

AD-A152 221

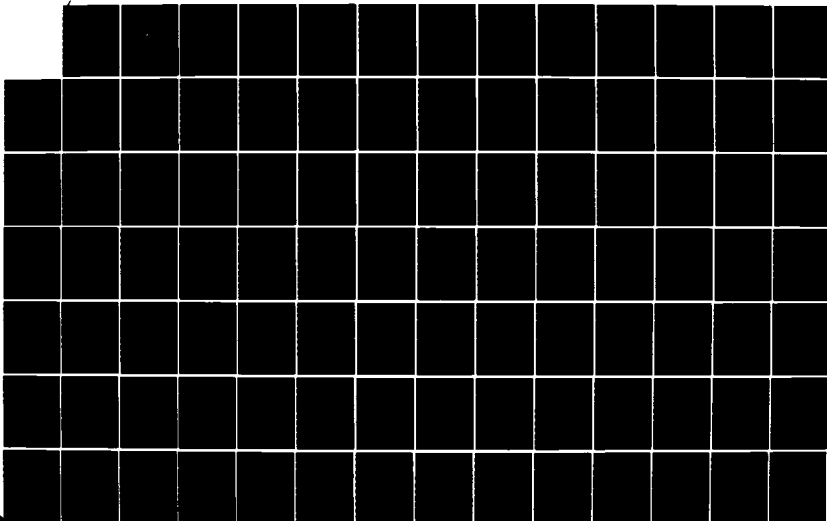
DEVELOPMENT OF GRAPHICAL TIME RESPONSE USING THE
OPTSYSX PROGRAM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY
CA H A DIEL SEP 84

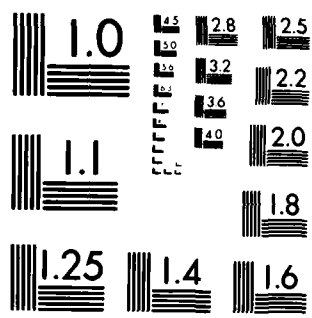
1/3

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A152 221

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

DTIC
ELECTE

APR 4 1985

D

Approved for public release; distribution unlimited.

DTIC FILE COPY

85 03 18 122

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
AD-A152221		
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).		

DD FORM 1473
1 JAN 73EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

1 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

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.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

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

Author:

Harry A. Diel
Harry A. Diel

Approved by:

Daniel J. Collins
D. J. Collins, Thesis Advisor

R. H. Nunn
R. H. Nunn, Second Reader

M. F. Platzter
M. F. Platzter, Chairman,
Department of Aeronautics

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



411

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.

*Additional programs for use with system
closed loop systems, optimization*

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 CPTimal 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, E, 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,EKR=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 one 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 "3" 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

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

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 $\dot{x}$ 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.

$$\dot{x} = [F+G*C]*x + [G]*uc \quad (2.2)$$

$$u = [C]*x + uc \quad (2.3)$$

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 x_e , $\dot{x}_e$ and $\tilde{x}$ are the state estimate vector, the derivative of the state estimate vector and the state reconstruction error, respectively.

¹The 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.

$$\dot{x} = [F+G*C]*x + [G]*uc \quad (2.2)$$

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

$$\dot{x}_e = [F]*x_e + [G]*u + [K]*(z - [H]*x_e) \quad (2.5)$$

$$u = [C]*x_e \quad (2.6)$$

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

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

$$\begin{bmatrix} \dot{x} \\ \dot{x}_e \end{bmatrix} = \begin{bmatrix} F & G*C \\ K*H & F+G*C-K*H \end{bmatrix} \begin{bmatrix} x \\ x_e \end{bmatrix} + \begin{bmatrix} G \\ G \end{bmatrix} * uc \quad (2.8)$$

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

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

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

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

$$\dot{x} = [F]*x_e + [G]*u + [K]*(z - [H]*x_e) \quad (2.5)$$

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

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.

$$\begin{bmatrix} \dot{x} \\ \dot{x}_e \end{bmatrix} = \begin{bmatrix} F & 0 \\ K*H & F-K*H \end{bmatrix} * \begin{bmatrix} x \\ x_e \end{bmatrix} + \begin{bmatrix} G \\ G \end{bmatrix} * u \quad (2.9)$$

a. System Selection

When the OPTCALC 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. OPEN LOOP TIME RESPONSE.
2. CLOSED LOOP TIME RESPONSE.
3. CLOSED LOOP FILTER ONLY TIME RESPONSE.
4. CLOSED 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 ($2*N_s$). The $[G]$ matrix is also augmented by repeating the first N_s rows again beginning at row N_s+1 , making the new $[G]$ matrix dimensions ($2*N_s, N_c$).

The augmented system can be evaluated by simply doubling the old system row and column dimension (N_s) 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 N_s and N_c , 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 (u_c) and each state (x) and state estimate ($\hat{x}$). 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 CPTSYS 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 CPTSYS EXEC gives the user the options to:

1. RUN OPTSYSX 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 lower case letters or numbers 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 CPTSYS...:
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:

$$\dot{X} = \{F\} * X + \{G\} * U + \{GAM\} * (W + W_0)$$

MEASUREMENT EQUATION--

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

REGULATOR PERFORMANCE INDEX--

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

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 ?

n

TYPE "YES" OR "NO".

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 FLAG/PARAMETER SETTINGS FOR THIS RUN ARE AS FOLLOWS:

IOL	IQ	IR	ISS	IM	ITF1	ITF2	ITF3	IFDFW	IE	IDDEBUG
3	0	0	0	1	0	0	0	0	0	0

ISSET	IDSTAE	IPSD	IYU	INORM	IREG	NS	NC	NOB	NG
0	0	0	0	0	0	4	0	0	0

ORDER OF SYSTEM = 4
NUMBER OF CONTROLS = 0
NUMBER OF OBSERVATIONS = 0
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" OR "NO".

n

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

0.0	0.10000E+01	0.0	0.0
0.0	-0.41500E+00	-0.11100E-01	0.0
0.98000E+01	-0.14300E+01	-0.19800E-01	0.0
0.0	0.0	0.10000E+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 (CUTPUT 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
 $XDCT = \{F\} * X + \{G\} * UC$
2. CLOSED LOOP TIME RESPONSE
 $XDCT = \{F - G * C\} * X + \{G\} * UC, \quad U = \{C\} * X$
3. OPTIMIZED FILTER CLOSED LOOP SYSTEM RESPONSE.

$$\begin{aligned} \dot{X} &= \{F\} * X + \{G\} * U, & Z &= \{H\} * X \\ \dot{X}_H &= \{F\} * X_H + \{G\} * U + \{K\} * \{Z - H * X_H\} \end{aligned}$$

4. FILTER + REGULATOR CLOSED LOOP SYSTEM RESPONSE.
 $\dot{X} = \{F + G * C\} * X + \{G\} * U, \quad Z = \{H\} * X$
 $\dot{X}_H = \{F\} * X_H + \{G\} * U + \{K\} * \{Z - H * X_H\}, \quad U = \{C\} * X_H$

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 PCINTS DO YOU WANT TO CALCULATE?

?
500

DOES THE SYSTEM UTILIZE A DRIVING FUNCTION (CONTRCL 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 APEAS.

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.

3
5

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

ALL CCNROLS, 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	0.4033316D+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	0.3550559D+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	-0.99887356D-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	J.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
F {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., $\dot{U} = -C^*X$)
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., $\dot{U} = -C^*X$)
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., $U = -C \cdot X$)
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. REPLCT 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 3.5 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. DCWN 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. REPLCT 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. REPLCT 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 DISSEFA 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 DISPLA 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 DISPLA METAFILE THAT YOU JUST CREATED?
(Y OR N)

Y
B (120) F/O
DASD 001 LINKED R/O; R/W BY MVS
Z (001) F/C - OS
DASD 001 DETACHED
CREATING NEW FILE:
CREATING NEW FILE:
FUN 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.

OPEN LOOP SYSTEM
EXAMPLE 2
MODERN CONTROL THEORY

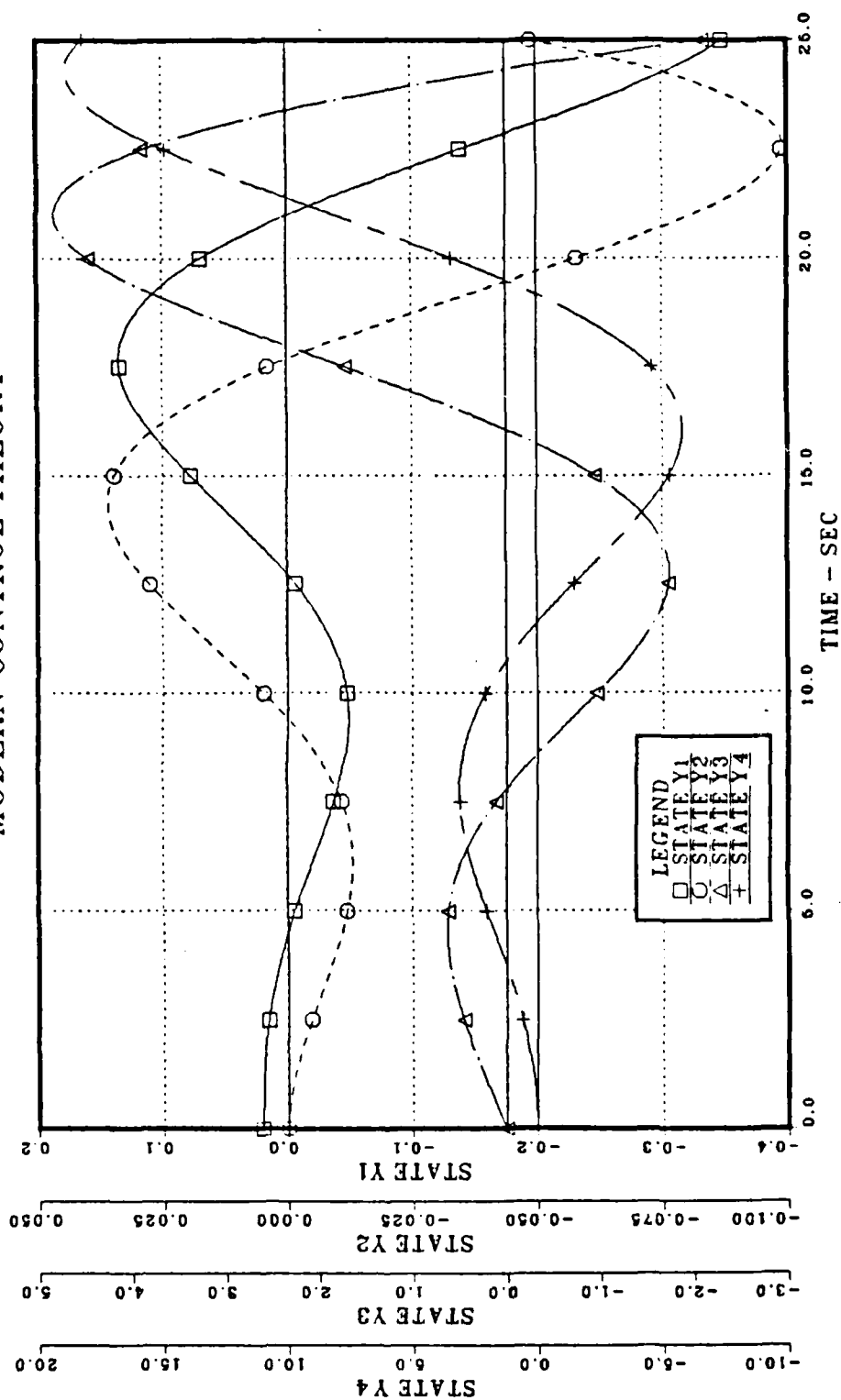


Figure 3.1 Open-loop Time Response

E. 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=C.01/0.02 20:19:44
optsys
```

THE OPTSYS EXEC CONTROLS A TRIO OF PROGRAMS:

1. OPTSYSX FCRTRAN (SYSTEM ANALYSIS)
2. OPTCALC FCRTRAN (CALCULATE TIME RESPONSE)
3. OPTPLOT FCRTRAN (DISPLA 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 CCNTAINS 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 CPTSYS 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:

$$\dot{X} = \{F\} * X + \{G\} * U + \{GAM\} * (W + W_0)$$

MEASUREMENT EQUATION--

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

REGULATOR PERFORMANCE INDEX--

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

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

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.

?
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 $10.0^{**} \{-1E\}$ 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
PRICR 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.

DO YOU DESIRE REGULATOR SYNTHESIS ONLY?

TYPE "YES" OR "NO".

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.

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".

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".

DO YOU WISH TO SAVE THE "E"-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

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 NOISE 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

ISSET	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 OBSERVATIONS = 4

NUMBER OF PROCESS NOISE 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" OR "NO".

n

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

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

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

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

?
1

?
 0 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..

0.1000E+01	0.0	0.0	0.0
0.0	0.1000E+01	0.0	0.0
0.0	0.0	0.1000E+01	0.0
0.0	0.0	0.0	0.1000E+01

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}

?
0

THE ELEMENT G(1, 1)=

?
6.27

THE ELEMENT G(2, 1)=

?
9.8

THE ELEMENT G(3, 1)=

?
0

THE ELEMENT G(4, 1)=

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}

?
131.3

THE ELEMENT B(1, 1)=

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
0.6270E+01
0.9800E+01
0.0

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

0.1313E+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-LCCP 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	-1.397574D-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-LCCP 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...

-8.5492E-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
 $\dot{X} = \{F\} * X + \{G\} * UC$
2. CLOSED LOOP TIME RESPONSE
 $\dot{X} = \{F - G * C\} * X + \{G\} * UC, \quad U = \{C\} * X$
3. OPTIMIZED FILTER CLOSED LOOP SYSTEM RESPONSE.
 $\dot{X} = \{F\} * X + \{G\} * UC, \quad Z = \{H\} * X$
 $\dot{X} = \{F\} * X + \{G\} * U + \{K\} * \{Z - H * X\}$
4. FILTER + REGULATOR CLOSED LOOP SYSTEM RESPONSE.
 $\dot{X} = \{F + G * C\} * X + \{G\} * UC, \quad Z = \{H\} * X$
 $\dot{X} = \{F\} * X + \{G\} * U + \{K\} * \{Z - H * X\}, \quad U = \{C\} * X$

SELECT 1, 2, 3 OR 4.

3
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?

?
0

INPUT START TIME IN SECONDS. (NORMALLY 0.0)

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

?
25

INPUT STOP TIME IN SECONDS.

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.

?
500

HOW MANY PCINTS DO YOU WANT TO CALCULATE?

DOES THE SYSTEM UTILIZE A DRIVING FUNCTION (CONIROL 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.

5.3

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

ALL CCNTROLS, STATES AND STATE ESTIMATES CAN BE PLOTTED.

TIME	U(1)	X(1)	X(2)	X(3)
0.0	0.0	0.20000 00D-01	0.0	0.0
0.50	0.0	0.11031 08D-01	-0.28350 04D-01	0.44382 30D-01
1.00	0.0	-0.31240 39D-02	-0.24990 14D-01	0.79231 24D-01
1.50	0.0	-0.12142 95D-01	-0.10444 92D-01	0.68503 92D-01
2.00	0.0	-0.13720 04D-01	0.34159 74D-02	0.25130 11D-01
2.50	0.0	-0.97362 74D-02	0.11372 85D-01	0.24639 73D-01
3.00	0.0	-0.34552 04D-02	0.12764 82D-01	0.59914 16D-01
3.50	0.0	0.22423 63D-02	0.94651 82D-02	0.71728 35D-01
4.00	0.0	0.56808 82D-02	0.41768 01D-02	0.62097 10D-01
4.50	0.0	0.64850 95D-02	-0.74959 87D-03	0.39545 11D-01
5.00	0.0	0.52317 22D-02	-0.39090 40D-02	0.14118 85D-01
5.50	0.0	0.29374 93D-02	-0.49262 79D-02	0.63085 58D-02
6.00	0.0	0.59766 76D-03	-0.42028 51D-02	0.17762 40D-01
6.50	0.0	-0.11062 74D-02	-0.25249 67D-02	0.20100 80D-01
7.00	0.0	-0.19035 54D-02	-0.69643 301D-03	0.15809 86D-01
7.50	0.0	-0.18757 04D-02	0.70525 77D-03	0.84177 29D-02
8.00	0.0	-0.13141 53D-02	0.14216 27D-02	0.11316 61D-02
8.50	0.0	-0.56469 48D-03	0.14803 70D-02	0.39745 07D-02
9.00	0.0	0.91051 25D-04	0.10899 64D-02	0.61703 45D-02
9.50	0.0	0.49518 28D-03	0.51755 73D-03	0.58192 87D-02
10.00	0.0	0.61687 27D-03	-0.85062 41D-05	0.38981 43D-02
10.50	0.0	0.51774 99D-03	-0.35158 87D-03	0.15036 36D-02
11.00	0.0	0.30233 67D-03	-0.47495 32D-03	0.50664 95D-03
11.50	0.0	0.73033 14D-04	-0.41817 81D-03	0.16809 06D-02
12.00	0.0	-0.98853 95D-04	-0.25951 92D-03	0.19649 39D-02
12.50	0.0	-0.18313 20D-03	-0.79951 28D-04	0.15814 98D-02
13.00	0.0	-0.18538 26D-03	0.61252 30D-04	0.87065 65D-03
13.50	0.0	-0.13307 46D-03	0.13627 40D-03	0.15090 12D-03
14.00	0.0	-0.60089 64D-04	0.14603 67D-03	0.36588 26D-03
14.50	0.0	0.53076 24D-05	0.11008 14D-03	0.59934 13D-03
15.00	0.0	0.46744 27D-04	0.54514 21D-04	0.57887 39D-03
15.50	0.0	0.60412 64D-04	0.21811 62D-05	0.39747 40D-03
16.00	0.0	0.51867 17D-04	-0.32842 78D-04	0.16294 61D-03
16.50	0.0	0.31203 03D-04	-0.46330 21D-04	0.38451 10D-04
17.00	0.0	0.85704 07D-05	-0.41751 56D-04	0.15964 41D-03
17.50	0.0	-0.87893 15D-05	-0.26629 10D-04	0.19296 83D-03
18.00	0.0	-0.17647 26D-04	-0.89652 53D-05	0.15888 05D-03
18.50	0.0	-0.18348 21D-04	0.52437 73D-05	0.90382 70D-04
19.00	0.0	-0.13485 38D-04	0.13059 90D-04	0.19145 84D-04
19.50	0.0	-0.63771 76D-05	0.14407 18D-04	0.33222 73D-04
20.00	0.0	0.14426 43D-06	0.11111 254D-04	0.57997 97D-04
20.50	0.0	0.43880 11D-05	0.57233 06D-05	0.57451 93D-04
21.00	0.0	0.59036 99D-05	0.52121 49D-06	0.40424 31D-04
21.50	0.0	0.51868 79D-05	-0.30485 85D-05	0.17506 08D-04
22.00	0.0	0.32109 31D-05	-0.45100 07D-05	0.26338 78D-05
22.50	0.0	0.98082 65D-06	-0.41618 88D-05	0.15104 81D-04
23.00	0.0	-0.76927 17D-06	-0.27256 18D-05	0.18917 76D-04
23.50	0.0	-0.16961 69D-05	-0.99097 12D-06	0.15934 80D-04
24.00	0.0	-0.18131 70D-05	0.43647 08D-06	0.93513 35D-05
24.50	0.0	-0.13640 03D-05	0.12478 88D-05	0.23119 75D-05
25.00	0.0	-0.67309 20D-06	0.14190 68D-05	0.29848 95D-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)

P {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., X_1 , X_2 , ETC)
2. FEEDBACK CONTROL (IE., $U = -C*X$)
3. CONTROL INPUT (IE., U_1 , U_2 , ETC.)
4. STATE ESTIMATE (OBSERVER) (IE., $\hat{X}_1$, $\hat{X}_2$, ETC.)
5. STATE RECONSTRUCTION ERROR (IE., $X_1 - \hat{X}_1$, $X_2 - \hat{X}_2$, 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., $\dot{U} = -C \cdot X$)
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., $\dot{U} = -C \cdot X$)
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., X_1 , X_2 , ETC)
2. FEEDBACK CONTROL (IE., $U = -C \cdot X$)
3. CONTROL INPUT (IE., U_1 , U_2 , ETC.)
4. STATE ESTIMATE (OBSERVER) (IE., $\hat{X}_1$, $\hat{X}_2$, ETC.)
5. STATE RECONSTRUCTION ERROR (IE., $X_1 - \hat{X}_1$, $X_2 - \hat{X}_2$, 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
(E) => 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
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 DISSELA 9.0 -- 16300 VECTORS GENERATED IN 1 PLOT FRAMES
PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CA.
1883 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 DISSFLA 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
 E (120) R/O
 DASD 001 LINKED R/O; R/W BY MVS; R/O BY 0085P
 Z (001) F/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 follows
 as figure 3.2.

CLOSED LOOP SYSTEM
EXAMPLE 3
MODERN CONTROL THEORY

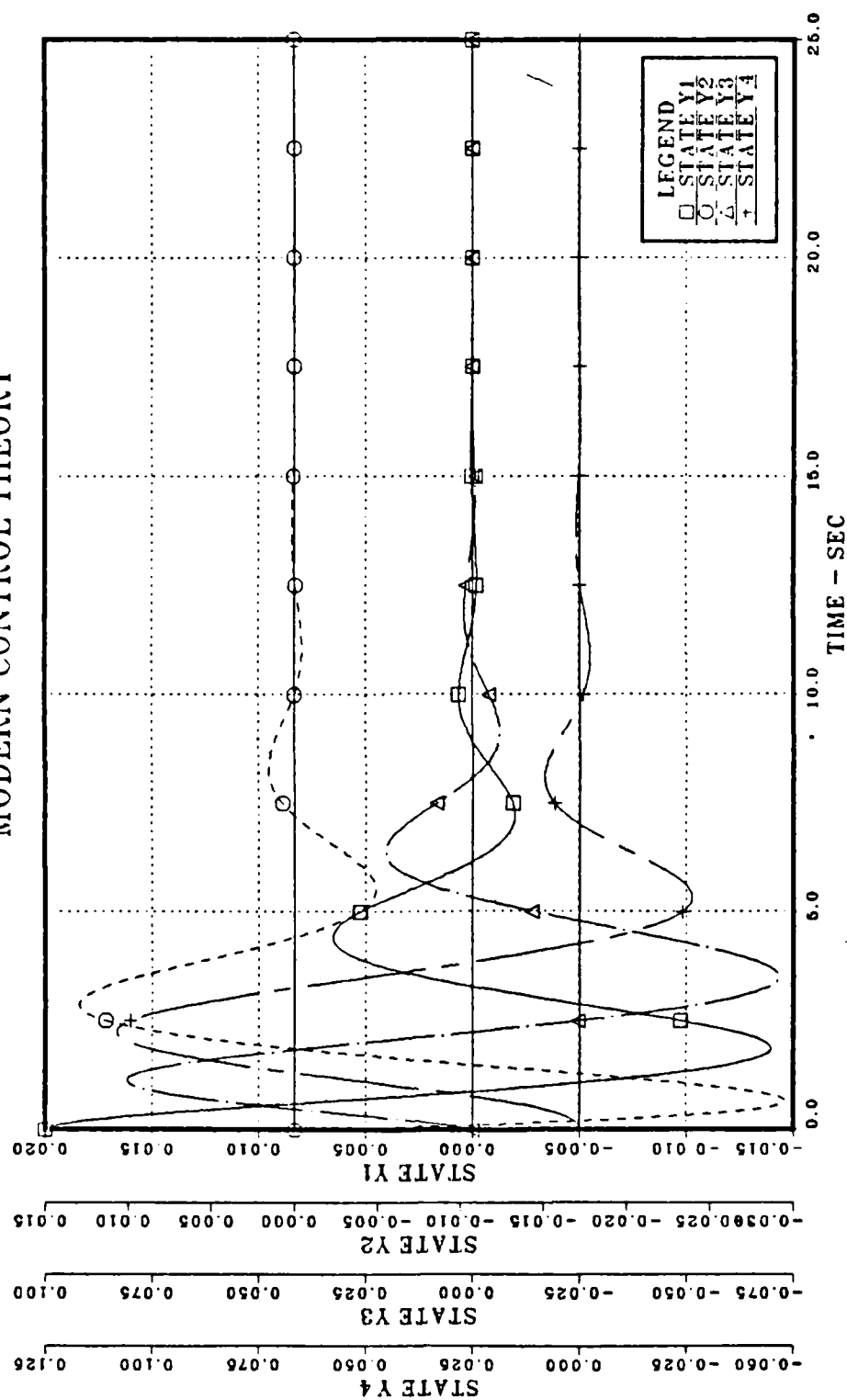


Figure 3.2 Closed-loop Time Response

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 lower case letters or numbers 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 (DISPLA 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 OPTSYSX...

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:

$$\dot{X} = \{F\} * X + \{G\} * U + \{GAM\} * (W + W_0)$$

MEASUREMENT EQUATION--

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

REGULATOR PERFORMANCE INDEX--

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

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" VIA THE

METHOD DESCRIBED ON THE PREVIOUS SCREEN?

TYPE "YES" OR "NO".

n

GENERAL OPTSYSX OPTIONS:

- OPTION 1 -- SYSTEM ANALYSIS WITHOUT
CPEN-LOOP EIGENSYSTEM CALCULATIONS.
- OPTION 2 -- SYSTEM ANALYSIS WITH OPEN-LOOP
EIGENSYSTEM CALCULATIONS.
- OPTION 3 -- CPEN-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 10.0×10^{-6} 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".

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	IDRUG
0	0	2	0	0	0	0	0	0	0	0

ISST	IDSTAE	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

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

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

0.0	0.1000E+01
0.0	-0.4600E+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..

0.1000E+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..

0.1000E+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..
0.10C0D+01

    EIGENSYSTEM OF OPTIMAL REGULATOR.....
    C-LOCF OPTIMAL REG. E-VALUES...DET(SI-F+G*C) ..
-1.71206D-01:-4.59681D+00:

    C-LOCF RIGHT EIGENVECTOR MATRIX.....M....
-9.856588D-01 -2.125703D-01
-1.687503D-01 9.771458D-01

    CCONTROL EIGENVECTOR MATRIX.....C*M..
-9.496319D-01 3.957155D-03

    C-LCCP 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
-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" 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" CR "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-IOCP SUBOPT. EST. E-VALUES...DET(SI-F+K*H)...

-5.00000D+01, 4.99984D+01:

C-LCCP 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-IOCP 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
 $XDCT = \{F\} * X + \{G\} * UC$
2. CLOSED LOOP TIME RESPONSE
 $XDCT = \{F - G * C\} * X + \{G\} * UC, \quad U = \{C\} * X$
3. OPTIMIZED FILTER CLOSED LOOP SYSTEM RESPONSE.
 $XDCT = \{F\} * X + \{G\} * UC, \quad Z = \{H\} * X$
 $XHDOT = \{F\} * XH + \{G\} * U + \{K\} * \{Z - H * XH\}$
4. FILTER + REGULATOR CLOSED LOOP SYSTEM RESPONSE.
 $XDCT = \{F + G * C\} * X + \{G\} * UC, \quad Z = \{H\} * X$
 $XHDOT = \{F\} * XH + \{G\} * U + \{K\} * \{Z - H * XH\}, \quad U = \{C\} * XH$

3
3
SELECT 1, 2, 3 OR 4.

THE (K*H) MATRIX

95.40000 0.0
4561.00000 0.0

THE CCMBINED 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 PCINTS 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	0.2689106D+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 -0.10000000D+02 C.1131009D+00 0.6197598D-01 0.1247416D+00
0.05 -0.10000000D+02 0.1133273D+00 0.1371451D-01 0.1217242D+00
0.06 -0.10000000D+02 0.1132681D+00-0.3323315D-01 0.1187650D+00
0.07 -0.10000000D+02 0.1129310D+00-0.7890278D-01 0.1161034D+00
0.07 -0.10000000D+02 0.1123237D+00-0.1233292D+00 0.1137974D+00
0.08 -0.10000000D+02 0.1114535D+00-0.1665461D+00 0.1117983D+00
0.08 -0.10000000D+02 0.1103275D+00-0.2085866D+00 0.1100069D+00
0.09 -0.10000000D+02 0.1089528D+00-0.2494827D+00 0.1083096D+00
0.10 -0.10000000D+02 0.1073360D+00-0.2892654D+00 0.1066016D+00
0.10 -0.10000000D+02 0.1054837D+00-0.3279652D+00 0.1047973D+00
0.11 -0.10000000D+02 0.1034025D+00-0.3656114D+00 0.1028338D+00
0.11 -0.10000000D+02 0.1010984D+00-0.4022329D+00 0.1006703D+00
0.12 -0.10000000D+02 0.9857769D-01-0.4378574D+00 0.9828370D-01
0.13 -0.10000000D+02 0.9584610D-01-0.4725121D+00 0.9566496D-01
0.13 -0.10000000D+02 0.9290943D-01-0.5062234D+00 0.9281438D-01
0.14 -0.10000000D+02 0.8977326D-01-0.5390170D+00 0.8973812D-01
0.14 -0.10000000D+02 0.8644301D-01-0.5709178D+00 0.8644545D-01
0.15 -0.10000000D+02 0.8292398D-01-0.6019503D+00 0.8294671D-01
0.16 -0.10000000D+02 0.7922130D-01-0.6321380D+00 0.7925213D-01
0.16 -0.10000000D+02 0.7533997D-01-0.6615038D+00 0.7537114D-01
0.17 -0.10000000D+02 0.7128485D-01-0.6900703D+00 0.7131205D-01
0.17 -0.10000000D+02 0.6706068D-01-0.7178591D+00 0.6708207D-01
0.18 -0.10000000D+02 0.6267205D-01-0.7448914D+00 0.6268741D-01
0.19 -0.10000000D+02 C.5812345D-01-0.7711879D+00 0.5813344D-01
0.19 -0.10000000D+02 0.5341923D-01-0.7967685D+00 0.5342495D-01
0.20 -0.10000000D+02 0.4856362D-01-0.8216527D+00 0.4856623D-01
0.20 -0.10000000D+02 0.4356075D-01-0.8458595D+00 0.4356131D-01
0.21 -0.10000000D+02 0.3841463D-01-0.8694073D+00 0.3841398D-01
0.22 -0.10000000D+02 0.3312915D-01-0.8923141D+00 0.3312793D-01
0.22 -0.10000000D+02 0.2770811D-01-0.9145973D+00 0.2770673D-01
0.23 -0.10000000D+02 0.2215519D-01-0.9362739D+00 0.2215392D-01
0.23 -0.10000000D+02 0.1647400D-01-0.9573604D+00 0.1647295D-01
0.24 -0.10000000D+02 0.1066802D-01-0.9778729D+00 0.1066723D-01
0.25 -0.10000000D+02 0.4740642D-02-0.9978270D+00 0.4740106D-02
0.25 -0.10000000D+02-0.1304821D-02-0.1017238D+01-0.1305149D-02
0.26 -0.10000000D+02-0.7465155D-02-0.1036120D+01-0.7465326D-02
0.26 -0.10000000D+02-0.1373724D-01-0.1054489D+01-0.1373730D-01
0.27 -0.10000000D+02-0.2011802D-01-0.1072357D+01-0.2011801D-01
0.28 -0.10000000D+02-0.2660455D-01-0.1089739D+01-0.2660450D-01
0.28 -0.10000000D+02-0.3319394D-01-0.1106648D+01-0.3319388D-01
0.29 -0.10000000D+02-0.3988340D-01-0.1123096D+01-0.3988334D-01
0.29 -0.10000000D+02-0.4667020D-01-0.1139097D+01-0.4667015D-01
0.30 -0.10000000D+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) F/O
C (121) F/O
E (122) F/O

```

... Your Fortran program is now being loaded ...
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., X_1 , X_2 , ETC)
2. FEEDBACK CONTROL (IE., $U = -C \cdot X$)
3. CONTROL INPUT (IE., U_1 , U_2 , ETC.)
4. STATE ESTIMATE (OBSERVER) (IE., $\hat{X}_1$, $\hat{X}_2$, ETC.)
5. STATE RECONSTRUCTION ERROR (IE., $X_1 - \hat{X}_1$,
 $X_2 - \hat{X}_2$, 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
 {C} => GAMMA
 {D} => DELTA
 {E} => EPSILON
 {F} => PHI
 {G} => CHI
 {H} => THETA

angular position - {X} 1

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

1. STATE VARIABLE (IE., X_1 , X_2 , ETC)
2. FEEDBACK CONTROL (IE., $U = -C \cdot X$)
3. CONTROL INPUT (IE., U_1 , U_2 , ETC.)
4. STATE ESTIMATE (OBSERVER) (IE., $\hat{X}_1$, $\hat{X}_2$, ETC.)
5. STATE RECONSTRUCTION ERROR (IE., $X_1 - \hat{X}_1$,
 $X_2 - \hat{X}_2$, 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
{E} => 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
{E} => 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
{E} => BETA
{F} => PHI
{Q} => THETA

linear optimal control systems
>> USING A PFE-ALLOCATED DATASET FOR UNIT FT17F001.

THE FOLLOWING OPTIONS ARE AVAILABLE.

1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
2. REPICT 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. PLOT 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. REFLECT 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. REFLECT 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?
4

THE FOLLOWING OPTIONS ARE AVAILABLE.

1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
2. REFLECT 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?
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 DISSELA 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

3?
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 DISPLA 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 DISPLA METAFILE THAT YOU JUST CREATED?
(Y OR N)

Y
B (120) R/C
DASD 001 LINKED R/O; R/W BY MVS
Z (001) R/C - 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 RECCFDING OF TERMINAL SESSION

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

FILTER ONLY CLOSED LOOP
EXAMPLE 4.1
LINEAR OPTIMAL CONTROL SYSTEMS

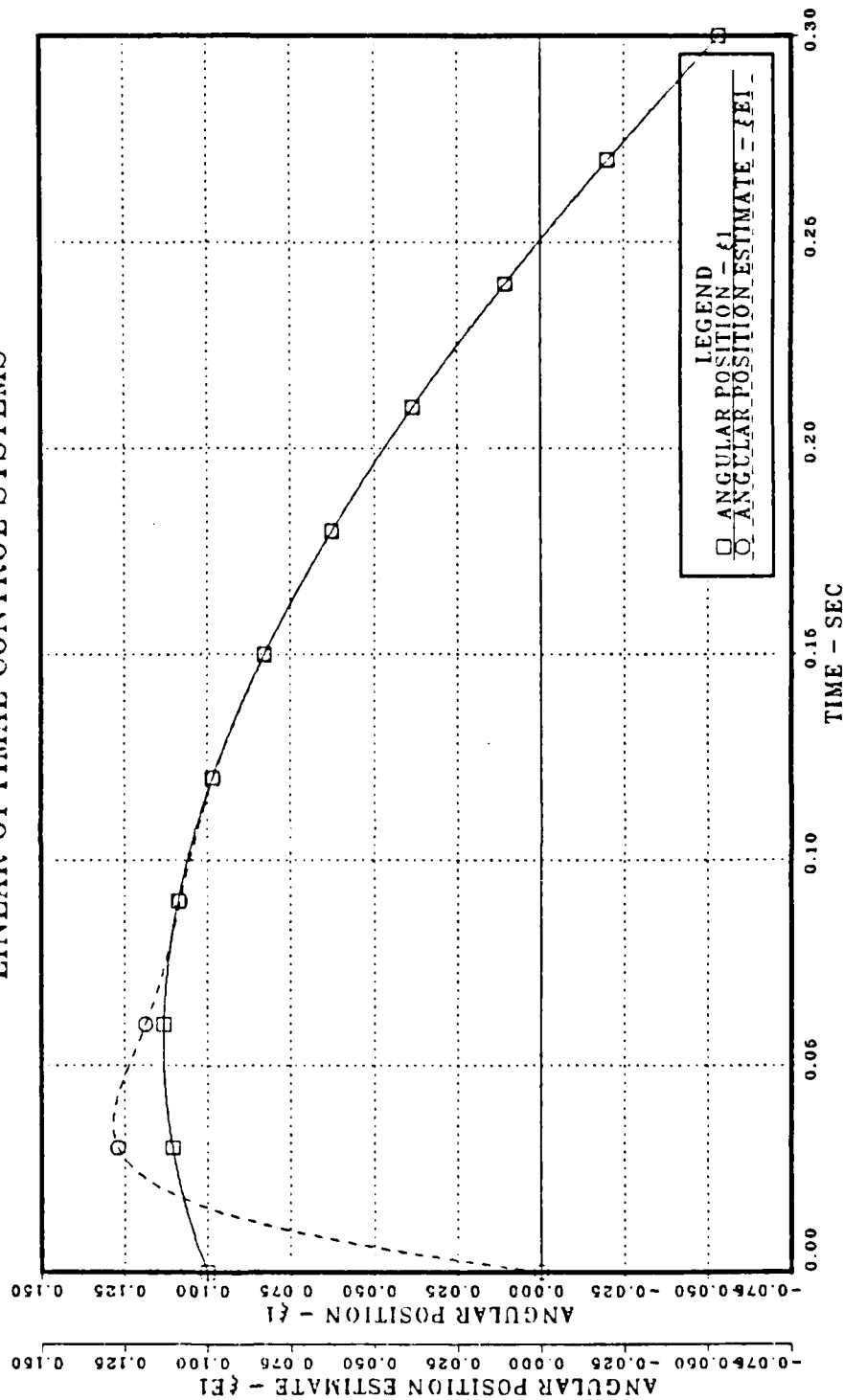


Figure 3.3 Filter Closed-loop Time Response

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 FCRTRAN (CALCULATE TIME RESPONSE)
3. OPTPLOT FCRTRAN (DISPLA 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 CCNTAINS 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:

$$\dot{X} = \{F\} * X + \{G\} * U + \{GAM\} * (W + W_0)$$

MEASUREMENT EQUATION--

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

REGULATOR PERFORMANCE INDEX--

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

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 -- LOCAL 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.

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

COMPENSATOR TRANSFER FUNCTION OPTIONS:

OPTICN 1 -- NO COMP. TRANSFER FUNCTIONS COMPUTED.

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

OPTICN 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.}

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

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

n
TYPE "YES" OR "NO".

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 10.0×10^{-6} IS CONSIDERED ZERO.

THE DEFAULT VALUE OF THIS PARAMETER {IE} IS 6.
IN OTHER WORDS, $IE = 1.0 \times 10^{-6}$.

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

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

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.}

n
TYPE "YES" OR "NO".

DO YOU DESIRE TO PRINT THE EULER-LAGRANGE EIGENSYSTEM
FOR TO DECOMPOSITION {FOR CHECKING THE PROGRAM}?

n
TYPE "YES" OR "NO".

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	IDDEBUG
0	0	3	0	0	0	0	0	0	0	0
ISSET	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?

n TYPE "YES" OR "NO".

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

0.0 0.1000D+01
0.0 -0.4600E+01

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

1.00000 0.0

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

n TYPE "YES" CR "NO".

MEASUREMENT SCALING MATRIX.....H..

0.1000E+01 0.0

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

0.0
0.78700

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

n TYPE "YES" CR "NO".

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?

n TYPE "YES" OR "NO".

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

0.0
0.7870D+00

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

-2.5410D+02 -1.9570E+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 FIGHT EIGENVECTOR MATRIX.....M.....
 -5.000980D-02 -4.999602D-02
 1.000000D+00 0.0

CCNTFCL EIGENVECTOR 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
 0.10000D+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" CR "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" CR "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)..

```

-5.00000E+01, 4.99984E+01:

C-ICOF RIGHT EIGENVECTOR MATRIX.....M....

9.953957D-03 -1.096216D-02
1.00000CD+00 0.0

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

9.953957D-03 -1.096216D-02

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

0.0 1.00000CD+00
-9.122292D+C1 9.080291D-01

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

7.150503D-06 2.271000D-04
2.271000D-C4 1.181151D-02

RMS VALUES OF THE ESTIMATION ERROR.....

2.674042D-03 1.0868C8D-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 CPTSYSX:

1. OPEN LOOP TIME RESPONSE
 $\dot{X} = \{F\} * X + \{G\} * UC$
2. CLOSED LOOP TIME RESPONSE
 $\dot{X} = \{F - G * C\} * X + \{G\} * UC, \quad U = \{C\} * X$
3. OPTIMIZED FILTER CLOSED LOOP SYSTEM RESPONSE.
 $\dot{X} = \{F\} * X + \{G\} * UC, \quad Z = \{H\} * X$
 $\dot{X}_H = \{F\} * X_H + \{G\} * U + \{K\} * \{Z - H * X_H\}$
4. FILTER + REGULATOR CLOSED LOOP SYSTEM RESPONSE.
 $\dot{X} = \{F + G * C\} * X + \{G\} * UC, \quad Z = \{H\} * X$
 $\dot{X}_H = \{F\} * X_H + \{G\} * U + \{K\} * \{Z - H * X_H\}, \quad U = \{C\} * X_H$

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.

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	0.4310795D-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	-0.4330414D-03	0.3030484D-01	-0.4330414D-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	-0.6765634D-03	0.4125983D-02	-0.6765634D-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	-0.5453529D-03	-0.4489304D-02	-0.5453529D-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) E/O
C (121) E/O
E (122) E/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., $U = -C \cdot X$)
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

AD-A152 221

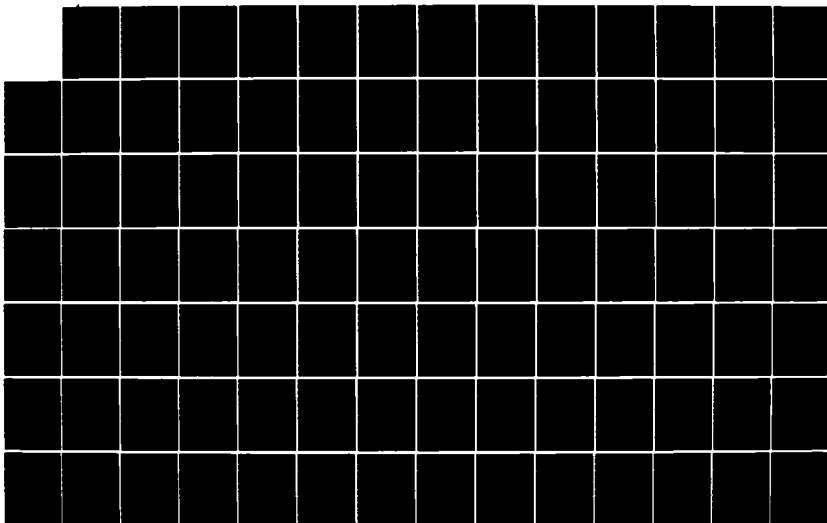
DEVELOPMENT OF GRAPHICAL TIME RESPONSE USING THE
OPTSYSX PROGRAM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY
CA H A DIEL SEP 84

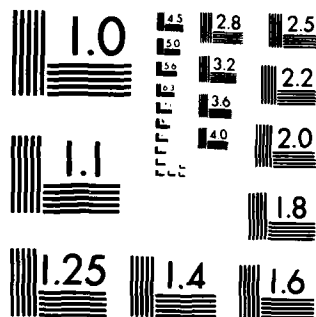
2/3

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

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
(E) => BETA
(F) => PHI
(C) => 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., $\dot{u} = -C \cdot X$)
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
(O) => 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., $\dot{u} = -C \cdot X$)
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
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
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.

3
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

2
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

3
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 POSITION OF THE LEGEND.
10. EDITING COMPLETE.

SELECT A NUMBER BETWEEN 1 AND 10.

3
5

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

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

2
1

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

2
-.025

WHAT IS THE NEW Y-MAX VALUE?

2
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.

3
10

THE FOLLOWING OPTIONS ARE AVAILABLE.

1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
2. REFLECT 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
4

THE FOLLOWING OPTIONS ARE AVAILABLE.

1. BEGIN NEW GRAPH OF OTHER CONTROLS, STATES, OR ESTIMATES.
2. REFLECT 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
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 DISSELA 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 DISPLA 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 DISPLA 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

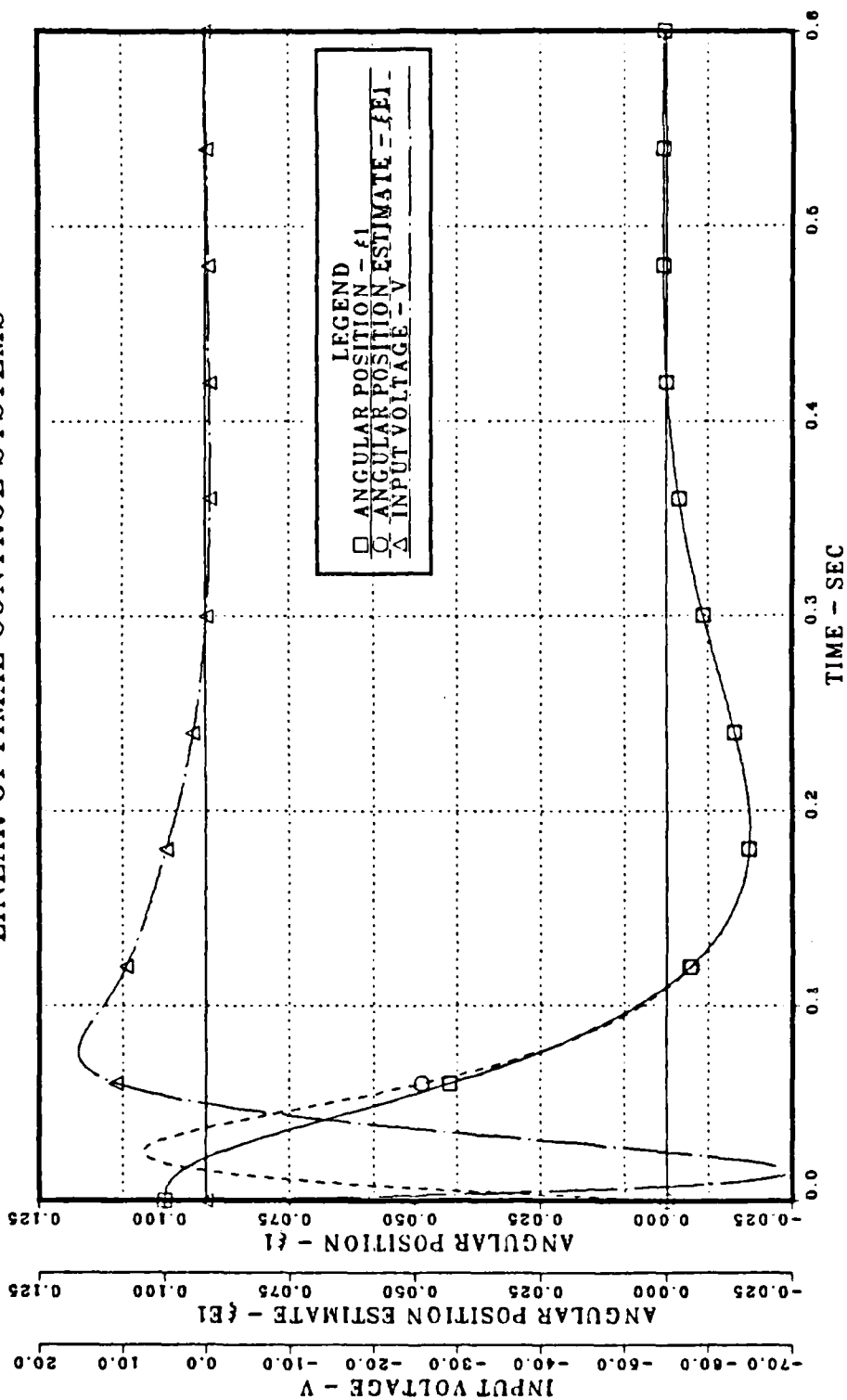


Figure 3.4 Filter plus Regulator Closed-loop Time Response

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 constraints 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.

X-29A LONGITUDINAL SYSTEM 82 X 82 MATRIX

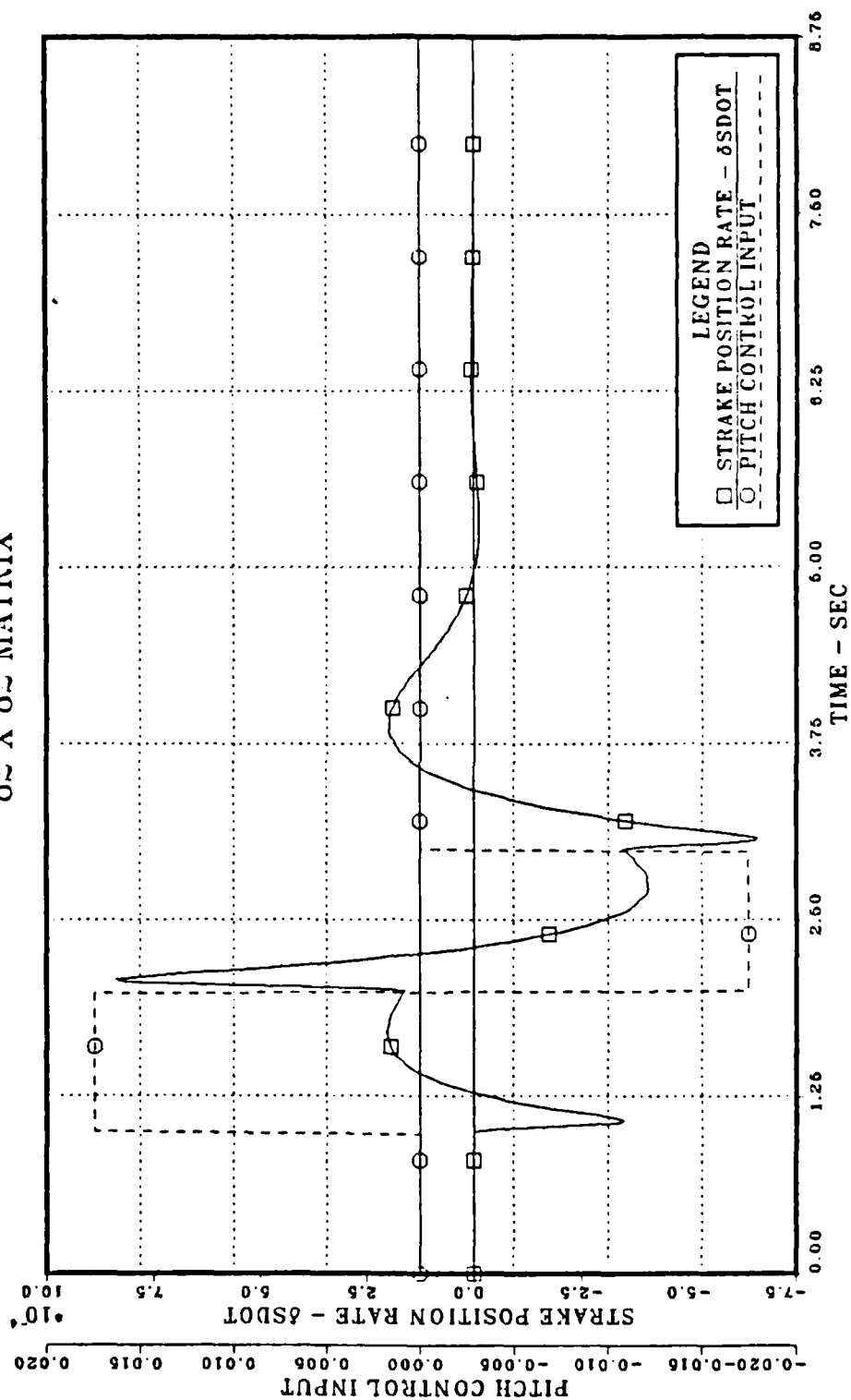


Figure 4.1 X-29A Longitudinal Time Response

X-29A LONGITUDINAL SYSTEM 82 X 82 MATRIX

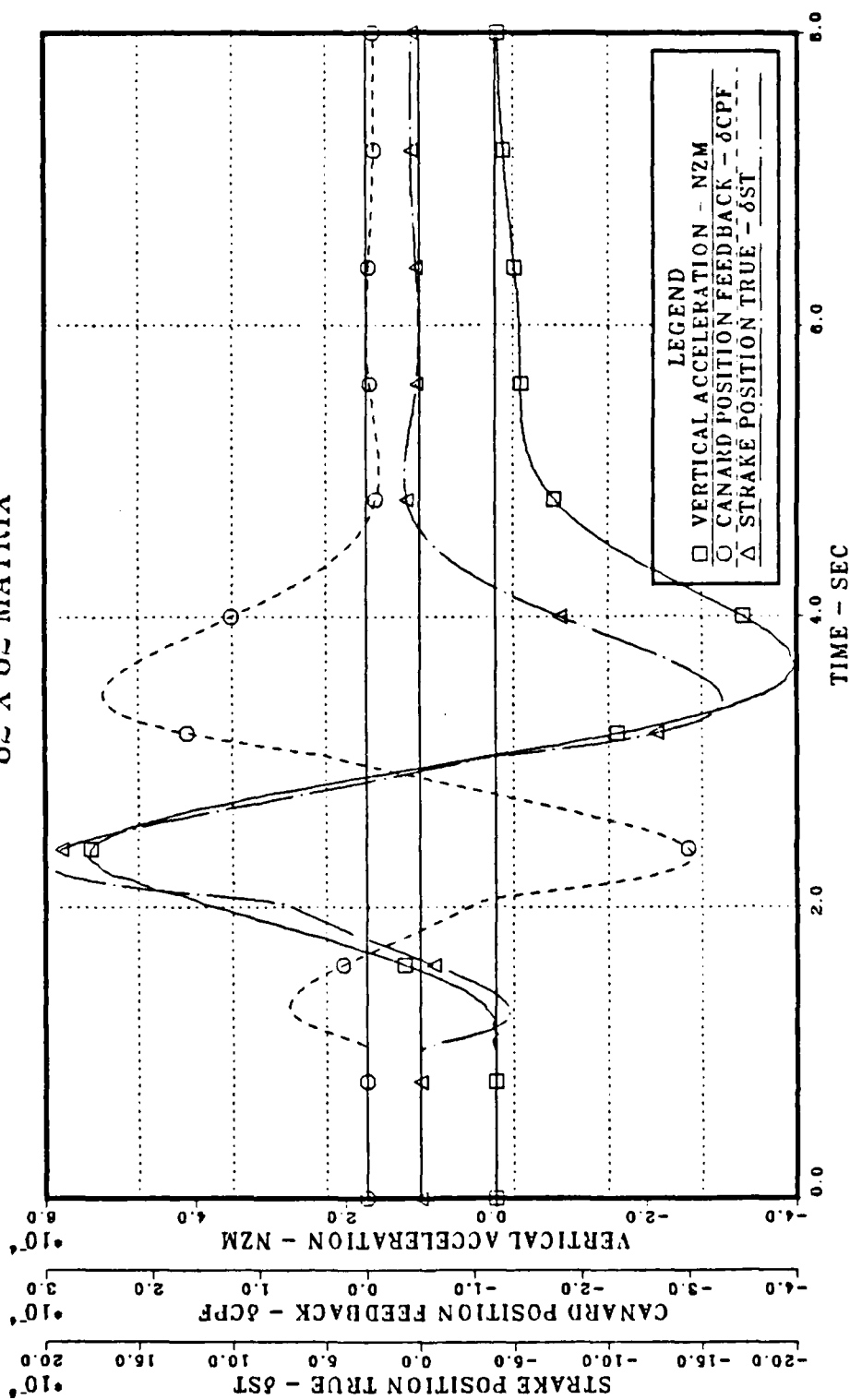


Figure 4.2 X-29A Longitudinal Time Response

X-29A LONGITUDINAL SYSTEM 82 X 82 MATRIX

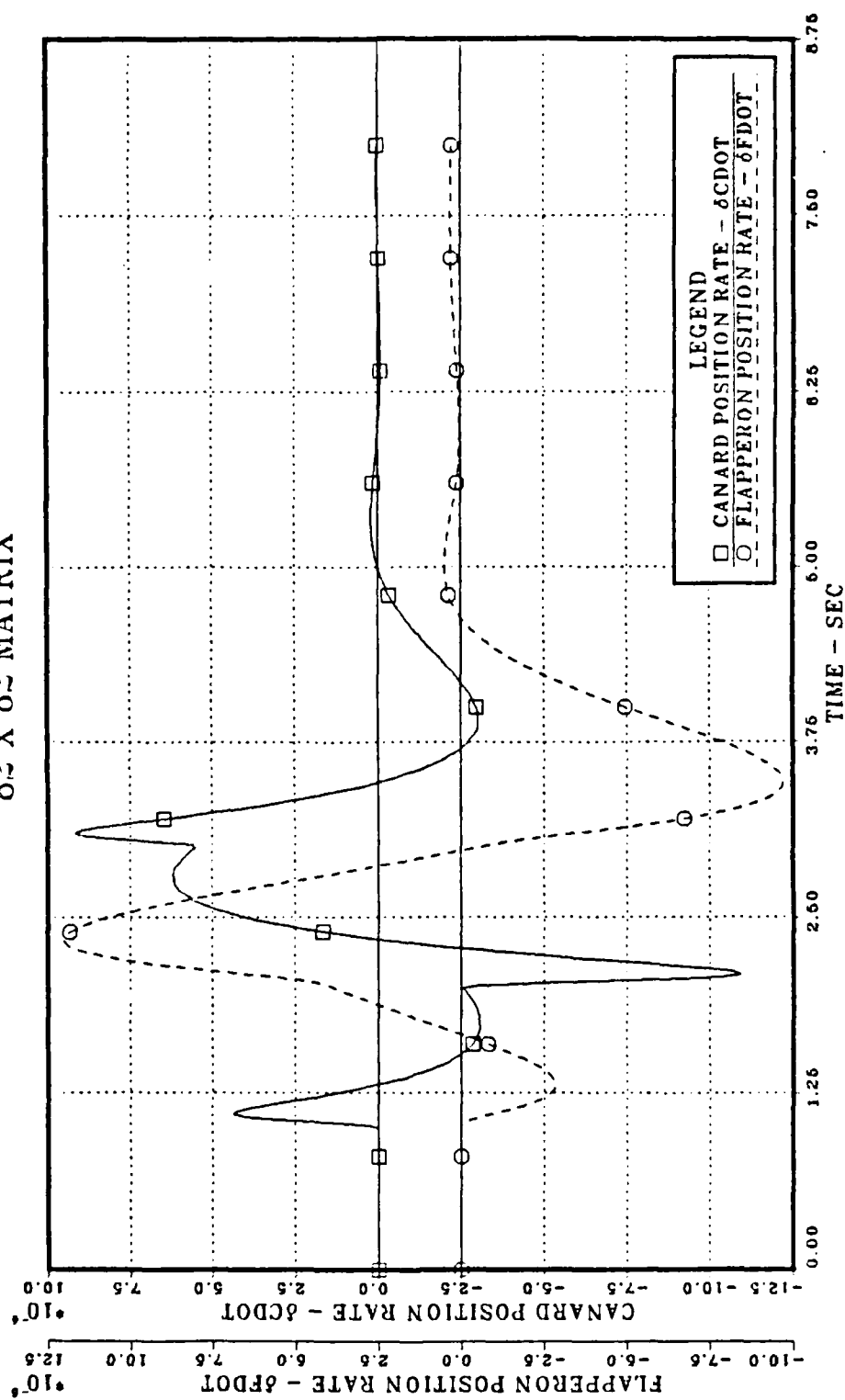


Figure 4.3 X-29A Longitudinal Time Response

X-29A LONGITUDINAL SYSTEM 82 X 82 MATRIX

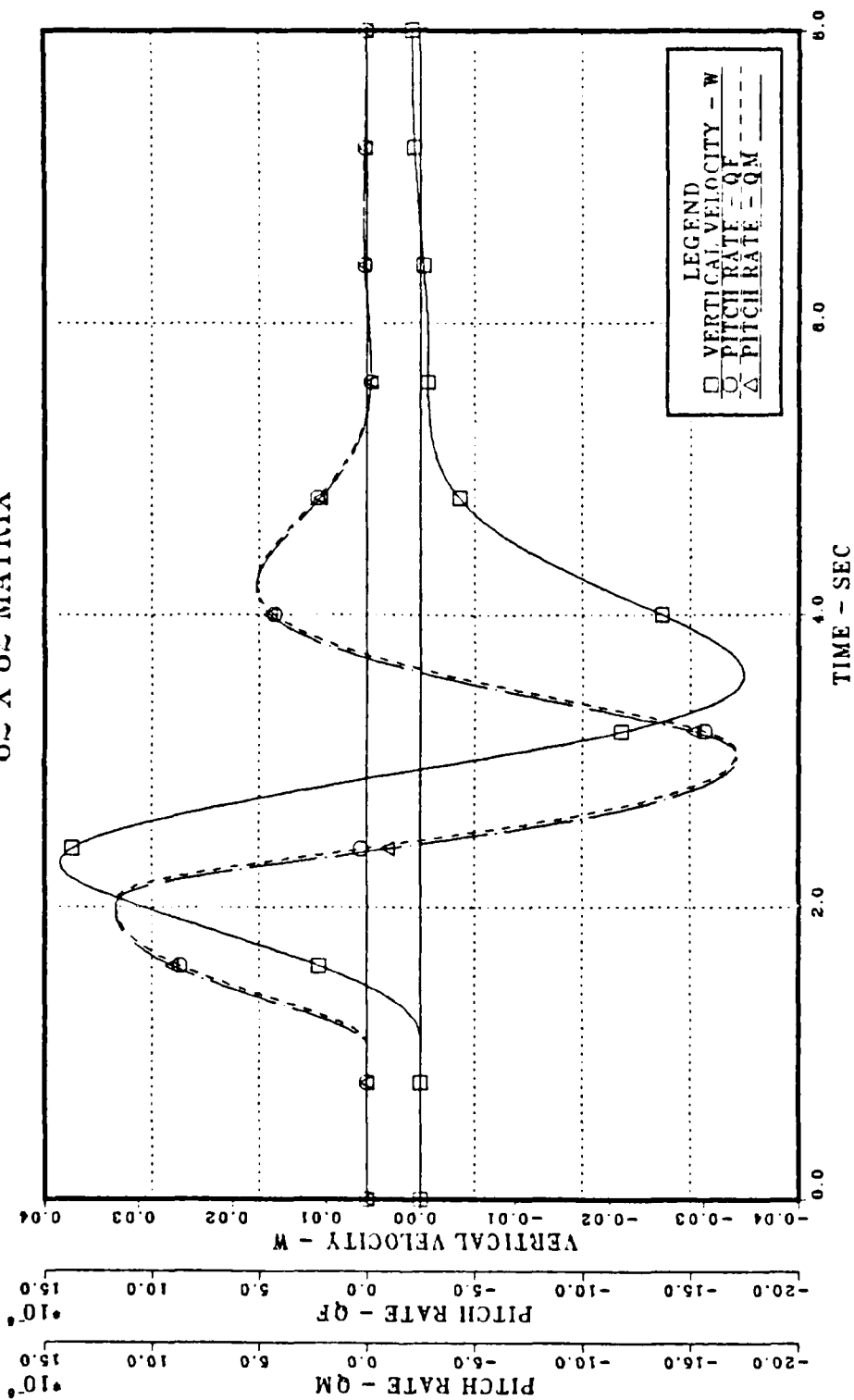


Figure 4.4 X-29A Longitudinal Time Response

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 (N_c) is given as zero, but this is not a viable solution for systems which use a driving function.

APPENDIX A. OPTSYS EXEC LISTING

```

STRACE OFF
*****
*
*               THE OPTSYS EXEC
*
*   CONTROLS THE OPTSYSX, OPTCALC AND OPTPLOT
*   TC 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 &GOTC -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 {DISPLA 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 "OETMAT 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 &GOTC -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 CPTSYSX TO GO
TO YOUR TERMINAL S(CREEN) OR TO A D(ISK) FILE?
(S OR D)

```

-ENDFCUR
&READ VARS &ANS
&IF .&ANS EQ .S &GOTC -FOURTH

```

```

&IF .&ANS EQ .D &GOTC -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
&GOTC -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 "
* BEFCRE LOADING OPTCALC
*

FINDSTAK OPTMAT DATA A1 C01 0 LIM002 ALL GROUP1
\$READ VARS \$A1 \$A2
\$IF . \$A2 EQ .0 \$EXIT \$RC
\$TYPE ICADING OPTCALC...
FILEDEF 8 DISK OPTPLCT DATA A1 (PERM
FILEDEF 9 DISK OPTMAT DATA A1 (PERM
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLDP NONIMSL
LOAD CPTCALC (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 OPTPLCT DATA A1 (PERM
EXEC DISSPLA OPTPLOT
-EIGHTH

*
* CHECK FOR FILE "DISSPLA METAFIIE A4" ON
* THE USER'S DISK BEFORE GOING TO DISSPOP
*

RENAME DISSPLA METAFIIE A4 OPTSYS METAFIIE A4
\$IF \$RC NE 0 \$GOTO -TENTH
RENAME OPTSYS METAFIIE A4 DISSPLA METAFIIE 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?
(TC 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!!

```
-ENDGCODEY
CP SLEEP 3 SEC
CLRSCRN
&EXIT &RC
```



```

C-----
1UG,ISET,IREG,IPSD,IYU,INORM
C-----
C  DATA IV,Y',IZ,N'
C-----
C  SUPPRESS INDIVIDUAL UNDERFLOW, OVERFLOW, DIVIDE CHECK, AND DECIMAL
C  CONVERT ERROR MESSAGES; PROVIDE SUMMARY OF ERRORS ONLY.
C-----
C  CALL ERFSET (207,256,-1,1,1,209)
C  CALL ERFSET (215,256,-1,1)
C-----
C  INITIALIZE SAVE FLAGS.
C-----
ISAF=0
ISAG=0
ISAH=0
ISAM=0
ISAA=0
ISAB=0
ISET=0
C-----

C-----SCRN1-----
10  CALL FRICMS ('CLRSCRN ')
    WRITE (5,64C)
    CALL RCCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 20
    GO TO 30
20  WRITE (5,10C)
    GO TO 10
30  CONTINUE
    IF (IANS.EQ.IZ) GO TO 630
C-----SCRN2-----
40  CALL FRICMS ('CLRSCRN ')
    WRITE (5,65C)
    CALL RCCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
    GO TO 40
50  WRITE (5,10C)
    GO TO 40
60  CONTINUE
    IF (IANS.EQ.IZ) GO TO 630
C-----ISET-----
70  CALL FRICMS ('CLRSCRN ')
    WRITE (5,66C)
    CALL RCCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 80
    GO TO 50
80  WRITE (5,10C)
    GO TO 70
90  CONTINUE

```

```
IF (IANS.EQ.IY) ISET=1
```

```
C-----  
C INITIALIZE SYSTEM FLAGS.  
C-----
```

```
100 CONTINUE
```

```
IR=0
```

```
IOL=0
```

```
IQ=0
```

```
IR=0
```

```
ISS=0
```

```
IM=0
```

```
ITF1=0
```

```
ITF2=0
```

```
ITF3=0
```

```
IFCFW=0
```

```
IE=0
```

```
IDSTAB=C
```

```
IDBUG=0
```

```
IPSC=0
```

```
IYLC=0
```

```
INCRM=0
```

```
IREG=0
```

```
NS=0
```

```
NC=0
```

```
NUB=0
```

```
NG=0
```

```
IRCMAT=C
```

```
C-----  
CALL FRICMS ('CLRSCRN ')
```

```
WRITE (5,67C)
```

```
CALL RCINT (IANS)
```

```
IOL=IANS-1
```

```
IF (IOL.EQ.2) GO TC 350
```

```
C-----  
CALL FRICMS ('CLRSCRN ')
```

```
WRITE (5,68C)
```

```
CALL RCCHAR (IANS)
```

```
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 12C
```

```
GO TO 120
```

```
WRITE (5,100C)
```

```
GO TO 11C
```

```
CONTINUE
```

```
IF (IANS.EQ.IY) IQ=1
```

```
IF (IANS.EQ.IZ) IQ=0
```

```
IF (ICL.EQ.3) GO TC 200
```

```
C-----  
CALL FRICMS ('CLRSCRN ')
```

```
WRITE (5,69C)
```

```

CALL RCINT (IANS)
IR=IANS-1
C-----ISS-----
140 CALL FRICMS ('CLRSCRN ')
    WRITE (5,70C)
    CALL RCCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 15C
    GO TO 16C
150 WRITE (5,10C0)
    GO TO 140
160 CONTINUE
    IF (IANS.EQ.IY) ISS=1
    IF (IANS.EC.IZ) ISS=0
C-----JM-----
170 WRITE (5,71C)
    CALL RCCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 18C
    GO TO 190
180 WRITE (5,10C0)
    GO TO 170
190 CONTINUE
    IF (IANS.EQ.IY) IM=1
    IF (IANS.EC.IZ) IM=0
200 CONTINUE
    IF (IOL.EQ.3) IM=1
C-----ITF1-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,72C)
CALL RCINT (IANS)
ITF1=IANS-1
IF (IOL.EQ.3) GO TC 240
C-----ITF2-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,73C)
CALL RCINT (IANS)
ITF2=IANS-1
IF (IOL.EQ.3) GO TC 240
C-----ITF3-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,74C)
CALL RCINT (IANS)
ITF3=IANS-1
C-----IFDFW-----
210 CALL FRICMS ('CLRSCRN ')
    WRITE (5,75C)
    CALL RCCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 22C
    GO TO 23C

```

```

220 WRITE (5,1000)
230 GO TO 210
CONTINUE
IF (IANS.EQ.IY) IFCF=1
IF (IANS.EC.IZ) IFCF=0
C-----IE-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,760)
CALL RICEAL (ANSR)
IE=IDINT(ANSR)
IF (IOL.EC.3) GO TC 300
C-----IDSTAB-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,770)
CALL RICEAL (IANS)
IF (IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 250
GO TO 260
WRITE (5,1000)
250 GO TO 240
CONTINUE
IF (IANS.EQ.IY) IDSTAB=1
IF (IANS.EC.IZ) IDSTAB=0
C-----IDEBUG-----
WRITE (5,780)
CALL RICEAL (IANS)
IF (IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 280
GO TO 290
WRITE (5,1000)
GO TO 270
CONTINUE
IF (IANS.EQ.IY) IDEBUG=1
IF (IANS.EC.IZ) IDEBUG=0
300 CONTINUE
C-----IPSD-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,790)
CALL RICEAL (IANS)
IPSC=IANS
IF (IPSC.EQ.3) IPSD=0
IF (IPSC.EQ.0) GO TC 310
C-----IYU-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,800)
CALL RICEAL (IANS)
IYU=IANS-1
C-----INORM-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,820)

```

```

310 CALL RCFEAL (ANSR)
    INCRM=IDINT(ANSR)
    IF ((ICL.EC.3) GO TO 350
----- IREG-----
320 CALL FRICMS ('CLRSCRN ')
    WRITE (5,810)
    CALL RCFEAL (ANSR)
    IF ((IANS.NE.1V).AND.(IANS.NE.1Z)) GO TO 330
    GO TO 340
330 WRITE (5,1000)
    GO TO 320
340 CONTINUE
    IF (IANS.EQ.1V) IREG=1
    IF (IANS.EQ.1Z) IREG=0
350 CALL RCFEAL (ANSR)
    IF ((ISAF.EC.1).AND.(IRDMAT.EQ.1)) GO TO 360
----- NS-----
    CALL FRICMS ('CLRSCRN ')
    WRITE (5,820)
    CALL RCFEAL (ANSR)
    NS=IDINT(ANSR)
    IF ((ICL.EC.2) GO TO 390
360 IF ((ISAG.EC.1).AND.(IRDMAT.EQ.1)) GO TO 370
----- NC-----
    WRITE (5,830)
    CALL RCFEAL (ANSR)
    NC=IDINT(ANSR)
    IF ((ISAH.EC.1).AND.(IRDMAT.EQ.1)) GO TO 380
----- NOB-----
    WRITE (5,840)
    CALL RCFEAL (ANSR)
    NOB=IDINT(ANSR)
    IF ((IGAM.EC.1).AND.(IRDMAT.EQ.1)) GO TO 390
----- NG-----
    WRITE (5,850)
    CALL RCFEAL (ANSR)
    NG=IDINT(ANSR)
    CONTINUE
390 ----- FLAG SETTINGS-----
    CALL FRICMS ('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) IPSD,IYU,INORM,IREG,NS,NC,NOB,NG
    WRITE (6,900) NS,NC,NOB,NG
----- BEGIN CALCULATIONS-----

```

```

N2=2*NS
CALL INNER (NS,NC,NCB,NG,N2,ACL,B,BA,C,I,CR,C,C,Chl,CWR,D,FBGC,FBGE,
1G,GAM,GP,GN,HO,DI,C2,PRO,RM,RC,C,SC,WK,WI,W21,X,WNGRM,WNGRMI,D
2ESTAB,AA,BM,CM,JCF,RES,AY,BB,C,C,CP,CM,GV,HV,HU,USFORE,ISAF,ISAH,IS
3AG,IGAP,IRET,PRIT,NROW,NCOL,IRDMAT,ISAA,ISAE)
C-----
IF (IRET.EQ.1) GO TC 400
CALL WRINAT (BA,G,FC,GAM,FBGC,FBGE,AY,B,NS,NC,NOB,NG)
C-----
400 WRITE (5,53C)
CALL RICHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 410
410 GO TO 42C
WRITE (5,10C0)
420 GO TO 4C
CONTINUE
IF (IANS.EQ.IY) GO TO 430
IF (IANS.EQ.IZ) GO TO 630
C-----
430 CONTINUE
IF (IRET.EQ.1) GO TC 100
IF (ISET.EQ.1) GC TC 100
CALL FRICMS (CLSCRN *)
440 WRITE (5,94C)
CALL RICHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 45C
450 GO TO 46C
WRITE (5,10C0)
460 GO TO 44C
CONTINUE
IF (IANS.EQ.IY) ISAF=1
IF (IANS.EQ.IZ) ISAF=0
C-----
470 IF (NOB.EQ.0) GO TC 500
CALL FRICMS (CLSCRN *)
WRITE (5,55C)
CALL RICHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 48C
480 GO TO 49C
WRITE (5,10C0)
490 GO TO 47C
CONTINUE
IF (IANS.EQ.IY) ISAF=1
IF (IANS.EQ.IZ) ISAF=0
500 CONTINUE
C-----
IF (INC.EQ.0) GO TO 540
CALL FRICMS (CLSCRN *)

```

```

510 WRITE (5,56C)
    CALL RLCFAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 520
520 GO TO 57C
    WRITE (5,10C0)
    GO TO 51C
530 CONTINUE
    IF (IANS.EQ.IY) ISAG=1
    IF (IANS.EQ.IZ) ISAG=0
540 CONTINUE
    -----IGAM-----
    IF (NG.EQ.0) GO TO 580
    CALL FRICMS ('CLRSCRN ')
550 WRITE (5,57C)
    CALL RLCFAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TC 56C
560 GO TO 57C
    WRITE (5,10C0)
    GO TO 55C
570 CONTINUE
    IF (IANS.EQ.IY) IGAM=1
    IF (IANS.EQ.IZ) IGAM=0
580 CONTINUE
    -----ISAA-----
    CALL FRICMS ('CLRSCRN ')
590 WRITE (5,58C)
    CALL RLCFAR (IANS)
    IF ((IANS.EQ.IY).OR.(IANS.EQ.IZ)) GC TC 600
    WRITE (5,10C0)
    GO TO 59C
600 CONTINUE
    IF (IANS.EQ.IY) ISAA=1
    IF (IANS.EQ.IZ) ISAA=0
    -----ISAB-----
    CALL FRICMS ('CLRSCRN ')
610 WRITE (5,55C)
    CALL RLCFAR (IANS)
    IF ((IANS.EQ.IY).OR.(IANS.EQ.IZ)) GC TC 620
    WRITE (5,10C0)
    GO TO 61C
620 CONTINUE
    IF (IANS.EQ.IY) ISAB=1
    IF (IANS.EQ.IZ) ISAB=0
    GO TO 1C0
    -----TERMINATE-----
630 WRITE (5,1010)
    STCP
    -----

```



```

730 2GN 2 -- PCLES, RESIDUES, AND ZEROS COMPUTED.//,10X,42HCPTICN 3 --
3 ONLY FCLES AND ZERCS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PULES A
4ND RESIDUES COMPUTED.//,10X,32HSESELECT AN OPTION: 1, 2, 3, CR 4.)
740 1 1 -- NOISE TRANSFER FUNCTION COMPUTED.//,10X,49HCPTICN 2 --
2 PULES AND RESIDUES AND ZEROS COMPUTED.//,10X,48HCPTICN 3 -- ONLY PG
3UES AND ZERCS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PULES AND RESID
4UES COMPUTED.//,10X,32HSESELECT AN OPTION: 1, 2, 3, CR 4.)
1 OPTION 1 -- NO COMP. TRANSFER FUNCTIONS COMPUTED.//,10X,48HCPTICN 3 -- C
2 2 -- FCLES, RESIDUES, AND ZERCS COMPUTED.//,10X,42HCPTICN 4 -- C
3 ONLY PCLES AND ZEROS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PCLES AND
4 X RESIDUES COMPUTED.//,10X,45HCPTICN 4 -- ONLY PCLES AND
5GN CAN BE.//,10X,45HCPTICN 4 -- ONLY PCLES AND
6 FILTER.//,10X,45HCPTICN 4 -- ONLY PCLES AND
7 OPTION 1, 2, 3, CR 4.)
750 1 1 -- NOISE TRANSFER FUNCTION COMPUTED.//,10X,42HCPTICN 3 --
3 ONLY FCLES AND ZERCS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PULES A
4ND RESIDUES COMPUTED.//,10X,32HSESELECT AN OPTION: 1, 2, 3, CR 4.)
760 1 1 -- NOISE TRANSFER FUNCTION COMPUTED.//,10X,49HCPTICN 2 --
2 PULES AND RESIDUES AND ZEROS COMPUTED.//,10X,48HCPTICN 3 -- ONLY PG
3UES AND ZERCS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PULES AND RESID
4UES COMPUTED.//,10X,32HSESELECT AN OPTION: 1, 2, 3, CR 4.)
1 OPTION 1 -- NO COMP. TRANSFER FUNCTIONS COMPUTED.//,10X,48HCPTICN 3 -- C
2 2 -- FCLES, RESIDUES, AND ZERCS COMPUTED.//,10X,42HCPTICN 4 -- C
3 ONLY PCLES AND ZEROS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PCLES AND
4 X RESIDUES COMPUTED.//,10X,45HCPTICN 4 -- ONLY PCLES AND
5GN CAN BE.//,10X,45HCPTICN 4 -- ONLY PCLES AND
6 FILTER.//,10X,45HCPTICN 4 -- ONLY PCLES AND
7 OPTION 1, 2, 3, CR 4.)
770 1 1 -- NOISE TRANSFER FUNCTION COMPUTED.//,10X,42HCPTICN 3 --
3 ONLY FCLES AND ZERCS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PULES A
4ND RESIDUES COMPUTED.//,10X,32HSESELECT AN OPTION: 1, 2, 3, CR 4.)
780 1 1 -- NOISE TRANSFER FUNCTION COMPUTED.//,10X,49HCPTICN 2 --
2 PULES AND RESIDUES AND ZEROS COMPUTED.//,10X,48HCPTICN 3 -- ONLY PG
3UES AND ZERCS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PULES AND RESID
4UES COMPUTED.//,10X,32HSESELECT AN OPTION: 1, 2, 3, CR 4.)
790 1 1 -- NOISE TRANSFER FUNCTION COMPUTED.//,10X,42HCPTICN 3 --
3 ONLY FCLES AND ZERCS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PULES A
4ND RESIDUES COMPUTED.//,10X,32HSESELECT AN OPTION: 1, 2, 3, CR 4.)
800 1 1 -- NOISE TRANSFER FUNCTION COMPUTED.//,10X,49HCPTICN 2 --
2 PULES AND RESIDUES AND ZEROS COMPUTED.//,10X,48HCPTICN 3 -- ONLY PG
3UES AND ZERCS COMPUTED.//,10X,45HCPTICN 4 -- ONLY PULES AND RESID
4UES COMPUTED.//,10X,32HSESELECT AN OPTION: 1, 2, 3, CR 4.)

```



```

960 4 INDIVIDUAL MATRIX ELEMENTS.//,15X,19HTYPE "YES" OR "NO".)
      FORMAT (//,5X,48HCC YOU WISH TO SAVE THE "G"-MATRIX FROM THE LAST
170 1//,5X,24HRUN TO BE USED IN THE FOLLOWING RUN?//,5X,39HNOTE: THE M
      2ATRIX WILL BE REDISPLAYED AT//,5X,24HTHE PRCPER INPUT SEQUENCE INT
      3ERVAL//,5X,40H AND YOU WILL HAVE THE OPTION CF CHANGING//,5X,27HIND
970 4 INDIVIDUAL MATRIX ELEMENTS.//,15X,19HTYPE "YES" OR "NO".)
      FORMAT (//,5X,52HCC YOU WISH TO SAVE THE "GAMMA"-MATRIX FROM THE
171 1LAST//,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?//,5X,39HNOTE: T
      2HE MATRIX WILL BE REDISPLAYED AT//,5X,24HTHE PRCPER INPUT SEQUENCE
      3INTERVAL//,5X,40H AND YOU WILL HAVE THE OPTION CF CHANGING//,5X,27
980 4 INDIVIDUAL MATRIX ELEMENTS.//,15X,19HTYPE "YES" OR "NO".)
      FORMAT (//,5X,48HCC YOU WISH TO SAVE THE "A"-MATRIX FROM THE LAST
172 1//,5X,24HRUN TO BE USED IN THE FOLLOWING RUN?//,5X,39HNOTE: THE M
      2ATRIX WILL BE REDISPLAYED AT//,5X,24HTHE PRCPER INPUT SEQUENCE INT
      3ERVAL//,5X,40H AND YOU WILL HAVE THE OPTION CF CHANGING//,5X,27HIND
990 4 INDIVIDUAL MATRIX ELEMENTS.//,15X,19HTYPE "YES" OR "NO".)
      FORMAT (//,5X,48HCC YOU WISH TO SAVE THE "E"-MATRIX FROM THE LAST
173 1//,5X,24HRUN TO BE USED IN THE FOLLOWING RUN?//,5X,39HNOTE: THE M
      2ATRIX WILL BE REDISPLAYED AT//,5X,24HTHE PRCPER INPUT SEQUENCE INT
      3ERVAL//,5X,40H AND YOU WILL HAVE THE OPTION CF CHANGING//,5X,27HIND
1000 4 INDIVIDUAL MATRIX ELEMENTS.//,15X,19HTYPE "YES" OR "NO".)
1010 4 INDIVIDUAL MATRIX ELEMENTS.//,15X,19HTYPE "YES" OR "NO".)
      FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
      END
C=====
C SUBROUTINE SETUP (EA,G,GAM,NS,NC,NG)
C=====
      IMPLICIT REAL*8(A-F,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,IFDUF,W,IE,JDSTAB,ICEB
      IUG,ISET,IPEG,IPSD,IYU,INORM
C-----
C FILE DEFINITIONS
C-----
      CALL FRICMS ('FILEDEF','03','A','DISK','X29A82')
      DATA
1
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 "CUMMY" 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.
      READ (1,50) ((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)
      CONTINUE
10

```

```

20 CONTINUE
C-----
C THESE ARE EXAMPLES OF SEVERAL POSSIBLE METHODS OF ARRAY GENERATION
C WITHIN A 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)=C.CD+00
      CONTINUE
30 CONTINUE
40 G(52,1)= 0.3620D+07
   G(77,1)=-C.1591D+02
   G(78,1)= 0.2448D+0C
   G(79,1)= 0.2448D+00
   G(81,1)= 0.1000D+00
      RETURN
C-----
50 FORMAT (5(E12.4))
      ENCL
C=====
C SUBROUTINE CHECK (EPS,AC,NG,NO,IRET)
C CHECKS THE CONSISTENCY OF REQUESTED OPTIONS.
C=====
      DOUBLE PRECISION EPS
      COMMON /PROG/ IOL,IG,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDE8
      LOG ISET,IREG,IPSD,IYU,INDRM
      10 SET MICAL ANALYSIS WHEN OL EIGENSYS CR OL TF REQUESTED
      IF (IM.EC. 1 .AND. ICL.EC. 0) IOL=1
      IF (ICL.EC. 3 .CR. ITF1.NE. 0) IM=1
C-----
      IF (INC.NE. 0 .OR. IOL.GE. 2) GO TO 10
      WRITE (5,50)
      IRET=1
      RETURN
      CONTINUE
10 C-----
C-----TRANSFER FUNCTION CHECKS-----
      IF (IE.EQ. 0) IE=6
      EPS=10.**(-IE)
C-----
      IF (ITF1.EC. 0 .OR. NC.NE. 0) GC TO 20
      WRITE (5,100)
      IRET=1
      RETURN
C-----
20 IF (ITF3.EC. 0) GC TO 30

```

```

IF (IREG .EQ. 0 .AND. (NC .NE. 0 .AND. NG .NE. 0)) GO TO 30
WRITE (5,110)
IRET=1
RETURN
CONTINUE

30-----NOISE IF-----
IF (ITF2 .EQ. 0) GC TC 40
IF (NG.NE. 0 .AND. NC .NE. 0) GC TC 40
WRITE (5,120)
IRET=1
RETURN

C-----DESTABILIZATION RESTRICTIONS-----
40 IF (IDSTAB .EQ. 0) GO TC 50
IF (NC .EQ. 0) GC TC 50
IF (NG .NE. 0) IREG=1
WRITE (5,130)
IF (IREG .EQ. 1) GC TO 50
IRET=1
RETURN
CONTINUE

50-----PSD INPUT-----
IF (IPSC .EQ. 0) GC TC 80
IF (IPSC .LT. 0 .OR. IPSD .GT. 3) GC TC 60
IF (IPLC .LT. 0 .OR. IPU .GT. 3) GO TC 60
IF (INCRM .LT. 0 .OR. INORM .GT. NG+NO) GO TO 60
GO TO 70
WRITE (5,140)
IRET=1
RETURN

70 IF (IREG .EQ. 0 .AND. NC .NE. 0) GC TO 80
WRITE (5,150)
IRET=1
RETURN
CONTINUE
RETURN

80-----
C-----
90 FORMAT (//,5X,49H H - MATRIX MUST BE INPUT, I.E. "NG" MUST BE > 0.
1//)
100 FORMAT (//,5X,46F(G) MATRIX MUST BE INPUT, I.E. NC MUST BE > 0.,//,
110X,26HC COMPUTE OPEN LOOP T. F.,//)
110 FORMAT (//,5X,48H REGULATOR AND FILTER SYNTHESIS MUST BE REQUESTED,
1//,5X,44F IN THE SAME RUN TO COMPUTE COMPENSATOR T. F.,//,5X,47HI.E.
2IREG.MUST = 0.: "NC" AND "NG" MUST BE > 0.,//)
120 FORMAT (//,5X,51H NCISE T. F. CALCULATED ONLY WHEN REGULATOR DESIGN
1ED.,//,5X,47HI.E. IREG.MUST = 1.: "NC" AND "NG" MUST BE > 0.,//)
130 FORMAT (//,5X,47H DESTABILIZATION OPTION DESIGNED FOR A REGULATOR,
1.5X,38F IF FILTER BUT NOT BOTH SIMULTANEOUSLY.,//,5X,55H IF "NG" > 0
2. THE REGULATOR OPTION IS AUTOMATICALLY SET ,//)

```

```

140  FORMAT (//,5X,49H ***** INCONSISTENT PSD INPUT FLAGS *****
150  1//)
150  1X,42H1C COMPUTE THE PSD OF A CONTROLLED SYSTEM //,10X,42H1.E. 1REG
2 MUST BE 0. AND "NC" MUST BE > 0.//)
END
C=====
C  SUPEROUTLINE INNER (NS,AC,NO,NG,A2,ACL,B,EA,C1,CR,CQ,CW1,CWK,D,FBGC,
1 FBGE,G,GAM,GM,GN,HC,D1,D2,PRO,RM,RC,C,SC,WR,W1,W2,X,WNORM,MNC
2 RMI,DESTAB,AA,BM,CM,JCF,RES,AY,BB,CC,CP,GM,CV,HV,HU,DSTCRE,ISAF,IS
3 AH,ISAG,IGAM,IREF,PRTI,NRUM,NCLL,IRCMAT,ISAF,ISAB)
C=====
C  IMPLICIT REAL*8(A-F,C-Z)
C=====
C  DIMENSION ACL(NS,NS),B(NC,NC),BA(NS,NS),CI(NS),CR(NS),CC(NS,NS),CW
1 I(NS),C(R(NS),FBGL(NC,NS),FBGE(NS,NC),G(NS,NS),GM(NS,NS),PRO(NS,NS
2 ),RC(NC,NG),SC(NS,NS),WR(N2),WI(N2),MI(NS,NS),W21(NS,NS),X(N2,N2)
3 ,GN(NS,NS),FO(NC,NS),D1(N2),D2(N2),RM(N2,N2),Q(NG,NG),D(NC,NC),GAM
4 (NS,NG),WNORM(NS,NS),WNCRMI(NS,NS),DESTAB(NS),AA(NS,NS),BM(NS,NC),
5 CM(NC,NS),JCF(N2),RES(N2),AY(NC,NC),BB(N2),CC(N2),CP(NS),GW(N2,NG)
6 ,GV(N2,NC),HY(NC,N2),HU(NC,N2),DSTCRE(NS,NS),PRTI(16,16)
C-----
C  COMMON /PROG/ IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEB
1 UG,ISET,IREF,I PSD,IYU,INORM
C-----
C  REAL*4 FMT(20)
C-----
C  OUTPUT OPTIONS-----
C  ICL=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-----
C  NSC=NS*NS
C  MH=NS
C  M=N2
C  CALL CHECK (EPS,NC,NG,NC,IREF)
C  IF (IREF.EQ.1) RETURN
C  IF (ISET.EQ.1) GC TC 20
C  CALL REPAIR (BA,G,H,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG,IRDMAT)
C  CALL REALF (NS,ISAF,BA)
C  IF (ICSTAB.EQ.0) GC TC 10
C  WRITE (1,18CO)
C  CALL RFEAL (ANSR)
C  DSTAB=ANSR

```

```

10  DO 10 I=1,NS
    DESTAB(I)=DESTAB
    CONTINUE
20  GO TO 30
30  CALL SETLP (BA,G,GAP,NS,NG,NC)
    CONTINUE
40  WRITE (6,1380)
    DO 40 I=1,NS
    WRITE (6,1350) (BA(I,J),J=1,NS)
    IF (IDSTAB.EQ.0) GC TC 50
    WRITE (6,1400)
    WRITE (6,1350) (DESTAB(I),I=1,NS)
50  CONTINUE
C-----EIGENSYSTEM CF THE OPEN LCUP DYNAMICS-----
    IF (ICL.EC.0.AND.IC.EQ.0) GO TO 90
    IF (ICL.EC.C.AND.NC.NE.0) GO TC 90
    DO 60 I=1,NS
    DO 60 J=1,NS
    GN(I,J)=BAC(I,J)
    CALL BALANC (NS,NS,GN,LCW,IHIGH,D1)
    CALL GRTHES (NS,NS,LOW,IHIGH,GN,D2)
    CALL GRTRAN (NS,NS,LOW,IHIGH,GN,D2,SC)
    CALL HCF2 (NS,NS,LCW,IHIGH,GN,CWR,CW1,SC,IERR)
    IF (IERR.NE.0) CALL EREXIT (NS,GN,IERR)
    CALL BALBAK (NS,NS,LOW,IHIGH,D1,NS,SC)
    CALL BALNCRALIZE AND PRINT CPEN LOOP EIGENSYSTEM
C-----
    IWRITE=1
    CALL CNCRM (CWR,CW1,SC,NS,IWRITE,NSC,DCL,D1,D2,WNORM,WNCRMI,HG,CM,
1ND,NS)
    IF (ICL.EC.2) RETURN
    IF (IC.EC.0.OR.(NC.NE.0.OR.IDSTAB.GT.0)) GO TO 50
    DO 70 I=1,NS
    IF (CWR(I).LT.0.) GC TC 70
    WRITE (5,1450)
    RETURN
70  CONTINUE
    IF (ICL.EC.3) GC TC 130
    DO 80 I=1,NS
    DO 80 J=1,NS
    W1(I,J)=SC(I,J)
    CALL MINV (NSQ,W1,NS,CDD,D1,D2)
    CONTINUE
90  IF (IDSTAB.EQ.0) GC TC 130
C-----FORM U * DIAG(DESTAB) * L-INV-----
    DO 100 I=1,NS
    DO 100 J=1,NS
    AA(I,J)=WNCRM(I,J)*DESTAB(J)
100  DO 120 I=1,NS

```

```

110      GO 120 J=1,NS
      DDD=0. CC
      DO 110 K=1,NS
      DDC=DDC+AA(I,K)*WNCRMI(K,J)
      DSTORE(I,J)=DDD
120      BA(I,J)=BA(I,J)+DDD
130      CONTINUE
      IF (NO. EC. 0) GO TO 145
      CALL READC (NU,NS,ISAH,HO)
      WRITE (6,1440)
      DO 140 I=1,NO
140      WRITE (6,1350) (HU(I,J),J=1,NS)
145      IF (IM. NE. 1) GC TO 150
      CALL MCCE (WNORM,HC,CM,NS,NC,NS,2)
150      CONTINUE
      IF (IFCF. EC. 0) GO TO 170
      CALL READC (NU,NC,C)
      WRITE (6,1470)
      DO 160 I=1,NO
160      WRITE (6,1350) (D(I,J),J=1,NC)
170      CONTINUE
      NOB=0
      IF (NC. EC. 0) GC TO 590
      IF (ICL. EC. 3) GC TC 270
      IF (IR. NE. 1. AND. IR. NE. 3) GO TO 210
      IF (ISET. EC. 1) GC TC 180
      CALL READC (NS,NC,ISAG,G)
180      CONTINUE
      CALL READCFB (NC,NS,FBGC)
      WRITE (6,1460)
      DO 190 I=1,NS
190      WRITE (6,1350) (G(I,J),J=1,NC)
      IF (IM. NE. 1) GO TO 200
      CALL MCCE (WNORMI,G,BM,NS,NS,NC,0)
200      CONTINUE
      GO TO 300
210      DO 220 I=1,NS
      DO 220 J=1,NS
      AM(I+MF,J)=0.0
220      CALL READCAY (NO,ISAA,AY)
      DO 240 I=1,NO
      DO 240 J=1,NS
      DDD=0. CC
      DO 230 K=1,NO
      DDC=DDC+AY(I,K)*HU(K,J)
230      AA(I,J)=CCD
240      WRITE (6,1460)
      DO 250 I=1,NO

```

```

2250 WRITE (C,1350) (AY(I,J),J=1,N0)
      DU 260 I=1,NS
      DO 260 J=1,NS
      DO 260 K=1,N0
      RM(I+MH,J)=RM(I+MH,J)+AA(K,I)*FC(K,J)
      IF (ISET.EQ.1) GO TO 280
      CALL REZLG(NS,NC,ISAG,G)
      CONTINUE
      IF (ICL.EQ.3) GO TC 290
      CALL REZCB(NC,ISAB,B)
      WRITE (C,1400)
      DO 300 I=1,NS
      WRITE (C,1350) (G(I,J),J=1,NC)
      IF (IM.NE.1) GO TO 310
      CALL MCEE(MNORMI,G,BM,NS,NS,NC,O)
      CONTINUE
      IF (ICL.EQ.3) GO TC 340
      WRITE (C,1410)
      DO 320 I=1,NC
      WRITE (C,1350) (B(I,J),J=1,NC)
      IF (ITFI.EQ.O) GO TC 350
      IF (OPEN LOOP TRANSFER FUNCTIONS-----)
      WRITE (C,1500)
      ITFX=1
      CALL TF(NS,NS,NSQ,BA,AA,NC,G,BM,NC,HQ,CM,IFDFW,D,BB,CC,CP,W,R,M,I,C
      1WR,CWI,SC,JCF,RES,D1,D2,DDD,EPS,ITFI,ITFX)
      IF (ICL.NE.3) GO TC 360
      IF (ING.EQ.O) RETURN
      GO TO 260
      CONTINUE
      IF (IR.EC.1.OR.IR.EC.3) GO TO 500
      C---CALCULATION OF CONTROL GAINS:FORMATION OF CONTROL HAMILTONIAN-----
      F-----GM*BI*GMT-----
      A-----FT-----
      ***F AND FT ARE THE UPEN LOOP
      ***BI IS NCXNC MATRIX AND TRANSPOSE
      ***A IS NCXNC CONTROL WEIGHTING
      ***M IS THE NSXNS STATE WEIGHTING
      ***GM IS THE NSXNC CONTROL
      ***DISTRIBUTION MATRIX-----
      CO 370 I=1,NC
      DU 370 J=1,MH
      PRO(I,J)=G(J,I)/B(I,I)
      DO 380 I=1,MH
      DO 380 J=1,MH

```

```

RM(I,J+MF)=0.D0
DO 380 K=1,NC
  RM(I,J+MF)=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+MF,J+MF)=-BA(J,I)
C-----M21 BLOCK-----
390 RM(I+MF,J)=-RM(I+MH,J)
C-----M12 BLOCK IS DEFINED IN LINE 430 ABOVE-----
400 CONTINUE
    IF (IDEEEG.EQ.0) GO TO 410
    WRITE(6,1510)
    CALL RAFRNT(M,M,M,S,RP,4,(9(1X,1PC13.6)))
    CALL BALANC(M,M,M,R,LOW,HIGH,D1)
    CALL CRTFES(M,M,M,LCW,HIGH,RM,D2)
    CALL ORTRAN(M,M,M,LCW,HIGH,RM,D2,X)
    CALL HCF2(M,M,M,LCW,HIGH,RM,WR,WI,X,IERR)
    IF (IERR.NE.0) CALL EXEXIT(M,RM,IERR)
    CALL BALEAK(M,M,M,LOW,HIGH,D1,M,X)
    CALL BUG DIAGNOSTICS ON EULER-LAGRANGE EQUATIONS
C-----
    IF (IDEEEG.EQ.0) GO TO 430
    WRITE(6,1520)
    DO 420 I=1,M
      WRITE(6,1530) WR(I),WI(I)
      WRITE(6,1540)
      CALL RAFRNT(M,M,M,S,X,4,(9(1X,1PC13.6)))
      CONTINUE
      IF (IDSTAB.EQ.1) GC TC 440
      IF (NOB.EC.C) WRITE(6,1550)
      IF (NGE.NE.0) WRITE(6,1560)
      IF (NCB.NE.0) GC TC 750
      CALL REGAIN(M,NS,NC,NCB,WR,WI,X,GN,WI,RM,W21,D1,CWR,LWI,SC,MHS,D2
1)
C-----CHECK EIGENVECTORS-----
    IF (IDEEEG.EQ.0) GO TO 450
    WRITE(6,1570)
    CALL RAFRNT(NS,NS,NS,S,SC,4,(9(1X,1PC13.6)))
    CONTINUE
C-----RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-----
    IF (IDSTAB.EQ.0) GO TO 470
    DO 460 I=1,NS
      BA(I,I)=BA(I,I)-DESTAB(I)
      IR=1
    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
      PRC(I,J)=0.0
      DO 480 K=1,MH
      PRC(I,J)=PRC(I,J)+G(K,I)*GN(K,J)
      FBGC(I,J)=-PRC(I,J)/B(I,I)
      IF (ILSTAB.EQ.1) GO TO 500
C-----NORMALIZE AND PRINT OPT. REG. CLCSED LOCF EIGENSYSTEM-----
      IWRITE=2
      CALL CACRM (CWR,CW,I,SC,NS,IWRITE,NSC,DCL,D1,D2,WNORM,WNCRMI,FBGC,
      IAA,NC,NS)
C-----THE OPTIMUM FEEDBACK CONTROL GAINS-----
      DO 500 IWRITE (6,1580)
      DO 510 I=1,NC
      FBGC(I,J)=FBGC(I,J),J=1,NS) (FBGC(I,J),J=1,NS)
C-----COMPLETE LOCAL C MATRIX OPEN LCCP U-INVERSE SAVED IN WNORMI&-----
      IF (IM.NE.1) GO TO 530
C-----
C IN COMPUTING LOCAL C RECOMPUTE U OPEN LCCP SINCE WNORM USED TO STUKE
C U & U-INV FOR CLOSED LCCP SYSTEMS; WNORMI USED TO SAVE U-INV OPEN LCCP
C-----
      DO 520 I=1,NS
      DO 520 J=1,NS
      WNCRM(I,J)=WNORMI(I,J)
      CALL MINV (NSQ,WNORM,NS,DDD,D1,D2)
      CALL MCLE (WNORM,FECC,AA,NS,NC,NS,3)
      CONTINUE
C-----THE CLCSED LOOP DYNAMICS MATRIX-----
      DO 550 I=1,NS
      DO 550 J=1,NS
      SUM=0.0
      DO 540 K=1,NC
      SUM=SUM+G(I,K)*FBGC(K,J)
      ACL(I,J)=BA(I,J)+SUM
      IWRITE (6,1600)
      CALL RAFRNT (MH,MH,MH,5,ACL,4,(5(1X,1PC13.6)))
      IF (IR.NE.1.AND. IR.NE.3) GC TO 590
      DO 560 I=1,NS
      DO 560 J=1,NS
      GN(I,J)=ACL(I,J)
      CALL BALANC (NS,NS,GN,LW,IGH,IGH,D1)
      CALL GRTRAN (NS,NS,LW,IGH,IGH,D2,SC)
      CALL GRTRN (NS,NS,LW,IGH,IGH,D2,SC)
      CALL HCR2 (NS,NS,LW,IGH,IGH,D1,NS,SC,IERR)
      IF (IERR.NE.0) CALL EREXIT (NS,GN,IERR)
      CALL BALBAK (NS,NS,LW,IGH,IGH,D1,NS,SC)

```



```

C-----
IF (ICSTAB.EQ.1) GO TO 770
      NORMALIZE AND PRINT OPT. ESTIMATOR EIGENSYSTEM-----
      IWRITE=4
      CALL CACRM (GR,CI,PRC,NS,IWRITE,NSC,DDC,D1,C2,WNURM,WNOKMI,HU,AA,
        INO,NS)
770      CO 780 I=1,MH
          DO 780 J=1,NO
          PRC(I,J)=HC(J,I)/RC(J,J)
780      DO 790 I=1,MH
          DO 790 J=1,NO
          FBGE(I,J)=0.00
          DO 790 K=1,MH
          FBGE(I,J)=FBGE(I,J)+GN(I,K)*PRC(K,J)
790      IF (ICSTAB.EQ.1) GO TO 810
          WRITE (C,1670)
          CALL RAPANT (MH,MH,MH,5,GN,4,(5(1X,1PD13.6)))
          WRITE (C,1680)
          DO 800 I=1,MH
          X(I,I)=LCRT(GN(I,I))
          DO 810 I=1,MH
          WRITE (C,1650) (X(I,I),I=1,MH)
          DO 820 I=1,MH
          WRITE (C,1640) (FBGE(I,J),J=1,NC)
800      C-----COMPLETE MODAL K MATRIX OPEN LOCP U-INV SAVED IN WNOFMI8-----
          IF (IM.NE.1) GO TO 830
          CALL MCCE (WNOFMI,FBGE,AA,MH,MF,NO,4)
830      CONTINUE FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-----
          RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-----
          IF (ICSTAB.EQ.0) GO TO 850
          DO 840 I=1,NS
          BA(I,J)=EA(I,J)-DSTCRE(I,J)
840      IR=2
          CONTINUE
          DO 870 I=1,NS
          DO 870 J=1,NS
          SUM=0.0
          CO 860 K=1,NO
          SUM=SUM+FBGE(I,K)*HC(K,J)
860      PRO(I,J)=BA(I,J)-SUM
870      WRITE (C,1650)
          CALL RAPANT (NS,NS,NS,5,PRC,4,(5(1X,1PD13.6)))
          IF (IR.LT.2) GO TO 850
          CALL BALANC (NS,NS,PRC,LOW,HIGH,D1)
          CALL ORTFAN (NS,NS,LOW,HIGH,PRC,D2)
          CALL CRTFAN (NS,NS,LOW,HIGH,PRC,D2,GM)
          CALL HCR2 (NS,NS,LCH,HIGH,PRO,CR,C1,GM,IERR)
          IF (IERR.NE.0) CALL EREXIT (NS,PRC,IERR)

```

```

CALL BALEAK (NS,NS,LC, HIGH,D1,NS,GM)
WRITE (6,1560)
C-----NORMALIZE AND PRINT SUBOPT. ESTIMATOR EIGENSYSTEM-----
IWRITE=1
CALL CNORM (CR,CI,GM,NS,IWRITE,NSQ,DD,CI,D2,WNGRM,WNGRMI,HC,AA,
1NO,NS)
DO 880 I=1,NS
IF (CR(I)-LT,0.0) GO TO 880
WRITE (6,1660)
RETURN
CONTINUE
GO TO 500
IF (I,CI,0) GO TO 1260
DO 910 I=1,NJ
DO 510 I=1,MH
PRC(I,J)=0.0
DO 910 K=1,NO
PRC(I,J)=PRC(I,J)+RC(I,K)*FBGE(J,K)
DO 920 I=1,MH
DO 520 I=1,MH
CQ(I,J)=C.DC
DO 920 K=1,NO
CQ(I,J)=CQ(I,J)-FBGE(I,K)*PRO(K,J)
CONTINUE
C-----THE RMS STATE AND CONTROL RESPONSES-----
IR=IR+1
GO TO (1C50,109C,94C,540), IR
DO 950 I=1,NS
DO 550 J=1,NG
X(I,J)=C.0
DO 950 K=1,NG
X(I,J)=X(I,J)+GAM(I,K)*Q(K,J)
DO 970 I=1,NS
DO 570 J=1,NS
SUM=0.0
DO 960 K=1,NG
SUM=SUM-X(I,K)*GAM(J,K)
PRC(I,J)=SUM+CQ(I,J)
CQ(I,J)=SUM
CQ(J,I)=SUM
W21(I,J)=GM(I,J)
W21(J,I)=GM(J,I)
CALL MINV (NSQ,W21,NS,DD,D1,D2)
CALL SCCV (NS,GM,W21,CR,CI,NS,GM,W21,CR,CI,FRU,GN)
WRITE (6,1670)
CALL RAFFAT (MH,MH,MH,5,GN,4,(5(1X,1FC13.6)))
WRITE (6,1680)

```

```

980      CO 980 J=1,MH
      X(I,I)=DSQRT(GN(I,I))
      WRITE (6,1650) (X(I,I),I=1,MH)
      IF (IC.EC.O) GO TO 1260
      DO 1000 I=1,NC
      DO 1000 J=1,NS
      SUM=0.C
      DO 990 K=1,NS
      SUM=SUM+FBGC(I,K)*GN(K,J)
      X(I,J)=SUM
      DO 1020 I=1,NS
      DO 1020 J=1,NS
      SUM=0.C
      IF (INC.EC.O) GO TO 1020
      DO 1010 K=1,NC
      SUM=SUM+G(I,K)*X(K,J)
      PRCC(I,J)=CQ(I,J)+SUM
      CALL SCCV (NS,SC,W11,CWR,CWI,NS,GM,W21,CR,CI,PRO,BA)
      IF (INC.EC.O) GO TO 1040
      DO 1030 I=1,NC
      DO 1030 J=1,NS
      W21(I,J)=0.C
      DO 1030 K=1,NS
      W21(I,J)=W21(I,J)+FBGC(I,K)*BA(J,K)
      DO 1060 I=1,NS
      DO 1060 J=1,NS
      SUM=0.C
      IF (INC.EC.O) GO TO 1060
      DO 1050 K=1,NC
      SUM=SUM+G(I,K)*W21(K,J)
      PRCC(I,J)=SUM
      DO 1070 I=1,NS
      DO 1070 J=1,NS
      PRCC(I,J)=PRCC(I,J)+CC(I,J)+PRO(J,I)
      PRCC(J,I)=PRCC(I,J)
      CALL SCCV (NS,SC,W11,CWR,CWI,NS,SC,W11,CWR,CWI,PRO,CQ)
      DO 1080 I=1,NS
      DO 1080 J=1,NS
      GM(I,J)=CC(I,J)-BA(I,J)+GN(I,J)
      GM(J,I)=GM(I,J)
      GO TO 1100
      CALL SCCV (NS,SC,W11,CWR,CWI,NS,SC,W11,CWR,CWI,CQ,GM)
      IF (INC.EC.O) GO TO 1150
      DO 1120 I=1,NS
      DO 1120 J=1,NC
      PRO(I,J)=0.CO
      CO 1110 K=1,NS
      PRCC(I,J)=PRCC(I,J)+GP(I,K)*FBGC(J,K)

```

```

1120 CONTINUE I=1,NC
    DO 1140 J=1,NC
    SC(I,J)=0.00
    DO 1130 K=1,NS
    SC(I,J)=SC(I,J)+FBGC(I,K)*PRO(K,J)
1140 CONTINUE
    IF (IREG .EC. 0) GC TO 1170
    DO 1160 J=1,NS
    DO 1160 J=1,NS
    CQ(I,J)=GM(I,J)
    GO TO 1150
1170 WRITE (6,1700)
    CALL RAFAINT (MH,MH,MH,5,GM,4,*(5(1X,1PD13.6)))
    IF (IR.GT.2) GO TO 1190
    DO 1180 J=1,MH
    DO 1180 J=1,MH
    CQ(I,J)=GN(I,J)+GM(I,J)
1180 CONTINUE
    WRITE (6,1710)
    CALL RAFAINT (MH,MH,MH,5,CQ,4,*(5(1X,1PD13.6)))
    IF (NC.EC.0) GO TO 1210
    WRITE (6,1720)
    DO 1200 J=1,NC
    DO 1200 J=1,NC (SC(I,J),J=1,NC)
1200 WRITE (6,1730)
1210 DO 1220 I=1,NS
1220 CQ(I,I)=LSQRT(CQ(I,I))
    IF (NC.EC.0) GC TO 1240
    DO 1230 I=1,NC
    SC(I,I)=LSQRT(SC(I,I))
1230 SC(I,I)=LSQRT(SC(I,I))
1240 WRITE (6,1740)
    DO 1250 I=1,NS
    IF (I.LF.NC) WRITE (6,1750) CQ(I,I),SC(I,I)
    IF (I.GT.NC) WRITE (6,1750) CQ(I,I)
1250 CONTINUE
1260 IF (ITF3 .EC. 0) GC TO 1290
    IF (ITF3 .EC. 0) COMPENSATOR FROM MEAS TO INPLT AND COMPUTE TF
    DO 1280 I=1,NS
    DO 1280 J=1,NS
    SUM=0.00
    DO 1270 K=1,ND
    SUM=SUM+FBGE(I,K)*HC(K,J)
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,
    1WK,WI,CAR,CMT,SC,JCF,RES,UI,D2,EUD,EPS,ITF3,ITFX)

```

```

1290 CONTINUE
C-----COMPUTE PSD FUNCTIONS OF THE CONTROLLED SYSTEM-----
IF (IPSD.EC.O) GO TO 1310
IF (IYLC.LT.3) GO TO 1300
CALL PSSCAL (M,NS,RM,X,NC,GW,GV,FBGC,NC,HY,FU,HC,FBGE,NG,
1 GAM,ACL,BA,WR,WI,CI,D2,JCF,RES,Q,RC,BE,CC,1,IPSD,INDRM)
CALL PSSCAL (M,NS,RM,X,NC,GW,GV,FBGC,NC,HY,FU,HC,FBGE,NG,
1 GAM,ACL,BA,WR,WI,CI,D2,JCF,RES,Q,RC,BE,CC,2,IPSD,INDRM)
GO TO 1310
1300 CALL PSSCAL (M,NS,RM,X,NC,GW,GV,FBGC,NC,HY,FU,HC,FBGE,NG,
1 GAM,ACL,BA,WR,WI,CI,D2,JCF,RES,Q,RC,BE,CC,IVU,IPSD,INDRM)
1310 IF (ISS.EC.O) RETURN
IF (NC.NE.O) GO TO 1330
DO 1320 I=1,NS
DO 1320 J=1,NS
ACL(I,J)=BA(I,J)
CONTINUE
CALL MINV (ASQ,ACL,NS,ODD,O1,D2)
CALL READW (NG,WR)
WRITE (6,1770) (WR(I),I=1,NG)
1340 DO 1340 I=1,NS
DO 1340 J=1,NG
WI(I)=C.C
WI(I)=WI(I)+GAM(I,J)*WR(J)
DO 1360 I=1,NS
CR(I)=C.C
DO 1350 J=1,NS
CR(I)=CR(I)-ACL(I,J)*WI(J)
1350 WRITE (6,1350) CR(I)
DO 1370 I=1,NC
CI(I)=C.C
1370 CI(I)=CI(I)+FBGC(I,J)*CR(J)
WRITE (6,1750) (CI(I),I=1,NC)
RETURN
C-----
1370 FORMAT (2X,IP6D14.6/,2X,6D14.6)
1380 FORMAT (//,EX,45HOPEN LOOP DYNAMICS MATRIX.....F.,//)
1390 FORMAT (10(2X,OPD11.4))
1400 FORMAT (//,EX,45HFE CONTROL DISTRIBUTION MATRIX.....G.,//)
1410 FORMAT (//,EX,45HTHE CCST MATRIX.....B.,//)
1420 FORMAT (//,EX,45HPROCESS NOISE DENSITY - PROCESS NOISE.....C.,//)
1430 FORMAT (//,EX,45HPCWERSPECTRAL SCALING MATRIX.....F.,//)
1440 FORMAT (//,EX,45HMEASUREMENT SPECTRAL SCALING MATRIX.....R.,//)
1450 FORMAT (//,EX,45HPOWER SPECTRAL SCALING MATRIX.....A.,//)
1460 FORMAT (//,EX,45HMULTIPLY CCST MATRIX.....C.,//)
1470 FORMAT (//,EX,45HMEASUREMENT FEEDTHROUGH MATRIX.....C.,//)

```

```

1480  FORMAT (//,25X,28H...DESTABILIZATION CASE...//,10X,35HTHE FOLLO
1490  WING VALUES WILL BE ADDED DOWN,/,10X,45HTHE DIAGONAL OF THE "F" MA
1500  TRIX TO DESTABILIZE IT,/,10X,41HOPTIMAL GAINS FOR THE DESTABILIZE
1510  3D SYSTEM,/,10X,39HARE THEN USED AS FIXED SUEOPTIMAL GAINS,/,10X,28
1520  4HFOR THE SYSTEM CALCULATIONS,/,//)
1530  FORMAT (//,43H PROGRAM TERMINATING DUE TO UNSTABLE SYSTEM)
1540  FORMAT (//,5X,31HOPEN LOOP TRANSFER FUNCTIONS,/,//)
1550  FORMAT (//,5X,32H EULER-LAGRANGE SYSTEM MATRIX,/,//)
1560  FORMAT (//,5X,43HEIGENVALUES AND EIGENVECTORS OF THE 2N X 2N,/,5X,
1570  145HEULER-LAGRANGE SYSTEM AFTER FOR2,/,//)
1580  FORMAT (1X,1P2C13.6)
1590  FORMAT (1X)
1600  FORMAT (//,5X,41HEIGENSYSTEM OF OPTIMAL REGULATOR,/,//)
1610  FORMAT (//,5X,41HEIGENSYSTEM OF OPTIMAL ESTIMATOR,/,//)
1620  FORMAT (//,5X,39H EIGENVECTORS FROM RGAIN PFICR TO CNORM,/,//)
1630  1*GTS...//)
1640  FORMAT (10(2X,1P01.4))
1650  FORMAT (//,5X,45H THE CLOSED LOOP DYNAMICS MATRIX .....F-G#C...//)
1660  FORMAT (//,60H PROGRAM TERMINATING DUE TO UNSTABLE CLOSED LOOP
1670  1 SYSTEM)
1680  FORMAT (//,2X,45H TRANSFER FUNCTIONS THROUGH THE CLOSED-LOOP SY
1690  STEM,/,//)
1700  FORMAT (//,5X,45H FILTER STEADY STATE GAINS.....K.....//)
1710  FORMAT (1X,2X,1P6D14.6)
1720  FORMAT (//,5X,45H THE CLOSED LOOP FILTER DYNAMICS MATRIX IS.....//)
1730  FORMAT (//,5X,43H PROGRAM TERMINATING DUE TO UNSTABLE FILTER)
1740  FORMAT (//,5X,45H THE CCVARIANCE OF THE ESTIMATION ERROR...P...//)
1750  FORMAT (//,5X,45H RMS VALUES OF THE ESTIMATION ERROR.....//)
1760  FORMAT (15(1X,1P013.6))
1770  FORMAT (//,5X,45H THE CCVARIANCE OF THE ESTIMATE.....XHAT...//)
1780  FORMAT (//,5X,45H THE STATE COVARIANCE MATRIX.....X=XHAT + P.....//)
1790  FORMAT (//,5X,45H THE CCNTRL CCVARIANCE.....U=C#XHAT#C.....//)
1800  FORMAT (1P6D14.6)
1810  FORMAT (//,2X,18H STATE RMS RESPONSE,20X,20HCCNTRL RMS RESPONSE,/)
1820  FORMAT (1X,1P015.7,25X,015.7)
1830  FORMAT (//,5X,50HCOMPENSATOR TRANSFER FUNCTIONS FROM MEAS. TO INPU
1840  T,/,5X,22H.....//)
1850  FORMAT (//,5X,46HSTEADY DISTURBANCE VECTOR.....W.....//)
1860  1,10(1X,1P012.4//)
1870  FORMAT (//,5X,45HSTEADY STATE VALUES OF STATE VAR. ARE.....//)
1880  FORMAT (//,5X,47HSTEADY STATE CCNTRL IS.....//)
1890  1/10(1X,1P012.4//)
1900  FORMAT (//,5X,49HENTER THE MAGNITUDE OF THE DESTABILIZATION VECTOR
1910  1,/,8X,47HFC BE ADDED DOWN THE DIAGONAL OF THE "F"-MATRIX,/,8X,18HT
1920  2C DESTABILIZE IT,/,//)
1930  ENL
1940  C=====

```

```

SUBROUTINE RAPRNT (NMAX,M,N,L,A,ICIP,FMT)
REAL*8 J(NMAX,N)
DIMENSION FMT(10)
NU=L
DO 20 NL=1,N,L
IF (NU.GT.N) NU=N
DO 10 I=1,M
WRITE (6,FMT) (A(I,J),J=NL,NU)
WRITE (6,30)
NU=NU+L
RETURN
FORMAT (1X)
END
C=====
SUBROUTINE RGAIN (M,NS,NC,NOB,WR,WI,VF,GN,W11,TCB,W21,LT,C,CI,CT,M
IHS,MT)
IMPLICIT REAL*8 (A-H,C-Z)
DIMENSION WR(M),WI(M),VF(M,M),GN(NS,NS)
DIMENSION W11(NS,NS),TCB(M,M),W21(NS,NS),LT(NS),MT(NS)
DIMENSION C(NS),CI(NS),CT(NS,NS)
K=1
KP=1
KN=1
NRZEV=0
NCPZEV=C
IF (K.GT.M) GO TO 210
C CHECK FOR EIGVAL AT OR NEAR J-CMEGA AXIS TO INCLUDE IN E-L EIGSYS
C TURN FIRST ONE POSITIVE AND SECOND ONE NEGATIVE
C-----
EIGVR=LABS(WR(K))
IF (EIGVR-GE.1.C-10) GC TO 60
IF (WI(K)) 40,20,40
NRZEV=NRZEV+1
IF (NRZEV.GT.1) GO TO 30
WR(K)=EIGVR
GO TO 80
WR(K)=-EIGVR
WRITE (6,25C)
GO TO 150
NCPZEV=NCPZEV+1
IF (NCPZEV.GT.1) GC TO 50
WR(K)=EIGVR
WR(K+1)=EIGVR
GO TO 110
WR(K)=-EIGVR
WR(K+1)=-EIGVR
WRITE (6,300)

```

```

60 GO TO 18C
70 IF (WR(K)) 140,70,70
80 IF (WI(K)) 110,80,110
C-----EIGENVECTOR FOR REAL EIGENVALUE, POSITIVE-----
90 IF (NCE.EC.O) GC TC 100
100 DO 50 J=1,M
    TCE(J,KF)=VF(J,K)
    KP=KP+1
    K=K+1
    GO TO 10
C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, POSITIVE REAL PART-----
110 IF (NOB.EC.O) GO TC 130
120 DO 120 J=1,M
    FR=VF(J,K)
    FI=-VF(J,K+1)
    TCE(J,KF)=FR+FI
    TCE(J,KF+1)=FR-FI
    KP=KP+2
    K=K+2
    GO TO 1C
140 IF (WI(K)) 180,150,180
C-----EIGENVECTOR FOR REAL EIGENVALUE, NEGATIVE REAL PART-----
150 C(KN)=WR(K)
    CI(KN)=WI(K)
    IF (NOB.NE.O) GO TC 170
    KN=KN+1
    DO 160 J=1,M
    TCE(J,KN)=VF(J,K)
    KN=KN+1
    K=K+1
    GO TO 1C
C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, NEGATIVE REAL PART-----
180 RR=WR(K)
    RI=WI(K)
    C(KN)=RR
    C(KN+1)=RI
    CI(KN)=FI
    CI(KN+1)=-FI
    IF (NOB.NE.O) GO TC 200
    KN=KN+1
    DO 190 J=1,M
    FR=VF(J,K)
    FI=-VF(J,K+1)
    TCE(J,KN)=FR+FI
    TCE(J,KN+1)=FR-FI
    KN=KN+2
    K=K+2
    GO TO 1C
190
200

```

```

210 CONTINUE
C IF (NOB.NE.O) GO TO 240-----FORMATION OF W11-----
DO 220 I=1,NS
DO 220 J=1,NS
W11(I,J)=TCE(I,J+NS)
CT(I,J)=W11(I,J)-----FORMATION OF W21-----
DO 230 I=1,NS
DO 230 J=1,NS
W21(I,J)=TCE(I+NS,J+NS)
IF (NOB.EQ.O) GO TO 260
DO 250 I=1,NS
DO 250 J=1,NS
W21(I,J)=TCE(I,J)
W11(I,J)=TCE(I+NS,J)
250 CONTINUE
260-----INVERT W11-----
NSC=NS*NS
CALL MINV (NSQ,W11,NS,CETC,LT,MT)
C-----CALCULATE THE RGAIN MATRIX-----
DO 270 IL=1,NS
DO 270 JL=1,NS
GN(IL,JL)=O.DO
CO 270 KL=1,NS
GN(IL,JL)=GN(IL,JL)+W21(IL,KL)*W11(KL,JL)
270 IF (NCDE.EQ.O) RETURN
DO 280 I=1,NS
DO 280 J=1,NS
CT(I,J)=W11(J,I)
RETCN
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.)
C=====
SUBROUTINE MINV (NSQ,A,N,D,L,M)
IMPLICIT REAL*8 (A-H,C-Z)
DIMENSION A(NSQ),L(N),M(N)
DOUBLE PRECISION A,D,BIGA,HOLD
NM=N*N
L=1.OCC
NK=-N
DO 180 K=1,N
NK=NK+K
L(K)=K

```

```

P(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
BIGA=A(IJ)
L(K)=I
P(K)=J
CONTINUE
-----INTERCHANGE ROWS-----
J=L(K)
IF (J-K) 50,50,30
KI=K-N
DO 40 I=1,N
KI=KI+N
HOLD=-A(KI)
JI=KI-K+J
A(KI)=A(JI)
A(JI)=HOLD
-----INTERCHANGE COLUMNS-----
I=P(K)
IF (I-K) 80,80,60
JP=N*(I-1)
DO 70 J=1,N
JK=NK+J
JI=JP+J
HOLD=-A(JK)
A(JK)=A(JI)
A(JI)=HOLD
-----DIVIDE COLUMN BY MINUS PIVOT-----
----- (VALUE OF PIVOT ELEMENT IS CONTAINED IN BIGA) -----
IF (BIGA) 100,90,100
C=0.00C
RETURN
DO 120 I=1,N
IF (I-K) 110,120,110
IK=NK+I
A(IK)=A(IK)/(-BIGA)
CONTINUE
-----REDUCE MATRIX-----
DO 150 J=1,N
IK=NK+I
HOLD=A(IK)
IJ=I-N
DO 150 I=1,N

```



```

1  VR1(NR),VR2(NR),WR(NR,NR),WRI(NR,NR)
10  REAL*8 A,B,C,D,K1,K2,K3,K4
    CO 20 I=1,NL
    DO 20 J=1,NR
    X(I,J)=C
    DO 20 II=1,NL
    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)=C
    CO 30 I=1,NR
    Q(I,J)=Q(I,J)+X(I,J)*WRI(J,J)
40  CONTINUE
    I=1 (VL2(I)) 60,110,60
    IF J=1
    IF (VR2(J)) 80,90,80
    A=VL1(I)+VR1(J)
    B=-2.*VL2(I)*VR2(J)
    C=A**2+VL2(I)**2+VR2(J)**2
    E=C**2-E**2
    K1=A*C/E
    K2=-(VR2(J)*C+VL2(I)*B)/D
    K3=-((VF2(J)*B+VL2(I)*C)/D
    K4=-A*B/E
    I1=I+1
    J1=J+1
    X(I,J1)=+K1*C(I,J)+K2*Q(I,J1)+K3*Q(I1,J)+K4*Q(I1,J1)
    X(I1,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+2 100
    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*C(I,J)-K2*Q(I+1,J)
    X(I+1,J)=K2*Q(I,J)+K1*Q(I+1,J)
    J=J+1
    IF (J.LE.NR) GO TO 70
    I=I+2
    GO TO 160
100  J=1
    IF (VR2(J)) 130,140,130
110  A=VR1(J)+VL1(I)
120  B=A**2+VR2(J)**2
130  K1=A/B

```

```

140 K2=VR2(J)/B
      X(I,J)=K1*C(I,J)-K2*Q(I,J+1)
      X(I,J+1)=K2*Q(I,J)+K1*C(I,J+1)
      J=J+2
      GO TO 15C
145 X(I,J)=C(I,J)/(VR1(J)+VLL(I))
      J=J+1
150 IF (J.LE.NR) GC TO 120
      I=I+1
160 IF (I.LE.NL) GO TO 50
      DO 170 I=1,NL
      DU 17C J=1,NR
      C(I,J)=C
      DO 170 J=1,NL
      Q(I,J)=C(I,J)+WL(I,II)*X(II,J)
      DO 190 I=1,NL
      DO 190 J=1,NR
      X(I,J)=C
      DO 180 J=1,NR
      X(I,J)=X(I,J)+Q(I,JJ)*WR(J,JJ)
180 CONTINUE
190 RETURN
      ENC
=====
C SUBROUTINE MODE (MACRM,G,GNORM,NS,N1,N2,ICCN)
=====
C MACRM TRANSFORMATION MATRIX U OR L-INV
C NS NO. CF STATE
C NC CF INPUTS OR OUTPUTS
C ICCA CONFCL FLAG TO INDICATE WHICH TRANSFORMATION
C 0 = MCDAL G
C 1 = MCDAL GAMMA
C 2 = MCDAL H
C 3 = MCDAL C
C 4 = MCDAL K
C 5 = CONTROL EIGENVECTCR MATRIX
C 6 = MEASUREMENT EIGENVECTCR MATRIX
=====
C IMPLICIT REAL*8(A-H,C-Z)
C DIMENSION WNORM(NS,NS),G(N1,N2),GNCRM(N1,N2)
C DO 10 I=1,N1
C DO 10 J=1,N2
C GNCRM(I,J)=C
C IPCINT=ICCN+1
C DO 20 (2C,2C,90,50,20,50,90), IPCINT
C DO 30 I=1,N2
C DO 30 K=1,NS

```

```

30  GNCRM(I,J)=GNORM(I,J)*WNORM(I,K)*G(K,J)
40  GO TO (4C,70,90,50,80), IPCINT
50  WRITE (6,170)
60  DO 60 I=1,NS
   WR ITE (6,230) (GNORM(I,J),J=1,N2)
   RETURN
70  WR ITE (6,180)
   GO TO 50
80  WR ITE (6,240)
   GO TO 50
90  DO 100 J=1,NS
   DO 100 I=1,N1
   DO 100 K=1,NS
   GNCRM(I,J)=GNORM(I,J)+G(I,K)*WNORM(K,J)
100  GO TO (110,110,110,120,110,130,140), IPCINT
110  WR ITE (6,150)
   GO TO 150
120  WR ITE (6,200)
   GO TO 150
130  WR ITE (6,210)
   GO TO 150
140  WR ITE (6,220)
   DO 160 I=1,N1
150  WR ITE (6,230) (GNORM(I,J),J=1,NS)
160  RETURN
-----
170  FORMAT (//,5X,45HMCDCAL CONTROL DISTRIBUTION MATRIX...TI*G...//)
180  FORMAT (//,5X,50HMCDCAL PROCESS NOISE DISTRIBUTION MATRIX...TI*GAM...
190  1.,//)
200  FORMAT (//,5X,45HMCDCAL MEASUREMENT SCALING MATRIX...H(BAR)*I...//)
210  FORMAT (//,5X,45HMCDCAL CONTROL GAINS...C*TI...//)
220  FORMAT (//,5X,45HMCDCAL EIGENVECTOR MATRIX...C*M...//)
230  FORMAT (//,5X,45HMCDCAL MEASUREMENT EIGENVECTOR MATRIX...H(BAR)*M...//)
240  FORMAT (1X,(2X,1P6C14.6))
   ENC
=====
SUBROUTINE GNORM (WZ,WY,VEC,NS,IMRITE,NSQ,D1,D2,WNCRM,WNORMI,H
1C,CP,N1,N2)
=====
C=====
C      WZ(I)      REAL PART OF I-TH EIGENVALUE
C      WY(I)      COMPLEX PART OF I-TH EIGENVALUE
C      VEC        MATRIX OF RIGHT EIGENVECTORS STORED IN REAL FORM
C      NS         FROM HCF2
C               AC. OF STATES
C=====

```

```

C----- FLAG TO CONTROL FORMATS FOR DIFFERENT EIGENSYSTEMS=
C WNCFM NORMALIZED MATRIX U OF RIGHT EIGENVECTORS STORED
C BY COLUMNS IN REAL FCRM
C WNCRMI U-INVERSE 2*CONJUGATE OF LEFT EIGENVECTORS
C STORED BY ROW IN REAL FCRM
C NSC,DD,D1,D2 - ARGUMENTS PASSED TO MINV
C=====
IMPLICIT REAL*8 (A-F,C-Z)
REAL*8 FIELD,CCMPA,SEMCGL,RIGHT,FMT
DIMENSION W2(NS),WV(NS),VEC(NS,NS),WNORM(NS,NS),STOR
1E(4),D1(NS),D2(NS),FMT(14),HO(N1,N2),CM(N1,N2)
DATA FIELD/5E12.5/,COMMA/5H,.,./,SEMCGL/5F,.,./,FMT/
16H(1X,1F,13*1H /,SEMCGL/4H,.,./
C-----NORMALIZE COMPLEX EIGENVECTORS BY LARGEST ELEMENT-----
KK=0
LR=0
LC=0
DO 50 K=1,NS
IF (KK.EC.1) GO TO 40
IF (DABS(WV(K)).LT.1.D-10) GO TO 50
LC=LC+1
EMAX=0.LC
DO 20 I=1,NS
CMCC=VEC(I,K)**2+VEC(I,K+1)**2
IF (CMCC-EMAX) 20,10,10
EMAX=CMCC
N=I
CONTINUE
VMR=VEC(N,K)
VMI=VEC(N,K+1)
DO 30 I=1,NS
VI=VEC(I,K)
VJ=VEC(I,K+1)
VECRN=(VR*VMR+VI*VMI)/EMAX
VECMN=(-VR*VMI+VI*VMR)/EMAX
WNCRM(I,K)=VECRN
WNCRM(I,K+1)=VECMN
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(WV(K)).GE.1.D-10) GO TO 80
LR=LR+1
REPCD=C.10

```

```

60      GO 60 I=1,NS
      RMCD=VEC(I,K)**2+REMCD
      RMCD=DSCRT(REMCD)
      GO 70 I=1,NS
      RVEC=VEC(I,K)/RMCD
      WNCRM(I,K)=RVEC
      CONTINUE
      GO TO (50,100,110,120,130), 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)
      KK=C
      NPRTW=C
      NPRTW=1
      DO 180 I=1,NS
      IF (KK.EC.1) GO TO 170
      IF (DAE(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
      NPRTW=1
      NPRTW=NFRTW+1
      NPRTW=NFRTW+1
      IF (KK.EC.1) GO TO 160
      STCRE(NPRTW)=WZ(I)
      FMT(NFMTW)=FIELD
      NPRTW=NFMTW+1
      FMT(NFMTW)=SEMCCL
      GO TO 150
      STCRE(NPRTW)=WZ(I)
      FMT(NFMTW)=FIELD
      FMT(NFMTW+1)=COMMA
      STCRE(NPRTW+1)=WY(I)
      FMT(NFMTW+2)=FIELD
      FMT(NFMTW+3)=SEMCCL
      NPRTW=NFRTW+1
      GO TO 150
      KK=C
170

```

```

180 CONTINUE
    FMT(NFMTM+1)=SEMENT
    FMT(NFMTM+1)=RIGHT
    WRITE (6,FMT) (STORE(J),J=1,NPRTM)
    IF (IWRITE.EQ.1) GC TO 190
    WRITE (6,37C)
    GO TO 200
190 WRITE (6,38C)
    CALL RAPRNT (NS,NS,NS,6,WNORM,4, (6(1X,1PD13.6)))
    GO TO 210
210 CALL MCLC (WNORM,HC,CM,NS,N1,N2,5)
    GO TO 220
220 CALL MCLC (WNORM,HC,CM,NS,N1,N2,6)
    GO TO 230
230 CALL MCLC (WNORM,HC,CM,NS,N1,N2,6)
    WRITE (6,39C)
    GO TO 250
250 WRITE (6,40C)
    GO TO 260
260 WRITE (6,41C)
    GO TO 270
270 WRITE (6,42C)
    GO TO 280
280 WRITE (6,430)
    IF (IWRITE.GT.1) GO TO 310
    DO 300 I=1,NS
    DO 300 J=1,NS
    WNCRM(I,J)=WNCRM(I,J)
    CALL MINV (NSQ,WNORM,NS,OCQ,D1,D2)
    CALL RAPRNT (NS,NS,NS,6,WNCRM,4, (6(1X,1PD13.6)))
    RETURN
310 CALL MINV (NSQ,WNORM,NS,OCQ,D1,D2)
    CALL RAPRNT (NS,NS,NS,6,WNCRM,4, (6(1X,1PD13.6)))
    RETURN
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
    WNCRM(I,J)=WNCRM(I,J)
    CALL MINV (NSQ,WNORM,NS,OCQ,D1,D2)
    CALL RAPRNT (NS,NS,NS,6,WNCRM,4, (6(1X,1PD13.6)))
    RETURN
310 CALL MINV (NSQ,WNORM,NS,OCQ,D1,D2)
    CALL RAPRNT (NS,NS,NS,6,WNCRM,4, (6(1X,1PD13.6)))
    RETURN
C-----
320 FORMAT (//5X,42HOPEN LOOP EIGENVALUES.....DET (SI-F).....//)
330 FORMAT (//5X,46HFC-LOOP OPTIMAL REG. E-VALUES.....DET (SI-F+G*C).....//)
340 FORMAT (//5X,46HFC-LOOP SUBOPT. REG. E-VALUES.....DET (SI-F+G*C).....//)
350 FORMAT (//5X,46HFC-LOOP OPTIMAL. EST. E-VALUES.....DET (SI-F+K*H).....//)
360 FORMAT (//5X,46HFC-LOOP SUBOPT. EST. E-VALUES.....DET (SI-F+K*H).....//)
370 FORMAT (//5X,46HOPEN LCCP RIGHT EIGENVECTOR MATRIX.....T.....//)
380 FORMAT (//5X,46HFC-LOOP LCCP RIGHT EIGENVECTOR MATRIX.....T-INV.....//)
390 FORMAT (//5X,46HOPEN LCCP LEFT EIGENVECTOR MATRIX.....M.....//)
400 FORMAT (//5X,46HFC-LOOP LCCP LEFT EIGENVECTOR MATRIX.....M-INV.....//)
410 FORMAT (//5X,46HFC-LOOP SUBOPT. REG. LEFT E-VECTOR MATRIX.....M-INV.....//)
420 FORMAT (//5X,46HFC-LOOP SUBOPT. REG. LEFT E-VECTOR MATRIX.....M-INV.....//)
430 FORMAT (//5X,51HFC-LOOP SUBOPT. FILTER LEFT E-VECTOR MATRIX.....M-INV.....//)
    1.,./

```

```

C=====
ENC
SUBROUTINE TF (N,NM,NSC,A,AA,M,B,BM,L,C,CM,IFCFW,D,EB,CC,CP,
1 EVF,EVI,PR,PI,SC,JCF,RES,D1,D2,LDL,EPS,ITF,ITFX)
IMPLICIT REAL*8(A-H,I-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),CF(N),EVR(N),EVI(N),PR(N),PI(N),SC(N,N),JCF(N),RES(N),D1(N
2),E2(N)
C--SAVE COMPUTATION ON OL AND CL SYS WITH MODAL WORK DONE IN CPTSYS-----
IF (ITF) .EQ. 1) GC TC 50
IF (ITFX) .EQ. 2) GC TC 10
CALL PCLES (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
DJ 20 J=1,N
AA(I,J)=SC(I,J)
20 DO 30 I=1,L
DO 30 J=1,N
CM(I,J)=C.CC
30 DO 30 K=1,N
CM(I,J)=CM(I,J)+C(I,K)*AA(K,J)
CALL MINV (NSC,AA,N,DDC,D1,D2)
40 DO 40 I=1,N
DO 40 J=1,M
BM(I,J)=C.CO
50 DU 40 K=1,N
EM(I,J)=EM(I,J)+AA(I,K)*B(K,J)
CONTINUE
60 RETURN
ENC
C=====
SUBROUTINE POLES (N,NM,A,AA,M,B,L,C,EVR,EVI,D1,D2,JCF,SC)
IMPLICIT REAL*8(A-H,I-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
AA(I,J)=A(I,J)
CALL BALANC (NM,N,LCM,IGH,D1)
CALL CRTFES (NM,N,LCW,IGH,AA,D2)
CALL CRTFRN (NM,N,LCW,IGH,AA,D2,SC)
CALL HCRF2 (NM,N,LCW,IGH,AA,EVR,EVI,SC,IERR)
10

```

```

IF (IEFF.NE. 0) GC TC 30
CALL BALBAK (NM,N,LCW,IHIGH,D1,N,SC)
WRITE (6,40)
DO 20 I=1,N
  WRITE (6,50) EVR(I),EVI(I)
  RETURN
  WRITE (6,60)
  RETURN
C-----
40  FORMAT (//,28H IF DENCMINATOR EIGENVALUES:./)
50  FORMAT (//,2X,3H (,F13.6,4H)+J(,F13.6,1H))
60  FORMAT (35H FAILURE IN HQR2, CALCULATING PGLES)
ENC
C=====
SUBROUTINE ZEROS (K1,K2,IFDFW,N,NM,A,AA,M,B,L,C,U,BB,CC,CP,EVR,EVI,
1, L1,C2,EPS)
  IMPLICIT REAL*8(A-F,C-Z)
  DIMENSION A(N,N),AA(N,N),B(N,M),C(L,N),L(L,M),BB(N),CC(N),CP(N),EVI
1(N),EVI(N),DI(N),C2(N)
  DOUBLE PRECISION SCL,DABS
  DO 10 I=1,N
    BB(I)=E(I,K1)
    CC(I)=C(K2,I)
    DO 10 J=1,N
      AA(I,J)=A(I,J)
      WRITE (6,50) K1,K2
      IF (IFLFW.EQ. 0) GC TC 20
      H=C(K2,K1)
      IF (DAES(H).LE.EPS) GO TO 20
      JJ=N
      GO TO 50
      NN=N-1
      DO 30 I=1,NN
        H=SCL(A,EB,CC)
        CALL CCMP (N, NM, AA, CC, CP)
        IF (DAES(H).GT.EPS) GC TO 40
        CONTINUE
        H=SCL(A,EB,CC)
        WRITE (6,100) H
        GO TO 70
        JJ=N-1
        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,C2)
        CALL FLF (NM,N,LCW,IHIGH,AA,EVR,EVI,IERR)
        IF (IEFF.NE. 0) GC TC 80
        WRITE (6,120)

```

```

60 DO 60 I=1,N
70 WRITE (4,130) EVR(I),EVI(I)
80 RETURN
80 WRITE (5,140)
80 RETURN
C=====
90 FORMAT (//,17H IF FOR INPUT NC,13,15F AND OUTPUT NU.,13,1H:)
100 FORMAT (//,5X,27HNC FINITE ZEROS: IF GAIN =,E12.4)
110 FORMAT (//,3X,20HCRCLER CF NUMERATOR =,13,9X,SHIF 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 (52F FAILURE IN HQR CALCULATING TRANSFER FUNCTION ZEROES)
    1ENC
C=====
SUBROUTINE ACOMP (N,NM,A,B,C,H)
    REAL*8 A,B,C,H
    DIMENSION A(NM,N),E(N),C(N)
    DO 10 I=1,N
    DO 10 J=1,N
        A(I,J)=A(I,J)-B(I)*C(J)/H
    10 RETURN
    10ENC
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)=C
    DO 10 J=1,N
        CC(I)=(C(I)+C(J)*A(J,I))
    10 CC(1)=C(1)
    DO 20 I=1,N
        C(I)=CC(I)
    20 RETURN
    20ENC
C=====
FUNCTION SCL(N,B,C)
    REAL*8 E,C,SCL
    DIMENSION B(N),C(N)
    SCL=0.
    DO 10 I=1,N
        SCL=SCL+C(I)*B(I)
    10 RETURN
    10ENC
C=====
SUBROUTINE RESID (K1,K2,N,JCF,M,BM,L,CM,PR,PI,RES,BB,CC,IPT)
    IMPLICIT REAL*8(A-H,C-Z)
    DIMENSION JCF(N),BM(N,M),CM(L,N),PR(N),PI(N),RES(N),BB(N),CC(N),PR

```

```

17(4) SN/8P* SIN(B*T /, R1/8H * /, R2/8H*EXP(A*T) /, ED/1H) /
DATA ZFC/ C.D.O / T1 / 4H*T** /, BLANK/ 8H /, CS/ 8H* COS(B*T /
C-----TEMPORARY MOD TILL JCF IS CALCULATED-----
10 DO 10 I=1,N
C JCF(I)=C
C-----TEMPORARY MOD-----
20 IF (IPT .EQ. 1) WRITE (5,170)
C-----TEMPORARY MOD-----
C-----LOOP THROUGH THE POLES-----
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)*B8(I)+CC(I+1)*B8(I+1)
RES(I+1)=CC(I)*B8(I+1)-CC(I+1)*B8(I)
IF (IPT .EQ. 0) GO TO 40
PR(I)=ELANK
PR(I+1)=R2
PR(I+2)=R2
IF (PI(I) .EQ. 0.D0) PRT(2)=BLANK
PR(I+3)=C
PR(I+4)=C
WRITE (6,180) PR(I),PI(I),RES(I), (PRT(J),J=1,4)
I=I+1
PR(I+3)=SN
WRITE (6,180) PR(I),PI(I),RES(I), (PRT(J),J=1,4)
GO TO 30
I=I+1
GO TO 30
C-----CONTINUE-----
C-----COMPUTE SIMPLE REAL POLE RESIDUE-----
RES(I)=CC(I)*B8(I)
IF (IPT .EQ. 0) GO TO 30
PR(I)=R1
PR(I+2)=R2
PR(I+3)=ELANK
PR(I+4)=ELANK
WRITE (6,180) PR(I),PI(I),RES(I), (PRT(J),J=1,4)
GO TO 30
C-----LOCK AHEAD TO DETERMINE SIZE OF THE JORDAN BLOCK-----
60 K=1
KT=N-1
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+1)*BB(I+2)
      RES(I+3)=CC(I)*BB(I+3)-CC(I+1)*BB(I+1)*BB(I+2)
      IF (IPT .EQ. 0) GO TO 100
      PR I(1)=R1
      PR T(2)=R2
      IF (DABS(PR(I)) .GT. 1.D-10) GO TO 90
      PR T(1)=ELANK
      PR T(2)=ELANK
      PR T(3)=ELANK
      PR T(4)=ELANK
      WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
      PR T(3)=SN
      I=I+1
      WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
      PR T(1)=T1
      PR T(2)=R2
      IF (DABS(PR(I)) .LT. 1.D-10) PR T(2)=BLANK
      PR T(3)=CS
      I=I+1
      WRITE (6,190) PR(I),PI(I),RES(I),PRT(1),K,(PRT(J),J=2,4)
      PR T(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-----
      CONTINUE
      KT=I+K-1
      NN=0
      DO 130 J=I,KT
      NN=NN+1
      RES(J)=ZERO
      DO 120 J=J,KT
      RES(J)=RES(J)+BB(JJ)*CC(JJ-NN+1)
120 CONTINUE
130 IF (IPT .EQ. 0) GO TO 150
      NN=0
      PR T(1)=T1
      PR T(2)=R2

```

```

140 PR T(3)=ELANK
150 PR T(4)=ELANK
160 DO 140 J=1,KT
WRITE (6,190) PR(J),PI(J),RES(J),PRT(1),NN,(PRT(JJ),JJ=2,4)
NN=NN+1
GO TO 3C
I=KT
GO TO 3C
CONTINUE
RETURN
C-----
170 FORMAT (//,3X,22HRESIDUES AT THE POLES:/,T16,9HP O L E S,T41,15HR
180 1E S I D L E S/,T9,7HREAL(A),T26,7HIMAG(B))
190 FORMAT (/,4X,1H(,F13.6,4H)+J(,F13.6,1H),4X,1H(,F13.6,1H),5A8,A1J
190 12A8,A1)
ENC
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 ABS
LOGICAL NOCCNV
DATA RADIX/24210C0C0C000000/
C-----
B2=RADIX**RADIX
K=1
L=N
GO TO 6C
IN-LINE PROCEDURE FOR ROW AND COLUMN EXCHANGE-----
10 SCALE(M)=J
IF (J.EC. M) GO TO 40
DO 20 I=1,L
F=A(I,J)
A(I,J)=A(I,M)
A(I,M)=F
CONTINUE
DO 30 I=K,N
F=A(J,I)
A(J,I)=A(M,I)
A(M,I)=F
CONTINUE
GO TO (50,90), IEXC
C-----
40 SEARCH FOR ROWS ISCLATING AN EIGENVALUE AND PUSH THEM DOWN-----
50 IF (L.EC. 1) GO TO 230
L=L-1
DO 80 JJ=1,L

```

```

70 J=L+1-JJ
   DO 70 I=1,L
   IF (I.EQ. J) GO TC 70
   IF (A(J,I) .NE. 0.000) GO TO 80
   CONTINUE

80 M=L
   IEXC=1
   GO TO 1C
   CONTINUE
   GO TO 100
C----- SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE AND PUSH THEM LEFT-----
90 K=K+1
100 DO 120 I=K,L
   DO 110 I=K,L
   IF (I.EQ. J) GO TO 110
   IF (A(I,J) .NE. 0.000) GO TO 120
   CONTINUE
110 M=K
   IEXC=2
   GO TO 1C
   CONTINUE
C----- NOW BALANCE THE SUBMATRIX IN ROWS K TO L-----
120 DO 130 I=K,L
130 SCALE(I)=1.000
C----- ITERATIVE LOOP FOR NCRM REDUCTION-----
140 NUCCNV=.FALSE.
   DO 220 I=K,L
   C=C.ODC
   R=0.0DC
   DO 150 J=K,L
   IF (J.EQ. I) GO TC 150
   C=C+DAE(A(I,J,I))
   R=R+DAE(A(I,J))
   CONTINUE
150 IF (C.EQ. C.000 .OR. R.EQ. 0.000) GO TO 220
C----- GUARD AGAINST ZERO C OR R DUE TO UNDERFLOW-----
   IF (C.EQ. 0.000) GO TO 170
   G=R/RACIX
   F=1.0DC
   S=C+R
160 IF (C.GE. G) GO TO 170
   F=F*ACIX
   C=C*B2
   GO TO 160
170 G=R*ACIX
180 IF (C.LT. G) GO TO 190
   F=F/RACIX
   C=C/B2
   GO TO 180

```

```

C-----NCW BALANCE-----
190 IF ((C + R) / F .GE. 0.95D0 * S) GO TO 220
   G=1.0DC/F
   SCALE(I)=SCALE(I)*F
   NOCCNV=.TRUE.
   DO 200 J=K,N
     A(I,J)=A(I,J)*G
   DO 210 J=1,L
     A(J,I)=A(J,I)*F
   CONTINUE
220 IF (NOCCNV) GO TO 140
230 LOW=K
   IGH=L
   RETURN
   END
C=====
SUBROUTINE ORTHES (NM,N,LOW,IGH,A,CRT)
  INTEGER I,J,M,N,II,JJ,LA,MP,NM,IGH,KP1,LOW
  REAL*8 A(NM,N),ORT(IGH)
  REAL*8 F,G,F,SCALE
  REAL*8 CSCRT,DABS,DSIGN
  LA=IGH-1
  KP1=LOW+1
  IF (LA.LT. KP1) GC TO 190
  DO 50 M=KP1,LA
    H=0.0DC
    ORT(M)=C.ODD
    SCALE=C.CDD
    DO 10 I=M,IGH
      SCALE=SCALE+DABS(A(I,M-1))
      IF (SCALE .EQ. 0.0DC) GO TO 90
      MP=M+IGH
      DO 20 II=M, IGH
        I=MP-II
        ORT(I)=A(I,M-1)/SCALE
        H=ORT(I)*ORT(I)
      CONTINUE
      G=-DSIGN(DSCRT(H),ORT(M))
      H=F-ORT(M)*G
      ORT(M)=CRT(M)-G
    DO 50 M=N,N
      F=0.0DC
      DO 30 II=M, IGH
        I=MP-II
        F=F+ORT(I)*A(I,J)
      CONTINUE
30

```

```

F=F/H
DO 40 I=M,IGH
A(I,J)=A(I,J)-F*CRT(I)
50 CONTINUE
C-----FORM (I-(U*UT)/H)*A*(I-(U*LT)/H)-----
DO 80 I=1,IGH
F=0.0DC
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
A(I,J)=A(I,J)-F*CRT(J)
70 CONTINUE
ORT(M)=SCALE*ORT(M)
80 A(M,M-1)=SCALE*G
CONTINUE
90 RETURN
100 ENC
C=====
SUBROUTINE CRTAN (NM,N,LOW,IGH,A,ORT,Z)
INTEGER I,J,N,KL,MP,MP,NM,IGH,LCW,MP1
REAL*8 A(NM,IGH),ORT(IGH),Z(NM,N)
REAL*8 G
C-----INITIALIZE Z TO IDENTITY MATRIX-----
DO 20 I=1,N
DO 10 J=1,N
Z(I,J)=C.0DC
Z(I,I)=1.0DC
20 CONTINUE
KL=IGH-LCW-1
IF (KL.LT.1) GC TC 80
DO 70 MP=1,KL
MP=IGH-MP
IF (A(MP,MP-1) .EQ. 0.0DC) GO TO 70
MP1=MP+1
DO 30 I=MP1,IGH
CRT(I)=A(I,MP-1)
30 DO 60 J=MP,IGH
G=0.0DC
DO 40 I=MP,IGH
G=G+ORT(I)*Z(I,J)
40 CONTINUE
C-----DIVISOR BELCW IS NEGATIVE OF H FORMED IN ORTHS.-----
C-----DOUBLE DIVISION AVOIDS POSSIBLE UNDERFLOW-----
G=(G / CRT(MP))/A(MP,MP-1)
DO 50 I=MP,IGH
Z(I,J)=Z(I,J)+G*ORT(I)
50

```

```

60 CONTINUE
70 CONTINUE
80 RETURN
C=====
SUBROUTINE FQR2 (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 F(NM,N),WR(N),WI(N),Z(NM,N)
REAL*8 P,Q,R,S,T,W,X,Y,RA,SA,VI,VR,ZZ,NCRM,MACHEP
REAL*8 CSQRT,DABS,DSIGN
INTEGER PINO
LOGICAL NOTLAS
COMPLEX*16Z3
COMPLEX*16CCMPLX
REAL*8 CREAL,DIMAG
C----- STATEMENT FUNCTIONS ENABLE EXTRACTION OF REAL AND IMAGINARY-----
C----- PARTS OF DOUBLE PRECISION COMPLEX NUMBERS-----
DREAL(Z3)=Z3
DIMAG(Z3)=(C.0D0,-1.0D0)*Z3
DATA MACHEP/2341000000000000000/
IERR=0
NCRM=0.0D0
K=1
C----- STORE RCCTS ISOLATED BY BALANC AND COMPUTE MATRIX NORM-----
DO 20 I=1,N
DO 10 J=K,N
10 NORM=NCRF+CABS(H(I,J))
K=I
IF (I .GE. LOW .AND. I .LE. IGH) GO TO 20
WR(I)=F(I,I)
WI(I)=C.CD0
20 CONTINUE
EN=IGH
T=0.0DC
C----- SEARCH FOR NEXT EIGENVALUES-----
30 IF (EN .LT. LOW) GC TO 290
ITS=0
NA=EN-1
ENM2=NA-1
C----- LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT-----
40 DO 50 LL=LCH,EN
L=EN+LC-LL
IF (L .EQ. LOW) GO TO 60
S=CABS(F(L-1,L-1))+CABS(H(L,L))
IF (S .EQ. 0.0D0) S=NCRM
IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 60
50 CONTINUE

```

```

C-----FORM SHIFT-----
60 X=H(EN,EN)
   IF (L.EC. EN) GO TC 220
   Y=H(NA,NA)
   W=H(EN,NA)*F(NA,EN)
   IF (L.EC. NA) GO TC 230
   IF (ITS.EQ. 30) GO TO 500
   IF (ITS.NE. 10) .AND. ITS.NE. 20) GO TC 80
C-----FORM EXCEPTIONAL SHIFT-----
   T=T+X
   DO 70 I=LCH,EN
   H(I,I)=F(I,I)-X
   S=LABS(F(EN,NA))+DABS(F(NA,ENM2))
   X=0.75DC*S
   Y=X
   W=-0.4375DC*S*S
   ITS=ITS+1
70 C-----LOCK FOR TWO CONSECUTIVE SMALL SUB-DIAGNAL ELEMENTS.-----
   DO 80 MP=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)) .LF. 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)=0.0DC
   IF (I.EC. MP2) GO TO 110
   H(I,I-3)=0.0DC
110 CONTINUE
C-----DOUBLE QR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN-----
   DO 210 K=M,NA
   NO TLAS=K.NE.NA
   IF (K.EC. M) GO TC 120
   P=H(K,K-1)
   Q=H(K+1,K-1)
   R=C.0DC
   IF (NCTLAS) R=H(K+2,K-1)

```

```

X=DABS(F)+DABS(Q)+DABS(R)
IF (X .EC. 0.0D0) GO TO 210
P=P/X
Q=Q/X
R=R/X
S=DSIGN(DSQRT(P*P+Q*Q+R*R),P)
IF (K .EC. M) GO TO 130
H(K,K-1)=-S*X
GO TO 140
.NE. M) H(K,K-1)=-H(K,K-1)
120 P=P+S
130 X=P/S
140 Y=Q/S
ZZ=R/S
Q=Q/P
R=R/P
C-----ROW MODIFICATION-----
DO 160 J=K,N
P=H(K,J)+C*H(K+1,J)
IF (.NOT. NOTLAS) GC TO 150
P=P+R*H(K+2,J)
H(K+2,J)=H(K+2,J)-P*ZZ
H(K+1,J)=H(K+1,J)-P*Y
H(K,J)=H(K,J)-P*X
CONTINUE
J=MINO(EN,K+3)
C-----COLUMN MODIFICATION-----
DO 180 I=1,J
P=X*H(I,K)+Y*H(I,K+1)
IF (.NOT. NOTLAS) GC TO 170
P=P+ZZ*H(I,K+2)
H(I,K+2)=H(I,K+2)-P*R
H(I,K+1)=H(I,K+1)-P*Q
H(I,K)=H(I,K)-P
CONTINUE
C-----ACCUMULATE TRANSFORMATIONS-----
DO 200 I=LCM,IGH
P=X*Z(I,K)+Y*Z(I,K+1)
IF (.NOT. NOTLAS) GO TO 190
P=P+ZZ*Z(I,K+2)
Z(I,K+2)=Z(I,K+2)-P*R
Z(I,K+1)=Z(I,K+1)-P*Q
Z(I,K)=Z(I,K)-P
CONTINUE
GO TO 4C
C-----ONE ROOT FOUND-----
220 H(EN,EN)=X+T

```

```

      WR(EN)=F(EN,EN)
      WI(EN)=C.OOO
      EN=NA
      GO TO 30
C-----TWO ROOTS FOUND-----
230  P=(Y - X)/2.OOO
      C=P*P+W
      ZZ=DSQRT(DABS(Q))
      H(EN,EN)=X+T
      X=H(EN,EN)
      H(NA,NA)=Y+T
      IF (Q .LT. C.OOO) GO TO 270
C-----REAL PAIR-----
      ZZ=P+CSIGN(ZZ,P)
      WR(NA)=X+ZZ
      WR(EN)=WR(NA)
      IF (ZZ .NE. O.OOO) WR(EN)=X-W/ZZ
      WI(NA)=C.OOO
      WI(EN)=C.OOO
      X=F(EN,NA)
      S=CABS(X)+CABS(ZZ)
      P=X/S
      C=ZZ/S
      R=DSQRT(F*P+Q*Q)
      P=P/R
      C=C/R
C-----ROW MODIFICATION-----
      DO 240 J=NA,N
      ZZ=H(NA,J)
      H(NA,J)=C*ZZ+P*H(EN,J)
      H(EN,J)=C*H(EN,J)-P*ZZ
240  CONTINUE
C-----COLUMN MODIFICATION-----
      DO 250 I=1,EN
      ZZ=H(I,NA)
      H(I,NA)=C*ZZ+P*H(I,EN)
      H(I,EN)=C*H(I,EN)-P*ZZ
250  CONTINUE
C-----ACCUMULATE TRANSFORMATIONS-----
      DO 260 I=LON,IGH
      ZZ=Z(I,NA)
      Z(I,NA)=C*ZZ+P*Z(I,EN)
      Z(I,EN)=C*Z(I,EN)-P*ZZ
260  CONTINUE
      GO TO 2EC
C-----COMPLEX PAIR-----
270  WR(NA)=X+P
      WR(EN)=X+P

```

```

280      WI(NA)=ZZ
      WI(EN)=-ZZ
      EN=ENM2
      GO TO 3C
C-----ALL ROOTS FOUND: BACKSUBSTITUTE TO FIND-----
C-----VECTORS OF UPPER TRIANGULAR FORM-----
290      IF (NORM .EQ. 0.000) GC TO 510
      DO 450 NA=1,N
      EN=N+1-NA
      P=WR(EN)
      Q=WI(EN)
      NA=EN-1
      IF (Q) 370,300,450
C-----REAL VECTOR-----
300      M=EN
      H(EN,EN)=1.000
      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
      R=R+H(I,J)*P(J,EN)
310      IF (WI(I) .GE. 0.000) GO TO 330
320      ZZ=W
      Z=R
      GO TO 360
330      M=I
      IF (WI(I) .NE. 0.000) GO TC 340
      T=W
      IF (W .EQ. 0.000) T=MACHEP*NORM
      H(I,EN)=-R/T
      GO TO 360
C-----SOLVE REAL EQUATIONS-----
340      X=H(I,I+1)
      Y=H(I+1,I)
      Q=(WR(I)-P)+WI(I)*WI(I)
      T=(X*Y-ZZ*Y)/Q
      H(I,EN)=T
      IF (DABS(X) .LE. DABS(ZZ)) GO TO 350
      H(I+1,EN)=(-R-W*T)/X
      GO TO 360
350      H(I+1,EN)=(-S-Y*T)/ZZ
360      CONTINUE
C-----END REAL VECTOR-----
C-----COMPLEX VECTOR-----
      GO TO 450

```

```

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)=C/H(EN,NA)
H(NA,EN)=-P/H(EN,NA)
GO TO 390
380 Z3=DCMPLX(0.0D0,-H(NA,EN))/DCMPLX(H(NA,NA)-F,Q)
H(NA,NA)=DREAL(Z3)
H(NA,EN)=CIPAG(Z3)
390 H(EN,NA)=0.0D0
H(EN,EN)=1.0D0
ENP2=NA-1
IF (ENP2 .EQ. 0) GC TO 450
DO 440 I=1,ENM2
I=NA-I
M=H(I,I)-P
RA=0.0D0
SA=H(I,EN)
DO 400 J=M,NA
RA=RA+H(I,J)*H(J,NA)
SA=SA+H(I,J)*H(J,EN)
CONTINUE
400 IF (W(I)) .GE. 0.0D0) GO TO 410
ZZ=W
R=RA
S=SA
GO TO 440
410 M=I
IF (W(I)) .NE. 0.0D0) GO TO 420
Z3=DCMPLX(-RA,-SA)/DCMPLX(W,Q)
H(I,NA)=DREAL(Z3)
H(I,EN)=CIMAG(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)+W(I)*W(I)-C*Q
VI=(WR(I)-P)*2.0D0*C
IF (VR .EQ. 0.0D0 .AND. VI .EQ. 0.0D0) VR=MACHEP*NORM*(DABS(W)+D
1ABS(Q)+DABS(X)+DABS(Y)+DABS(Z3))
Z3=DCMPLX(X*VR-Z*RA+Q*SA,X*S-ZZ*SA-C*RA)/DCMPLX(VR,VI)
H(I,NA)=DREAL(Z3)
H(I,EN)=CIMAG(Z3)
IF (DABS(X) .LE. DABS(Z3)+DABS(Q)) GC TO 430
H(I+1,NA)=-RA-W*H(I,NA)+Q*H(I,EN))/X
H(I+1,EN)=-SA-W*H(I,EN)-Q*H(I,NA))/X
GO TO 440

```

```

430 Z3=DCMPLX(-R-Y*H(I,NA),-S-Y*H(I,EN))/DCMPLX(ZZ,Q)
H(I+1,NA)=DREAL(Z3)
H(I+1,EN)=DIMAG(Z3)
CONTINUE
C-----END COMPLEX VECTOR-----
440 CONTINUE
C-----END BACK SUBSTITUTION. VECTORS OF ISOLATED ROOTS-----
450 CONTINUE
DO 470 I=1,N
IF (I .GE. LOW .AND. I .LE. IGH) GC TO 470
DO 460 J=I,N
Z(I,J)=F(I,J)
CONTINUE
C-----MULTIPLY BY TRANSFORMATION MATRIX TO GIVE-----
C-----VECTORS OF ORIGINAL FULL MATRIX.-----
DO 490 J=LCW,N
J=N+LOW-JJ
M=MINO(J,IGH)
DO 490 I=LCW,IGH
ZZ=0.0CC
DO 480 K=LOW,M
ZZ=ZZ+Z(I,K)*H(K,J)
Z(I,J)=ZZ
CONTINUE
GO TO 510
C-----SET ERROR-->NO CONVERGENCE TO AN-----
C-----EIGENVALUE AFTER 30 ITERATIONS-----
500 IERR=EN
510 RETURN
END
C=====
SUBROUTINE HALBAK (NM,N,LOW,IGH,SCALE,M,Z)
INTEGER I,J,K,M,N,II,NM,IGH,LOW
REAL*8 SCALE(N),Z(NP,M),S
IF (M .EQ. 0) GO TO 60
IF (IGH .EQ. LOW) GC TC 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=1.000/SCALE(I).-----
DO 10 J=1,M
Z(I,J)=Z(I,J)*S
CONTINUE
DO 50 II=1,N
I=II
IF (I .GE. LOW .AND. I .LE. IGH) GC TO 50
IF (I .LT. LOW) I=LOW-II
K=SCALE(I)

```

```

IF (K .EQ. 1) GO TO 50
DO 40 J=1,M
S=Z(I,J)
Z(I,J)=Z(K,J)
Z(K,J)=S
CONTINUE
RETURN
END

```

40
50
60

```

C=====
SUBROUTINE PQR (NM,N,LCW,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 P,Q,R,S,T,W,X,Y,ZZ,NORM,MACHEP
REAL*8 CSORT,DABS,DSIGN
INTEGER MINO
LOGICAL NOTLAS
DATA MACHEP/Z3410000000000000000/
IERR=0
NORM=0.CDO
K=1

```

C----- STCFE ROOTS ISOLATED BY BALANC AND COMPUTE MATRIX VORM-----

```

DO 20 I=1,N
DO 10 J=K,N
NORM=NLCFM+DABS(H(I,J))
K=I
IF (I .GE. LOW .AND. I .LE. IGH) GO TO 20
WR(I)=H(I,I)
WI(I)=C.CDO
CONTINUE
EN=IGH
T=0.0DC

```

C----- SEARCH FOR NEXT EIGENVALUES-----

```

30 IF (EN .LT. LOW) GC TO 250
IT S=0
NA=EN-1
ENM2=NA-1

```

C----- LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT-----

```

40 DO 50 LL=LCH,EN
L=EN+LCH-LL
IF (L .EQ. LOW) GO TO 60
S=DABS(H(L-1,L-1))+DABS(H(L,L))
IF (S .EQ. C.CDO) S=NORM
IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 60
CONTINUE

```

C----- FORM SHIFT-----

```

50 X=H(EN,EN)
60 IF (L .EQ. EN) GO TO 200

```

```

C-----
Y=H(NA,NA)*F(NA,EN)
H=H(EN,NA)*F(NA,EN)
IF (L.EC.NA) GO TO 210
IF (ITS.EC.30) GC TC 240
IF (ITS.NE.10.AND.ITS.NE.20) GO TC 80
FORM EXCEPTIONAL SHIFT-----
T=T+X
DO 70 I=LOW,EN
H(I,I)=F(I,I)-X
S=CABS(H(EN,NA))+DABS(H(NA,ENM2))
X=0.75E0+S
Y=X
W=-0.4375E0*S*S
ITS=ITS+1
LCCK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS.-----
DO 90 MN=L,ENM2
M=ENM2+L-MN
ZZ=H(M,M)
R=X-ZZ
S=Y-ZZ
P=(R-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.EC.L) GO TO 100
IF (DABS(H(M,M-1))*(DABS(Q)+DABS(R))+LE.MACHEP*DABS(P)
1 * (DABS(H(M-1,M-1))+DABS(ZZ)+DABS(H(M+1,M+1)))) GO TO 100
CONTINUE
MP2=M+2
DO 110 I=MP2,EN
H(I,I-2)=0.000
IF (I.EC.MP2) GO TO 110
H(I,I-3)=0.000
CONTINUE
DOUBLE QR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN-----
DO 190 K=M,NA
NO TLAS=K.NE.NA
IF (K.EC.M) GO TC 120
P=F(K,K-1)
Q=H(K+1,K-1)
R=C.000
IF (NC TLAS) R=H(K+2,K-1)
X=DABS(P)+DABS(Q)+DABS(R)
IF (X.EC.C.000) GO TO 190
P=P/X

```

```

120 Q=Q/X
    R=R/X
    S=DSQRT(P*P+C*Q+R*R),P)
    IF (K-1) GO TC 130
    H(K,K-1)=-S*X
    GO TO 140
130 IF (L-1) H(K,K-1)=-H(K,K-1)
140 P=P+S
    X=P/S
    Y=C/S
    ZZ=R/S
    Q=C/P
    R=R/P
C-----ROW MODIFICATION-----
    DO 160 J=K,EN
    P=H(K,J)+C*F(K+1,J)
    IF (.NOT. NCTLAS) GO TC 150
    P=P+R*H(K+2,J)
    H(K+2,J)=H(K+2,J)-P*ZZ
    H(K+1,J)=H(K+1,J)-P*Y
    H(K,J)=H(K,J)-P*X
    CONTINUE
    J=MINO(EA,K+3)
C-----COLUMN MODIFICATION-----
    DO 180 I=L,J
    P=X*H(I,K)+Y*H(I,K+1)
    IF (.NOT. NCTLAS) GO TO 170
    P=P+ZZ*F(I,K+2)
    H(I,K+2)=H(I,K+2)-P*R
    H(I,K+1)=H(I,K+1)-P*Q
    H(I,K)=H(I,K)-P
    CONTINUE
    CONTINUE
    GO TO 40
C-----ONE ROOT FOUND-----
200 WR(EN)=X+T
    W(EN)=C.0DC
    EN=NA
    GO TO 30
C-----TWO ROOTS FOUND-----
210 P=(Y-X)/2.0D0
    Q=P*P+K
    ZZ=DSQRT(DABS(Q))
    X=X+T
    IF (Q.LT. C.0D0) GO TC 220
    ZZ=P+LSIGN(ZZ,P)
    WR(NA)=X+ZZ
C-----REAL PAIR-----

```

```

WR(EN)=WR(NA)
IF (ZZ.NE.0.000) WR(EN)=X-W/ZZ
WI(NA)=C.0DC
WI(EN)=C.0DC
GO TO 230
-----COMPLEX PAIR-----
220 WR(NA)=X+P
    WR(EN)=X+P
    WI(NA)=ZZ
    WI(EN)=-ZZ
    EN=ENM2
    GO TO 30
230 -----SET ERRCR--- NO CONVERGENCE TO AN-----
    -----EIGENVALUE AFTER 30 ITERATIONS-----
240 IERR=EN
250 RETURN
    ENC
=====
1 SUBROUTINE PSDCAL (N2,NS,FA,X,NC,GW,GV,C,NC,HY,HU,H,
2 FBGE,NG,GAM,ACL,F,WR,WI,DI,D2,JCF,RES,Q,R,BB,CC,IYU,
3 IPSD,INORM)
=====
    PSDCAL COMPUTES THE PSD OF OUTPUTS OR CONTROLS OF
    A CONTROLLED SYSTEM
=====
    IYL= 1      OUTPUT PSD
    = 2      CCTRL PSD
    = 3      BOTH OUTPUT AND CONTROL PSD
    IPSD=1      PSD
    =2      PSD AND TF RESIDUES
    INCRM= 1,2,... NG NORMALIZED BY ITH PROCESS NOISE
    NG+1,... NG+NC NORMALIZED BY ITH MEAS NOISE
=====
    DOUBLE PRECISION FA,X,GW,GV,C,HY,H,FBGE,GAM,ACL,F,WR,WI,DI,D2,RES,
188 CC,C,R,PSD,W,DNORM,DNI,EMAX,ELUG,EMCC,DW,ST,DM,RE,A,I,HU,DW1
    COMPLEX X #16ZD,ZN,ZZ
    DIMENSION FA(N2,N2),X(N2,N2),GW(N2,NG),C(NG,NS),HY(NG,N2),H(NG,NS)
1,FBGE(NS,NO),GAM(NS,NG),ACL(NS,NS),F(NS,NS),WR(N2),WI(N2),DI(N2),D
22(N2),RES(N2),Q(NG,NG),R(NG,NO),PSD(30),W(30),BB(N2),CC(N2),GV(N2,
3NO),HU(AC,N2),DW1(4)
    INTEGER JCF(N2)
    DATA DW1/1.0,2.0,5.0,10.0/
    IF (IYL.EQ.0) IYL=1
    IF (INCRM.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
NSC=N2*N2
C-----COMPUTE EIGENSYSTEM OF CONTROLLED SYSTEM; FORM FA-----
DO 10 I=1,NS
DO 10 J=1,NS
FA(I,J)=ACL(I,J)
FA(NS+I,J)=0.DO
DO 30 I=1,NS
DO 30 J=1,NS
ST=0.DO
DO 20 K=1,NC
ST=ST+EEGE(I,K)*H(K,J)
FA(I,NS+J)=-ST
C 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,LCM,IHIGH,D1)
CALL ORTHES (N2,N2,LOW,IHIGH,FA,D2)
CALL ORTRAN (N2,N2,LOW,IHIGH,FA,D2,X)
CALL HCF2 (N2,N2,LCM,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-----DETERMINE MCDAL MATRICES-----
IF (IYU.EQ. 1) GO TO 60
C-----HSEBU-----
DO 50 I=1,NC
DO 50 J=1,N2
ST=0.DO
DO 40 K=1,NS
ST=ST-C(I,K)*X(K,J)
HU(I,J)=ST
GO TO 50
C-----HSEBY-----
DO 60 I=1,NC
DO 80 J=1,N2
ST=0.DO
DO 70 K=1,NS
ST=ST+H(I,K)*X(K,J)-H(I,K)*X(NS+K,J)
HY(I,J)=ST
CALL RAPRNT (NO,NO,N2,9,HY,4,(9(1X,1PD13.6)))
C-----DEBUG ABOVE-----
CALL MINV (NSQ,X,N2,ST,D1,D2)
C 90 CALL RAPRNT (N2,N2,N2,5,X,4,(5(1X,1PD13.6)))
C-----DEBUG ABOVE-----

```

```

C-----GSUBM-----
DO 110 I=1,N2
DO 110 J=1,NG
ST=0.000
DO 100 K=1,AS
SW=ST-X(I,NS+K)*GAM(K,J)
100  CALL RAIPRNT (N2,N2,NG,5,GW,4,'(9(1X,1PD13.6))')
110  C-----DETERMINE BANDWIDTH OF CONTROLLED SYSTEM-----
C-----
      IF (INCRM .LE. NG) DNORM=1.00/C(INCRM,INCRM)
      IF (INORM .GT. NG) DNORM=1.00/R(INORM-NG,INCRM-NG)
      EMAX=0.00
DO 120 I=1,N2
EMCD=DAES(WR(I)**2 +WI(I)**2)
IF (EMCD .GT. EMAX) EMAX=EMCD
120  CONTINUE
EMCD=DSQRT(EMAX)
EMCD=2*EMCD
C-----RCUND UP TO NEAREST 2,4,5,8,10-----
ELCG=DLCG10(EMOD)
IF (ELCG .LT. 0.00) IPOW=-IDINT(DABS(ELCG) + 1)
IF (ELCG .GE. 0.00) IPOW=ICINT(ELCG)
EMAX=EMCD*10**(-IPOW)
IF (EMAX .GT. 2.00) EMCD=2.00
IF (EMAX .GT. 4.00) EMCD=4.00
IF (EMAX .GT. 5.00) EMCD=5.00
IF (EMAX .GT. 8.00) EMCD=8.00
IF (EMAX .GT. 10.00) EMCD=10.00
EMAX=EMCD*10**IPOW
OW=EMAX/20.00
C-----ADD 10 POINTS 3 DECADES UP-----
IF (EMCD .LT. 5.0) GO TO 130
EMAX=1.001
IK=3
GO TO 140
EMAX=5.00
130  IK=2
140  CONTINUE
C-----STORE 30 FREQUENCIES-----
DO 150 I=1,20
W(I)=OW*(I-1)
DO 160 J=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

```

```

160      W(IP+JJ)=CW1(IX)*10**((IPGW+I-1+JJ+IK-2)
CONTINUE
IX=MOD(IK,3)+1
W(30)=CW1(IX)*10**((IPGW+3+IK-2)
C-----LARGE LOOP THRU OUTPUTS-----
      IF (IYL .EQ. 1) NL=NO
      IF (IYL .EQ. 2) NL=NC
      DO 310 L=1,NL
      CO 170 I=1,30
      PSD(I)=C.D0
C-----LOOP THRU PROCESS NOISE-----
      DO 220 I=1,NG
      DN1=DNCRM*(I, I)
      IF (IYL .EQ. 1) .AND. IPT .EQ. 1) WRITE (6,350) I,L
      IF (IYL .EQ. 2) .AND. IPT .EQ. 1) WRITE (6,380) I,L
      IF (IYL .EQ. 1) CALL RESID (I,L,N2,JCF,NG,GW,NL,HY,WR,WI,
1RES,BB,CC,IPT)
      IF (IYL .EQ. 2) CALL RESID (I,L,N2,JCF,NG,GW,NL,HU,WR,WI,
1RES,BB,CC,IPT)
      CO 210 K=1,20
      ZZ=DCMFLX(0.D0,0.D0)
      CM=W(K)
      DO 200 II=1,N2
      IF (W(II)) 200,180,190
      ZD=DCMFLX(-WR(II),CM-WI(II))
      ZZ=RES(II)/ZD+ZZ
      GO TO 200
      RE=WR(II)
      AI=WI(II)
      ZD=DCMFLX(RE**2+AI**2-QM**2,-2*CO*RE*QM)
      ZN=DCMFLX(RES(II+1)*AI-RES(II)*RE,RES(II)*QM)
      ZZ=ZZ+ZN/ZD
CONTINUE
      PSD(K)=FSC(K)+DN1*(ZZ*DCONJG(ZZ))
C-----GSUBV-----
      DO 240 I=1,N2
      DO 240 J=1,NO
      ST=0.D0
      DU 230 K=1,NS
      ST=ST+X(I,K)*FBGE(K,J)+X(I,NS+K)*FBGE(K,J)
      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=DNCRM*(I, I)
      IF (IYL .EQ. 1) .AND. IPT .EQ. 1) WRITE (6,370) I,L
      IF (IYL .EQ. 2) .AND. IPT .EQ. 1) WRITE (6,380) I,L

```

```

IF (IYL.EQ.1) CALL RESID (I,L,N2,JCF,NQ,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=DCMFLX(0.00,0.00)
CM=W(K)
DU 270 II=1,N2
IF (WI(II)) 270,250,260
250  ZD=DCMFLX(-WR(II),CM-WI(II))
ZZ=ZZ+RES(II)/ZD
GO TO 270
RE=WR(II)
AI=WI(II)
ZD=DCMFLX(RE**2 + AI**2 -OM**2,-2.DJ*RE*OM)
ZN=DCMFLX(RES(II+1)*AI-RES(II)*RE,RES(II)*OM)
ZZ=ZZ+ZN/ZD
CONTINUE
IF (IYL.EQ.2 .CR. 1 .NE. LJ) GO TO 280
PSC(K)=FSC(K)+DNI
PSC(K)=FSC(K)+DNI*(ZZ*DCONJG(ZZ))
CONTINUE
CONTINUE
IF (IYL.EQ.1) WRITE (6,390) L
IF (IYL.EQ.2) WRITE (6,400) L
WRITE (6,410) (W(I),PSC(I),I=1,30)
CONTINUE
RETURN
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 NO.,I3
1,/)
350  FORMAT (/,38F TRANSFER FUNCTION FROM PRCESS NOISE ,I2,3H TG,13H ME
1ASUREMENT ,I2,/)
360  FORMAT (/,38F TRANSFER FUNCTION FROM PRCESS NOISE ,I2,3H TG,9H CON
1TRCL ,I2,/)
370  FORMAT (/,36F TRANSFER FUNCTION FROM MEASUREMENT ,I2,16H TU MEASURE
1MENT ,I2,/)
380  FORMAT (/,36F TRANSFER FUNCTION FROM MEASUREMENT ,I2,12H TO CONTROL
1,I2,/)
390  FORMAT (/,14F PSD OF OUTPUT,I3,32H FORCED BY ALL NOISE -(RAD FREQ,,
115HNORMALIZED PSC)/)
400  FORMAT (/,15H PSD OF CCNTROL,I3,32H FORCED BY ALL NOISE -(RAD FREQ,
1,15HNORMALIZED PSD)/)
410  FORMAT (4(IX,1H,E11.4,1H,E11.4,1F))

```

```

=====
C=====
ENC
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 RAFRNT (N,N,N,5,A,4,'(9(1X,1PD13.6))')
RETURN
FORMAT (35H FAILURE IN HQR2 ON EIGENVALUE NC. ,I3)
ENC
10
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,IISAF
DATA IY,YI,IIZ,NZ/
IF (IISAF.EQ.1) GO TO 40
WRITE (5,130)
DO 20 I=1,NS
DO 10 J=1,NS
WRITE (5,120) I,J
CALL RDEAL (ANSR)
BA(I,J)=ANSR
CONTINUE
CONTINUE
10
20
C-----
CALL FRICMS ('CLRSCRN ')
CONTINUE
WRITE (5,140)
CALL MATPRT (BA,NS,NS)
50
WRITE (5,150)
CALL RICHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
GO TO 70
WRITE (5,160)
GO TO 50
CONTINUE
IF (IANS.EQ.IZ) GO TO 110
IF (IANS.EQ.IY) GO TO 80
WRITE (5,170)
CALL RDCINT (IANS)
K=IANS
80
WRITE (5,180)

```

```

CALL RCINT (IANS)
L=IANS
WRITE (5,120) K,L
CALL RDEAL (ANSR)
DUM=ANSR
DO 100 I=1,NS
DO 90 J=1,NS
IF ((I.EC.K).AND.(J.EQ.L)) BA(I,J)=DUM
CONTINUE
GO TO 30
CONTINUE
CALL FRICMS ('CLSRCN ')
RETURN
C-----
120 FORMAT (5X,14H THE ELEMENT F(I,I2,1H,I2,2H)=)
130 FORMAT (/,5X,36H ENTER THE SYSTEM MATRIX "F"-MATRIX &,/,10X,41H DIM
130 LENSION = # STATES NS& X # STATES NS&)
140 FORMAT (/,15X,33H THE SYSTEM MATRIX "F"-MATRIX &,/,/)
150 FORMAT (/,5X,19H YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
150 ENT?/,10X,19H TYPE "YES" CR "NO".)
160 FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
170 FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
180 FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
180 I.)
ENC
C=====
SUBROUTINE READH (NG,NS,ISAH,H0)
INTERACTIVELY INPUTS THE "H" MATRIX MEASUREMENT SCALING MATRIX &.
C=====
REAL*8 FO(NO,NS),DUM,ANSR
INTEGER IANS,I,J,K,L,ISAH
DATA IV,Y',I2,N'
C-----
THIS IS AN EXAMPLE OF ONE POSSIBLE METHOD OF ARRAY GENERATION
WITHIN THE PROGRAM ITSELF. FOR VERY LARGE DATA ARRAYS, THIS METHOD
MAY BE PREFERABLE TO SOME USERS OVER INTERACTIVE ENTRY OF EACH
INDIVIDUAL ELEMENT.
C-----
DO 2 I=1,11
DO 1 J=1,62
FO(I,J) = 0.0D+00
FO(1,1) = 0.11520D+00
FO(2,75) = 0.5730D+02
FO(3,74) = 0.1000D+01
FO(4,63) = 0.5730D+02
FO(5,62) = 0.1000D+01
FO(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
C3     GO TO 50
C-----
C      IF (ISAF.EQ.1) GO TO 40
C      WRITE (5,12C)
C      DO 20 I=1,NC
C      DO 10 J=1,NS
C      WRITE (5,11C) I,J
C      CALL RCFEAL (ANSR)
C      HO(I,J)=ANSR
C      CONTINUE
C      CONTINUE
C-----
C      CALL FRTCMS ('CLRSCRN ')
C      CONTINUE
C      WRITE (5,13C)
C      CALL MATPRF (HG,NO,NS)
C      WRITE (5,14C)
C      CALL RCFPAR (IANS)
C      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
C      GO TO 70
C      WRITE (5,15C)
C      GO TO 50
C      CONTINUE
C      IF (IANS.EQ.IZ) GO TO 100
C      WRITE (5,16C)
C      CALL RCLINT (IANS)
C      K=IANS
C      WRITE (5,17C)
C      CALL RCLINT (IANS)
C      L=IANS
C      WRITE (5,11C) K,L
C      CALL RCFEAL (ANSR)
C      DUM=ANSR
C      DO 90 I=1,NO
C      DO 80 J=1,NS
C      IF ((I.EQ.K).AND.(J.EQ.L)) HO(I,J)=DUM
C      CONTINUE
C      CONTINUE
C      GO TO 30
C      CONTINUE

```

```

CALL FRTCMS ('CLRSCRN ')
RETURN
-----
110 FORMAT (5X,14H THE ELEMENT H(I,I2,1H,I2,2H)=)
120 FORMAT (//,5X,50H ENTER THE MEASUREMENT SCALING MATRIX "H"-MATRIX&
130 1//,10X,47H DIMENSION = # OBSERVATIONS NO& X # STATES NS&)
140 FORMAT (//,10X,46H THE MEASUREMENT SCALING MATRIX "H"-MATRIX&...//
150 1//)
160 FORMAT (//,5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
170 1ENT?//,10X,19H TYPE "YES" CR "NO".)
180 FORMAT (//,1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
190 FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
200 1)
210 FORMAT (5X,52H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
220 1)
230 1)
240 1)
250 1)
260 1)
270 1)
280 1)
290 1)
300 1)
310 1)
320 1)
330 1)
340 1)
350 1)
360 1)
370 1)
380 1)
390 1)
400 1)
410 1)
420 1)
430 1)
440 1)
450 1)
460 1)
470 1)
480 1)
490 1)
500 1)
510 1)
520 1)
530 1)
540 1)
550 1)
560 1)
570 1)
580 1)
590 1)
600 1)
610 1)
620 1)
630 1)
640 1)
650 1)
660 1)
670 1)
680 1)
690 1)
700 1)
710 1)
720 1)
730 1)
740 1)
750 1)
760 1)
770 1)
780 1)
790 1)
800 1)
810 1)
820 1)
830 1)
840 1)
850 1)
860 1)
870 1)
880 1)
890 1)
900 1)
910 1)
920 1)
930 1)
940 1)
950 1)
960 1)
970 1)
980 1)
990 1)
1000 1)

```

```

100 WRITE (5,130) K,L
110 CALL RCFREAL (ANSR)
120 DUM=AN
130 DO 80 I=1,NC
140 DO 70 J=1,NC
150 IF ((I.EQ.K).AND.(J.EQ.L)) D(I,J)=DUM
160 CONTINUE
170 GO TO 30
180 CONTINUE
190 CALL FRICMS ('CLRSCRN ')
200 RETURN
210
220 FORMAT (5X,14H THE ELEMENT D(I,12,1H,I,12,2H)=)
230 FORMAT (//,5X,54H ENTER THE MEASUREMENT FEEDTHROUGH MATRIX / FEEDFOR
240 1WARD,/,5X,34H DISTRIBUTION MATRIX "D"-MATRIX&.,//,8X,49H DIMENSION
250 2 = # OBSERVATIONS NO& X # CONTROLS NC&)
260 FORMAT (//,5X,50H THE FEEDFORWARD DISTRIBUTION MATRIX "D"-MATRIX&
270 1.,//)
280 FORMAT (//,5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
290 1ENT?,//,10X,19H TYPE "YES" OR "NO".)
300 FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
310 FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
320 FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
330 1.)
340 ENC
350
360 SUBROUTINE READG (NS,NC,ISAG,G)
370 INTERACTIVELY INPUTS THE "G" MATRIX CONTROL DISTRIBUTION MATRIX&=
380
390 REAL*8 G(NS,NC),DUM,ANSR
400 INTEGER I,ANS,I,J,K,L,ISAG
410 DATA IY,Y,/,I Z,N,/,
420 IF (ISAG.EQ.1) GO TO 40
430 WRITE (5,120)
440 DO 20 I=1,NS
450 DO 10 J=1,NC
460 WRITE (5,110) I,J
470 CALL RCFREAL (ANSR)
480 G(I,J)=ANSR
490 CONTINUE
500 CONTINUE
510
520 CALL FRICMS ('CLRSCRN ')
530 CONTINUE
540 WRITE (5,130)
550 CALL MATPART (G,NS,NC)
560 WRITE (5,140)

```

```

60 CALL RCCPAR (IANS)
   IF ((IANS.NE.IV).AND.(IANS.NE.IZ)) GO TO 60
   WRITE (5,150)
   GO TO 70
70 CONTINUE
   IF (IANS.EQ.IZ) GO TO 100
   WRITE (5,160)
   CALL RLINT (IANS)
   K=IANS
   WRITE (5,170)
   CALL RLINT (IANS)
   L=IANS
   WRITE (5,110) K,L
   CALL RLREAL (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 FRICMS ('CLRSCRN ')
   RETURN
-----
110 FORMAT (5X,14H THE ELEMENT G(I2,1H,I2,2H)=)
120 FORMAT (//,5X,51H ENTER THE CONTROL DISTRIBUTION MATRIX "G"-MATRIX &
1//,1CX,43H DIMENSION = # STATES NS X # CONTROLS NC &
130 FORMAT (//,1CX,47H THE CONTROL DISTRIBUTION MATRIX "G"-MATRIX &...
1//)
140 FORMAT (//5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT? //,10X,19H TYPE "YES" OR "NO".)
150 FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
160 FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
170 FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
1.)
   ENC
C=====
SUBROUTINE READFB (NC,NS,FBG)
C INPUTS THE "C" FEEDBACK GAIN CONTROL MATRIX &.
C=====
REAL*8 FBGC(NC,NS),DUM,ANSR
INTEGER IANS,I,J,K,L
DATA IV,Y,/,I2/,N:/
WRITE (5,110)
DO 20 I=1,NC
DO 10 J=1,NS

```

```

10  WRITE (5,10C) I,J
20  CALL RCREAL (ANSR)
30  FBGC(I,J)=ANSR
40  CONTINUE
50  CONTINUE
60  CALL FRTCMS ('CLRSCRN ')
70  WRITE (5,12C)
80  CALL MATFRT (FBGC,NC,NS)
90  WRITE (5,13C)
100  CALL RCFAR (IANS)
110  IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
120  GO TO 60
130  WRITE (5,140)
140  GO TO 40
150  CONTINUE
160  IF (IANS.EQ.IZ) GO TO 90
170  WRITE (5,15C)
180  CALL RLINT (IANS)
190  K=IANS
200  WRITE (5,16C)
210  CALL RLINT (IANS)
220  L=IANS
230  WRITE (5,10C) K,L
240  CALL RCFEAL (ANSR)
250  DUM=ANSR
260  DO 80 I=1,NC
270  DO 70 J=1,NS
280  IF ((I.EC.K).AND.(J.EQ.L)) FBGC(I,J)=DUM
290  CONTINUE
300  CONTINUE
310  GO TO 30
320  CALL FRTCMS ('CLRSCRN ')
330  RETURN
340  FORMAT (5X,14H THE ELEMENT C(I,12,1H,I2,2H)=)
350  FORMAT (//,5X,52H ENTER THE FEEDBACK GAIN CONTROL MATRIX "C"--MATRIX
360  1X,10X,44H DIMENSION = # CONTROLS NC&X # STATES NS&.)
370  FORMAT (//,10X,45H THE FEEDBACK GAIN CONTROL MATRIX "C"--MATRIX&,//
380  1)
390  FORMAT (//,5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
400  1EN?,//,10X,19H TYPE "YES" OR "NO".)
410  FORMAT (//,10X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
420  FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
430  FORMAT (5X,53H ENTER THE COLUMN NUMBER CF THE ELEMENT TO BE CHANGED
440  1.)
450  END

```

```

=====
C SUBROUTINE READAY (NG,ISAA,AY)
C INPUTS THE "A" MATRIX DIAGONAL OUTPUT COST MATRIX&
C =====
      REAL*8 AY(NG,NG),CUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IV,Y,I2,N//
      IF (ISAA.EQ.1) GC TC 30
      WRITE (5,11C)
      DO 20 I=1,NC
      DO 10 J=1,NC
      WRITE (5,10C) I,J
      CALL RCFEAL (ANSR)
      AY(I,J)=ANSR
      CONTINUE
      CONTINUE
      CALL FRICMS ('CLRSCRN ')
      WRITE (5,12C)
      CALL MATPR (AY,NG,NG)
      WRITE (5,13C)
      CALL RCCHAR (IANS)
      IF ((IANS.NE.IV).AND.(IANS.NE.IZ)) GO TO 50
      GO TO 6C
      WRITE (5,14C)
      GO TO 40
      CONTINUE
      IF (IANS.EQ.IZ) GO TO 90
      WRITE (5,15C)
      CALL RCINT (IANS)
      K=IANS
      WRITE (5,16C)
      CALL RCINT (IANS)
      L=IANS
      WRITE (5,100) K,L
      CALL RCREAL (ANSR)
      CUM=ANSR
      DO 80 I=1,NC
      DO 70 J=1,NC
      IF ((I.EC.K).AND.(J.EQ.L)) AY(I,J)=CUM
      CONTINUE
      GO TO 30
      CONTINUE
      CALL FRICMS ('CLRSCRN ')
      RETURN
      FORMAT (5X,14H THE ELEMENT A(I,12,1H,12,2H)=)
=====

```

```

110 FORMAT (//,5X,54HENTER THE OUTPUT MEASUREMENT COST MATRIX "A"-MAT
1R1X&.,//,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.)
C=====
C      SUBROUTINE READB (NC,ISAB,B)
C      INPUTS THE "B" MATRIX CONTROL COST WEIGHTING MATRIX&.
C=====
      REAL*8 E(NC,NC),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IV,Y,Z/N:/
      IF (ISAB.EQ.1) GC TC 20
      WRITE (5,90)
      DO 10 I=1,NC
      DO 10 J=1,NC
      WRITE (5,80) I,J
      CALL RCREAL (ANSR)
      B(I,J)=ANSR
      GO TO 50
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,100)
      CALL MATPR1 (B,NC,NC)
      WRITE (5,110)
      CALL RCCHAR (IANS)
      IF ((IANS.NE.IV).AND.(IANS.NE.IZ)) GO TO 40
      GO TO 50
      WRITE (5,120)
      GO TO 30
      CONTINUE
      IF (IANS.EQ.IZ) GO TO 70
      WRITE (5,130)
      CALL RCINT (IANS)
      K=IANS
      WRITE (5,140)
      CALL RCINT (IANS)
      L=IANS
      WRITE (5,80) K,L
      CALL RCREAL (ANSR)
      DUM=ANSR
      DO 60 I=1,NC

```



```

      WRITE (5,16C)
      CALL RCINT (IANS)
      K=IANS
      WRITE (5,17C)
      CALL RCINT (IANS)
      L=IANS
      WRITE (5,11C) K,L
      CALL RCREAL (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
      CONTINUE
      GO TO 3C
      CONTINUE
      CALL FRICMS ('CLRSCRN ')
      RETURN
C-----
110  FORMAT (5X,16H THE ELEMENT GAM(I2,1H,I2,2H)=)
120  FORMAT (/,5X,36H ENTER THE PROCESS NOISE DISTRIBUTION,/,5X,24HMATRI
1X "GAMMA"-MATRIX,/,2X,56HDIMENSION = # STATES NS& X # PROCESS
2NDISE SCRCES NG&)
130  FORMAT (/,10X,37H THE PROCESS NOISE DISTRIBUTION MATRIX,/,10X,19H
1"GAMMA"-MATRIX,/,10X,19H)
140  FORMAT (/,5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
1ENT?/,10X,19H TYPE "YES" OR "NO".)
150  FORMAT (/,1X,51HWARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
160  FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
170  FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
1.)
      ENC
C=====
      SUBROUTINE READQ (NG,Q)
      INTERACTIVELY INPUTS THE "Q" MATRIX NOISE WEIGHTING MATRIX &
C=====
      REAL*8 C(NG,NG),DUM,ANSR
      INTEGER IANS,I2,J,K,L
      DATA IY,IY,/,12/,N?
      WRITE (5,11C)
      DO 20 I=1,NG
      DO 10 J=1,NG
      WRITE (5,10C) I,J
      CALL RCREAL (ANSR)
      C(I,J)=ANSR
      CONTINUE
      CONTINUE
      10
      20
C-----

```

```

30 CALL FRTCMS ('CLRSCRN ')
   WRITE (5,120)
   CALL MATPR1 (Q,NG,NG)
40  WRITE (5,130)
   CALL RCCHAR (IANS)
   IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
50  GO TO 60
   WRITE (5,140)
60  GO TO 40
   CONTINUE
   IF (IANS.EC.IZ) GO TO 90
   WRITE (5,150)
   CALL RCINT (IANS)
   K=IANS
   WRITE (5,160)
   CALL RCINT (IANS)
   L=IANS
   WRITE (5,100) K,L
   CALL RCREAL (ANSR)
   DUM=ANSR
   DO 80 I=1,NG
   DO 70 J=1,NG
70  IF ((I.EQ.K).AND.(J.EQ.L)) Q(I,J)=DUM
80  CONTINUE
90  CONTINUE
   GO TO 30
   CONTINUE
   CALL FRTCMS ('CLRSCRN ')
   RETURN
C-----
100 FORMAT (5X,14H THE ELEMENT Q(,I2,1H,12,2H)=)
110 FORMAT (//,5X,44H ENTER THE PROCESS NOISE PSD WEIGHTING MATRIX,/,5X
1,12H "C" MATRIX,/,/,5X,42H DIMENSION = # PROCESS NOISE SOURCES NG&
2,1X,17X,27H#PROCESS NOISE SOURCES NG&)
120 FORMAT (//,5X,42H THE PROCESS NOISE WEIGHTING MATRIX...Q...//)
130 FORMAT (//,5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
1ENT?//,10X,19H TYPE "YES" OR "NO".)
140 FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
150 FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160 FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
1.)
C=====
C SUBROUTINE READR (NO,RC)
C INTERACTIVELY INPUTS THE "R" MATRIX=
C MEASUREMENT NOISE DISTRIBUTION MATRIX.&
C=====
REAL*8 RC(NO,NO),DUM,ANSR

```

```

10  INTEGER IANS,I,J,K,L
11  DATA IV,Y,Z,N:
12  WRITE (5,50)
13  DO 10 I=1,NO
14  DO 10 J=1,NC
15  WRITE (5,60) I,J
16  CALL RCLREAL (ANSR)
17  RC(I,J)=ANSR
18  -----
19  CALL FRICMS ('CLRSCRN ')
20  WRITE (5,100)
21  CALL MATPR (RC,NO,NO)
22  WRITE (5,110)
23  CALL RCLREAL (IANS)
24  IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 40
25  GO TO 50
26  WRITE (5,120)
27  GO TO 30
28  CONTINUE
29  IF (IANS.EQ.IZ) GO TO 70
30  WRITE (5,130)
31  CALL RCLINT (IANS)
32  K=IANS
33  WRITE (5,140)
34  CALL RCLINT (IANS)
35  L=IANS
36  WRITE (5,80) K,L
37  CALL RCLREAL (ANSR)
38  DUP=ANSR
39  DO 60 I=1,NO
40  DO 60 J=1,NC
41  IF ((I.EQ.K).AND.(J.EQ.L)) RC(I,J)=DUP
42  GO TO 20
43  CONTINUE
44  CALL FRICMS ('CLRSCRN ')
45  RETURN
46  -----
47  FORMAT (5X,14H THE ELEMENT R(I2,1H,I2,2H)=)
48  FORMAT (7,5X,60H ENTER THE MEASUREMENT NOISE DISTRIBUTION MATRIX "
49  1R" MATRIX NO.//,5X,53H DIMENSION = # CESERVATIONS NO& X # OBSERVATIO
50  2NS NO&)
51  FORMAT (//,15X,50H THE MEASUREMENT NOISE DISTRIBUTION MATRIX.....R.
52  1...//)
53  FORMAT (//5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
54  1ENT?//,10X,19H TYPE "YES" OR "NO".)
55  FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
56  FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
57  FORMAT (5X,52H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED

```

AD-A152 221

DEVELOPMENT OF GRAPHICAL TIME RESPONSE USING THE
OPTSYSX PROGRAM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY
CA H A DIEL SEP 84

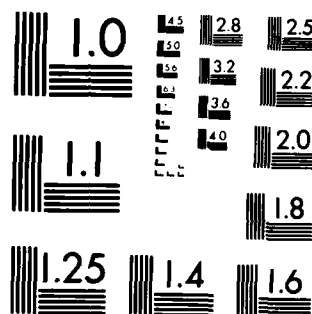
3/3

UNCLASSIFIED

F/G 9/2

NL

										END			



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

1) ENC
C=====
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,IJ,K,L
DATA IY,YO,IJ,N:7
WRITE (5,11C)
DO 20 I=1,NS
DO 10 J=1,NC
WRITE (5,10C) I,J
CALL RCFEAL (ANSR)
FBGE(I,J)=ANSR
CONTINUE
CONTINUE
10
20
C-----
30 CALL FRTCMS ('CLRSCRN ')
WRITE (5,12C)
CALL MATPRT (FBGE,NS,NC)
40 WRITE (5,13C)
CALL RLCFAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
50 GO TO 60
WRITE (5,14C)
60 GO TO 40
CONTINUE
IF (IANS.EQ.IZ) GO TO 90
WRITE (5,15C)
CALL RLCINT (IANS)
K=IANS
WRITE (5,16C)
CALL RLCINT (IANS)
L=IANS
WRITE (5,100) K,L
CALL RCFEAL (ANSR)
DUM=ANSR
DO 80 I=1,NS
DO 70 J=1,NC
IF ((I.EC.K).AND.(J.EQ.L)) FBGE(I,J)=DUM
70 CONTINUE
80 CONTINUE
90 GO TO 30
CONTINUE
CALL FRTCMS ('CLRSCRN ')
RETURN
C-----

```

```

100  FORMAT (5X,14H THE ELEMENT K(I,12,1H,I2,2H)=)
110  FORMAT (//,5X,54H ENTER THE FEEDBACK GAIN ESTIMATOR MATRIX "K"-MATR
110  1IX6,/,10X,48H DIMENSION = # STATES NS6 X # OBSERVATIONS NO6,.)
120  FORMAT (//,15X,47H THE FEEDBACK GAIN ESTIMATOR MATRIX "K"-MATRIX6,
120  1//)
130  FORMAT (//,5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELE
130  1MENT? //,10X,19H TYPE "YES" OR "NO".)
140  FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
150  FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160  FORMAT (5X,52H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
160  1)
1)
C=====
C SUBROUTINE READW (NG,WR)
C INTERACTIVELY INPUTS THE "WO" MATRIX STEADY DISTURBANCE VECTOR =
C MATRIX6 ELEMENT BY ELEMENT.
C=====
REAL*8 WR(NG),DUM,ANSR
INTEGER IANS,I,K
DATA IY,YY,IZ/'N' /
WRITE (5,10C)
DO 10 I=1,NG
WRITE (5,80) I
CALL RDREAL (ANSR)
WR(I)=ANSR
CONTINUE
10
C-----
20  CALL FRIGMS ('CLASCRN ')
WRITE (5,11C)
WRITE (5,50) (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 (5,13C)
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.EC.K) WR(I)=DUM
CONTINUE
60  GO TO 20

```

```

70 CONTINUE
   CALL FRTCMS ('CLRSCRN ')
   RETURN
C-----
80  FORMAT (5X,15H THE ELEMENT W0(,I2,2H)=)
90  FORMAT (F12.5)
100 FORMAT (//,5X,57H ENTER THE STEADY DISTURBANCE VECTOR MATRIX "W0"-M
110 1A TRIX, (//,10X,44H DIMENSION= # PROCESS NOISE SOURCES NGC X 1)
   FORMAT (//,15X,53H THE STEADY DISTURBANCE VECTOR MATRIX "W0"-MATRI
120 1X, (//,10X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
130 1ENT?, (//,10X,19H TYPE "YES" OR "NO".)
140 FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
   END
C=====
C SUBROUTINE RCREAL -- INTERACTIVELY READS A REAL NUMBER REPLY
C INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS A NULL
C STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
SUBROUTINE RCREAL (ANSR)
  REAL*8 ANSR
  INTEGER COUNT
C-----
  COUNT=0
  CONTINUE
  COUNT=COUNT+1
  IF (COUNT.LT.3) GO TO 20
  WRITE (5,60)
  GO TO 40
20 CONTINUE
  READ (5,*,END=30,ERR=30) ANSR
  RETURN
30 REWIND 5
  WRITE (5,50)
  GO TO 10
40 CONTINUE
  STOP
C-----
50  FORMAT (1X,64H WARNING: NULL STRINGS ARE NOT ALLOWED, ENTER A NUME
   1RICAL VALUE.)
60  FORMAT (//,5X,47H PROGRAM TERMINATION - TWO NULL STRINGS ENTERED )
   END
C=====
C SUBROUTINE RCINT -- INTERACTIVELY READS AN INTEGER REPLY
C INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS AN IMPROPER
C DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====

```

```

C-----
SUBROUTINE R0INT ( IANS)
INTEGER CCOUNT, IANS
C-----
10  COUNT=0
    CONTINUE
    COUNT=CCOUNT+1
    IF (CCOUNT.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 RDCPAR -- INTERACTIVELY READS A CHARACTER STRING REPLY =
C  ('YES' OR 'NO') INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY =
C  ENTERS A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY =
C=====
SUBROUTINE RDCPAR ( IANS)
INTEGER CCOUNT, IANS
DATA IV,'Y',I Z/'N'/'
C-----
10  COUNT=0
    CONTINUE
    COUNT=CCOUNT+1
    IF (CCOUNT.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

```



```

150 FORMAT (6F12.5,/,6F12.5,/,3F12.5,/,/)
160 FORMAT (6F12.5,/,6F12.5,/,4F12.5,/,/)
C=====
C SLBROUTINE RDMATF -- READS THE FLAGS AND MATRIX SIZES FROM
C THE DATA FILE CN FILEDEF 9. ASKS IF YOU WANT TO USE THE MATRICES.
C=====
SUBROUTINE RDMATF (NS,NC,NOB,NG,ISAF,ISAG,ISAH,IGAM,ISAA,ISAB,IRDM
1AT)
INTEGER NS,NC,NOB,NG,ISAF,ISAG,ISAH,IGAM,IRDMAT,INO,IANS,K
REWIND 5
READ (9,240,END=30,ERR=30) K,IANS
IF (IANS.EQ.1) GC TC 10
GO TO 30
READ (5,250) NS,NC,NOB,NG
WRITE (5,255)
CALL FRICMS ('CLRSCRN ')
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
RETURN
C-----ISAF-----
30 CALL FRICMS ('CLRSCRN ')
40 WRITE (5,270)
50 CALL RCCHAR (IANS)
IF ((IANS.EQ.1YES).OR.(IANS.EQ.INO)) GC TO 70
60 WRITE (5,330)
70 GO TO 50
CONTINUE
IF (IANS.EQ.IYES) ISAF=1
IF (IANS.EQ.INO) ISAF=0
C-----ISAH-----
80 IF (INO.EQ.0) GO TO 110
90 CALL FRICMS ('CLRSCRN ')
WRITE (5,280)
CALL RCCHAR (IANS)
IF ((IANS.EQ.1YES).OR.(IANS.EQ.INO)) GC TO 100
WRITE (5,330)

```

```

100 GO TO EC
CONTINUE
IF (IANS.EC.IYES) ISAH=1
IF (IANS.EQ.INO) ISAH=0
110 CONTINUE-----ISAG-----
IF (NG.EC.0) GO TO 150
CALL FRICMS ('CLRSCRN ')
WRITE (5,29C)
CALL RDCPAR (IANS)
IF ((IANS.EC.IYES).CR.(IANS.EQ.INO)) GO TO 140
WRITE (5,33C)
GO TO 120
140 CONTINUE
IF (IANS.EC.IYES) ISAG=1
IF (IANS.EQ.INO) ISAG=0
150 CONTINUE-----IGAM-----
IF (NG.EC.0) GO TO 190
CALL FRICMS ('CLRSCRN ')
WRITE (5,30C)
CALL RDCPAR (IANS)
IF ((IANS.EC.IYES).OR.(IANS.EQ.INO)) GO TO 180
WRITE (5,33C)
GO TO 160
180 CONTINUE
IF (IANS.EC.IYES) IGAM=1
IF (IANS.EQ.INO) IGAM=0
190 CONTINUE-----ISAA-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,31C)
CALL RDCPAR (IANS)
IF ((IANS.EC.IYES).OR.(IANS.EQ.INO)) GO TO 210
WRITE (5,33C)
GO TO 200
210 CONTINUE
IF (IANS.EC.IYES) ISAA=1
IF (IANS.EQ.INO) ISAA=0
C-----ISAB-----
CALL FRICMS ('CLRSCRN ')
WRITE (5,32C)
CALL RDCPAR (IANS)
IF ((IANS.EC.IYES).OR.(IANS.EQ.INO)) GO TO 230
WRITE (5,33C)
GO TO 220
230 CONTINUE
IF (IANS.EC.IYES) ISAB=1

```

```

IF (IANS.EQ.INO) I SAB=0
RETURN
C-----
240  FORMAT (11,3X,11)
250  FORMAT (415)
255  FORMAT (//////)
260  FORMAT (//////)
1//,12X,42H THE "F", "G", "H", "GAM", "A" AND "B" MATRICES
2FOLLOWING OPTIONS ARE AVAILABLE: //,15X,38H1. USE ALL OF THE SAME MA
3TRICES AGAIN. //,15X,2. USE SELECTED MATRICES AGAIN. //,15X,3.
415X,3. INPUT ALL NEW MATRICES. //,10X,17H ENTER 1, 2, OR 3.
5//,10X,3. NOTE: EACH SAVED MATRIX WILL BE REDISPLAYED AT
6//,10X,34H THE PROPER INPUT SEQUENCE INTERVAL
7//,10X,40H AND YOU WILL HAVE THE OPTION OF CHANGING //,10X,
827H INDIVIDUAL MATRIX ELEMENTS. //,10X,48HDC YOU WISH TO SAVE THE
9FORMAT (////,5X,48HDC YOU WISH TO SAVE THE "F" MATRIX FROM THE LAST
1//,5X, WILL BE REDISPLAYED AT //,5X,34H THE PROPER INPUT SEQUENCE IN
2AT RIX //,5X,40H AND YOU WILL HAVE THE OPTION OF CHANGING //,5X,27HIND
3ERVAL //,5X,40HDC YOU WISH TO SAVE THE "G" MATRIX FROM THE LAST
4INDIVIDUAL MATRIX ELEMENTS. //,15X,19H TYPE "YES" OR "NO".
5FORMAT (////,5X,48HDC YOU WISH TO SAVE THE "H" MATRIX FROM THE LAST
1//,5X, WILL BE REDISPLAYED AT //,5X,34H THE PROPER INPUT SEQUENCE IN
2AT RIX //,5X,40H AND YOU WILL HAVE THE OPTION OF CHANGING //,5X,27HIND
3ERVAL //,5X,40HDC YOU WISH TO SAVE THE "GAMMA" MATRIX FROM THE LAST
4INDIVIDUAL MATRIX ELEMENTS. //,15X,19H TYPE "YES" OR "NO".
5FORMAT (////,5X,48HDC YOU WISH TO SAVE THE "A" MATRIX FROM THE LAST
1LAST //,5X, WILL BE REDISPLAYED AT //,5X,34H THE PROPER INPUT SEQUENCE IN
2THE MATRIX //,5X,40H AND YOU WILL HAVE THE OPTION OF CHANGING //,5X,27
3INTERVAL //,5X,40HDC YOU WISH TO SAVE THE "A" MATRIX FROM THE LAST
4INDIVIDUAL MATRIX ELEMENTS. //,15X,19H TYPE "YES" OR "NO".
5FORMAT (////,5X,48HDC YOU WISH TO SAVE THE "B" MATRIX FROM THE LAST
1//,5X, WILL BE REDISPLAYED AT //,5X,34H THE PROPER INPUT SEQUENCE IN
2AT RIX //,5X,40H AND YOU WILL HAVE THE OPTION OF CHANGING //,5X,27HIND
3ERVAL //,5X,40HDC YOU WISH TO SAVE THE "B" MATRIX FROM THE LAST
4INDIVIDUAL MATRIX ELEMENTS. //,15X,19H TYPE "YES" OR "NO".
5FORMAT (////,5X,48HDC YOU WISH TO SAVE THE "A" MATRIX FROM THE LAST
1//,5X, WILL BE REDISPLAYED AT //,5X,34H THE PROPER INPUT SEQUENCE IN
2AT RIX //,5X,40H AND YOU WILL HAVE THE OPTION OF CHANGING //,5X,27HIND
3ERVAL //,5X,40HDC YOU WISH TO SAVE THE "B" MATRIX FROM THE LAST
4INDIVIDUAL MATRIX ELEMENTS. //,15X,19H TYPE "YES" OR "NO".
5FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
ENC
C=====

```

```

C SUBROUTINE RDMAT -- READS THE F, G, H, GAM, A AND B MATRICES FROM
C MATRICES FROM THE DATA FILE OPTMAT ON FILEDEF 5.
C=====
SUBROUTINE RDMAT(BA,G,HQ,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG,IKDMAT)
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.EC.O) RETURN
REWIND 5
READ (5,20) 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,NOI),I=1,NOI)
READ (9,10) ((GAM(I,J),J=1,NGI),I=1,NSI)
READ (9,10) ((FBGC(I,J),J=1,NCI),I=1,NCI)
READ (9,10) ((FBGE(I,J),J=1,NOI),I=1,NOI)
READ (9,10) ((AY(I,J),J=1,NCI),I=1,NCI)
READ (9,10) ((B(I,J),J=1,NCI),I=1,NCI)
RETURN
10 FORMAT(4(C20.13))
20 FORMAT(4I5)
ENC
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,HQ,GAM,FBGC,FBGE,AY,E,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)
INTEGER NS,NC,NG,I,J,IANS,INC,IYES
DATA IYES/5,7,INO/N/
WRITE (5,90)
CALL FRTCMS('CLRSCRN ')
WRITE (5,100)
CALL RLCFAR(IANS)
IF ((IANS.EC.INO).OR.(IANS.EQ.IYES)) GO TO 20
WRITE (5,110)
GO TO 10
IF (IANS.EC.INO) RETURN
REWIND 5
I=0
IANS = 1
WRITE (9,140) I, IANS
WRITE (5,120) NS,NC,NO,NG
WRITE (5,130) ((BA(I,J),J=1,NS),I=1,NS)

```

```

WRITE(S,130) ((G(I,J),J=1,NC),I=1,NS)
WRITE(S,130) ((H(I,J),J=1,NS),I=1,NC)
WRITE(S,130) ((GAM(I,J),J=1,NG),I=1,NS)
WRITE(S,130) ((FBGC(I,J),J=1,NS),I=1,NC)
WRITE(S,130) ((FBGE(I,J),J=1,NO),I=1,NS)
WRITE(S,130) ((AY(I,J),J=1,NO),I=1,NC)
WRITE(S,130) ((B(I,J),J=1,NC),I=1,NC)
STCP
-----
90  FORMAT(//////////)
100  FORMAT(///,10X,DO YOU WISH EVALUATING? ,/,24X,(Y OR N),/,
112X,OF THE SYSTEM YOU ARE LOGGED ON AT A DUAL SCREEN? ,/,
2///,5X,NOTE: YOU MUST BE LOGGED ON AT A DUAL SCREEN? ,/,
310X,(TEK 618) TERMINAL TO UTILIZE THIS MODE. ,///,10X,
458H THE F (SYSTEM), G (CONTROL), H (OBSERVABLES), GAM (NCISE), ,/,
513X,53FA (OUTPUT COST) AND B (CONTROL CTS) MATRICES WILL BE ,/,
616X,SAVED FOR REENTRY TO THE MAIN CPTSYS PROGRAM. ?)
110  FORMAT(10X,29HYOL MUST ANSWER (Y)ES OR (N)O )
120  FORMAT(4I5)
130  FORMAT(4C20.13)
140  FORMAT(11,3X,11)
END

```

```

*****
CPTCALC
CALCULATES THE GRAPHICAL TIME RESPONSE OF A
HIGH ORDER SYSTEM UNDER THE CNTRL OF THE
OPTSYS EXEC
BY
H. A. DIEL
15 JUL 1984
*****
=====
MAIN PROGRAM - PERFORMS INTEGRATION OVER THE DESIRED TIME SPAN
=====
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(4000),U(10),IWK(100),IYPE(10),FBG
2E(32,32),HK(32,32),DUMMY(32,32),BG(32,32)
DATA IVES/.V./,INO/.N./
COMMON F,G,L,NC
COMMON /A/ IYPE,DREND,DRBEG,UMAX
EXTERNAL FCN,FCNJ
=====
SUPPRESS INDIVIDUAL UNDERFLOW, OVERFLOW, DIVIDE CHECK, AND DECIMAL =
CONVERT ERROR MESSAGES: PROVIDE SUMMARY OF ERRORS ONLY.
=====
CALL ERRSET (207,256,-1,1,1,209)
CALL ERRSET (215,256,-1,1,1)
=====
READ IN DATA FILE
=====
10 REMIND 5
READ (5,600) I,J
CALL FRICMS (I,CLSCRN *)
WRITE (5,280)
READ (5,590) NS,NC,NO,NG
CALL RENAT (F,G,HO,FBGC,FBGE,NS,NC,NO,NG)
=====
DISPLAY INPUT MATRICES
=====
CALL NEHSCR
WRITE (5,290)
=====

```



```

70      G(I+NS,K)=G(I,K)
      CONTINUE
      DO 80 J=1,NS
      DUMMY(I,J)=F(I,J)
      DUMMY(I+NS,J)=HK(I,J)
      DUMMY(I,J+NS)=0.0000
      DUMMY(I+NS,J+NS)=F(I,J)-HK(I,J)
      CONTINUE
      NS2=2*NS
      DO 90 I=1,NS2
      DO 90 J=1,NS2
      F(I,J)=DUMMY(I,J)
      CONTINUE
      CALL NEWSCR
      WRITE(I,380)
      CALL MATPRT (F,NS2,NS2)
      CALL NEWSCR
      WRITE(I,390)
      CALL MATPRT (G,NS2,NC)
      CALL NEWSCR
      GO TO 140

C----- FILTER & REGULATOR CLOSED LOOP SYSTEM -----
C-----
100     CALL FRICMS ('CLRSCRN ')
      CALL MAMULT (FBGE,NS,NC,H0,NS,HK)
      CALL MAMULT (G,NS,NC,FBGC,NS,BG)
      CALL NEWSCR
      WRITE(I,360)
      CALL MATPRT (BG,NS,NS)
      CALL NEWSCR
      WRITE(I,370)
      CALL MATPRT (HK,NS,NS)
      DO 120 I=1,NS
      DO 110 K=1,NC
      G(I+NS,K)=G(I,K)
      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)
      CONTINUE
      NS2=2*NS
      DO 130 I=1,NS2
      DO 130 J=1,NS2
      F(I,J)=DUMMY(I,J)
      CONTINUE
110
120
130

```

```

CALL NEWSCR
WRITE (5,38C)
CALL MATPRT (F,NS2,NS2)
CALL NEWSCR
WRITE (5,39C)
CALL MATPRT (G,NS2,NC)
CALL NEWSCR
TESTIME=

```

```

C-----INPLT INTEGRATION START AND STOP TIMES
C-----

```

```

140 CALL FRICMS ('CLRSCRN ')
    WRITE (5,40C)
    CALL RCFEAL (T)
    CALL FRICMS ('CLRSCRN ')
    WRITE (5,41C)
    CALL RCFEAL (TSTEP)

```

```

C-----INPLT NUMBER OF POINTS TO CALCULATE
C-----

```

```

150 CALL FRICMS ('CLRSCRN ')
    WRITE (5,42C)
    CALL RCINT (NPTS)
    IF (NPTS.GT.500) GO TO 150
    NPTS=NPTS+1

```

```

C-----SELECT DRIVING FUNCTION & START & STOP TIMES
C-----

```

```

WRITE (5,43C)
CALL RECHAR (IANS)
DO 160 I=1,NC
  U(I)=0.0000
  ITYPE(I)=1
  DRBEG(I)=0.00
  DREND(I)=0.00
  UMAX(I)=0.00
  CONTINUE
  IF (IANS.EQ.IYES) GO TO 170
  GO TO 210
DO 200 I=1,NC
  CALL FRICMS ('CLRSCRN ')
  WRITE (5,44C) I
  CALL RCINT (IANS)
  IF ((IANS-GE.1).AND.(IANS.LE.2)) GO TO 190
  WRITE (5,45C)
  GO TO 180
  ITYPE(I)=IANS
  CALL FRICMS ('CLRSCRN ')

```

```

160 CONTINUE
    IF (IANS.EQ.IYES) GO TO 170
    GO TO 210

```

```

170 DO 200 I=1,NC
    CALL FRICMS ('CLRSCRN ')
    WRITE (5,44C) I
    CALL RCINT (IANS)
    IF ((IANS-GE.1).AND.(IANS.LE.2)) GO TO 190
    WRITE (5,45C)
    GO TO 180
    ITYPE(I)=IANS
    CALL FRICMS ('CLRSCRN ')

```

```

180 CONTINUE
    IF (IANS.EQ.IYES) GO TO 170
    GO TO 210
DO 200 I=1,NC
  CALL FRICMS ('CLRSCRN ')
  WRITE (5,44C) I
  CALL RCINT (IANS)
  IF ((IANS-GE.1).AND.(IANS.LE.2)) GO TO 190
  WRITE (5,45C)
  GO TO 180
  ITYPE(I)=IANS
  CALL FRICMS ('CLRSCRN ')

```



```

C MITER = 2-FINITE DIFFERENCE, = 3-DIRECTIONAL DERIV, = 0-FUNCTIONAL DER
C METH = 1-ACAMS METHOD, = 2-STIFF SYSTEM - GEAR METHOD
C METH=2
INDEX=1
TOL=1.0E-10
NPTS=NPTS-1
DEL TAT=(TSTOP - T)/DFLOAT(NPTS)
H=1.0D-10
WRITE (5,570) (T,(U(K),K=1,NC),(X(J),J=1,NS2))
CALL FRICMS (,CLRSCRN ,)
WRITE (5,54C)
WRITE (5,58C) T,U(1),X(1),X(2),X(3)
ICCOUNT=NPTS/50
DO 270 I=1,NPTS
TEND=T+DEL TAT
CALL DGEAR (NS2,FCN,FCNJ,T,H,X,TEND,TOL,METH,MITER,
1 INDEX,IWK,IWK,IER)
IF (MOL(I,ICOUNT)).EQ.0) WRITE (5,58C) T,U(1),X(1),X(2),X(3)
WRITE (5,570) (T,(U(K),K=1,NC),(X(J),J=1,NS2))
IF (IER.EQ.C) GO TC 270
IF (IER.EC.66) WRITE (5,550)
IF (IER.NE.132) AND.(IER.NE.133)) GO TC 270
TOL=TOL*.0001
WRITE (5,560)
CONTINUE
270 STCP
C=====
280 FORMAT (//10X,44H DURING THIS SECTION OF THE PROGRAM YOU WILL: ,//
110X,44H- SELECT THE TYPE OF SYSTEM RESPONSE TO PLOT,/,12X,45H( OPEN
2 LCOP,1 CLSSED LOOP, GR FILTER/REGULATORS),/,10X,55H- PROVIDE START
3 AND STOP TIMES 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,28H CLEAR THE SCREEN TO CONTINUE)
FORMAT (//,15X,14H THE F MATRIX,///)
FORMAT (//,15X,14H THE G MATRIX,///)
FORMAT (//,15X,14H THE H MATRIX,///)
FORMAT (//,10X,51H THE FOLLOWING PLOTTING OPTIONS ARE AVAILABLE IF
1THE,/,1CX,45H REQUIRED MATRICES WERE CALCULATED IN OPTS V SX: ,//,15X,
227H1. CPEN LOOP TIME RESPONSE,/,20X,21HXDCT = F&*X + G&*UC,/,15X,
3X,29H2. CLSSED LOOP TIME RESPONSE,/,20X,21HXDCT = F&*X + G&*UC,/,15X,
4UC,15X,37HXDCT = F&*X + G&*UC,/,15X,61H4. OPTIMIZED FIL
5 RESPONSE,/,20X,37HXDCT = F&*X + G&*UC,/,15X,61H4. OPTIMIZED FIL
6HX FCT = F&*X + G&*UC,/,15X,61H4. OPTIMIZED FIL
7TER + REGULATOR CLOSED LOOP SYSTEM K&*Z - H&*X&E,/,15X,61H4. OPTIMIZED FIL
8C&*X + G&*UC,/,15X,61H4. OPTIMIZED FIL

```

```

340 95U + K8* Z - H*XH6, THE K MATRIX, //, 10X, 20+SELECT 1, 2, 3 OR 4.)
350 FORMAT (//, 15X, 14H THE AUGMENTED F MATRIX (F+G*C), //)
360 FORMAT (//, 15X, 16H THE (G*C) MATRIX, //)
370 FORMAT (//, 15X, 16H THE (K*H) MATRIX, //)
380 FORMAT (//, 15X, 16H THE COMBINED SYSTEM F MATRIX (2*NS X 2*NS), //)
390 FORMAT (//, 15X, 16H THE AUGMENTED G MATRIX (2*NS X NC), //)
400 FORMAT (//, 10X, 33HAT WHAT TIME DO YOU WANT TO START, //, 10X, 31HTHE T
1 TIME RESFCNSE CALCULATIONS?, //, 10X, 43HINPUT START TIME IN SECONDS.
2 (NORMALY 0.0))
410 FORMAT (//, 10X, 32HAT WHAT TIME DO YOU WANT TO STOP, //, 10X, 31HTHE TI
1 ME RESPCNSE CALCULATIONS?, //, 10X, 29H INPUT STOP TIME IN SECONDS.)
420 FORMAT (//, 10X, 47HTHIS PROGRAM DIVIDES THE TIME INTERVAL YOU HAVE,
1/, 10X, 45HJUST SPECIFIED INTO UP TO 500 SMALL INTERVALS FOR, //, 10X, 4
28H THE INTEGRATION AND PLOTTING ROUTINES. IN ORDER, //, 10X, 50HTO SAV
3E COMPUTER TIME, THE CURVE FIDELITY 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 (CCNTR
1CL, 8H INPUT)?, //, 25X, 13HYES OR (N)O)
440 FORMAT (//, 10X, 46H2O INPUT, //, 10X, 19H 2. RAMP INPUT, //, 10X, 29H
1, 10X, 19H 1. STEP INPUT, //, 10X, 28H MUST BE BETWEEN 1 AND 2.)
450 ENTER YOUR SELECTION, ANSWER 1 OR 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
1 STOP?, //, 10X, 33H INPUT THE STOP TIME IN SECONDS.)
480 FORMAT (//, 10X, 28HWHAT IS THE MAXIMUM VALUE CF, //, 10X, 23HCRIVING FUN
1CT IEN NUMBER, I2, 2H ?)
490 FORMAT (//, 10X, 49HDOES THE SYSTEM START WITH ALL INITIAL CONDITION
1S, 8H = 0.0 ?), //, 25X, 14HYES OR (N)O?)
500 FORMAT (//, 10X, 36HWHAT IS THE INITIAL CCNDITION FOR X(, 12, 3H) ?)
510 FORMAT (//, 10X, 39HWHAT IS THE INITIAL CCNDITION FOR X(, 12, 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 CCNDITIONS, //, 12X, 12H5. CONTINUE, //, 10X,
532+SELECT A NUMBER BETWEEN 1 AND 5.)
530 FORMAT (//, 36HYOUR ANSWER MUST BE BETWEEN 1 AND 5.)
540 FORMAT (//, 10X, 41HTHE FOLLOWING INFORMATION IS PROVIDED ONLY, //, 10X, 4
16HFCR AN INDICATION OF PROPER PROGRAM OPERATION, //, 5X, 56HALL CNT
2RULS, STATES AND STATE OF ESTIMATES CAN BE PLOTTED, //, 5X, 4HTIME, I1X,
34HU(1), 10X, 4HX(1), 10X, 4HX(2), 10X, 4HX(3), //)
550 FORMAT (//, 80HPIER = 66 INDICATES THAT STEP SIZE WAS SUCCESSFULLY RED
1UCED TC INCREASE ACCURACY, //, 18HNO ACTION REQUIRED)
560 FORMAT (//, 51HPIER = 132 OR 133 INDICATES THAT CONVERGENCE WAS NOT, 35

```

```

1H ACHIEVED AFTER REDUCING STEP SIZE.//,21HCCNVERGENCE TOLERANCE,3
26H IS BEING REDUCED BY A FACTOR OF TEN,/,22HFOR ANOTHER ATTEMPT AT
3,13H CCNVERGENCE.)
570 FORMAT (5E14.7)
580 FORMAT (1X,F8.2,5X,4E14.7)
590 FORMAT (515)
600 FORMAT (11,3X,11)
    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,C-Z)
DIMENSION X(32),XDOT(32),F(32,32),G(32,10),L(10)
COMMON F,G,U,NC

```

```

C=====
C CALL DRIVER (T,U,NC)
C SPECIAL DRIVER FOR XV-29A MATRICES
C=====

```

```

IF (NS.LT.77) GO TO 15
IF (T.LT.1.0) U(1)=0.000
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)=0.000
DO 10 I=1,NC
U(I)=U(1)
CONTINUE

```

```

10 CONTINUE
15 CONTINUE
C=====

```

```

DO 40 J=1,NS
XDOT(J)=0.000
DO 20 I=1,NC
XDOT(J)=XDOT(J)+G(J,I)*U(I)
CONTINUE
DO 30 K=1,NS
XDOT(J)=XDOT(J)+F(J,K)*X(K)
CONTINUE
RETURN
END

```

```

20 CONTINUE
30 CONTINUE
40 CONTINUE

```

```

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,C-Z)
DIMENSION X(32),PD(32,32)

```

```

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,C-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 (IT.LT.DRBEG(I)) U(I)=0.0
IF (T.LT.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 RCMAT -- READS THE F, G, H, C AND K MATRICES
C FROM THE DATA FILE CPTMAT DATA ON FILEDEF 9.
C=====
SUBROUTINE RCMAT (BA,G,HO,FBGC,FBGE,NS,NC,NG)
IMPLICIT REAL*8(A-H,C-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))
ENC
C=====
C SUBROUTINE MATPRT -- DISPLAYS A TWO-DIMENSIONAL ARRAY (16 COLUMNS MAX)
C IN VARIABLE SCREEN FORMAT FOR USER EASE IN ROW IDENTIFICATION.
C=====
SUBROUTINE MATPRT (PRIT,NRCW,NCOL)
IMPLICIT REAL*8 (A-H,C-Z)
DIMENSION PRIT(32,32)
C-----

```

```

IF (NCCL.EQ.0) NCOL=1
IF (NCCL.EQ.1) WRITE (5,10) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.2) WRITE (5,20) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.3) WRITE (5,30) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.4) WRITE (5,40) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.5) WRITE (5,50) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.6) WRITE (5,60) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.7) WRITE (5,70) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.8) WRITE (5,80) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.9) WRITE (5,90) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.10) WRITE (5,100) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.11) WRITE (5,110) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.12) WRITE (5,120) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.13) WRITE (5,130) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.14) WRITE (5,140) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.15) WRITE (5,150) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCCL.EQ.16) WRITE (5,160) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
RETURN

```

```

100 FORMAT (F12.5)
200 FORMAT (2F12.5)
300 FORMAT (3F12.5)
400 FORMAT (4F12.5)
500 FORMAT (5F12.5)
600 FORMAT (6F12.5)
700 FORMAT (6F12.5,/.F12.5,///)
800 FORMAT (6F12.5,/.2F12.5,///)
900 FORMAT (6F12.5,/.3F12.5,///)
1000 FORMAT (6F12.5,/.4F12.5,///)
1100 FORMAT (6F12.5,/.5F12.5,///)
1200 FORMAT (6F12.5,/.6F12.5,///)
1300 FORMAT (6F12.5,/.6F12.5,/.F12.5,///)
1400 FORMAT (6F12.5,/.6F12.5,/.2F12.5,///)
1500 FORMAT (6F12.5,/.6F12.5,/.3F12.5,///)
1600 FORMAT (6F12.5,/.6F12.5,/.4F12.5,///)
END

```

```

C=====
C SUBROUTINE MAMULT (A,NAROWS,NACOLS,B,NBCOLS,C)
C MULTIPLIES TWO MATRICES C=C*A
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=0.0
DO 10 K=1,NACOLS
SUM=SUM+A(I,K)*B(K,J)
CONTINUE

```

```

20      C(I,J)=SUM
      CONTINUE
      RETURN
      END
=====
C      SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY
C      INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS AN IMPROPER
C      DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
      SUBROUTINE RDINT ( IANS)
      INTEGER COUNT
C-----
      COUNT=C
      CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 50
      CONTINUE
      READ (5,*,END=40,ERR=40) IANS
      IF (IANS) 4C,40,30
      CONTINUE
      RETURN
      REWIND 5
      WRITE (5,70)
      GO TO 10
      CONTINUE
      STOP
C-----
60      FORMAT (//,5X,49HPRCGRAM TERMINATION - TWO IMPROPER DATA ENTRIES
70      1)
      FORMAT (1X,56HWARNING: IMPROPER DATA ENTRY ENTER A POSITIVE INTE
      GER.)
      END
=====
C      SUBROUTINE RDCPAR -- INTERACTIVELY READS A CHARACTER STRING REPLY
C      (.YES. OR .NO.) INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY
C      ENTERS A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY
C=====
      SUBROUTINE RDCPAR ( IANS)
      INTEGER CCOUNT
C-----
      COUNT=C
      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"
10CR "NC".)
60 FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED )
70 FORMAT (A1)
   END
C=====
C SUBROUTINE RCREAL -- INTERACTIVELY READS A REAL NUMBER REPLY
C INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS A NULL
C STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
SUBROUTINE ROREAL (ANSR)
   REAL*8 ANSR
   INTEGER CCLNT
C-----
10 COLNT=0
   CONTINUE
   COUNT=CCLNT+1
   IF (CCLNT.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,60HWARNING: NULL STRINGS ARE NOT ALLOWED, ENTER A NUME
10RICAL VALUE.)
60 FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED )
   END
C=====
C SUBROUTINE RDCPST -- INTERACTIVELY READS A CHARACTER STRING REPLY
C UP TO 40 CHARACTERS LONG AND FORMATS THE CHARACTER STRING FOR USE
C BY A GISSPLA PRINT ROUTINE.
C=====

```

```

SUBROUTINE RDCHST (CHST)
INTEGER CHST(11),I
DATA IEL/1,ICCL/'$' '/'
C-----
CALL GETCHS (CHST)
CHST(11) = IBL
DO 10 I = 1,11
  IF (CHST(I).NE.IBL) GC TC 10
  CHST(I) = IDOL
  GO TC 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=C
CONTINUE
COUNT=COUNT+1
IF (COUNT.LT.3) GO TO 20
WRITE (5,60)
GO TO 40
CONTINUE
REWIND 5
READ (5,70,END=30,ERR=30) (CHST(I),I = 1,10)
RETURN
REWIND 5
WRITE (5,50)
GO TO 10
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,'4TH PROGRAM TERMINATION - TWO NULL STRINGS ENTERED ')
70 FORMAT (10A4)
END
C=====
C SLBROUTINE NEWSCR -- CLEARS THE SCREEN WITHOUT ERASING THE =
C PREVIOUS SCREEN'S INFORMATION. =
C=====
SUBROUTINE NEWSCR

```

WRITE (5,10)
CALL FRTCMS (' CLRS CRN ')
RETURN

C-----
10 FORMAT (//////////)

END


```

CALL RCIAT (IANS)
IF (IANS.EQ.1) GO TC 50
IF (IANS.EQ.2) GO TC 210
GO TO 20
IF (ICLCGR.EQ.0) GC TC 50
=====
C== READ IN DATA AGAIN IF OLD GRAPH DATA WAS USED
C==
IOLDGR=C
REWIND E
READ (8,139C) NS,NC,NPTS,TEST
N = NS
IF (TEST.EQ.3) N=2*NS
READ (8,138C) ((FBGC(I,J),J=1,NS),I=1,NC)
DO 40 I=1,NPTS
READ (8,138C) TIME(I),(U(I,J),J=1,NC),(DATA(I,K),K=1,N)
CONTINUE
NPTSDA=NPTS
C=====
C== SELECT NUMBER OF CURVES TO PLOT CN GRAPH
C==
50 WRITE (5,56C)
CALL RCIAT (NCURVS)
IF ((NCURVS.GE.1).AND.(NCURVS.LE.4)) GC TO 60
WRITE (5,57C) NCURVS
GO TO 50
CALL FRICMS ('CLRSCRN ')
XPAGE=8.5
YPAGE=6.0
DELTAX=C.0
DELTAY=C.0
SCALEH=1.0
N=1
CALL SELCRV (N,C1,C1MIN,C1MAX,TITLE1,DATA,U,FBGC,NS,NC,NPTS,TEST)
IF (NCURVS.EQ.1) GC TO 70
N=2
CALL SELCRV (N,C2,C2MIN,C2MAX,TITLE2,DATA,U,FBGC,NS,NC,NPTS,TEST)
IF (NCURVS.EQ.2) GC TO 70
N=3
CALL SELCRV (N,C3,C3MIN,C3MAX,TITLE3,DATA,U,FBGC,NS,NC,NPTS,TEST)
IF (NCURVS.EQ.3) GC TO 70
N=4
CALL SELCRV (N,C4,C4MIN,C4MAX,TITLE4,DATA,U,FBGC,NS,NC,NPTS,TEST)
CALL FRICMS ('CLRSCRN ')
C=====
C== SELECT NUMBER OF HEADINGS FOR GRAPH
C==
70 HEAD2(1)=IHEAD
C=====

```

```

80      HEAD3(1)=IHEAD
      WRITE (5,980)
      CALL RCREAL (DOUBLP)
      NHEAD=ICINT(DOUBLP)
      IF ((NHEAD.EQ.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 PLCT CURVES
      C=====
100     CALL FRTCMS ('CLRSCRN ')
110     CALL TEK618
      CALL HWFCT ('AUTO')
      CALL HWSCAL ('SCREEN')
      GO TO 120
115     IF (ICMPRS.EQ.1) GO TO 120
      CALL CCMPRS
      ICMPRS=1
120     CALL PAGE (XPAGE,YPAGE)
      CALL NCERDR
      CALL TRIFLX
      XPOS=.2+.3*(XPAGE+YPAGE)/19.5
      YORIGN=.1+.35*(XPAGE+YPAGE)/19.5
      CALL PHYSOR (XORIGN,YORIGN)
      XAXIS=XPAGE-XORIGN-.3
      YAXIS=YPAGE-YORIGN-.3
      IF (NFEAC.EQ.0) YAXIS=YPAGE-YORIGN-.3
      CALL AREA2D (XAXIS,YAXIS)
      CALL FRAME
      HI TE=(XPAGE+YPAGE)*.006*SCALEH
      IF (HI TE.LT.0.01) HI TE=0.01
      CALL HEIGHT (HI TE)
      CALL BASALF ('STAN CARD')
      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,ISTEP,AXIS)
TENC=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 (TTITLE1,LGND1,1)
IF (NCURVS.EQ.1) GC TC 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 (TTITLE2,LGND1,2)
CALL RESET ('DASH')
IF (NCURVS.EQ.2) GC TC 130
CALL YGRAXS (C3MIN,SCALE,C3MAX,YAXIS,TITLE3,100,-2.0*XPOS,0.0)
CALL RLVEC (TSTART,0.0,TEND,0.0,0000)
CALL CFNCLT
CALL LEGLIN
CALL CURVE (TIME,C3,NPTS,MARKRS)
CALL LINES (TTITLE3,LGND1,3)
CALL RESET ('CHND01')
IF (NCURVS.EQ.3) GC TC 130
CALL YGRAXS (C4MIN,SCALE,C4MAX,YAXIS,TITLE4,100,-3.0*XPOS,0.0)
CALL RLVEC (TSTART,0.0,TEND,0.0,0000)
CALL CFNCLSH
CALL LEGLIN
CALL CURVE (TIME,C4,NPTS,MARKRS)
CALL LINES (TTITLE4,LGND1,4)
CALL RESET ('CHNDSF')
=====
C===== PRINT LEGEND
C=====
C=====
130 CALL LINESP (1.8)
X1=XLEGND(LGND1,NCURVS)
Y1=YLEGND(LGND1,NCURVS)
XLED=XAXIS-0.2-X1+DELTA X
YLED=Y+0.2+DELTA Y
CALL LEGND (LGND1,NCURVS,XLED,YLED)
=====
C===== PRINT GRID LEAVING BOX AROUND LEGEND
C=====
C=====
CALL BLFEC (XLED-0.1,YLED-0.1,X1+.2,Y1+.2,2.)
CALL CCT
CALL GRID (1,1)

```

```

140 CALL RESET ('DOT')
    CALL ENLCR (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)
    CONTINUE
140 CALL ENDPL (0)
    IF (ICMPRS.EQ.1) GC TO 880
C=====
C MAIN OPTICNS MENU
C=====
150 CALL FRICMS ('CLRSCRN ')
160 WRITE (5,1040)
    CALL RLINT (IANS)
    IF ((IANS.GE.1).AND.(IANS.LE.5)) GC TO 170
    CALL FRICMS ('CLRSCRN ')
    WRITE (5,1010) IANS
    GO TO 160
170 CALL FRICMS ('CLRSCRN ')
    GO TO (180,210,270,840,850), IANS
C=====
C BEGIN A NEW GRAPH
C=====
180 WRITE (5,1040)
    CALL RLCHAR (IANS)
    IF (IANS.EQ.IYES) GC TO 190
    IF (IANS.EQ.INO) GC TO 200
    WRITE (5,1050)
    GO TO 180
190 CALL FILECV (NPTS,NCURVS,NHEAD,HEAD1,HEAD2,HEAD3,TITLE1,
    1,TITLE2,TITLE3,TITLE4,XPAGE,YPAGE,DELTA X,DELTA Y,SCALEH,
    2,C1MIN,C1MAX,C2MIN,C2MAX,C3MIN,C3MAX,C4MIN,C4MAX,
    3,TIME,C1,C2,C3,C4)
    GO TO 200
200 GO TO 20
C=====
C READ CURVE DATA FROM FILE AND PLOT CURVE
C=====
210 WRITE (5,1040)
    CALL RLCHAR (IANS)
    IF (IANS.EQ.IYES) GC TO 220
    IF (IANS.EQ.INO) GC TO 230
    WRITE (5,1050)
    GO TO 210
220 CALL FILECV (NPTS,NCURVS,NHEAD,HEAD1,HEAD2,HEAD3,TITLE1,
    1,TITLE2,TITLE3,TITLE4,XPAGE,YPAGE,DELTA X,DELTA Y,SCALEH,

```

```

230 2C1MIN,C1MAX,C2MIN,C2MAX,C3MIN,C3MAX,C4MIN,C4MAX,
    3TIME,C1,C2,C3,C4)
    WRITE (5,1060)
    READ (5,1410,END=265,ERR=265) (NAMFIL)
    CALL FRIGMS (FILEDEF,04,DISK,NAMFIL,DATA,
1A1)
    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 (INHEAD.EQ.1) GO TO 240
    READ (4,1400,END=260,ERR=260) (HEAD2(I),I=1,11)
    IF (INHEAD.EQ.2) GO TO 240
    READ (4,1400,END=260,ERR=260) (HEAD3(I),I=1,11)
    READ (4,1400,END=260,ERR=260) (TITLE1(I),I=1,11)
    IF (INCLRV.EQ.1) GC TO 250
    READ (4,1400,END=260,ERR=260) (TITLE2(I),I=1,11)
    IF (INCLRV.EQ.2) GC TO 250
    READ (4,1400,END=260,ERR=260) (TITLE3(I),I=1,11)
    IF (INCLRV.EQ.3) GC TO 250
    READ (4,1400,END=260,ERR=260) (TITLE4(I),I=1,11)
    READ (4,1380,END=260,ERR=260) XPAGE,YPAGE,DELTA X,DELTA Y,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)
    IOLCGR=1
    GO TO 240
240 WRITE (5,1080) NAMFIL
    WRITE (5,1090)
    CALL RLCCHAR (IANS)
    GO TO 150
250 WRITE (5,1375)
    WRITE (5,1050)
    CALL RLCCHAR (IANS)
    GO TO 150
C=====
C MAKE CORRECTIONS TO EXISTING GRAPH
C=====
270 CALL FRIGMS ('CLRSCRN ')
280 WRITE (5,1020)
    CALL RLCINT (IANS)
    IF ((IANS.GE.1).AND.(IANS.LE.10)) GC TO 290
    CALL FRIGMS ('CLRSCRN ')
    WRITE (5,1030) IANS
    GO TO 280
290 CALL FRIGMS ('CLRSCRN ')
    GO TO (300,380,440,560,750,730,800,820,830,150), IANS

```

```

C=====
C      CHANGE CURVE VARIABLE
C=====
300  CALL FRICMS ('CLRCSRN ')
    IF ((ICLGR-EQ.0) GC TO 310
    WRITE (5,1100)
    GO TO 280
310  ICLRVS=NCURVS
    IF ((NCLRVS.LT.4) ICURVS=NCURVS+1
    WRITE (5,1110) ICURVS
    CALL RCINT (IANS)
    IF ((IANS.GE.1).AND.(IANS.LE.ICURVS)) GC 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,TEST)
    GO TO 370
340  N=2
    CALL SELCRV (N,C2,C2MIN,C2MAX, TITLE2,DATA,U,FBGC,NS,NC,NPTS,TEST)
    GO TO 370
350  N=3
    CALL SELCRV (N,C3,C3MIN,C3MAX, TITLE3,DATA,U,FBGC,NS,NC,NPTS,TEST)
    GO TO 370
360  N=4
    CALL SELCRV (N,C4,C4MIN,C4MAX, TITLE4,DATA,U,FBGC,NS,NC,NPTS,TEST)
    GO TO 370
370  GO TO 270
C=====
C      DELETE CURVE
C=====
380  WRITE (5,1120) NCURVS
    CALL RCINT (IANS)
    IF ((IANS.GE.1).AND.(IANS.LE.5)) GC TO (390,400,410,420,430), IANS
    WRITE (5,1130)
    GO TO 280
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 RCINT (ICRV)
    IF ((ICRV.GE.1).AND.(ICRV.LE.NCURVS)) GO TO 450

```

```

450 WRITE (5,1150) NCURVS
GO TO 440
460 WRITE (5,1160)
GO TO (460,470,480,490), ICRV
CALL CLRCFPR (TITLE1)
470 GO TO 500
CALL CLRCFPR (TITLE2)
480 GO TO 500
CALL CLRCFPR (TITLE3)
490 GO TO 500
CALL CLRCFPR (TITLE4)
500 CALL RLCHST (CHST)
DO 550 J=1,11
510 GO TO (510,520,530,540), ICRV
TITLE1(J)=CFST(J)
520 GO TO 550
TITLE2(J)=CHST(J)
530 GO TO 550
TITLE3(J)=CFST(J)
540 GO TO 550
TITLE4(J)=CFST(J)
550 CONTINUE
GO TO 270
C=====
C EDIT GRAPH HEADING
C=====
560 WRITE (5,1170)
CALL RLINT (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 RLINT (IHOG)
IF ((IHOG-GE.1).AND.(IHOG.LE.3)) GO TO 580
WRITE (5,1200)
GO TO 570
580 IF (NHEAD-LT.IHOG) NHEAD=IHOG
CALL FRTCMS ('CLRCRN ')
WRITE (5,1210)
GO TO (590,600,610), IHOG
590 CALL CLRCFPR (HEAD1)
GO TO 620
600 CALL CLRCFPR (HEAD2)
GO TO 620
610 CALL CLRCFPR (HEAD3)
620 CALL RLCHST (CHST)
DO 650 J=1,11
IF (IHOG-NE.1) GC TC 630

```

```

630 HEAD1(J)=CHST(J)
IF (IHCG.NE.2) GC TC 640
640 HEAD2(J)=CHST(J)
IF (IHCG.NE.3) GO TO 650
650 HEAD3(J)=CHST(J)
CONTINUE
GO TO 270
660 CALL FRICMS (' CLRSCRN ')
670 WRITE (5,1220) NHEAD
WRITE (5,1230) IANS)
CALL RCINT (IANS)
IF ((IANS-GE.1).AND.(IANS.LE.NHEAD)) GC TO (680,700,720), IANS
WRITE (5,1230) NHEAD
GO TO 670
680 DO 690 I=1,11
HEAD1(I)=HEAD2(I)
690 CONTINUE
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(NPTSCA)
CALL RCREAL (DOUBLP)
ANS=SNGL(DOUBLP)
IF ((ANS.GT.TSTART).AND.(ANS.LE.TIME(NPTSDA))) GO TO 740
WRITE (5,1250) ANS,TSTART,TIME(NPTSCA)
GO TO 730
740 NPTS=FIX(FLOAT(NPTSDA)*ANS/TIME(NPTSDA))
TSTOP=ANS
IF (NPTS.GT.NPTSDA) NPTS=NPTSDA
GO TO 270
C=====
C CHANGE CURVE Y DIRECT ION SCALE
C=====
750 WRITE (5,1260)
CALL RCINT (IANS)
WRITE (5,1270)
CALL RCREAL (DOUBLP)
YMIN=SNGL(DOUBLP)
WRITE (5,1280)
CALL RCREAL (DOUBLP)
YMAX=SNGL(DOUBLP)
IF (IANS.NE.1) GO TO 760
C1 MIN=YMIN

```

```

760 C1 MAX=YMAX
    IF ((ANS.NE.2)) GO TC 770
    C2 MIN=YMIN
    C2 MAX=YMAX
770 IF ((ANS.NE.3)) GO TC 780
    C3 MIN=YMIN
    C3 MAX=YMAX
780 IF ((ANS.NE.4)) GO TC 790
    C2 MIN=YMIN
    C2 MAX=YMAX
    GO TO 270
C=====
C      CHANGE PLOT SIZE
C=====
800 WRITE (5,1290)
    WRITE (5,1300)
    CALL CLREAL (XPAGE)
    CALL CLREAL (DCUBLP)
    ANS=SNGL(DCUBLP)
    IF ((ANS.LT.0.1)).OR.((ANS.GT.21.0)) GO TC 80C
    XPAGE=ANS
810 WRITE (5,1310)
    WRITE (5,1320)
    CALL CLREAL (YPAGE)
    CALL CLREAL (DCUBLP)
    ANS=SNGL(DCUBLP)
    IF ((ANS.LT.0.1)).OR.((ANS.GT.21.0)) GO TC 810
    YPAGE=ANS
    GO TO 270
C=====
C      CHANGE THE LETTERING HEIGHT
C=====
820 WRITE (5,1320)
    CALL CLREAL (SCALEF)
    CALL CLREAL (DCUBLP)
    SCALEH=SNGL(DCUBLP)
    IF (SCALEF.GT.0.0) GO TO 270
    WRITE (5,1330)
    GO TO 820
C=====
C      MOVE LEGEND BOX
C=====
830 WRITE (5,1340)
    CALL CLREAL (DCUBLP)
    ANS=SNGL(DCUBLP)
    DELTAX=DELTAX+ANS
    CALL FRICMS (CLRSCRN *)
    WRITE (5,1350)

```

```

CALL RCREAL (DCUBLP)
ANS=SNGL(DCUBLP)
DELTAY=DELTAY+ANS
GO TO 27C
=====
C=====
C      PLOT THE REVISED GRAPH
C=====
C=====
840  IF (ICMPRS.EQ.1) CALL DUNEPL
    ICMPRS=C
    GO TO 11C
=====
C=====
C      QUIT OR MAKE METAFILE MENU
C=====
C=====
850  WRITE (5,1040)
    CALL RECFAR (IANS)
    IF (IANS.EQ.1YES) GC TC 860
    IF (IANS.EQ.1NO) GC TC 870
    WRITE (5,1050)
    GO TO 88C
=====
860  CALL FILECV (NPITS,NCURVS,NHEAD,HEAD1,HEAD2,HEAD3,TITLE1
      1,TITLE2,TITLE3,TITLE4,XPAGE,YPAGE,DELTAY,SCALEH,
      2C1MIN,C1MAX,C2MIN,C2MAX,C3MIN,C3MAX,C4MIN,C4MAX,
      3TIME,C1,C2,C3,C4)
    CALL DUNEPL
    CALL FRICMS ('CLRSCRN ')
    WRITE (5,11360)
    CALL RDINT (IANS)
    IF ((IANS.GE.1).AND.(IANS.LE.2)) GO TO (890,930), IANS
    WRITE (5,11370)
    GO TO 88C
=====
C=====
C      MAKE METAFILE OF ANY PREVIOUSLY FILED CURVES
C=====
C=====
890  WRITE (5,1060)
    READ (5,1410)END=265,ERR=265) (NAMFIL)
    CALL FRICMS ('FILEDEF ',04
      1,A1
    WRITE (5,1070) NAMFIL
    REWIND 4
    READ (4,1350,END=920,ERR=920) NPITS,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)
    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)

```

```

910 IF (NCURVS.EQ.2) GC TC 910 (TITLE3(I),I=1,11)
    READ (4,1400,END=920,ERR=920)
    IF (NCURVS.EQ.3) GC TC 910
    READ (4,1400,END=920,ERR=920)
    READ (4,1380,END=920,ERR=920) XPAGE,YPAGE,DELTA X,DELTA Y,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 RECHAR (IANS)
    GO TO 88C
930 IF (ICMPRS.EQ.1) CALL DONEPL
    STOP
-----
940 FORMAT (//,10X,34HTHIS PORTION OF THE PROGRAM PLOTS:,,15X,13H- TH
1E STATES,,15X,26H- EXTERNAL CONTROL INPUTS,,15X,26H- FEEDBACK
2CONTROL ERRORS,,15X,39H- 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,739HFCW MANY VARIABLES DO YOU WISH TO PLOT MUST BE BETWEEN 1
970 FORMAT (//,10X,56HTHE NUMBER OF VARIABLES TO PLOT MUST BE BETWEEN 1
1 AND 4.,//,15X,1H(,13,31H) 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(,13,31H) 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 5. GRAPH:,,10X,27H 4. PLOT REVISED GRAPH ON THE TEK618:,,10X,
444H 5. CLIT AND/OR MAKE METAFILE OF THE CURVES:,,14X,17HPREVIOUSL
5Y SAVED:,,32HSELECT A NUMBER BETWEEN 1 AND 5.)
1010 FORMAT (//,15HYOUR INPUT OF (,13,38H) IS NOT WITHIN THE RANGE OF 1
1 TO 5.)
1020 FORMAT (//,20X,19HTHE GRAPH EDIT MENU:,,10X,56H 1. CHANGE VARIABLE
1S CR 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 HEADINGS(S):,,10X,28H 5. CHANGE THE Y-AXIS SCALE:,,10X,31
4H 6. CHANGE THE TIME AXIS SCALE:,,10X,45H 7. CHANGE PLOT SIZE: (3

```

```

1030 SEFAULT IS 8.5 X 6.0);/,10X,32H 8. CHANGE THE LETTERING HEIGHT.:/,1
1040 60X,34H 5. CHANGE POSITION OF THE LEGEND.:/,10X,21H10. EDITING COMP
1050 7LETE.:/,3FSELECT A NUMBER BETWEEN 1 AND 10.
1060 1000 10.
1070 1000 10.
1080 1000 10.
1090 1000 10.
1100 1000 10.
1110 1000 10.
1120 1000 10.
1130 1000 10.
1140 1000 10.
1150 1000 10.
1160 1000 10.
1170 1000 10.
1180 1000 10.
1190 1000 10.
1200 1000 10.
1210 1000 10.
1220 1000 10.
1230 1000 10.
1240 1000 10.

```

```

1250 1AN/,10X,12HEND TIME OF ,F9.2,9H SECCNDS.,/,10X,24HDATA IS AVAILABL
1260 2E UP TO ,F9.2,9H SECCNDS.,/,15X,34HWHAT IS THE DESIRED NEW END TI
1270 3ME.?)
1280 4FORMAT (/,10X,15HYOUR ANSWER GF ,F9.2,26H IS NOT WITHIN THE ALLOW
1290 5ED/,10X,9H RANGE OF ,F9.2,4H TO ,F9.2,9H SECCNDS.)
1300 6FORMAT (/,10X,49HON WHICH CURVE DO YOU WANT TO CHANGE THE Y-SCALE?,
1310 7/,17X,33HENTER CURVE NUMBER-1, 2, 3, CR 4)
1320 8FORMAT (/,10X,42HWHAT IS THE NEW Y-MIN VALUE AT THE ORIGIN?)
1330 9FORMAT (/,10X,28HWHAT IS THE NEW Y-MAX VALUE?)
1340 10FORMAT (/,15X,51HWHAT IS THE DESIRED DIMENSION IN THE "X" DIRECTION
1350 11?)
1360 12FORMAT (/,10X,32H -THE MAX LENGTH IS 21.0 INCHES.)
1370 13FORMAT (/,5X,51HWHAT IS THE DESIRED DIMENSION IN THE "Y" DIRECTION
1380 14?)
1390 15FORMAT (/,10X,45HTHE LETTERING SCALE FACTOR WILL BE MULTIPLIED/,1
1400 1610X,35HTIMES THE CURRENT LETTERING HEIGHT.,/,5X,58HI.E. A NUMBER
1410 17GREATER THAN 1.0 INCREASES, AND VICE VERSA.,/,10X,30HWHAT SCALE FA
1420 18CTOR OR DC YOU WANT?)
1430 19FORMAT (/,37HYOUR ANSWER MUST BE GREATER THAN 0.0.)
1440 20FORMAT (/,10X,35HCHCK MANY INCHES IN THE X DIRECTION ,/,10X,36H(LEF
1450 21T CR RIGHT), DO YOU WANT TO MOVE,/,10X,45HMOVE THE LEGEND BOX FROM
1460 22ITS PRESENT POSITION,/,10X,40HNOTE: 1. DEFAULT PLOT SIZE IS 8.5 X
1470 233 X 6.0,/,10X,26H 2. LEFT IS NEGATIVE ,/,10X,23H 3. UP IS PO
1480 24SITIVE)
1490 25FORMAT (/,10X,35HCHCK MANY INCHES IN THE Y DIRECTION ,/,10X,33H(UP
1500 26OR DOWN), DC YOU WANT TO MOVE,/,10X,45HMOVE THE LEGEND BOX FROM IT
1510 27S PRESENT POSITION,/,10X,40HNOTE: 1. DEFAULT PAGE SIZE IS 8.5 X
1520 2836.0,/,10X,26H 2. DOWN IS NEGATIVE ,/,10X,23H 3. UP IS PO
1530 29SITIVE)
1540 30FORMAT (/,10X,36HTHE FOLLOWING OPTIONS ARE AVAILABLE: ,/,10X,44H1
1550 31MAKE METAFILE OF PREVIOUSLY SAVED CURVE.,/,10X,9H2. QUIT.,/,1
1560 325X,12HENTER 1 OR 2)
1570 33FORMAT (/,10X,33HYCUR INPUT MUST BE EITHER 1 OR 2.)
1580 34FORMAT (/,10X,33HYCUR INPUT NULL STRING NOT ALLOWED. TRY AGAIN.)
1590 35FORMAT (5E14.7)
1600 36FORMAT (5I15)
1610 37FORMAT (11A4)
1620 38FORMAT (2A4)
1630 39ENC
1640 40C=====
1650 41SUBROUTINE SELCRV (I,C,CMIN,CMAX,TITLE,DATA,U,FBGC,NS,NC,NPTS,
1660 421TEST)
1670 43C=====
1680 44C=====
1690 45C=====
1700 46C=====
1710 47C=====
1720 48C=====
1730 49C=====
1740 50C=====
1750 51C=====
1760 52C=====
1770 53C=====
1780 54C=====
1790 55C=====
1800 56C=====
1810 57C=====
1820 58C=====
1830 59C=====
1840 60C=====
1850 61C=====
1860 62C=====
1870 63C=====
1880 64C=====
1890 65C=====
1900 66C=====
1910 67C=====
1920 68C=====
1930 69C=====
1940 70C=====
1950 71C=====
1960 72C=====
1970 73C=====
1980 74C=====
1990 75C=====
2000 76C=====
2010 77C=====
2020 78C=====
2030 79C=====
2040 80C=====
2050 81C=====
2060 82C=====
2070 83C=====
2080 84C=====
2090 85C=====
2100 86C=====
2110 87C=====
2120 88C=====
2130 89C=====
2140 90C=====
2150 91C=====
2160 92C=====
2170 93C=====
2180 94C=====
2190 95C=====
2200 96C=====
2210 97C=====
2220 98C=====
2230 99C=====
2240 100C=====
2250 101C=====
2260 102C=====
2270 103C=====
2280 104C=====
2290 105C=====
2300 106C=====
2310 107C=====
2320 108C=====
2330 109C=====
2340 110C=====
2350 111C=====
2360 112C=====
2370 113C=====
2380 114C=====
2390 115C=====
2400 116C=====
2410 117C=====
2420 118C=====
2430 119C=====
2440 120C=====
2450 121C=====
2460 122C=====
2470 123C=====
2480 124C=====
2490 125C=====
2500 126C=====
2510 127C=====
2520 128C=====
2530 129C=====
2540 130C=====
2550 131C=====
2560 132C=====
2570 133C=====
2580 134C=====
2590 135C=====
2600 136C=====
2610 137C=====
2620 138C=====
2630 139C=====
2640 140C=====
2650 141C=====
2660 142C=====
2670 143C=====
2680 144C=====
2690 145C=====
2700 146C=====
2710 147C=====
2720 148C=====
2730 149C=====
2740 150C=====
2750 151C=====
2760 152C=====
2770 153C=====
2780 154C=====
2790 155C=====
2800 156C=====
2810 157C=====
2820 158C=====
2830 159C=====
2840 160C=====
2850 161C=====
2860 162C=====
2870 163C=====
2880 164C=====
2890 165C=====
2900 166C=====
2910 167C=====
2920 168C=====
2930 169C=====
2940 170C=====
2950 171C=====
2960 172C=====
2970 173C=====
2980 174C=====
2990 175C=====
3000 176C=====
3010 177C=====
3020 178C=====
3030 179C=====
3040 180C=====
3050 181C=====
3060 182C=====
3070 183C=====
3080 184C=====
3090 185C=====
3100 186C=====
3110 187C=====
3120 188C=====
3130 189C=====
3140 190C=====
3150 191C=====
3160 192C=====
3170 193C=====
3180 194C=====
3190 195C=====
3200 196C=====
3210 197C=====
3220 198C=====
3230 199C=====
3240 200C=====
3250 201C=====
3260 202C=====
3270 203C=====
3280 204C=====
3290 205C=====
3300 206C=====
3310 207C=====
3320 208C=====
3330 209C=====
3340 210C=====
3350 211C=====
3360 212C=====
3370 213C=====
3380 214C=====
3390 215C=====
3400 216C=====
3410 217C=====
3420 218C=====
3430 219C=====
3440 220C=====
3450 221C=====
3460 222C=====
3470 223C=====
3480 224C=====
3490 225C=====
3500 226C=====
3510 227C=====
3520 228C=====
3530 229C=====
3540 230C=====
3550 231C=====
3560 232C=====
3570 233C=====
3580 234C=====
3590 235C=====
3600 236C=====
3610 237C=====
3620 238C=====
3630 239C=====
3640 240C=====
3650 241C=====
3660 242C=====
3670 243C=====
3680 244C=====
3690 245C=====
3700 246C=====
3710 247C=====
3720 248C=====
3730 249C=====
3740 250C=====
3750 251C=====
3760 252C=====
3770 253C=====
3780 254C=====
3790 255C=====
3800 256C=====
3810 257C=====
3820 258C=====
3830 259C=====
3840 260C=====
3850 261C=====
3860 262C=====
3870 263C=====
3880 264C=====
3890 265C=====
3900 266C=====
3910 267C=====
3920 268C=====
3930 269C=====
3940 270C=====
3950 271C=====
3960 272C=====
3970 273C=====
3980 274C=====
3990 275C=====
4000 276C=====
4010 277C=====
4020 278C=====
4030 279C=====
4040 280C=====
4050 281C=====
4060 282C=====
4070 283C=====
4080 284C=====
4090 285C=====
4100 286C=====
4110 287C=====
4120 288C=====
4130 289C=====
4140 290C=====
4150 291C=====
4160 292C=====
4170 293C=====
4180 294C=====
4190 295C=====
4200 296C=====
4210 297C=====
4220 298C=====
4230 299C=====
4240 300C=====
4250 301C=====
4260 302C=====
4270 303C=====
4280 304C=====
4290 305C=====
4300 306C=====
4310 307C=====
4320 308C=====
4330 309C=====
4340 310C=====
4350 311C=====
4360 312C=====
4370 313C=====
4380 314C=====
4390 315C=====
4400 316C=====
4410 317C=====
4420 318C=====
4430 319C=====
4440 320C=====
4450 321C=====
4460 322C=====
4470 323C=====
4480 324C=====
4490 325C=====
4500 326C=====
4510 327C=====
4520 328C=====
4530 329C=====
4540 330C=====
4550 331C=====
4560 332C=====
4570 333C=====
4580 334C=====
4590 335C=====
4600 336C=====
4610 337C=====
4620 338C=====
4630 339C=====
4640 340C=====
4650 341C=====
4660 342C=====
4670 343C=====
4680 344C=====
4690 345C=====
4700 346C=====
4710 347C=====
4720 348C=====
4730 349C=====
4740 350C=====
4750 351C=====
4760 352C=====
4770 353C=====
4780 354C=====
4790 355C=====
4800 356C=====
4810 357C=====
4820 358C=====
4830 359C=====
4840 360C=====
4850 361C=====
4860 362C=====
4870 363C=====
4880 364C=====
4890 365C=====
4900 366C=====
4910 367C=====
4920 368C=====
4930 369C=====
4940 370C=====
4950 371C=====
4960 372C=====
4970 373C=====
4980 374C=====
4990 375C=====
5000 376C=====
5010 377C=====
5020 378C=====
5030 379C=====
5040 380C=====
5050 381C=====
5060 382C=====
5070 383C=====
5080 384C=====
5090 385C=====
5100 386C=====
5110 387C=====
5120 388C=====
5130 389C=====
5140 390C=====
5150 391C=====
5160 392C=====
5170 393C=====
5180 394C=====
5190 395C=====
5200 396C=====
5210 397C=====
5220 398C=====
5230 399C=====
5240 400C=====
5250 401C=====
5260 402C=====
5270 403C=====
5280 404C=====
5290 405C=====
5300 406C=====
5310 407C=====
5320 408C=====
5330 409C=====
5340 410C=====
5350 411C=====
5360 412C=====
5370 413C=====
5380 414C=====
5390 415C=====
5400 416C=====
5410 417C=====
5420 418C=====
5430 419C=====
5440 420C=====
5450 421C=====
5460 422C=====
5470 423C=====
5480 424C=====
5490 425C=====
5500 426C=====
5510 427C=====
5520 428C=====
5530 429C=====
5540 430C=====
5550 431C=====
5560 432C=====
5570 433C=====
5580 434C=====
5590 435C=====
5600 436C=====
5610 437C=====
5620 438C=====
5630 439C=====
5640 440C=====
5650 441C=====
5660 442C=====
5670 443C=====
5680 444C=====
5690 445C=====
5700 446C=====
5710 447C=====
5720 448C=====
5730 449C=====
5740 450C=====
5750 451C=====
5760 452C=====
5770 453C=====
5780 454C=====
5790 455C=====
5800 456C=====
5810 457C=====
5820 458C=====
5830 459C=====
5840 460C=====
5850 461C=====
5860 462C=====
5870 463C=====
5880 464C=====
5890 465C=====
5900 466C=====
5910 467C=====
5920 468C=====
5930 469C=====
5940 470C=====
5950 471C=====
5960 472C=====
5970 473C=====
5980 474C=====
5990 475C=====
6000 476C=====
6010 477C=====
6020 478C=====
6030 479C=====
6040 480C=====
6050 481C=====
6060 482C=====
6070 483C=====
6080 484C=====
6090 485C=====
6100 486C=====
6110 487C=====
6120 488C=====
6130 489C=====
6140 490C=====
6150 491C=====
6160 492C=====
6170 493C=====
6180 494C=====
6190 495C=====
6200 496C=====
6210 497C=====
6220 498C=====
6230 499C=====
6240 500C=====
6250 501C=====
6260 502C=====
6270 503C=====
6280 504C=====
6290 505C=====
6300 506C=====
6310 507C=====
6320 508C=====
6330 509C=====
6340 510C=====
6350 511C=====
6360 512C=====
6370 513C=====
6380 514C=====
6390 515C=====
6400 516C=====
6410 517C=====
6420 518C=====
6430 519C=====
6440 520C=====
6450 521C=====
6460 522C=====
6470 523C=====
6480 524C=====
6490 525C=====
6500 526C=====
6510 527C=====
6520 528C=====
6530 529C=====
6540 530C=====
6550 531C=====
6560 532C=====
6570 533C=====
6580 534C=====
6590 535C=====
6600 536C=====
6610 537C=====
6620 538C=====
6630 539C=====
6640 540C=====
6650 541C=====
6660 542C=====
6670 543C=====
6680 544C=====
6690 545C=====
6700 546C=====
6710 547C=====
6720 548C=====
6730 549C=====
6740 550C=====
6750 551C=====
6760 552C=====
6770 553C=====
6780 554C=====
6790 555C=====
6800 556C=====
6810 557C=====
6820 558C=====
6830 559C=====
6840 560C=====
6850 561C=====
6860 562C=====
6870 563C=====
6880 564C=====
6890 565C=====
6900 566C=====
6910 567C=====
6920 568C=====
6930 569C=====
6940 570C=====
6950 571C=====
6960 572C=====
6970 573C=====
6980 574C=====
6990 575C=====
7000 576C=====
7010 577C=====
7020 578C=====
7030 579C=====
7040 580C=====
7050 581C=====
7060 582C=====
7070 583C=====
7080 584C=====
7090 585C=====
7100 586C=====
7110 587C=====
7120 588C=====
7130 589C=====
7140 590C=====
7150 591C=====
7160 592C=====
7170 593C=====
7180 594C=====
7190 595C=====
7200 596C=====
7210 597C=====
7220 598C=====
7230 599C=====
7240 600C=====
7250 601C=====
7260 602C=====
7270 603C=====
7280 604C=====
7290 605C=====
7300 606C=====
7310 607C=====
7320 608C=====
7330 609C=====
7340 610C=====
7350 611C=====
7360 612C=====
7370 613C=====
7380 614C=====
7390 615C=====
7400 616C=====
7410 617C=====
7420 618C=====
7430 619C=====
7440 620C=====
7450 621C=====
7460 622C=====
7470 623C=====
7480 624C=====
7490 625C=====
7500 626C=====
7510 627C=====
7520 628C=====
7530 629C=====
7540 630C=====
7550 631C=====
7560 632C=====
7570 633C=====
7580 634C=====
7590 635C=====
7600 636C=====
7610 637C=====
7620 638C=====
7630 639C=====
7640 640C=====
7650 641C=====
7660 642C=====
7670 643C=====
7680 644C=====
7690 645C=====
7700 646C=====
7710 647C=====
7720 648C=====
7730 649C=====
7740 650C=====
7750 651C=====
7760 652C=====
7770 653C=====
7780 654C=====
7790 655C=====
7800 656C=====
7810 657C=====
7820 658C=====
7830 659C=====
7840 660C=====
7850 661C=====
7860 662C=====
7870 663C=====
7880 664C=====
7890 665C=====
7900 666C=====
7910 667C=====
7920 668C=====
7930 669C=====
7940 670C=====
7950 671C=====
7960 672C=====
7970 673C=====
7980 674C=====
7990 675C=====
8000 676C=====
8010 677C=====
8020 678C=====
8030 679C=====
8040 680C=====
8050 681C=====
8060 682C=====
8070 683C=====
8080 684C=====
8090 685C=====
8100 686C=====
8110 687C=====
8120 688C=====
8130 689C=====
8140 690C=====
8150 691C=====
8160 692C=====
8170 693C=====
8180 694C=====
8190 695C=====
8200 696C=====
8210 697C=====
8220 698C=====
8230 699C=====
8240 700C=====
8250 701C=====
8260 702C=====
8270 703C=====
8280 704C=====
8290 705C=====
8300 706C=====
8310 707C=====
8320 708C=====
8330 709C=====
8340 710C=====
8350 711C=====
8360 712C=====
8370 713C=====
8380 714C=====
8390 715C=====
8400 716C=====
8410 717C=====
8420 718C=====
8430 719C=====
8440 720C=====
8450 721C=====
8460 722C=====
8470 723C=====
8480 724C=====
8490 725C=====
8500 726C=====
8510 727C=====
8520 728C=====
8530 729C=====
8540 730C=====
8550 731C=====
8560 732C=====
8570 733C=====
8580 734C=====
8590 735C=====
8600 736C=====
8610 737C=====
8620 738C=====
8630 739C=====
8640 740C=====
8650 741C=====
8660 742C=====
8670 743C=====
8680 744C=====
8690 745C=====
8700 746C=====
8710 747C=====
8720 748C=====
8730 749C=====
8740 750C=====
8750 751C=====
8760 752C=====
8770 753C=====
8780 754C=====
8790 755C=====
8800 756C=====
8810 757C=====
8820 758C=====
8830 759C=====
8840 760C=====
8850 761C=====
8860 762C=====
8870 763C=====
8880 764C=====
8890 765C=====
8900 766C=====
8910 767C=====
8920 768C=====
8930 769C=====
8940 770C=====
8950 771C=====
8960 772C=====
8970 773C=====
8980 774C=====
8990 775C=====
9000 776C=====
9010 777C=====
9020 778C=====
9030 779C=====
9040 780C=====
9050 781C=====
9060 782C=====
9070 783C=====
9080 784C=====
9090 785C=====
9100 786C=====
9110 787C=====
9120 788C=====
9130 789C=====
9140 790C=====
9150 791C=====
9160 792C=====
9170 793C=====
9180 794C=====
9190 795C=====
9200 796C=====
9210 797C=====
9220 798C=====
9230 799C=====
9240 800C=====
9250 801C=====
9260 802C=====
9270 803C=====
9280 804C=====
9290 805C=====
9300 806C=====
9310 807C=====
9320 808C=====
9330 809C=====
9340 810C=====
9350 811C=====
9360 812C=====
9370 813C=====
9380 814C=====
9390 815C=====
9400 816C=====
9410 817C=====
9420 818C=====
9430 819C=====
9440 820C=====
9450 821C=====
9460 822C=====
9470 823C=====
9480 824C=====
9490 825C=====
9500 826C=====
9510 827C=====
9520 828C=====
9530 829C=====
9540 830C=====
9550 831C=====
9560 832C=====
9570 833C=====
9580 834C=====
9590 835C=====
9600 836C=====
9610 837C=====
9620 838C=====
9630 839C=====
9640 840C=====
9650 841C=====
9660 842C=====
9670 843C=====
9680 844C=====
9690 845C=====
9700 846C=====
9710 847C=====
9720 848C=====
9730 849C=====
9740 850C=====
9750 851C=====
9760 852C=====
9770 853C=====
9780 854C=====
9790 855C=====
9800 856C=====
9810 857C=====
9820 858C=====
9830 859C=====
9840 860C=====
9850 861C=====
9860 862C=====
9870 863C=====
9880 864C=====
9890 865C=====
9900 866C=====
9910 867C=====
9920 868C=====
9930 869C=====
9940 870C=====
9950 871C=====
9960 872C=====
9970 873C=====
9980 874C=====
9990 875C=====
10000 876C=====
10010 877C=====
10020 878C=====
10030 879C=====
10040 880C=====
10050 881C=====
10060 882C=====
10070 883C=====
10080 884C=====
10090 885C=====
10100 886C=====
10110 887C=====
10120 888C=====
10130 889C=====
10140 890C=====
10150 891C=====
10160 892C=====
10170 893C=====
10180 894C=====
10190 895C=====
10200 896C=====
10210 897C=====
10220 898C=====
10230 899C=====
10240 900C=====
10250 901C=====
10260 902C=====
10270 903C=====
10280 904C=====
10290 905C=====
10300 906C=====
10310 907C=====
10320 908C=====
10330 909C=====
10340 910C=====
10350 911C=====
10360 912C=====
10370 913C=====
10380 914C=====
10390 915C=====
10400 916C=====
10410 917C=====
10420 918C=====
10430 919C=====
10440 920C=====
10450 921C=====
10460 922C=====
10470 923C=====
10480 924C=====
10490 925C=====
10500 926C=====
10510 927C=====
10520 928C=====
10530 929C=====
10540 930C=====
10550 931C=====
10560 932C=====
10570 933C=====
10580 934C=====
10590 935C=====
10600 936C=====
10610 937C=====
10620 938C=====
10630 939C=====
10640 940C=====
10650 941C=====
10660 942C=====
10670 943C=====
10680 944C=====
10690 945C=====
10700 946C=====
10710 947C=====
10720 948C=====
10730 949C=====
10740 950C=====
10750 951C=====
10760 952C=====
10770 953C=====
10780 954C=====
10790 955C=====
10800 956C=====
10810 957C=====
10820 958C=====
10830 959C=====
10840 960C=====
10850 961C=====
10860 962C=====
10870 963C=====
10880 964C=====
10890 965C=====
10900 966C=====
10910 967C=====
10920 968C=====
10930 969C=====
10940 970C=====
10950 971C=====
10960 972C=====
10970 973C=====
10980 974C=====
10990 975C=====
11000 976C=====
11010 977C=====
11020 978C=====
11030 979C=====
11040 980C=====
11050 981C=====
11060 982C=====
11070 983C=====
11080 984C=====
11090 985C=====
11100 986C=====
11110 987C=====
11120 988C=====
11130 989C=====
11140 990C=====
11150 991C=====
11160 992C=====
11170 993C=====
11180 994C=====
11190 995C=====
11200 996C=====
11210 997C=====
11220 998C=====
11230 999C=====
11240 1000C=====
11250 1001C=====
11260 1002C=====
11270 1003C=====
11280 1004C=====
11290 1005C=====
11300 1006C=====
11310 1007C=====
11320 1008C=====
11330 1009C=====
11340 1010C=====
11350 1011C=====
11360 1012C=====
11370 1013C=====
11380 1014C=====
11390 1015C=====
11400 1016C=====
11410 1017C=====
11420 1018C=====
11430 1019C=====
11440 1020C=====
11450 1021C=====
11460 1022C=====
11470 1023C=====
11480 1024C=====
11490 1025C=====
11500 1026C=====
11510 1027C=====
11520 1028C=====
11530 1029C=====
11540 1030C=====
11550 1031C=====
11560 1032C=====
11570 1033C=====
11580 1034C=====
11590 1035C=====
11600 1036C=====
11610 1037C=====
11620 1038C=====
11630 1039C=====
11640 1040C=====
11650 1041C=====
11660 1042C=====
11670 1043C=====
11680 1044C=====
11690 1045C=====
11700 1046C=====
11
```

```

=====
C=====
C      CALL FRICMS ('CLRSCRN ')
C      SELECT TYPE OF VARIABLE TO PLOT
C=====
10=====
C      WRITE (5,260) I
C      CALL RCINT (ITYPE)
C      IF ((ITYPE.GE.1).AND.(ITYPE.LE.5)) GO TO 20
C      GO TO 10
20=====
C      GO TO (30,60,120,160,200), ITYPE
C=====
C      SELECT STATE VARIABLE
C=====
30=====
C      WRITE (5,290) I
C      CALL RCINT (IANS)
C      IF ((IANS.GE.1).AND.(IANS.LE.NS)) GC TO 40
C      WRITE (5,320) NS,IANS
C      GO TO 30
40=====
C      C(1)=DATA(1,IANS)
C      CMIN=C(1)
C      CMAX=C(1)
C      DO 50 J=2,NPTS
C      C(J)=DATA(J,IANS)
C      IF (C(J).LT.CMIN) CMIN=C(J)
C      IF (C(J).GT.CMAX) CMAX=C(J)
C      CONTINUE
50=====
C      GO TO 240
C=====
C      SELECT FEEDBACK <C>*X
C=====
60=====
C      CALL FRICMS ('CLRSCRN ')
C      WRITE (5,270) I
C      CALL RCINT (IANS)
C      IF ((IANS.GE.1).AND.(IANS.LE.NS)) GC TC 80
C      WRITE (5,320) NS,IANS
C      GO TO 70
80=====
C      CONTINUE
C      DO 100 I=1,NPTS
C      C(I)=0.
C      DO 90 J=1,NS
C      C(I)=C(I)+FBGC(IANS,J)*DATA(I,J)
C      IF (I.EST.NE.3) C(I)=C(I)+FBGC(IANS,J)*DATA(I,NS+J)
C      CONTINUE
90=====
C      CONTINUE
C      CMAX=C(1)
C      CMIN=C(1)
C      DO 110 I=2,NPTS
C      IF (C(I).LT.CMIN) CMIN=C(I)
C      IF (C(I).GT.CMAX) CMAX=C(I)
100=====

```

```

110 CONTINUE
C=====
C      GO TO 240
C      SELECT CONTROL INPUT
C=====
120 CALL FRICMS ('CLRSCRN ')
130 WRITE (5,280) I
    CALL RDIINT (IANS)
    IF ((IANS.GE.1).AND.(IANS.LE.NC)) GC TC 140
    WRITE (5,320) NC
    GO TO 130
140 C(I)=U(I,IANS)
    CMIN=C(I)
    CMAX=C(I)
    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) GC TC 170
    WRITE (5,300)
    GO TO 10
170 WRITE (5,310) I
    CALL RDIINT (IANS)
    IF ((IANS.GE.1).AND.(IANS.LE.NS)) GC TO 180
    WRITE (5,320) NS,IANS
    GO TO 170
180 C(I)=DATA(1,IANS+NS)
    CMIN=C(I)
    CMAX=C(I)
    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 RDIINT (IANS)

```

```

220 IF ((IANS-GE.1).AND.(IANS.LE.NS)) GC TO 220
    WRITE (5,320) NS,IANS
    GO TO 210
    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)
    CONTINUE
    CALL FRICMS ('CLRSCRN ')
    IF (CMIN.NE.CMAX) GC TO 250
    WRITE (5,330)
    CMAX=1.C+CMIN
    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,12,1H?,//,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.,X*AT1,X*AT2,ETC.),//,15X,60H5. STATE RECONSTRUCTION ERROR (
5IE.,X1-X*AT1,X2-X*AT2,ETC.),//,10X,18HENTER 1,2,3,4 OR 5)
270 FORMAT (//,10X,50HWAT IS THE SUBSCRIPT OF THE FEEDBACK CONTROL TH
1AT,//,10X,30HYOU WANT TO PLOT AS THE NUMBER,12,15H CURVE VS TIME?,/
FORMAT (//,10X,47HWAT IS THE SUBSCRIPT CF THE CONTROL INPUT THAT,/
1,10X,30HYOU WANT TO PLOT AS THE NUMBER,13,15H CURVE VS TIME?,/
290 FORMAT (//,10X,48HWAT IS THE SUBSCRIPT OF THE STATE VARIABLE THAT
1,//,10X,30HYOU WANT TO PLOT AS THE NUMBER,13,15H CURVE VS TIME?,/
300 FORMAT (//,30X,8HERROR???,15X,24HSTATE ESTIMATES WERE NOT,12H CAL
1CULATE.)
310 FORMAT (//,10X,48HWAT IS THE SUBSCRIPT OF THE STATE ESTIMATE THAT
1,//,10X,30HYOU WANT TO PLOT AS THE NUMBER,13,15H CURVE VS TIME?,/
320 FORMAT (//,10X,35H THE SUBSCRIPT MUST BE BETWEEN 1 AND,13,2H .,1H(,1
12,30H) WAS THE NUMBER YOU SELECTED.)
330 FORMAT (//,10X,48HTHIS VARIABLE IS A CONSTANT THROUGHOUT THE TIME
1,10HEVALUATED.)
340 FORMAT (//,10X,42HWAT IS THE CURVE LABEL FOR THIS VARIABLE?,/)
350 FORMAT (//,10X,34HNOTE: 1. 40 CHARACTERS MAX LENGTH,10X,55H
1 2. ENCLGSEC IN PARENTHESIS,10X,27H
2 3. 10X,27H (B) => BETA,10X,27H
3 4 PH I,10X,27H (Q) => THETA)
360 FORMAT (//,10X,43HWAT IS THE SUBSCRIPT OF THE RECONSTRUCTION,10X
1,41HERRCR THAT YOU WANT TO PLOT AS THE NUMBER,13,7H CURVE?)

```

```

=====
ENC
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)
CONTINUE
DO 20 I=1,501
C1(I)=C2(I)
CONTINUE
C1MAX=C2MAX
C1MIN=C2MIN
RETURN
ENC
C=====
SUBROUTINE HEADS (HEAD,N)
C  GETS THE HEADING CHARACTER STRING
C=====
INTEGER HEAD(11),N
CALL FRTCMS ('CLRSRN ')
WRITE (5,10) N
WRITE (5,20)
CALL RDCHST (HEAD)
RETURN
C=====
10  FORMAT (/,10X,35HWHAT IS THE DESIRED HEADING NUMBER ,11,1H?,/)
20  FORMAT (/,10X,34HNOTE: 1. 40 CHARACTERS MAX LENGTH,/,10X,55H
1    2. GREEK SYMBOLS WILL BE PRINTED FOR ANY LETTERS,/,10X,34H
3/,10X,27H (B) => BETA ,/,10X,27H IE. {A} => ALPHA ,
4  PHI ,/,10X,27H (Q) => THETA )
ENC
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)
ENC
C=====
SUBROUTINE CURREAL (ANS)
C  DISPLAYS THE CURRENT VALUE OF A REAL VARIABLE

```



```

20 CONTINUE
   WRITE (4,50) XPAGE,YPAGE,DELTA X,DELTA Y,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)
25 RETURN
   WRITE (5,50)
   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 (5I15)
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 INADVERTENTLY ENTERS AN IMPROPER
C DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
SUBROUTINE RDINT (IANS)
  INTEGER COUNT
C-----
  COUNT=C
  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.)

```

```

=====
C SUBROUTINE ROCHAR -- INTERACTIVELY READS A CHARACTER STRING REPLY
C (YES, OR NO) INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY
C ENTERS A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY
C=====
SUBROUTINE ROCHAR (IANS)
  INTEGER CCUNT
  COUNT=0
  CONTINUE
  COUNT=CCUNT+1
  IF (CCUNT.LT.3) GO TO 20
  WRITE (5,60)
  GO TO 40
  CONTINUE
  REWIND
  REAC (5,70,END=30,ERR=30) IANS
  RETURN
  REWIND
  WRITE (5,50)
  GO TO 10
  CONTINUE
  STOP
=====
C SUBROUTINE RCHWARNING: NULL STRINGS ARE NOT ALLOWED, ENTER "YES"
C OR "NO"
C
C FORMAT (1X,60HWARNING: NULL STRINGS ARE NOT ALLOWED, ENTER "YES"
C FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED )
C FORMAT (A1)
C END
=====
C SUBROUTINE RCREAL -- INTERACTIVELY READS A REAL NUMBER REPLY
C INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS A NULL
C STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
SUBROUTINE RCREAL (ANSR)
  REAL*8 ANSR
  INTEGER CCUNT
  COUNT=0
  CONTINUE
  COUNT=CCUNT+1
  IF (CCUNT.LT.3) GO TO 20
  WRITE (5,60)
  GO TO 40
  CONTINUE
  REAC (5,*,END=30,ERR=30) ANSR
  RETURN
=====

```

```

30  REWIND 5
    WRITE (5,50)
    GO TO 10
40  CONTINUE
    STCP
C-----
50  FORMAT (1X,64HWARNING: NULL STRINGS ARE NOT ALLOWED, ENTER A NOME
    1RICAL VALUE.)
60  FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED )
    ENCL
C=====
C  SUBROUTINE RDCST -- INTERACTIVELY READS A CHARACTER STRING REPLY
C  UP TO 40 CHARACTERS LONG AND FORMATS THE CHARACTER STRING FOR USE
C  BY A DISPLA PRINT ROUTINE.
C=====
C  SUBROUTINE RDCST (CHST)
    INTEGER CHST(1),I
    DATA IEL/,I,ICCL/'$ ' /
C-----
    CALL GETCHS (CHST)
    CHST(1) = IBL
    DO 10 I = 1,11
        IF (CHST(I),NE.IBL) GO TO 10
        CHST(I) = IDCL
        GO TO 20
    10 CONTINUE
    20 RETURN
C-----
    ENCL
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=====
C  SUBROUTINE GETCHS (CHST)
    INTEGER COUNT,CHST(20),I
C-----
    COUNT=C
    CONTINUE
    COUNT=COUNT+1
    IF (COUNT,LT.3) GO TO 20
    WRITE (5,60)
    GO TO 40
    CONTINUE
    20 CONTINUE
    READ (5,70,END=30,ERR=30) (CHST(I),I = 1,10)
    RETURN
    30 REWIND 5

```

```

40      WRITE (5,50)
      GO TO 10
      CONTINUE
      STOP
C-----
50      FORMAT (1X,'WARNING: NULL STRINGS ARE NOT ALLOWED, THE PROGRAM',
60      1/, 'WILL TERMINATE IF ANOTHER NULL STRING IS ENTERED.')
```

FORMAT (///,5X,4HPROGRAM 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 (OPTSYS) 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

END

FILMED

5-85

DTIC

