matrix.

The full terminal session is recorded below, with user input at the left margin in <u>lower case letters</u> or <u>numbers</u> below each "?".

```
record on
BEGIN RECORDING OF TERMINAL SESSION
R; T=0.01/0.02 20:55:40
optsys

     THE OPTSYS EXEC CONTROLS A TRIO OF PROGRAMS:

          1. OPTSYSX FORTRAN   (SYSTEM ANALYSIS)
          2. OPTCALC FORTRAN   (CALCULATE TIME RESPONSE)
          3. OPTPLOT FORTRAN   (DISSPLA PLOTTING ROUTINE)

    EACH PROGRAM PASSES INFORMATION TO THE NEXT PROGRAM
    THROUGH A DATA FILE WRITTEN TO THE USERS DISK.  IN THIS
    CASE, THESE FILES ARE "OPTMAT DATA" AND "OPTPLOT DATA".
    THE SIZE OF THESE FILES VARY WITH THE SYSTEM ORDER, AND
    CAN USE ABOUT 20% OF THE USERS DISK SPACE.  THEREFORE
    ENSURE THAT SUFFICIENT DISK SPACE IS AVAILABLE.

        - TYPE "E" TO EXIT, ANY OTHER ENTRY TO CONTINUE -

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

     YOU HAVE A DATA FILE NAMED 'OPTMAT DATA' ON YOUR A
     DISK  THAT WAS PREVIOUSLY GENERATED BY THE  OPTSYS
     PROGRAM  AND  CONTAINS THE F, G, H, GAMMA, A AND B
     MATRICES FROM THAT RUN.

     IF  YOU WOULD LIKE TO USE THESE SAME MATRICES  FOR
     THIS  RUN, THE  OPTSYS  PROGRAM  WILL  READ IN THE
     DESIRED DATA AT THE APPROPRIATE TIME,
```

```
                    IF YOU TYPE (Y)ES.

      ANY OTHER INPUT WILL RESULT IN THAT FILE BEING ERASED!

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


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

     DO YOU WANT THE NUMERICAL OUTPUT FROM OPTSYSX TO GO
       TO YOUR TERMINAL S(CREEN) OR TO A D(ISK) FILE?
                        (S OR D)

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

s


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

          OUTPUT WILL COME TO YOUR TERMINAL SCREEN.

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

LOADING OPTSYS....
EXECUTION BEGINS...
  OPTSYSX IS A COMPLETELY INTERACTIVE OPTIMAL SYSTEMS CONTROL
     PROGRAM. IT WILL SOLVE NUMEROUS CONTROL PROBLEMS ON THE
     FOLLOWING TYPES OF SYSTEMS CONTROL EQUATIONS:

               XDOT = {F}*X + {G}*U + {GAM}*(W+W0)

                    MEASUREMENT EQUATION--

               Z = {H}*X + {D}*U + V

                    REGULATOR PERFORMANCE INDEX--

               J = 1/2 * INTEGRAL (Y *{A}*Y + U *{B}*U) DT

                    STATE FEEDBACK GAIN DEFINITION--

                         U = -{C}*X

               DO YOU WISH TO CONTINUE?  TYPE "YES" OR "NO".
y
                         --DATA ENTRY--

     ALTHOUGH OPTSYSX IS SPECIFICALLY DESIGNED TO READ
     ALL MATRIX DATA INTERACTIVELY, SEVERAL ALTERNATE
     METHODS ARE AVAILABLE TO USERS:

       METHOD 1--THE "F","G", AND "GAMMA" MATRICES
          MAY BE READ FROM SEPARATE DATA FILES.

       METHOD 2--THE "F","G", AND "GAMMA" MATRICES MAY BE
          EXPLICITLY DEFINED WITHIN SUBROUTINE "SETUP".

       {NOTE: IN EITHER CASE, THE USER SHOULD OBTAIN A COPY
              OF THE PROGRAM LISTING AND EXAMINE
              THE EXAMPLES CONTAINED IN S/R "SETUP".}

          DO YOU WISH TO CONTINUE?  TYPE "YES" OR "NO".
y


     DO YOU WISH TO INPUT THE "F", "G", AND "GAMMA"
          MATRICES FROM SUBROUTINE "SETUP" IAW THE
```

```
            METHOD DESCRIBED ON THE PREVIOUS SCREEN?

                 TYPE "YES" OR "NO".
n
                                  GENERAL OPTSYSX OPTIONS:

           OPTION 1 -- SYSTEM ANALYSIS WITHOUT
                       OPEN-LOOP EIGENSYSTEM CALCULATIONS.

           OPTION 2 -- SYSTEM ANALYSIS WITH OPEN-LOOP
                       EIGENSYSTEM CALCULATIONS.

           OPTION 3 -- OPEN-LOOP EIGENSYSTEM FOUND
                       AND PROGRAM TERMINATES.
                       {"F"-MATRIX ENTRY FOLLOWS IMMEDIATELY.}

           OPTION 4 -- MODAL DISTRIBUTION MATRICES COMPUTED
                       WITHOUT FILTER OR REGULATOR SYNTHESIS
                       OR STEADY-STATE ANALYSIS.

                SELECT AN OPTION: 1,2,3, OR 4.
?
1

    DO YOU DESIRE RMS VALUES OF STATE AND CONTROL?

            TYPE "YES" OR "NO".
n

                        OPTSYSX LQR/CLASSICAL OPTIONS:

           OPTION 1 -- OPTIMAL FILTER AND/OR REGULATOR
                       SYNTHESIS WITH NO EXTERNAL "C" OR "K"
                       MATRIX INPUT.

           OPTION 2 -- OPTIMAL FILTER AND/OR REGULATOR
                       SYNTHESIS WITH EXTERNAL "C"
                       MATRIX INPUT.

           OPTION 3 -- OPTIMAL FILTER AND/OR REGULATOR
                       SYNTHESIS WITH EXTERNAL "K"
                       MATRIX INPUT.

           OPTION 4 -- OPTIMAL FILTER AND/OR REGULATOR
                       SYNTHESIS WITH EXTERNAL "C" AND "K"
                       MATRIX INPUT.

           SELECT AN OPTION: 1, 2, 3, OR 4.
?
3

    DO YOU WISH TO DETERMINE THE STEADY-STATE RESPONSE
       FOR A CONSTANT DISTURBANCE?

            TYPE "YES" OR "NO".
n
    DO YOU WISH TO DETERMINE THE MODAL DISTRIBUTION
       AND GAIN MATRICES?

            TYPE "YES" OR "NO".
n

  OPEN-LOOP TRANSFER FUNCTION OPTIONS:

      OPTION 1 -- NO OPEN-LOOP TRANSFER FUNCTIONS COMPUTED.
```

```
        OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.
        OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.
        OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.
        SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

   NOISE TRANSFER FUNCTION OPTIONS:
        OPTION 1 -- NO NOISE TRANSFER FUNCTIONS COMPUTED.
        OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.
        OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.
        OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.
        SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

     COMPENSATOR TRANSFER FUNCTION OPTIONS:
        OPTION 1 -- NO COMP. TRANSFER FUNCTIONS COMPUTED.
        OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.
        OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.
        OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.
             {NOTE: A COMPENSATOR TRANSFER FUNCTION CAN BE
                    COMPUTED ONLY IF BOTH A REGULATOR
                    AND FILTER ARE SYNTHESIZED
                    AND/OR INPUT.}
        SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

     WILL A FEED-FORWARD DISTRIBUTION MATRIX
     {"D" - MATRIX} BE INPUT ?
                TYPE "YES" OR "NO".
n
   THIS OPTION DETERMINES THE CRITERIA FOR DECIDING WHEN A
   MARKOV PARAMETER IS ZERO-THE MARKOV PARAMETER INDICATES
   THE ORDER OF THE NUMERATOR POLYNOMIAL OF EACH TRANSFER
   FUNCTION.
     ALL "N" ZEROS OF THIS POLYNOMIAL ARE PRINTED OUT AND
     THIS TEST TELLS HOW MANY EXTRA ROOTS EXIST AT Z = 0.
     LESS THAN 10.0**{-IE} IS CONSIDERED ZERO.
        THE DEFAULT VALUE OF THIS PARAMETER {IE} IS 6.
        IN OTHER WORDS, IE = 1.0E-6.
      IF YOU DESIRE A DIFFERENT MARKOV CRITERIA,
      TYPE THE INTEGER VALUE.
      IF YOU DESIRE THE DEFAULT VALUE, TYPE "0" {ZERO}
?
```

```
0

 DO YCU DESIRE TO SYNTHESIZE A STABLE FILTER {OR REGULATOR} BY
     DESTABILIZING THE ORIGINAL SYSTEM?

        {NOTE:WORKS FCR FILTER OR REGULATOR BUT NOT FCR BOTH
              IN THE SAME RUN.}

           TYPE "YES" CR "NO".
n
     DC YOU DESIRE TC PRINT THE EULER-LAGRANGE EIGENSYSTEM
        PRIOR TO DECCMPOSITION {FOR CHECKING THE PROGRAM}?

           TYPE "YES" CR "NO".
n

  POWER SPECTRAL DENSITY {PSD} OPTION 1 :

 OPTICN 1 -- COMPUTE THE PSD OF THE OUTPUTS AND/OR THE
             CONTROLS OF THE CONTROLLED SYSTEM WHEN FORCED BY
             PROCESS AND MEASUREMENT NOISE.  {NOTE: BOTH A
             REGULATCR AND A FILTER MUST BE RESIDENT IN THE
             PROGRAM TO USE THIS OPTION.}

 OPTION 2 -- SAME AS OPTION 1 ABOVE BUT ONLY PRINT THE
             RESIDUES OF EACH TRANSFER FUNCTION
             USED IN THE PSD COMPUTATION.

 OPTICN 3 -- NOT DESIRED.

           SELECT AN OPTION: 1, 2, OR 3.
?
3

     DO YOU DESIRE REGULATOR SYNTHESIS ONLY?

           TYPE "YES" CR "NO".

n

     ENTER THE # OF STATES {NS} OF THE SYSTEM MATRIX
     {"F"-MATRIX}.
?
2

   ENTER THE # OF CONTROLS {NC} OF THE CONTROL SYSTEM MODEL
   {"G"-MATRIX}.
?
1

     ENTER THE # OF MEASUREMENTS OR OBSERVATIONS {NO} OF THE
     {"H"-MATRIX}.
?
1

     ENTER THE # OF PROCESS NOISE SOURCES {NG} OF THE
     {"GAMMA"-MATRIX}.
?
1
    FLAG/PARAMETER SETTINGS FOR THIS RUN ARE AS FOLLOWS:

IOL   IQ   IR   ISS   IM   ITF1   ITF2   ITF3   IFDFW   IE   IDEBUG
 0     0    2    0     0     0      0      0      0      0     0
```

```
ISET   IDSTAB  IPSD   IYU   INORM   IREG   NS   NC   NOB   NG
   0       0      0     0      0       0    2    1     1    1

 ORDER OF SYSTEM =   2
 NUMBER CF CONTROLS =  1
 NUMBER OF OBSERVATICNS =   1
 NUMBER CF PROCESS NCISE SOURCES =  1

     ENTER THE SYSTEM MATRIX {"F"-MATRIX}

          DIMENSION = # STATES {NS} X # STATES {NS}
     THE ELEMENT F( 1, 1)=
?
0
     THE ELEMENT F( 1, 2)=
?
1
     THE ELEMENT F( 2, 1)=
?
0
     THE ELEMENT F( 2, 2)=
?
-4.6

                THE SYSTEM MATRIX {"F"-MATRIX}...

     0.0           1.00000
     0.0          -4.60000

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" CR "NO".
n

    CPEN LOOP DYNAMICS MATRIX.................F..

  0.0           0.1000D+01
  0.0          -0.4600D+01

     ENTER THE MEASUREMENT SCALING MATRIX {"H"-MATRIX}.

          DIMENSION = # OBSERVATIONS {NO} X # STATES {NS}
     THE ELEMENT H( 1, 1)=
?
1
     THE ELEMENT H( 1, 2)=
?
0

          THE MEASUREMENT SCALING MATRIX {"H"-MATRIX}...

     1.00000     0.0

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" CR "NO".
n
```

```
   MEASUREMENT SCALING MATRIX................H..
 0.1000D+01   0.0

     ENTER THE OUTPUT MEASUREMENT COST MATRIX {"A"-MATRIX}.
     DIMENSION = # OBSERVATIONS {NO} X # OBSERVATIONS {NO}
     THE ELEMENT A( 1, 1)=
?
1

     THE OUTPUT MEASUREMENT COST MATRIX {"A"-MATRIX}...

     1.00000

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" OR "NO".
n

   OUTPUT COST MATRIX.........................A..
 0.1000D+01
     ENTER THE CONTROL DISTRIBUTION MATRIX {"G"-MATRIX}.
          DIMENSION = # STATES {NS} X # CONTROLS {NC}
     THE ELEMENT G( 1, 1)=
?
0
     THE ELEMENT G( 2, 1)=
?
0.787

          THE CONTROL DISTRIBUTION MATRIX {"G"-MATRIX}...

     0.0
     0.78700

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" OR "NO".
n
     ENTER THE CONTROL COST WEIGHTING MATRIX {"B"-MATRIX}
          DIMENSION = # CONTROLS {NC} X # CONTROLS {NC}
     THE ELEMENT B( 1, 1)=
?
1

          THE CONTROL COST MATRIX..........B...

     1.00000

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" OR "NO".
n
```

```
    THE CONTROL DISTRIBUTION MATRIX...........G..

  0.0
  0.7870D+00

    THE CONTROL COST MATRIX....................B..

  0.1000D+01

    EIGENSYSTEM OF OPTIMAL REGULATOR.........

    C-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G*C)..

-1.71206D-01:-4.59681D+00:

    C-LOOP RIGHT EIGENVECTOR MATRIX..........M....

 9.856588D-01 -2.125703D-01
-1.687503D-01  9.771458D-01

    CONTROL EIGENVECTOR MATRIX..............C*M..

   -9.496319D-01  3.957155D-03

    C-LOOP OPT. REG. LEFT E-VECTOR MATRIX..M-INV..

 1.053798D+00  2.292453D-01
 1.819879D-01  1.062979D+00

 THE OPTIMAL FEEDBACK GAIN CONTROL MATRIX...C=BINV*GT*S...

 -1.0000D+00  -2.1349D-01

    THE CLOSED LOOP DYNAMICS MATRIX ......F-G*C..

 0.0           1.000000D+00
-7.870000D-01 -4.768018D+00

     ENTER THE PROCESS NOISE DISTRIBUTION
     MATRIX {"GAMMA"-MATRIX}.
  DIMENSION = # STATES {NS} X # PROCESS NOISE SOURCES {NG}
     THE ELEMENT GAM( 1, 1) =
?
0
     THE ELEMENT GAM( 2, 1) =
?
0.1

          THE PROCESS NOISE DISTRIBUTION MATRIX
          {"GAMMA"-MATRIX}...

     0.0
     0.10000

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?

          TYPE "YES" OR "NO".
n
```

```
     ENTER THE PROCESS NOISE PSD WEIGHTING MATRIX
     {"Q"MATRIX}.
     DIMENSION = # PROCESS NOISE SOURCES {NG} X
                 #PROCESS NOISE SOURCES {NG}
     THE ELEMENT Q( 1, 1)=
?
10

     THE PROCESS NOISE WEIGHTING MATRIX.....Q..

    10.00000

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" OR "NO".
n

    PROCESS NOISE DISTRIBUTION MATRIX.....GAMMA..

  0.0
  0.1000D+00

    POWER SPECTRAL DENSITY - PROCESS NOISE....Q..

  0.1000D+02
 ENTER THE MEASUREMENT NOISE DISTRIBUTION MATRIX {"R"MATRIX}.
     DIMENSION = # OBSERVATIONS {NO} X # OBSERVATIONS {NO}
     THE ELEMENT R( 1, 1)=
?
0.0000001

       THE MEASUREMENT NOISE DISTRIBUTION MATRIX.....R...

     0.00000
     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" OR "NO".
n

    POWER SPECTRAL DENSITY-MEASUREMENT NOISE..R..
  0.1000D-06
     ENTER THE FEEDBACK GAIN ESTIMATOR MATRIX {"K"-MATRIX}.
          DIMENSION = # STATES {NS} X # OBSERVATIONS {NO}.
     THE ELEMENT K( 1, 1)=
?
95.4
     THE ELEMENT K( 2, 1)=
?
4561

          THE FEEDBACK GAIN ESTIMATOR MATRIX {"K"-MATRIX}
    95.40000
```

```
  4561.00000

      DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
           TYPE "YES" OR "NO".
n

    FILTER STEADY STATE GAINS................K...

    9.540000D+01
    4.561000D+03

    THE CLOSED LOOP FILTER DYNAMICS MATRIX IS....

-9.540000D+01  1.000000D+00
-4.561000D+03 -4.600000D+00

    EIGENSYSTEM OF OPTIMAL ESTIMATOR.........

    C-LOOP SUBOPT. EST. E-VALUES...DET(SI-F+K*H)..

-5.00000D+01, 4.99984D+01:

    C-LOOP RIGHT EIGENVECTOR MATRIX..........M....

 9.953957D-03 -1.096216D-02
 1.000000D+00  0.0

    MEASUREMENT EIGENVECTOR MATRIX.....H(BAR)*M..

    9.953957D-03 -1.096216D-02

    C-LOOP SUBOPT. FILTER LEFT E-VECTOR MATRIX..M-INV..

 0.0           1.000000D+00
-9.122292D+01  9.080291D-01

    THE COVARIANCE OF THE ESTIMATION ERROR....P..

 7.150503D-06  2.271000D-04
 2.271000D-04  1.181151D-02

    RMS VALUES OF THE ESTIMATION ERROR...........

 2.674042D-03  1.086808D-01
          DO YOU WISH TO OBTAIN A TIME RESPONSE
            OF THE SYSTEM YOU ARE EVALUATING?
                        (Y OR N)

      NOTE: YOU MUST BE LOGGED ON AT A DUAL SCREEN
           (TEK 618) TERMINAL TO UTILIZE THIS MODE.

   THE F (SYSTEM), G (CONTROL), H (OBSERVABLES), GAM (NOISE),
      A (OUTPUT COST) AND B (CONTROL COST) MATRICES WILL BE
         SAVED FOR REENTRY TO THE MAIN OPTSYS PROGRAM.
y
```

```
*************************************************************************
          IF YOU ARE DISSATISFIED WITH THE RESULTS
          THUS FAR AND WOULD LIKE TO EXIT TO CMS,

                    -TYPE 'Y' TO EXIT-

              (ANY OTHER INPUT TO CONTINUE)
*************************************************************************

LOADING OPTCALC....
EXECUTION BEGINS...

   DURING THIS SECTION OF THE PROGRAM YOU WILL:

  - SELECT THE TYPE OF SYSTEM RESPONSE TO PLOT
    (OPEN LOOP, CLOSED LOOP, OR FILTER/REGULATOR)
  - PROVIDE START AND STOP TIME FOR PLOTTING CALCULATIONS
  - SELECT THE TYPE OF DRIVING FUNCTION(S) (STEP OR RAMP)
  - PROVIDE START AND STOP TIMES FOR THE DRIVING FUNCTION(S)
  - PROVIDE DRIVING FUNCTION MAGNITUDE(S).

                 CLEAR THE SCREEN TO CONTINUE

                    THE F MATRIX

     0.0             1.00000
     0.0            -4.60000

                    THE G MATRIX

     0.0
     0.78700

                    THE C MATRIX

   -1.00000         -0.21349

                    THE H MATRIX

    1.00000          0.0

                    THE K MATRIX

   95.40000
 4561.00000

 THE FOLLOWING PLOTTING OPTIONS ARE AVAILABLE IF THE
 REQUIRED MATRICES WERE CALCULATED IN OPTSYSX:

    1.  OPEN LOOP TIME RESPONSE
         XDOT =  {F}*X +  {G}*UC

    2.  CLOSED LOOP TIME RESPONSE
         XDOT =  {F-G*C}*X +  {G}*UC,    U =  {C}*X

    3.  OPTIMIZED FILTER CLOSED LOOP SYSTEM RESPONSE.
         XDOT =  {F}*X +  {G}*UC,         Z =  {H}*X
         XHDOT =  {F}*XH +{G}U +  {K}*{Z - H*XH}

    4.  FILTER + REGULATOR CLOSED LOOP SYSTEM RESPONSE.
         XDOT =  {F+G*C}*X +  {G}*UC,                Z =  {H}*X
         XHDOT =  {F}*XH +{G}U +  {K}*{Z - H*XH},    U =  {C}*XH
```

```
          SELECT 1, 2, 3 OR 4.
?
3


                 THE (K*H) MATRIX

   95.40000      0.0
 4561.00000      0.0


                 THE COMBINED SYSTEM F MATRIX (2*NS X 2*NS)

    0.0           1.00000      0.0           0.0
    0.0          -4.60000      0.0           0.0
   95.40000       0.0        -95.40000       1.00000
 4561.00000       0.0      -4561.00000      -4.60000

                 THE AUGMENTED G MATRIX (2*NS X NC)

    0.0
    0.78700
    0.0
    0.78700

          AT WHAT TIME DO YOU WANT TO START
          THE TIME RESPONSE CALCULATIONS?

          INPUT START TIME IN SECONDS. (NORMALLY 0.0)
?
0


          AT WHAT TIME DO YOU WANT TO STOP
          THE TIME RESPONSE CALCULATIONS?

            INPUT STOP TIME IN SECONDS.
?
0.3

          THIS PROGRAM DIVIDES THE TIME INTERVAL YOU HAVE
          JUST SPECIFIED INTO UP TO 500 SMALL INTERVALS FOR
          THE INTEGRATION AND PLOTTING ROUTINES.  IN ORDER
          TO SAVE COMPUTER TIME, THE NUMBER OF POINTS CAN BE
          CAN BE REDUCED WITH SOME LOSS IN CURVE FIDELITY.

          HOW MANY POINTS DO YOU WANT TO CALCULATE?
?
500

  DOES THE SYSTEM UTILIZE A DRIVING FUNCTION (CONTROL INPUT)?

                      (Y)ES OR (N)O
Y

          TWO TYPES OF FUNCTIONS CAN BE USED AS DRIVERS.
                1. STEP INPUT
                2. RAMP INPUT

ENTER YOUR SELECTION, 1 OR 2. FOR DRIVING FUNCTION NUMBER 1
?
1

     AT WHAT TIME DO YOU DESIRE INPUT NUMBER 1 TO START?

            INPUT THE START TIME IN SECONDS.
```

```
?
0
     AT WHAT TIME DO YOU DESIRE INPUT NUMBER 1 TO STOP?
              INPUT THE STOP TIME IN SECONDS.
?
0.4
          WHAT IS THE MAXIMUM VALUE OF
          DRIVING FUNCTION NUMBER 1 ?
?
-10

DOES THE SYSTEM START WITH ALL INITIAL CONDITIONS = 0.0 ?
                         (Y)ES OR (N)O?
n
          WHAT IS THE INITIAL CONDITION FOR X( 1) ?
?
0.1
          WHAT IS THE INITIAL CONDITION FOR XHAT( 1) ?
?
0
          WHAT IS THE INITIAL CONDITION FOR X( 2) ?
?
0.5
          WHAT IS THE INITIAL CONDITION FOR XHAT( 2) ?
?
0

          THIS IS YOUR LAST OPPORTUNITY TO
          MAKE CHANGES IN THE FOLLOWING AREAS.
            1.   SELECT ANOTHER TYPE OF SYSTEM TO PLOT
                 (OPEN, CLOSED, FILTER OR FILTER/REGULATOR)
            2.   START AND STOP TIMES
            3.   DRIVING FUNCTIONS
            4.   INITIAL CONDITIONS
            5.   CONTINUE
          SELECT A NUMBER BETWEEN 1 AND 5.
?
5

          THE FOLLOWING INFORMATION IS PROVIDED ONLY
          FOR AN INDICATION OF PROPER PROGRAM OPERATION.
     ALL CONTROLS, STATES AND STATE ESTIMATES CAN BE PLOTTED.
 TIME       U(1)            X(1)             X(2)             X(3)

 0.0    0.0             0.1000000D+00  0.5000000D+00  0.0
 0.01 -0.1000000D+02  0.1028186D+00  0.4398144D+00  0.4973463D-01
 0.01 -0.1000000D+02  0.1052810D+00  0.3812672D+00  0.8502367D-01
 0.02 -0.1000000D+02  0.1073970D+00  0.3243138D+00  0.1078565D+00
 0.02 -0.1000000D+02  0.1091759D+00  0.2689108D+00  0.1209438D+00
 0.03 -0.1000000D+02  0.1106270D+00  0.2150160D+00  0.1270318D+00
 0.04 -0.1000000D+02  0.1117590D+00  0.1625884D+00  0.1285212D+00
 0.04 -0.1000000D+02  0.1125809D+00  0.1115880D+00  0.1273039D+00
```

```
0.05 -0.1000000D+02 0.1131009D+00 0.6197598D-01 0.1247416D+00
0.05 -0.1000000D+02 0.1133273D+00 0.1371451D-01 0.1217242D+00
0.06 -0.1000000D+02 0.1132681D+00-0.3323315D-01 0.1187650D+00
0.07 -0.1000000D+02 0.1129310D+00-0.7890278D-01 0.1161034D+00
0.07 -0.1000000D+02 0.1123237D+00-0.1233292D+00 0.1137974D+00
0.08 -0.1000000D+02 0.1114535D+00-0.1665461D+00 0.1117983D+00
0.08 -0.1000000D+02 0.1103275D+00-0.2085866D+00 0.1100069D+00
0.09 -0.1000000D+02 0.1089528D+00-0.2494827D+00 0.1083096D+00
0.10 -0.1000000D+02 0.1073360D+00-0.2892654D+00 0.1066016D+00
0.10 -0.1000000D+02 0.1054837D+00-0.3279652D+00 0.1047973D+00
0.11 -0.1000000D+02 0.1034025D+00-0.3656114D+00 0.1028338D+00
0.11 -0.1000000D+02 0.1010984D+00-0.4022329D+00 0.1006703D+00
0.12 -0.1000000D+02 0.9857769D-01-0.4378574D+00 0.9828370D-01
0.13 -0.1000000D+02 0.9584610D-01-0.4725121D+00 0.9566496D-01
0.13 -0.1000000D+02 0.9290943D-01-0.5062234D+00 0.9281438D-01
0.14 -0.1000000D+02 0.8977326D-01-0.5390170D+00 0.8973812D-01
0.14 -0.1000000D+02 0.8644301D-01-0.5709178D+00 0.8644545D-01
0.15 -0.1000000D+02 0.8292398D-01-0.6019503D+00 0.8294671D-01
0.16 -0.1000000D+02 0.7922130D-01-0.6321380D+00 0.7925213D-01
0.16 -0.1000000D+02 0.7533997D-01-0.6615038D+00 0.7537114D-01
0.17 -0.1000000D+02 0.7128485D-01-0.6900703D+00 0.7131205D-01
0.17 -0.1000000D+02 0.6706068D-01-0.7178591D+00 0.6708207D-01
0.18 -0.1000000D+02 0.6267205D-01-0.7448914D+00 0.6268741D-01
0.19 -0.1000000D+02 0.5812345D-01-0.7711879D+00 0.5813344D-01
0.19 -0.1000000D+02 0.5341923D-01-0.7967685D+00 0.5342495D-01
0.20 -0.1000000D+02 0.4856362D-01-0.8216527D+00 0.4856623D-01
0.20 -0.1000000D+02 0.4356075D-01-0.8458595D+00 0.4356131D-01
0.21 -0.1000000D+02 0.3841463D-01-0.8694073D+00 0.3841398D-01
0.22 -0.1000000D+02 0.3312915D-01-0.8923141D+00 0.3312793D-01
0.22 -0.1000000D+02 0.2770811D-01-0.9145973D+00 0.2770673D-01
0.23 -0.1000000D+02 0.2215519D-01-0.9362739D+00 0.2215392D-01
0.23 -0.1000000D+02 0.1647400D-01-0.9573604D+00 0.1647295D-01
0.24 -0.1000000D+02 0.1066802D-01-0.9778729D+00 0.1066723D-01
0.25 -0.1000000D+02 0.4740642D-02-0.9978270D+00 0.4740106D-02
0.25 -0.1000000D+02-0.1304821D-02-0.1017238D+01-0.1305149D-02
0.26 -0.1000000D+02-0.7465155D-02-0.1036120D+01-0.7465326D-02
0.26 -0.1000000D+02-0.1373724D-01-0.1054489D+01-0.1373730D-01
0.27 -0.1000000D+02-0.2011802D-01-0.1072357D+01-0.2011801D-01
0.28 -0.1000000D+02-0.2660455D-01-0.1089739D+01-0.2660450D-01
0.28 -0.1000000D+02-0.3319394D-01-0.1106648D+01-0.3319388D-01
0.29 -0.1000000D+02-0.3988340D-01-0.1123096D+01-0.3988334D-01
0.29 -0.1000000D+02-0.4667020D-01-0.1139097D+01-0.4667015D-01
0.30 -0.1000000D+02-0.5355170D-01-0.1154662D+01-0.5355166D-01

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

        IF YOU ARE DISSATISFIED WITH THE RESULTS
        THUS FAR AND WOULD LIKE TO EXIT TO CMS,

              -TYPE 'Y' TO EXIT-

           (ANY OTHER INPUT TO CONTINUE)

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

B (120) R/O
C (121) R/O
E (122) R/O

         ... Your Fortran program is now being loaded ...
               ... execution will soon follow ...
EXECUTION BEGINS...

         THIS PORTION OF THE PROGRAM PLOTS:
              - THE STATES,
```

```
          - EXTERNAL CONTROL INPUTS,
          - FEEDBACK CONTROL INPUTS,
          - STATE ESTIMATES AND
          - RECONSTRUCTION ERRORS
     FROM THE DATA THAT YOU JUST CALCULATED.

     THE CAPABILITY IS ALSO AVAILABLE TO REVIEW ANY
     GRAPHS THAT YOU HAD PREVIOUSLY SAVED AS DATA
     FILES ON YOUR DISK.

          CLEAR THE SCREEN TO CONTINUE.


     THE FOLLOWING OPTIONS ARE AVAILABLE:

          1. PLOT THE DATA YOU JUST CALCULATED.
          2. PLOT A CURVE THAT YOU PREVIOUSLY SAVED.

     ENTER 1 OR 2
?
1


     YOU MAY PLOT UP TO 4 SYSTEM VARIABLES VS TIME.
HOW MANY VARIABLES DO YOU WISH TO PLOT?
?
2


 WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 1?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC)

     ENTER 1,2,3,4 OR 5
?
1


     WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
     YOU WANT TO PLOT AS THE NUMBER  1 CURVE VS TIME?
?
1


   WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

    NOTE:  1. 40 CHARACTERS MAX LENGTH
           2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
              ENCLOSED IN PARENTHESES.
              IE. (A) => ALPHA
                  (B) => BETA
                  (F) => PHI
                  (Q) => THETA
angular position - (x)1

WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 2?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC)
```

```
          ENTER 1,2,3,4 OR 5
?
4

          WHAT IS THE SUBSCRIPT OF THE STATE ESTIMATE THAT
          YOU WANT TO PLOT AS THE NUMBER  2 CURVE VS TIME?
?
1

     WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE. (A) => ALPHA
                   (B) => BETA
                   (F) => PHI
                   (Q) => THETA
angular position estimate -  (x)e1

          YOU MAY USE UP TO 3 HEADINGS.
          HOW MANY HEADINGS DO YOU DESIRE ON THIS GRAPH?

                0, 1, 2 OR 3
?
3

     WHAT IS THE DESIRED HEADING NUMBER 1?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE. (A) => ALPHA
                   (B) => BETA
                   (F) => PHI
                   (Q) => THETA
filter only closed loop

     WHAT IS THE DESIRED HEADING NUMBER 2?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE. (A) => ALPHA
                   (B) => BETA
                   (F) => PHI
                   (Q) => THETA
example 4.1

     WHAT IS THE DESIRED HEADING NUMBER 3?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE. (A) => ALPHA
                   (B) => BETA
                   (F) => PHI
                   (Q) => THETA
linear optimal control systems
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.

   THE FOLLOWING OPTIONS ARE AVAILABLE.

  1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
  2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
```

```
  3. EDIT THE CURRENT GRAPH.
  4. PLOT REVISED GRAPH ON THE TEK618.
  5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
       PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
3

              THE GRAPH EDIT MENU

   1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
   2. DELETE CURVE FROM CURRENT PLOT.
   3. EDIT CURVE TITLE(S).
   4. EDIT PAGE HEADING(S).
   5. CHANGE THE Y-AXIS SCALE.
   6. CHANGE THE TIME AXIS SCALE.
   7. CHANGE PLOT SIZE. (DEFAULT IS 8.5 X 6.0)
   8. CHANGE THE LETTERING HEIGHT.
   9. CHANGE POSITION OF THE LEGEND.
  10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.
?
5

     ON WHICH CURVE DO YOU WANT TO CHANGE THE Y-SCALE?

            ENTER CURVE NUMBER- 1, 2, 3, OR 4
?
1

          WHAT IS THE NEW Y-MIN VALUE AT THE ORIGIN?
?
-.075

          WHAT IS THE NEW Y-MAX VALUE?
?
0.15

              THE GRAPH EDIT MENU

   1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
   2. DELETE CURVE FROM CURRENT PLOT.
   3. EDIT CURVE TITLE(S).
   4. EDIT PAGE HEADING(S).
   5. CHANGE THE Y-AXIS SCALE.
   6. CHANGE THE TIME AXIS SCALE.
   7. CHANGE PLOT SIZE. (DEFAULT IS 8.5 X 6.0)
   8. CHANGE THE LETTERING HEIGHT.
   9. CHANGE POSITION OF THE LEGEND.
  10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.
?
10

    THE FOLLOWING OPTIONS ARE AVAILABLE.

 1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
 2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
 3. EDIT THE CURRENT GRAPH.
 4. PLOT REVISED GRAPH ON THE TEK618.
 5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
    PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
4
```

```
     THE FOLLOWING OPTIONS ARE AVAILABLE.

  1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
  2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
  3. EDIT THE CURRENT GRAPH.
  4. PLOT REVISED GRAPH ON THE TEK618.
  5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
     PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
3

            THE GRAPH EDIT MENU

    1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
    2. DELETE CURVE FROM CURRENT PLOT.
    3. EDIT CURVE TITLE(S).
    4. EDIT PAGE HEADING(S).
    5. CHANGE THE Y-AXIS SCALE.
    6. CHANGE THE TIME AXIS SCALE.
    7. CHANGE PLOT SIZE. (DEFAULT IS 8.5 X 6.0)
    8. CHANGE THE LETTERING HEIGHT.
    9. CHANGE POSITION OF THE LEGEND.
   10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.
?
5

     ON WHICH CURVE DO YOU WANT TO CHANGE THE Y-SCALE?

              ENTER CURVE NUMBER- 1, 2, 3, OR 4
?
1

            WHAT IS THE NEW Y-MIN VALUE AT THE ORIGIN?
?
-.06

            WHAT IS THE NEW Y-MAX VALUE?
?
0.15

                THE GRAPH EDIT MENU

    1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
    2. DELETE CURVE FROM CURRENT PLOT.
    3. EDIT CURVE TITLE(S).
    4. EDIT PAGE HEADING(S).
    5. CHANGE THE Y-AXIS SCALE.
    6. CHANGE THE TIME AXIS SCALE.
    7. CHANGE PLOT SIZE. (DEFAULT IS 8.5 X 6.0)
    8. CHANGE THE LETTERING HEIGHT.
    9. CHANGE POSITION OF THE LEGEND.
   10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.
?
10

    THE FOLLOWING OPTIONS ARE AVAILABLE.

  1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
  2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
  3. EDIT THE CURRENT GRAPH.
  4. PLOT REVISED GRAPH ON THE TEK618.
  5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
```

```
    PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
4

    THE FOLLOWING OPTIONS ARE AVAILABLE.

 1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
 2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
 3. EDIT THE CURRENT GRAPH.
 4. PLOT REVISED GRAPH ON THE TEK618.
 5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
    PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
5

     DO YOU WANT TO SAVE THE CURRENT GRAPH DATA TO
     BE USED LATER TO GENERATE A METAFILE?

                   Y OR N

    NOTE: A METAFILE IS REQUIRED FOR SMOOTH VERSATEC PLOTS.
    THERE WILL BE AN OPPORTUNITY TO GENERATE A METAFILE
    JUST BEFORE EXITING THIS PROGRAM.
filteron

     YOUR ANSWER MUST BE "Y" OR "N".

     DO YOU WANT TO SAVE THE CURRENT GRAPH DATA TO
     BE USED LATER TO GENERATE A METAFILE?

                   Y OR N

    NOTE: A METAFILE IS REQUIRED FOR SMOOTH VERSATEC PLOTS.
    THERE WILL BE AN OPPORTUNITY TO GENERATE A METAFILE
    JUST BEFORE EXITING THIS PROGRAM.
y

     WHAT FILE NAME DO YOU WANT THE CURVE DATA STORED UNDER?
                   (8 CHARACTERS MAX)
filteron

          THE CURVE DATA IS BEING FILED UNDER FILTERON DATA
END OF DISSPLA 9.0 -- 43644 VECTORS GENERATED IN 3 PLOT FRAMES
    PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CA.
8493 VIRTUAL STORAGE REFERENCES; 9 READS; 0 WRITES.

          THE FOLLOWING OPTIONS ARE AVAILABLE:

          1.  MAKE METAFILE OF PREVIOUSLY SAVED CURVE.
          2.  QUIT.

              ENTER 1 OR 2
?
1

          WHAT FILE NAME IS THE DATA STORED UNDER?
filteron

THE CURVE DATA IS BEING LOADED FROM FILE FILTERON DATA
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.
```

```
          THE FOLLOWING OPTIONS ARE AVAILABLE:

          1.  MAKE METAFILE OF PREVIOUSLY SAVED CURVE.
          2.  QUIT.

              ENTER 1 OR 2
?
2
END OF DISSPLA 9.0 -- 14919 VECTORS GENERATED IN 1 PLOT FRAMES
   PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CA.
2874 VIRTUAL STORAGE REFERENCES; 9 READS; 0 WRITES.
DASD 121 DETACHED
DASD 122 DETACHED
DASD 120 DETACHED

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

    DO YOU WANT A VRSTEC PLOTTER SMOOTH COPY OF THE
      THE DISSPLA METAFILE THAT YOU JUST CREATED?
                        (Y OR N)

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

y
B (120) R/O
DASD 001 LINKED R/O; R/W BY MVS
Z (001) R/O - OS
DASD 001 DETACHED
CREATING NEW FILE:
CREATING NEW FILE:
PUN FILE 6910  TO  MVS      COPY 001   NOHOLD
DASD 120 DETACHED

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

   YOUR GRAPH(S) CAN BE PICKED UP AT THE COMPUTER CENTER.

     THE GRAPH(S) WILL BE ADDRESSED TO "POP (USER ID)".

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

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

             DO YOU WANT TO

          1. RUN OPTSYSX AGAIN
          2. RUN THE PLOT PROGRAM USING THE SAME MATRICES?
            (TO PLOT ANOTHER TYPE OF SYSTEM (OPEN/CLOSED))
          3. QUIT

                 ENTER 1, 2 OR 3

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

3

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

                  HAVE A GOOD DAY!!

****************************************************************
R; T=13.37/21.24 21:23:09
record off
END RECORDING OF TERMINAL SESSION
```

The graphical output generated by this example follows as figure 3.3.

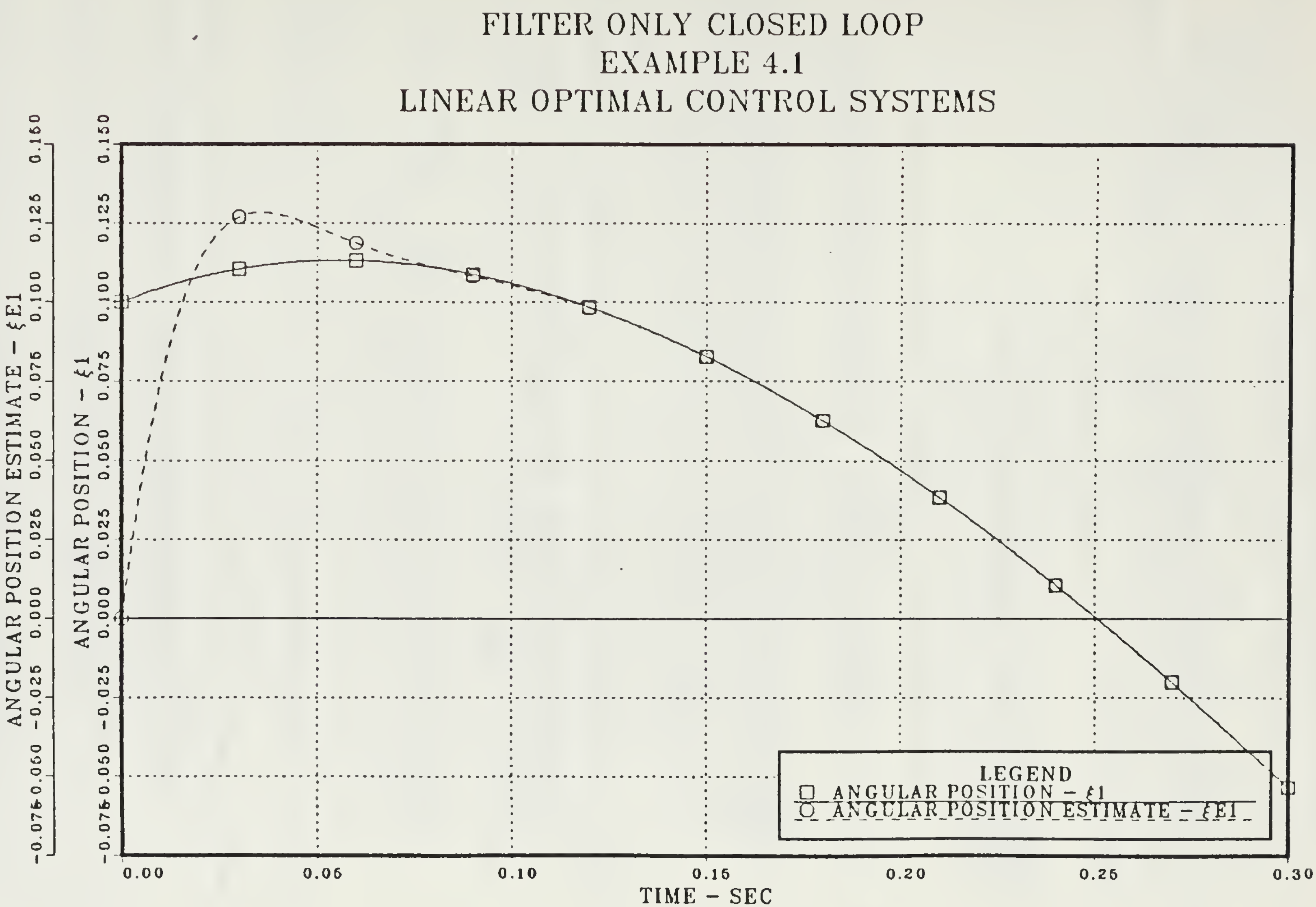

Figure 3.3 Filter Closed-loop Time Response

D. FILTER AND REGULATOR CLOSED LOOP TIME RESPONSE

The following filter and regulator example was taken from [Ref. 7 pp. 382 - 384].

The full terminal session is recorded below, with user input at the left margin in lower case letters or numbers below each "?".

```
record on
BEGIN RECORDING OF TERMINAL SESSION
R; T=0.01/0.02 09:08:30
optsys

      THE OPTSYS EXEC CONTROLS A TRIO OF PROGRAMS:

         1. OPTSYSX FORTRAN   (SYSTEM ANALYSIS)
         2. OPTCALC FORTRAN   (CALCULATE TIME RESPONSE)
         3. OPTPLOT FORTRAN   (DISSPLA PLOTTING ROUTINE)

    EACH PROGRAM PASSES INFORMATION TO THE NEXT PROGRAM
    THROUGH A DATA FILE WRITTEN TO THE USERS DISK.  IN THIS
    CASE, THESE FILES ARE "OPTMAT DATA" AND "OPTPLOT DATA".
    THE SIZE OF THESE FILES VARY WITH THE SYSTEM ORDER, AND
    CAN USE ABOUT 20% OF THE USERS DISK SPACE.  THEREFORE
    ENSURE THAT SUFFICIENT DISK SPACE IS AVAILABLE.

       - TYPE "E" TO EXIT, ANY OTHER ENTRY TO CONTINUE -

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

     YOU HAVE A DATA FILE NAMED 'OPTMAT DATA' ON YOUR A
     DISK  THAT WAS PREVIOUSLY GENERATED BY THE  OPTSYS
     PROGRAM  AND  CONTAINS THE F, G, H, GAMMA, A AND B
     MATRICES FROM THAT RUN.

     IF  YOU WOULD LIKE TO USE THESE SAME MATRICES  FOR
     THIS  RUN, THE  OPTSYS  PROGRAM  WILL  READ IN THE
     DESIRED DATA AT THE APPROPRIATE TIME,

                     IF YOU TYPE (Y)ES.

     ANY OTHER INPUT WILL RESULT IN THAT FILE BEING ERASED!
****************************************************************
y

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

      DO YOU WANT THE NUMERICAL OUTPUT FROM OPTSYSX TO GO
        TO YOUR TERMINAL S(CREEN) OR TO A D(ISK) FILE?
                          (S OR D)

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

s

****************************************************************
```

```
          OUTPUT WILL COME TO YOUR TERMINAL SCREEN.

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

LOADING OPTSYS....
EXECUTION BEGINS...
  OPTSYSX IS A COMPLETELY INTERACTIVE OPTIMAL SYSTEMS CONTROL
     PROGRAM. IT WILL SOLVE NUMEROUS CONTROL PROBLEMS ON THE
     FOLLOWING TYPES OF SYSTEMS CONTROL EQUATIONS:

          XDOT = {F}*X + {G}*U + {GAM}*(W+W0)

               MEASUREMENT EQUATION--

          Z = {H}*X + {D}*U + V

               REGULATOR PERFORMANCE INDEX--

          J = 1/2 * INTEGRAL (Y *{A}*Y + U *{B}*U) DT

               STATE FEEDBACK GAIN DEFINITION--

                       U = -{C}*X

             DO YOU WISH TO CONTINUE?  TYPE "YES" OR "NO".
y
                       --DATA ENTRY--

     ALTHOUGH OPTSYSX IS SPECIFICALLY DESIGNED TO READ
     ALL MATRIX DATA INTERACTIVELY, SEVERAL ALTERNATE
     METHODS ARE AVAILABLE TO USERS:

     METHOD 1--THE "F","G", AND "GAMMA" MATRICES
        MAY BE READ FROM SEPARATE DATA FILES.

        METHOD 2--THE "F","G", AND "GAMMA" MATRICES MAY BE
           EXPLICITLY DEFINED WITHIN SUBROUTINE "SETUP".

        {NOTE: IN EITHER CASE, THE USER SHOULD OBTAIN A COPY
               OF THE PROGRAM LISTING AND EXAMINE
               THE EXAMPLES CONTAINED IN S/R "SETUP".}

         DO YOU WISH TO CONTINUE?  TYPE "YES" OR "NO".
y

     DO YOU WISH TO INPUT THE "F", "G", AND "GAMMA"
         MATRICES FROM SUBROUTINE "SETUP" IAW THE
         METHOD DESCRIBED ON THE PREVIOUS SCREEN?

             TYPE "YES" OR "NO".
n
                  GENERAL OPTSYSX OPTIONS:

       OPTION 1 -- SYSTEM ANALYSIS WITHOUT
                   OPEN-LOOP EIGENSYSTEM CALCULATIONS.


       OPTION 2 -- SYSTEM ANALYSIS WITH OPEN-LOOP
                   EIGENSYSTEM CALCULATIONS.

       OPTION 3 -- OPEN-LOOP EIGENSYSTEM FOUND
                   AND PROGRAM TERMINATES.
                   {"F"-MATRIX ENTRY FOLLOWS IMMEDIATELY.}

       OPTION 4 -- MODAL DISTRIBUTION MATRICES COMPUTED
                   WITHOUT FILTER OR REGULATOR SYNTHESIS
                   OR STEADY-STATE ANALYSIS.
```

```
              SELECT AN OPTION: 1,2,3, OR 4.
?


     DO YOU DESIRE RMS VALUES OF STATE AND CONTROL?
          TYPE "YES" OR "NO".
n
                         CPTSYSX LQR/CLASSICAL OPTIONS:
          OPTION 1 -- OPTIMAL FILTER AND/OR REGULATOR
                      SYNTHESIS WITH NO EXTERNAL "C" OR "K"
                      MATRIX INPUT.
          OPTION 2 -- OPTIMAL FILTER AND/OR REGULATOR
                      SYNTHESIS WITH EXTERNAL "C"
                      MATRIX INPUT.
          OPTION 3 -- OPTIMAL FILTER AND/OR REGULATOR
                      SYNTHESIS WITH EXTERNAL "K"
                      MATRIX INPUT.
          OPTION 4 -- OPTIMAL FILTER AND/OR REGULATOR
                      SYNTHESIS WITH EXTERNAL "C" AND "K"
                      MATRIX INPUT.
          SELECT AN OPTION: 1, 2, 3, OR 4.
?
4

     DO YOU WISH TO DETERMINE THE STEADY-STATE RESPONSE
        FOR A CONSTANT DISTURBANCE?
          TYPE "YES" OR "NO".
n
     DO YOU WISH TO DETERMINE THE MODAL DISTRIBUTION
        AND GAIN MATRICES?
          TYPE "YES" OR "NO".
n

  OPEN-LOOP TRANSFER FUNCTION OPTIONS:
       OPTION 1 -- NO OPEN-LOOP TRANSFER FUNCTIONS COMPUTED.
          OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.
          OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.
          OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.
          SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

  NOISE TRANSFER FUNCTION OPTIONS:
       OPTION 1 -- NO NOISE TRANSFER FUNCTIONS COMPUTED.
       OPTION 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.
       OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.
       OPTION 4 -- ONLY POLES AND RESIDUES COMPUTED.
```

```
        SELECT AN OPTICN: 1, 2, 3, OR 4.
?
1

  COMPENSATCR TRANSFER FUNCTION OPTIONS:

        OPTICN 1 -- NO COMP. TRANSFER FUNCTIONS COMPUTED.

        OPTICN 2 -- PCLES, RESIDUES, AND ZEROS COMPUTED.

        OPTION 3 -- ONLY POLES AND ZEROS COMPUTED.

        OPTICN 4 -- ONLY POLES AND RESIDUES COMPUTED.

              {NOTE: A COMPENSATOR TRANSFER FUNCTION CAN BE
                     COMPUTED ONLY IF BOTH A REGULATOR
                     AND FILTER ARE SYNTHESIZED
                     AND/OR INPUT.}

           SELECT AN OPTION: 1, 2, 3, OR 4.
?
1

     WILL A FEED-FORWARD DISTRIBUTION MATRIX
     {"D" - MATRIX} BE INPUT ?

                TYPE "YES" OR "NO".
n

  THIS OPTION DETERMINES THE CRITERIA FOR DECIDING WHEN A
  MARKOV PARAMETER IS ZERO-THE MARKOV PARAMETER INDICATES
  THE ORDER OF THE NUMERATOR POLYNOMIAL OF EACH TRANSFER
  FUNCTICN.
  ALL "N" ZEROS OF THIS POLYNOMIAL ARE PRINTED OUT AND
  THIS TEST TELLS HOW MANY EXTRA ROOTS EXIST AT Z = 0.
  LESS THAN 10.0**{-IE} IS CONSIDERED ZERO.


  THE DEFAULT VALUE OF THIS PARAMETER {IE} IS 6.
  IN OTHER WORDS, IE = 1.0E-6.

        IF YOU DESIRE A DIFFERENT MARKOV CRITERIA,
        TYPE THE INTEGER VALUE.

        IF YOU DESIRE THE DEFAULT VALUE, TYPE "0" {ZERO}
?
0

 DO YOU DESIRE TO SYNTHESIZE A STABLE FILTER {OR REGULATOR} BY
     DESTABILIZING THE ORIGINAL SYSTEM?

        {NOTE:WORKS FOR FILTER OR REGULATOR BUT NOT FOR BOTH
              IN THE SAME RUN.}

           TYPE "YES" OR "NO".
n
     DO YOU DESIRE TO PRINT THE EULER-LAGRANGE EIGENSYSTEM
        PRIOR TO DECOMPOSITION {FOR CHECKING THE PROGRAM}?

           TYPE "YES" OR "NO".
n

  POWER SPECTRAL DENSITY {PSD} OPTION 1 :

OPTICN 1 -- COMPUTE THE PSD OF THE OUTPUTS AND/OR THE
```

```
            CONTROLS OF THE CONTROLLED SYSTEM WHEN FORCED BY
            PROCESS AND MEASUREMENT NOISE.  {NOTE: BOTH A
            REGULATOR AND A FILTER MUST BE RESIDENT IN THE
            PROGRAM TO USE THIS OPTION.}

OPTION 2 -- SAME AS OPTION 1 ABOVE BUT ONLY PRINT THE
            RESIDUES OF EACH TRANSFER FUNCTION
            USED IN THE PSD COMPUTATION.

OPTION 3 -- NOT DESIRED.

          SELECT AN OPTION: 1, 2, OR 3.
?
3

     DO YOU DESIRE REGULATOR SYNTHESIS ONLY?

          TYPE "YES" OR "NO".

n

          THE "F", "G", "H", "GAM", "A" AND "E" MATRICES
             FROM YOUR PREVIOUS OPTSYS RUN WERE SAVED.

          THE FOLLOWING OPTIONS ARE AVAILABLE:
               1. USE ALL OF THE SAME MATRICES AGAIN.
               2. USE SELECTED MATRICES AGAIN.
               3. INPUT ALL NEW MATRICES.

          ENTER 1, 2, OR 3.

          NOTE: EACH SAVED MATRIX WILL BE REDISPLAYED AT
          THE PROPER INPUT SEQUENCE INTERVAL
          AND YOU WILL HAVE THE OPTION OF CHANGING
          INDIVIDUAL MATRIX ELEMENTS.
?
    FLAG/PARAMETER SETTINGS FOR THIS RUN ARE AS FOLLOWS:

IOL   IQ   IR   ISS   IM   ITF1   ITF2   ITF3   IFDFW   IE   IDEBUG

 0    0    3    0     0    0      0      0      0       0    0

ISET   IDSTAB   IPSD   IYU   INORM   IREG   NS   NC   NOB   NG

  0      0       0      0     0       0      2    1    1     1

 ORDER OF SYSTEM =  2

 NUMBER OF CONTROLS =  1

 NUMBER OF OBSERVATIONS =  1

 NUMBER OF PROCESS NOISE SOURCES =  1

               THE SYSTEM MATRIX {"F"-MATRIX}...

     0.0          1.00000
     0.0         -4.60000

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
```

```
          TYPE "YES" OR "NO".
n

    OPEN LOOP DYNAMICS MATRIX.................F..

  0.0          0.1000D+01
  0.0         -0.4600D+01

          THE MEASUREMENT SCALING MATRIX {"H"-MATRIX}...

      1.00000     0.0

      DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" CR "NO".
n

    MEASUREMENT SCALING MATRIX................H..

  0.1000D+01   0.0

          THE CONTROL DISTRIBUTION MATRIX {"G"-MATRIX}...

      0.0
      0.78700

      DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" CR "NO".
n

      ENTER THE FEEDBACK GAIN CONTROL MATRIX {"C"-MATRIX}.
          DIMENSION = # CONTROLS {NC} X # STATES {NS}.
      THE ELEMENT C( 1, 1)=
?
-254.1
      THE ELEMENT C( 1, 2)=
?
-19.57

          THE FEEDBACK GAIN CONTROL MATRIX {"C"-MATRIX}

  -254.10000   -19.57000

      DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" OR "NO".
n

    THE CONTROL DISTRIBUTION MATRIX...........G..

  0.0
  0.7870D+00

 THE OPTIMAL FEEDBACK GAIN CONTROL MATRIX...C=BINV*GT*S...
 -2.5410D+02  -1.9570D+01
```

```
   THE CLOSED LOOP DYNAMICS MATRIX ......F-G*C..

 0.0            1.000000D+00
-1.999767D+02 -2.000159D+01

   C-IOCP SUBOPT. REG. E-VALUES...DET(SI-F+G*C)..

-1.00008D+01, 9.99804D+00:

   C-LOCP RIGHT EIGENVECTOR MATRIX..........M....

-5.000980D-02 -4.999602D-02
 1.000000D+00  0.0

   CCNTFCL EIGENVECTCR MATRIX..............C*M..

  -6.862510D+00  1.270399D+01

   C-ICCP SUBOPT-REG. LEFT E-VECTOR MATRIX..M-INV

 0.0            1.000000D+00
-2.000159D+01 -1.000276D+00

         THE PROCESS NOISE DISTRIBUTION MATRIX
         {"GAMMA"-MATRIX}...

    0.0
    0.10000

    DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
         TYPE "YES" CR "NO".
n

    ENTER THE PROCESS NOISE PSD WEIGHTING MATRIX
    {"Q"MATRIX}.

    DIMENSION = # PROCESS NOISE SOURCES {NG} X
                #PROCESS NOISE SOURCES {NG}
    THE ELEMENT Q( 1, 1)=
?
10

    THE PROCESS NOISE WEIGHTING MATRIX.....Q..

   10.00000

    DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
         TYPE "YES" CR "NO".
n

   PROCESS NOISE DISTRIBUTION MATRIX.....GAMMA..

  0.0
  0.1000D+00
```

```
    POWER SPECTRAL DENSITY - PROCESS NOISE....Q..
  0.1000D+02
ENTER THE MEASUREMENT NOISE DISTRIBUTION MATRIX {"R"MATRIX}.
     DIMENSION = # OBSERVATIONS {NO} X # OBSERVATIONS {NO}
     THE ELEMENT R( 1, 1)=
?
0.0000001

     THE MEASUREMENT NOISE DISTRIBUTION MATRIX.....R...
     0.00000

     DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?

          TYPE "YES" OR "NO".
n

    POWER SPECTRAL DENSITY-MEASUREMENT NOISE..R..
  0.1000D-06
     ENTER THE FEEDBACK GAIN ESTIMATOR MATRIX {"K"-MATRIX}.
          DIMENSION = # STATES {NS} X # OBSERVATIONS {NO}.
     THE ELEMENT K( 1, 1)=
?
95.4
     THE ELEMENT K( 2, 1)=
?
4561

     THE FEEDBACK GAIN ESTIMATOR MATRIX {"K"-MATRIX}

    95.40000
  4561.00000

   DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?
          TYPE "YES" OR "NO".
n

    FILTER STEADY STATE GAINS................K...

    9.540000D+01
    4.561000D+03

    THE CLOSED LOOP FILTER DYNAMICS MATRIX IS....

-9.540000D+01  1.000000D+00
-4.561000D+03 -4.600000D+00


    EIGENSYSTEM OF OPTIMAL ESTIMATOR.........

    C-LOOP SUBOPT. EST. E-VALUES...DET(SI-F+K*H)..
```

```
-5.00000D+01, 4.99984D+01:

   C-LOOP RIGHT EIGENVECTOR MATRIX..........M....

 9.953957D-03 -1.096216D-02
 1.000000D+00  0.0

    MEASUREMENT EIGENVECTOR MATRIX.....H(BAR)*M..

    9.953957D-03 -1.096216D-02

    C-LOOP SUBOPT. FILTER LEFT E-VECTOR MATRIX..M-INV..

 0.0            1.000000D+00
-9.122292D+01   9.080291D-01

    THE COVARIANCE OF THE ESTIMATION ERROR....P..

 7.150503D-06   2.271000D-04
 2.271000D-04   1.181151D-02

    RMS VALUES OF THE ESTIMATION ERROR...........

 2.674042D-03   1.086808D-01

          DO YOU WISH TO OBTAIN A TIME RESPONSE
           OF THE SYSTEM YOU ARE EVALUATING?

                       (Y OR N)

   NOTE: YOU MUST BE LOGGED ON AT A DUAL SCREEN
         (TEK 618) TERMINAL TO UTILIZE THIS MODE.

  THE F (SYSTEM), G (CONTROL), H (OBSERVABLES), GAM (NOISE),
     A (OUTPUT COST) AND B (CONTROL COST) MATRICES WILL BE
        SAVED FOR REENTRY TO THE MAIN OPTSYS PROGRAM.
Y

****************************************************************
      IF YOU ARE DISSATISFIED WITH THE RESULTS
      THUS FAR AND WOULD LIKE TO EXIT TO CMS,

             -TYPE 'Y' TO EXIT-

          (ANY OTHER INPUT TO CONTINUE)
****************************************************************

LOADING OPTCALC....
EXECUTION BEGINS...

  DURING THIS SECTION OF THE PROGRAM YOU WILL:

  - SELECT THE TYPE OF SYSTEM RESPONSE TO PLOT
    (OPEN LOOP, CLOSED LOOP, OR FILTER/REGULATOR)
  - PROVIDE START AND STOP TIME FOR PLOTTING CALCULATIONS
  - SELECT THE TYPE OF DRIVING FUNCTION(S) (STEP OR RAMP)
  - PROVIDE START AND STOP TIMES FOR THE DRIVING FUNCTION(S)
  - PROVIDE DRIVING FUNCTION MAGNITUDE(S).
```

```
                 CLEAR THE SCREEN TO CONTINUE

                    THE F MATRIX

     0.0             1.00000
     0.0            -4.60000

                    THE G MATRIX

     0.0
     0.78700

                    THE C MATRIX

  -254.10000       -19.57000

                    THE H MATRIX

     1.00000         0.0

                    THE K MATRIX

    95.40000
  4561.00000


    THE FOLLOWING PLOTTING OPTIONS ARE AVAILABLE IF THE
    REQUIRED MATRICES WERE CALCULATED IN OPTSYSX:

   1.   OPEN LOOP TIME RESPONSE
         XDOT =  {F}*X  +  {G}*UC

   2.   CLOSED LOOP TIME RESPONSE
         XDOT =  {F-G*C}*X +  {G}*UC,     U =   {C}*X

   3.   OPTIMIZED FILTER CLOSED LOOP SYSTEM RESPONSE.
         XDOT =  {F}*X  +  {G}*UC,         Z =  {H}*X
         XHDOT =  {F}*XH  +{G} U  +  {K}*{Z  -  H*XH}

   4.   FILTER + REGULATOR CLOSED LOOP SYSTEM RESPONSE.
         XDOT =  {F+G*C}*X  +  {G}*UC,                 Z =  {H}*X
         XHDOT =  {F}*XH  +{G} U  +  {K}*{Z  -  H*XH},    U =  {C}*XH

           SELECT 1, 2, 3 OR 4.
?
4


                 THE (G*C) MATRIX

     0.0             0.0
  -199.97670       -15.40159

                 THE (K*H) MATRIX

    95.40000         0.0
  4561.00000         0.0

           THE COMBINED SYSTEM F MATRIX (2*NS X 2*NS)

     0.0             1.00000        0.0           0.0
     0.0            -4.60000    -199.97670    -15.40159
```

```
   95.40000      0.0        -95.40000     1.00000
 4561.00000      0.0      -4760.97670   -20.00159

              THE AUGMENTED G MATRIX (2*NS X NC)

    0.0
    0.78700
    0.0
    0.78700


         AT WHAT TIME DO YOU WANT TO START
         THE TIME RESPONSE CALCULATIONS?

         INPUT START TIME IN SECONDS. (NORMALLY 0.0)
?
0


         AT WHAT TIME DO YOU WANT TO STOP
         THE TIME RESPONSE CALCULATIONS?

           INPUT STOP TIME IN SECONDS.
?
0.6

     THIS PROGRAM DIVIDES THE TIME INTERVAL YOU HAVE
     JUST SPECIFIED INTO UP TO 500 SMALL INTERVALS FOR
     THE INTEGRATION AND PLOTTING ROUTINES.  IN ORDER
     TO SAVE COMPUTER TIME, THE NUMBER OF POINTS CAN BE
     CAN BE REDUCED WITH SOME LOSS IN CURVE FIDELITY.

       HOW MANY POINTS DO YOU WANT TO CALCULATE?
?
500

DOES THE SYSTEM UTILIZE A DRIVING FUNCTION (CONTROL INPUT)?

                         (Y)ES OR (N)O
n

DOES THE SYSTEM START WITH ALL INITIAL CONDITIONS = 0.0 ?

                      (Y)ES OR (N)O?
n

         WHAT IS THE INITIAL CONDITION FOR X( 1)  ?
?
0.1

         WHAT IS THE INITIAL CONDITION FOR XHAT( 1) ?
?
0

         WHAT IS THE INITIAL CONDITION FOR X( 2)  ?
?
0

         WHAT IS THE INITIAL CONDITION FOR XHAT( 2) ?
?
0

         THIS IS YOUR LAST OPPORTUNITY TO
         MAKE CHANGES IN THE FOLLOWING AREAS.

           1.  SELECT ANOTHER TYPE OF SYSTEM TO PLOT
```

(OPEN, CLOSED, FILTER OR FILTER/REGULATOR)

2. START AND STOP TIMES
3. DRIVING FUNCTIONS
4. INITIAL CONDITIONS
5. CONTINUE

SELECT A NUMBER BETWEEN 1 AND 5.

?
5

THE FOLLOWING INFORMATION IS PROVIDED ONLY
FOR AN INDICATION OF PROPER PROGRAM OPERATION.

ALL CONTROLS, STATES AND STATE ESTIMATES CAN BE PLOTTED.

TIME	U(1)	X(1)	X(2)	X(3)
0.0	0.0	0.1000000D+00	0.0	0.0
0.01	0.0	0.9816154D-01	-0.4069147D+00	0.8100300D-01
0.02	0.0	0.8962622D-01	-0.9917358D+00	0.1042014D+00
0.04	0.0	0.7542258D-01	-0.1323812D+01	0.9379455D-01
0.05	0.0	0.5899229D-01	-0.1375381D+01	0.7124587D-01
0.06	0.0	0.4310993D-01	-0.1252435D+01	0.4867715D-01
0.07	0.0	0.2920410D-01	-0.1060393D+01	0.3055684D-01
0.08	0.0	0.1767465D-01	-0.8635309D+00	0.1722332D-01
0.10	0.0	0.8388221D-02	-0.6887453D+00	0.7571909D-02
0.11	0.0	0.1034654D-02	-0.5412268D+00	0.4310799D-03
0.12	0.0	-0.4696796D-02	-0.4175739D+00	-0.4997717D-02
0.13	0.0	-0.9063199D-02	-0.3129904D+00	-0.9154016D-02
0.14	0.0	-0.1227031D-01	-0.2238930D+00	-0.1226195D-01
0.16	0.0	-0.1448987D-01	-0.1481346D+00	-0.1445494D-01
0.17	0.0	-0.1587379D-01	-0.8444929D-01	-0.1584467D-01
0.18	0.0	-0.1656121D-01	-0.3190280D-01	-0.1654535D-01
0.19	0.0	-0.1668037D-01	0.1041880D-01	-0.1667477D-01
0.20	0.0	-0.1634808D-01	0.4351168D-01	-0.1634779D-01
0.22	0.0	-0.1566875D-01	0.6844137D-01	-0.1567017D-01
0.23	0.0	-0.1473380D-01	0.8629163D-01	-0.1473518D-01
0.24	0.0	-0.1362184D-01	0.9811538D-01	-0.1362266D-01
0.25	0.0	-0.1239918D-01	0.1049015D+00	-0.1239950D-01
0.26	0.0	-0.1112072D-01	0.1075564D+00	-0.1112077D-01
0.28	0.0	-0.9831066D-02	0.1068970D+00	-0.9831012D-02
0.29	0.0	-0.8565537D-02	0.1036486D+00	-0.8565474D-02
0.30	0.0	-0.7351302D-02	0.9844669D-01	-0.7351261D-02
0.31	0.0	-0.6208421D-02	0.9184092D-01	-0.6208403D-02
0.32	0.0	-0.5150854D-02	0.8429932D-01	-0.5150849D-02
0.34	0.0	-0.4187403D-02	0.7621404D-01	-0.4187405D-02
0.35	0.0	-0.3322593D-02	0.6790754D-01	-0.3322596D-02
0.36	0.0	-0.2557462D-02	0.5963914D-01	-0.2557464D-02
0.37	0.0	-0.1890282D-02	0.5161190D-01	-0.1890283D-02
0.38	0.0	-0.1317192D-02	0.4397938D-01	-0.1317192D-02
0.40	0.0	-0.8327506D-03	0.3685230D-01	-0.8327506D-03
0.41	0.0	-0.4304142D-03	0.3030484D-01	-0.4304141D-03
0.42	0.0	-0.1029384D-03	0.2438053D-01	-0.1029383D-03
0.43	0.0	0.1572873D-03	0.1909772D-01	0.1572873D-03
0.44	0.0	0.3579670D-03	0.1445443D-01	0.3579670D-03
0.46	0.0	0.5066818D-03	0.1043279D-01	0.5066818D-03
0.47	0.0	0.6107220D-03	0.7002862D-02	0.6107220D-03
0.48	0.0	0.6769634D-03	0.4125983D-02	0.6769634D-03
0.49	0.0	0.7117801D-03	0.1757615D-02	0.7117801D-03
0.50	0.0	0.7209884D-03	-0.1502632D-03	0.7209884D-03
0.52	0.0	0.7098176D-03	-0.1647187D-02	0.7098176D-03
0.53	0.0	0.6829017D-03	-0.2782618D-02	0.6829017D-03
0.54	0.0	0.6442879D-03	-0.3604693D-02	0.6442879D-03
0.55	0.0	0.5974586D-03	-0.4159284D-02	0.5974586D-03
0.56	0.0	0.5453629D-03	-0.4489304D-02	0.5453629D-03

```
     0.58   0.0  0.4904551D-03-0.4634245D-02 0.4904551D-03
     0.59   0.0  0.4347377D-03-0.4629897D-02 0.4347377D-03
     0.60   0.0  0.3798064D-03-0.4508230D-02 0.3798064D-03

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

         IF YOU ARE DISSATISFIED WITH THE RESULTS
         THUS FAR AND WOULD LIKE TO EXIT TO CMS,

                -TYPE 'Y' TO EXIT-

            (ANY OTHER INPUT TO CONTINUE)

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

B (120) R/O
C (121) R/O
E (122) R/O

          ... Your Fortran program is now being loaded ...
                ... execution will soon follow ...
EXECUTION BEGINS...

          THIS PORTION OF THE PROGRAM PLOTS:
               - THE STATES,
               - EXTERNAL CONTROL INPUTS,
               - FEEDBACK CONTROL INPUTS,
               - STATE ESTIMATES AND
               - RECONSTRUCTION ERRORS
          FROM THE DATA THAT YOU JUST CALCULATED.

          THE CAPABILITY IS ALSO AVAILABLE TO REVIEW ANY
          GRAPHS THAT YOU HAD PREVIOUSLY SAVED AS DATA
          FILES ON YOUR DISK.

               CLEAR THE SCREEN TO CONTINUE.

          THE FOLLOWING OPTIONS ARE AVAILABLE:

               1. PLOT THE DATA YOU JUST CALCULATED.
               2. PLOT A CURVE THAT YOU PREVIOUSLY SAVED.

          ENTER 1 OR 2
?
1

          YOU MAY PLOT UP TO 4 SYSTEM VARIABLES VS TIME.
HOW MANY VARIABLES DO YOU WISH TO PLOT?
?
3

WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 1?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC)

          ENTER 1,2,3,4 OR 5
?
1
```

```
          WHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
          YOU WANT TO PLOT AS THE NUMBER  1 CURVE VS TIME?
?
1

      WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

      NOTE:  1. 40 CHARACTERS MAX LENGTH
             2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
                ENCLOSED IN PARENTHESES.
                IE. (A) => ALPHA
                    (B) => BETA
                    (F) => PHI
                    (Q) => THETA
angular position - (x)1

WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 2?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC)

          ENTER 1,2,3,4 OR 5
?
4

          WHAT IS THE SUBSCRIPT OF THE STATE ESTIMATE THAT
          YOU WANT TO PLOT AS THE NUMBER  2 CURVE VS TIME?
?
1

     WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

     NOTE:  1. 40 CHARACTERS MAX LENGTH
            2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
               ENCLOSED IN PARENTHESES.
               IE. (A) => ALPHA
                   (B) => BETA
                   (F) => PHI
                   (Q) => THETA
angular position estimate - (x)e1

WHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS CURVE NUMBER 3?

   1. STATE VARIABLE (IE., X1, X2, ETC)
   2. FEEDBACK CONTROL (IE., U = -C*X)
   3. CONTROL INPUT (IE., U1, U2, ETC.)
   4. STATE ESTIMATE (OBSERVER) (IE., XHAT1, XHAT2, ETC.)
   5. STATE RECONSTRUCTION ERROR (IE., X1-XHAT1,
      X2-XHAT2, ETC)

          ENTER 1,2,3,4 OR 5
?
2

          WHAT IS THE SUBSCRIPT OF THE FEEDBACK CONTROL THAT
          YOU WANT TO PLOT AS THE NUMBER 3 CURVE VS TIME?
?
```

```
1

    WHAT IS THE CURVE LABEL FOR THIS VARIABLE?

    NOTE:  1. 40 CHARACTERS MAX LENGTH
           2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
              ENCLOSED IN PARENTHESES.
              IE. (A) => ALPHA
                  (B) => BETA
                  (F) => PHI
                  (Q) => THETA
input voltage - v

          YOU MAY USE UP TO 3 HEADINGS.
          HOW MANY HEADINGS DO YOU DESIRE ON THIS GRAPH?

                    0, 1, 2 OR 3
?
3

    WHAT IS THE DESIRED HEADING NUMBER 1?

    NOTE:  1. 40 CHARACTERS MAX LENGTH
           2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
              ENCLOSED IN PARENTHESES.
              IE. (A) => ALPHA
                  (B) => BETA
                  (F) => PHI
                  (Q) => THETA
filter + regulator closed loop system

    WHAT IS THE DESIRED HEADING NUMBER 2?

    NOTE:  1. 40 CHARACTERS MAX LENGTH
           2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
              ENCLOSED IN PARENTHESES.
              IE. (A) => ALPHA
                  (B) => BETA
                  (F) => PHI
                  (Q) => THETA
example 5.1

    WHAT IS THE DESIRED HEADING NUMBER 3?

    NOTE:  1. 40 CHARACTERS MAX LENGTH
           2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS
              ENCLOSED IN PARENTHESES.
              IE. (A) => ALPHA
                  (B) => BETA
                  (F) => PHI
                  (Q) => THETA
linear optimal control systems
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.

     THE FOLLOWING OPTIONS ARE AVAILABLE.

1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
3. EDIT THE CURRENT GRAPH.
4. PLOT REVISED GRAPH ON THE TEK618.
5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
   PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
3
```

```
                THE GRAPH EDIT MENU

     1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
     2. DELETE CURVE FROM CURRENT PLOT.
     3. EDIT CURVE TITLE(S).
     4. EDIT PAGE HEADING(S).
     5. CHANGE THE Y-AXIS SCALE.
     6. CHANGE THE TIME AXIS SCALE.
     7. CHANGE PLOT SIZE. (DEFAULT IS 8.5 X 6.0)
     8. CHANGE THE LETTERING HEIGHT.
     9. CHANGE POSITICN OF THE LEGEND.
    10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.
?
9

          HOW MANY INCHES IN THE X DIRECTION
          (LEFT OR RIGHT), DO YOU WANT TO MOVE
          MOVE THE LEGEND BOX FROM ITS PRESENT POSITION

          NOTE: 1. DEFAULT PLOT SIZE IS  8.5 X 6.0
                2. LEFT IS NEGATIVE
                3. RIGHT IS POSITIVE
?
0

          HOW MANY INCHES IN THE Y DIRECTION
          (UP OR DOWN), DO YOU WANT TO MOVE
          MOVE THE LEGEND BOX FROM ITS PRESENT POSITION

          NOTE: 1. DEFAULT PAGE SIZE IS  8.5 X 6.0
                2. DOWN IS NEGATIVE
                3. UP IS POSITIVE
?
2

                THE GRAPH EDIT MENU

     1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
     2. DELETE CURVE FROM CURRENT PLOT.
     3. EDIT CURVE TITLE(S).
     4. EDIT PAGE HEADING(S).
     5. CHANGE THE Y-AXIS SCALE.
     6. CHANGE THE TIME AXIS SCALE.
     7. CHANGE PLOT SIZE. (DEFAULT IS 8.5 X 6.0),
     8. CHANGE THE LETTERING HEIGHT.
     9. CHANGE POSITICN OF THE LEGEND.
    10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.
?
5

      ON WHICH CURVE DO YOU WANT TO CHANGE THE Y-SCALE?

              ENTER CURVE NUMBER- 1, 2, 3, OR 4
?
1

          WHAT IS THE NEW Y-MIN VALUE AT THE ORIGIN?
?
-.025

          WHAT IS THE NEW Y-MAX VALUE?
?
0.125

                THE GRAPH EDIT MENU
```

```
     1. CHANGE VARIABLES OR ADD A CURVE ON THE CURRENT PLOT.
     2. DELETE CURVE FROM CURRENT PLOT.
     3. EDIT CURVE TITLE(S).
     4. EDIT PAGE HEADING(S).
     5. CHANGE THE Y-AXIS SCALE.
     6. CHANGE THE TIME AXIS SCALE.
     7. CHANGE PLOT SIZE. (DEFAULT IS 8.5 X 6.0),
     8. CHANGE THE LETTERING HEIGHT.
     9. CHANGE POSITION OF THE LEGEND.
    10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.
?
10

       THE FOLLOWING OPTIONS ARE AVAILABLE.

 1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
 2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
 3. EDIT THE CURRENT GRAPH.
 4. PLOT REVISED GRAPH ON THE TEK618.
 5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
    PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
4

     THE FOLLOWING OPTIONS ARE AVAILABLE.

 1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
 2. REPLOT PREVIOUSLY SAVED GRAPH DATA.
 3. EDIT THE CURRENT GRAPH.
 4. PLOT REVISED GRAPH ON THE TEK618.
 5. QUIT AND/OR MAKE METAFILE OF THE CURVES.
    PREVIOUSLY SAVED.

SELECT A NUMBER BETWEEN 1 AND 5.
?
5

          DO YOU WANT TO SAVE THE CURRENT GRAPH DATA TO
          BE USED LATER TO GENERATE A METAFILE?

                     Y OR N

    NOTE: A METAFILE IS REQUIRED FOR SMOOTH VERSATEC PLOTS.
    THERE WILL BE AN OPPORTUNITY TO GENERATE A METAFILE
    JUST BEFORE EXITING THIS PROGRAM.
y

     WHAT FILE NAME DO YOU WANT THE CURVE DATA STORED UNDER?
                     (8 CHARACTERS MAX)
filtereg

          THE CURVE DATA IS BEING FILED UNDER FILTEREG DATA


END OF DISSPLA 9.0 -- 26332 VECTORS GENERATED IN 2 PLOT FRAMES
    PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CA.
 5020 VIRTUAL STORAGE REFERENCES; 9 READS; 0 WRITES.

          THE FOLLOWING OPTIONS ARE AVAILABLE:

          1.  MAKE METAFILE OF PREVIOUSLY SAVED CURVE.
          2.  QUIT.
```

```
                ENTER 1 OR 2
?
1

          WHAT FILE NAME IS THE DATA STORED UNDER?
filtereg

THE CURVE DATA IS BEING LOADED FROM FILE FILTEREG DATA
>> USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.

          THE FOLLOWING OPTIONS ARE AVAILABLE:

          1.  MAKE METAFILE OF PREVIOUSLY SAVED CURVE.
          2.  QUIT.

                ENTER 1 OR 2
?
2
END OF DISSPLA 9.0 -- 13201 VECTORS GENERATED IN 1 PLOT FRAMES
    PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CA.
2772 VIRTUAL STORAGE REFERENCES; 9 READS; 0 WRITES.

DASD121 DETACHED
DASD 122 DETACHED
DASD 120 DETACHED

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

     DO YOU WANT A VRSTEC PLOTTER SMOOTH COPY OF THE
       THE DISSPLA METAFILE THAT YOU JUST CREATED?
                          (Y OR N)

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

y
B (120) R/O
DASD 001 LINKED R/O; R/W BY MVS; R/O BY 0700P
Z (001) R/C - OS
DASD 001 DETACHED
CREATING NEW FILE:
CREATING NEW FILE:
PUN FILE 8317  TO  MVS      COPY 001   NOHOLD
DASD 120 DETACHED

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

     YOUR GRAPH(S) CAN BE PICKED UP AT THE COMPUTER CENTER.

       THE GRAPH(S) WILL BE ADDRESSED TO "POP (USER ID)".

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

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

                   DO YOU WANT TO

          1. RUN OPTSYSX AGAIN
          2. RUN THE PLOT PROGRAM USING THE SAME MATRICES?
             (TO PLOT ANOTHER TYPE OF SYSTEM (OPEN/CLOSED))
          3. QUIT

                     ENTER 1, 2 OR 3
```

```
****************************************************************

3

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

                    HAVE A GOOD DAY!!

****************************************************************
R; T=19.00/31.53 09:37:38
record off
END RECORDING OF TERMINAL SESSION
```

The graphical output generated by this example follows as figure 3.4.

FILTER + REGULATOR CLOSED LOOP SYSTEM
EXAMPLE 5.1
LINEAR OPTIMAL CONTROL SYSTEMS

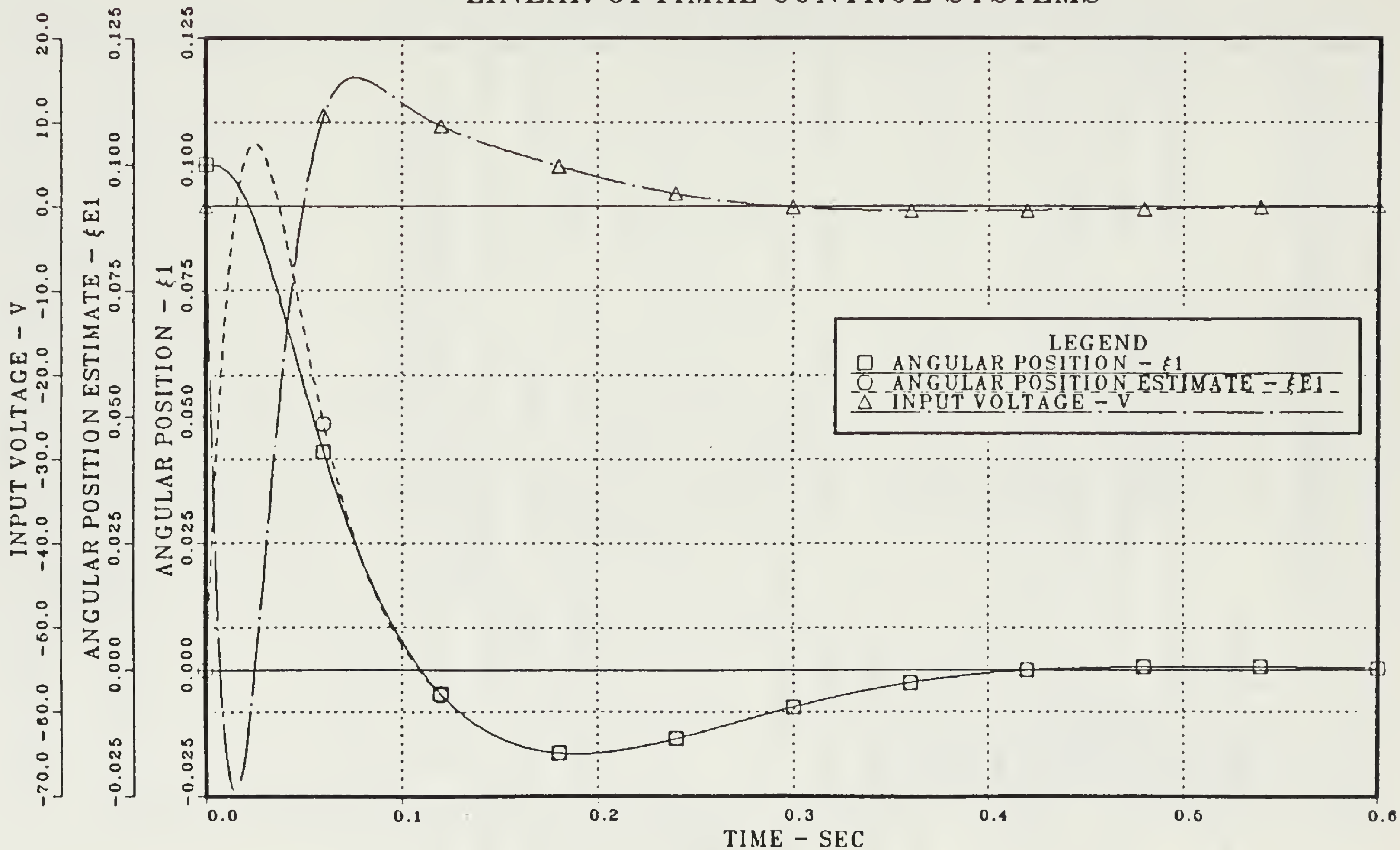

Figure 3.4   Filter plus Regulator Closed-loop Time Response

IV. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

As an ultimate evaluation of the computational abilities of OPTSYSX, the program was tested using an 82 X 82 matrix of aircraft longitudinal motion equations for the X-29A experimental forward-swept wing Fighter aircraft prototype, provided by NASA-Edwards.

For a system of equations of this magnitude, all program arrays were re-dimensioned, and a 2-megabyte virtual machine size was required.

The graphical time response curves generated from the X-29A longitudinal system matrix follow as figures 4.1, 4.2, 4.3 and 4.4. The accuracy of these time response curves is mixed. All of the states shown have the correct waveforms, but differ in a scale factor of approximately times 10.0. Unfortunately the data supplied by NASA was not explicit regarding how the control input was applied, and whether any additional gains were used in their simulation of the system. Time constrains did not allow the clarification of these problem areas, however the results of the X-29A longitudinal system are encouraging and should be a topic of further research.

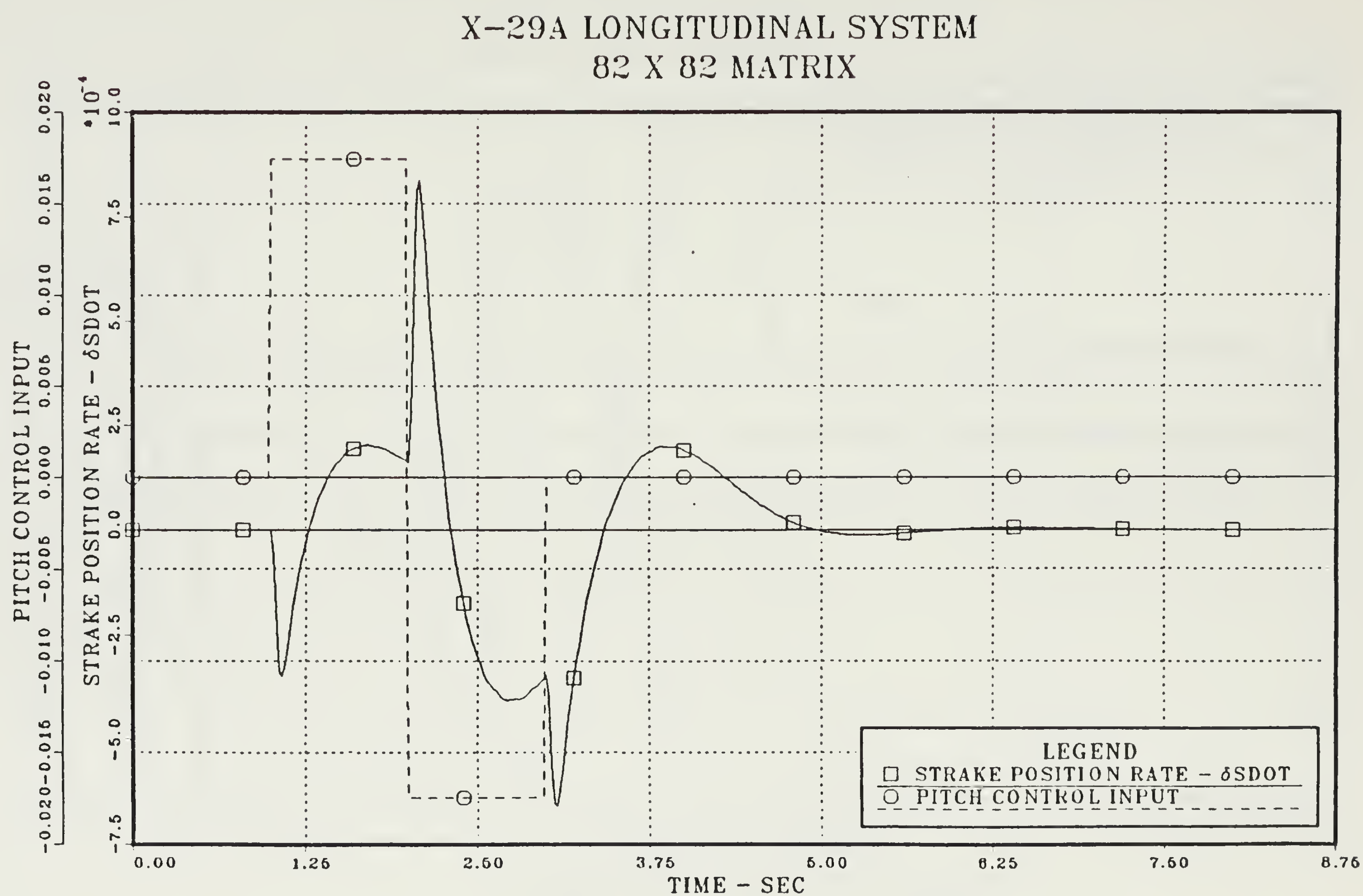

Figure 4.1 X-29A Longitudinal Time Response

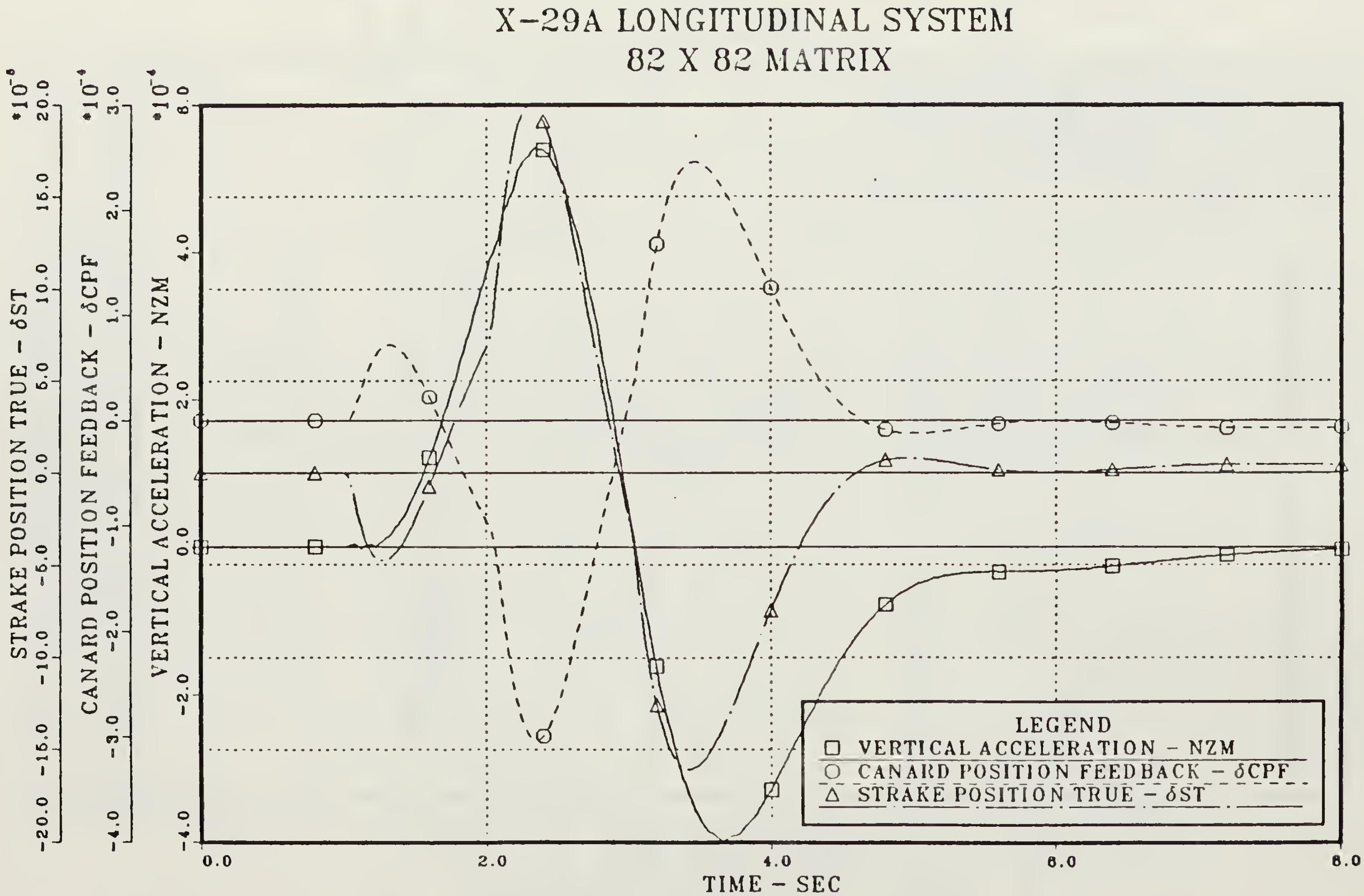

Figure 4.2 X-29A Longitudinal Time Response

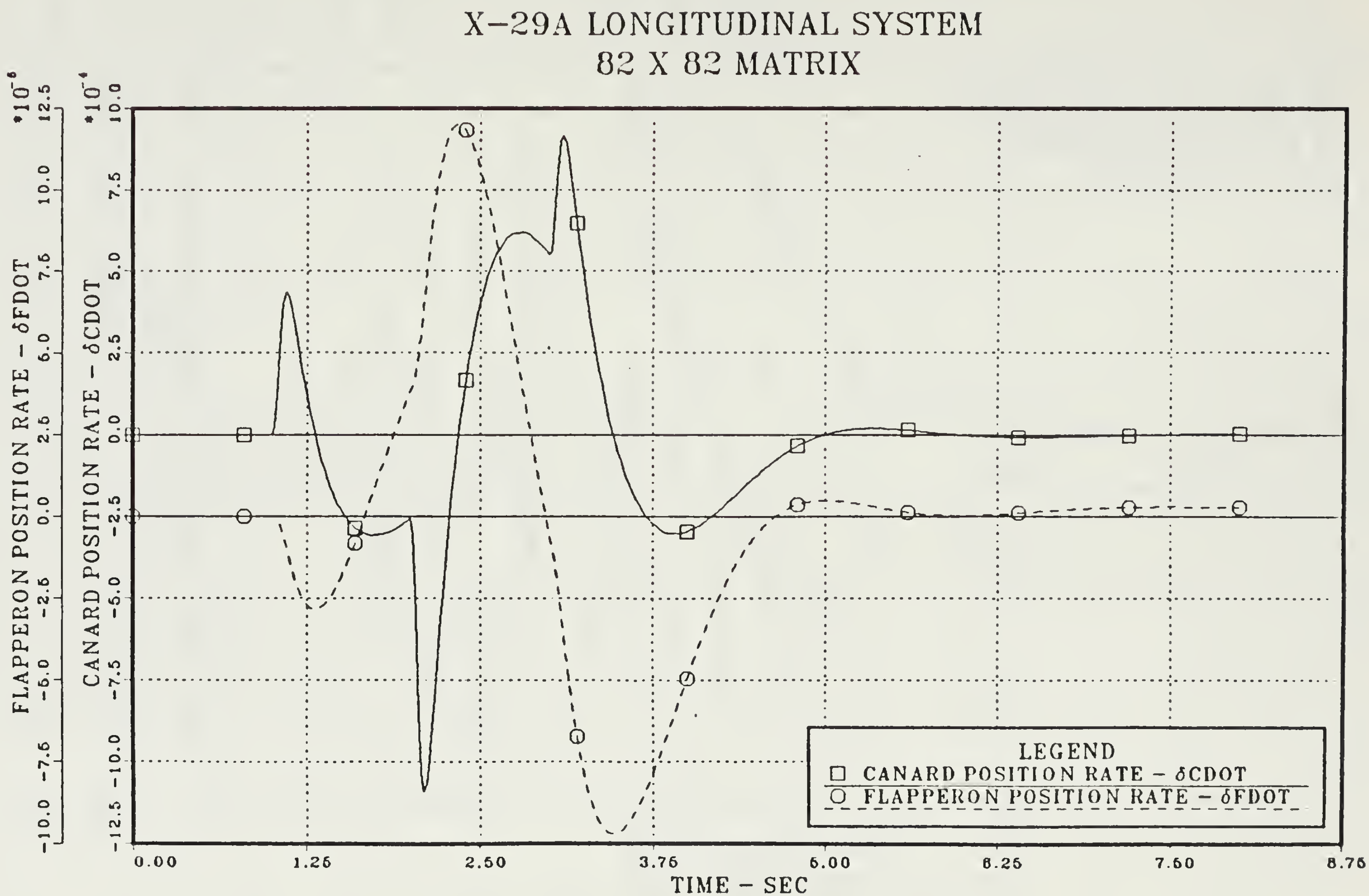

Figure 4.3 X-29A Longitudinal Time Response

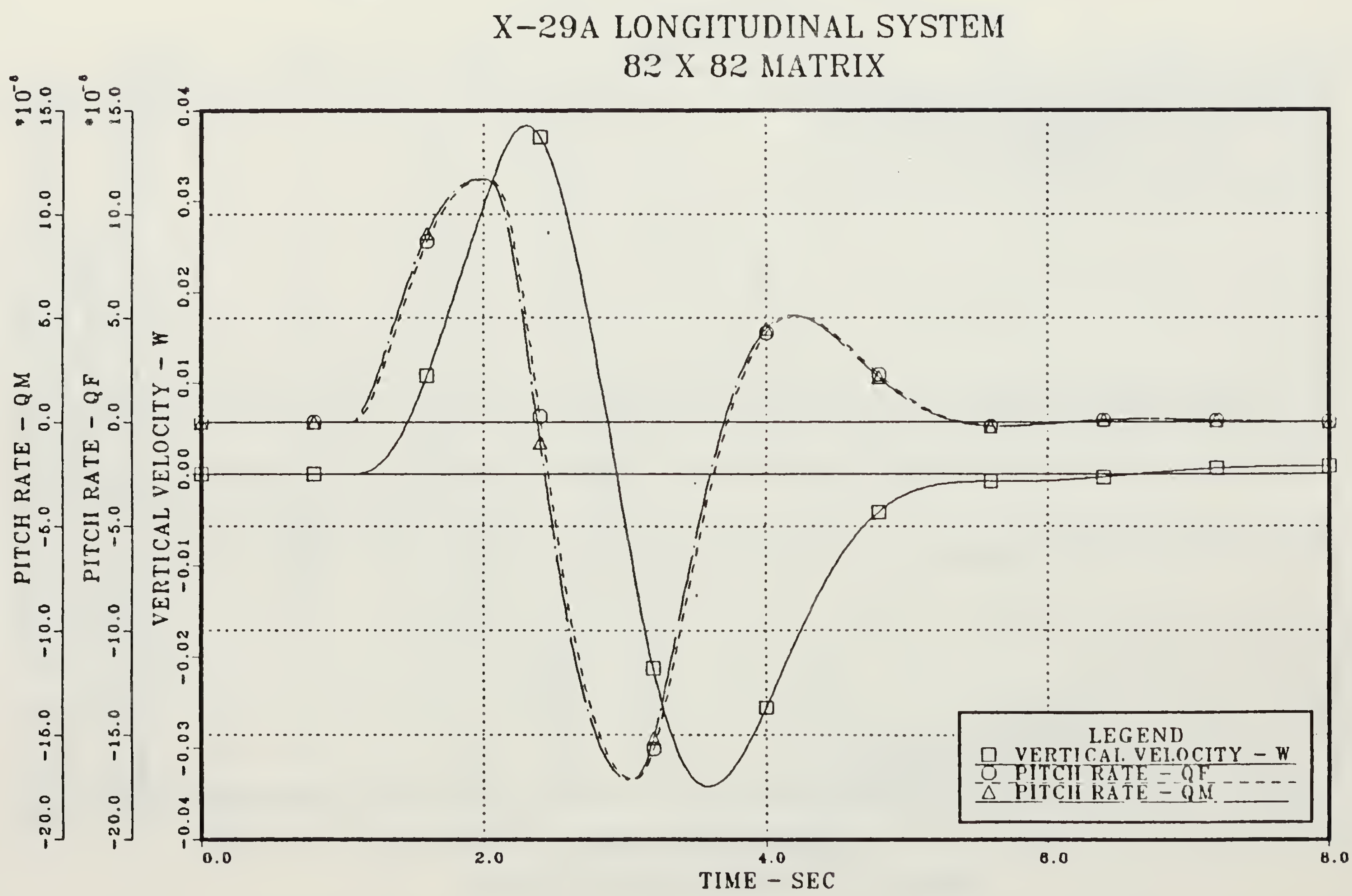

Figure 4.4 X-29A Longitudinal Time Response

It is hoped that control system instructors will encourage their students to use this interactive graphical time response program for all applicable class projects; and that its enhanced capabilities will stimulate both interest in and research on basic systems control problems, as well as more advanced designs.

B. RECOMMENDATIONS

Based on the results of this thesis, three areas emerged as possibilities for further research and study:

1. Program Availability

The use of OPTSYSX and similar design programs should be encouraged in all undergraduate and graduate level courses involved in the analysis and design of control systems. Toward this end, it is recommended that the OPTSYS time response programs be placed in the non-IMSL library of subroutines, making it easily available to all potential users.

2. Program Memory Requirements

When configured for large matrix operations (98X98) the OPTSYSX program requires over 2 megabytes of virtual memory. Virtual machines of this size are not normally available to a user. The memory usage for matrix storage is a possible area of improvement in the efficiency of the OPTSYSX program design. All matrices calculated in OPTSYSX (except DUMMY matrices) are still available when the run is finished. This simplifies program operation but uses an excessive amount of memory. Memory usage should be studied and program modifications should be made to reduce the excessive memory requirements.

3. Further Modifications

Program sequencing during optimal filter synthesis should be modified. At the present time a regulator must be designed or supplied when a filter is designed. Various test runs indicate that this problem can be overridden if the number of controls (Nc) is given as zero, but this is not a viable solution for systems which use a driving function.

APPENDIX A. OPTSYS EXEC LISTING

```
&TRACE OFF
*******************************************************************
*
*                    THE OPTSYS EXEC
*
*          CONTROLS THE OPTSYSX, OPTCALC AND OPTPLOT
*          TO DETERMINE THE TIME RESPONSE OF A SYSTEM.
*
*                      BY H. A. DIEL
*                 VERSION 1.0   16 JULY 1984
*
*******************************************************************
*******************************************************************
*                                                                 *
*       CHECK FOR USER'S VM SIZE = > THAN 1 MEGBYTE               *
*                                                                 *
*******************************************************************
VMSIZE
&IF &RC GE 1024 &GOTO -TWO
CLRSCRN
&BEGTYPE -ENDTHREE

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

        YOU MUST HAVE A 1M OR LARGER VIRTUAL MACHINE
               TO RUN THIS OPTSYS PROGRAM

             TO DEFINE A 1M VIRTUAL MACHINE:

          DEFINE STORAGE 1M          (PRESS ENTER)
          I CMS                      (PRESS ENTER)
          OPTSYS                     (PRESS ENTER)

             FOR SYSTEMS LARGER THAN 32 X 32
          OBTAIN A LISTING OF THE OPTSYS PROGRAM
     AND FOLLOW INSTRUCTIONS CONTAINED IN THE LISTING.

*******************************************************************
-ENDTHREE
&EXIT &RC
-TWO
CLRSCRN
&BEGTYPE -ENDZERO

   THE OPTSYS EXEC CONTROLS A TRIO OF PROGRAMS:

      1. OPTSYSX FORTRAN  (SYSTEM ANALYSIS)
      2. OPTCALC FORTRAN  (CALCULATE TIME RESPONSE)
      3. OPTPLOT FORTRAN  (DISSPLA PLOTTING ROUTINE)

   EACH PROGRAM PASSES INFORMATION TO THE NEXT PROGRAM
   THROUGH A DATA FILE WRITTEN TO THE USERS DISK.  IN THIS
   CASE, THESE FILES ARE "OPTMAT DATA" AND "OPTPLOT DATA".
   THE SIZE OF THESE FILES VARY WITH THE SYSTEM ORDER, AND
   CAN USE ABOUT 20% OF THE USERS DISK SPACE.  THEREFORE
   ENSURE THAT SUFFICIENT DISK SPACE IS AVAILABLE.
```

```
     - TYPE "E" TO EXIT, ANY OTHER ENTRY TO CONTINUE -
-ENDZERO
&READ VARS &ANS
&IF .&ANS EQ .E &EXIT &RC
*****************************************************************
*                                                               *
*  ALLOW THE USE OF AN OLD "OPTMAT DATA A1"                     *
*                                                               *
*****************************************************************
RENAME OPTMAT DATA A1 OPTSYS DATA A1
&IF &RC NE 0 &GOTO -FIRST
RENAME OPTSYS DATA A1 OPTMAT DATA A1
CLRSCRN
&BEGTYPE -ENDONE


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

  YOU HAVE A DATA FILE NAMED 'OPTMAT DATA' ON YOUR A
  DISK  THAT WAS PREVIOUSLY GENERATED BY THE  OPTSYS
  PROGRAM  AND  CONTAINS THE F, G, H, GAMMA, A AND B
  MATRICES FROM THAT RUN.

  IF  YOU WOULD LIKE TO USE THESE SAME MATRICES  FOR
  THIS  RUN, THE  OPTSYS  PROGRAM  WILL  READ IN THE
  DESIRED DATA AT THE APPROPRIATE TIME,

                 IF YOU TYPE (Y)ES.

  ANY OTHER INPUT WILL RESULT IN THAT FILE BEING ERASED!

*****************************************************************
-ENDONE
&READ VARS &ANS
&IF .&ANS EQ .Y &GOTO -ONE
-FIRST
*****************************************************************
*                                                               *
*  ERASE THE OLD "OPTMAT DATA A1" DATA FILE                     *
* PLACE "000 0" IN THE NEW "OPTMAT DATA FILE"                   *
* TO ACT AS A FLAG FOR OPTSYSX  AND OPTCALC                     *
*                                                               *
*****************************************************************
ERASE OPTMAT DATA A1
&STACK 000 0
FILESTCK OPTMAT DATA A1 F 80 1
-ONE
-THIRD
CLRSCRN
&BEGTYPE -ENDFOUR



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

   DO YOU WANT THE NUMERICAL OUTPUT FROM OPTSYSX TO GO
     TO YOUR TERMINAL S(CREEN) OR TO A D(ISK) FILE?
                    (S OR D)

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

-ENDFOUR
&READ VARS &ANS
&IF .&ANS EQ .S &GOTO -FOURTH
```

```
&IF .&ANS EQ .D &GOTO -FIFTH
CLRSCRN
&BEGTYPE -ENDFIVE




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

           YOU MUST ANSWER S(CREEN) OR D(ISK).

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

-ENDFIVE
CP SLEEP 3 SEC
&GOTO -THIRD
-FOURTH
CLRSCRN
&BEGTYPE -ENDSIX




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

        OUTPUT WILL COME TO YOUR TERMINAL SCREEN.

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

-ENDSIX
CP SLEEP 1 SEC
&TYPE LOADING OPTSYS....
FILEDEF 06 TERM (RECFM FA BLKSIZE 133
FILEDEF 8 DISK OPTPLOT DATA A1 (PERM
FILEDEF 9 DISK OPTMAT DATA A1 (PERM
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLDP  NONIMSL
LOAD OPTSYSX (START
&GOTO -FIVE
-FIFTH
CLRSCRN
&BEGTYPE -ENDSEVEN




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

     OUTPUT WILL GO TO DISK FILE 'OUTPUTX LISTING A1'

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

-ENDSEVEN
CP SLEEP 1 SEC
&TYPE LOADING OPTSYS....
FILEDEF 06 DISK OUTPUTX LISTING A1
FILEDEF 8 DISK OPTPLOT DATA A1 (PERM
FILEDEF 9 DISK OPTMAT DATA A1 (PERM
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLDP  NONIMSL
LOAD OPTSYSX (START
-FIVE
```

```
CLRSCRN
&BEGTYPE -ENDEIGHT




*****************************************************************
      IF YOU ARE DISSATISFIED WITH THE RESULTS
      THUS FAR AND WOULD LIKE TO EXIT TO CMS,

             -TYPE 'Y' TO EXIT-

          (ANY OTHER INPUT TO CONTINUE)
*****************************************************************
-ENDEIGHT
&READ VARS &ANS
&IF .&ANS EQ .Y &EXIT &RC
-SIXTH
*****************************************************************
*                                                               *
*        CHECK FOR DATA IN THE FILE "OPTMAT DATA "              *
*                 BEFORE LOADING OPTCALC                        *
*                                                               *
*****************************************************************
FINDSTAK OPTMAT DATA A1 C01 0 LIM002 ALL GROUP1
&READ VARS &A1 &A2
&IF .&A2 EQ .0 &EXIT &RC
&TYPE LOADING OPTCALC....
FILEDEF 8 DISK OPTPLOT DATA A1 (PERM
FILEDEF 9 DISK OPTMAT DATA A1 (PERM
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLDP  NONIMSL
LOAD OPTCALC (START
-SEVENTH
CLRSCRN
&BEGTYPE -ENDNINE



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


      IF YOU ARE DISSATISFIED WITH THE RESULTS
      THUS FAR AND WOULD LIKE TO EXIT TO CMS,

             -TYPE 'Y' TO EXIT-

          (ANY OTHER INPUT TO CONTINUE)

*****************************************************************
-ENDNINE
&READ VARS &ANS
&IF .&ANS EQ .Y &EXIT &RC
FILEDEF 8 DISK OPTPLOT DATA A1 (PERM
EXEC DISSPLA OPTPLOT
-EIGHTH
*****************************************************************
*                                                               *
*        CHECK FOR FILE "DISSPLA METAFILE A4" ON                *
*        THE USER'S DISK BEFORE GOING TO DISSPOP                *
*                                                               *
*****************************************************************
RENAME DISSPLA METAFILE A4 OPTSYS METAFILE A4
&IF &RC NE 0 &GOTO -TENTH
RENAME OPTSYS METAFILE A4 DISSPLA METAFILE A4
```

```
CLRSCRN
&BEGTYPE -ENDTEN




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

    DO YOU WANT A VRSTEC PLOTTER SMOOTH COPY OF THE
      THE DISSPLA METAFILE THAT YOU JUST CREATED?
                         (Y OR N)

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

-ENDTEN
&READ VARS &ANS
&IF .&ANS EQ .Y &GOTO -NINTH
&IF .&ANS EQ .N &GOTO -TENTH
CLRSCRN
&BEGTYPE -ENDELEVEN




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

               YOU MUST ANSWER Y(ES) OR N(O).

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

-ENDELEVEN
CP SLEEP 4 SEC
&GOTO -EIGHTH
-NINTH
EXEC DISSPOP VRSTEC
CLRSCRN
&BEGTYPE -ENDTWELVE




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

   YOUR GRAPH(S) CAN BE PICKED UP AT THE COMPUTER CENTER.

     THE GRAPH(S) WILL BE ADDRESSED TO "POP (USER ID)".

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

-ENDTWELVE
CP SLEEP 5 SEC
-TENTH
CLRSCRN
&BEGTYPE -ENDTHIRTEEN



******************************************************************
```

```
                 DO YOU WANT TO

        1. RUN OPTSYSX AGAIN
        2. RUN THE PLOT PROGRAM USING THE SAME MATRICES?
          (TO PLOT ANOTHER TYPE OF SYSTEM (OPEN/CLOSED))
        3. QUIT

                   ENTER 1, 2 OR 3

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

-ENDTHIRTEEN
&READ VARS &ANS
&IF .&ANS EQ .1 &GOTO -THIRD
&IF .&ANS EQ .2 &GOTO -SIXTH
CLRSCRN
&BEGTYPE -ENDGOODBY



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


                 HAVE A GOOD DAY!!

*************************************************************
-ENDGOODBY
CP SLEEP 3 SEC
CLRSCRN
&EXIT &RC
```

```
C***********************************************************************
C***********************************************************************
C************                                               ************
C**********                  CPTSYSX                          **********
C********               BY JOHN G. HODEN                        ********
C********                                                       ********
C********     THIS PROGRAM IS A COMPLETELY INTERACTIVE          ********
C********     OPTIMAL SYSTEMS CONTROL DESIGN/SYNTHESIS          ********
C********     PROGRAM CAPABLE OF HANDLING VERY LARGE (80X80)+   ********
C********     MULTIVARIABLE SYSTEMS OF LINEAR EQUATIONS.        ********
C********                                                       ********
C**********         VERSION 1.8   11 MAR 1984                 **********
C************         MODIFIED BY H. A. DIEL                ************
C*************             17 JULY 1984                    *************
C***********************************************************************
C***********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
      INTEGER IANS,IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,ICEBUG,ISET,
     1IPSC,IYL,INCRM,NS,NC,NCB,NG,IREG,ICSTAB,IRET,NROW,NCOL,ISAF,ISAH,I
     2SAG,IGAM,IRDMAT
C-----------------------------------------------------------------------
C     INCREASED DIMENSIONS -- LARGE ORDER SYSTEM.
C-----------------------------------------------------------------------
C     DIMENSION ACL(82,82),B(41,41),BA(82,82),CI(82),CR(82),CC(82,82),
C    *CWI(82),CWR(82),FBGC(41,82),FBGE(82,41),G(82,82),GM(82,82),
C    *PRO(82,82),RC(41,41),SC(82,82),WR(164),WI(164),W11(82,82),
C    *W21(82,82),X(164,164),GN(82,82),HO(41,82),D1(164),D2(164),
C    *RM(164,164),Q(41,41),GAM(82,41),WNCRM(82,82),WNORMI(82,82),
C    *DESTAB(82),AA(82,82),BM(82,41),CM(41,82),D(41,41),DSTURE(82,82),
C    *JCF(164),RES(164),AY(82,82),BB(164),CC(164),CP(82),GW(164,41),
C    *GV(164,41),FY(41,164),HU(41,164),PRTT(16,16),DUM(82,85)
C-----------------------------------------------------------------------
C     STANDARD PROGRAM DIMENSIONS.
C-----------------------------------------------------------------------
      DIMENSION ACL(32,32),B(32,32),BA(32,32),CI(32),CR(32),CC(32,32),CW
     1I(32),CWR(32),FBGC(32,32),FBGE(32,32),G(32,32),GM(32,32),PRO(32,32
     2),RC(32,32),SC(32,32),WR(64),WI(64),W11(32,32),W21(32,32),X(64,64)
     3,GN(32,32),HO(32,32),D1(64),D2(64),RM(64,64),Q(32,32),GAM(32,32),W
     4NORM(32,32),WNORMI(32,32),DESTAB(32),AA(32,32),BM(32,32),CM(32,32)
     5,D(32,32),DSTORE(32,32),JCF(64),RES(64),AY(32,32),BB(64),CC(64),CP
     6(32),GW(64,64),GV(64,64),HY(64,64),HU(64,64),PRTT(16,16),DUM(32,32
     7)
C-----------------------------------------------------------------------
      EQUIVALENCE (W11(1,1),GW(1,1)), (W11(1,1),GV(1,1)), (W21(1,1),HY(1
     1,1)), (W21(1,1),HU(1,1))
C-----------------------------------------------------------------------
      COMMON /PROG/ IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,ICEB
```

```
     1UG,ISET,IREG,IPSD,IYU,INORM
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-----------------------------------------------------------------------
      CALL ERRSET (207,256,-1,1,1,209)
      CALL ERRSET (215,256,-1,1)
C-----------------------------------------------------------------------
C   INITIALIZE SAVE FLAGS.                                              =
C-----------------------------------------------------------------------
      ISAF=0
      ISAG=0
      ISAH=0
      IGAM=0
      ISAA=0
      ISAB=0
      ISET=0
C--------------------------------------------------SCRN1-----------------
10    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,640)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 20
      GO TO 30
20    WRITE (5,1000)
      GO TO 10
30    CONTINUE
      IF (IANS.EQ.IZ) GO TO 630
C--------------------------------------------------SCRN2-----------------
40    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,650)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
      GO TO 60
50    WRITE (5,1000)
      GO TO 40
60    CONTINUE
      IF (IANS.EQ.IZ) GO TO 630
C--------------------------------------------------ISET------------------
70    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,660)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 80
      GO TO 90
80    WRITE (5,1000)
      GO TO 70
90    CONTINUE
```

```
      IF (IANS.EQ.IY) ISET=1
C-----------------------------------------------------------------------
C  INITIALIZE SYSTEM FLAGS.                                             =
C-----------------------------------------------------------------------
100   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
      IPSC=0
      IYL=0
      INCRM=0
      IREG=0
      NS=0
      NC=0
      NOB=0
      NG=0
      IRDMAT=0
C------------------------------------------------------IOL-------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,670)
      CALL RDINT (IANS)
      IOL=IANS-1
      IF (IOL.EQ.2) GO TO 350
C------------------------------------------------------IQ--------------------
      CALL FRTCMS ('CLRSCRN ')
110   WRITE (5,680)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 120
      GO TO 130
120   WRITE (5,1000)
      GO TO 110
130   CONTINUE
      IF (IANS.EQ.IY) IQ=1
      IF (IANS.EQ.IZ) IQ=0
      IF (IOL.EQ.3) GO TO 200
C------------------------------------------------------IR--------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,690)
```

```
      CALL RDINT (IANS)
      IR=IANS-1
C---------------------------------------------------ISS--------------------
      CALL FRTCMS ('CLRSCRN ')
140   WRITE (5,700)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 150
      GO TO 160
150   WRITE (5,1000)
      GO TO 140
160   CONTINUE
      IF (IANS.EQ.IY) ISS=1
      IF (IANS.EQ.IZ) ISS=0
C---------------------------------------------------IM---------------------
170   WRITE (5,710)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 180
      GO TO 190
180   WRITE (5,1000)
      GO TO 170
190   CONTINUE
      IF (IANS.EQ.IY) IM=1
      IF (IANS.EQ.IZ) IM=0
200   CONTINUE
      IF (IOL.EQ.3) IM=1
C---------------------------------------------------ITF1-------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,720)
      CALL RDINT (IANS)
      ITF1=IANS-1
C     IF (IOL.EQ.3) GO TO 240
C---------------------------------------------------ITF2-------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,730)
      CALL RDINT (IANS)
      ITF2=IANS-1
C     IF (IOL.EQ.3) GO TO 240
C---------------------------------------------------ITF3-------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,740)
      CALL RDINT (IANS)
      ITF3=IANS-1
C---------------------------------------------------IFDFW------------------
      CALL FRTCMS ('CLRSCRN ')
210   WRITE (5,750)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 220
      GO TO 230
```

```
220   WRITE (5,1000)
      GO TO 210
230   CONTINUE
      IF (IANS.EQ.IY) IFDFW=1
      IF (IANS.EQ.IZ) IFDFW=0
C--------------------------------------------------IE-------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,760)
      CALL RDREAL (ANSR)
      IE=IDINT(ANSR)
      IF (IOL.EQ.3) GO TO 300
C--------------------------------------------------IDSTAB----------------
      CALL FRTCMS ('CLRSCRN ')
240   WRITE (5,770)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 250
      GO TO 260
250   WRITE (5,1000)
      GO TO 240
260   CONTINUE
      IF (IANS.EQ.IY) IDSTAB=1
      IF (IANS.EQ.IZ) IDSTAB=0
C--------------------------------------------------IDEBUG----------------
270   WRITE (5,780)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 280
      GO TO 290
280   WRITE (5,1000)
      GO TO 270
290   CONTINUE
      IF (IANS.EQ.IY) IDEBUG=1
      IF (IANS.EQ.IZ) IDEBUG=0
300   CONTINUE
C--------------------------------------------------IPSD------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,790)
      CALL RDINT (IANS)
      IPSD=IANS
      IF (IPSD.EQ.3) IPSD=0
      IF (IPSD.EQ.0) GO TO 310
C--------------------------------------------------IYU-------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,800)
      CALL RDINT (IANS)
      IYU=IANS-1
C--------------------------------------------------INORM-----------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,920)
```

```
      CALL RDREAL (ANSR)
      INORM=IDINT(ANSR)
310   IF (IOL.EQ.3) GO TO 350
C----------------------------------------------------IREG------------------
      CALL FRTCMS ('CLRSCRN ')
320   WRITE (5,810)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 330
      GO TO 340
330   WRITE (5,1000)
      GO TO 320
340   CONTINUE
      IF (IANS.EQ.IY) IREG=1
      IF (IANS.EQ.IZ) IREG=0
350   CALL RDMATF (NS,NC,NOB,NG,ISAF,ISAG,ISAH,IGAM,ISAA,ISAB,IRDMAT)
      IF ((ISAF.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 360
C----------------------------------------------------NS--------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,820)
      CALL RDREAL (ANSR)
      NS=IDINT(ANSR)
360   IF (IOL.EQ.2) GO TO 390
      IF ((ISAG.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 370
C----------------------------------------------------NC--------------------
      WRITE (5,830)
      CALL RDREAL (ANSR)
      NC=IDINT(ANSR)
370   IF ((ISAH.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 380
C----------------------------------------------------NOB-------------------
      WRITE (5,840)
      CALL RDREAL (ANSR)
      NOB=IDINT(ANSR)
380   IF ((IGAM.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 390
C----------------------------------------------------NG--------------------
      WRITE (5,850)
      CALL RDREAL (ANSR)
      NG=IDINT(ANSR)
390   CONTINUE
C----------------------------------------------------FLAG SETTINGS---------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,860)
      WRITE (6,870)
      WRITE (6,880) IOL,IC,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDEBUG,ISET
     1,ICSTAE
      WRITE (6,890)
      WRITE (6,900) IPSD,IYU,INORM,IREG,NS,NC,NOB,NG
      WRITE (6,910) NS,NC,NOB,NG
C----------------------------------------------------BEGIN CALCULATIONS-----
```

```
      N2=2*NS
      CALL INNER (NS,NC,NCB,NG,N2,ACL,B,BA,CI,CR,CC,CWI,CWR,D,FBGC,FBGE,
     1G,GAM,GM,GN,HO,D1,C2,PRO,RM,RC,Q,SC,WR,WI,W11,W21,X,WNORM,WNORMI,D
     2ESTAB,AA,BM,CM,JCF,RES,AY,BB,CC,CP,GW,GV,HY,HU,DSTORE,ISAF,ISAH,IS
     3AG,IGAM,IRET,PRTT,NROW,NCOL,IRDMAT,ISAA,ISAE)
C-----------------------------------------------------------------------
      IF (IRET.EQ.1) GO TO 400
      CALL WRTMAT (BA,G,HC,GAM,FBGC,FBGE,AY,B,NS,NC,NOB,NG)
C---------------------------------------------------IRET-----------------
400   WRITE (5,930)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 410
      GO TO 420
410   WRITE (5,1000)
      GO TO 400
420   CONTINUE
      IF (IANS.EQ.IY) GO TO 430
      IF (IANS.EQ.IZ) GO TO 630
C---------------------------------------------------ISAF-----------------
430   CONTINUE
      IF (IRET.EQ.1) GO TO 100
      IF (ISET.EQ.1) GO TO 100
      CALL FRTCMS ('CLRSCRN ')
440   WRITE (5,940)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 450
      GO TO 460
450   WRITE (5,1000)
      GO TO 440
460   CONTINUE
      IF (IANS.EQ.IY) ISAF=1
      IF (IANS.EQ.IZ) ISAF=0
C---------------------------------------------------ISAH-----------------
      IF (NOB.EQ.0) GO TO 500
      CALL FRTCMS ('CLRSCRN ')
470   WRITE (5,950)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 480
      GO TO 490
480   WRITE (5,1000)
      GO TO 470
490   CONTINUE
      IF (IANS.EQ.IY) ISAH=1
      IF (IANS.EQ.IZ) ISAH=0
500   CONTINUE
C---------------------------------------------------ISAG-----------------
      IF (NC.EQ.0) GO TO 540
      CALL FRTCMS ('CLRSCRN ')
```

```
510   WRITE (5,560)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 520
      GO TO 530
520   WRITE (5,1000)
      GO TO 510
530   CONTINUE
      IF (IANS.EQ.IY) ISAG=1
      IF (IANS.EQ.IZ) ISAG=0
540   CONTINUE
C---------------------------------------------------------IGAM------------------
      IF (NG.EQ.0) GO TO 580
      CALL FRTCMS ('CLRSCRN ')
550   WRITE (5,570)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 560
      GO TO 570
560   WRITE (5,1000)
      GO TO 550
570   CONTINUE
      IF (IANS.EQ.IY) IGAM=1
      IF (IANS.EQ.IZ) IGAM=0
580   CONTINUE
C---------------------------------------------------------ISAA------------------
      CALL FRTCMS ('CLRSCRN ')
590   WRITE (5,580)
      CALL RDCHAR (IANS)
      IF ((IANS.EQ.IY).OR.(IANS.EQ.IZ)) GO TO 600
      WRITE (5,1000)
      GO TO 590
600   CONTINUE
      IF (IANS.EQ.IY) ISAA=1
      IF (IANS.EQ.IZ) ISAA=0
C---------------------------------------------------------ISAB------------------
      CALL FRTCMS ('CLRSCRN ')
610   WRITE (5,590)
      CALL RDCHAR (IANS)
      IF ((IANS.EQ.IY).OR.(IANS.EQ.IZ)) GO TO 620
      WRITE (5,1000)
      GO TO 610
620   CONTINUE
      IF (IANS.EQ.IY) ISAB=1
      IF (IANS.EQ.IZ) ISAB=0
      GO TO 100
C---------------------------------------------------------TERMINATE-------------
630   WRITE (5,1010)
      STOP
C-------------------------------------------------------------------------------
```

```
640   FORMAT (5X,59HOPTSYSX IS A COMPLETELY INTERACTIVE OPTIMAL SYSTEMS
     1CONTROL,/,8X,55HPROGRAM. IT WILL SCLVE NUMEROUS CONTROL PRCBLEMS O
     2N THE,/,8X,45HFCLLCWING TYPES CF SYSTEMS CCNTROL EQUATIONS:,//,15X
     3,35HXDOT =  F&*X +  G&*U +  GAM&*(W+WO),//,20X,22HMEASUREMENT ECUA
     4TICN--,//,15X,21HZ =  H&*X +  D&*U + V,//,2CX,29HREGULATOR PERFCRM
     5ANCE INDEX--,//,15X,42HJ = 1/2 * INTEGRAL (Y9* A&*Y + U9* B&*U)DT,
     6//,20X,32HSTATE FEEDBACK GAIN DEFINITICN--,//,25X,10HU = - C&*X,//
     7,15X,45HDC YOU WISH TO CONTINUE?  TYPE "YES" OR "NO".)
650   FORMAT (25X,14H--DATA ENTRY--,//,5X,49HALTHCUGH OPTSYSX IS SPECIFI
     1CALLY DESIGNED TO READ,/,5X,48HALL MATRIX DATA INTERACTIVELY, SEVE
     2RAL ALTERNATE,/,5X,31HMETHCDS ARE AVAILABLE TO USERS:,//,10X,43HME
     3THOD 1--THE "F","G", AND "GAMMA" MATRICES,/,13X,37HMAY BE READ FRO
     4M SEPARATE CATA FILES.,//,10X,50HMETHOD 2--THE "F","G", AND "GAMMA
     5" MATRICES MAY BE,/,13X,45HEXPLICITLY DEFINED WITHIN SUBROUTINE "S
     6ETUP".,//,10X,52H NOTE: IN EITHER CASE, THE USER SHCULD OBTAIN A C
     7OPY,/,17X,34HOF THE PROGRAM LISTING AND EXAMINE,/,17X,39HTHE EXAMP
     8LES CONTAINED IN S/R "SETUP".&,//,10X,45HDO YOU WISH TO CONTINUE?
     9 TYPE "YES" OR "NO".)
660   FORMAT (//,5X,46HDC YCU WISH TC INPUT THE "F", "G", AND "GAMMA",/,
     110X,40HMATRICES FRCM SUBROUTINE "SETUP" IAW THE,/,10X,40HMETHOD DE
     2SCRIBED CN THE PREVIOUS SCREEN?,//,15X,19HTYPE "YES" OR "NO".)
670   FORMAT (25X,24HGENERAL OPTSYSX CPTICNS:,//,10X,35HOPTION 1 -- SYST
     1EM ANALYSIS WITHOUT,/,22X,35HOPEN-LCOP EIGENSYSTEM CALCULATIONS.,/
     2/,10X,42HCPTION 2 -- SYSTEM ANALYSIS WITH OPEN-LOOP,/,22X,25HEIGEN
     3SYSTEM CALCULATIONS.,//,10X,39HCPTICN 3 -- CPEN-LOOP EIGENSYSTEM F
     4CUND,/,22X,23HAND PRCGRAM TERMINATES.,/,22X,39H "F"-MATRIX ENTRY F
     5OLLCWS IMMEDIATELY.&,//,10X,48HCPTICN 4 -- MODAL DISTRIBUTION MATR
     6ICES CCMPUTED,/,22X,37HWITHOUT FILTER CR REGULATOR SYNTHESIS,/,22X
     7,25HOR STEADY-STATE ANALYSIS.,//,15X,30HSELECT AN OPTION: 1,2,3, O
     8R 4.)
680   FORMAT (//,5X,46HDC YCU DESIRE RMS VALUES OF STATE AND CONTROL?,//
     1,10X,19HTYPE "YES" CR "NO".)
690   FORMAT (/,2CX,30HOPTSYSX LCR/CLASSICAL CPTICNS:,//,10X,43HCPTION 1
     1 -- OPTIMAL FILTER AND/CR REGULATOR,/,22X,37HSYNTHESIS WITH NO EXT
     2ERNAL "C" OR "K",/,22X,13HMATRIX INPUT.,//,10X,43HOPTION 2 -- OPTI
     3MAL FILTER AND/OR REGULATOR,/,22X,27HSYNTHESIS WITH EXTERNAL "C",/
     4,22X,13HMATRIX INPUT.,//,10X,43HOPTION 3 -- OPTIMAL FILTER AND/OR
     5REGULATCR,/,22X,27HSYNTHESIS WITH EXTERNAL "K",/,22X,13HMATRIX INP
     6UT.,//,10X,43HOPTICN 4 -- OPTIMAL FILTER AND/OR REGULATCR,/,22X,35
     7HSYNTHESIS WITH EXTERNAL "C" AND "K",/,22X,13HMATRIX INPUT.,//,10X
     8,32HSELECT AN OPTION: 1, 2, 3, CR 4.)
700   FORMAT (//,5X,50HDC YCU WISH TC DETERMINE THE STEADY-STATE RESPCNS
     1E,/,8X,27HFCR A CONSTANT DISTURBANCE?,//,10X,19HTYPE "YES" OR "NO"
     2.)
710   FORMAT (5X,47HDO YCU WISH TO DETERMINE THE MODAL DISTRIBUTION,/,8X
     1,18HAND GAIN MATRICES?,//,10X,19HTYPE "YES" OR "NO".)
720   FORMAT (//,5X,36HOPEN-LOOP TRANSFER FUNCTION OPTIONS:,//,10X,53HOP
     1TICN 1 -- NC OPEN-LCOP TRANSFER FUNCTICNS CCMPUTED.,//,10X,48HOPTI
```

```
     2ON 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.,//,10X,42HOPTION 3 --
     3 ONLY POLES AND ZEROS COMPUTED.,//,10X,45HOPTION 4 -- ONLY POLES A
     4ND RESIDUES COMPUTED.,//,10X,32HSELECT AN OPTION: 1, 2, 3, OR 4.)
730  FORMAT (//,5X,32HNOISE TRANSFER FUNCTION OPTIONS:,//,10X,49HOPTION
     1 1 -- NO NOISE TRANSFER FUNCTIONS COMPUTED.,//,10X,48HOPTION 2 --
     2POLES, RESIDUES, AND ZEROS COMPUTED.,//,10X,42HOPTION 3 -- ONLY PO
     3LES AND ZEROS COMPUTED.,//,10X,45HOPTION 4 -- ONLY POLES AND RESID
     4UES COMPUTED.,//,10X,32HSELECT AN OPTION: 1, 2, 3, OR 4.)
740  FORMAT (//,5X,38HCOMPENSATOR TRANSFER FUNCTION OPTIONS:,//,10X,49H
     1OPTION 1 -- NO COMP. TRANSFER FUNCTIONS COMPUTED.,//,10X,48HOPTION
     2 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.,//,10X,42HOPTION 3 -- O
     3NLY POLES AND ZEROS COMPUTED.,//,10X,45HOPTION 4 -- ONLY POLES AND
     4 RESIDUES COMPUTED.,//,15X,45H NOTE: A COMPENSATOR TRANSFER FUNCTI
     5ON CAN BE,/,22X,33HCOMPUTED ONLY IF BOTH A REGULATOR,/,22X,26HAND
     6FILTER ARE SYNTHESIZED,/,22X,14HAND/OR INPUT.&,//,10X,32HSELECT AN
     7 OPTION: 1, 2, 3, OR 4.)
750  FORMAT (//,5X,39HWILL A FEED-FORWARD DISTRIBUTION MATRIX,/,5X,25H
     1"D" - MATRIX& BE INPUT ?,//,15X,19HTYPE "YES" OR "NO".)
760  FORMAT (/,5X,63H THIS OPTION DETERMINES THE CRITERIA FOR DECIDING
     1WHEN A MARKOV,/,8X,58HPARAMETER IS ZERO-THE MARKOV PARAMETER INDIC
     2ATES THE ORDER,/,8X,54HOF THE NUMERATOR POLYNOMIAL OF EACH TRANSFE
     3R FUNCTION.,//,8X,52HALL "N" ZEROS OF THIS POLYNOMIAL ARE PRINTED
     4OUT AND,/,8X,52HTHIS TEST TELLS HOW MANY EXTRA ROOTS EXIST AT Z =
     50.,/,8X,41HLESS THAN 10.0** -IE& IS CONSIDERED ZERO.,//,8X,47H THE
     6 DEFAULT VALUE OF THIS PARAMETER  IE& IS 6.,/,8X,28HIN OTHER WORDS
     7, IE = 1.0E-6.,//,10X,66HIF YOU DESIRE A DIFFERENT MARKOV CRITERIA
     8, TYPE THE INTEGER VALUE.,//,10X,48HIF YOU DESIRE THE DEFAULT VALU
     9E, TYPE "0"  ZERO&)
770  FORMAT (//,5X,61HDO YOU DESIRE TO SYNTHESIZE A STABLE FILTER  OR R
     1EGULATOR& BY,/,8X,34HDESTABILIZING THE ORIGINAL SYSTEM?,//,12X,52H
     2 NOTE:WORKS FOR FILTER OR REGULATOR BUT NOT FOR BOTH,/,20X,17HIN T
     3HE SAME RUN.&,//,10X,19HTYPE "YES" OR "NO".)
780  FORMAT (5X,53HDO YOU DESIRE TO PRINT THE EULER-LAGRANGE EIGENSYSTE
     1M,/,8X,50HPRIOR TO DECOMPOSITION  FOR CHECKING THE PROGRAM&?,//,10
     2X,19HTYPE "YES" OR "NO".)
790  FORMAT (//,5X,39HPOWER SPECTRAL DENSITY  PSD& OPTION 1 :,//,10X,53
     1HOPTION 1 -- COMPUTE THE PSD OF THE OUTPUTS AND/OR THE,/,22X,48HCO
     2NTROLS OF THE CONTROLLED SYSTEM WHEN FORCED BY,/,22X,45HPROCESS AN
     3D MEASUREMENT NOISE.   NOTE: BOTH A,/,22X,46HREGULATOR AND A FILTE
     4R MUST BE RESIDENT IN THE,/,22X,28HPROGRAM TO USE THIS OPTION.&,//
     5,10X,53HOPTION 2 -- SAME AS OPTION 1 ABOVE BUT ONLY PRINT THE,/,22
     6X,34HRESIDUES OF EACH TRANSFER FUNCTION,/,22X,28HUSED IN THE PSD C
     7OMPUTATION.,//,10X,24HOPTION 3 -- NOT DESIRED.,///,10X,29HSELECT A
     8N OPTION: 1, 2, OR 3.)
800  FORMAT (//,5X,39HPOWER SPECTRAL DENSITY  PSD& OPTION 2 :,//,10X,35
     1HOPTION 1 -- PSD OUTPUT NOT DESIRED.,//,10X,38HOPTION 2 -- COMPUTE
     2 ONLY OUTPUT PSD/S.,//,10X,39HOPTION 3 -- COMPUTE ONLY CONTROL PSD
     3/S.,//,10X,50HOPTION 4 -- COMPUTE BOTH OUTPUT AND CONTROL PSD/S.,/
```

```
     4/,15X,32HSELECT AN CPTICN: 1, 2, 3, CR 4.)
810   FORMAT (//,5X,39HDO YOU DESIRE REGULATCR SYNTHESIS ONLY?,//,10X,19
     1HTYPE "YES" OR "NO".,//)
820   FORMAT (/,5X,47HENTER THE # OF STATES  NS& CF THE SYSTEM MATRIX,/,
     15X,13H "F"-MATRIX&.)
830   FORMAT (/,5X,56HENTER THE # OF CCNTRCLS  NC& OF THE CONTROL SYSTEM
     1 MCDEL,/,5X,13H "G"-MATRIX&.)
840   FORMAT (/,5X,54HENTER THE # OF MEASUREMENTS OR OBSERVATIONS  NO& O
     1FTHE,/,5X,13H "H"-MATRIX&.)
850   FORMAT (/,5X,48HENTER THE # OF PROCESS NOISE SOURCES  NG& OF THE,/
     1,5X,17H "GAMMA"-MATRIX&.)
860   FORMAT (5X,52HFLAG/PARAMETER SETTINGS FCR THIS RUN ARE AS FOLLOWS:
     1,/)
870   FORMAT (1X,3HIOL,2X,2HIC,2X,2HIR,2X,3HISS,2X,2HIM,2X,4HITF1,2X,4HI
     1TF2,2X,4HITF3,2X,5HIFDFW,2X,2HIE,2X,6HICEBUG,2X,4HISET,2X,6HIDSTAB
     2,/)
880   FORMAT (1X,I2,3X,I2,3X,I2,2X,I2,3X,I2,3X,I2,4X,I2,4X,I2,4X,I2,4X,I
     12,3X,I2,6X,I2,5X,I2,/)
890   FORMAT (1X,4HIPSD,2X,3HIYU,2X,5HINCRM,2X,4HIREG,2X,2HNS,2X,2HNC,2X
     1,3HNOB,2X,2HNG,/)
900   FORMAT (2X,I2,3X,I2,4X,I2,5X,I2,3X,I2,2X,I2,3X,I2,2X,I2,//)
910   FORMAT (2X,17HORDER OF SYSTEM =,I3,//,2X,20HNUMBER CF CCNTROLS =,I
     13,//,2X,24HNUMBER OF CBSERVATIONS =,I3,//,2X,33HNUMBER CF PROCESS
     2NOISE SCURCES =,I3,////)
920   FORMAT (5X,53HDETERMINE THE NORMALIZATICN PARAMETER  INCRM& FOR TH
     1E,/,5X,55HPCWER SPECTRAL DENSITY  PSD& CPTICN YOU HAVE PREVIOUSLY,
     2/,5X,52HCHOSEN. TWO PSD NORMALIZATICN METHODS ARE AVAILABLE:,//,10
     3X,54HMETHOD 1 -- PSD IS NORMALIZED BY THE I-NORM/TH PROCESS,/,21X,
     429HNOISE MINUS "Q" INORM,INORM&.,/,21X,49H NOTE: "Q" IS AN CPTIMAL
     5 STATE WEIGHTING MATRIX.&,/,21X,34HIN THIS METHOD, INORM =0,1,...,
     6NG.,//,10X,63HMETHOD 2 -- PSD IS NCRMALIZED BY THE  INORM -NG&/TH
     7MEASUREMENT,/,21X,39HNCISE MINUS "R" INCRM - NG,INORM - NG&.,/,21X
     8,51H NOTE: "R" IS AN CPTIMAL CCNTRCL WEIGHTING MATRIX.&,/,21X,44HI
     9N THIS METHCD, INORM = NG + 1,...,NG + NOB.,//,10X,51HSELECT AN IN
     $TEGER FRCM 0 - 16 REPRESENTING YOUR PSD,/,15X,27HNORMALIZATION REQ
     $UIREMENTS.,//,10X,53HIF PSD NORMALIZATICN IS NOT DESIREL ENTER "0"
     $  ZERC&.)
930   FORMAT (5X,43HANALYSIS COMPLETE. DC YOU WANT ANOTHER RUN?,/,15X,19
     1HTYPE "YES" OR "NO".)
940   FORMAT (///,5X,48HDC YCU WISH TC SAVE THE "F"-MATRIX FROM THE LAST
     1,/,5X,36HRUN TO BE USED IN THE FOLLCWING RUN?,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PRCPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YCU WILL HAVE THE OPTION CF CHANGING,/,5X,27HIND
     4IVICUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" CR "NO".)
950   FORMAT (///,5X,48HDC YCU WISH TO SAVE THE "F"-MATRIX FROM THE LAST
     1,/,5X,36HRUN TO BE USED IN THE FOLLCWING RUN?,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PRCPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YCU WILL HAVE THE CPTION CF CHANGING,/,5X,27HIND
```

```
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
960   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "G"-MATRIX FROM THE LAST
     1,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
970   FORMAT (///,5X,52HDO YOU WISH TO SAVE THE "GAMMA"-MATRIX FROM THE
     1LAST,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: T
     2HE MATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE
     3 INTERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27
     4HINDIVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
980   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "A"-MATRIX FROM THE LAST
     1,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
990   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "E"-MATRIX FROM THE LAST
     1,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
1000  FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY   ENTER "YES" OR "NO".)
1010  FORMAT (//,41H........CPTSYSX IS NOW TERMINATED........,//)
      END
C=====================================================================
      SUBROUTINE SETUP (EA,G,GAM,NS,NC,NG)
C=====================================================================
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BA(NS,NS),G(NS,NC),GAM(NS,NG),DUM(82,85)
      COMMON /PROG/ IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,ICEB
     1UG,ISET,IREG,IPSD,IYU,INORM
C---------------------------------------------------------------------
C  FILE DEFINITIONS
C---------------------------------------------------------------------
      CALL FRTCMS ('FILEDEF ','03      ','DISK    ','X29A82  ',
     1             'DATA    ','A       ')
C---------------------------------------------------------------------
C  THIS IS AN EXAMPLE OF A 82 X 85 DATA FILE X29A82 DATA A1 READ FROM
C  A USER'S DISK AND CONVERTED (FROM A "DUMMY" ARRAY NAMED 'DUM') TO A
C  SYMMETRIC ARRAY.  THE FORMAT STATEMENT MUST MATCH YOUR DISK DATA
C  FORMAT OR THE PROGRAM WILL FAIL   NOTE: ALL PROGRAM DIMENSIONS
C  MUST BE ENLARGED ACCORDINGLY FOR A SYSTEM OF THIS SIZE.
C---------------------------------------------------------------------
      READ (3,50) ((DUM(I,J),J=1,85),I=1,NS)
      DO 20 I=1,NS
      DO 10 J=1,NS
      BA(I,J)=DUM(I,J)
10    CONTINUE
```

```
20     CONTINUE
C-----------------------------------------------------------------------
C  THESE ARE EXAMPLES OF SEVERAL POSSIBLE METHODS OF ARRAY GENERATION
C  WITHIN SUBROUTINE SETUP.THE "GAM" ARRAY WAS SET TO ZERO SINCE NO
C  "NOISE" WAS PRESENT, AND THE NON-ZERO ELEMENTS OF THE "G" ARRAY WERE
C  EXPLICITLY DEFINED. THEY COULD ALSO BE READ FROM FILES AS ABOVE.
C-----------------------------------------------------------------------
       DO 40 I=1,NS
       DO 30 J=1,NC
       GAM(I,J)=0.0D+00
       G(I,J)=0.0D+00
30     CONTINUE
40     CONTINUE
       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
       RETURN
C----------------------------------------------------------------------
50     FORMAT (5(E12.4))
       END
C=======================================================================
       SUBROUTINE CHECK (EPS,NC,NG,NO,IRET)
C      CHECKS THE CONSISTENCY OF REQUESTED OPTIONS.                   =
C=======================================================================
       DOUBLE PRECISION EPS
       COMMON /PROG/ IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEB
      1UG,ISET,IREG,IPSD,IYU,INORM
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 (NC .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)
      IRET=1
      RETURN
C------------------DESTABILIZATION RESTRICTIONS------------------------
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-----------------------------------------------------------------------
90    FORMAT (//,5X,49H H - MATRIX MUST BE INPUT, I.E. "NG" 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,48HREGULATOR AND FILTER SYNTHESIS MUST BE REQUESTED,
     1/,5X,44HIN THE SAME RUN TO COMPUTE COMPENSATOR T. F.,/,5X,47HI.E.
     2IREG MUST = 0.; "NC" AND "NG" MUST BE > 0.,//)
120   FORMAT (//,5X,51HNOISE T. F. CALCULATED ONLY WHEN REGULATOR DESIGN
     1ED,/,5X,47HI.E. IREG MUST = 1.; "NC" AND "NG" MUST BE > 0.,//)
130   FORMAT (//,5X,47HDESTABILIZATION OPTION DESIGNED FOR A REGULATOR,/
     1,5X,38HOR FILTER BUT NOT BOTH SIMULTANEOUSLY.,//,5X,55HIF "NG" > 0
     2. THE REGULATOR OPTION IS AUTOMATICALLY SET ,//)
```

```
140   FORMAT (//,5X,49H ********** INCONSISTENT PSD INPUT FLAGS ********
     1,//)
150   FORMAT (//,5X,44HBOTH A REGULATOR AND FILTER MUST BE RESIDENT,/,10
     1X,42HTO COMPUTE THE PSD OF A CONTROLLED SYSTEM ,/,10X,42HI.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,CW,CWI,CWR,D,FBGC,
     1FBGE,G,GAM,GM,GN,HO,D1,D2,PRO,RM,RC,C,SC,WR,WI,W11,W21,X,WNORM,WNO
     2RMI,DESTAB,AA,BM,CM,JCF,RES,AY,BB,CC,CP,GW,GV,HY,HU,DSTORE,ISAF,IS
     3AH,ISAG,IGAM,IRET,PRTT,NROW,NCOL,IRDMAT,ISAA,ISAB)
C=====================================================================
      IMPLICIT REAL*8(A-H,O-Z)
C---------------------------------------------------------------------
      DIMENSION ACL(NS,NS),B(NC,NC),BA(NS,NS),CI(NS),CR(NS),CW(NS,NS),CW
     1I(NS),CWR(NS),FBGC(NC,NS),FBGE(NS,NC),G(NS,NS),GM(NS,NS),PRO(NS,NS
     2),RC(NC,NO),SC(NS,NS),WR(N2),WI(N2),W11(NS,NS),W21(NS,NS),X(N2,N2)
     3,GN(NS,NS),HO(NO,NS),D1(N2),D2(N2),RM(N2,N2),Q(NG,NG),D(NO,NC),GAM
     4(NS,NG),WNORM(NS,NS),WNORMI(NS,NS),DESTAB(NS),AA(NS,NS),BM(NS,NC),
     5CM(NO,NS),JCF(N2),RES(N2),AY(NC,NO),BB(N2),CC(N2),CP(NS),GW(N2,NG)
     6,GV(N2,NC),HY(NO,N2),HU(NC,N2),DSTORE(NS,NS),PRTT(16,16)
C---------------------------------------------------------------------
      COMMON /PROC/ IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEB
     1UG,ISET,IREG,IPSD,IYU,INORM
C---------------------------------------------------------------------
      REAL*4 FMT(20)
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
      M=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,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG,IRDMAT)
      CALL READF (NS,ISAF,BA)
      IF (IDSTAB.EQ.0) GO TO 10
      WRITE (5,1800)
      CALL RDREAL (ANSR)
      DSTAB=ANSR
```

```
      DO 10 I=1,NS
      DESTAB(I)=DSTAB
10    CONTINUE
      GO TO 30
20    CALL SETUP (BA,G,GAM,NS,NG,NC)
30    CONTINUE
      WRITE (6,1380)
      DO 40 I=1,NS
40    WRITE (6,1390) (BA(I,J),J=1,NS)
      IF (IDSTAB.EQ.0) GO TO 50
      WRITE (6,1480)
      WRITE (6,1390) (DESTAB(I),I=1,NS)
50    CONTINUE
C-------------EIGENSYSTEM OF THE OPEN LOOP DYNAMICS------------------
      IF (ICL.EQ.0.AND.IC.EQ.0) GO TO 90
      IF (ICL.EQ.0.AND.NC.NE.0) GO TO 90
      DO 60 I=1,NS
      DO 60 J=1,NS
60    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 HQR2 (NS,NS,LOW,IHIGH,GN,CWR,CWI,SC,IERR)
      IF (IERR.NE.0) CALL EREXIT (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,NSC,DDD,D1,D2,WNORM,WNORMI,HO,CM,
     1NO,NS)
      IF (IOL.EQ.2) RETURN
      IF (IC.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 (ICL.EQ.3) GO TO 130
      DO 80 I=1,NS
      DO 80 J=1,NS
80    W11(I,J)=SC(I,J)
      CALL MINV (NSQ,W11,NS,DDD,D1,D2)
90    CONTINUE
      IF (IDSTAB.EQ.0) GO TO 130
C----------------- FORM U * DIAG(DESTAB) * U-INV---------------------
      DO 100 J=1,NS
      DO 100 I=1,NS
100   AA(I,J)=WNORM(I,J)*DESTAB(J)
      DO 120 I=1,NS
```

```
      DO 120 J=1,NS
      DDD=0.DC
      DO 110 K=1,NS
110   DDD=DDD+AA(I,K)*WNORMI(K,J)
      DSTORE(I,J)=DDD
120   BA(I,J)=BA(I,J)+DDD
130   CONTINUE
      IF (NO.EQ.0) GO TO 145
      CALL READH (NO,NS,ISAH,HO)
      WRITE (6,1440)
      DO 140 I=1,NO
140   WRITE (6,1390) (HO(I,J),J=1,NS)
145   IF (IM.NE.1) GO TO 150
      CALL MCDE (WNORM,HO,CM,NS,NC,NS,2)
150   CONTINUE
      IF (IFDFW.EQ.0) GO TO 170
      CALL READD (NO,NC,D)
      WRITE (6,1470)
      DO 160 I=1,NO
160   WRITE (6,1390) (D(I,J),J=1,NC)
170   CONTINUE
      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)
180   CONTINUE
      CALL READFB (NC,NS,FBGC)
      WRITE (6,1400)
      DO 190 I=1,NS
190   WRITE (6,1390) (G(I,J),J=1,NC)
      IF (IM.NE.1) GO TO 200
      CALL MCDE (WNORMI,G,BM,NS,NS,NC,0)
200   CONTINUE
      GO TO 330
210   DO 220 I=1,NS
      DO 220 J=1,NS
220   RM(I+MH,J)=0.0
      CALL READAY (NO,ISAA,AY)
      DO 240 I=1,NO
      DO 240 J=1,NS
      DDD=0.DC
      DO 230 K=1,NO
230   DDD=DDD+AY(I,K)*HO(K,J)
240   AA(I,J)=DDD
      WRITE (6,1460)
      DO 250 I=1,NO
```

```
250   WRITE (6,1390) (AY(I,J),J=1,NO)
      DO 260 I=1,NS
      DO 260 J=1,NS
      DO 260 K=1,NO
260   RM(I+MH,J)=RM(I+MH,J)+AA(K,I)*HC(K,J)
270   IF (ISET.EQ.1) GO TO 280
      CALL READG (NS,NC,ISAG,G)
280   CONTINUE
      IF (ICL.EQ.3) GO TO 290
      CALL READB (NC,ISAB,B)
290   WRITE (6,1400)
      DO 300 I=1,NS
300   WRITE (6,1390) (G(I,J),J=1,NC)
      IF (IM.NE.1) GO TO 310
      CALL MCDE (WNORMI,G,BM,NS,NS,NC,0)
310   CONTINUE
      IF (ICL.EQ.3) GO TO 340
      WRITE (6,1410)
      DO 320 I=1,NC
320   WRITE (6,1390) (B(I,J),J=1,NC)
330   IF (ITF1.EQ.0) GO TO 350
C-----------------OPEN LOOP TRANSFER FUNCTIONS-----------------------
340   WRITE (6,1500)
      ITFX=1
      CALL TF (NS,NS,NSQ,BA,AA,NC,G,BM,NC,HO,CM,IFDFW,D,BB,CC,CP,WR,WI,C
     1WR,CWI,SC,JCF,RES,D1,D2,DDD,EPS,ITF1,ITFX)
350   IF (ICL.NE.3) GO TO 360
      IF (NG.EQ.0) RETURN
      GO TO 600
360   CONTINUE
      IF (IR.EQ.1.OR.IR.EQ.3) GO TO 500
C---CALCULATION OF CONTROL GAINS:FORMATION OF CONTROL HAMILTONIAN-------
C
C    --                          --          ***F AND FT ARE THE OPEN LOOP
C   |                              |            DYNAMICS MATRIX AND TRANSPOSE
C   |      F         -GM*BI*GMT    |         ***BI IS NCXNC CONTROL WEIGHTING
C   |                              |            MATRIX
C   |                              |         ***A IS THE NSXNS STATE WEIGHTING
C   |                              |            MATRIX
C   |     -A             -FT       |
C    --                          --          ***GM IS THE NSXNC CONTROL
C                                               DISTRIBUTION MATRIX
C-----------------------------------------------------------------------
      DO 370 I=1,NC
      DO 370 J=1,MH
370   PRO(I,J)=G(J,I)/B(I,I)
      DO 380 I=1,MH
      DO 380 J=1,MH
```

```
      RM(I,J+MH)=0.D0
      DO 380 K=1,NC
380   RM(I,J+MH)=RM(I,J+MH)-G(I,K)*PRC(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,S,RM,4,'(9(1X,1PD13.6))')
410   CALL BALANC (M,M,RM,LOW,IHIGH,D1)
      CALL ORTHES (M,M,LOW,IHIGH,RM,D2)
      CALL ORTRAN (M,M,LOW,IHIGH,RM,D2,X)
      CALL HQR2 (M,M,LOW,IHIGH,RM,WR,WI,X,IERR)
      IF (IERR .NE. 0) CALL EREXIT (M,RM,IERR)
      CALL BALBAK (M,M,LOW,IHIGH,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
420   WRITE (6,1530) WR(I),WI(I)
      WRITE (6,1540)
      CALL RAPRNT (M,M,M,S,X,4,'(9(1X,1PD13.6))')
430   CONTINUE
      IF (IDSTAB .EQ. 1) GO TO 440
      IF (NOB.EQ.0) WRITE (6,1550)
      IF (NOB.NE.0) WRITE (6,1560)
440   IF (NOB.NE. 0) GO TO 750
      CALL RGAIN (M,NS,NC,NOB,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,S,SC,4,'(9(1X,1PD13.6))')
450   CONTINUE
C------RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-------
      IF (IDSTAB .EQ. 0) GO TO 470
      DO 460 I=1,NS
460   BA(I,I)=BA(I,I)-DESTAB(I)
      IR=1
470   CONTINUE
C--------------------CALCULATION OF FEEDBACK GAIN------------------------
```

```
C------------FEEDBACK GAINS--->  U = -(BINVERSE)*GT*GN&----------------
C-----------------------CALCULATE GT-----------------------------------
      DO 490 I=1,NC
      DO 490 J=1,NS
      PRO(I,J)=0.D0
      DO 480 K=1,MH
480   PRO(I,J)=PRO(I,J)+G(K,I)*GN(K,J)
490   FBGC(I,J)=-PRO(I,J)/B(I,I)
      IF (IDSTAB .EQ. 1) GO TO 500
C--------NORMALIZE AND PRINT OPT. REG. CLOSED LOOP EIGENSYSTEM---------
      IWRITE=2
      CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSC,DDD,D1,D2,WNORM,WNORMI,FBGC,
     1AA,NC,NS)
C---------------THE OPTIMUM FEEDBACK CONTROL GAINS---------------------
500   WRITE (6,1580)
      DO 510 I=1,NC
510   WRITE (6,1590) (FBGC(I,J),J=1,NS)
C------COMPUTE MODAL C MATRIX  OPEN LOOP U-INVERSE SAVED IN WNORMI&-----
      IF (IM .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
520   WNORM(I,J)=WNORMI(I,J)
      CALL MINV (NSQ,WNORM,NS,DDD,D1,D2)
      CALL MCLE (WNORM,FBGC,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.D0
      DO 540 K=1,NC
540   SUM=SUM+G(I,K)*FBGC(K,J)
550   ACL(I,J)=BA(I,J)+SUM
      WRITE (6,1600)
      CALL RAPRNT (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
560   GN(I,J)=ACL(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 HQR2 (NS,NS,LOW,IHIGH,GN,CWR,CWI,SC,IERR)
      IF (IERR .NE. 0) CALL EREXIT (NS,GN,IERR)
      CALL BALBAK (NS,NS,LOW,IHIGH,D1,NS,SC)
```

```
C-------NORMALIZE AND PRINT CLOSED LOOP SUBOPT. REG. EIGENSYSTEM--------
      IWRITE=3
      CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSC,DDC,D1,D2,WNORM,WNORMI,FBGC,
     1AA,NC,NS)
      DO 570 I=1,NS
      IF (CWR(I).LT.0.0) GO TO 570
      WRITE (5,1610)
      RETURN
570   CONTINUE
      IF (IQ.NE.1) GO TO 590
      DO 580 I=1,NS
      DO 580 J=1,NS
580   W11(I,J)=SC(I,J)
      CALL MINV (NSQ,W11,NS,DDD,D1,D2)
590   NOE=NC
      IF ( NG .EQ. 0) RETURN
600   IF (ISET.EQ.1) GO TO 610
      CALL READG2 (NS,NG,IGAM,GAM)
610   CONTINUE
      IF (ICL .EQ. 3) GO TO 620
      CALL READQ (NG,Q)
620   WRITE (6,1420)
      DO 630 I=1,NS
630   WRITE (6,1390) (GAM(I,J),J=1,NG)
      IF (IM.NE.1) GO TO 640
      CALL MCDE (WNORMI,GAM,AA,NS,NS,NG,1)
640   CONTINUE
      IF (ICL .EQ. 3) RETURN
      WRITE (6,1430)
      DO 650 I=1,NG
650   WRITE (6,1390) (Q(I,J),J=1,NG)
      IF ((IQ.EQ.0).AND.(NG.EQ.0)) GO TO 1260
      DO 660 I=1,NG
      DO 660 J=1,NS
      PRO(I,J)=0.D0
      DO 660 K=1,NG
660   PRO(I,J)=PRO(I,J)+Q(I,K)*GAM(J,K)
      DO 670 I=1,NS
      DO 670 J=1,NS
      CQ(I,J)=0.D0
      DO 670 K=1,NG
670   CQ(I,J)=CQ(I,J)-GAM(I,K)*PRO(K,J)
      IF (IREG .EQ. 1) GO TO 690
C---CALCULATION OF FILTER GAINS:FORMATION OF ESTIMATION HAMILTONIAN-----
C                                                                      =
C    T--                     --T        ***F AND FT ARE SAME AS FOR    =
C    |                         |           CONTROL HAMILTONIAN         =
C    |    F          -GM*Q*GMT |        ***Q IS NGXNG STATE DISTURBANCE =
```

```
C     |                                  |            RCOVARIANCE                   =
C     |                                  |         ***R IS NOXNO MEASUREMENT NOISE  =
C     |                                  |            RCOVARIANCE                   =
C     | -HCT*RIN*HC       -FT            |         ***HO IS NOXNS MEASUREMENT MATRIX=
C    ---                               ---         ***GM IS NSXNG STATE DISTURBANCE =
C                                                     DISTRIBUTION MATRIX           =
C-------------------------------------------------------------------------------------
      CALL READR (NO,RC)
      WRITE (6,1450)
      DO 680 I=1,NO
680   WRITE (6,1390) (RC(I,J),J=1,NO)
690   IF (ITF2 .EQ. 0) GO TO 700
C--------------------NOISE TRANSFER FUNCTIONS-----------------------------------------
      WRITE (6,1620)
      ITFX=2
      IZERO=0
      CALL TF (NS,NS,NSQ,ACL,AA,NG,GAM,BM,NO,HO,CM,IZERO,D,BB,CC,CP,WR,
     1WI,CWR,CWI,SC,JCF,RES,D1,D2,DDD,EPS,ITF2,ITFX)
      IF (IREG .EQ. 1) RETURN
700   CONTINUE
      IF (IREG .EQ. 1) GO TO 930
      IF (IR.LT.2) GO TO 710
      CALL READFE (NS,NO,FBGE)
      GO TO 810
710   CONTINUE
C-----------THE MEASUREMENT MATRIX (HCT*RIN*HC===>SC----------------------------------
      DO 720 I=1,NO
      DO 720 J=1,MH
720   PRC(I,J)=HC(I,J)/RC(I,I)
      DO 730 I=1,MH
      DO 730 J=1,MH
      RM(I+MH,J)=0.D0
      DO 730 K=1,NO
730   RM(I+MH,J)=RM(I+MH,J)-HC(K,I)*PRC(K,J)
C----------------------------GM*Q*GMT===>CC-------------------------------------------
      DO 740 I=1,NS
      DO 740 J=1,NS
      RM(I,J)=BA(I,J)
      RM(I+MH,J+MH)=-BA(J,I)
740   RM(I,J+NS)=CQ(I,J)
      GO TO 400
C-GO BACK TO 450 TO SET UP THE FILTER HAMILTONIAN; CALC.THE FILTER GAINS
750   CALL RGAIN (M,NS,NC,NCB,WR,WI,X,GN,GM ,RM,W21,D1,CR,CI,FRO,MHS,D2)
C-------------------------CHECK EIGENVECTORS------------------------------------------
      IF (IDEBUG .EQ. 0) GO TO 760
      WRITE (6,1570)
      CALL RAFFNT (NS,NS,NS,9,PRC,4,'(9(1X,1PD13.6))')
760   CONTINUE
```

```
      IF (IDSTAB .EQ. 1) GO TO 770
C------------NORMALIZE AND PRINT OPT. ESTIMATOR EIGENSYSTEM-------------
      IWRITE=4
      CALL CNORM (CR,CI,PRO,NS,IWRITE,NSC,DDC,D1,D2,WNORM,WNORMI,HO,AA,
     1NO,NS)
770   DO 780 I=1,MH
      DO 780 J=1,NO
780   PRO(I,J)=+HC(J,I)/RC(J,J)
      DO 790 I=1,MH
      DO 790 J=1,NO
      FBGE(I,J)=0.D0
      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) (X(I,I),I=1,MH)
810   WRITE (6,1630)
      DO 820 I=1,MH
820   WRITE (6,1640) (FBGE(I,J),J=1,NO)
C--------COMPLTE 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
C------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)=EA(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)*HC(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,PRO,LOW,IHIGH,D1)
      CALL ORTHES (NS,NS,LOW,IHIGH,PRO,D2)
      CALL ORTRAN (NS,NS,LOW,IHIGH,PRO,D2,GM)
      CALL HQR2 (NS,NS,LOW,IHIGH,PRO,CR,CI,GM,IERR)
      IF (IERR .NE. 0) CALL EREXIT (NS,PRO,IERR)
```

```
      CALL BALEAK (NS,NS,LCW,IHIGH,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,HC,AA,
     1NO,NS)
      DO 880 I=1,NS
      IF (CR(I).LT.0.0) GO TO 880
      WRITE (5,1660)
      RETURN
880   CONTINUE
      GO TO 900
890   IF (IQ.EQ.0) GO TO 1260
900   DO 910 I=1,NO
      DO 910 J=1,MH
      PRO(I,J)=0.D0
      DO 910 K=1,NO
910   PRO(I,J)=PRO(I,J)+RC(I,K)*FBGE(J,K)
      DO 920 I=1,MH
      DO 920 J=1,MH
      CQ(I,J)=0.D0
      DO 920 K=1,NO
920   CQ(I,J)=CQ(I,J)-FBGE(I,K)*PRO(K,J)
930   CONTINUE
C-------------------THE RMS STATE AND CONTROL RESPONSES-----------------
      IR=IR+1
      GO TO (1050,1090,940,940), IR
940   DO 950 I=1,NS
      DO 950 J=1,NG
      X(I,J)=0.0
      DO 950 K=1,NG
950   X(I,J)=X(I,J)+GAM(I,K)*Q(K,J)
      DO 970 I=1,NS
      DO 970 J=I,NS
      SUM=0.0
      DO 960 K=1,NG
960   SUM=SUM-X(I,K)*GAM(J,K)
      PRC(I,J)=SUM+CQ(I,J)
      PRC(J,I)=PRC(I,J)
      CQ(I,J)=SUM
      CQ(J,I)=SUM
      W21(I,J)=GM(I,J)
970   W21(J,I)=GM(J,I)
      CALL MINV (NSQ,W21,NS,DDD,D1,D2)
      CALL SCCV (NS,GM,W21,CR,CI,NS,GM,W21,CR,CI,PRO,GN)
      WRITE (6,1670)
      CALL RAPRNT (MH,MH,MH,5,GN,4,'(5(1X,1PD13.6))')
      WRITE (6,1680)
```

```
      DO 980 I=1,MH
980   X(I,I)=DSQRT(GN(I,I))
      WRITE (6,1690) (X(I,I),I=1,MH)
      IF (IG.EQ.0) GO TO 1260
      DO 1000 I=1,NC
      DO 1000 J=1,NS
      SUM=0.0
      DO 990 K=1,NS
990   SUM=SUM+FBGC(I,K)*GN(K,J)
1000  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
1010  SUM=SUM+G(I,K)*X(K,J)
1020  PRO(I,J)=CQ(I,J)+SUM
      CALL SCCV (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
1030  W21(I,J)=W21(I,J)+FBGC(I,K)*BA(J,K)
1040  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
1050  SUM=SUM+G(I,K)*W21(K,J)
1060  PRO(I,J)=SUM
      DO 1070 I=1,NS
      DO 1070 J=I,NS
      PRO(I,J)=PRO(I,J)+CQ(I,J)+PRO(J,I)
1070  PRO(J,I)=PRO(I,J)
      CALL SCCV (NS,SC,W11,CWR,CWI,NS,SC,W11,CWR,CWI,PRO,CQ)
      DO 1080 I=1,NS
      DO 1080 J=I,NS
      GM(I,J)=CQ(I,J)-BA(I,J)-BA(J,I)+GN(I,J)
1080  GM(J,I)=GM(I,J)
      GO TO 1100
1090  CALL SCCV (NS,SC,W11,CWR,CWI,NS,SC,W11,CWR,CWI,CQ,GM)
1100  IF (NC.EQ.0) GO TO 1150
      DO 1120 I=1,NS
      DO 1120 J=1,NC
      PRO(I,J)=0.D0
      DO 1110 K=1,NS
1110  PRO(I,J)=PRO(I,J)+GM(I,K)*FBGC(J,K)
```

```
1120  CONTINLE
      DO 114C I=1,NC
      DO 1140 J=1,NC
      SC(I,J)=0.C0
      DO 113C K=1,NS
1130  SC(I,J)=SC(I,J)+FBGC(I,K)*PRO(K,J)
1140  CONTINLE
1150  IF (IREG .EG. 0) GC TC 1170
      DO 1160 I=1,NS
      DO 116C J=1,NS
1160  CQ(I,J)=GM(I,J)
      GO TO 1190
1170  WRITE (6,17C0)
      CALL RAFRNT (MH,MH,MH,5,GM,4,'(5(1X,1PD13.6))')
      IF (IR.GT.2) GO TO 1190
      DO 118C I=1,MH
      DO 1180 J=1,MH
1180  CQ(I,J)=GN(I,J)+GM(I,J)
1190  CONTINLE
      WRITE (6,1710)
      CALL RAFRNT (MH,MH,MH,5,CQ,4,'(5(1X,1PD13.6))')
      IF (NC.EC.0) GO TO 1210
      WRITE (6,1720)
      DO 120C I=1,NC
1200  WRITE (6,1730) (SC(I,J),J=1,NC)
1210  DO 122C I=1,NS
1220  CQ(I,I)=DSQRT(CQ(I,I))
      IF (NC.EC.0) GC TO 1240
      DO 123C I=1,NC
1230  SC(I,I)=DSQRT(SC(I,I))
1240  WRITE (6,1740)
      DO 125C 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  CONTINLE
1260  IF (ITF3 .EC. 0) GC TC 1290
C-------FORM CCMPENSATCR FRCM MEAS TC INPLT AND CCMPUTE TF-------------
      CO 128C I=1,NS
      CO 128C J=1,NS
      SUM=0.DC
      CO 1270 K=1,NO
1270  SUM=SUM+FBGE(I,K)*HC(K,J)
1280  CQ(I,J)=ACL(I,J)-SUM
      WRITE (6,1760)
      ITFX=3
      IZERO=C
      CALL TF (NS,NS,NSQ,CQ,AA,NC,FBGE,BM,NC,FBGC,CM,IZERO,D,EB,CC,CP,
     1WR,WI,CWR,CWI,SC,JCF,RES,D1,D2,DDD,EPS,ITF3,ITFX)
```

```
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,FBGC,NC,HY,HU,HC,FBGE,NG,
      1 GAM,ACL,BA,WR,WI,D1,D2,JCF,RES,Q,RC,BB,CC,1   ,IPSD,INORM)
       CALL PSDCAL (M,NS,RM,X,NC,GW,GV,FBGC,NC,HY,HY,HO,FBGE,NG,
      1 GAM,ACL,BA,WR,WI,D1,D2,JCF,RES,Q,RC,BB,CC,2   ,IPSD,INORM)
       GO TO 1310
1300   CALL PSDCAL (M,NS,RM,X,NC,GW,GV,FBGC,NC,HY,HU,HO,FBGE,NG,
      1 GAM,ACL,BA,WR,WI,D1,D2,JCF,RES,Q,RC,BB,CC,IYU,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
1320   ACL(I,J)=BA(I,J)
1330   CONTINUE
       CALL MINV (NSQ,ACL,NS,DDD,D1,D2)
       CALL READW (NG,WR)
       WRITE (6,1770) (WR(I),I=1,NG)
       WRITE (6,1780)
       DO 1340 I=1,NS
       WI(I)=0.0
       DO 1340 J=1,NG
1340   WI(I)=WI(I)+GAM(I,J)*WR(J)
       DO 1360 I=1,NS
       CR(I)=0.0
       DO 1350 J=1,NS
1350   CR(I)=CR(I)-ACL(I,J)*WI(J)
1360   WRITE (6,1390) CR(I)
       DO 1370 I=1,NC
       CI(I)=0.0
       DO 1370 J=1,NS
1370   CI(I)=CI(I)+FBGC(I,J)*CR(J)
       WRITE (6,1790) (CI(I),I=1,NC)
       RETURN
C-----------------------------------------------------------------------
C670   FORMAT (2X,1P6D14.6,/,2X,6D14.6)
1380   FORMAT (//,5X,45HOPEN LOOP DYNAMICS MATRIX.................F..,//)
1390   FORMAT (10(2X,0PD11.4))
1400   FORMAT (//,5X,45HTHE CONTROL DISTRIBUTION MATRIX...........G..,//)
1410   FORMAT (//,5X,45HTHE CONTROL COST MATRIX...................B..,//)
1420   FORMAT (//,5X,45HPROCESS NOISE DISTRIBUTION MATRIX.....GAMMA..,//)
1430   FORMAT (//,5X,45HPOWER SPECTRAL DENSITY - PROCESS NOISE....Q..,//)
1440   FORMAT (//,5X,45HMEASUREMENT SCALING MATRIX................H..,//)
1450   FORMAT (//,5X,45HPOWER SPECTRAL DENSITY-MEASUREMENT NOISE..R..,//)
1460   FORMAT (//,5X,45HOUTPUT COST MATRIX........................A..,//)
1470   FORMAT (//,5X,45HMEASUREMENT FEEDTHROUGH MATRIX............D..,//)
```

```
1480  FORMAT (//,25X,28H....DESTABILIZATION CASE....,//,10X,39HTHE FOLLO
     1WING VALUES WILL BE ADDED DOWN,/,10X,49HTHE DIAGONAL OF THE "F" MA
     2TRIX TO DESTABILIZE IT.,/,10X,41HOPTIMAL GAINS FOR THE DESTABILIZE
     3D SYSTEM,/,10X,39HARE THEN USED AS FIXED SUBOPTIMAL GAINS,/,10X,28
     4HFOR THE SYSTEM CALCULATIONS.,//)
1490  FORMAT (///43H PROGRAM TERMINATING DUE TO UNSTABLE SYSTEM)
1500  FORMAT (//,5X,31HOPEN LOOP TRANSFER FUNCTIONS...)
1510  FORMAT (//,5X,32H EULER-LAGRANGE SYSTEM MATRIX...,//)
1520  FORMAT (//,5X,43HEIGENVALUES AND EIGENVECTORS OF THE 2N X 2N,/,5X,
     145HEULER-LAGRANGE SYSTEM AFTER HQR2.............,//)
1530  FORMAT (1X,1P2D13.6)
1540  FORMAT (1X)
1550  FORMAT (//,5X,41HEIGENSYSTEM OF OPTIMAL REGULATOR..........,//)
1560  FORMAT (//,5X,41HEIGENSYSTEM OF OPTIMAL ESTIMATOR..........,//)
1570  FORMAT (//,5X,39H EIGENVECTORS FROM RGAIN PRIOR TO CNORM,//)
1580  FORMAT (//,2X,57HTHE OPTIMAL FEEDBACK GAIN CONTROL MATRIX...C=BINV
     1*GT*S...,//)
1590  FORMAT (10(2X,1PD11.4))
1600  FORMAT (//,5X,45HTHE CLOSED LOOP DYNAMICS MATRIX ......F-G*C...,//)
1610  FORMAT (///,60H   PROGRAM TERMINATING DUE TO UNSTABLE CLOSED LOOP
     1   SYSTEM)
1620  FORMAT (//,2X,'NOISE TRANSFER FUNCTIONS THROUGH THE CLOSED-LOOP SY
     1STEM..',//)
1630  FORMAT (//,5X,45HFILTER STEADY STATE GAINS.................K...,//)
1640  FORMAT (1X,2X,1P6D14.6)
1650  FORMAT (//,5X,45HTHE CLOSED LOOP FILTER DYNAMICS MATRIX IS.....,//)
1660  FORMAT (////,43H PROGRAM TERMINATING DUE TO UNSTABLE FILTER)
1670  FORMAT (//,5X,45HTHE COVARIANCE OF THE ESTIMATION ERROR....P...,//)
1680  FORMAT (//,5X,45HRMS VALUES OF THE ESTIMATION ERROR............,//)
1690  FORMAT ((5(1X,1PD13.6)))
1700  FORMAT (//,5X,45HTHE COVARIANCE OF THE ESTIMATE.........XHAT...,//)
1710  FORMAT (//,5X,45HTHE STATE COVARIANCE MATRIX.....X=XHAT + P...,//)
1720  FORMAT (//,5X,45HTHE CONTROL COVARIANCE.........U=C*XHAT*CT...,//)
1730  FORMAT (1P6D14.6)
1740  FORMAT (//,2X,18HSTATE RMS RESPONSE,20X,20HCONTROL RMS RESPONSE,/)
1750  FORMAT (1X,1PD15.7,25X,D15.7)
1760  FORMAT (//,5X,50HCOMPENSATOR TRANSFER FUNCTIONS FROM MEAS. TO INPU
     1T,/,5X,52H.....................U/Z = -C*(SI-F+G*C+K*H)INV*K...,//)
1770  FORMAT (//,5X,46HSTEADY DISTURBANCE VECTOR...................W..,//
     1,10(1X,1PD12.4/))
1780  FORMAT (//,5X,45HSTEADY STATE VALUES OF STATE VAR. ARE.........,//)
1790  FORMAT (//,5X,47HSTEADY STATE CONTROL IS ........................,/
     1/10(1X,1PD12.4/))
1800  FORMAT (//,5X,49HENTER THE MAGNITUDE OF THE DESTABILIZATION VECTOR
     1,/,8X,47HTO BE ADDED DOWN THE DIAGONAL OF THE "F"-MATRIX,/,8X,18HT
     2O DESTABILIZE IT.,//)
      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
10    WRITE (6,FMT) (A(I,J),J=NL,NU)
      WRITE (6,30)
20    NU=NU+L
      RETURN
30    FORMAT (1X)
      END
C==========================================================================
      SUBROUTINE RGAIN (M,NS,NC,NOB,WR,WI,VF,GN,W11,TCB,W21,LT,C,CI,CT,M
     1HS,MT)
      IMPLICIT REAL*8 (A-H,O-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--------------------------------------------------------------------------
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
20    NRZEV=NRZEV+1
      IF (NRZEV.GT.1) GO TO 30
      WR(K)=EIGVR
      GO TO 80
30    WR(K)=-EIGVR
      WRITE (6,290)
      GO TO 150
40    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 1EC
60    IF (WR(K)) 140,70,70
70    IF (WI(K)) 110,80,110
C-----------------EIGENVECTOR FOR REAL EIGENVALUE,FOSITIVE---------------
80    IF (NOE.EC.0) GC TC 100
      DO 90 J=1,M
90    TCE(J,KF)=VF(J,K)
100   KP=KP+1
      K=K+1
      GO TO 10
C-----------EIGENVECTCR FCR CCMPLEX EIGENVALUE,POSITIVE REAL FART-------
110   IF (NOB.EQ.0) GO TC 130
      CO 120 J=1,M
      FR=VF(J,K)
      FI=-VF(J,K+1)
      TCB(J,KF)=FR+FI
120   TCB(J,KF+1)=FR-FI
130   KP=KP+2
      K=K+2
      GO TO 1C
140   IF (WI(K)) 180,150,180
C------------EIGENVECTCR FCR REAL EIGENVALUE,NEGATIVE REAL PART---------
150   C(KN)=WR(K)
      CI(KN)=WI(K)
      IF (NOB.NE.0) GO TC 170
      KNS=KN+NS
      DO 160 J=1,M
160   TCB(J,KNS)=VF(J,K)
170   KN=KN+1
      K=K+1
      GO TO 1C
C------------EIGENVECTCR FCR COMPLEX EIGENVALLE,NEGATIVE REAL PART------
180   RR=WR(K)
      RI=WI(K)
      C(KN)=RR
      C(KN+1)=RR
      CI(KN)=RI
      CI(KN+1)=-RI
      IF (NOB.NE.C) GO TC 200
      KNS=KN+NS
      CO 190 J=1,M
      FR=VF(J,K)
      FI=-VF(J,K+1)
      TCB(J,KNS)=FR+FI
190   TCB(J,KNS+1)=FR-FI
200   KN=KN+2
      K=K+2
      GO TO 1C
```

```
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)=TCE(I,J+NS)
220    CT(I,J)=W11(I,J)
C-------------------------FORMATION OF W21----------------------------
       DO 230 I=1,NS
       DO 230 J=1,NS
230    W21(I,J)=TCE(I+NS,J+NS)
240    IF (NOB.EQ.0) GO TO 260
       DO 250 I=1,NS
       DO 250 J=1,NS
       W21(I,J)=-TCB(I,J)
250    W11(I,J)=TCE(I+NS,J)
260    CONTINUE
C---------------------------INVERT W11--------------------------------
       NSQ=NS*NS
       CALL MINV (NSQ,W11,NS,DETC,LT,MT)
C--------------------CALCULATE THE RGAIN MATRIX-----------------------
       DO 270 IL=1,NS
       DO 270 JL=1,NS
       GN(IL,JL)=0.D0
       DO 270 KL=1,NS
270    GN(IL,JL)=GN(IL,JL)+W21(IL,KL)*W11(KL,JL)
       IF (NOE.EQ.0) RETURN
       DO 280 I=1,NS
       DO 280 J=1,NS
280    CT(I,J)=W11(J,I)
       RETURN
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,O-Z)
       DIMENSION A(NSQ),L(N),M(N)
       DOUBLE PRECISION A,D,BIGA,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
      M(K)=J
20    CONTINUE
C-----------------------------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)
40    A(JI)=HOLD
C-----------------------------INTERCHANGE COLUMNS-------------------------
50    I=M(K)
      IF (I-K) 80,80,60
60    JP=N*(I-1)
      DO 70 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
70    A(JI)=HOLD
C-------------------------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
C-----------------------------REDUCE MATRIX-------------------------------
      DO 150 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 150 J=1,N
```

```
      IJ=IJ+N
      IF (I-K) 130,150,130
130   IF (J-K) 140,150,140
140   KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
150   CONTINUE
C-----------------------------DIVIDE ROW BY PIVOT-------------------------
      KJ=K-N
      DO 170 J=1,N
      KJ=KJ+N
      IF (J-K) 160,170,160
160   A(KJ)=A(KJ)/BIGA
170   CONTINUE
C------------------------------PRODUCT OF PIVOTS--------------------------
      D=D*BIGA
C--------------------------REPLACE PIVOT BY RECIPROCAL--------------------
      A(KK)=(1.0D0)/BIGA
180   CONTINUE
C----------------------FINAL ROW AND COLUMN INTERCHANGE-------------------
      K=N
190   K=(K-1)
      IF (K) 260,260,200
200   I=L(K)
      IF (I-K) 230,230,210
210   JQ=N*(K-1)
      JR=N*(I-1)
      DO 220 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
220   A(JI)=HOLD
230   J=M(K)
      IF (J-K) 190,190,240
240   KI=K-N
      DO 250 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
250   A(JI)=HOLD
      GO TO 190
260   K=0
      RETURN
      END
C=========================================================================
      SUBROUTINE SCOV (NL,WL,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
 10   DO 20 I=1,NL
      DO 20 J=1,NR
      X(I,J)=0.
      DO 20 II=1,NL
 20   X(I,J)=X(I,J)+WLI(I,II)*Q(II,J)
      DO 40 I=1,NL
      DO 40 J=1,NR
      Q(I,J)=0.
      DO 30 JJ=1,NR
 30   Q(I,J)=Q(I,J)+X(I,JJ)*WRI(J,JJ)
 40   CONTINUE
      I=1
 50   IF (VL2(I)) 60,110,60
 60   J=1
 70   IF (VR2(J)) 80,90,80
 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
 90   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
100   IF (J.LE.NR) GO TO 70
      I=I+2
      GO TO 160
110   J=1
120   IF (VR2(J)) 130,140,130
130   A=VR1(J)+VL1(I)
      B=A**2+VR2(J)**2
      K1=A/B
```

```
      K2=VR2(J)/B
      X(I,J)=K1*Q(I,J)-K2*Q(I,J+1)
      X(I,J+1)=K2*Q(I,J)+K1*Q(I,J+1)
      J=J+2
      GO TO 150
140   X(I,J)=Q(I,J)/(VR1(J)+VL1(I))
      J=J+1
150   IF (J.LE.NR) GO TO 120
      I=I+1
160   IF (I.LE.NL) GO TO 50
      DO 170 I=1,NL
      DO 170 J=1,NR
      Q(I,J)=0.
      DO 170 II=1,NL
170   Q(I,J)=Q(I,J)+WL(I,II)*X(II,J)
      DO 190 I=1,NL
      DO 190 J=1,NR
      X(I,J)=0.
      DO 180 JJ=1,NR
180   X(I,J)=X(I,J)+Q(I,JJ)*WR(J,JJ)
190   CONTINUE
      RETURN
      END
C=======================================================================
      SUBROUTINE MODE (WNORM,G,GNORM,NS,N1,N2,ICON)
C                                                                      =
C  WNORM  TRANSFORMATION MATRIX U OR L-INV                             =
C  NS     NO. OF STATE                                                 =
C  NC     NO.  OF INPUTS OR OUTPUTS                                    =
C  ICON  CONTROL FLAG TO INDICATE WHICH TRANSFORMATION                 =
C          0 =  MODAL G                                                =
C          1 =  MODAL GAMMA                                            =
C          2 =  MODAL H                                                =
C          3 =  MODAL C                                                =
C          4 =  MODAL K                                                =
C          5 =  CONTROL EIGENVECTOR MATRIX                             =
C          6 =  MEASUREMENT EIGENVECTOR MATRIX                         =
C=======================================================================
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION WNORM(NS,NS),G(N1,N2),GNORM(N1,N2)
      DO 10 I=1,N1
      DO 10 J=1,N2
10    GNORM(I,J)=0.
      IPOINT=ICON+1
      GO TO (20,20,90,90,20,90,90), IPOINT
20    DO 30 J=1,N2
      DO 30 I=1,NS
      DO 30 K=1,NS
```

```
30      GNORM(I,J)=GNORM(I,J)+WNORM(I,K)*G(K,J)
        GO TO (40,70,90,90,80), IPCINT
40      WRITE (6,170)
50      DO 60 I=1,NS
60      WRITE (6,230) (GNORM(I,J),J=1,N2)
        RETURN
70      WRITE (6,180)
        GO TO 50
80      WRITE (6,240)
        GO TO 50
90      DO 100 J=1,NS
        DO 100 I=1,N1
        DO 100 K=1,NS
100     GNORM(I,J)=GNORM(I,J)+C(I,K)*WNORM(K,J)
        GO TO (110,110,110,120,110,130,140), IPCINT
110     WRITE (6,190)
        GO TO 150
120     WRITE (6,200)
        GO TO 150
130     WRITE (6,210)
        GO TO 150
140     WRITE (6,220)
150     DO 160 I=1,N1
160     WRITE (6,230) (GNORM(I,J),J=1,NS)
        RETURN
C-----------------------------------------------------------------------
170     FORMAT (//,5X,45HMODAL CONTROL DISTRIBUTION MATRIX......TI*G..,//)
180     FORMAT (//,5X,50HMODAL PROCESS NOISE DISTRIBUTION MATRIX...TI*GAM.
       1.,//)
190     FORMAT (//,5X,45HMODAL MEASUREMENT SCALING MATRIX...H(BAR)*T..,//)
200     FORMAT (//,5X,45HTHE MODAL CONTROL GAINS.................C*T..,//)
210     FORMAT (//,5X,45HCONTROL EIGENVECTOR MATRIX..............C*M..,//)
220     FORMAT (//,5X,45HMEASUREMENT EIGENVECTOR MATRIX.....H(BAR)*M..,//)
230     FORMAT (1X,(2X,1P6E14.6))
240     FORMAT (//,5X,45HMODAL FILTER STEADY STATE GAINS........TI*K..,//)
        END
C=======================================================================
        SUBROUTINE CNORM (WZ,WY,VEC,NS,IWRITE,NSQ,DED,D1,D2,WNCRM,WNORMI,H
       1G,CM,N1,N2)
C                                                                      =
C           WZ(I)       REAL PART OF I-TH EIGENVALUE                   =
C                                                                      =
C           WY(I)       COMPLEX PART OF I-TH EIGENVALUE                =
C                                                                      =
C           VEC         MATRIX OF RIGHT EIGENVECTORS STORED IN REAL FORM =
C                        FROM HQR2                                     =
C           NS          NO. OF STATES                                  =
C                                                                      =
```

```
C          IWRITE     FLAG TO CONTROL FORMATS FOR DIFFERENT EIGHENSYSTEMS=
C                                                                        =
C          WNORM      NORMALIZED MATRIX U OF RIGHT EIGENVECTORS STORED   =
C                      BY COLUMNS IN REAL FORM                           =
C          WNORMI     U-INVERSE 2*CONGUGATE OF LEFT EIGENVECTORS         =
C                       STORED BY ROW IN REAL FORM                       =
C          NSC,CDD,D1,D2 - ARGUMENTS PASSED TO MINV                      =
C=========================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 FIELD,COMMA,SEMCOL,RIGHT,FMT
      DIMENSION W2(NS),WY(NS),VEC(NS,NS),WNORM(NS,NS),WNORMI(NS,NS),STOR
     1E(6),D1(NS),D2(NS),FMT(14),HO(N1,N2),CM(N1,N2)
      DATA FIELD/5HE12.5/,COMMA/5H,',',/,SEMCOL/5H,':',/,RIGHT/1H)/,FMT/
     16H(1X,1F,13*1H /,SEMEND/4H,':'/
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
      CMOD=VEC(I,K)**2+VEC(I,K+1)**2
      IF (CMOD-EMAX) 20,10,10
10    EMAX=CMOD
      M=I
20    CONTINUE
      VMR=VEC(M,K)
      VMI=VEC(M,K+1)
      DO 30 I=1,NS
      VR=VEC(I,K)
      VI=VEC(I,K+1)
      VECRN=(VR*VMR+VI*VMI)/EMAX
      VECIN=(-VR*VMI+VI*VMR)/EMAX
      WNORM(I,K)=VECRN
      WNORM(I,K+1)=VECIN
30    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
      RENOD=0.D0
```

```
      DO 60 I=1,NS
60    REMOD=VEC(I,K)**2+REMOD
      RMOD=DSQRT(REMOD)
      DO 70 I=1,NS
      RVEC=VEC(I,K)/RMOD
      WNORM(I,K)=RVEC
70    CONTINUE
80    CONTINUE
      GO TO (90,100,110,120,130), IWRITE
90    WRITE (6,320)
      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
      NFMTW=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(NFMTW+1)=RIGHT
      WRITE (6,FMT) (STORE(J),J=1,NPRTW)
      NPRTW=0
      NFMTW=1
150   NPRTW=NPRTW+1
      NFMTW=NFMTW+1
      IF (KK.EQ.1) GO TO 160
      STORE(NPRTW)=WZ(I)
      FMT(NFMTW)=FIELD
      NFMTW=NFMTW+1
      FMT(NFMTW)=SEMCOL
      GO TO 180
160   STORE(NPRTW)=WZ(I)
      FMT(NFMTW)=FIELD
      FMT(NFMTW+1)=COMMA
      STORE(NPRTW+1)=WY(I)
      FMT(NFMTW+2)=FIELD
      FMT(NFMTW+3)=SEMCOL
      NFMTW=NFMTW+3
      NPRTW=NPRTW+1
      GO TO 180
170   KK=0
```

```
180    CONTINUE
       FMT(NFMTW)=SEMEND
       FMT(NFMTW+1)=RIGHT
       WRITE (6,FMT) (STORE(J),J=1,NPRTW)
       IF (IWRITE.NE.1) GO TO 190
       WRITE (6,370)
       GO TO 200
190    WRITE (6,380)
200    CALL RAPRNT (NS,NS,NS,6,WNORM,4,'(6(1X,1PD13.6))')
       GO TO (230,210,210,220,220), IWRITE
210    CALL MODE (WNORM,HC,CM,NS,N1,N2,5)
       GO TO 230
220    CALL MODE (WNORM,HC,CM,NS,N1,N2,6)
230    GO TO (240,250,260,270,280), IWRITE
240    WRITE (6,390)
       GO TO 290
250    WRITE (6,400)
       GO TO 290
260    WRITE (6,410)
       GO TO 290
270    WRITE (6,420)
       GO TO 290
280    WRITE (6,430)
C-----------------SAVE L-INVERSE OPEN LOOP IN WNORMI-----------------
290    IF (IWRITE .GT. 1) GO TO 310
       DO 300 I=1,NS
       DO 300 J=1,NS
300    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)..,//)
330    FORMAT (//5X,46HC-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G*C)..,//)
340    FORMAT (//5X,46HC-LOOP SUBOPT. REG. E-VALUES...DET(SI-F+G*C)..,//)
350    FORMAT (//5X,46HC-LOOP OPTIMAL EST. E-VALUES...DET(SI-F+K*H)..,//)
360    FORMAT (//5X,46HC-LOOP SUBOPT. EST. E-VALUES...DET(SI-F+K*H)..,//)
370    FORMAT (//5X,46HOPEN LOOP RIGHT EIGENVECTOR MATRIX........T....,//)
380    FORMAT (//5X,46HC-LOOP RIGHT EIGENVECTOR MATRIX..........M....,//)
390    FORMAT (//5X,46HOPEN LOOP LEFT EIGENVECTOR MATRIX......T-INV..,//)
400    FORMAT (//5X,46HC-LOOP OPT. REG. LEFT E-VECTOR MATRIX..M-INV..,//)
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.,//)
```

```
      END
C=====================================================================
      SUBROUTINE TF (N,NM,NSC,A,AA,M,B,BM,L,C,CM,IFDFW,D,BB,CC,CP,
     1      EVR,EVI,PR,PI,SC,JCF,RES,D1,D2,DDD,EPS,ITF,ITFX)
      IMPLICIT REAL*8(A-H,O-Z)
      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 CPTSYS-----
      IF (ITFX .EQ. 1) GO TO 50
      IF (ITFX .EQ. 2) GO TO 10
      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,L
      DO 30 J=1,N
      CM(I,J)=0.D0
      DO 30 K=1,N
30    CM(I,J)=CM(I,J)+C(I,K)*AA(K,J)
      CALL MINV (NSQ,AA,N,DDD,D1,D2)
      DO 40 I=1,N
      DO 40 J=1,M
      BM(I,J)=0.D0
      DO 40 K=1,N
40    BM(I,J)=BM(I,J)+AA(I,K)*B(K,J)
50    CONTINUE
      DO 60 I=1,M
      DO 60 J=1,L
      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)
      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)
      IMPLICIT REAL*8(A-H,O-Z)
      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
10    AA(I,J)=A(I,J)
      CALL BALANC (NM,N,AA,LOW,IHIGH,D1)
      CALL ORTHES (NM,N,LOW,IHIGH,AA,D2)
      CALL ORTRAN (NM,N,LOW,IHIGH,AA,D2,SC)
      CALL HQR2 (NM,N,LOW,IHIGH,AA,EVR,EVI,SC,IERR)
```

```
      IF (IERR .NE. 0) GC TC 30
      CALL BALBAK (NM,N,LCW,IHIGH,D1,N,SC)
      WRITE (6,40)
      CO 20 I=1,N
20    WRITE (6,50) EVR(I),EVI(I)
      RETURN
30    WRITE (5,60)
      RETURN
C-----------------------------------------------------------------------
40    FORMAT (///,28H TF DENCMINATOR EIGENVALLES:,/)
50    FORMAT (/,2X,3H  (,F13.6,4H)+J(,F13.6,1H))
60    FORMAT (35H FAILURE IN HQR2, CALCULATING POLES)
      END
C=======================================================================
      SUBROUTINE ZEROS (K1,K2,IFDFW,N,NM,A,AA,M,B,L,C,D,BB,CC,CP,EVR,EVI
     1,     D1,D2,EPS)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSICN 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 SCL,DABS
      DO 10 I=1,N
      BB(I)=B(I,K1)
      CC(I)=C(K2,I)
      DO 10 J=1,N
10    AA(I,J)=A(I,J)
      WRITE (6,9C) K1,K2
      IF (IFDFW .EQ. 0) GC TC 20
      H=D(K2,K1)
      IF (DABS(H).LE.EPS) GO TO 20
      JJ=N
      GO TO 5C
20    NN=N-1
      DO 30 I=1,NN
      H= SCL(N,BB,CC)
      CALL CCCMP (N,NM,AA,CC,CP)
      IF (DABS(H).GT.EPS) GC TO 40
30    CONTINUE
      H=SCL(N,BB,CC)
      WRITE (6,10C) H
      GO TO 7C
40    JJ=N-I
50    WRITE (6,110) JJ,H
      CALL ACCMP (N,NM,AA,BB,CC,H)
      CALL BALANC (NM,N,AA,LCW,IHIGH,D1)
      CALL ORTHES (NM,N,LCW,IHIGH,AA,D2)
      CALL HQR (NM,N,LOW,IHIGH,AA,EVR,EVI,IERR)
      IF (IERR .NE. 0) GC TC 80
      WRITE (6,12C)
```

```
      DO 60 I=1,N
60    WRITE (6,130) EVR(I),EVI(I)
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)
      END
C==========================================================================
      SUBROUTINE ACOMP (N,NM,A,B,C,H)
      REAL*8 A,B,C,H
      DIMENSION A(NM,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,NM,A,C,CC)
      REAL*8 A,C,CC
      DIMENSION A(NM,N),C(N),CC(N)
      DO 10 I=1,N
      CC(I)=0.
      DO 10 J=1,N
10    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
10    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/
C-------------TEMPORARY MOD TILL JCF IS CALCULATED--------------------
      DO 10 I=1,N
10    JCF(I)=0
C-----------------------TEMPORARY MOD----------------------------------
      IF (IPT .EQ. 1) WRITE (6,170)
      DO 20 I=1,N
      BB(I)=BM(I,K1)
20    CC(I)=CM(K2,I)
C---------------------LOOP THROUGH THE POLES--------------------------
      I=0
30    I=I+1
      IF (I .GT. N) GO TO 160
      IF (JCF(I) .EQ. 1) GO TO 60
      IF (DABS(PI(I)) .LT. 1.D-10) GO TO 50
C--------COMPUTE SIMPLE COMPLEX 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
```

```
70    K=K+1
80    CONTINUE
      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)+CC(I+2)*BB(I+3)-CC(I+3)*BB(I+
     12)
      RES(I+2)=CC(I)*BB(I+3)+CC(I+1)*BB(I+2)
      RES(I+3)=CC(I)*BB(I+3)-CC(I+1)*BB(I+2)
      IF (IPT .EQ. 0) GO TO 100
      PRT(1)=R1
      PRT(2)=R2
      IF (DABS(PR(I)) .GT. 1.D-10) GO TO 90
      PRT(1)=BLANK
      PRT(2)=BLANK
90    PRT(3)=CS
      PRT(4)=EC
      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
100   I=I+3
      GO TO 30
C----COMPUTE REPEATED REAL POLE 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
120   RES(J)=RES(J)+BB(JJ)*CC(JJ-NN+1)
130   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 (6,190)  PR(J),PI(J),RES(J),PRT(1),NN,(PRT(JJ),JJ=2,4)
140   NN=NN+1
      GO TO 30
150   I=KT
      GO TO 30
160   CONTINUE
      RETURN
C----------------------------------------------------------------------
170   FORMAT (//,3X,22HRESIDUES AT THE POLES:/,T16,9HP O L E S,T41,15HR
     1E S I D U E S,/,T9,7HREAL(A),T26,7HIMAG(B))
180   FORMAT (/,4X,1H(,F13.6,4H)+J(,F13.6,1H),4X,1H(,F13.6,1H),3A8,A1)
190   FORMAT (/,4X,1H(,F13.6,4H)+J(,F13.6,1H),4X,1H(,F13.6,1H),A4,I2,2X,
     12A8,A1)
      END
C======================================================================
      SUBROUTINE BALANC (NM,N,A,LOW,IGH,SCALE)
      INTEGER I,J,K,L,M,N,JJ,NM,IGH,LOW,IEXC
      REAL*8 A(NM,N),SCALE(N)
      REAL*8 C,F,G,R,S,B2,RADIX
      REAL*8 DABS
      LOGICAL NOCONV
      DATA RADIX/Z4210000000000000/
C----------------------------------------------------------------------
      B2=RADIX*RADIX
      K=1
      L=N
      GO TO 60
C-------------IN-LINE PROCEDURE FOR ROW AND COLUMN EXCHANGE------------
10    SCALE(M)=J
      IF (J .EQ. M) GO TO 40
      DO 20  I=1,L
      F=A(I,J)
      A(I,J)=A(I,M)
      A(I,M)=F
20    CONTINUE
      DO 30  I=K,N
      F=A(J,I)
      A(J,I)=A(M,I)
      A(M,I)=F
30    CONTINUE
40    GO TO (50,90), IEXC
C-----SEARCH FOR ROWS ISOLATING AN EIGENVALUE AND PUSH THEM DOWN-------
50    IF (L .EQ. 1) GO TO 230
      L=L-1
60    DO 80  JJ=1,L
```

```
      J=L+1-JJ
      DO 70 I=1,L
      IF (I .EQ. J) GO TO 70
      IF (A(J,I) .NE. 0.0D0) GO TO 80
70    CONTINUE
      M=L
      IEXC=1
      GO TO 10
80    CONTINUE
      GO TO 100
C-------SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE AND PUSH THEM LEFT---
90    K=K+1
100   DO 120 J=K,L
      DO 110 I=K,L
      IF (I .EQ. J) GO TO 110
      IF (A(I,J) .NE. 0.0D0) GO TO 120
110   CONTINUE
      M=K
      IEXC=2
      GO TO 10
120   CONTINUE
C----------------NOW BALANCE THE SUBMATRIX IN ROWS K TO L--------------
      DO 130 I=K,L
130   SCALE(I)=1.0D0
C----------------ITERATIVE LOOP FOR NORM REDUCTION-----------------------
140   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(J,I))
      R=R+DABS(A(I,J))
150   CONTINUE
C----------------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
160   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
```

```
C-------------------------------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
200   A(I,J)=A(I,J)*G
      DO 210 J=1,L
210   A(J,I)=A(J,I)*F
220   CONTINUE
      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
10    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)
20    CONTINUE
      G=-DSIGN(DSQRT(H),ORT(M))
      H=H-ORT(M)*G
      ORT(M)=ORT(M)-G
C-------------------------FORM (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)
30    CONTINUE
```

```
      F=F/H
      DO 40 I=M,IGH
40    A(I,J)=A(I,J)-F*ORT(I)
50    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)
60    CONTINUE
      F=F/H
      DO 70 J=M,IGH
70    A(I,J)=A(I,J)-F*ORT(J)
80    CONTINUE
      ORT(M)=SCALE*ORT(M)
      A(M,M-1)=SCALE*G
90    CONTINUE
100   RETURN
      END
C======================================================================
      SUBROUTINE ORTRAN (NM,N,LOW,IGH,A,ORT,Z)
      INTEGER I,J,N,KL,MM,MP,NM,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
10    Z(I,J)=0.0D0
      Z(I,I)=1.0D0
20    CONTINUE
      KL=IGH-LOW-1
      IF (KL .LT. 1) GO TO 80
      DO 70 MM=1,KL
      MP=IGH-MM
      IF (A(MP,MP-1) .EQ. 0.0D0) GO TO 70
      MP1=MP+1
      DO 30 I=MP1,IGH
30    ORT(I)=A(I,MP-1)
      DO 60 J=MP,IGH
      G=0.0D0
      DO 40 I=MP,IGH
40    G=G+ORT(I)*Z(I,J)
C----------------DIVISOR BELOW IS NEGATIVE OF H FORMED IN ORTHES.-------
C----------------DOUBLE DIVISION AVOIDS POSSIBLE UNDERFLOW-------------
      G=(G / ORT(MP))/A(MP,MP-1)
      DO 50 I=MP,IGH
50    Z(I,J)=Z(I,J)+G*ORT(I)
```

```
60    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,NM,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 DSQRT,DABS,DSIGN
      INTEGER MINO
      LOGICAL NOTLAS
      COMPLEX *16Z3
      COMPLEX *16DCMPLX
      REAL*8 DREAL,DIMAG
C----------STATEMENT FUNCTIONS ENABLE EXTRACTION OF REAL AND IMAGINARY--
C----------PARTS OF DOUBLE PRECISION COMPLEX NUMBERS--------------------
      DREAL(Z3)=Z3
      DIMAG(Z3)=(0.0D0,-1.0D0)*Z3
      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
10    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
20    CONTINUE
      EN=IGH
      T=0.0D0
C----------------SEARCH FOR NEXT EIGENVALUES----------------------------
30    IF (EN .LT. LOW) GO TO 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=NORM
      IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 60
50    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 TO 500
        IF (ITS .NE. 10 .AND. ITS .NE. 20) GO TO 80
C----------------------FORM EXCEPTIONAL SHIFT---------------------------
        T=T+X
        DO 70 I=LOW,EN
70      H(I,I)=H(I,I)-X
        S=DABS(H(EN,NA))+DABS(H(NA,ENM2))
        X=0.75D0*S
        Y=X
        W=-0.4375D0*S*S
80      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
90      CONTINUE
100     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
110     CONTINUE
C--------DOUBLE QR 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
120   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
150   H(K+1,J)=H(K+1,J)-P*Y
      H(K,J)=H(K,J)-P*X
160   CONTINUE
      J=MIN0(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
170   H(I,K+1)=H(I,K+1)-P*Q
      H(I,K)=H(I,K)-P
180   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
190   Z(I,K+1)=Z(I,K+1)-P*Q
      Z(I,K)=Z(I,K)-P
200   CONTINUE
210   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=DSQRT(DABS(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
280   EN=ENM2
      GO TO 3C
C----------------ALL ROOTS FOUND.  BACKSUBSTITUTE TO FIND--------------
C----------------VECTORS OF UPPER TRIANGULAR FORM-----------------------
290   IF (NORM .EQ. 0.0D0) GC TO 510
      DO 450 NN=1,N
      EN=N+1-NN
      P=WR(EN)
      Q=WI(EN)
      NA=EN-1
      IF (Q) 370,300,450
C--------------------------REAL VECTOR----------------------------------
300   M=EN
      H(EN,EN)=1.0D0
      IF (NA .EQ. 0) GC TC 450
      DO 360 II=1,NA
      I=EN-II
      W=H(I,I)-P
      R=H(I,EN)
      IF (M .GT. NA) GO TO 320
      DO 310 J=M,NA
310   R=R+H(I,J)*H(J,EN)
320   IF (WI(I) .GE. 0.0D0) GO TO 330
      ZZ=W
      S=R
      GO TO 360
330   M=I
      IF (WI(I) .NE. 0.0D0) GO TC 340
      T=W
      IF (W .EQ. 0.0D0) T=MACHEP*NORM
      H(I,EN)=-R/T
      GO TO 360
C-------------------------SOLVE REAL EQUATICNS--------------------------
340   X=H(I,I+1)
      Y=H(I+1,I)
      Q=(WR(I) - P)*(WR(I) - P)+WI(I)*WI(I)
      T=(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 36C
350   H(I+1,EN)=(-S - Y * T)/ZZ
360   CONTINUE
C--------------------------END REAL VECTCR------------------------------
      GO TO 450
C--------------------------COMPLEX VECTOR-------------------------------
```

```
370    M=NA
C-----------------LAST VECTOR COMPONENT CHOSEN IMAGINARY SO THAT---------
C-----------------EIGENVECTOR MATRIX IS TRIANGULAR-----------------------
       IF (DABS(H(EN,NA)) .LE. DABS(H(NA,EN))) GO TO 380
       H(NA,NA)=Q/H(EN,NA)
       H(NA,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(NA,EN)=DIMAG(Z3)
390    H(EN,NA)=0.0D0
       H(EN,EN)=1.0D0
       ENM2=NA-1
       IF (ENM2 .EQ. 0) GO TO 450
       DO 440 II=1,ENM2
       I=NA-II
       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)
400    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=(WR(I) - P)*(WR(I) - P)+WI(I)*WI(I)-Q*Q
       VI=(WR(I) - P)*2.0D0*Q
       IF (VR .EQ. 0.0D0 .AND. VI .EQ. 0.0D0) VR=MACHEP*NORM*(DABS(W) + D
      1ABS(Q) + DABS(X) + DABS(Y) + DABS(ZZ))
       Z3=DCMPLX(X*R-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)
440    CONTINUE
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
460    Z(I,J)=H(I,J)
470    CONTINUE
C-----------------MULTIPLY BY TRANSFORMATION MATRIX TO GIVE-------------
C-----------------VECTORS OF ORIGINAL FULL MATRIX.-------------------------
       DO 490 JJ=LOW,N
       J=N+LOW-JJ
       M=MIN0(J,IGH)
       DO 490 I=LOW,IGH
       ZZ=0.0DC
       DO 480 K=LOW,M
480    ZZ=ZZ+Z(I,K)*H(K,J)
       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,NM,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)
C-----------------LEFT HAND EIGENVECTORS ARE BACK TRANSFORMED------------
C-----------------IF THE FOREGOING STATEMENT IS REPLACED BY--------------
C-----------------S=1.0D0/SCALE(I).----------------------------------------
       DO 10 J=1,M
10     Z(I,J)=Z(I,J)*S
20     CONTINUE
30     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
40    CONTINUE
50    CONTINUE
60    RETURN
      END
C=========================================================================
      SUBROUTINE HQR (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
      REAL*8 H(NM,N),WR(N),WI(N)
      REAL*8 P,Q,R,S,T,W,X,Y,ZZ,NORM,MACHEP
      REAL*8 DSQRT,DABS,DSIGN
      INTEGER MINO
      LOGICAL NOTLAS
      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
10    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
20    CONTINUE
      EN=IGH
      T=0.0D0
C-----------------------SEARCH FOR NEXT EIGENVALUES---------------------
30    IF (EN .LT. LOW) GO TO 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-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=NORM
      IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 60
50    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 TO 240
      IF (ITS .NE. 10 .AND. ITS .NE. 20) GO TO 80
C------------------------FORM EXCEPTIONAL SHIFT------------------------
      T=T+X
      DO 70 I=LOW,EN
70    H(I,I)=H(I,I)-X
      S=DABS(H(EN,NA))+DABS(H(NA,ENM2))
      X=0.75D0*S
      Y=X
      W=-0.4375D0*S*S
80    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
90    CONTINUE
100   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
110   CONTINUE
C-----DOUBLE QR 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
120   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,EN
      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
150   H(K+1,J)=H(K+1,J)-P*Y
      H(K,J)=H(K,J)-P*X
160   CONTINUE
      J=MIN0(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+2)-P*R
170   H(I,K+1)=H(I,K+1)-P*Q
      H(I,K)=H(I,K)-P
180   CONTINUE
190   CONTINUE
      GO TO 40
C----------------------------ONE ROOT FOUND---------------------------
200   WR(EN)=X+T
      WI(EN)=0.0D0
      EN=NA
      GO TO 30
C----------------------------TWO 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
230   EN=ENM2
      GO TO 30
C-----------------SET ERRCR -- NO CONVERGENCE TO AN-----------------------
C-----------------EIGENVALUE AFTER 30 ITERATIONS---------------------------
240   IERR=EN
250   RETURN
      END
C==========================================================================
      SUBROUTINE PSDCAL (N2,NS,FA,X,NC,GW,GV ,C,NC,HY,HU,H,
     1  FBGE,NG,GAM,ACL,F,WR,WI,D1,D2,JCF,RES,Q,R,BB,CC,IYU,
     2  IPSD,INORM)
C==========================================================================
C     =  PSDCAL COMPUTES THE PSD OF OUTPUTS OR CONTROLS OF                 =
C     =   A CONTROLLED SYSTEM                                              =
C     =                                                                    =
C     =      IYU= 1      OUTPUT PSD                                        =
C     =         = 2      CONTROL PSD                                       =
C     =         = 3      BOTH OUTPUT AND CONTROL PSD                       =
C     =                                                                    =
C     =      IPSD=1      PSD                                               =
C     =          =2      PSD AND TF RESIDUES                               =
C     =                                                                    =
C     =                                                                    =
C     =      INORM=      1,2,... NG  NORMALIZED BY ITH PROCESS NOISE       =
C     =                  NG+1,... NG+NC NORMALIZED BY ITH MEAS NOISE    =
C==========================================================================
      DOUBLE PRECISION FA,X,GW,GV,C,HY,H,FBGE,GAM,ACL,F,WR,WI,D1,D2,RES,
     1BB,CC,Q,R,PSD,W,DNORM,DN1,EMAX,ELOG,EMCC,DW,ST,OM,RE,AI,HU,DW1
      COMPLEX *16ZD,ZN,ZZ
      DIMENSION FA(N2,N2),X(N2,N2),GW(N2,NG),C(NC,NS),HY(NO,N2),H(NO,NS)
     1,FBGE(NS,NO),GAM(NS,NG),ACL(NS,NS),F(NS,NS),WR(N2),WI(N2),D1(N2),D
     22(N2),RES(N2),Q(NG,NG),R(NC,NO),PSD(30),W(30),BB(N2),CC(N2),GV(N2,
     3NO),HU(NC,N2),DW1(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 (IPSC .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)
10    FA(NS+I,J)=0.D0
      DO 30 I=1,NS
      DO 30 J=1,NS
      ST=0.DC
      DO 20 K=1,NC
20    ST=ST+FEGE(I,K)*H(K,J)
      FA(I,NS+J)=-ST
30    FA(NS+I,NS+J)=F(I,J)-ST
      CALL RAPRNT (N2,N2,N2,9,FA,4,'(9(1X,1PD13.6))')
C-------------------------------DEBUG ABOVE-------------------------------
      CALL BALANC (N2,N2,FA,LOW,IHIGH,D1)
      CALL ORTHES (N2,N2,LOW,IHIGH,FA,D2)
      CALL ORTRAN (N2,N2,LOW,IHIGH,FA,D2,X)
      CALL HQR2 (N2,N2,LOW,IHIGH,FA,WR,WI,X,IERR)
      IF (IERR .NE. 0) GO TO 320
      CALL BALBAK (N2,N2,LOW,IHIGH,D1,N2,X)
      CALL RAPRNT (N2,N2,N2,9,X,4,'(9(1X,1PD13.6))')
C------------DEBUG ABOVE;DETERMINE MODAL MATRICES------------------------
      IF (IYU .EQ. 1) GO TO 60
C---------------------------HSUBU-----------------------------------------
      DO 50 I=1,NC
      DO 50 J=1,N2
      ST=0.DC
      DO 40 K=1,NS
40    ST=ST-C(I,K)*X(K,J)
50    HU(I,J)=ST
      GO TO 90
C---------------------------HSUBY-----------------------------------------
60    DO 80 I=1,NO
      DO 80 J=1,N2
      ST=0.DC
      DO 70 K=1,NS
70    ST=ST+H(I,K)*X(K,J)-H(I,K)*X(NS+K,J)
80    HY(I,J)=ST
      CALL RAPRNT (NO,NO,N2,9,HY,4,'(9(1X,1PD13.6))')
C---------------------------DEBUG ABOVE-----------------------------------
90    CALL MINV (NSQ,X,N2,ST,D1,D2)
      CALL RAPRNT (N2,N2,N2,9,X,4,'(9(1X,1PD13.6))')
C---------------------------DEBUG ABOVE-----------------------------------
```

```
C-----------------------------GSUBW--------------------------------------
      DO 110 I=1,N2
      DO 110 J=1,NG
      ST=0.0D0
      DO 100 K=1,NS
100   ST=ST-X(I,NS+K)*GAM(K,J)
110   GW(I,J)=ST
      CALL RAPRNT (N2,N2,NG,9,GW,4,'(9(1X,1PD13.6))')
C---------DEBUG ABOVE; USE SELECTED NORMALIZATION-----------------------
      IF (INORM .LE. NG) DNORM=1.D0/Q(INORM,INORM)
      IF (INORM .GT. NG) DNORM=1.D0/R(INORM-NG,INORM-NG)
C----------------DETERMINE BANDWIDTH OF CONTROLLED SYSTEM---------------
      EMAX=0.D0
      DO 120 I=1,N2
      EMOD=DABS(WR(I)**2 +WI(I)**2)
      IF (EMOD .GT. EMAX) EMAX=EMOD
120   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.D1
      IK=3
      GO TO 140
130   EMAX=5.D0
      IK=2
140   CONTINUE
C-----------------------STORE 30 FREQUENCIES----------------------------
      DO 150 I=1,20
150   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)=CW1(IX)*10**(IPOW+I-1+JJ+IK-2)
160   CONTINUE
      IX=MOD(IK,3)+1
      W(30)=CW1(IX)*10**(IPOW+3 +IK-2)
C---------------------LARGE LOOP THRU OUTPUTS------------------------
      IF (IYU .EQ. 1) NL=NO
      IF (IYU .EQ. 2) NL=NC
      DO 310 L=1,NL
      DO 170 I=1,30
170   PSD(I)=0.D0
C---------------------LOOP THRU PROCESS NOISE------------------------
      DO 220 I=1,NG
      DN1=DNORM*Q(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=DCMPLX(0.D0,0.D0)
      OM=W(K)
      DO 200 II=1,N2
      IF (WI(II)) 200,180,190
180   ZD=DCMPLX(-WR(II),OM-WI(II))
      ZZ=RES(II)/ZD+ZZ
      GO TO 200
190   RE=WR(II)
      AI=WI(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
200   CONTINUE
210   PSD(K)=PSD(K)+DN1*(ZZ*DCONJG(ZZ))
220   CONTINUE
C-----------------------------GSUBV----------------------------------
      DO 240 I=1,N2
      DO 240 J=1,NO
      ST=0.D0
      DO 230 K=1,NS
230   ST=ST+X(I,K)*FBGE(K,J)+X(I,NS+K)*FBGE(K,J)
240   GV(I,J)=ST
      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*R(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 (IYL.EQ.1) CALL RESID (I,L,N2,JCF,NO,GV,NL,HY,WR,WI,RES,
     1     BB,CC,IPT)
      IF (IYL.EQ.2) CALL RESID (I,L,N2,JCF,NC,GV,NL,HU,WR,WI,RES,
     1     BB,CC,IPT)
      DO 290 K=1,30
      ZZ=DCMPLX(0.D0,0.D0)
      CM=W(K)
      DO 270 II=1,N2
      IF (WI(II)) 270,250,260
250   ZD=DCMPLX(-WR(II),CM-WI(II))
      ZZ=ZZ+RES(II)/ZD
      GO TO 270
260   RE=WR(II)
      AI=WI(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 (IYL .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 (IYL .EQ. 1) WRITE (6,390) L
      IF (IYU .EQ. 2) WRITE (6,400) L
      WRITE (6,410) (W(I),PSD(I),I=1,30)
310   CONTINUE
      RETURN
320   CONTINUE
      CALL EREXIT (N2,FA,IERR)
      RETURN
C-----------------------------------------------------------------------
330   FORMAT (/,41H SUBSEQUENT PSD IS NORMALIZED BY MEAS NO.,I3,//)
340   FORMAT (/,50H SUBSEQUENT PSD IS NORMALIZED BY PROCESS NOISE NO.,I3
     1,//)
350   FORMAT (/38H TRANSFER FUNCTION FROM PROCESS NOISE ,I2,3H TO,13H ME
     1ASUREMENT ,I2,//)
360   FORMAT (/38H TRANSFER FUNCTION FROM PROCESS NOISE ,I2,3H TO,9H CON
     1TROL ,I2,//)
370   FORMAT (/36H TRANSFER FUNCTION FROM MEASUREMENT ,I2,16H TO MEASURE
     1MENT ,I2,//)
380   FORMAT (/36H TRANSFER FUNCTION FROM MEASUREMENT ,I2,12H TO CONTROL
     1 ,I2,//)
390   FORMAT (/14H PSD OF OUTPUT,I3,32H  FORCED BY ALL NOISE-(RAD FREQ,,
     115HNORMALIZED PSD)/)
400   FORMAT (/15H PSD OF CONTROL,I3,32H  FORCED BY ALL NOISE-(RAD FREQ,
     1,15HNORMALIZED PSD)/)
410   FORMAT (4(1X,1H(,E11.4,1H,,E11.4,1H)))
```

```
      END
C=====================================================================
      SUBROUTINE EREXIT (N,A,IERR)
C     EREXIT RETURNS THE NUMBER OF THE EIGENVALUE WHERE HQR2          =
C     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
10    FORMAT (35H FAILURE IN HQR2 ON EIGENVALUE NO. ,I3)
      END
C=====================================================================
      SUBROUTINE READF (NS,ISAF,BA)
C     INTERACTIVELY INPUTS THE "F" MATRIX ELEMENT BY ELEMENT.         =
C=====================================================================
      REAL*8  BA(NS,NS),DUM,ANSR
      INTEGER I,J,K,L,IANS,ISAF
      DATA IY/'Y'/,IZ/'N'/
      IF (ISAF.EQ.1) GO TO 40
      WRITE (5,130)
      DO 20 I=1,NS
      DO 10 J=1,NS
      WRITE (5,120) I,J
      CALL RDREAL (ANSR)
      BA(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C---------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
40    CONTINUE
      WRITE (5,140)
      CALL MATPRT (BA,NS,NS)
50    WRITE (5,150)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
      GO TO 70
60    WRITE (5,160)
      GO TO 50
70    CONTINUE
      IF (IANS.EQ.IZ) GO TO 110
      IF (IANS.EQ.IY) GO TO 80
80    WRITE (5,170)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,180)
```

```
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,120) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 100 I=1,NS
      DO 90 J=1,NS
      IF ((I.EQ.K).AND.(J.EQ.L)) BA(I,J)=DUM
90    CONTINUE
100   CONTINUE
      GO TO 30
110   CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
120   FORMAT (5X,14HTHE ELEMENT F(,I2,1H,,I2,2H)=)
130   FORMAT (/,5X,36HENTER THE SYSTEM MATRIX  "F"-MATRIX&,//,10X,41HDIM
     1ENSION = # STATES  NS& X # STATES  NS&)
140   FORMAT (//,15X,33HTHE SYSTEM MATRIX  "F"-MATRIX&...,//)
150   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
160   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY   ENTER "YES" OR "NO".)
170   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
180   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C=======================================================================
      SUBROUTINE READH (NO,NS,ISAH,HO)
C     INTERACTIVELY INPUTS THE "H" MATRIX  MEASUREMENT SCALING MATRIX&.=
C=======================================================================
      REAL*8  HO(NO,NS),DUM,ANSR
      INTEGER IANS,I,J,K,L,ISAH
      DATA IY/'Y'/,IZ/'N'/
C-----------------------------------------------------------------------
C  THIS IS AN EXAMPLE OF ONE POSSIBLE METHOD OF ARRAY GENERATION       =
C  WITHIN THE PROGRAM ITSELF.  FOR VERY LARGE DATA ARRAYS, THIS METHOD =
C  MAY BE PREFERABLE TO SOME USERS OVER INTERACTIVE ENTRY OF EACH      =
C  INDIVIDUAL ELEMENT.                                                 =
C-----------------------------------------------------------------------
C     DO 2 I=1,11
C        DO 1 J=1,82
C             HO(I,J)    = 0.0D+00
C             HO(1,1)    = 0.11520D+00
C             HO(2,75)   = 0.5730D+02
C             HO(3,74)   = 0.1000D+01
C             HO(4,63)   = 0.5730D+02
C             HO(5,62)   = 0.1000D+01
C             HO(6,76)   = 0.5730D+02
```

```
C                 HO(7,44)  = 0.5730D+02
C                 HO(8,45)  = 0.5730D+02
C                 HO(9,46)  = 0.5730D+02
C                 HO(10,47) = 0.5730D+02
C                 HO(11,48) = 0.5730D+02
C1          CONTINUE
C2      CONTINUE
C       GO TO 90
C3      CONTINUE
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 RDREAL (ANSR)
        HO(I,J)=ANSR
10      CONTINUE
20      CONTINUE
C----------------------------------------------------------------------
30      CALL FRTCMS ('CLRSCRN ')
40      CONTINUE
        WRITE (5,130)
        CALL MATPRT (HO,NO,NS)
50      WRITE (5,140)
        CALL RDCHAR (IANS)
        IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
        GO TO 70
60      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 RDREAL (ANSR)
        DUM=ANSR
        DO 90 I=1,NO
        DO 80 J=1,NS
        IF ((I.EQ.K).AND.(J.EQ.L)) HO(I,J)=DUM
80      CONTINUE
90      CONTINUE
        GO TO 30
100     CONTINUE
```

```
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
110   FORMAT (5X,14HTHE ELEMENT H(,I2,1H,,I2,2H)=)
120   FORMAT (/,5X,50HENTER THE MEASUREMENT SCALING MATRIX  "H"-MATRIX&.
     1,//,10X,47HDIMENSION = # OBSERVATIONS  NO& X # STATES  NS&)
130   FORMAT (//,10X,46HTHE MEASUREMENT SCALING MATRIX  "H"-MATRIX&...,/
     1/)
140   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
150   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY   ENTER "YES" OR "NO".)
160   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
170   FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1)
      END
C=======================================================================
      SUBROUTINE READD (NO,NC,D)
C     INPUTS THE "D" MATRIX  MEASUREMENT FEED-FORWARD DIST. MATRIX&.   =
C=======================================================================
      REAL*8  D(NC,NC),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,110)
      DO 20 I=1,NC
      DO 10 J=1,NC
      WRITE (5,100) I,J
      CALL RDREAL (ANSR)
      D(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C-----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,120)
      CALL MATPRT (D,NC,NC)
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
```

```
      WRITE (5,100) K,L
      CALL RCREAL (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
      WRITE (5,120)
      DO 20 I=1,NS
      DO 10 J=1,NC
      WRITE (5,110) I,J
      CALL RCREAL (ANSR)
      G(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C-----------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
40    CONTINUE
      WRITE (5,130)
      CALL MATPRT (G,NS,NC)
50    WRITE (5,140)
```

```
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
      GO TO 70
60    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 RDREAL (ANSR)
      DUM=ANSR
      DO 90 I=1,NS
      DO 80 J=1,NC
      IF ((I.EQ.K).AND.(J.EQ.L)) G(I,J)=DUM
80    CONTINUE
90    CONTINUE
      GO TO 30
100   CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C------------------------------------------------------------------------
110   FORMAT (5X,14HTHE ELEMENT G(,I2,1H,,I2,2H)=)
120   FORMAT (/,5X,51HENTER THE CONTROL DISTRIBUTION MATRIX  "G"-MATRIX&
     1.,//,10X,43HDIMENSION = # STATES  NS& X # CONTROLS  NC&)
130   FORMAT (//,10X,47HTHE CONTROL DISTRIBUTION MATRIX  "G"-MATRIX&...,
     1//)
140   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
150   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY   ENTER "YES" OR "NO".)
160   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
170   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C========================================================================
      SUBROUTINE READFB (NC,NS,FBGC)
C     INPUTS THE "C"     FEEDBACK GAIN CONTROL MATRIX&.                 =
C========================================================================
      REAL*8  FBGC(NC,NS),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,110)
      DO 20 I=1,NC
      DO 10 J=1,NS
```

```
      WRITE (5,100) I,J
      CALL RDREAL (ANSR)
      FBGC(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C---------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,120)
      CALL MATPRT (FBGC,NC,NS)
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
      WRITE (5,100) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 80 I=1,NC
      DO 70 J=1,NS
      IF ((I.EQ.K).AND.(J.EQ.L)) FBGC(I,J)=DUM
70    CONTINUE
80    CONTINUE
      GO TO 30
90    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C---------------------------------------------------------------------
100   FORMAT (5X,14HTHE ELEMENT C(,I2,1H,,I2,2H)=).
110   FORMAT (/,5X,52HENTER THE FEEDBACK GAIN CONTROL MATRIX  "C"-MATRIX
     1&.,//,10X,44HDIMENSION = # CONTROLS  NC& X # STATES  NS&.)
120   FORMAT (//,10X,45HTHE FEEDBACK GAIN CONTROL MATRIX  "C"-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 READAY (NO,ISAA,AY)
C     INPUTS THE "A" MATRIX   DIAGONAL OUTPUT COST MATRIX&.          =
C=====================================================================
      REAL*8  AY(NO,NO),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      IF (ISAA.EQ.1) GO TO 30
      WRITE (5,110)
      DO 20 I=1,NO
      DO 10 J=1,NO
      WRITE (5,100) I,J
      CALL RDREAL (ANSR)
      AY(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C---------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,120)
      CALL MATPRT (AY,NO,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
      WRITE (5,100) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 80 I=1,NO
      DO 70 J=1,NO
      IF ((I.EQ.K).AND.(J.EQ.L)) AY(I,J)=DUM
70    CONTINUE
80    CONTINUE
      GO TO 30
90    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C---------------------------------------------------------------------
100   FORMAT (5X,14HTHE ELEMENT A(,I2,1H,,I2,2H)=)
```

```
110    FORMAT (//,5X,54HENTER THE OUTPUT MEASUREMENT COST MATRIX  "A"-MAT
      1RIX&.,/,5X,53HDIMENSION = # OBSERVATIONS  NO& X # OBSERVATIONS  NO
      2&)
120    FORMAT (//,5X,50HTHE OUTPUT MEASUREMENT COST MATRIX  "A"-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 READB (NC,ISAB,B)
C      INPUTS THE "B" MATRIX  CONTROL COST WEIGHTING MATRIX&.          =
C======================================================================
       REAL*8  B(NC,NC),DUM,ANSR
       INTEGER IANS,I,J,K,L
       DATA IY/'Y'/,IZ/'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 RDREAL (ANSR)
10     B(I,J)=ANSR
C----------------------------------------------------------------------
20     CALL FRTCMS ('CLRSCRN ')
       WRITE (5,100)
       CALL MATPRT (B,NC,NC)
30     WRITE (5,110)
       CALL RDCHAR (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)
       L=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 FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
80    FORMAT (5X,14HTHE ELEMENT B(,I2,1H,,I2,2H)=)
90    FORMAT (/,5X,52HENTER THE CONTROL COST WEIGHTING MATRIX  "B"-MATRI
     1X&,/,10X,45HDIMENSION = # CONTROLS  NC& X # CONTROLS  NC&)
100   FORMAT (//,10X,37HTHE CONTROL COST MATRIX..........B...,//)
110   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
120   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY   ENTER "YES" OR "NO".)
130   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
140   FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1)
      END
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/'Y'/,IZ/'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
10    CONTINUE
20    CONTINUE
C-----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
40    CONTINUE
      WRITE (5,130)
      CALL MATPRT (GAM,NS,NG)
50    WRITE (5,140)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
      GO TO 70
60    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 RDREAL (ANSR)
      DUM=ANSR
      DO 90 I=1,NS
      DO 80 J=1,NG
      IF ((I.EQ.K).AND.(J.EQ.L)) GAM(I,J)=DUM
80    CONTINUE
90    CONTINUE
      GO TO 30
100   CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
110   FORMAT (5X,16HTHE ELEMENT GAM(,I2,1H,,I2,2H)=)
120   FORMAT (/,5X,36HENTER THE PROCESS NOISE DISTRIBUTION,/,5X,24HMATRI
     1X "GAMMA"-MATRIX&.,/,2X,56HDIMENSION = # STATES  NS& X # PROCESS
     2NOISE SOURCES  NG&)
130   FORMAT (//,10X,37HTHE PROCESS NOISE DISTRIBUTION MATRIX,/,10X,19H
     1"GAMMA"-MATRIX&...,//)
140   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
150   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY   ENTER "YES" OR "NO".)
160   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
170   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C=======================================================================
      SUBROUTINE READQ (NG,Q)
C     INTERACTIVELY INPUTS THE "Q" MATRIX  NOISE WEIGHTING MATRIX&     =
C=======================================================================
      REAL*8  Q(NG,NG),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,110)
      DO 20 I=1,NG
      DO 10 J=1,NG
      WRITE (5,100) I,J
      CALL RDREAL (ANSR)
      Q(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C-----------------------------------------------------------------------
```

```
30    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,120)
      CALL MATPRT (Q,NG,NG)
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
      WRITE (5,100) K,L
      CALL RDREAL (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
70    CONTINUE
80    CONTINUE
      GO TO 30
90    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C----------------------------------------------------------------------
100   FORMAT (5X,14HTHE ELEMENT Q(,I2,1H,,I2,2H)=)
110   FORMAT (//,5X,44HENTER THE PROCESS NOISE PSD WEIGHTING MATRIX,/,5X
     1,12H "Q"MATRIX&.,//,5X,42HDIMENSION = # PROCESS NOISE SOURCES  NG&
     2 X,/,17X,27H#PROCESS NOISE SOURCES  NG&)
120   FORMAT (//,5X,42HTHE PROCESS NOISE WEIGHTING MATRIX.....Q..,//)
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 READR (NO,RC)
C     INTERACTIVELY INPUTS THE "R" MATRIX=
C      MEASUREMENT NOISE DISTRIBUTION MATRIX.&                        =
C======================================================================
      REAL*8  RC(NO,NO),DUM,ANSR
```

```
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,90)
      DO 10 I=1,NO
      DO 10 J=1,NC
      WRITE (5,80) I,J
      CALL RDREAL (ANSR)
10    RC(I,J)=ANSR
C-----------------------------------------------------------------------
20    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,100)
      CALL MATPRT (RC,NO,NO)
30    WRITE (5,110)
      CALL RDCHAR (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)
      L=IANS
      WRITE (5,80) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 60 I=1,NO
      DO 60 J=1,NC
60    IF ((I.EQ.K).AND.(J.EQ.L)) RC(I,J)=DUM
      GO TO 20
70    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
80    FORMAT (5X,14HTHE ELEMENT R(,I2,1H,,I2,2H)=)
90    FORMAT (/,5X,60HENTER THE MEASUREMENT NOISE DISTRIBUTION MATRIX  "
     1R"MATRIX&.,//,5X,53HDIMENSION = # OBSERVATIONS  NO& X # OBSERVATIO
     2NS  NO&)
100   FORMAT (//,15X,50HTHE MEASUREMENT NOISE DISTRIBUTION MATRIX.....R.
     1..,//)
110   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
120   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY   ENTER "YES" OR "NO".)
130   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
140   FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
```

```
      1)
       END
C=============================================================================
       SUBROUTINE READFE (NS,NO,FBGE)
C      INTERACTIVELY INPUTS THE "K"  FEEDBACK GAIN ESTIMATOR MATRIX&     =
C=============================================================================
       REAL*8  FBGE(NS,NO),DUM,ANSR
       INTEGER IANS,I,J,K,L
       DATA IY/'Y'/,IZ/'N'/
       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
10     CONTINUE
20     CONTINUE
C-----------------------------------------------------------------------------
30     CALL FRTCMS ('CLRSCRN ')
       WRITE (5,120)
       CALL MATPRT (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
       WRITE (5,100) K,L
       CALL RDREAL (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
70     CONTINUE
80     CONTINUE
       GO TO 30
90     CONTINUE
       CALL FRTCMS ('CLRSCRN ')
       RETURN
C-----------------------------------------------------------------------------
```

```
100   FORMAT (5X,14HTHE ELEMENT K(,I2,1H,,I2,2H)=)
110   FORMAT (/,5X,54HENTER THE FEEDBACK GAIN ESTIMATOR MATRIX  "K"-MATR
     1IX&.,//,10X,48HDIMENSION = # STATES  NS& X # OBSERVATIONS  NO&.)
120   FORMAT (//,15X,47HTHE FEEDBACK GAIN ESTIMATOR MATRIX  "K"-MATRIX&,
     1//)
130   FORMAT (//,5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELE
     1MENT?,//,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,52HENTER THE COLUMN NUMBER CF THE ELEMENT TO BE CHANGED
     1)
      END
C=====================================================================
      SUBROUTINE READW (NG,WR)
C     INTERACTIVELY INPUTS THE "WO" MATRIX  STEADY DISTURBANCE VECTOR =
C     MATRIX& ELEMENT BY ELEMENT.                                     =
C=====================================================================
      REAL*8  WR(NG),DUM,ANSR
      INTEGER IANS,I,K
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,100)
      DO 10 I=1,NG
      WRITE (5,80) I
      CALL RDREAL (ANSR)
      WR(I)=ANSR
10    CONTINUE
C---------------------------------------------------------------------
20    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,110)
      WRITE (5,90) (WR(I),I=1,NG)
30    WRITE (5,120)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 40
      GO TO 50
40    WRITE (5,130)
      GO TO 30
50    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
60    CONTINUE
      GO TO 20
```

```
70     CONTINUE
       CALL FRTCMS ('CLRSCRN ')
       RETURN
C-----------------------------------------------------------------------
80     FORMAT (5X,15HTHE ELEMENT WO(,I2,2H)=)
90     FORMAT (F12.5)
100    FORMAT (/,5X,57HENTER THE STEADY DISTURBANCE VECTOR MATRIX  "WO"-M
      1ATRIX&.,//,10X,44HDIMENSION = # PROCESS NOISE SOURCES  NG& X 1)
110    FORMAT (//,15X,53HTHE STEADY DISTURBANCE VECTOR MATRIX  "WO"-MATRI
      1X&...,//)
120    FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
      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 INADVERTENLY 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-----------------------------------------------------------------------
       COUNT=0
10     CONTINUE
       COUNT=COUNT+1
       IF (COUNT.LT.3) GO TO 20
       WRITE (5,60)
       GO TO 40
20     CONTINUE
       READ (5,*,END=30,ERR=30) ANSR
       RETURN
30     REWIND 5
       WRITE (5,50)
       GO TO 10
40     CONTINUE
       STOP
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 INADVERTENLY ENTERS AN IMPROPER=
C  DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.      =
C=======================================================================
```

```
      SUBROUTINE RDINT (IANS)
      INTEGER CCUNT,IANS
C-----------------------------------------------------------------------
      COUNT=C
10    CONTINUE
      COUNT=CCUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 5C
20    CONTINUE
      READ (5,*,END=40,ERR=40) IANS
      IF (IANS) 4C,40,30
30    CONTINUE
      RETURN
40    REWIND 5
      WRITE (5,70)
      GO TO 1C
50    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 INADVERTENLY    =
C  ENTERS  A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY=
C=======================================================================
      SUBROUTINE RDCHAR (IANS)
      INTEGER CCUNT,IANS
C     DATA IY/'Y'/,IZ/'N'/
C-----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=CCUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      REWIND 5
      READ (5,70,END=30,ERR=30) IANS
      RETURN
30    REWIND 5
      WRITE (5,50)
      GO TO 1C
40    CONTINUE
```

```
      STCP
C------------------------------------------------------------------------
50    FORMAT (1X,60HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER "YES"
     1OR "NO".)
60    FORMAT (///,5X,47HPROGRAM TERMINATICN - TWO NULL STRINGS ENTEREC )
70    FORMAT (A1)
      END
C========================================================================
C  SLBROUTINE MATPRT -- CISPLAYS 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,O-Z)
      DIMENSICN PRTT(NROW,NCOL)
C------------------------------------------------------------------------
      IF (NCCL.EQ.0)  NCOL=1
      IF (NCCL.EQ.1)  WRITE (5,10)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.2)  WRITE (5,20)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.3)  WRITE (5,30)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.4)  WRITE (5,40)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.5)  WRITE (5,50)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.6)  WRITE (5,60)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.7)  WRITE (5,70)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.8)  WRITE (5,80)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.9)  WRITE (5,90)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.10) WRITE (5,100) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.11) WRITE (5,110) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.12) WRITE (5,120) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.13) WRITE (5,130) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.14) WRITE (5,140) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.15) WRITE (5,150) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.16) WRITE (5,160) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      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,/,F12.5,//)
80    FORMAT (6F12.5,/,2F12.5,//)
90    FORMAT (6F12.5,/,3F12.5,//)
100   FORMAT (6F12.5,/,4F12.5,//)
110   FORMAT (6F12.5,/,5F12.5,//)
120   FORMAT (6F12.5,/,6F12.5,//)
130   FORMAT (6F12.5,/,6F12.5,/,F12.5,//)
140   FORMAT (6F12.5,/,6F12.5,/,2F12.5,//)
```

```
150    FORMAT (6F12.5,/,6F12.5,/,3F12.5,//)
160    FORMAT (6F12.5,/,6F12.5,/,4F12.5,//)
       ENC
C==========================================================================
C  SLBROUTINE RDMATF -- READS THE FLAGS AND MATRIX SIZES FROM            =
C  THE DATA FILE CN FILECEF 9.  ASKS IF YOU WANT TO USE THE MATRICES.    =
C==========================================================================
       SUBROUTINE RDMATF (NS,NC,NOB,NG,ISAF,ISAG,ISAH,IGAM,ISAA,ISAB,IRDM
      1AT)
       DATA IYES/'Y'/,INO/'N'/
C      INTEGER NS,NC,NOB,NG,ISAF,ISAG,ISAH,IGAM,IRDMAT,INO,IANS,K
       REWIND 9
       READ (9,240,END=30,ERR=30) K,IANS
       IF (IANS.EQ.1) GC TC 10
       GO TO 3C
10     READ (9,250) NS,NC,NOB,NG
       WRITE (5,255)
       CALL FRTCMS ('CLRSCRN ')
20     WRITE (5,260)
       CALL RCINT (IANS)
       IF (IANS.GT.3) GO TO 20
       IF (IANS.EQ.3) GO TC 30
       IRDMAT=1
       IF (IANS.EQ.2) GO TO 40
       ISAF=1
       ISAG=1
       ISAH=1
       IGAM=1
       ISAA=1
       ISAB=1
30     RETURN
C----------------------------------------------------------ISAF--------------------
40     CALL FRTCMS ('CLRSCRN ')
50     WRITE (5,27C)
       CALL RCCHAR (IANS)
       IF ((IANS.EC.IYES).OR.(IANS.EQ.INO)) GC TO 70
60     WRITE (5,330)
       GO TO 50
70     CONTINUE
       IF (IANS.EQ.IYES) ISAF=1
       IF (IANS.EC.INO) ISAF=0
C----------------------------------------------------------ISAH--------------------
       IF (NOB.EQ.0) GO TO 110
       CALL FRTCMS ('CLRSCRN ')
80     WRITE (5,28C)
       CALL RCCHAR (IANS)
       IF ((IANS.EC.IYES).OR.(IANS.EQ.INO)) GC TO 100
90     WRITE (5,33C)
```

```
      GO TO 80
100   CONTINUE
      IF (IANS.EQ.IYES) ISAH=1
      IF (IANS.EQ.INO) ISAH=0
110   CONTINUE
C----------------------------------------------------ISAG--------------------
      IF (NC.EQ.0) GO TO 150
      CALL FRTCMS ('CLRSCRN ')
120   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
150   CONTINUE
C----------------------------------------------------IGAM--------------------
      IF (NG.EQ.0) GO TO 190
      CALL FRTCMS ('CLRSCRN ')
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
C----------------------------------------------------ISAA--------------------
      CALL FRTCMS ('CLRSCRN ')
200   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
C----------------------------------------------------ISAB--------------------
      CALL FRTCMS ('CLRSCRN ')
220   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
C-----------------------------------------------------------------------
240   FORMAT (I1,3X,I1)
250   FORMAT (4I5)
255   FORMAT (///////////////////////)
260   FORMAT (//,10X,46HTHE "F", "G", "H", "GAM", "A" AND "B" MATRICES,
     1/,12X,42H FROM YOUR PREVIOUS OPTSYS RUN WERE SAVED.,//,10X,36HTHE
     2FOLLOWING OPTIONS ARE AVAILABLE:,/15X,38H1. USE ALL OF THE SAME MA
     3TRICES AGAIN.,/,15X,'2. USE SELECTED MATRICES AGAIN.',/,
     415X,'3. INPUT ALL NEW MATRICES.',//,10X,17HENTER 1, 2, OR 3.
     5,//,10X,'NOTE: EACH SAVED MATRIX WILL BE REDISPLAYED AT'
     6,/,10X,34HTHE PROPER INPUT SEQUENCE INTERVAL
     7,/,10X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,10X,
     827HINDIVIDUAL MATRIX ELEMENTS.)
270   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "F"-MATRIX FROM THE LAST
     1,/,5X,  'RUN TO BE USED IN THIS RUN?'         ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
280   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "H"-MATRIX FROM THE LAST
     1,/,5X,  'RUN TO BE USED IN THIS RUN?'         ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
290   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "G"-MATRIX FROM THE LAST
     1,/,5X,  'RUN TO BE USED IN THIS RUN?'         ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
300   FORMAT (///,5X,52HDO YOU WISH TO SAVE THE "GAMMA"-MATRIX FROM THE
     1LAST,/,5X,  'RUN TO BE USED IN THIS RUN?'         ,//,5X,39HNOTE: T
     2HE MATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE
     3 INTERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27
     4HINDIVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
310   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "A"-MATRIX FROM THE LAST
     1,/,5X,  'RUN TO BE USED IN THIS RUN?'         ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
320   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "B"-MATRIX FROM THE LAST
     1,/,5X,  'RUN TO BE USED IN THIS RUN?'         ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL 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, GAM, A AND B MATRICES FROM    =
C   MATRICES FROM THE DATA FILE OPTMAT ON FILEDEF 9.                      =
C=========================================================================
      SUBROUTINE RDMAT(BA,G,HO,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG,IRDMAT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BA(NS,NS),G(NS,NC),HO(NO,NS),GAM(NS,NG),FBGC(NC,NS),
     1AY(NO,NC),B(NC,NC),FBGE(NS,NO)
      IF(IRDMAT.EQ.0) RETURN
      REWIND 9
      READ (9,20) K,IANS
      READ (9,20) NSI,NCI,NOI,NGI
      READ(9,10) ((BA(I,J),J=1,NSI),I=1,NSI)
      READ(9,10) ((G(I,J),J=1,NCI),I=1,NSI)
      READ(9,10) ((HO(I,J),J=1,NSI),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) ((FBGE(I,J),J=1,NOI),I=1,NSI)
      READ(9,10) ((AY(I,J),J=1,NOI),I=1,NCI)
      READ(9,10) ((B(I,J),J=1,NCI),I=1,NCI)
      RETURN
C-------------------------------------------------------------------------
10    FORMAT(4(D20.13))
20    FORMAT(4I5)
      END
C=========================================================================
C   SUBROUTINE WRTMAT -- WRITES THE F, G, HO & GAM MATRICES TO            =
C   THE DATA FILE OPTMAT ON FILEDEF 9.                                    =
C=========================================================================
      SUBROUTINE WRTMAT(BA,G,HO,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BA(NS,NS),G(NS,NC),HO(NO,NS),GAM(NS,NG),FBGC(NC,NS),
     1AY(NO,NC),B(NC,NC),FBGE(NS,NO)
C     INTEGER NS,NC,NO,NG,I,J,IANS,INO,IYES
      DATA IYES/'Y'/,INO/'N'/
      WRITE (5,90)
      CALL FRTCMS('CLRSCRN ')
10    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
      REWIND 9
      I = 0
      IANS = 1
      WRITE(9,140) I,IANS
      WRITE(9,120) NS,NC,NO,NG
      WRITE(9,130) ((BA(I,J),J=1,NS),I=1,NS)
```

```
      WRITE(9,130) ((G(I,J),J=1,NC),I=1,NS)
      WRITE(9,130) ((HO(I,J),J=1,NS),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,NS)
      WRITE(9,130) ((AY(I,J),J=1,NO),I=1,NC)
      WRITE(9,130) ((B(I,J),J=1,NC),I=1,NC)
      STOP
C-----------------------------------------------------------------------
90    FORMAT(//////////////////////)
100   FORMAT(///,10X,'DO YOU WISH TO OBTAIN A TIME RESPONSE',/,
     112X,'OF THE SYSTEM YOU ARE EVALUATING?',/,24X,'(Y OR N)',
     2///,5X,'NOTE: YOU MUST BE LOGGED ON AT A DUAL SCREEN',/,
     310X,'(TEK 618) TERMINAL TO UTILIZE THIS MODE.',///,10X,
     458HTHE F (SYSTEM), G (CONTROL), H (OBSERVABLES), GAM (NOISE),,/,
     513X,53HA (OUTPUT COST) AND B (CONTROL COST) MATRICES WILL BE,/,
     616X,'SAVED FOR REENTRY TO THE MAIN OPTSYS PROGRAM.')
110   FORMAT(10X,29HYOU MUST ANSWER (Y)ES OR (N)O  )
120   FORMAT(4I5)
130   FORMAT(4D20.13)
140   FORMAT(I1,3X,I1)
      END
```

```
C*************************************************************************
C
C                              OPTCALC
C
C          CALCULATES THE GRAPHICAL TIME RESPONSE OF A
C          HIGH ORDER SYSTEM UNDER THE CONTROL OF THE
C                         OPTSYS EXEC
C
C                              BY
C                          H. A. DIEL
C                         15 JUL 1984
C
C*************************************************************************
C=========================================================================
C=========================================================================
C
C MAIN PROGRAM - PERFORMS INTEGRATION OVER THE DESIRED TIME SPAN
C
C=========================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION F(32,32),G(32,10),HO(32,32),FBGC(32,32),DREND(10),DRBEG(
     110),UMAX(10),XDOT(32),X(32),WK(44000),U(10),IWK(100),ITYPE(10),FBG
     2E(32,32),HK(32,32),DUMMY(32,32),BG(32,32)
      DATA IYES/'Y'/,INO/'N'/
      COMMON F,G,L,NC
      COMMON /A/ ITYPE,DREND,DRBEG,UMAX
      EXTERNAL FCN,FCNJ
C-------------------------------------------------------------------------
C-------------------------------------------------------------------------
C  SUPPRESS INDIVIDUAL UNDERFLOW, OVERFLOW, DIVIDE CHECK, AND DECIMAL  =
C  CONVERT ERROR MESSAGES; PROVIDE SUMMARY OF ERRORS ONLY.             =
C-------------------------------------------------------------------------
      CALL ERRSET (207,256,-1,1,1,209)
      CALL ERRSET (215,256,-1,1)
C-------------------------------------------------------------------------
C         READ IN DATA FILE
C-------------------------------------------------------------------------
10    REWIND 9
      READ (9,600) I,J
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,280)
      READ (9,590) NS,NC,NO,NG
      CALL RDMAT (F,G,HO,FBGC,FBGE,NS,NC,NO,NG)
C-------------------------------------------------------------------------
C         DISPLAY INPUT MATRICES
C-------------------------------------------------------------------------
      CALL NEWSCR
      WRITE (5,290)
```

```
      CALL MATPRT (F,NS,NS)
      CALL NEWSCR
      WRITE (5,300)
      CALL MATPRT (G,NS,NC)
      CALL NEWSCR
      WRITE (5,310)
      CALL MATPRT (FBGC,NC,NS)
      CALL NEWSCR
      IF (NO.EQ.0) GO TO 20
      WRITE (5,320)
      CALL MATPRT (HO,NO,NS)
      CALL NEWSCR
      WRITE (5,340)
      CALL MATPRT (FBGE,NS,NC)
C-----------------------------------------------------------------------
C          SELECT SYSTEM CALCULATION OPTION
C-----------------------------------------------------------------------
      CALL NEWSCR
20    WRITE (5,330)
      CALL RDINT (IANS)
      IF (IANS.GT.4) GO TO 20
      IESTIM=IANS
      NS2=NS
      GO TO (140,30,60,100), IESTIM
C-----------------------------------------------------------------------
C          SIMPLE CLOSED LOOP
C-----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
      DO 50 I=1,NS
      DO 50 J=1,NS
      SUM=0.0D0
      DO 40 K=1,NC
40    SUM=SUM+G(I,K)*FBGC(K,J)
50    F(I,J)=F(I,J)+SUM
      WRITE (5,350)
      CALL MATPRT (F,NS,NS)
      CALL NEWSCR
      GO TO 140
C-----------------------------------------------------------------------
C          FILTER ONLY CLOSED LOOP SYSTEM
C-----------------------------------------------------------------------
60    CALL FRTCMS ('CLRSCRN ')
      CALL MAMULT (FBGE,NS,NC,HO,NS,HK)
      CALL NEWSCR
      WRITE (5,370)
      CALL MATPRT (HK,NS,NS)
      DO 80 I=1,NS
      DO 70 K=1,NC
```

```
      G(I+NS,K)=G(I,K)
70    CONTINUE
      DO 80 J=1,NS
      DUMMY(I,J)=F(I,J)
      DUMMY(I+NS,J)=HK(I,J)
      DUMMY(I,J+NS)=0.0D00
      DUMMY(I+NS,J+NS)=F(I,J)-HK(I,J)
80    CONTINUE
      NS2=2*NS
      DO 90 I=1,NS2
      DO 90 J=1,NS2
      F(I,J)=DUMMY(I,J)
90    CONTINUE
      CALL NEWSCR
      WRITE (5,380)
      CALL MATPRT (F,NS2,NS2)
      CALL NEWSCR
      WRITE (5,390)
      CALL MATPRT (G,NS2,NC)
      CALL NEWSCR
      GO TO 140
C------------------------------------------------------------------------
C        FILTER & REGULATOR CLOSED LOOP SYSTEM
C------------------------------------------------------------------------
100   CALL FRTCMS ('CLRSCRN ')
      CALL MAMULT (FBGE,NS,NC,HO,NS,HK)
      CALL MAMULT (G,NS,NC,FBGC,NS,BG)
      CALL NEWSCR
      WRITE (5,360)
      CALL MATPRT (BG,NS,NS)
      CALL NEWSCR
      WRITE (5,370)
      CALL MATPRT (HK,NS,NS)
      DO 120 I=1,NS
      DO 110 K=1,NC
      G(I+NS,K)=G(I,K)
110   CONTINUE
      DO 120 J=1,NS
      DUMMY(I,J)=F(I,J)
      DUMMY(I+NS,J)=HK(I,J)
      DUMMY(I,J+NS)=BG(I,J)
      DUMMY(I+NS,J+NS)=F(I,J)+BG(I,J)-HK(I,J)
120   CONTINUE
      NS2=2*NS
      DO 130 I=1,NS2
      DO 130 J=1,NS2
      F(I,J)=DUMMY(I,J)
130   CONTINUE
```

```
      CALL NEWSCR
      WRITE (5,380)
      CALL MATPRT (F,NS2,NS2)
      CALL NEWSCR
      WRITE (5,390)
      CALL MATPRT (G,NS2,NC)
      CALL NEWSCR
      IESTIM=3
C-----------------------------------------------------------------------
C         INPUT INTEGRATION START AND STOP TIMES
C-----------------------------------------------------------------------
140   CALL FRTCMS ('CLRSCRN ')
      WRITE (5,400)
      CALL RDREAL (T)
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,410)
      CALL RDREAL (TSTOP)
C-----------------------------------------------------------------------
C         INPUT NUMBER OF POINTS TO CALCULATE
C-----------------------------------------------------------------------
      CALL FRTCMS ('CLRSCRN ')
150   WRITE (5,420)
      CALL RDINT (NPTS)
      IF (NPTS.GT.500) GO TO 150
      NPTS=NPTS+1
C-----------------------------------------------------------------------
C         SELECT DRIVING FUNCTION & START & STOP TIMES
C-----------------------------------------------------------------------
      WRITE (5,430)
      CALL RDCHAR (IANS)
      DO 160 I=1,NC
      U(I)=0.0D00
      ITYPE(I)=1
      DRBEG(I)=0.D0
      DREND(I)=0.D0
      UMAX(I)=0.D0
160   CONTINUE
      IF (IANS.EQ.IYES) GO TO 170
      GO TO 210
170   DO 200 I=1,NC
180   CALL FRTCMS ('CLRSCRN ')
      WRITE (5,440) I
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.2)) GO TO 190
      WRITE (5,450)
      GO TO 180
190   ITYPE(I)=IANS
      CALL FRTCMS ('CLRSCRN ')
```

```
      WRITE (5,460) I
      CALL RDREAL (ANS)
      DREEG(I)=ANS
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,470) I
      CALL RDREAL (ANS)
      DREND(I)=ANS
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,480) I
      CALL RDREAL (ANS)
      UMAX(I)=ANS
200   CONTINUE
C-----------------------------------------------------------------------
C         INPUT INITIAL CONDITIONS
C-----------------------------------------------------------------------
210   CALL FRTCMS ('CLRSCRN ')
      WRITE (5,490)
      CALL RDCHAR (IANS)
      IF (IANS.EQ.INO) GO TO 230
      DO 220 I=1,NS
      X(I)=0.0D0
220   CONTINUE
      GO TO 240
230   DO 240 I=1,NS
      WRITE (5,500) I
      CALL RDREAL (ANS)
      X(I)=ANS
      IF (IESTIM.NE.3) GO TO 240
      WRITE (5,510) I
      CALL RDREAL (ANS)
      X(I+NS)=ANS
240   CONTINUE
C-----------------------------------------------------------------------
C         LAST CHANCE FOR CORRECTIONS
C-----------------------------------------------------------------------
      CALL FRTCMS ('CLRSCRN ')
250   WRITE (5,520)
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.5)) GO TO (10,140,180,200,260), IANS
      WRITE (5,530)
      CALL NEWSCR
      GO TO 250
260   CONTINUE
C-----------------------------------------------------------------------
C         INTEGRATE OVER THE DESIRED TIME SPAN
C-----------------------------------------------------------------------
      WRITE (8,590) NS,NC,NPTS,IESTIM
      WRITE (8,570) ((FBGC(I,J),J=1,NS),I=1,NC)
```

```
C   MITER = 2-FINITE DIFERENCE, = 3-DIRECTIONAL DERIV, = 0-FUNCTIONAL DER
        MITER=2
C   METH = 1-ADAMS METHOD, = 2-STIFF SYSTEM - GEAR METHOD
        METH=2
        INDEX=1
        TOL=1.0D-10
        NPTS=NPTS-1
        DELTAT=(TSTOP - T)/DFLOAT(NPTS)
        H=1.0D-10
        WRITE (8,570) (T,(U(K),K=1,NC),(X(J),J=1,NS2))
        CALL FRTCMS ('CLRSCRN ')
        WRITE (5,540)
        WRITE (5,580) T,U(1),X(1),X(2),X(3)
        ICOUNT=NPTS/50
        DO 270 I=1,NPTS
        TEND=T+DELTAT
        CALL DGEAR (NS2,FCN,FCNJ,T,H,X,TEND,TOL,METH,MITER,
       1 INDEX,IWK,WK,IER)
        IF (MOD(I,ICOUNT).EQ.0) WRITE (5,580) T,U(1),X(1),X(2),X(3)
        WRITE (8,570) (T,(U(K),K=1,NC),(X(J),J=1,NS2))
        IF (IER.EQ.0) GO TO 270
        IF (IER.EQ.66) WRITE (5,550)
        IF ((IER.NE.132).AND.(IER.NE.133)) GO TO 270
        TOL=TOL*1.0D01
        WRITE (5,560)
270     CONTINUE
        STOP
C=======================================================================
280     FORMAT (//10X,44HDURING THIS SECTION OF THE PROGRAM YOU WILL:,/,/,
       110X,44H- SELECT THE TYPE OF SYSTEM RESPONSE TO PLOT,/,12X,45H(OPEN
       2 LOOP, CLOSED LOOP, OR FILTER/REGULATOR),/,10X,55H- PROVIDE START
       3AND STOP TIME FOR PLOTTING CALCULATIONS,/,10X,55H- SELECT THE TYPE
       4 OF DRIVING FUNCTION(S) (STEP OR RAMP),/,10X,58H- PROVIDE START AN
       5D STOP TIMES FOR THE DRIVING FUNCTION(S),/,10X,40H- PROVIDE DRIVIN
       6G FUNCTION MAGNITUDE(S).,//,15X,28HCLEAR THE SCREEN TO CONTINUE)
290     FORMAT (//,15X,14H THE F MATRIX ,//)
300     FORMAT (//,15X,14H THE G MATRIX ,//)
310     FORMAT (//,15X,14H THE C MATRIX ,//)
320     FORMAT (//,15X,13H THE H MATRIX,//)
330     FORMAT (//,10X,51HTHE FOLLOWING PLOTTING OPTIONS ARE AVAILABLE IF
       1THE,/,10X,45HREQUIRED MATRICES WERE CALCULATED IN OPTSYSX:,//,15X,
       227H1.  OPEN LOOP TIME RESPONSE,/20X,21HXDOT =   F&*X +   G&*UC,//,15
       3X,29H2.  CLOSED LOOP TIME RESPONSE,/20X,39HXDOT =   F-G*C&*X +   G&*
       4UC,    U =    C&*X,//,15X,49H3.  OPTIMIZED FILTER CLOSED LOOP SYSTEM
       5 RESPONSE.,/,20X,37HXDOT =   F&*X +   G&*UC,        Z =   H&*X,/,20X,37
       6HXHDOT =   F&*XH + G&U +  K&* Z - H*XH&,//,15X,61H4.  OPTIMIZED FIL
       7TER + REGULATOR CLOSED LOOP SYSTEM RESPONSE.,/,20X,50HXDOT =   F+G*
       8C&*X +   G&*UC,                   Z =   H&*X,/,20X,51HXHDOT =   F&*XH + G
```

```
      9&U +  K&* Z - H*XH&,   U =  C&*XH,//,10X,20HSELECT 1, 2, 3 OR 4.)
340   FORMAT (//,15X,14H THE K MATRIX ,//)
350   FORMAT (//,15X,32H THE AUGMENTED F MATRIX  (F+G*C),///)
360   FORMAT (//,15X,16HTHE (G*C) MATRIX,//)
370   FORMAT (//,15X,16HTHE (K*H) MATRIX,//)
380   FORMAT (//,15X,42HTHE COMBINED SYSTEM F MATRIX (2*NS X  2*NS),///)
390   FORMAT (//,15X,34HTHE AUGMENTED G MATRIX (2*NS X NC),//)
400   FORMAT (//,10X,33HAT WHAT TIME DO YOU WANT TO START,/,10X,31HTHE T
     1IME RESPONSE CALCULATIONS?,//,10X,43HINPUT START TIME IN SECONDS.
     2(NORMALLY 0.0))
410   FORMAT (//,10X,32HAT WHAT TIME DO YOU WANT TO STOP,/,10X,31HTHE TI
     1ME RESPONSE CALCULATIONS?,//,10X,29H  INPUT STOP TIME IN SECONDS.)
420   FORMAT (//,10X,47HTHIS PROGRAM DIVIDES THE TIME INTERVAL YOU HAVE,
     1/,10X,49HJUST SPECIFIED INTO UP TO 500 SMALL INTERVALS FOR,/,10X,4
     28HTHE INTEGRATION AND PLOTTING ROUTINES.  IN ORDER,/,10X,50HTO SAV
     3E COMPUTER TIME, THE NUMBER OF POINTS CAN BE,/,10X,48HCAN BE REDUC
     4ED WITH SOME LOSS IN CURVE FIDELITY.,//,10X,41HHOW MANY POINTS DO
     5YOU WANT TO CALCULATE?)
430   FORMAT (/,10X,51HDOES THE SYSTEM UTILIZE A DRIVING FUNCTION (CONTR
     1OL,8H INPUT)?,//,25X,13H(Y)ES OR (N)O)
440   FORMAT (//,10X,46HTWO TYPES OF FUNCTIONS CAN BE USED AS DRIVERS.,/
     1,10X,19H      1. STEP INPUT,/,10X,19H      2. RAMP INPUT,//10X,29H
     2ENTER YOUR SELECTION, 1 OR 2.,28H FOR DRIVING FUNCTION NUMBER,I2)
450   FORMAT (/,10X,36HYOUR ANSWER MUST BE BETWEEN 1 AND 2.)
460   FORMAT (/,10X,39HAT WHAT TIME DO YOU DESIRE INPUT NUMBER,I2,10H TO
     1 START?,//,10X,34H  INPUT THE START TIME IN SECONDS.)
470   FORMAT (/,10X,39HAT WHAT TIME DO YOU DESIRE INPUT NUMBER,I2,9H TO
     1STOP?,//,10X,33H  INPUT THE STOP TIME IN SECONDS.)
480   FORMAT (/,10X,28HWHAT IS THE MAXIMUM VALUE OF,/,10X,23HDRIVING FUN
     1CTION NUMBER,I2,2H ?)
490   FORMAT (//,10X,49HDOES THE SYSTEM START WITH ALL INITIAL CONDITION
     1S,8H = 0.0 ?,//,25X,14H(Y)ES OR (N)O?)
500   FORMAT (/,10X,36HWHAT IS THE INITIAL CONDITION FOR X(,I2,3H) ?)
510   FORMAT (/,10X,39HWHAT IS THE INITIAL CONDITION FOR XHAT(,I2,3H) ?)
520   FORMAT (/,10X,32HTHIS IS YOUR LAST OPPORTUNITY TO,/,10X,36HMAKE CH
     1ANGES IN THE FOLLOWING AREAS.,//,12X,41H1.  SELECT ANOTHER TYPE OF
     2 SYSTEM TO PLOT,/,16X,42H(OPEN, CLOSED, FILTER OR FILTER/REGULATOR
     3),//,12X,24H2.  START AND STOP TIMES,//,12X,21H3.  DRIVING FUNCTIO
     4NS,//,12X,22H4.  INITIAL CONDITIONS,//,12X,12H5.  CONTINUE,//,10X,
     532HSELECT A NUMBER BETWEEN 1 AND 5.)
530   FORMAT (/,36HYOUR ANSWER MUST BE BETWEEN 1 AND 5.)
540   FORMAT (/,10X,41HTHE FOLLOWING INFORMATON IS PROVIDED ONLY,/,10X,4
     16HFOR AN INDICATION OF PROPER PROGRAM OPERATION.,//,5X,56HALL CONT
     2ROLS, STATES AND STATE ESTIMATES CAN BE PLOTTED.,//,5X,4HTIME,11X,
     34HU(1),10X,4HX(1),10X,4HX(2),10X,4HX(3),/)
550   FORMAT (/80HIER = 66 INDICATES THAT STEP SIZE WAS SUCCESSFULLY RED
     1UCED TO INCREASE ACCURACY.,/,18HNO ACTION REQUIRED)
560   FORMAT (/51HIER = 132 OR 133 INDICATES THAT CONVERGENCE WAS NOT,35
```

```
     1H ACHIEVED AFTER REDUCING STEP SIZE.,//,21HCONVERGENCE TOLERANCE,3
     26H IS BEING REDUCED BY A FACTOR OF TEN,/,22HFOR ANOTHER ATTEMPT AT
     3,13H CONVERGENCE.)
570   FORMAT (5E14.7)
580   FORMAT (1X,F8.2,5X,4E14.7)
590   FORMAT (5I5)
600   FORMAT (I1,3X,I1)
      END
C=====================================================================
C    SUBROUTINE FCN - USED BY IMSL SUBROUTINE DGEAR TO EVALUATE THE   =
C    SYSTEM UNDER INVESTIGATION.                                      =
C=====================================================================
      SUBROUTINE FCN (NS,T,X,XDOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(32),XDOT(32),F(32,32),G(32,10),U(10)
      COMMON F,G,U,NC
      CALL DRIVER (T,U,NC)
C*********************************************************************
C     SPECIAL DRIVER FOR XV-29A MATRICES
C*********************************************************************
      IF (NS.LT.77) GO TO 15
      IF (T.LT.1.0) U(1)=0.0D0
      IF (T.GE.1.0) U(1)=.1745D-01
      IF (T.GE.2.0) U(1)=-.1745D-01
      IF (T.GE.3.0) U(1)=0.0D0
      DO 10 I=1,NC
      U(I)=U(1)
10    CONTINUE
C*********************************************************************
15    CONTINUE
      DO 40 J=1,NS
      XDOT(J)=0.0D0
      DO 20 I=1,NC
      XDOT(J)=XDOT(J)+G(J,I)*U(I)
20    CONTINUE
      DO 30 K=1,NS
      XDOT(J)=XDOT(J)+F(J,K)*X(K)
30    CONTINUE
40    CONTINUE
      RETURN
      END
C=====================================================================
C    SUBROUTINE FCNJ - USED BY IMSL SUBROUTINE DGEAR TO EVALUATE THE  =
C    SYSTEM UNDER INVESTIGATION. (JUST A DUMMY SUBROUTINE.)           =
C=====================================================================
      SUBROUTINE FCNJ (NS,T,X,PD)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(32),PD(32,32)
```

```
      RETURN
      END
C=======================================================================
C   SUBROUTINE DRIVER - FORMS THE SPECIFIED DRIVING FUNCTION AND       =
C   RETURNS THE RESULT TO THE MAIN PROGRAM.                            =
C=======================================================================
      SUBROUTINE DRIVER (T,U,NC)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION U(10),DREND(10),DRBEG(10),UMAX(10),ITYPE(10)
      COMMON /A/ ITYPE,DREND,DRBEG,UMAX
      DO 20 I=1,NC
      IF (ITYPE(I).EQ.2) GO TO 10
      IF (T.LT.DRBEG(I)) U(I)=0.0
      IF (T.GE.DRBEG(I)) U(I)=UMAX(I)
      IF (T.GT.DREND(I)) U(I)=0.0
      GO TO 20
10    IF ((T.GE.DRBEG(I)).AND.(T.LE.DREND(I))) U(I)=(T-DRBEG
     1(I))*UMAX(I)/(DREND(I)-DRBEG(I))
      IF ((T.LT.DRBEG(I)).OR.(T.GT.DREND(I))) U(I)=0.0
20    CONTINUE
      RETURN
      END
C=======================================================================
C  SUBROUTINE RDMAT -- READS THE F, G, H, C AND K MATRICES             =
C  FROM THE DATA FILE OPTMAT DATA ON FILEDEF 9.                        =
C=======================================================================
      SUBROUTINE RDMAT (BA,G,HO,FBGC,FBGE,NS,NC,NC,NG)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BA(32,32),G(32,10),HO(32,32),GAM(32,32),FBGC(32,32),FBGE
     1(32,32)
      READ (9,10) ((BA(I,J),J=1,NS),I=1,NS)
      READ (9,10) ((G(I,J),J=1,NC),I=1,NS)
      READ (9,10) ((HO(I,J),J=1,NS),I=1,NC)
      READ (9,10) ((GAM(I,J),J=1,NG),I=1,NS)
      READ (9,10) ((FBGC(I,J),J=1,NS),I=1,NC)
      READ (9,10) ((FBGE(I,J),J=1,NO),I=1,NS)
      RETURN
C-----------------------------------------------------------------------
10    FORMAT (4(D20.13))
      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,O-Z)
      DIMENSION PRTT(32,32)
C-----------------------------------------------------------------------
```

```
      IF (NCCL.EQ.0) NCOL=1
      IF (NCCL.EQ.1) WRITE (5,10) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.2) WRITE (5,20) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.3) WRITE (5,30) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EC.4) WRITE (5,40) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.5) WRITE (5,50) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.6) WRITE (5,60) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.7) WRITE (5,70) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EC.8) WRITE (5,80) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.9) WRITE (5,90) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.10) WRITE (5,100) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.11) WRITE (5,110) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.12) WRITE (5,120) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.13) WRITE (5,130) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EQ.14) WRITE (5,140) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EC.15) WRITE (5,150) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCCL.EC.16) WRITE (5,160) ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      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,/,F12.5,//)
80    FORMAT (6F12.5,/,2F12.5,//)
90    FORMAT (6F12.5,/,3F12.5,//)
100   FORMAT (6F12.5,/,4F12.5,//)
110   FORMAT (6F12.5,/,5F12.5,//)
120   FORMAT (6F12.5,/,6F12.5,//)
130   FORMAT (6F12.5,/,6F12.5,/,F12.5,//)
140   FORMAT (6F12.5,/,6F12.5,/,2F12.5,//)
150   FORMAT (6F12.5,/,6F12.5,/,3F12.5,//)
160   FORMAT (6F12.5,/,6F12.5,/,4F12.5,//)
      END
C=======================================================================
      SUBROUTINE MAMULT (A,NAROWS,NACOLS,B,NBCOLS,C)
C     MULTIPLIES TWO MATRICES          C& =  A*B&                        =
C=======================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(32,32),B(32,32),C(32,32)
      DO 20 I=1,NAROWS
      DO 20 J=1,NBCOLS
      SUM=0.0D0
      DO 10 K=1,NACOLS
      SUM=SUM+A(I,K)*B(K,J)
10    CONTINUE
```

```
      C(I,J)=SUM
20    CONTINUE
      RETURN
      END
C==========================================================================
C  SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY               =
C  INTO A FORTRAN PROGRAM.  IF THE USER INADVERTENLY ENTERS AN IMPROPER=
C  DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.        =
C==========================================================================
      SUBROUTINE RDINT (IANS)
      INTEGER COUNT
C--------------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 50
20    CONTINUE
      READ (5,*,END=40,ERR=40) IANS
      IF (IANS) 40,40,30
30    CONTINUE
      RETURN
40    REWIND 5
      WRITE (5,70)
      GO TO 10
50    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 INADVERTENLY     =
C  ENTERS  A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY=
C==========================================================================
      SUBROUTINE RDCHAR (IANS)
      INTEGER COUNT
C--------------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
```

```
20    CONTINUE
      REWIND 5
      READ (5,70,END=30,ERR=30) IANS
      RETURN
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 RDREAL -- INTERACTIVELY READS A REAL NUMBER REPLY        =
C  INTO A FORTRAN PROGRAM.  IF THE USER INADVERTENLY 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-----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      READ (5,*,END=30,ERR=30) ANSR
      RETURN
30    REWIND 5
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
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 RDCHST -- INTERACTIVELY READS A CHARACTER STRING REPLY   =
C  UP TO 40 CHARACTERS LONG AND FORMATS THE CHARACTER STRING FOR USE   =
C  BY A DISSPLA PRINT ROUTINE.                                         =
C=======================================================================
```

```
      SUBROUTINE RDCHST (CHST)
      INTEGER CHST(11),I
      DATA IBL/'    '/,IDOL/'$   '/
C-----------------------------------------------------------------------
      CALL GETCHS (CHST)
      CHST(11) = IBL
      DO 10 I =1,11
         IF(CHST(I).NE.IBL) GO TO 10
         CHST(I) = IDOL
         GO TO 20
10    CONTINUE
20    RETURN
C-----------------------------------------------------------------------
      END
C=======================================================================
C  SUBROUTINE GETCHS -- INTERACTIVELY READS A CHARACTER STRING REPLY   =
C  UP TO 40 CHARACTERS LONG.  IF THE USER INADVERTENTLY ENTERS A NULL  =
C  STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY               =
C=======================================================================
      SUBROUTINE GETCHS (CHST)
      INTEGER COUNT,CHST(20),I
C-----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      REWIND 5
      READ (5,70,END=30,ERR=30) (CHST(I),I = 1,10)
      RETURN
30    REWIND 5
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
C-----------------------------------------------------------------------
50    FORMAT (1X,'WARNING:  NULL STRINGS ARE NOT ALLOWED, THE PROGRAM',
     1/,'WILL TERMINATE IF ANOTHER NULL STRING IS ENTERED.')
60    FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED )
70    FORMAT (10A4)
      END
C=======================================================================
C  SUBROUTINE NEWSCR -- CLEARS THE SCREEN WITHOUT ERASING THE          =
C  PREVIOUS SCREEN'S INFORMATION.                                      =
C=======================================================================
      SUBROUTINE NEWSCR
```

```
      WRITE (5,10)
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C----------------------------------------------------------------
10    FORMAT(////////////////////)
      END
```

```
C*********************************************************************
C
C                              OPTPLOT
C
C              DISSPLA PLOTTING ROUTINE TO BE USED WITH
C                       OPTSYSX AND OPTCALC
C                      UNDER THE OPTSYS EXEC
C
C                                BY
C                            H. A. DIEL
C                           25 JULY 1984
C
C*********************************************************************
C=====================================================================
C=====================================================================
C
C  MAIN PROGRAM - CONTROLS DISSPLA PLOTTING SOFTWARE PACKAGE
C
C=====================================================================
      IMPLICIT REAL*4 (A-H,O-Z)
      REAL*8 DCLBLP
      INTEGER TITLE1,TITLE2,TITLE3,TITLE4,HEAD1,HEAD2,HEAD3,CHST
      DIMENSION LGND1(501),TITLE1(11),TITLE2(11),TITLE3(11),TITLE4(11),H
     1EAD1(11),HEAD2(11),HEAD3(11),CHST(11),DATA(501,83),U(501,10),FBGC(
     210,83),TIME(501),C1(501),C2(501),C3(501),C4(501),NAMFIL(2)
      DATA ICMPRS/0/,IHEAD/4H    $/,ICLDGR/0/,IYES/1HY/,INO/1HN/
C---------------------------------------------------------------------
C  SUPPRESS INDIVIDUAL UNDERFLOW, OVERFLOW, DIVIDE CHECK, AND DECIMAL  =
C  CONVERT ERROR MESSAGES; PROVIDE SUMMARY OF ERRORS ONLY.             =
C---------------------------------------------------------------------
      CALL ERRSET (207,256,-1,1,1,209)
      CALL ERRSET (215,256,-1,1)
C---------------------------------------------------------------------
C=====================================================================
C     READ IN "OPTPLOT DATA A1"
C=====================================================================
      READ (8,1390) NS,NC,NPTS,IEST
      N=NS
      IF (IEST.EQ.3) N=2*NS
      READ (8,1380) ((FBGC(I,J),J=1,NS),I=1,NC)
      DO 10 I=1,NPTS
      READ (8,1380) TIME(I),(U(I,J),J=1,NC),(DATA(I,K),K=1,N)
10    CONTINUE
      NPTSDA=NPTS
C=====================================================================
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,940)
20    WRITE (5,950)
```

```
      CALL RDINT (IANS)
      IF (IANS.EQ.1) GO TO 50
      IF (IANS.EQ.2) GO TO 210
      GO TO 20
30    IF (IOLDGR.EQ.0) GO TO 50
C==========================================================================
C     READ IN DATA AGAIN IF OLD GRAPH DATA WAS USED
C==========================================================================
      IOLDGR=0
      REWIND 8
      READ (8,1390) NS,NC,NPTS,IEST
      N = NS
      IF (IEST.EQ.3) N=2*NS
      READ (8,1380) ((FBGC(I,J),J=1,NS),I=1,NC)
      DO 40 I=1,NPTS
      READ (8,1380) TIME(I),(U(I,J),J=1,NC),(DATA(I,K),K=1,N)
40    CONTINUE
      NPTSDA=NPTS
C==========================================================================
C        SELECT NUMBER OF CURVES TO PLOT ON GRAPH
C==========================================================================
50    WRITE (5,960)
      CALL RDINT (NCURVS)
      IF ((NCURVS.GE.1).AND.(NCURVS.LE.4)) GO TO 60
      WRITE (5,970) NCURVS
      GO TO 50
60    CALL FRTCMS ('CLRSCRN ')
      XPAGE=8.5
      YPAGE=6.0
      DELTAX=0.0
      DELTAY=0.0
      SCALEH=1.0
      N=1
      CALL SELCRV (N,C1,C1MIN,C1MAX,TITLE1,DATA,U,FBGC,NS,NC,NPTS,IEST)
      IF (NCURVS.EQ.1) GO TO 70
      N=2
      CALL SELCRV (N,C2,C2MIN,C2MAX,TITLE2,DATA,U,FBGC,NS,NC,NPTS,IEST)
      IF (NCURVS.EQ.2) GO TO 70
      N=3
      CALL SELCRV (N,C3,C3MIN,C3MAX,TITLE3,DATA,U,FBGC,NS,NC,NPTS,IEST)
      IF (NCURVS.EQ.3) GO TO 70
      N=4
      CALL SELCRV (N,C4,C4MIN,C4MAX,TITLE4,DATA,U,FBGC,NS,NC,NPTS,IEST)
70    CALL FRTCMS ('CLRSCRN ')
C==========================================================================
C        SELECT NUMBER OF HEADINGS FOR GRAPH
C==========================================================================
      HEAD2(1)=IHEAD
```

```
      HEAD3(1)=IHEAD
80    WRITE (5,980)
      CALL RCREAL (DOUBLP)
      NHEAD=IDINT(DOUBLP)
      IF ((NHEAD.GE.0).AND.(NHEAD.LE.3)) GO TO 90
      WRITE (5,990) NHEAD
      GO TO 80
90    IF (NHEAD.EQ.0) GO TO 100
      N=1
      CALL HEADS (HEAD1,N)
      IF (NHEAD.EQ.1) GO TO 100
      N=2
      CALL HEADS (HEAD2,N)
      IF (NHEAD.EQ.2) GO TO 100
      N=3
      CALL HEADS (HEAD3,N)
C=======================================================================
C     PLOT CURVES
C=======================================================================
100   CALL FRTCMS ('CLRSCRN ')
110   CALL TEK618
      CALL HWROT ('AUTO')
      CALL HWSCAL ('SCREEN')
      GO TO 120
115   IF (ICMPRS.EQ.1) GO TO 120
      CALL COMPRS
      ICMPRS=1
120   CALL PAGE (XPAGE,YPAGE)
      CALL NOBRDR
      CALL TRIPLX
      XPOS=.2+.3*(XPAGE+YPAGE)/19.5
      XORIGN=.1+XPOS*FLOAT(NCURVS)
      YORIGN=.3+.35*(XPAGE+YPAGE)/19.5
      CALL PHYSOR (XORIGN,YORIGN)
      XAXIS=XPAGE-XORIGN-.3
      YAXIS=YPAGE-YORIGN-.3-(XPAGE+YPAGE)/19.5
      IF (NHEAD.EQ.0) YAXIS=YPAGE-YORIGN-.3
      CALL AREA2D (XAXIS,YAXIS)
      CALL FRAME
      HITE=(XPAGE+YPAGE)*.006*SCALEH
      IF (HITE.LT.0.01) HITE=0.01
      CALL HEIGHT (HITE)
      CALL BASALF ('STANDARD')
      CALL MIXALF ('L/CGREEK')
      CALL XNAME ('TIME - SEC$',100)
      CALL YNAME (TITLE1,100)
      TSTART=TIME(1)
      TSTOP=TIME(NPTS)
```

```
      CALL AXSPLT (TSTART,TSTOP,XAXIS,TBEGIN,TSTEP,AXIS)
      TEND=TBEGIN+TSTEP*AXIS
      CALL GRAF (TBEGIN,TSTEP,TEND,C1MIN,'SCALE',C1MAX)
      CALL RLVEC (TSTART,0.0,TEND,0.0,0000)
      IF (NPTS.LE.200) CALL RASPLN (5.0)
      CALL LEGLIN
      MARKRS=NPTS/10
      CALL CURVE (TIME,C1,NPTS,MARKRS)
      CALL LINES (TITLE1,LGND1,1)
      IF (NCURVS.EQ.1) GO TO 130
      CALL YGRAXS (C2MIN,'SCALE',C2MAX,YAXIS,TITLE2,100,-XPOS,0.0)
      CALL RLVEC (TSTART,0.0,TEND,0.0,0000)
      CALL DASH
      CALL LEGLIN
      CALL CURVE (TIME,C2,NPTS,MARKRS)
      CALL LINES (TITLE2,LGND1,2)
      CALL RESET ('DASH')
      IF (NCURVS.EQ.2) GO TO 130
      CALL YGRAXS (C3MIN,'SCALE',C3MAX,YAXIS,TITLE3,100,-2.0*XPOS,0.0)
      CALL RLVEC (TSTART,0.0,TEND,0.0,0000)
      CALL CHNDOT
      CALL LEGLIN
      CALL CURVE (TIME,C3,NPTS,MARKRS)
      CALL LINES (TITLE3,LGND1,3)
      CALL RESET ('CHNDOT')
      IF (NCURVS.EQ.3) GO TO 130
      CALL YGRAXS (C4MIN,'SCALE',C4MAX,YAXIS,TITLE4,100,-3.0*XPOS,0.0)
      CALL RLVEC (TSTART,0.0,TEND,0.0,0000)
      CALL CHNDSH
      CALL LEGLIN
      CALL CURVE (TIME,C4,NPTS,MARKRS)
      CALL LINES (TITLE4,LGND1,4)
      CALL RESET ('CHNDSH')
C=====================================================================
C          PRINT LEGEND
C=====================================================================
130   CALL LINESP (1.8)
      X1=XLEGND(LGND1,NCURVS)
      Y1=YLEGND(LGND1,NCURVS)
      XLED=XAXIS-0.2-X1+DELTAX
      YLED=0.2+DELTAY
      CALL LEGEND (LGND1,NCURVS,XLED,YLED)
C=====================================================================
C          PRINT GRID LEAVING BOX AROUND LEGEND
C=====================================================================
      CALL BLREC (XLED-0.1,YLED-0.1,X1+.2,Y1+.2,2.)
      CALL DOT
      CALL GRID (1,1)
```

```
      CALL RESET ('DOT')
      CALL ENDGR (0)
      IF (NHEAD.EQ.0) GO TO 140
      CALL HEADIN (HEAD1,100,1.5,NHEAD)
      IF (NHEAD.EQ.1) GO TO 140
      CALL HEADIN (HEAD2,100,1.5,NHEAD)
      IF (NHEAD.EQ.2) GO TO 140
      CALL HEADIN (HEAD3,100,1.5,NHEAD)
140   CONTINUE
      CALL ENDPL (0)
      IF (ICMPRS.EQ.1) GO TO 880
C=======================================================================
C    MAIN OPTIONS MENU
C=======================================================================
150   CALL FRTCMS ('CLRSCRN ')
160   WRITE (5,1000)
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.5)) GO TO 170
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,1010) IANS
      GO TO 160
170   CALL FRTCMS ('CLRSCRN ')
      GO TO (180,210,270,840,850), IANS
C=======================================================================
C     BEGIN A NEW GRAPH
C=======================================================================
180   WRITE (5,1040)
      CALL RDCHAR (IANS)
      IF (IANS.EQ.IYES) GO TO 190
      IF (IANS.EQ.INO) GO TO 200
      WRITE (5,1050)
      GO TO 180
190   CALL FILECV (NPTS,NCURVS,NHEAD,HEAD1,HEAD2,HEAD3,TITLE1
     1,TITLE2,TITLE3,TITLE4,XPAGE,YPAGE,DELTAX,DELTAY,SCALEH,
     2C1MIN,C1MAX,C2MIN,C2MAX,C3MIN,C3MAX,C4MIN,C4MAX,
     3TIME,C1,C2,C3,C4)
200   GO TO 30
C=======================================================================
C     READ CURVE DATA FROM FILE AND PLOT CURVE
C=======================================================================
210   WRITE (5,1040)
      CALL RDCHAR (IANS)
      IF (IANS.EQ.IYES) GO TO 220
      IF (IANS.EQ.INO) GO TO 230
      WRITE (5,1050)
      GO TO 210
220   CALL FILECV (NPTS,NCURVS,NHEAD,HEAD1,HEAD2,HEAD3,TITLE1
     1,TITLE2,TITLE3,TITLE4,XPAGE,YPAGE,DELTAX,DELTAY,SCALEH,
```

```
     2C1MIN,C1MAX,C2MIN,C2MAX,C3MIN,C3MAX,C4MIN,C4MAX,
     3TIME,C1,C2,C3,C4)
230   WRITE (5,1060)
      READ (5,1410,END=265,ERR=265) (NAMFIL)
      CALL FRTCMS ('FILEDEF ','04      ','DISK    ',NAMFIL,'DATA    ',
     1'A1      ')
      WRITE (5,1070) NAMFIL
      REWIND 4
      READ (4,1390,END=260,ERR=260) NPTS,NCURVS,NHEAD
      READ (4,1400,END=260,ERR=260) (HEAD1(I),I=1,11)
      IF (NHEAD.EQ.1) GO TO 240
      READ (4,1400,END=260,ERR=260) (HEAD2(I),I=1,11)
      IF (NHEAD.EQ.2) GO TO 240
      READ (4,1400,END=260,ERR=260) (HEAD3(I),I=1,11)
240   READ (4,1400,END=260,ERR=260) (TITLE1(I),I=1,11)
      IF (NCURVS.EQ.1) GO TO 250
      READ (4,1400,END=260,ERR=260) (TITLE2(I),I=1,11)
      IF (NCURVS.EQ.2) GO TO 250
      READ (4,1400,END=260,ERR=260) (TITLE3(I),I=1,11)
      IF (NCURVS.EQ.3) GO TO 250
      READ (4,1400,END=260,ERR=260) (TITLE4(I),I=1,11)
250   READ (4,1380,END=260,ERR=260) XPAGE,YPAGE,DELTAX,DELTAY,SCALEH
      READ (4,1380,END=260,ERR=260) C1MIN,C1MAX,C2MIN,C2MAX
      READ (4,1380,END=260,ERR=260) C3MIN,C3MAX,C4MIN,C4MAX
      READ (4,1380,END=260,ERR=260) (TIME(I),C1(I),C2(I),C3(I),C4(I),I=1
     1,NPTS)
      IOLDGR=1
      GO TO 840
260   WRITE (5,1080) NAMFIL
      WRITE (5,1090)
      CALL RDCHAR (IANS)
      GO TO 150
265   WRITE (5,1375)
      WRITE (5,1090)
      CALL RDCHAR (IANS)
      GO TO 150
C=====================================================================
C     MAKE CORRECTIONS TO EXISTING GRAPH
C=====================================================================
270   CALL FRTCMS ('CLRSCRN ')
280   WRITE (5,1020)
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.10)) GO TO 290
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,1030) IANS
      GO TO 280
290   CALL FRTCMS ('CLRSCRN ')
      GO TO (300,380,440,560,750,730,800,820,830,150), IANS
```

```
C=====================================================================
C         CHANGE CURVE VARIABLE
C=====================================================================
300    CALL FRTCMS ('CLRCSRN ')
       IF (IOLDGR.EQ.0) GO TO 310
       WRITE (5,1100)
       GO TO 280
310    ICURVS=NCURVS
       IF (NCURVS.LT.4) ICURVS=NCURVS+1
       WRITE (5,1110) ICURVS
       CALL RDINT (IANS)
       IF ((IANS.GE.1).AND.(IANS.LE.ICURVS)) GO TO 320
       WRITE (5,1150) ICURVS
       GO TO 310
320    IF (IANS.EQ.(NCURVS+1)) NCURVS=IANS
       GO TO (330,340,350,360), IANS
330    N=1
       CALL SELCRV (N,C1,C1MIN,C1MAX,TITLE1,DATA,U,FBGC,NS,NC,NPTS,IEST)
       GO TO 370
340    N=2
       CALL SELCRV (N,C2,C2MIN,C2MAX,TITLE2,DATA,U,FBGC,NS,NC,NPTS,IEST)
       GO TO 370
350    N=3
       CALL SELCRV (N,C3,C3MIN,C3MAX,TITLE3,DATA,U,FBGC,NS,NC,NPTS,IEST)
       GO TO 370
360    N=4
       CALL SELCRV (N,C4,C4MIN,C4MAX,TITLE4,DATA,U,FBGC,NS,NC,NPTS,IEST)
370    GO TO 270
C=====================================================================
C         DELETE CURVE
C=====================================================================
380    WRITE (5,1120) NCURVS
       CALL RDINT (IANS)
       IF ((IANS.GE.1).AND.(IANS.LE.5)) GO TO (390,400,410,420,430), IANS
       WRITE (5,1130)
       GO TO 380
390    CALL CRVEXC (C1,C1MIN,C1MAX,TITLE1,C2,C2MIN,C2MAX,TITLE2)
400    CALL CRVEXC (C2,C2MIN,C2MAX,TITLE2,C3,C3MIN,C3MAX,TITLE3)
410    CALL CRVEXC (C3,C3MIN,C3MAX,TITLE3,C4,C4MIN,C4MAX,TITLE4)
420    NCURVS=NCURVS-1
430    GO TO 270
C=====================================================================
C         EDIT CURVE TITLE
C=====================================================================
440    WRITE (5,1140)
       WRITE (5,1150) NCURVS
       CALL RDINT (ICRV)
       IF ((ICRV.GE.1).AND.(ICRV.LE.NCURVS)) GO TO 450
```

```
      WRITE (5,1150) NCURVS
      GO TO 440
450   WRITE (5,1160)
      GO TO (460,470,480,490), ICRV
460   CALL CLRCHR (TITLE1)
      GO TO 500
470   CALL CLRCHR (TITLE2)
      GO TO 500
480   CALL CLRCHR (TITLE3)
      GO TO 500
490   CALL CLRCHR (TITLE4)
500   CALL RDCHST (CHST)
      DO 550 J=1,11
      GO TO (510,520,530,540), ICRV
510   TITLE1(J)=CHST(J)
      GO TO 550
520   TITLE2(J)=CHST(J)
      GO TO 550
530   TITLE3(J)=CHST(J)
      GO TO 550
540   TITLE4(J)=CHST(J)
550   CONTINUE
      GO TO 270
C=====================================================================
C          EDIT GRAPH HEADING
C=====================================================================
560   WRITE (5,1170)
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.4)) GO TO (570,570,660,270), IANS
      WRITE (5,1180)
      GO TO 560
570   WRITE (5,1190)
      CALL RDINT (IHDG)
      IF ((IHDG.GE.1).AND.(IHDG.LE.3)) GO TO 580
      WRITE (5,1200)
      GO TO 570
580   IF (NHEAD.LT.IHDG) NHEAD=IHDG
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,1210)
      GO TO (590,600,610), IHDG
590   CALL CLRCHR (HEAD1)
      GO TO 620
600   CALL CLRCHR (HEAD2)
      GO TO 620
610   CALL CLRCHR (HEAD3)
620   CALL RDCHST (CHST)
      DO 650 J=1,11
      IF (IHDG.NE.1) GO TO 630
```

```
      HEAD1(J)=CHST(J)
630   IF (IHDG.NE.2) GO TO 640
      HEAD2(J)=CHST(J)
640   IF (IHDG.NE.3) GO TO 650
      HEAD3(J)=CHST(J)
650   CONTINUE
      GO TO 270
660   CALL FRTCMS ('CLRSCRN ')
670   WRITE (5,1220)
      WRITE (5,1230) NHEAD
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.NHEAD)) GO TO (680,700,720), IANS
      WRITE (5,1230) NHEAD
      GO TO 670
680   DO 690 I=1,11
      HEAD1(I)=HEAD2(I)
690   CONTINUE
700   DO 710 I=1,11
      HEAD2(I)=HEAD3(I)
710   CONTINUE
720   NHEAD=NHEAD-1
      GO TO 270
C=====================================================================
C     CHANGE TIME SCALE
C=====================================================================
730   WRITE (5,1240) NPTS,TSTOP,TIME(NPTSDA)
      CALL RDREAL (DOUBLP)
      ANS=SNGL(DOUBLP)
      IF ((ANS.GT.TSTART).AND.(ANS.LE.TIME(NPTSDA))) GO TO 740
      WRITE (5,1250) ANS,TSTART,TIME(NPTSDA)
      GO TO 730
740   NPTS=IFIX(FLOAT(NPTSDA)*ANS/TIME(NPTSDA))
      TSTOP=ANS
      IF (NPTS.GT.NPTSDA) NPTS=NPTSDA
      GO TO 270
C=====================================================================
C     CHANGE CURVE Y DIRECTION SCALE
C=====================================================================
750   WRITE (5,1260)
      CALL RDINT (IANS)
      WRITE (5,1270)
      CALL RDREAL (DOUBLP)
      YMIN=SNGL(DOUBLP)
      WRITE (5,1280)
      CALL RDREAL (DOUBLP)
      YMAX=SNGL(DOUBLP)
      IF (IANS.NE.1) GO TO 760
      C1MIN=YMIN
```

```
      C1MAX=YMAX
760   IF (IANS.NE.2) GO TO 770
      C2MIN=YMIN
      C2MAX=YMAX
770   IF (IANS.NE.3) GO TO 780
      C3MIN=YMIN
      C3MAX=YMAX
780   IF (IANS.NE.4) GO TO 790
      C2MIN=YMIN
      C2MAX=YMAX
790   GO TO 270
C=====================================================================
C         CHANGE PLOT SIZE
C=====================================================================
800   WRITE (5,1290)
      WRITE (5,1300)
      CALL CUREAL (XPAGE)
      CALL RDREAL (DOUBLP)
      ANS=SNGL(DOUBLP)
      IF ((ANS.LT.0.1).OR.(ANS.GT.21.0)) GO TO 800
      XPAGE=ANS
810   WRITE (5,1310)
      WRITE (5,1300)
      CALL CUREAL (YPAGE)
      CALL RDREAL (DOUBLP)
      ANS=SNGL(DOUBLP)
      IF ((ANS.LT.0.1).OR.(ANS.GT.21.0)) GO TO 810
      YPAGE=ANS
      GO TO 270
C=====================================================================
C         CHANGE THE LETTERING HEIGHT
C=====================================================================
820   WRITE (5,1320)
      CALL CUREAL (SCALEH)
      CALL RDREAL (DOUBLP)
      SCALEH=SNGL(DOUBLP)
      IF (SCALEH.GT.0.0) GO TO 270
      WRITE (5,1330)
      GO TO 820
C=====================================================================
C         MOVE LEGEND BOX
C=====================================================================
830   WRITE (5,1340)
      CALL RDREAL (DOUBLP)
      ANS=SNGL(DOUBLP)
      DELTAX=DELTAX+ANS
C     CALL FRTCMS ('CLRSCRN ')
      WRITE (5,1350)
```

```
      CALL RDREAL (DOUBLP)
      ANS=SNGL(DOUBLP)
      DELTAY=DELTAY+ANS
      GO TO 270
C=======================================================================
C         PLOT THE REVISED GRAPH
C=======================================================================
840   IF (ICMPRS.EQ.1) CALL DONEPL
      ICMPRS=0
      GO TO 110
C=======================================================================
C     QUIT OR MAKE METAFILE MENU
C=======================================================================
850   WRITE (5,1040)
      CALL RDCHAR (IANS)
      IF (IANS.EQ.IYES) GO TO 860
      IF (IANS.EQ.INO) GO TO 870
      WRITE (5,1050)
      GO TO 850
860   CALL FILECV (NPTS,NCURVS,NHEAD,HEAD1,HEAD2,HEAD3,TITLE1
     1,TITLE2,TITLE3,TITLE4,XPAGE,YPAGE,DELTAX,DELTAY,SCALEH,
     2C1MIN,C1MAX,C2MIN,C2MAX,C3MIN,C3MAX,C4MIN,C4MAX,
     3TIME,C1,C2,C3,C4)
870   CALL DONEPL
880   CALL FRTCMS ('CLRSCRN ')
      WRITE (5,1360)
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.2)) GO TO (890,930), IANS
      WRITE (5,1370)
      GO TO 880
C=======================================================================
C     MAKE METAFILE OF ANY PREVIOUSLY FILED CURVES
C=======================================================================
890   WRITE (5,1060)
      READ (5,1410,END=265,ERR=265) (NAMFIL)
      CALL FRTCMS ('FILEDEF ','04      ','DISK    ',NAMFIL,'DATA    ',
     1'A1      ')
      WRITE (5,1070) NAMFIL
      REWIND 4
      READ (4,1390,END=920,ERR=920) NPTS,NCURVS,NHEAD
      READ (4,1400,END=920,ERR=920) (HEAD1(I),I=1,11)
      IF (NHEAD.EQ.1) GO TO 900
      READ (4,1400,END=920,ERR=920) (HEAD2(I),I=1,11)
      IF (NHEAD.EQ.2) GO TO 900
      READ (4,1400,END=920,ERR=920) (HEAD3(I),I=1,11)
900   READ (4,1400,END=920,ERR=920) (TITLE1(I),I=1,11)
      IF (NCURVS.EQ.1) GO TO 910
      READ (4,1400,END=920,ERR=920) (TITLE2(I),I=1,11)
```

```
      IF (NCURVS.EQ.2) GC TC 910
      READ (4,1400,END=920,ERR=920) (TITLE3(I),I=1,11)
      IF (NCURVS.EQ.3) GC TC 910
      READ (4,1400,END=920,ERR=920) (TITLE4(I),I=1,11)
910   READ (4,1380,END=920,ERR=920) XPAGE,YPAGE,DELTAX,DELTAY,SCALEH
      READ (4,1380,END=920,ERR=920) C1MIN,C1MAX,C2MIN,C2MAX
      READ (4,1380,END=920,ERR=920) C3MIN,C3MAX,C4MIN,C4MAX
      READ (4,1380,END=920,ERR=920) (TIME(I),C1(I),C2(I),C3(I),C4(I),I=1
     1,NPTS)
      GO TO 115
920   WRITE (5,1080) NAMFIL
      WRITE (5,1090)
      CALL RCCHAR (IANS)
      GO TO 880
930   IF (ICMPRS.EQ.1) CALL DONEPL
      STOP
C-----------------------------------------------------------------------
940   FORMAT (//,10X,34HTHIS PORTION OF THE PROGRAM PLOTS:,/,15X,13H- TH
     1E STATES,,/,15X,26H- EXTERNAL CONTROL INPUTS,,/,15X,26H- FEEDBACK
     2CONTROL INPUTS,,/,15X,21H- STATE ESTIMATES AND,/,15X,23H- RECONSTR
     3UCTION ERRORS,/,10X,39HFROM THE DATA THAT YOU JUST CALCULATED.,//,
     410X,46HTHE CAPABILITY IS ALSO AVAILABLE TO REVIEW ANY,/,10X,44HGRA
     5PHS THAT YOU HAD PREVIOUSLY SAVED AS DATA,/,10X,19HFILES ON YOUR D
     6ISK.,//,15X,29HCLEAR THE SCREEN TO CONTINUE.////////)
950   FORMAT (///,10X,36HTHE FOLLOWING OPTIONS ARE AVAILABLE:,//,15X,37H
     11. PLOT THE DATA YOU JUST CALCULATED.,/,15X,42H2. PLOT A CURVE THA
     2T YOU PREVIOUSLY SAVED.,//,10X,12HENTER 1 OR 2)
960   FORMAT (//,10X,46HYOU MAY PLOT UP TO 4 SYSTEM VARIABLES VS TIME.,1
     10X,/39HHOW MANY VARIABLES DO YOU WISH TO PLOT?)
970   FORMAT (/,10X,56HTHE NUMBER OF VARIABLES TO PLOT MUST BE BETWEEN 1
     1 AND 4.,/,15X,1H(,I3,31H) WAS THE NUMBER YOU REQUESTED.)
980   FORMAT (//,10X,29HYOU MAY USE UP TO 3 HEADINGS.,/,10X,46HHOW MANY
     1HEADINGS DO YOU DESIRE ON THIS GRAPH?,//,20X,12H0, 1, 2 OR 3)
990   FORMAT (/,10X,47HTHE NUMBER OF HEADINGS MUST BE BETWEEN 0 AND 3.,/
     1,10X,1H(,I3,31H) WAS THE NUMBER YOU REQUESTED.)
1000  FORMAT (/,10X,36HTHE FOLLOWING OPTIONS ARE AVAILABLE.,//,10X,60H 1
     1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.,/,10X,3
     29H 2. REPLOT PREVIOUSLY SAVED GRAPH DATA.,/,10X,27H 3. EDIT THE CU
     3RRENT GRAPH.,/,10X,37H 4. PLOT REVISED GRAPH ON THE TEK618.,/,10X,
     444H 5. QUIT AND/OR MAKE METAFILE OF THE CURVES.,/,14X,17HPREVIOUSL
     5Y SAVED.,//,32HSELECT A NUMBER BETWEEN 1 AND 5.)
1010  FORMAT (/,15HYOUR INPUT OF (,I3,38H) IS NOT WITHIN THE RANGE OF 1
     1TO 5.   ,/)
1020  FORMAT (/,20X,19HTHE GRAPH EDIT MENU,//,10X,56H 1. CHANGE VARIABLE
     1S OR ADD A CURVE ON THE CURRENT PLOT.,/,10X,35H 2. DELETE CURVE FR
     2OM CURRENT PLOT.,/,10X,24H 3. EDIT CURVE TITLE(S).,/,10X,25H 4. ED
     3IT PAGE HEADING(S).,/,10X,28H 5. CHANGE THE Y-AXIS SCALE.,/,10X,31
     4H 6. CHANGE THE TIME AXIS SCALE.,/,10X,45H 7. CHANGE PLOT SIZE. (D
```

```
      5EFAULT IS 8.5 X 6.0),/,10X,32H 8. CHANGE THE LETTERING HEIGHT.,/,1
      60X,34H 9. CHANGE POSITION OF THE LEGEND.,/,10X,21H10. EDITING COMP
      7LETE.,//,33HSELECT A NUMBER BETWEEN 1 AND 10.)
1030  FORMAT (/,15HYOUR INPUT OF (,I3,38H) IS NOT WITHIN THE RANGE OF 1
     1TO 10. ,/)
1040  FORMAT (/,10X,45HDO YOU WANT TO SAVE THE CURRENT GRAPH DATA TO,/,1
     10X,37HBE USED LATER TO GENERATE A METAFILE?,//,20X,6HY OR N,//,10X
     2,55HNOTE: A METAFILE IS REQUIRED FOR SMOOTH VERSATEC PLOTS.,/,10X,
     352HTHERE WILL BE AN OPPORTUNITY TO GENERATE A METAFILE ,/,10X,33HJ
     4UST BEFORE EXITING THIS PROGRAM.)
1050  FORMAT (/,5X,31HYOUR ANSWER MUST BE "Y" OR "N".)
1060  FORMAT (//,10X,40HWHAT FILE NAME IS THE DATA STORED UNDER?)
1070  FORMAT (/,41HTHE CURVE DATA IS BEING LOADED FROM FILE ,2A4,5H DATA
     1)
1080  FORMAT (/,10X,24HREAD ERROR ON FILE NAME ,2A4,/,10X,19H RECHECK FI
     1LE NAME.)
1090  FORMAT (//,20X,23H(ANY INPUT TO CONTINUE))
1100  FORMAT (/,5X,47HTHIS FUNCTION IS NOT AVAILABLE ON AN OLD GRAPH.)
1110  FORMAT (/,10X,38HWHICH CURVE DO YOU WANT TO ADD/CHANGE?,/,10X,32HY
     1OUR INPUT MUST BE BETWEEN 1 AND,I2,1H.)
1120  FORMAT (/,10X,I2,41H CURVES ARE PLOTTED ON THE CURRENT GRAPH,,/,10
     1X,34HWHICH CURVE DO YOU WANT TO DELETE?,//,15X,10H1. CURVE 1,/,15X
     2,10H2. CURVE 2,/,15X,10H3. CURVE 3,/,15X,10H4. CURVE 4,/,15X,22H5.
     3 RETURN TO EDIT MENU,//,10X,30HENTER A NUMBER BETWEEN 1 AND 5)
1130  FORMAT (//,35HYOUR ANSWER MUST BE BETWEEN 1 AND 5)
1140  FORMAT (//,10X,49HWHAT IS THE CURVE NUMBER OF THE TITLE YOU WISH T
     1O,8H REVISE?,/)
1150  FORMAT (/,10X,36HYOUR INPUT MUST BE BETWEEN (1) AND (,I1,2H).)
1160  FORMAT (//,10X,41HWHAT IS THE DESIRED LABEL FOR THIS CURVE?,/,10X,
     134HNOTE:  1. 40 CHARACTERS MAX LENGTH,/,17X,48H2. GREEK SYMBOLS WI
     2LL BE PRINTED FOR ANY LETTERS,/,19X,24HENCLOSED IN PARENTHESES.,/,
     320X,16HIE. (A) => ALPHA,/,24X,11H(B) => BETA,/,24X,10H(F) => PHI,/
     4,24X,12H(Q) => THETA)
1170  FORMAT (//,10X,31HYOU HAVE THE FOLLOWING OPTIONS:,//,15X,18H1.  AD
     1D A HEADING.,/,15X,21H2.  REVISE A HEADING.,/,15X,21H3.  DELETE A
     2HEADING.,/,15X,28H4.  RETURN TO THE EDIT MENU.,//,31HINPUT A NUMBE
     3R BETWEEN 1 AND 4.)
1180  FORMAT (/,15X,35HYOUR INPUT MUST BE BETWEEN 1 AND 4.)
1190  FORMAT (//,10X,43HWHICH HEADING DO YOU WISH TO REVISE OR ADD?,/)
1200  FORMAT (/,10X,40HYOUR INPUT MUST BE BETWEEN (1) AND (3). )
1210  FORMAT (//,10X,28HWHAT IS THE DESIRED HEADING?,/,10X,34HNOTE:  1.
     140 CHARACTERS MAX LENGTH,/,17X,48H2. GREEK SYMBOLS WILL BE PRINTED
     2 FOR ANY LETTERS,/,19X,24HENCLOSED IN PARENTHESES.,/,20X,16HIE. (A
     3) => ALPHA,/,24X,11H(B) => BETA,/,24X,10H(F) => PHI,/,24X,12H(Q) =
     4> THETA)
1220  FORMAT (//,10X,36HWHICH HEADING DO YOU WANT TO DELETE?)
1230  FORMAT (//,10X,33HYOUR INPUT MUST BE BETWEEN 1 AND ,I1,1H.)
1240  FORMAT (/,10X,21HTHE CURRENT PLOT HAS ,I3,23H POINTS PLOTTED WITH
```

```
     1AN/,10X,12HEND TIME OF ,F9.2,9H SECCNDS.,/,10X,24HDATA IS AVAILABL
     2E UP TO ,F9.2,9H SECONDS.,//,15X,34HWHAT IS THE DESIRED NEW END TI
     3ME.?)
1250  FORMAT (//,10X,15HYOUR ANSWER CF ,F9.2,26H IS NOT WITHIN THE ALLOW
     1ED/,10X,9HRANGE OF ,F9.2,4H TO ,F9.2,9H SECCNDS.)
1260  FORMAT (/10X,49HON WHICH CURVE DO YOU WANT TO CHANGE THE Y-SCALE?,
     1//17X,33HENTER CURVE NUMBER- 1, 2, 3, CR 4)
1270  FORMAT (/10X,42HWHAT IS THE NEW Y-MIN VALUE AT THE ORIGIN?)
1280  FORMAT (/10X,28HWHAT IS THE NEW Y-MAX VALUE?)
1290  FORMAT (/,5X,51HWHAT IS THE DESIRED DIMENSICN IN THE "X" DIRECTION
     1?)
1300  FORMAT (//,10X,32H -THE MAX LENGTH IS 21.0 INCHES.)
1310  FORMAT (/,5X,51HWHAT IS THE DESIRED DIMENSICN IN THE "Y" DIRECTION
     1?)
1320  FORMAT (//,10X,45HTHE LETTERING SCALE FACTCR WILL BE MULTIPLIED/,1
     10X,35HTIMES THE CURRENT LETTERING HEIGHT.,//,5X,58HI.E.  A NUMBER
     2GREATER THAN 1.0 INCREASES, AND VICE VERSA.//,10X,30HWHAT SCALE FA
     3CTOR DC YOU WANT?)
1330  FORMAT (/,37HYOUR ANSWER MUST BE GREATER THAN 0.0.)
1340  FORMAT (/,10X,35HHCW MANY INCHES IN THE X DIRECTION ,/,10X,36H(LEF
     1T CR RIGHT), DO YOU WANT TO MOVE,/,10X,45HMCVE THE LEGEND BOX FROM
     2 ITS PRESENT POSITION,//,10X,40HNOTE: 1. DEFAULT PLOT SIZE IS  8.5
     3 X 6.0,/,10X,26H       2. LEFT IS NEGATIVE ,/,10X,26H       3. RIGHT
     4 IS POSITIVE)
1350  FORMAT (/,10X,35HHCW MANY INCHES IN THE Y DIRECTION ,/,10X,33H(UP
     1OR DOWN), DC YOU WANT TO MOVE,/,10X,45HMOVE THE LEGEND BOX FROM IT
     2S PRESENT PCSITION,//,10X,40HNOTE: 1. DEFAULT PAGE SIZE IS  8.5 X
     36.0,/,10X,26H       2. DOWN IS NEGATIVE ,/,10X,23H       3. UP IS PO
     4SITIVE)
1360  FORMAT (//,10X,36HTHE FOLLCWING OPTIONS ARE AVAILABLE:,//,10X,44H1
     1.  MAKE METAFILE OF PREVIOUSLY SAVEC CURVE.,/,10X,9H2.  QUIT.,//,1
     25X,12HENTER 1 OR 2)
1370  FORMAT (/,10X,33HYCUR INPUT MUST BE EITHER 1 OR 2.)
1375  FORMAT (/,'ILLEGAL INPUT    NULL STRING NOT ALLOWED. TRY AGAIN.')
1380  FORMAT (5E14.7)
1390  FORMAT (5I5)
1400  FORMAT (11A4)
1410  FORMAT (2A4)
      ENC
C=====================================================================
      SUBROUTINE SELCRV (I,C,CMIN,CMAX,TITLE,CATA,U,FBGC,NS,NC,NPTS,
     1IEST)
C     SELECT VARIABLES TC PLCT
C=====================================================================
      IMPLICIT REAL*4 (A-H,O-Z)
      INTEGER TITLE,CHST
      DIMENSICN TITLE(11),CHST(11),DATA(501,83),U(501,10),C(501),FBGC(10
     1,83)
```

```
      CALL FRTCMS ('CLRSCRN ')
C=======================================================================
C         SELECT TYPE OF VARIABLE TO PLOT
C=======================================================================
10    WRITE (5,260) I
      CALL RDINT (ITYPE)
      IF ((ITYPE.GE.1).AND.(ITYPE.LE.5)) GO TO 20
      GO TO 10
20    GO TO (30,60,120,160,200), ITYPE
C=======================================================================
C         SELECT STATE VARIABLE
C=======================================================================
30    WRITE (5,290) I
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.NS)) GO TO 40
      WRITE (5,320) NS,IANS
      GO TO 30
40    C(1)=DATA(1,IANS)
      CMIN=C(1)
      CMAX=C(1)
      DO 50 J=2,NPTS
      C(J)=DATA(J,IANS)
      IF (C(J).LT.CMIN) CMIN=C(J)
      IF (C(J).GT.CMAX) CMAX=C(J)
50    CONTINUE
      GO TO 240
C=======================================================================
C         SELECT FEEDBACK  <C>*X
C=======================================================================
60    CALL FRTCMS ('CLRSCRN ')
70    WRITE (5,270) I
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.NS)) GO TO 80
      WRITE (5,320) NS,IANS
      GO TO 70
80    CONTINUE
      DO 100 I=1,NPTS
      C(I)=0.0
      DO 90 J=1,NS
      IF (IEST.NE.3) C(I)=C(I)+FBGC(IANS,J)*DATA(I,J)
      IF (IEST.EQ.3) C(I)=C(I)+FBGC(IANS,J)*DATA(I,NS+J)
90    CONTINUE
100   CONTINUE
      CMAX=C(1)
      CMIN=C(1)
      DO 110 I=2,NPTS
      IF (C(I).LT.CMIN) CMIN=C(I)
      IF (C(I).GT.CMAX) CMAX=C(I)
```

```
110   CONTINUE
      GO TO 240
C==================================================================
C          SELECT CONTROL INPUT
C==================================================================
120   CALL FRTCMS ('CLRSCRN ')
130   WRITE (5,280) I
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.NC)) GO TO 140
      WRITE (5,320) NC
      GO TO 130
140   C(1)=U(1,IANS)
      CMIN=C(1)
      CMAX=C(1)
      DO 150 J=2,NPTS
      C(J)=U(J,IANS)
      IF (C(J).LT.CMIN) CMIN=C(J)
      IF (C(J).GT.CMAX) CMAX=C(J)
150   CONTINUE
      GO TO 240
C==================================================================
C          SELECT STATE OBSERVER
C==================================================================
160   IF (IEST.EQ.3) GO TO 170
      WRITE (5,300)
      GO TO 10
170   WRITE (5,310) I
      CALL RDINT (IANS)
      IF ((IANS.GE.1).AND.(IANS.LE.NS)) GO TO 180
      WRITE (5,320) NS,IANS
      GO TO 170
180   C(1)=DATA(1,IANS+NS)
      CMIN=C(1)
      CMAX=C(1)
      DO 190 J=2,NPTS
      C(J)=DATA(J,IANS+NS)
      IF (C(J).LT.CMIN) CMIN=C(J)
      IF (C(J).GT.CMAX) CMAX=C(J)
190   CONTINUE
      GO TO 240
C==================================================================
C          SELECT RECONSTRUCTION ERROR
C==================================================================
200   IF (IEST.EQ.3) GO TO 210
      WRITE (5,300)
      GO TO 10
210   WRITE (5,360) I
      CALL RDINT (IANS)
```

```
      IF ((IANS.GE.1).AND.(IANS.LE.NS)) GO TO 220
      WRITE (5,320) NS,IANS
      GO TO 210
220   C(1)=DATA(1,IANS)-DATA(1,IANS+NS)
      CMIN=C(1)
      CMAX=C(1)
      DO 230 J=2,NPTS
      C(J)=DATA(J,IANS)-DATA(J,IANS+NS)
      IF (C(J).LT.CMIN) CMIN=C(J)
      IF (C(J).GT.CMAX) CMAX=C(J)
230   CONTINUE
240   CALL FRTCMS ('CLRSCRN ')
      IF (CMIN.NE.CMAX) GO TO 250
      WRITE (5,330)
      CMAX=1.0+CMIN
250   WRITE (5,340)
      WRITE (5,350)
      CALL RECFST (TITLE)
      RETURN
C------------------------------------------------------------------------
260   FORMAT (//,10X,46HWHICH TYPE OF VARIABLE DO YOU WISH TO PLOT AS ,1
     12HCURVE NUMBER,I2,1H?,//,15X,36H1. STATE VARIABLE (IE., X1, X2, ET
     2C),/,15X,35H2. FEEDBACK CONTROL (IE., U = -C*X),/,15X,36H3. CONTRO
     3L INPUT (IE., U1, U2, ETC.),/,15X,54H4. STATE ESTIMATE (OBSERVER)
     4(IE., XHAT1, XHAT2, ETC.),/,15X,60H5. STATE RECONSTRUCTION ERROR (
     5IE., X1-XHAT1, X2-XHAT2, ETC),//,10X,18HENTER 1,2,3,4 OR 5)
270   FORMAT (//,10X,50HWHAT IS THE SUBSCRIPT OF THE FEEDBACK CONTROL TH
     1AT,/,10X,30HYOU WANT TO PLOT AS THE NUMBER,I2,15H CURVE VS TIME?)
280   FORMAT (/,10X,47HWHAT IS THE SUBSCRIPT OF THE CONTROL INPUT THAT,/
     1,10X,30HYOU WANT TO PLOT AS THE NUMBER,I3,15H CURVE VS TIME?)
290   FORMAT (//,10X,48HWHAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
     1,/,10X,30HYOU WANT TO PLOT AS THE NUMBER,I3,15H CURVE VS TIME?)
300   FORMAT (//,30X,8HERROR???/,15X,24HSTATE ESTIMATES WERE NOT,12H CAL
     1CULATED.)
310   FORMAT (//,10X,48HWHAT IS THE SUBSCRIPT OF THE STATE ESTIMATE THAT
     1,/,10X,30HYOU WANT TO PLOT AS THE NUMBER,I3,15H CURVE VS TIME?)
320   FORMAT (/,10X,35HTHE SUBSCRIPT MUST BE BETWEEN 1 AND,I3,2H .,1H(,I
     12,30H) WAS THE NUMBER YOU SELECTED.)
330   FORMAT (//,10X,48HTHIS VARIABLE IS A CONSTANT THROUGHOUT THE TIME
     1,10HEVALUATED.)
340   FORMAT (//,10X,42HWHAT IS THE CURVE LABEL FOR THIS VARIABLE?,/)
350   FORMAT (/,10X,34HNOTE:  1. 40 CHARACTERS MAX LENGTH,/,10X,55H
     1  2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS,/,10X,34H
     2     ENCLOSED IN PARENTHESES.,/,10X,27H            IE. (A) => ALPHA ,
     3/,10X,27H              (B) => BETA  ,/,10X,27H             (F) =>
     4 PHI   ,/,10X,27H             (Q) => THETA )
360   FORMAT (/,10X,43HWHAT IS THE SUBSCRIPT OF THE RECONSTRUCTION,/,10X
     1,41HERROR THAT YOU WANT TO PLOT AS THE NUMBER,I3,7H CURVE?)
```

```
      END
C=====================================================================
      SUBROUTINE CRVEXC (C1,C1MIN,C1MAX,TITLE1,C2,C2MIN,C2MAX,TITLE2)
C     EXCHANGES DATA FROM C2 TO C1
C=====================================================================
      IMPLICIT REAL*4 (A-H,O-Z)
      INTEGER TITLE1,TITLE2
      DIMENSION TITLE1(11),TITLE2(11),C1(501),C2(501)
      DO 10 I=1,11
      TITLE1(I)=TITLE2(I)
10    CONTINUE
      DO 20 I=1,501
      C1(I)=C2(I)
20    CONTINUE
      C1MAX=C2MAX
      C1MIN=C2MIN
      RETURN
      END
C=====================================================================
      SUBROUTINE HEADS (HEAD,N)
C     GETS THE HEADING CHARACTER STRING
C=====================================================================
      INTEGER HEAD(11),N
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,10) N
      WRITE (5,20)
      CALL RDCHST (HEAD)
      RETURN
C---------------------------------------------------------------------
10    FORMAT (/,10X,35HWHAT IS THE DESIRED HEADING NUMBER ,I1,1H?,/)
20    FORMAT (/,10X,34HNOTE:  1. 40 CHARACTERS MAX LENGTH,/,10X,55H
     1  2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS,/,10X,34H
     2     ENCLOSED IN PARENTHESES.,/,10X,27H              IE. (A) => ALPHA ,
     3/,10X,27H                 (B) => BETA  ,/,10X,27H              (F) =>
     4 PHI    ,/,10X,27H                 (Q) => THETA )
      END
C=====================================================================
      SUBROUTINE CURINT (IANS)
C     DISPLAYS THE CURRENT VALUE OF AN INTEGER
C=====================================================================
      WRITE (5,10) IANS
      RETURN
C---------------------------------------------------------------------
10    FORMAT (/,10X,21HTHE CURRENT VALUE IS ,I10)
      END
C=====================================================================
      SUBROUTINE CUREAL (ANS)
C     DISPLAYS THE CURRENT VALUE OF A REAL VARIABLE
```

```
C=====================================================================
      WRITE (5,10) ANS
      RETURN
C---------------------------------------------------------------------
10    FORMAT (/,10X,21HTHE CURRENT VALUE IS ,F15.5)
      END
C=====================================================================
      SUBROUTINE CURCHR (CHST)
C     DISPLAYS THE CURRENT VALUE OF A CHARACTER STRING
C=====================================================================
      INTEGER CHST(11)
      WRITE (5,10) (CHST(I),I=1,11)
      RETURN
C---------------------------------------------------------------------
10    FORMAT (/,10X,32HTHE CURRENT CHARACTER STRING IS:,/,10X,11A4)
      END
C=====================================================================
C     WRITE CURVE DATA TO FILE USING FILEDEF 4
C=====================================================================
      SUBROUTINE FILECV (NPTS,NCURVS,NHEAD,HEAD1,HEAD2,HEAD3,TITLE1
     1,TITLE2,TITLE3,TITLE4,XPAGE,YPAGE,DELTAX,DELTAY,SCALEH,
     2C1MIN,C1MAX,C2MIN,C2MAX,C3MIN,C3MAX,C4MIN,C4MAX,
     3TIME,C1,C2,C3,C4)
      IMPLICIT REAL*4 (A-H,O-Z)
      INTEGER TITLE1,TITLE2,TITLE3,TITLE4,HEAD1,HEAD2,HEAD3
      DIMENSION TITLE1(11),TITLE2(11),TITLE3(11),TITLE4(11),HEAD1(11),HE
     1AD2(11),HEAD3(11),TIME(501),C1(501),C2(501),C3(501),C4(501),NAMFIL
     2(2)
5     WRITE (5,30)
      READ (5,80,END=25,ERR=25)  NAMFIL
      CALL FRTCMS ('FILEDEF ','04      ','DISK    ',NAMFIL,'DATA    ',
     1'A1      ')
      WRITE (5,40) NAMFIL
      REWIND 4
      WRITE (4,60) NPTS,NCURVS,NHEAD
      WRITE (4,70) (HEAD1(I),I=1,11)
      IF (NHEAD.EQ.1) GO TO 10
      WRITE (4,70) (HEAD2(I),I=1,11)
      IF (NHEAD.EQ.2) GO TO 10
      WRITE (4,70) (HEAD3(I),I=1,11)
10    CONTINUE
      WRITE (4,70) (TITLE1(I),I=1,11)
      IF (NCURVS.EQ.1) GO TO 20
      WRITE (4,70) (TITLE2(I),I=1,11)
      IF (NCURVS.EQ.2) GO TO 20
      WRITE (4,70) (TITLE3(I),I=1,11)
      IF (NCURVS.EQ.3) GO TO 20
      WRITE (4,70) (TITLE4(I),I=1,11)
```

```
20    CONTINUE
      WRITE (4,50) XPAGE,YPAGE,DELTAX,DELTAY,SCALEH
      WRITE (4,50) C1MIN,C1MAX,C2MIN,C2MAX
      WRITE (4,50) C3MIN,C3MAX,C4MIN,C4MAX
      WRITE (4,50) (TIME(I),C1(I),C2(I),C3(I),C4(I),I=1,NPTS)
      RETURN
25    WRITE (5,90)
      REWIND 5
      GO TO 5
C-----------------------------------------------------------------------
30    FORMAT (//,10X,48HWHAT FILE NAME DO YOU WANT THE CURVE DATA STORED
     1,7H UNDER?,/,20X,18H(8 CHARACTERS MAX))
40    FORMAT (/,10X,36HTHE CURVE DATA IS BEING FILED UNDER ,2A4,5H DATA)
50    FORMAT (5E14.7)
60    FORMAT (5I5)
70    FORMAT (11A4)
80    FORMAT (2A4)
90    FORMAT (/,15X,'ILLEGAL INPUT     TRY AGAIN.')
      END
C=======================================================================
C  SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY            =
C  INTO A FORTRAN PROGRAM.  IF THE USER INADVERTENLY ENTERS AN IMPROPER=
C  DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.      =
C=======================================================================
      SUBROUTINE RDINT (IANS)
      INTEGER COUNT
C-----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 50
20    CONTINUE
      READ (5,*,END=40,ERR=40) IANS
      IF (IANS) 40,40,30
30    CONTINUE
      RETURN
40    REWIND 5
      WRITE (5,70)
      GO TO 10
50    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 INADVERTENLY    =
C   ENTERS  A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY=
C========================================================================
      SUBROUTINE RDCHAR (IANS)
      INTEGER COUNT
C------------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      REWIND 5
      READ (5,70,END=30,ERR=30) IANS
      RETURN
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 RDREAL -- INTERACTIVELY READS A REAL NUMBER REPLY        =
C   INTO A FORTRAN PROGRAM.  IF THE USER INADVERTENLY 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------------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      READ (5,*,END=30,ERR=30) ANSR
      RETURN
```

```
30    REWIND 5
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
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 RDCHST -- INTERACTIVELY READS A CHARACTER STRING REPLY    =
C  UP TO 40 CHARACTERS LONG AND FORMATS THE CHARACTER STRING FOR USE    =
C  BY A DISSPLA PRINT ROUTINE.                                          =
C=======================================================================
      SUBROUTINE RDCHST (CHST)
      INTEGER CHST(11),I
      DATA IBL/'    '/,IDOL/'$   '/
C-----------------------------------------------------------------------
      CALL GETCHS (CHST)
      CHST(11) = IBL
      DO 10 I =1,11
         IF(CHST(I).NE.IBL) GO TO 10
         CHST(I) = IDOL
         GO TO 20
10    CONTINUE
20    RETURN
C-----------------------------------------------------------------------
      END
C=======================================================================
C  SUBROUTINE GETCHS -- INTERACTIVELY READS A CHARACTER STRING REPLY    =
C  UP TO 40 CHARACTERS LONG.  IF THE USER INADVERTENTLY ENTERS A NULL   =
C  STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY                =
C=======================================================================
      SUBROUTINE GETCHS (CHST)
      INTEGER COUNT,CHST(20),I
C-----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      REWIND 5
      READ (5,70,END=30,ERR=30) (CHST(I),I = 1,10)
      RETURN
30    REWIND 5
```

```
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
C-----------------------------------------------------------------------
50    FORMAT (1X,'WARNING:  NULL STRINGS ARE NOT ALLOWED, THE PROGRAM',
     1/,'WILL TERMINATE IF ANOTHER NULL STRING IS ENTERED.')
60    FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED )
70    FORMAT (10A4)
      END
C=======================================================================
C  SUBROUTINE NEWSCR -- CLEARS THE SCREEN WITHOUT ERASING THE         =
C  PREVIOUS SCREEN'S INFORMATION.                                     =
C=======================================================================
      SUBROUTINE NEWSCR
      WRITE (5,10)
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
10    FORMAT(//////////////////////)
      END
```

LIST OF REFERENCES

1. Hall, W. E., _Computational Methods for the Synthesis of Rotary- Wing VTOL Aircraft Control Systems_, Ph.D. Dissertation, Stanford Univ.,Aug. 1971.

2. Walker, R. A., _User's Manual for OPTSYS 4 at SCIP_, Stanford Univ., Aero/Astro Dept., Dec. 1979.

3. Liu, G., _User's Manual for OPTSYS 5 at CIT_, Stanford Univ., Aero/Astro Dept., Aug. 1982.

4. Hoden, J. G., _Interactive Implementation of the Optimal Systems Control Design Program (OPTSYSX) on the IBM/3033_, MS thesis, Naval Postgraduate School, Monterey, CA., Mar. 1984.

5. Bryson, A. E. and Ho, Y. C., _Applied Optimal Control_, Hemisphere Pub. Co., 1969, (2nd Printing, 1975).

6. Sandell, N. R. and Athens, M., _Modern Control Theory_, Center for Advanced Engineering Study, Massachusetts Institute of Technology, 1974.

7. Kwakernaak, H. and Sivan, R., _Linear Optimal Control Systems_, Wiley-Interscience, 1972.

BIBLIOGRAPHY

Lipschutz, S. and Poe, A., _Programming with FORTRAN_, Schaum's Outline Series, McGraw-Hill, 1978.

Melsa, J.L. and Jones, S.K., _Computer Programs for Computational Assistance in the Study of Linear Control Theory_, McGraw-Hill, 1973.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria Va 22314	2
2.	Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, Ca 93943	1
3.	Library, Code 0142 Naval Postgraduate School Monterey Ca 93943	2
4.	Professor D.J. Collins Code 67Co Department of Aeronautics Naval Postgraduate School Monterey, Ca 93943	5
5.	Professor R.H. Nunn Code 69Nn Department of Mechanical Engineering Naval Postgraduate School Monterey, Ca 93943	1
6.	CDR V.C. Gordon 315 Alameda Ave. Coronado, CA 92118	1
7.	CDR H.A. Diel AIR - 5421D Naval Air Systems Command Washington, DC 20361	2
8.	Professor A.E. Bryson Department of Aeronautics and Astronautics Stanford University Stanford, CA 94305	1

210164

Thesis
D567152 Diel
c.1
Development of graphical time response using the OPTSYSX Program.