

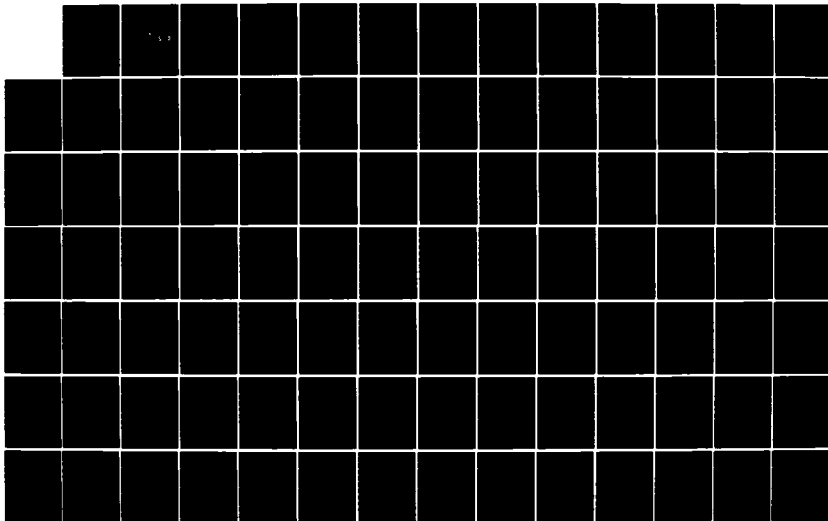
AD-A152 148

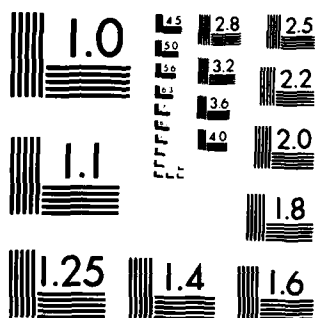
DEVELOPMENT OF GRAPHICAL POLE-ZERO ROOT-LOCUS BODE
NYQUIST AND NICHOLS RESPONSES USING THE OPTSYSX PROGRAM
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M H LAPTAS
SEP 84 F/G 9/2

1/3

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC
ELECTE
APR 5 1985
S B D

AD-A152 148

DTIC FILE COPY

THESIS

DEVELOPMENT OF GRAPHICAL POLE-ZERO,
ROOT-LOCUS, BODE, NYQUIST, AND NICHOLS
RESPONSES USING THE OPTSYSX PROGRAM

by

Michael Henry Laptas

September 1984

Thesis Advisor:

Daniel J. Collins

Approved for public release; distribution unlimited

85 03 18 015

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Development of Graphical Pole-Zero, Root-Locus, Bode, Nyquist, and Nichols Responses Using the OPTSYSX Program		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1984
7. AUTHOR(s) Michael Henry Laptas		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE September 1984
		13. NUMBER OF PAGES 219
		15. SECURITY CLASS. (of this report)
		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) OPTSYSX Nyquist Noise. Pole-Zero Nichols Compensator, Root-Locus Transfer function <i>etc.</i> Bode Open Loop		
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 (OPTSYS) originally obtained from Professor Arthur E. Bryson of Stanford University. This program has been subsequently redesigned to run interactively on the IBM 3033 VM/CMS by Lieutenant Commander John G. Hoden, and additions by Commander Harry A. Diel provide the user with a highly accurate graphic time response to a system designed using the OPTSYSX program.		

The addition of the FORTRAN program OPTGRAPH gives the user the capability to obtain classical analysis (Pole-Zero Map, Root Locus, Bode, Nyquist, and Nichols) of transfer functions, calculated for a system designed with the OPTSYSX program. The OPTGRAPH program uses high resolution precision plotting software to provide the user with highly accurate frequency response plots.

This series of programs permits users to rapidly carry out simulation, analysis, and design of Optimal Systems Control problems in a totally interactive mode.

Computer supplied by user - included - > p 1

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	

Distribution/	
...	
...	
...	



A-1

Approved for public release; distribution unlimited.

Development of Graphical Pole-Zero, Root-Locus, Bode,
Nyquist, and Nichols Responses using the
OPTSYSX Program

by

Michael H. Laptas
Lieutenant, United States Navy
B.S., Purdue University, Lafayette, 1977

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:

Michael H. Laptas
Michael H. Laptas

Approved by:

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

D.M. Layton
D.M. Layton, Second Reader

M.F. Platz

M.F. Platz, Chairman,
Department of Aeronautics

J.N. Dyer

J.N. Dyer,
Dean of Science and Engineering

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. This program has been subsequently redesigned to run interactively on the IBM 3033 VM/CMS by Lieutenant Commander John G. Hoden, and additions by Commander Harry A. Diel provide the user with a highly accurate graphic time response to a system designed using the OPTSYSX program.

The addition of the FORTRAN program OPTGRAPH gives the user the capability to obtain classical analysis (Pole-Zero Map, Root Locus, Bode, Nyquist, and Nichols) of transfer functions, calculated for a system designed with the OPTSYSX program. The OPTGRAPH program uses high resolution precision plotting software to provide the user with highly accurate frequency response plots.

This series of programs permits users to rapidly carry out simulation, analysis, and design of Optimal Systems Control problems in a totally interactive mode.

TABLE OF CONTENTS

I.	INTRODUCTION	9
II.	THE OPTSYSX COMPUTER PROGRAM	11
	A. BACKGROUND	11
	B. OVERVIEW	11
	C. OPTSYSX MODIFICATIONS	12
	D. OPTSYS EXEC MODIFICATIONS	13
III.	THE OPGRAPH PROGRAM	15
	A. PROGRAM OVERVIEW	15
	1. Program Language	15
	2. Graphics Package	15
	3. Library Subroutines	15
	4. Program Composition	16
	B. GENERAL PROGRAM OPERATION	16
	1. User Protection Features	16
	2. Large Order System	17
	C. SYSTEM/MODEL DESCRIPTION	18
	D. PROGRAM ORGANIZATION	19
	1. Main Program	19
	2. Program Sequencing	20
	3. Graphic and/or Tabular System Response	21
	E. INTERACTIVE EXAMPLES	23
	F. CHARACTERISTIC EQUATION ANALYSIS	24
	1. Pole-Zero Map	24
	2. Pole-Zero Map (Interactive Example)	25
	3. Root-Locus	33
	4. Root-Locus (Interactive Example)	34

G.	FREQUENCY RESPONSE ANALYSIS	43
1.	Bode Response	44
2.	Bode Response (Interactive Example) . . .	45
3.	Nyquist Response	51
4.	Nyquist Response (Interactive Example) . .	52
5.	Nichols Response	52
6.	Nichols Response (Interactive Example) . .	52
IV.	CONCLUSIONS AND RECOMMENDATIONS	59
A.	CONCLUSIONS	59
B.	RECOMMENDATIONS	60
1.	Root Finding Routines	60
2.	Program Memory Requirements	61
3.	Further Modifications	61
	APPENDIX A: OPTSYS EXEC LISTING	62
	APPENDIX B: OPTSYSX PROGRAM LISTING	71
	APPENDIX C: OPTGRAPH PROGRAM LISTING	160
	APPENDIX D: OPGRAPH LISTING	215
	LIST OF REFERENCES	217
	BIBLIOGRAPHY	218
	INITIAL DISTRIBUTION LIST	219

LIST OF FIGURES

3.1	Pole-Zero Map Example	53
3.2	Root Locus Response Example	54
3.3	Bode Response Example (Magnitude)	55
3.4	Bode Response Example (Phase)	56
3.5	Nyquist Response Example	57
3.6	Nichols Response Example	58

SYMBOLS

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

I. INTRODUCTION

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

The Optimal Systems control program was originally developed by Hall [Ref. 1] to support his research in rotary-wing aircraft control systems. Later program modifications were made by Walker [Ref. 2] and Liu [Ref. 3] of Stanford University, and are designated OPTSYS 4 and OPTSYS 5 respectively. OPTSYS modifications made by Hoden [Ref. 4] were primarily devoted to creating a user-friendly interactive version (OPTSYSX) of the OPTSYS 4 program. The latest modifications by Diel [Ref. 5] allowed the user to save matrices for subsequent runs of the OPTSYSX program, and formed data file sets for the time response program OPTCALC.

The intent of this thesis work was to develop an interactive program to plot the Pole-Zero map, Root-Locus, Bode, Nyquist, and Nichols responses to the open loop, closed loop noise, and compensator transfer functions of a State Variable Control System which has been developed using OPTSYS Program. Minor modifications to the OPTSYSX Program were necessary to construct a data file sets for plotting the Pole-Zero Map, and in calculations for the Root-Locus, Bode, Nyquist, and Nichols plots.

It is assumed that the user is familiar with the basic concepts of Control Theory and Optimal Systems Design. The symbol/naming conventions of Bryson [Ref. 6] are used in the discussion of program operations and descriptions of problems using the OPTSYS System.

An overview of the OPTSYSX Program capabilities and of modifications to the existing programs is presented first. This is followed by a description of the program (OPTGRAPH), which was developed to plot the Pole-Zero Map and to perform calculations for the Root-Locus, Bode, Nyquist, and Nichols plots.

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. Complete program listings for the OPTSYS EXEC program, OPTSYSX program, and OPTGRAPH program are included in appendices A,B, and C respectively.

II. THE OPTSYSX COMPUTER PROGRAM

A. BACKGROUND

OPTSYSX is a double-precision, interactive FORTRAN program employing modern control theory analysis techniques. Its capabilities include the calculation of the open-loop eigensystem, and the stationary closed loop system; the synthesis of regulators and filters; along with power spectral density, and modal distribution computations. The modifications introduced to the OPTSYSX program by this thesis work do not affect the program's original capabilities.

B. OVERVIEW

OPTSYSX is an extremely large and complex program with over 3000 lines of code. To use the program in its small version (dimensioned for a 32 X 32 "F", "G", and "H" matrices), the user must extend his Virtual Machine (VM) memory capacity beyond the default VM memory capacity of 720 kilobytes to 1024 kilobytes.

Any significant increase in the OPTSYSX program size, with the the resulting requirement for additional memory capacity, would cause user difficulties. For this reason the task of obtaining the Pole-Zero maps, Root-Locus, and Bode, Nyquist, Nichols plots for the open loop, noise and compensator transfer functions was relegated to a separate program (OPTGRAPH). Three data files (OPTGROL DATA, OPTGRNO DATA, and OPTGRCM DATA), containing the open loop, noise, and compensator transfer functions, are formed by the OPTSYS program to be transferred to the OPTGRAPH program. The OPTSYS EXEC program provides an interface between the

OPTSYSX program, the time response programs (OPTCALC and OPTPLCT), and the OPTGRAPH program.

C. OPTSYSX MODIFICATIONS

OPTSYSX program modifications consist of the addition of write statements, three flags, and a short routine to interpret user inputs.

Write statements, to input data to the OPTGROL, OPTGRNO, and OPTGRCM DATA files were added to the main program, subroutine INNER, subroutine ZEROS, and subroutine POLES. System information consisting of number of states (Ns), number of controls (Nc), number of measurements (No), number of process noise sources (Ng), type transfer function (ITFX), Markov Parameter (IE), and two flags to be used by the OPTSYS EXEC and the OPTGRAPH programs is inputted from the main program. The write statements added to the subroutine INNER, and subroutine POLES input the poles for the open loop and noise transfer functions, and compensator transfer function respectively. The zeros, the numerator order, the gain, and the input and output numbers for all three transfer functions are obtained from the subroutine ZEROS.

The first of the two flags, input with the system information from the main program to the three data files, serves as a marker for the OPTSYS EXEC to locate the correct data line for reading system inputs. The second flag, which is set by the routine to interpret user inputs, serves as a signal for the OPTSYS EXEC to either load the OPTGRAPH program if any one of the three transfer functions was calculated by the OPTSYSX program, and abort loading the OPTGRAPH program if none of the three transfer functions were calculated. The third flag (ITFX) is an existing OPTSYSX program flag. The flag (ITFX) is passed from the

subroutine TF to the subroutine ZEROS to identify the calculation of either an open loop, noise, or compensator transfer function zeros.

The short routine added to the main program of OPTSYSX reads the flags ITF1, ITF2, ITF3, and IRET. The flag sent to the CPTGROL, OPTGRNO, or OPTGRCM DATA files is set to allow loading of transfer function data file to the OPTGRAPH program if the user selects either of the transfer function options number two (calculate poles, residues, and zeros - ITF1 = 1, ITF2 = 1, or ITF3 = 1) number three (calculate poles, and zeros - ITF1 = 2, ITF2 = 2, or, ITF3 = 2) when calculating the open loop, noise, or compensator transfer functions. The selection of any other transfer function option will not calculate the transfer function zeros. The flag IRET is read to determine if the user desires to rerun a problem without exiting the OPTSYSX program. Should the user elect to rerun a problem without exiting OPTSYSX (IRET = 1), the three data files are set back to the beginning (REWIND) to accept the new problem system data.

D. OPTSYS EXEC MODIFICATIONS

The OPTSYS EXEC was written by Diel [Ref. 5] to interface between the OPTSYSX program and the time response programs OPTCALC and CPTPLOT. The modifications to the exec program were to add the option to select running the OPTGRAPH program to the exec menus and a short section to interface the OPTGRAPH program with the OPTSYSX program.

The added section to interface between the OPTSYSX and OPTGRAPH programs defined loader size (LDRTBLE), file definitions (FILEDEF), and text libraries (TXTLIB) for the CPTGRAPH program. The first line of the three data files is read to ensure that there has been at least one of the three transfer functions calculated by the OPTSYSX program prior to loading OPTGRAPH program.

The OPTSYS EXEC routine is also used to send plots to the VERSATEC plotter from the OPTPLOT program and the OPTGRAPH program.

III. THE OPGRAPH PROGRAM

A. PROGRAM OVERVIEW

CPTGRAPH is an interactive FORTRAN program which uses the transfer function poles and zeros calculated by OPTSYSX to calculate, tabulate, and plot the Pole-Zero Map, Pode, Nyquist, Nichols, and Root-Locus responses for either the open loop, closed loop noise, or compensator transfer functions.

1. Program Language

CPTGRAPH is programmed in FORTRAN following the conventions of the IBM System /360/370/ FORTRAN IV language. CPTGRAPH has been compiled and run under both FORTRAN IV (G1) and FORTRAN H (extended) compilers on the IBM 3033.

2. Graphics Package

This program uses the Display Integrated Software System and Plotting System Software (DISSPLA) developed and distributed by the Integrated Software Systems Corporation (ISSCC) of San Diego, California. The DISSPLA package is a library of FORTRAN subroutines using FORTRAN IV conventions.

3. Library Subroutines

CPTGRAPH uses the International Mathematical and Statistical Library (IMSL) subroutine ZRPOLY to calculate the roots of a polynomial equation. ZRPOLY is a double precision FORTRAN subroutine to which is input a polynomial with real coefficients (double precision) in terms of decreasing powers of the polynomial. The subroutine finds the roots of the polynomial and returns a double precision

complex array containing the roots. ZRPOLY is capable of accepting polynomials greater than 0 order and less than 101 order.

4. Program Composition

CPTGRAPH has one main program and 24 subroutines. The main program and its subroutines may be divided into four basic categories:

- 1) File Data Input
- 2) Interactive Data Input
- 3) Plot Setup and Sequencing
- 4) Calculation

A brief and general description of the program and its subroutines will follow in subsequent sections.

B. GENERAL PROGRAM OPERATION

OFTGRAPH was written to satisfy two specific objectives. The first is to create a program to be used as a instructional tool for students taking controls related courses, and the second is as method to assist in the analysis of actual large order control system problems.

To accomplish these objectives the emphasis while writing this program was to make the program as user oriented as possible by attempting to eliminate ambiguities and providing features which would protect the user from inadvertent wrong entries. An effort was also made to minimize the Virtual Machine (VM) memory required to execute the program to facilitate the analysis of a large order system.

1. User Protection Features

Three basic methods were used to protect the user from an abnormal program termination in the case of an inadvertent wrong entry. The first was to construct the

subroutines to display a summary of the user inputs and allowing changes prior to leaving the subroutine. For the case where the user input is expected to fall within a given range, the input is examined and the user is either allowed to continue for an input within the range or issued an error or warning and reasked the question for an input outside the range. The final method is used by the three subroutines which read the user input from the screen. The subroutines RDINT, RDREAL, and RICHAR expect an integer input, real number input, and a logical "yes" or "no" respectively. In the event that a "null" line is inadvertently entered, the user is issued a warning and allowed another opportunity to enter the correct input before abnormally terminating the program. The entry of two "null" lines also offers the user the option to exit the OPTGRAPH program at other than normal program exit points. The subroutine RDCHAR also examines the input and issues a warning if the input is not either "yes" or "no".

2. Large Order System

The large order control problem being considered for analysis by the OPTSYSX and the OPTGRAPH programs is the X-29A aircraft longitudinal axis backup mode system. This system has a (98 X 98) "F" matrix, a (2 X 98) "H" matrix, and a (98 X 1) "G" matrix.

To accommodate this large order system the OPTGRAPH program was dimensioned to accept a system with a maximum of 99 states (Ns), 12 controls (Nc), 12 process noise sources (Ng), and 12 measurements (No). The Virtual Machine (VM) memory requirements for the program were minimized by the reuse of memory locations allocated for storage of calculated data after that data has been either tabulated or plotted. This method of memory reuse has allowed the OPTGRAPH program to be dimensioned to accommodate a large

order system and still operate with less than the 1024 KILOBYTES VM memory required to run OPTSYSX in its small version.

C. SYSTEM/MODEL DESCRIPTION

The system equations used by the OPTSYSX program for the calculation of the transfer functions are of the state variable form. The system equations are:

system model

$$\dot{x} = [F]*x + [G]*u + [GAM]*w \quad (3.1)$$

measurement equation

$$z = [H]*x + [D]*u + v \quad (3.2)$$

estimator equation

$$\dot{x}_e = [F]*x_e + [G]*u + [K]*(z - [H]*x_e) \quad (3.3)$$

open loop transfer function

$$[H]*[s[I] - [F]]^{-1}*[G] \quad (3.4)$$

closed loop noise transfer function

$$[H]*[s[I] - [F]]^{-1}*[GAM] \quad (3.5)$$

compensator transfer function from measurement to input

$$[C]*([s[I] - [F]] + [G]*[C] + [K]*[H])^{-1}*[K] \quad (3.6)$$

where

u = Control vector (Ns X 1)

w = White process noise vector ($N_g \times 1$)
 x = State vector ($N_s \times 1$)
 $\dot{x}$ = Derivative of the state vector ($N_s \times 1$)
 y = Output vector ($N_o \times 1$)
 z = measurement vector ($N_o \times 1$)
 $[C]$ = control gain matrix (N_c, N_s)
 $[D]$ = control feed-forward distribution matrix (N_c, N_c)
of the control vector (u)
 $[F]$ = open-loop dynamics matrix (N_s, N_s)
(System matrix or Plant matrix)
 $[G]$ = control distribution matrix (N_s, N_c)
 $[GAM]$ = state disturbance distribution matrix (N_s, N_g)
 $[H]$ = measurement distribution matrix (N_o, N_s)
 $[I]$ = identity matrix (N_s, N_s)

D. PROGRAM ORGANIZATION

1. Main Program

The main program presents a menu allowing the user to select for analysis either the open loop transfer function, the closed loop noise transfer function, the compensator transfer function, or exit the OPTGRAPH program. The data file for the selected transfer function is examined by the main program to ensure that the file contains the required transfer function data. Should the data file be incomplete the program gives the user a warning message indicating the nature of the missing data and allows the user to select another option.

The main program menu:

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

Data is read from the selected file by the main program and formed into three arrays and seven integer variables. The exchange of data between the main program and the subroutines, and between the subroutines is by the extensive use of labeled common statements. The data transferred through the subroutine call statements is limited primarily to flags, and constants. The program moves to its primary sequencing subroutine (GRAPH) upon the completion of file data entry.

2. Program Sequencing

The subroutine GRAPH first presents the user with a summary of the the transfer system received from the data file. The user is then asked which transfer function he desires to to analyze using the conventions of the OPTSYSX program describing the transfer functions by input number and output number. A menu is presented which allows the user to select graphical and/or tabular system response (Pole-Zero map, Root-locus, Bode, Nyquist, or Nichols), select another transfer function, or exit the OPTGRAPH program. Upon completion of a graphical and/or tabular response option, or change of transfer function option, the program returns to the GRAPH menu to allow the user to choose another option or exit to the main program.

Subroutine GRAPH data summary and menu:

OPTGRAPH

THE OPEN LOOP TRANSFER SYSTEM OBTAINED FROM OPTSYS CONTAINS:

- | | |
|-----------------------|---|
| 1. NUMBER OF STATES = | 4 |
|-----------------------|---|

2. NUMBER OF CCNTROLS (INPUTS) = 1
3. NUMBER OF MEASUREMENTS (OUTPUTS) = 1
4. MARKOV PARAMETER = 10.0**- 6

CLEAR SCREEN TO CONTINUE

OPTGRAPH

AN OPEN LOOP POLE-ZERO, ROOT LOCUS, BODE, NYQUIST, AND/
OR NICHOLS PLOT IS DESIRED FOR:

INPUT # ?

OUTPUT # ?

OPTGRAPH

OPEN LOOP TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR DATA FOR:

1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME TRANSFER
FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT OPTGRAPH

ENTER OPTICN NUMBER.

3. Graphic and/or Tabular System Response

The selection of a graphical and/or tabular system response option for an Pole-Zero map, Root-Locus, Bode, open loop Nyquist, or open loop Nichols calls the subroutines

PZERO, RTLO, BODE, NYQST, or NICHOL respectively. While the response of these subroutines is different, their organization is similar.

The subroutines PZERO, RTLO, BODE, NYQST, and NICHOL primarily act as sequencing and plot setup subroutines. The interactive questioning of the user for graphic/tabular response information and the system response calculation has been relegated to other subroutines. The subroutines PZERO, RTLO, BODE, NYQST, and NICHOL are divided into two major sections with the first being devoted to producing a graphical output and the second to producing a tabular output.

The user is presented a subroutine menu offering the option for graphic response, tabular response, or exiting the subroutine. The program moves to either the first section (graphic response), second section (tabular response) or back to the subroutine GRAPH menu depending on the option selection. The program returns to the subroutine menu upon completion of either the graphic response, or tabular response section, allowing the user to select another response option, or exit the subroutine.

Features that are common to the five subroutines will be described here and features that are unique to one of the subroutines will be described in the following section under the appropriate response heading.

For a graphic response, the user is given the choice of two printer options (TEK 618, or VERSATEC). With the selection of the TEK 618, the plot page size is defaulted to 11 inches by 8.5 inches, and the plot will be presented on the TEK 618 screen. The VERSATEC option gives the user the option to create up to a 21 inch by 21 inch plot (maximum for the VERSATEC printer) and causes a DISPLA METAFILE to be created. A scaling subroutine (PSCALE) scales the plot heading, legend, etc. in proportion to the page width selected. The VERSATEC option offers the the advantage of a

high quality print from the VERSATEC printer using the VERSATEC printer option in the OPTSYS EXEC. The creation of a DISPLA METAFILE also gives the user (after he exits optsys) the opportunity to use printers other than the TEK 618 or VERSATEC. A major disadvantage with the VERSATEC option is that the user must exit the OPTGRAPH program before he can print a graphic system response. For this reason, it is recommended that the system response first be plotted on the TEK 618 prior to selecting the VERSATEC option.

For a tabular response, the user is presented a menu which gives him the choice of three devices (screen, printer, or disk) to send the tabular output. The user is asked if he desires to make any changes, after the tabular data has been calculated. If the answer is affirmative, he is presented a menu and allowed to make changes and rerun the tabular data routine.

Examples of these features are included in the following sections with the interactive examples for the system responses.

E. INTERACTIVE EXAMPLES

The open-loop, closed loop noise, and compensator transfer functions are handled identically by the OPTGRAPH program, with the only difference being the program and graph headings identifying the transfer function type. The frequency response programs for the Bode, Nyquist, and Nichols responses use identical setup and tabular data routines.

A good overall representation of the OPTGRAPH program capabilities will be demonstrated by single interactive terminal session examples for the pole-zero, root locus, and bode responses using selected transfer functions. Tabular

data examples will be provided for the Pole-Zero, Root Locus, and Bode responses, and graphic response examples will be provided the Pole-Zero, Root Locus, Bode, Nyquist, and Nichols responses.

F. CHARACTERISTIC EQUATION ANALYSIS

The subroutines PZERO and RTLO are similar in their basic formats. The pole and zero locations for the selected transfer function input and output numbers are read into two single dimension arrays. The extraneous zeros, calculated by the OPTSYSX program, are eliminated by comparing the zeros to the Markov parameter sent with the system information from the OPTSYSX program. For a graphical response, the user is asked to define the plot limits in terms of X-coordinates (real axis) and Y-coordinates (imaginary axis). Points for the plot which fall outside these limits will be ignored. System information (transfer function input number, output number, and (DC) gain) is listed in the graph legend.

1. Pole-Zero Map

The subroutine PZERO converts the double precision numbers for pole and zero locations to single precision for compatibility with the Graphics Package and plots the poles and zeros within the plot limits defined by the user for the graphical response routine. For tabular data response the user is given system information (numerator order, denominator order, and transfer function (DC) gain) for the selected transfer function input number and output number, and the pole and zero locations.

2. Pole-Zero Map (Interactive Example)

The following example of a simplified open loop transfer function for an airplane with an autopilot in the longitudinal mode was obtained from [Ref. 7,p.315].

The aircraft system is represented by:

$$[D] = [0]$$

$$[F] = \begin{bmatrix} 0. & 1. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 1. \\ 0. & 16. & -12. & -3. \end{bmatrix}$$

$$[G] = \begin{bmatrix} 0. \\ 0. \\ 1. \\ -2. \end{bmatrix}$$

$$[GAM] = [0]$$

$$[H] = [1. \quad 0. \quad 0. \quad 0.]$$

a. Example of Graphic Response (Terminal Session)

BEGIN RECORDING OF TERMINAL SESSION

R; T=0.01/0.02 19:40:17

EXECUTION BEGINS...

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

1

OPTGRAPH

THE OPEN LOOP TRANSFER SYSTEM OBTAINED FROM OPTSYS
CONTAINS:

1. NUMBER OF STATES = 4
2. NUMBER OF CONTROLS (INPUTS) = 1
3. NUMBER OF MEASUREMENTS (OUTPUTS) = 1
4. MARKOV PARAMETER = 10.0×10^{-6}

CLEAR SCREEN TO CONTINUE

OPTGRAPH

AN OPEN LOOP POLE-ZERO, ROOT LOCUS, BODE, NYQUIST,
AND? OR NICHOLS PLOT IS DESIRED FOR:

INPUT # ?

1

OUTPUT # ?

1

OPTGRAPH

OPEN LOOP TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP

2. ROOT-LOCUS
3. EODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT
CPTGRAPH

ENTER OPTICN NUMBER.

1

POLE-ZERO MAP

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTICN NUMBER.

1

POLE-ZERO MAP

OPEN LOOP TRANSFER FUNCTION

PLOTTER SELECTION AND PAGE SIZE

(NOTE: PAGE IS 11.0 INCHES BY 8.5 INCHES WITH
SELECTION OF TEK618)

WHICH PLOTTER DO YOU DESIRE:

1. TEK618
2. VERSATEC

CHOCSE OPTION 1 OR 2

2

PAGE SIZE (MAXIMUM = 21.0 INCHES BY 21.0
INCHES)

HEIGHT =

6

WIDTH =

8

PLOTTING LIMITS FOR GRAPH

X AXIS (REAL AXIS)

X MINIMUM = ?

-15

X MAXIMUM = ?

5

Y AXIS (IMAGINARY AXIS)

Y MINIMUM = ?

-5

Y MAXIMUM = ?

5

PLOTTING LIMITS FOR GRAPH

X AXIS (REAL AXIS)

X MINIMUM = -15.00

X MAXIMUM = 5.00

Y AXIS (IMAGINARY AXIS)

Y MINIMUM = -5.00

Y MAXIMUM = 5.00

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NO".

n

PLOT HEADING

HOW MANY LINES OF HEADING DO YOU DESIRE ?

(3MAX)

ENTER NUMBER OF LINES.

(ENTER "0" FOR NO HEADING)

3

A MAXIMUM OF 32 CHARACTERS PER LINE IS
ALLCWED

LINE 1 IS:

aircraft with an autopilot in

LINE 2 IS:
the longitudinal mode

LINE 3 IS:
ref. 7; page 315

PLOT HEADING

LINE 1 : AIRCRAFT WITH AN AUTOPILOT IN
LINE 2 : THE LONGITUDINAL MODE
LINE 3 : REF. 7; PAGE 315
DO YOU DESIRE TO MAKE ANY CHANGES ?
TYPE "YES" OR "NO".

n
>USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.
>USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.

A DISSPLA METAFILE HAS BEEN CREATED
CLEAR SCREEN TO CONTINUE

POLE-ZERO MAP
OPEN LOOP TRANSFER FUNCTION
DO YOU DESIRE TO MAKE ANY CHANGES TO:
1. PLOTTER / PAGE SIZE
2. GRAPH LIMITS
3. HEADING
4. NO ADDITIONAL CHANGES - PLOT POLE-ZERO
MAP
5. NO CHANGES - EXIT POLE-ZERO PLOTTING
ROUTINE
ENTER OPTION NUMBER.
5

POLE-ZERO MAP
OPEN LOOP TRANSFER FUNCTION
DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTICN NUMBER.

3

OPTGRAPH

OPEN LOOP TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT

CPTGRAPH

ENTER OPTICN NUMBER.

7

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTICN NUMBER.

4

```
R; T=2.80/4.67 19:45:15
record off
END RECORDING OF TERMINAL SESSION
```

The preceding example follows at the end of the chapter as figure 3.1 .

b. Example of Tabular Data (Terminal Session)

The entering and exiting option menus for the tabular data terminal session are identical to the graphic response terminal session option menus, and have been eliminated from the following terminal session.

```
BEGIN RECORDING OF TERMINAL SESSION
```

```
R; T=0.01/0.02 19:51:03
```

```
EXECUTION BEGINS...
```

```
POLE-ZERO MAP
```

```
OPEN LOCF TRANSFER FUNCTION
```

```
DO YOU DESIRE:
```

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

```
ENTER OPTICN NUMBER.
```

```
2
```

```
POLE-ZERO MAP
```

```
OPEN LOCF TRANSFER FUNCTION
```

```
DO YOU DESIRE TABULAR OUTPUT TO GO TO:
```

1. SCREEN
2. PRINTER
3. DISK (OPGRAPH LISTING)

```
ENTER OPTICN NUMBER.
```

```
1
```

POLE-ZERO MAP
 OPEN LOOP TRANSFER FUNCTION
 INPUT NUMBER = 1
 OUTPUT NUMBER = 1
 TRANSFER FUNCTION (DC) GAIN = 0.1000D+01
 DENOMINATOR ORDER = 4

POLE LOCATIONS

REAL PART	IMAGINARY PART
0.0000D+00	0.0000D+00
0.1000D+01	0.0000D+00
-0.2000D+01	0.3464D+01
-0.2000D+01	-0.3464D+01

NUMERATOR ORDER = 1

ZERO LOCATIONS

REAL PART	IMAGINARY PART
-0.1000D+01	0.0000D+00

DO YOU DESIRE TO CHANGE OUTPUT DEVICE?

TYPE "YES" OR "NC".

n

POLE-ZERO MAP
 OPEN LOOP TRANSFER FUNCTION
 DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTION NUMBER.

3

R; T=1.60/3.23 19:53:00

record off

END RECORDING OF TERMINAL SESSION

3. Root-Locus

The subroutine RTLO creates a closed loop system (with negative unity feed-back) from the selected transfer function input numbers and output numbers. Transfer function poles and zeros are formed into two separate polynomials by the subroutine MAKPOL. The numerator polynomial (zeros) is multiplied by the gains, and the two polynomials are added to form a single polynomial. Complex roots of the of the polynomial are calculated by the IMSL library subroutine ZRPCLY. The complex roots are separated into real and imaginary components for either plotting or tabulating.

The plotting routine first plots the transfer function pole and zero locations then plots the root locations as they are received from the subroutine ZRPOLY. For the plotting routine the gain interval specified by the user is divided into 2000 evenly spaced points for the subroutine ZRPOLY to calculate roots. The poles, zeros, and roots are plotted within the plot limits specified by the user.

For the tabular data routine the user is given the option to choose between 1 and 500 points for the subroutine ZRPOLY to calculate roots. The tabular data output provided for the user is system information (numerator order, denominator order, and transfer function (DC) gain) for the selected transfer function input number and output number, the pole and zero locations, and the root locus gains and roots

A positive unity feedback system may be analyzed by specifying a negative gain range vice a positive gain range.

4. Root-Locus (Interactive Example)

The following interactive example is of the open loop transfer function for an aircraft with an autopilot in the longitudinal mode described in Pole-Zero interactive section.

a. Example of Graphic Response (Terminal Session)

BEGIN RECORDING OF TERMINAL SESSION

R; T=0.01/0.02 20:28:37

EXECUTION BEGINS...

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

1

OPTGRAPH

THE OPEN LOOP TRANSFER SYSTEM OBTAINED FROM OPTSYS
CONTAINS:

- | | |
|---------------------------------------|---|
| 1. NUMBER OF STATES = | 4 |
| 2. NUMBER OF CONTROLS (INPUTS) = | 1 |
| 3. NUMBER OF MEASUREMENTS (OUTPUTS) = | 1 |
| 4. MARKOV PARAMETER = 10.0**- 6 | |

CLEAR SCREEN TO CONTINUE

OPTGRAPH

AN OPEN LOOP POLE-ZERO, ROOT LOCUS, BODE, NYQUIST,

AND/OR NICHOLS PLOT IS DESIRED FOR:

INPUT # ?

1

OUTPUT # ?

1

OPTGRAPH

OPEN LOCF TRANSFER FUNCTION

INPUT # = 1

CUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT

OPTGRAPH

ENTER OPTION NUMBER.

2

ROOT-LOCUS

OPEN LOCF TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTION NUMBER.

1

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

PLOTTER SELECTION AND PAGE SIZE

(NOTE: PAGE IS 11.0 INCHES BY 8.5 INCHES WITH
SELECTION OF TEK618)

WHICH PLOTTER DO YOU DESIRE:

1. TEK618

2. VERSATEC

CHOCSE OPTION 1 CR 2

2

PAGE SIZE (MAXIMUM = 21.0 INCHES BY 21.0
INCHES)

HEIGHT =

6

WIDTH =

8

PLOTTING LIMITS FOR GRAPH

X AXIS (REAL AXIS)

X MINIMUM = ?

-15

X MAXIMUM = ?

5

Y AXIS (IMAGINARY AXIS)

Y MINIMUM = ?

-5

Y MAXIMUM = ?

5

PLOTTING LIMITS FOR GRAPH

X AXIS (REAL AXIS)

X MINIMUM = -15.00

X MAXIMUM = 5.00

Y AXIS (IMAGINARY AXIS)

Y MINIMUM = -5.00

Y MAXIMUM = 5.00

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NC".

n

GAIN RANGE FOR ROOT-LOCUS PLOT

MINIMUM GAIN = ?

0

MAXIMUM GAIN = ?

500

GAIN RANGE FOR ROOT-LOCUS PLOT

MINIMUM GAIN = 0.000D+00

MAXIMUM GAIN = 0.500D+03

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NC".

n

PLOT HEADING

HOW MANY LINES OF HEADING DO YOU DESIRE ?

(3 MAX)

ENTER NUMBER OF LINES.

(ENTER "0" FOR NO HEADING)

3

A MAXIMUM OF 32 CHARACTERS PER LINE IS
ALLOWED

LINE 1 IS:

aircraft with an autopilot in

LINE 2 IS:

the longitudinal mode

LINE 3 IS:

ref. 7; page 315

PLOT HEADING

LINE 1 : AIRCRAFT WITH AN AUTOPILOT IN
LINE 2 : THE LONGITUDINAL MODE
LINE 3 : REF. 7; PAGE 315

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NO".

n

>USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.

>USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.

A DISPLA METAFILE HAS BEEN CREATED
CLEAR SCREEN TO CONTINUE

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE TO MAKE ANY CHANGES TO:

1. PLOTTER / PAGE SIZE
2. GRAPH LIMITS
3. HEADING
4. GAIN RANGE
5. NO ADDITIONAL CHANGES - PLOT ROOT-LOCUS
6. NO CHANGES - EXIT ROOT-LOCUS PLOTTING

ROUTINE

ENTER OPTION NUMBER.

6

RCOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTION NUMBER.

3

OPTGRAPH

OPEN LOOP TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT

OPTGRAPH

ENTER OPTION NUMBER.

7

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

4

R; T=12.36/15.14 20:32:54

record off

END RECORDING OF TERMINAL SESSION

The preceding example follows at the end of the chapter as figure 3.2 .

b. Example of tabular data (Terminal Session)

The entering and exiting option menus for the tabular data terminal session are identical to the graphic response terminal session option menus, and have been eliminated from the following terminal session.

BEGIN RECORDING OF TERMINAL SESSION

R; T=0.01/0.02 20:42:59

EXECUTION BEGINS...

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTION NUMBER.

2

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE TABULAR OUTPUT TO GO TO:

1. SCREEN
2. PRINTER
3. DISK (OPGRAPH LISTING)

ENTER OPTION NUMBER.

1

GAIN RANGE FOR ROOT-LOCUS PLOT

MINIMUM GAIN = ?

1

MAXIMUM GAIN = ?

5

GAIN RANGE FOR ROOT-LOCUS PLOT

MINIMUM GAIN = 0.100D+01

MAXIMUM GAIN = 0.500D+01

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NC".

n

HOW MANY POINTS DO YOU WANT TO TABULATE?

(500 IS THE MAXIMUM)

ENTER NUMBER OF POINTS.

5

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

INPUT NUMBER = 1

OUTPUT NUMBER = 1

TRANSFER FUNCTION (DC) GAIN = 0.1000D+01

OPEN LOOP TF POLES AND ZEROS

DENOMINATOR ORDER = 4

POLE LOCATIONS

REAL PART	IMAGINARY PART
0.0000D+00	0.0000D+00
0.1000D+01	0.0000D+00
-0.2000D+01	0.3464D+01
-0.2000D+01	-0.3464D+01

NUMERATOR ORDER = 1

ZERO LOCATIONS

REAL PART	IMAGINARY PART
-0.1000D+01	0.0000D+00

GAIN = 0.1000D+01

REAL PART	IMAGINARY PART
0.7074D-01	0.0000D+00
0.8962D+00	0.0000D+00
-0.1983D+01	0.3441D+01

-0.1983D+01 -0.3441D+01

GAIN = 0.1800D+01

REAL PART	IMAGINARY PART
0.1453D+00	0.0000D+00
0.7948D+00	0.0000D+00
-0.1970D+01	0.3422D+01
-0.1970D+01	-0.3422D+01

GAIN = 0.2600D+01

REAL PART	IMAGINARY PART
0.2576D+00	0.0000D+00
0.6552D+00	0.0000D+00
-0.1956D+01	0.3402D+01
-0.1956D+01	-0.3402D+01

GAIN = 0.3400D+01

REAL PART	IMAGINARY PART
0.4426D+00	0.1661D+00
0.4426D+00	-0.1661D+00
-0.1943D+01	0.3383D+01
-0.1943D+01	-0.3383D+01

GAIN = 0.4200D+01

REAL PART	IMAGINARY PART
0.4285D+00	0.3096D+00
0.4285D+00	-0.3096D+00
-0.1929D+01	0.3363D+01
-0.1929D+01	-0.3363D+01

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NO".

Y

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE TO MAKE ANY CHANGES TO:

1. OUTPUT DEVICE

2. GAIN RANGE
3. NUMBER OF POINTS CALCULATED
4. NO ADDITIONAL CHANGES - TABULATE DATA
5. EXIT ROOT-LCCUS TABULAR DATA ROUTINE

ENTER OPTICN NUMBER.

5

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTICN NUMBER.

3

R; T=1.63/3.40 20:44:54

record off

END RECORDING OF TERMINAL SESSION

G. FREQUENCY RESPONSE ANALYSIS

The calculations for the frequency response of the Bode, Nyquist, and Nichols responses are identical. The only difference is in the graphical presentation of the calculated data. The subroutines BODE, NYQST, and NICHOLS call the subroutine FREQ to calculate the transfer function magnitude and phase for the frequency range specified by the user.

The subroutine FREQ eliminates the extraneous zeros by comparing the zeros to the Markov parameter as described in the Characteristic Equation Analysis section. The frequency response magnitude and phase are calculated using double precision complex library functions following the

conventions of the IBM System /360/370/ FORTRAN IV language. The frequency response is calculated by determining the product of successive numerator zeros which have been combined with the frequency and divided by the product of successive denominator poles which also have been combined with the frequency. The magnitude for the complex result of the zeros, poles, and frequency calculations is determined using the FORTRAN double precision library functions for finding the magnitude of a complex argument. The phase of the of the complex result is determined by separating the real and imaginary parts of the complex results, and finding the phase angle using the double precision FORTRAN library function arctangent. The angle is converted from radians to degrees and is corrected for proper quadrant by checking the signs of the real and imaginary parts. The normal range for calculating phase angles by the subroutine is from +540 degrees to -540 degrees. This range is considered adequate for most application, but should these limits be exceeded the plot will jump with a vertical line to either +360 degrees or -360 degrees respectively before continuing with the plot.

For a graphic response the user specified frequency range is divided into 500 equally spaced points (on a logarithmic scale) for which to calculate the magnitude and phase. The user is given the option to select between 1 and 500 points for tabular data.

1. Bode Response

The graphic response section of the subroutine BODE creates two plots. The first plot is the response magnitude in decibels versus the frequency in radians per second. The magnitude from the subroutine FREQ is converted to decibels and plotted with its corresponding frequency. The second plot is of the phase in degrees versus frequency in radians

per second. The phase from the subroutine FREQ is plotted with its corresponding frequency. The frequency for both plots is plotted on a logarithmic scale. The scale range for both is set automatically to include the entire range of data to be plotted by the subroutine MINMAX which scans the data for the maximum and minimum values. System information (transfer function input number, output number, and (DC) gain) is listed in the graph legend.

The magnitude in decibels and the phase in degrees is searched for "0" decibel and 180 degree crossover points to calculate the phase and gain margins.

The tabular response section gives the user the option to select the the number of frequency points to be calculated between 1 and 500 for the frequency range he has specified. For tabular data response the user is given system information (numerator order, denominator order, and transfer function (DC) gain) for the selected transfer function input number and output number, and the frequency (radians per second), magnitude (decibels), phase (degrees), and the real and imaginary frequency response parts.

2. Code Response (Interactive Example)

The following example of compensator transfer function with a filter and regulator was obtained from [Ref. 8, pp. 382 - 384].

The compensated system is represented by:

$$[C] = \begin{bmatrix} -254.1 & -19.57 \end{bmatrix}$$

$$[D] = \begin{bmatrix} 0 \end{bmatrix}$$

$$[F] = \begin{bmatrix} 0. & 1.0 \\ 0. & -4.6 \end{bmatrix}$$

[G] = $\begin{bmatrix} 0. & \\ & 0.787 \end{bmatrix}$

[GAM] = $\begin{bmatrix} 0. & \\ & 0.1 \end{bmatrix}$

[H] = $\begin{bmatrix} 1. & 0. \end{bmatrix}$

[K] = $\begin{bmatrix} 95.4 & \\ & 4561. \end{bmatrix}$

[Q] = $\begin{bmatrix} 10. \end{bmatrix}$

a. Example of Graphic Response (Terminal Session)

BEGIN RECCRDING OF TERMINAL SESSION

R; T=0.01/0.02 18:30:23

graphics

EXECUTION BEGINS...

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

3

OPTGRAPH

THE COMPENSATOR TRANSFER SYSTEM OBTAINED FROM
OPTSYS CCNTAINS:

1. NUMBER OF STATES = 2
2. NUMBER OF CONTROLS (INPUTS) = 1
3. NUMBER OF MEASUREMENTS (OUTPUTS) = 1
4. MARKOV PARAMETER = 10.0×10^{-6}

CLEAR SCREEN TO CONTINUE

OPTGRAPH

AN OPEN LOOP POLE-ZERO, ROOT LOCUS, BODE, NYQUIST,
AND/OR NICHOLS PLOT IS DESIRED FOR:

INPUT # ?

1

OUTPUT # ?

1

OPTGRAPH

COMPENSATOR TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT

CPTGRAPH

ENTER OPTION NUMBER.

3

BODE PLOT
COMPENSATOR TRANSFER FUNCTION
DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTION NUMBER.

1

BODE PLOT
COMPENSATOR TRANSFER FUNCTION
PLOTTER SELECTION AND PAGE SIZE
(NOTE: PAGE IS 11.0 INCHES BY 8.5 INCHES WITH
SELECTION OF TEK618)

WHICH PLOTTER DO YOU DESIRE:

1. TEK618
2. VERSATEC

CHOOSE OPTION 1 OR 2

2

PAGE SIZE (MAXIMUM = 21.0 INCHES BY 21.0
INCHES)

HEIGHT =

5.65

WIDTH =

8.5

PLOTTING LIMITS FOR GRAPH

FREQUENCY RANGE - RADIANS PER SECOND

MINIMUM FREQUENCY = ?

0.1

MAXIMUM FREQUENCY = ?

1000000

PLOTTING LIMITS FOR GRAPH

FREQUENCY RANGE - RADIANS PER SECOND

MINIMUM FREQUENCY = 0.100E+00
MAXIMUM FREQUENCY = 0.100E+07
DO YOU DESIRE TO MAKE ANY CHANGES ?
TYPE "YES" OR "NO".

n

PLOT HEADING
HOW MANY LINES OF HEADING DO YOU DESIRE ?
(3 MAX)
ENTER NUMBER OF LINES.
(ENTER "0" FOR NO HEADING)

2

A MAXIMUM OF 32 CHARACTERS PER LINE IS
ALLOWED
LINE 1 IS:
filter simulation
LINE 2 IS:
ref. 8; pages 332 - 334

PLOT HEADING
LINE 1 : FILTER SIMULATION
LINE 2 : REF. 8; PAGES 332 - 334
DO YOU DESIRE TO MAKE ANY CHANGES ?
TYPE "YES" OR "NO".

n

A DISSPLA METAFILE HAS BEEN CREATED
CLEAR SCREEN TO CONTINUE

GAIN MARGIN AND PHASE MARGIN
PHASE CROSSOVER FREQUENCY IS 75.1184 RAD/SEC;
GAIN MARGIN IS -59.8906 DB
GAIN CROSSOVER FREQUENCY IS ***** RAD/SEC;
PHASE MARGIN IS -89.9480 DEG
CLEAR SCREEN TO CONTINUE

BODE PLOT
COMPENSATOR TRANSFER FUNCTION
DO YOU DESIRE TO MAKE ANY CHANGES TO:
1. PLOTTER / PAGE SIZE
2. GRAPH LIMITS (FREQUENCY RANGE)
3. HEADING
4. NO ADDITIONAL CHANGES - PLOT BODE PLOTS
5. NO CHANGES - EXIT BODE SUBROUTINE
ENTER OPTION NUMBER.
5

BODE PLOT
COMPENSATOR TRANSFER FUNCTION
DO YOU DESIRE:
1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE
ENTER OPTION NUMBER.
3

OPTGRAPH
COMPENSATOR TRANSFER FUNCTION
INPUT # = 1
OUTPUT # = 1
DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:
1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME

```
TRANSFER FUNCTION)
7.  ANOTHER TYPE TRANSFER FUNCTION /EXIT
    CPTGRAPH
ENTER OPTION NUMBER.
7
```

```
OPTGRAPH

DO YOU DESIRE TO ANALYZE:
1.  OPEN LOOP TRANSFER FUNCTION
2.  NOISE TRANSFER FUNCTION
3.  COMPENSATOR TRANSFER FUNCTION
4.  EXIT OPTGRAPH
ENTER OPTION NUMBER.
4
R; T=7.01/8.84 18:39:11
record off
END RECORDING OF TERMINAL SESSION
```

The preceding example follows at the end of the chapter as figure 3.3 and figure 3.4 .

b. Example of Tabular Data Listing File

The terminal session for the Bode tabular data response is similar to the Bode graphic response terminal session. Appendix D contains an example of tabular output sent to a disk (OPGRAPH LISTING) for five points between 10 and 100 radians/second.

3. Nyquist Response

The graphic response section of the subroutine NYQST creates a polar plot from the magnitude and phase data received from the subroutine FREQ. The plot is automatically scaled from a minimum radius of one by using the

subroutine MINMAX to determine the maximum phase and magnitude values. The gain / phase margin calculations and the tabular data output are identical to the subroutine BODE.

4. Nyquist Response (Interactive Example)

The following example is of the open loop transfer function for the filter simulation example described in the Bode interactive section.

This example follows at the end of the chapter as figure 3.5 .

5. Nichols Response

The graphic response section of the subroutine NICHCI creates a single rectangular plot of magnitude in decibels versus phase in degrees. The magnitude from the subroutine FREQ is converted to decibels and plotted with the corresponding magnitude. The plot is automatically scaled using the subroutine MINMAX to determine the maximum phase and magnitude values. The gain / phase margin calculations and the tabular data output are identical to the subroutine BODE.

6. Nichols Response (Interactive Example)

The following example is of the open loop transfer function for the filter simulation example described in the Bode interactive section.

This example follows at the end of the chapter as figure 3.6 .

AIRCRAFT WITH AN AUTOPILOT IN
THE LONGITUDINAL MODE

REF. 7; PAGE 315

OPEN LOOP TRANSFER POLE-ZERO MAP

INPUT # = 1
OUTPUT # = 1
DC GAIN = 1.000×10^0

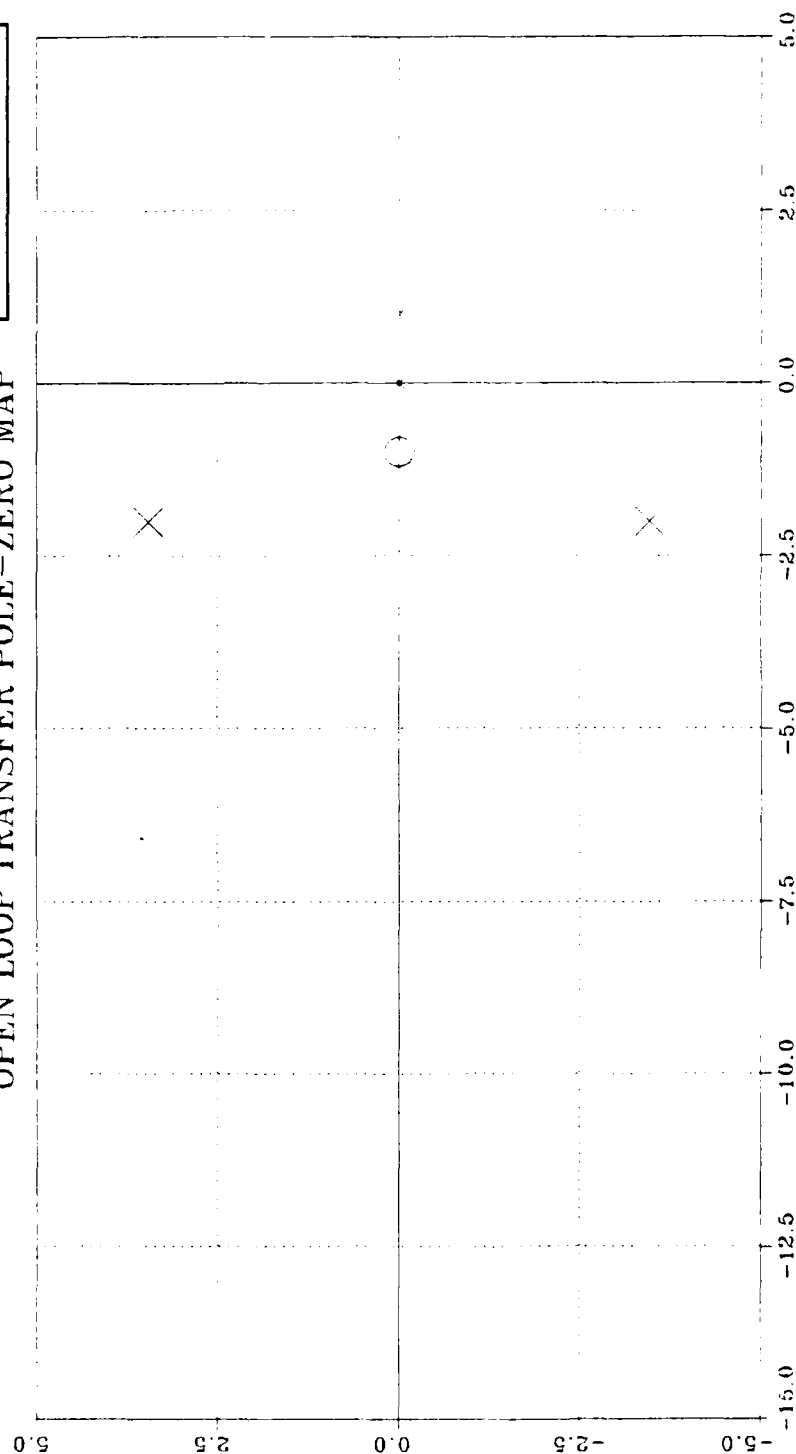


Figure 3.1 Pole-Zero Map Example.

AIRCRAFT WITH AN AUTO PILOT IN
THE LONGITUDINAL MODE
REF. 7; PAGE 315
ROOT-LOCUS PLOT (OPEN LOOP TF)

INPUT # = 1
OUTPUT # = 1
DC GAIN = 1.000×10^0

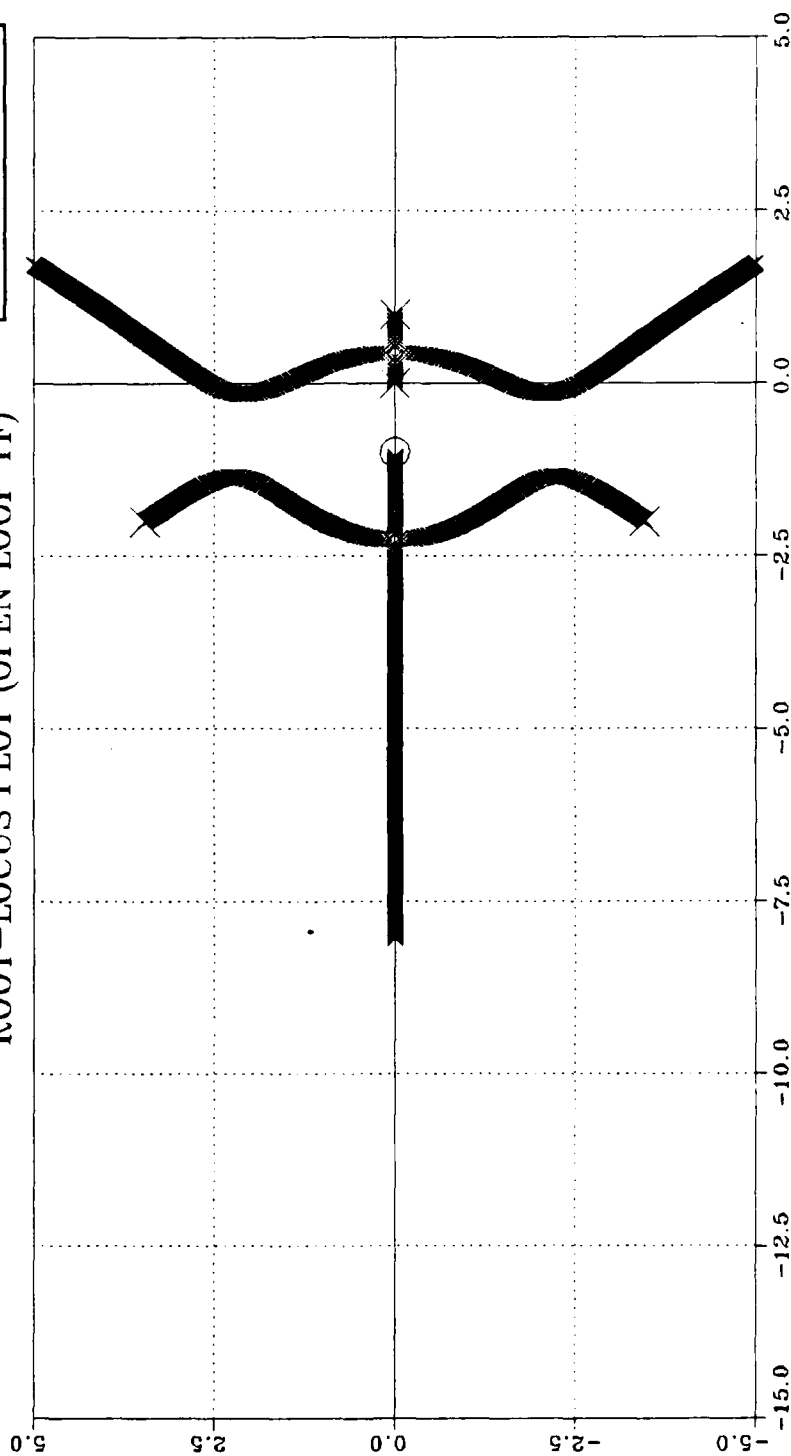


Figure 3.2 Root Locus Response Example.

FILTER SIMULATION
REF. 8; PAGES 332 - 334
COMPENSATOR TF BODE MAGNITUDE

INPUT # = 1
OUTPUT # = 1
DC GAIN = -1.135×10^0

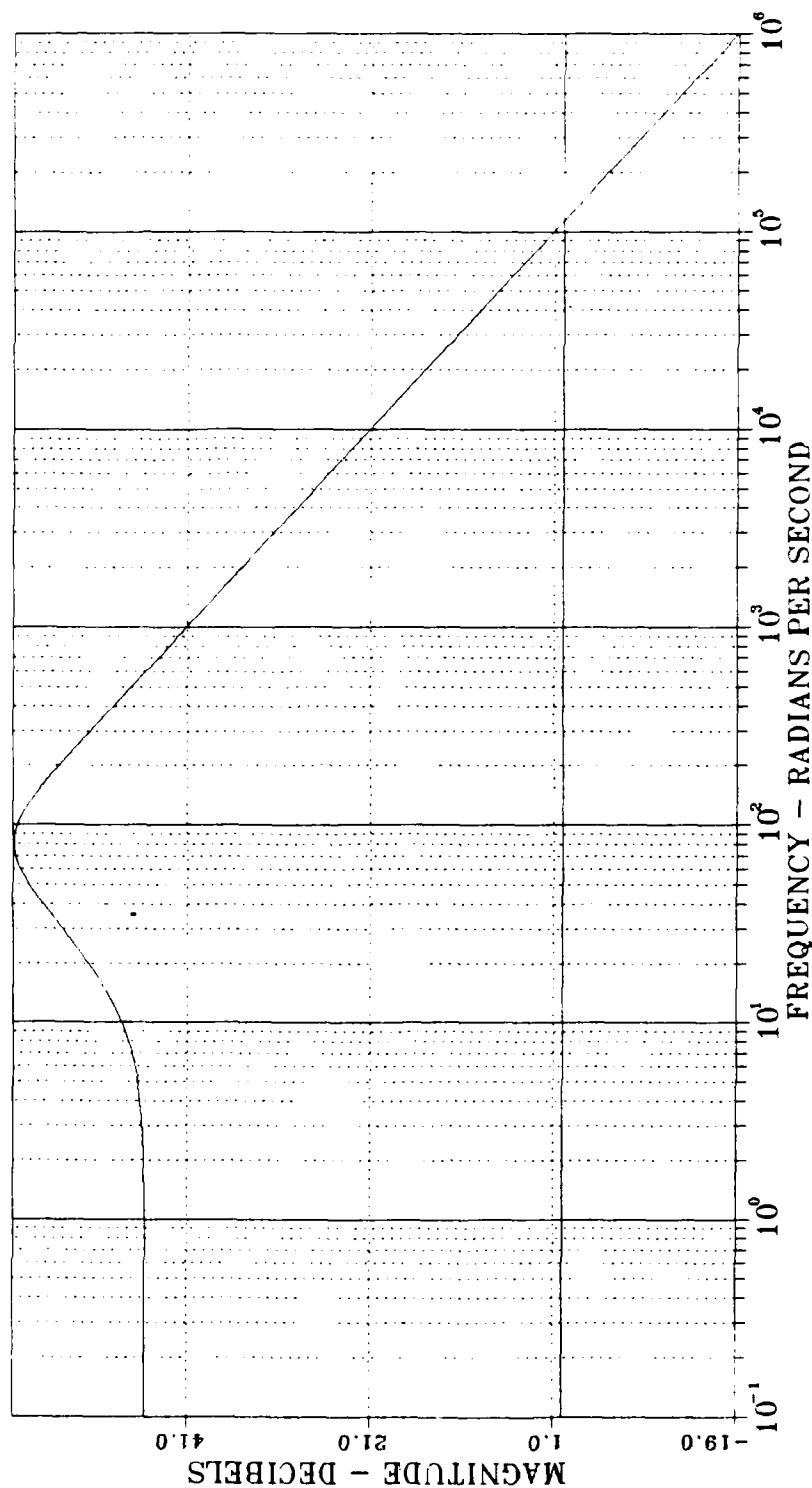


Figure 3.3 Bode Response Example (Magnitude).

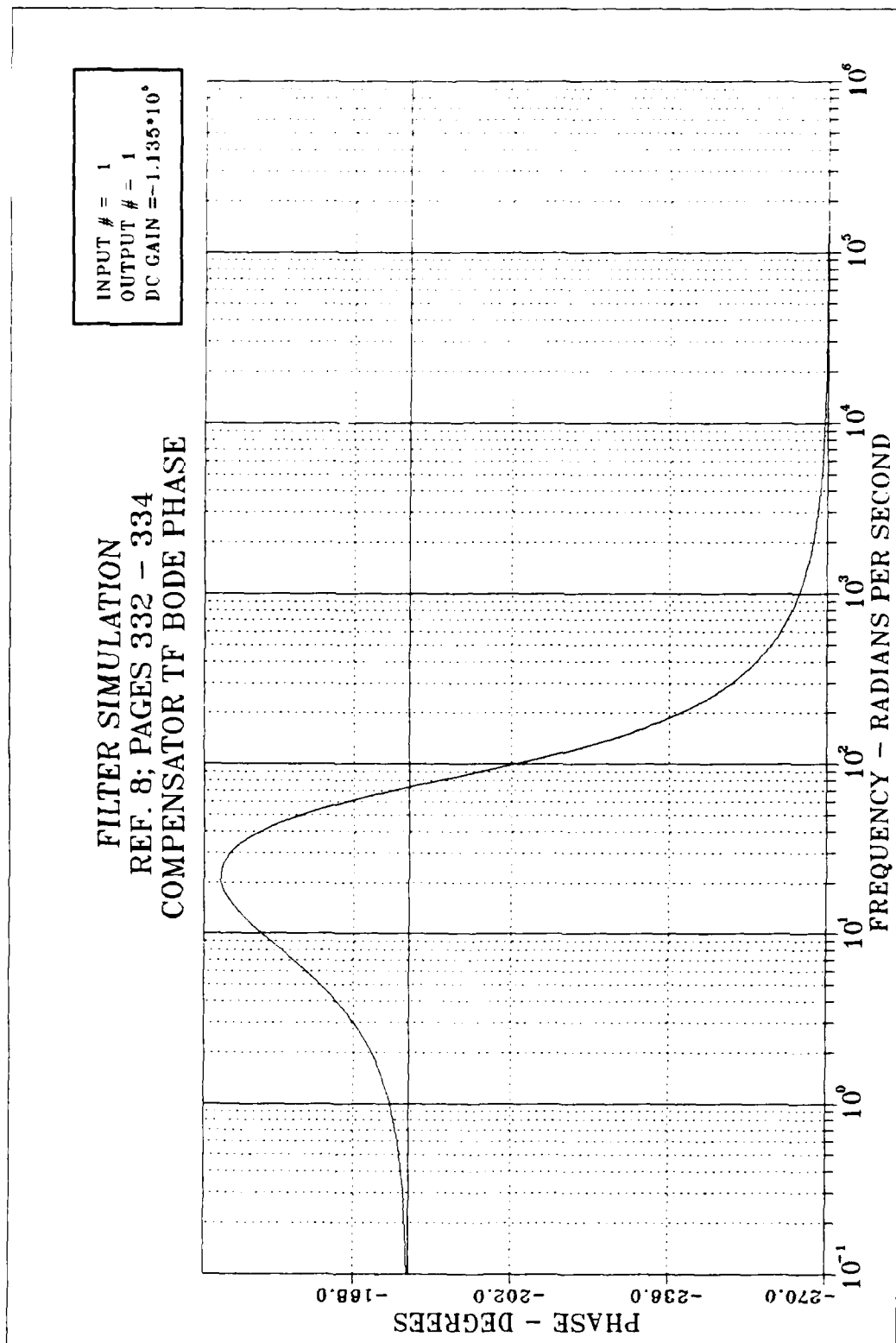


Figure 3.4 Bode Response Example (Phase).

FILTER SIMULATION
 REF. 8; PAGES 332 - 334
 OPEN LOOP TRANSFER NYQUIST

INPUT # = 1
 OUTPUT # = 1
 DC GAIN = 7.870×10^{-1}

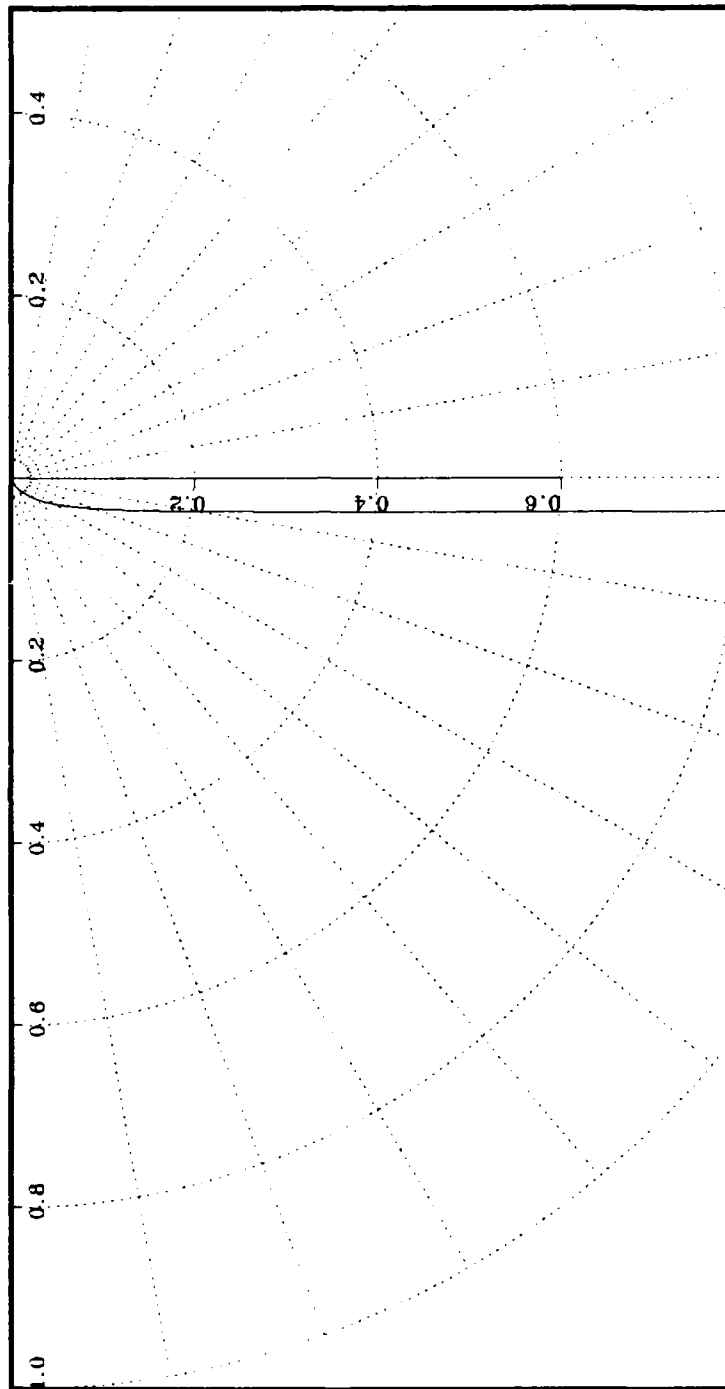


Figure 3.5 Nyquist Response Example.

FILTER SIMULATION
 REF. 8; PAGES 332 - 334
 OPEN LOOP TRANSFER NICHOLS

INPUT # = 1
 OUTPUT # = 1
 DC GAIN = $7.870 \cdot 10^{-1}$

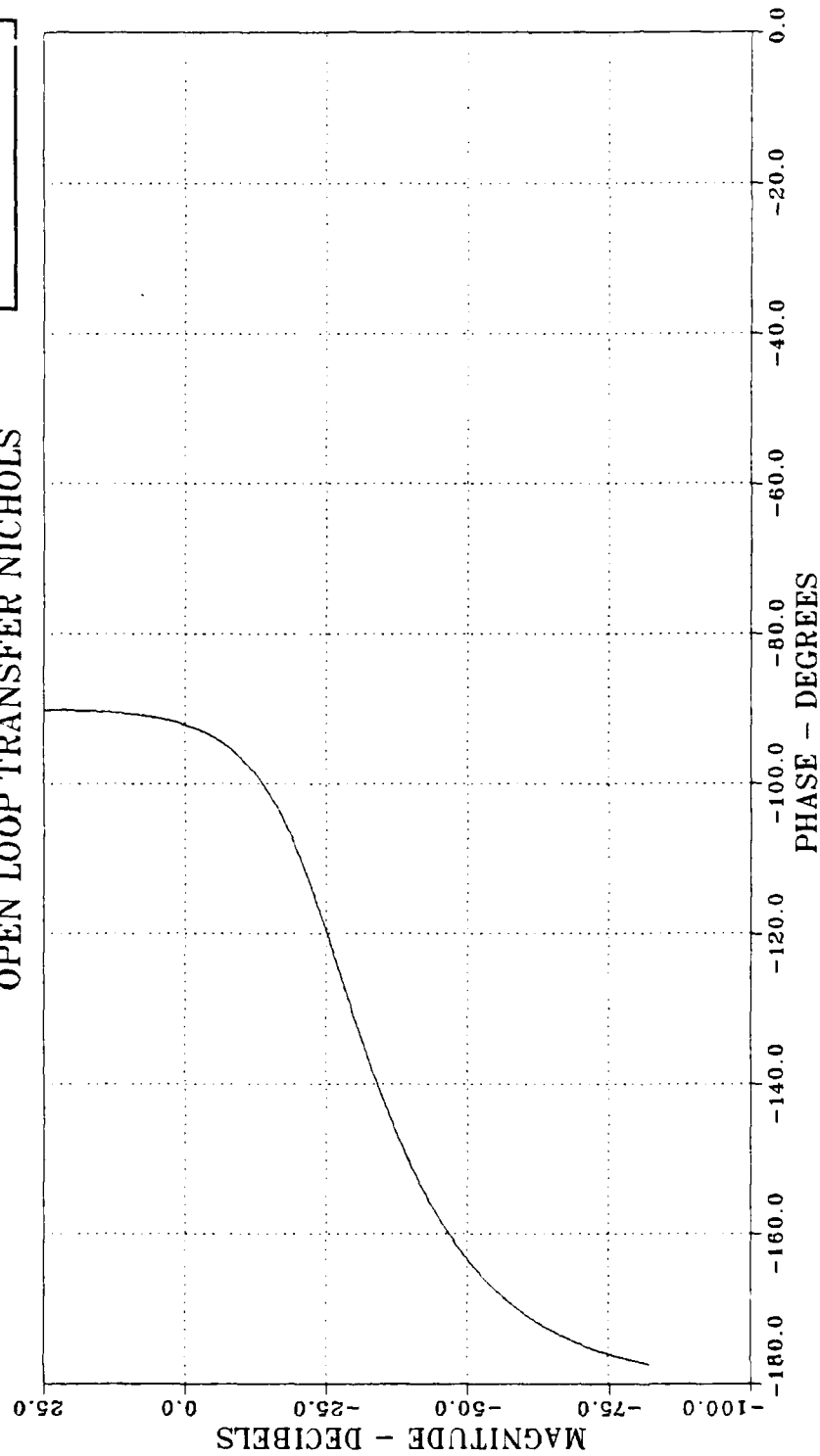


Figure 3.6 Nichols Response Example.

IV. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

An evaluation of the computational ability of OPTSYSX and OPTGRAPH, the program was tested using an 82 X 82 matrix, provided by NASA-Edwards, of longitudinal equations for the X-29A experimental forward-swept wing fighter aircraft prototype. The OPTSYSX program array had to be redimensioned and a 2-megabyte virtual machine size was required for a system of this size.

The open loop eigenvalues calculated by the OPTSYSX program compared favorably with the Eigensystem data supplied by NASA-Edwards, but the OPTSYSX program failed in calculating the open loop transfer function zeros. Extraneous zeros (calculated by OPTSYSX) were of the same approximate magnitude as the open loop system zeros. This prevented the identification and elimination of the extraneous zeros by comparison with the Markov parameter. The open loop zero identification was further complicated by not having available open loop transfer function zero information from NASA-Edwards to compare the OPTSYSX results to.

When requested for additional information NASA-Edwards provided a computer tape with a revised 98 X 98 matrix of longitudinal motion equations for the X-29A fighter aircraft. For this revised system, the OPTSYSX program requires a virtual machine size in excess of 2-megabyte. Time constraints have prevented running the OPTSYSX program with the revised system.

The OPTGRAPH program was tested using the OPTSYSX program's open loop Eigenvalue and transfer function output data for the X-29A 82 X 82 system of longitudinal motion

equations. OPTGRAPH provided excellent quality Bode plots for all of the open loop transfer function inputs and outputs and did not exhibit any computational difficulties as indicated by the lack of error messages at the end of the test. While the test results are encouraging as to the ability of the OPTGRAPH program to assist in the analysis of large order systems, the actual computational accuracy of the OPTGRAPH program can not be fully verified until accurate transfer function information can be obtained from the OPTSYSX program.

B. RECCMMENDATIONS

Based on the results of this Thesis the following areas emerged as possible areas for further research and study.

1. Root Finding Routines

The OPTSYSX program has two similar root finding subroutines (HQR and HQR2), to find the transfer function Eigensystem (poles) and zeros. The order of the transfer function numerator is calculated separately in the subroutine ZEROS.

The inclusion of the open loop transfer function eigensystem (pole locations), zero locations, transfer function numerator orders, and transfer function gains, for the revised (98 X 98) longitudinal mode system by NASA-Edwards should simplify the locating the problem in calculating the transfer function zeros.

The open loop system data from NASA-Edwards also included Bode magnitude and phase plots, and gain and phase margin information, which can be used to evaluate more conclusively the OPTGRAPH program's ability to help evaluate a large order system.

2. Program Memory Requirements

The OPTSYSX program requires in excess of 2-megabyte of virtual machine memory when configured for large matrix operations (98 X 98). Virtual machines with this memory capacity are not normally available to a user. The memory usage for matrix storage and manipulation is a possible area for the reduction in program memory requirement size. The OPTSYSX program retains most of its computational arrays in memory. This method simplifies programming but is extremely inefficient in its use of virtual machine memory.

3. Further Modifications

The area of modern digital controls should be further investigated. The computational abilities of the OPTGRAPH program make it readily adaptable, with minor modifications, for assisting in the analysis of transfer functions in the digital domains.

APPENDIX A
OPTSYS EXEC LISTING

```
      &TRACE OFF
*****
*
*           THE OPTSYS EXEC
*
*       CONTROLS THE OPTSYSX, OPTCALC AND OPTPLOT
*
*       TO DETERMINE THE TIME RESPONSE OF A SYSTEM.
*
*           BY H. A. DIEL
*
*       VERSION 1.0   16 JULY 1984
*
*       MODIFIED BY M. H. LAPTAS
*****
CLRSCRN
&BEGTYPE -ENDFOUR

*****

      CUIPUT FROM OPTSYSX WILL COME TO THE SCREEN IF YOU
      WISH LISK FILE
      ENTER  D      ANY OTHER INPUT YIELD SCREEN

*****

-ENDFOUR
&READ VARS &ANS
&IF .&ANS NE .D &GOTO -ONE
FILEDEF 06 DISK OUTPUTX LISTING A1
CLRSCRN
&BEGTYPE -ENDSIX
```

OUTPUT WILL GO TO DISK FILE 'OUTPUTX LISTING A1'

-ENDSIX

CP SLEEP 3 SEC

-ONE

CLRSCRN

CP LINK 0039P 191 251 RR

ACC 251 F

*

* ALLOW THE USE OF AN OLD "OPTMAT DATA A1"

*

RENAME OPTMAT DATA A1 OPTSYS DATA A1

EIF &RC NE 0 &GOTO -FIRST

RENAME OPTSYS DATA A1 OPTMAT DATA A1

FINDSTAK OPTMAT DATA A1 C01 0 LIM002 ALL GROUP1

EREAD VARS &A1 &A2

EIF .&A2 NE .1 &GOTO -START

CLRSCRN

EBEGTYPE -ENDTWO

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 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!

-ENDTWO

&READ VARS &ANS

&IF .&ANS EQ .Y &GOTO -START

-FIRST

* *

* ERASE THE OLD "CPTMAT 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

-START

CLRSCFN

&BEGTYPE -ENDONE

THE OPTSYS EXEC CONTROLS FOUR PROGRAMS:

- 1 OPTSYSX FCRTRAN (SYSTEM ANALYSIS)
GENERATES OPTMAT AND OPTGRAPH DATA SETS
- RERUN USES OPTMAT
- 2 OPTCALC FCRTRAN (CALCULATE TIME RESPONSE)
GENERATES OPTPLOT DATA SET USES OPTMAT FROM
OPTSYSX
- 3 OPTGRAPH FORTRAN (POLE-ZERO, ROOT-LOCUS, ECDE,

NYQUIST, ETC)

USES OPTGROL -OPTGRNO -OPTGRCM FROM OPTSYSX

4 EXIT

OPTPLOT FORTRAN IS THE FOURTH PROGRAM

USES OPTPLOT DATA SET FROM OPTCALC

THE SIZE OF THE DATA SETS VARY WITH THE SYSTEM
ORDER, AND CAN USE ABOUT 20% OF THE USERS DISK SPACE.
THEREFORE ENSURE THAT SUFFICIENT DISK SPACE IS
AVAILABLE.

IF DATA SET IS ALREADY AVAILABLE YOU MAY RUN ANY
PROGRAM

- ENTER 1 , 2 , 3 , 4 ANY OTHER INPUT
RETURNS TO MENU

-ENDONE

&READ VAES &ANS

&IF .&ANS EQ .1 &GOTO -OPTSYS

&IF .&ANS EQ .2 &GOTO -OPTCALC1

&IF .&ANS EQ .3 &GOTO -OPGRAPH1

&IF .&ANS EQ .4 &EXIT &RC

&GOTO -START

-OPTSYS

FILEDEF 8 DISK OPTPLCT DATA A1 (PERM

FILEDEF 9 DISK OPTMAT DATA A1 (PERM

FILEDEF 10 DISK OPTGRCL DATA A1 (PERM

FILEDEF 01 DISK OPTGRNO DATA A1 (PERM

FILEDEF 04 DISK OPTGRCM DATA A1 (PERM

GLOBAL TXTLIB VFORTLIE IMSLDP NONIMSL

&TYPE LCADING OPTSYS..... GENERAL STATE VARIABLE ANALYSIS

PROGRAM

&TYPE..... OPTMAT DATA A1 INPUT DATA SET ON RERUNS

OPTSYSX

CLRSCN

&TYPE... DATA SETS .. OPTMAT AND .. OPGRAPH CREATED

CP SLEEP 5 SEC

```

&GOTO -START
-OPTCALC1
*****
*
*          CHECK FOR DATA IN THE FILE "OPTMAT DATA "
*          BEFORE LOADING OPTCALC
*
*****
FINDSTAK OPTMAT DATA A1 C01 0 LIM002 ALL GROUP1
&READ VARS &A1 &A2
&IF .&A2 EQ .1 &GOTO -OPTCALC
&TYPE PROPER DATA FILE IS NOT AVAILABLE FOR OPTCALC
CP SLEEP 5 SEC
&GOTO -START
-OPTCALC
&TYPE.....OPTCALC.... TIME RESPONSE PROGRAM
FILEDEF 8 DISK OPTPLCI DATA A1 (PERM
FILEDEF 9 DISK OPTMAT DATA A1 (PERM
GLOBAL TXTLIB VFORTLIE IMSLDP NONIMSL
OPTCALC
&TYPE    .. OPTPLOT DATA A1    CREATED
CP SLEEP 5 SEC
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 &GOTO -START
&TYPE PLOTS NEXT
CP SLEEP 4 SEC
GLOBAL TXTLIB DISLIBVS 92DISLIB 92INTLIB VPORTLIB GRFLIB
      NCNIMSL CMSLIB
FILEDEF 8 DISK OPTPLOT DATA A1 (PERM
FILEDEF 11 DISK LBLANK TMP A3 (RECFM F LRECL 2400 BLKSIZE
      2400 BLKSIZE 2400 XTENT 600
FILEDEF 12 DISK ISSCCMAP MAPDTA * (RECFM F LRECL 400
      BLKSIZE 400
FILEDEF 13 DISK TABLET TMP A3 (RECFM VS LRECL 208 BLKSIZE
      208 XTENT 1000
FILEDEF 14 DISK LBLANK MAPDTA &LBMODE (RECFM VS LRECL 608
      BLKSIZE 608
FILEDEF 17 DISK DISSPIA SYSUT1 A3 (RECFM FB LRECL 2000
      BLCK 2000 XTENT 500
FILEDEF 18 DISK DISSPIA METAFILE A4 (RECFM VBS LRECL 19065
      BLCK 19069
OPTPLOT
&GOTO -META
-OPGRAPH1
*****
*
*          CHECK FOR DATA IN THE OPTGRAPH DATA SETS "
*          BEFORE LOADING OPGRAPH
*
*****
*****
RENAME OPTGROL DATA A1 OPTSYS DATA A1
&IF &RC NE 0 &GOTO -CFTNOISE
RENAME OPTSYS DATA A1 OPTGROL DATA A1

```

```

FINDSTAK OPTGRNO DATA A1 C01 0 LIM002 ALL GROUP1
&READ VARS &A1 &A2 &A3 &A4 &A5
&IF .&A2 NE .0 &GOTO -SIXTEEN
-OPTNCISE
RENAME OPTGRNO DATA A1 OPTSYS DATA A1
&IF &RC NE 0 &GOTO -CPTCOMP
RENAME OPTSYS DATA A1 OPTGRNO DATA A1
FINDSTAK OPTGRNO DATA A1 C01 0 LIM002 ALL GROUP1
&READ VARS &A1 &A2 &A3 &A4 &A5
&IF .&A2 NE .0 &GOTO -SIXTEEN
-OPTCCMP
RENAME OPTGRCM DATA A1 OPTSYS DATA A1
&IF &RC NE 0 &GOTO -NCDATA
RENAME OPTSYS DATA A1 OPTGRCM DATA A1
FINDSTAK OPTGRCM DATA A1 C01 0 LIM002 ALL GROUP1
&READ VARS &A1 &A2 &A3 &A4 &A5
&IF .&A2 NE .0 &GOTO -SIXTEEN
-NODATA
&PRINT PROPER DATA FILE IS NOT AVAILABLE FOR OPGRAPH
CP SLEEP 5 SEC
&GOTO -START
-SIXTEEN
&TYPE ...OPTGRAPH.... CLASSICAL ANALYSIS OF OPTSYS OUTPUT
SET IDRTLS 10
GLOBAL TXTLIB DISLIBVS 92DISLIB 92INTLIB VFORTLIB GRFLIB
      NONIMSL CMSLIB
SET IDRTBLS 10
FILEDEF 01 DISK OPTGRNO DATA A1 (PERM
FILEDEF 02 DISK OPGRAPH LISTING A1
FILEDEF 03 PRINTER (RECFM FA BLKSIZE 133 PERM
FILEDEF 04 DISK OPTGRCM DATA A1 (PERM
FILEDEF 06 TERM (RECFM FA BLKSIZE 133
FILEDEF 05 TERM (RECFM FA BLKSIZE 80
FILEDEF 10 DISK OPTGRCL DATA A1 (PERM
FILEDEF 11 DISK LBLANK TMP A3 (RECFM F LRECL 2400 BLKSIZE

```

```

2400 XTENT 600
FILEDEF 12 DISK ISSCCMAP MAPDTA * (RECFM F LRECL 400
    BLKSIZE 400
FILEDEF 13 DISK TABLET TMP A3 (RECFM VS LRECL 208 BLKSIZE
    208 XTENT 1000
FILEDEF 14 DISK LBLANK MAPDTA &LBMODE (RECFM VS LRECL 608
    BLKSIZE 608
FILEDEF 17 DISK DISSPLA SYSUT1 A3 (RECFM FB LRECL 2000
    BLCK 2000 XTENT 500
FILEDEF 18 DISK DISSPLA METAFILE A4 (RECFM VBS LRECL 19065
    BLCK 19069
CPTGRAPH
-META
*****
*
*          CHECK FOR FILE "DISSPLA METAFILE A4" ON
*          THE USER'S DISK BEFORE GOING TO DISSPOP
*
*****
RENAME DISSPLA METAFILE A4 OPTSYS METAFILE A4
&IF &RETCode NE 0 &GOTO -START
RENAME OPTSYS METAFILE A4 DISSPLA METAFILE A4
-EIGHT
CLRSCRN
&BEGTYPE -ENDTEN

```

DC 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 -START
&GOTO -EIGHT
-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
&GOTO -START

APPENDIX B
OPISYSX PROGRAM LISTING

This portion of the thesis contains the OPTSYSX FORTRAN program (88 pages).


```

C-----DATA IY/Y',IZ/N'/-----OPT00490
C-----SUPPRESS INDIVIDUAL UNDERFLOW, OVERFLOW, DIVICE CHECK, AND DECIMAL-----OPT00500
C-----CONVERT ERROR MESSAGES; PROVIDE SUMMARY OF ERRORS ONLY.-----OPT00510
C-----OPT00520
C-----OPT00530
C-----OPT00540
C-----CALL ERRSET (207,256,-1,1,1,209)-----OPT00550
C-----CALL ERRSET (215,256,-1,1,1)-----OPT00560
C-----OPT00570
C-----INITIALIZE SAVE FLAGS.-----OPT00580
C-----=-----OPT00590
C-----OPT00600
C-----OPT00610
C-----OPT00620
C-----OPT00630
C-----OPT00640
C-----OPT00650
C-----OPT00660
C-----OPT00670
C-----SCRNA-----OPT00680
C-----CALL FRICMS ('CLRSCRN ')-----OPT00690
C-----WRITE (5,8E)-----OPT00700
C-----CALL RDINT (IANS)-----OPT00710
C-----IF (IANS.EC.1) GO TO 20-----OPT00720
C-----IF (IANS.EQ.2) GO TO 10-----OPT00730
C-----GO TO 5-----OPT00740
C-----SCRNI-----OPT00750
C-----CALL FRICMS ('CLRSCRN ')-----OPT00760
C-----WRITE (5,8EC)-----OPT00770
C-----CALL RDCHAR (IANS)-----OPT00780
C-----IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 30-----OPT00790
C-----GO TO 40-----OPT00800
C-----WRITE (5,8EC)-----OPT00810
C-----GO TO 20-----OPT00820
C-----CONTINUE-----OPT00830
C-----IF (IANS.EC.IZ) GO TO 560-----SCRN2-----OPT00840
C-----CALL FRICMS ('CLRSCRN ')-----OPT00850
C-----WRITE (5,9C)-----OPT00860
C-----CALL RDCHAR (IANS)-----OPT00870
C-----IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60-----OPT00880
C-----GO TO 70-----OPT00890
C-----WRITE (5,8EC)-----OPT00900
C-----GO TO 50-----OPT00910
C-----CONTINUE-----OPT00920
C-----IF (IANS.EC.IZ) GO TO 560-----ISET-----OPT00930
C-----CALL FRICMS ('CLRSCRN ')-----OPT00940
C-----WRITE (5,91C)-----OPT00950
C-----OPT00960

```

5C	CALL RCCHAR (IANS)	OPT00970
	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 90	OPT00980
	GO TO 100	OPT00990
	WRITE (5,88C)	OPT01000
	GO TO 80	OPT01010
100	CONTINUE	OPT01020
C	IF (IANS.EC.IY) ISET=1	OPT01030
C	INITIALIZE SYSTEM FLAGS.	OPT01040
C		OPT01050
1C	CONTINUE	OPT01060
	ISET=0	OPT01070
	IQL=0	OPT01080
	IQ=0	OPT01090
	IR=0	OPT01100
	ISS=0	OPT01110
	IM=0	OPT01120
	ITF1=0	OPT01130
	ITF2=0	OPT01140
	ITF3=0	OPT01150
	IFDFW=0	OPT01160
	IF=0	OPT01170
	IDSTAB=0	OPT01180
	IDBUG=0	OPT01190
	IPSD=0	OPT01200
	IYU=0	OPT01210
	INCRM=0	OPT01220
	IREG=0	OPT01230
	NS=0	OPT01240
	NC=0	OPT01250
	NOR=0	OPT01260
	NC=0	OPT01270
	IRDMAT=0	OPT01280
C		OPT01290
	CALL FRTCMS ('CLRSCRN')	OPT01300
	WRITE (5,57C)	OPT01310
	CALL RDINT (IANS)	OPT01320
	ICL=IANS-1	OPT01330
	IF (IOL.EQ.2) GO TO 350	OPT01340
	IF (IOL.EQ.3) GO TO 200	OPT01350
C		OPT01360
	CALL FRTCMS ('CLRSCRN')	OPT01370
110	WRITE (5,58C)	OPT01380
	CALL RCCHAR (IANS)	OPT01390
	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 120	OPT01400
	GO TO 130	OPT01410
120	WRITE (5,88C)	OPT01420
	GO TO 110	OPT01430
		OPT01440

250	GO TO 260	OPT01930
	WRITE (5,88C)	OPT01940
260	GO TO 240	OPT01950
	CONTINUE	OPT01960
	IF ((IANS.EC.IY) IDSTAB=1	OPT01970
	IF ((IANS.EC.IZ) IDSTAB=0	OPT01980
C	-----	OPT01990
270	WRITE (5,66C)	OPT02000
	CALL RCHAR (IANS)	OPT02010
	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 280	OPT02020
	GO TO 290	OPT02030
280	WRITE (5,88C)	OPT02040
	GO TO 270	OPT02050
290	CONTINUE	OPT02060
	IF ((IANS.EC.IY) IDEBUG=1	OPT02070
	IF ((IANS.EC.IZ) IDEBUG=0	OPT02080
300	CONTINUE	OPT02090
C	-----	OPT02100
	CALL FRCMS ('CLRSCRN ')	OPT02110
320	WRITE (5,71C)	OPT02120
	CALL RCHAR (IANS)	OPT02130
	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 320	OPT02140
	GO TO 340	OPT02150
330	WRITE (5,88C)	OPT02160
	GO TO 320	OPT02170
340	CONTINUE	OPT02180
	IF ((IANS.EC.IY) IREG=1	OPT02190
	IF ((IANS.EC.IZ) IREG=0	OPT02200
350	CALL RDMATF(NS,NC,NCB,NG,ISAF,ISAG,ISAH,IGAM,ISAB,IRDMAT)	OPT02210
	IF ((ISAF.EC.1).AND.(IRDMAT.EC.1)) GO TO 352	OPT02220
C	-----	OPT02230
	CALL FRCMS ('CLRSCRN ')	OPT02240
	WRITE (5,72C)	OPT02250
	CALL RCREAL (ANSR)	OPT02260
	NS=IDINT (ANSR)	OPT02270
352	IF ((IOL.EQ.2)) GO TO 360	OPT02280
	IF ((ISAG.EC.1).AND.(IPDMAT.EC.1)) GO TO 354	OPT02290
C	-----	OPT02300
	WRITE (5,73C)	OPT02310
	CALL RCREAL (ANSR)	OPT02320
	NC=IDINT (ANSR)	OPT02330
354	IF ((ISAH.EC.1).AND.(IRDMAT.EC.1)) GO TO 356	OPT02340
C	-----	OPT02350
	WRITE (5,740)	OPT02360
	CALL RCREAL (ANSR)	OPT02370
	NOB=IDINT (ANSR)	OPT02380
356	IF ((IGAM.EC.1).AND.(IRDMAT.EC.1)) GO TO 360	OPT02390
C	-----	OPT02400

```

WRITE (5,75C)
CALL RDRREAL (ANSR)
NG=IDINT (ANSR)
CCNT INUE
IF (IOL.EQ.2) GO TO 364
IF (IOL.EQ.3) GO TC 310
C-----IPSD-----
IF (IREG.NE.C.OR.NC.EQ.0) GO TO 310
CALL FRTCMS ('CLRSCRN ')
WRITE (5,65C)
CALL RDINT (IANS)
IPSD=IANS
IF (IPSD.EQ.3) IPSD=0
IF (IPSD.EQ.0) GO TC 310
C-----IYU-----
CALL FRTCMS ('CLRSCRN ')
WRITE (5,77C)
CALL RDINT (IANS)
IYU=IANS-1
C-----INCRM-----
CALL FRTCMS ('CLRSCRN ')
WRITE (5,82C)
CALL RDRREAL (ANSR)
INORM=IDINT (ANSR)
CONTINUE
310 C-----ITF1-----
CALL FRTCMS ('CLRSCRN ')
WRITE (5,62C)
CALL RDINT (IANS)
ITF1=IANS-1
IF (IOL.EQ.3) GO TC 362
C-----ITF2-----
IF (IREG.NE.1) GO TC 315
CALL FRTCMS ('CLRSCRN ')
WRITE (5,63C)
CALL RDINT (IANS)
ITF2=IANS-1
C-----ITF3-----
IF (IREG.NE.0.OR.(NC.EQ.0.CR.NG.EQ.0)) GO TC 362
CALL FRTCMS ('CLRSCRN ')
WRITE (5,64C)
CALL RDINT (IANS)
ITF3=IANS-1
C-----IF-----
362 CCNT INUE
IF ((ITF1+ITF2+ITF3).EQ.0) GO TO 364
CALL FRTCMS ('CLRSCRN ')
WRITE (5,66C)

```

OPT02410
 OPT02420
 OPT02430
 OPT02440
 OPT02450
 OPT02460
 OPT02470
 OPT02480
 OPT02490
 OPT02500
 OPT02510
 OPT02520
 OPT02530
 OPT02540
 OPT02550
 OPT02560
 OPT02570
 OPT02580
 OPT02590
 OPT02600
 OPT02610
 OPT02620
 OPT02630
 OPT02640
 OPT02650
 OPT02660
 OPT02670
 OPT02680
 OPT02690
 OPT02700
 OPT02710
 OPT02720
 OPT02730
 OPT02740
 OPT02750
 OPT02760
 OPT02770
 OPT02780
 OPT02790
 OPT02800
 OPT02810
 OPT02820
 OPT02830
 OPT02840
 OPT02850
 OPT02860
 OPT02870
 OPT02880

```

CALL RDREAL (ANSR)
IE=IDINT(ANSR)
C-----FLAG SETTINGS-----
364 CONTINUE
CALL FRTCMS ('CLRSCRN ')
WRITE (6,76C)
WRITE (6,77C)
WRITE (6,78C) IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDEBUG,ISET
1,IDSTAB
WRITE (6,79C)
WRITE (6,80C) IPSD,IYU,INORM,IREG,NS,NC,NOB,NG
WRITE (6,81C) NS,NC,NOB,NG
C-----CPTGRAPH DATA FILES -----
ZERO = 0
CNE = 1
REWIND 10
ITFX = 1
WRITE (10,1000) ZERO,CNE,NS,NC,NOB,IE,ITFX
REWIND 1
ITFX = 2
WRITE (1,1000) ZERO,CNE,NS,NG,NOB,IE,ITFX
REWIND 4
ITFX = 3
WRITE (4,1000) ZERO,CNE,NS,NOB,NC,IE,ITFX
C-----BEGIN CALCULATIONS-----
N2=2*NS
CALL INNER (NS,NC,NOB,NG,N2,ACL,B,BA,CI,CR,CC,CWI,CWR,D,FBGC,FBGE,
1G,GAM,GM,GN,HC,DI,D2,PRO,RM,RC,Q,SC,WR,WI,W21,X,WNORM,WNORMI,D
2ESTAB,AA,BM,CM,JCF,RES,AY,8B,CC,CP,GM,GV,HY,HU,DSICRE,ISAF,ISAH,IS
3AG,IGAM,IRET,PRIT,NRCW,NCOL,IRDMAT,ISAA,ISAB)
C-----CPTGROL DATA-----
IF((ITF1.EQ.1).OR.(ITF1.EQ.2)) GO TO 396
END FILE 10
REWIND 10
ITFX = 1
WRITE (10,1000) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
C-----CPTGRNO DATA-----
396 IF((ITF2.EQ.1).OR.(ITF2.EQ.2)) GO TO 397
END FILE 1
REWIND 1
ITFX = 2
WRITE (1,1000) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
C-----CPTGRCM DATA-----
397 IF((ITF3.EQ.3).OR.(ITF3.EQ.2)) GO TO 398
END FILE 4
REWIND 4
ITFX = 3
WRITE (4,1000) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
OPT02890
OPT02900
OPT02910
OPT02920
OPT02930
OPT02940
OPT02950
OPT02960
OPT02970
OPT02980
OPT02990
OPT03000
OPT03010
OPT03020
OPT03030
OPT03040
OPT03050
OPT03060
OPT03070
OPT03080
OPT03090
OPT03100
OPT03110
OPT03120
OPT03130
OPT03140
OPT03150
OPT03160
OPT03170
OPT03180
OPT03190
OPT03200
OPT03210
OPT03220
OPT03230
OPT03240
OPT03250
OPT03260
OPT03270
OPT03280
OPT03290
OPT03300
OPT03310
OPT03320
OPT03330
OPT03340
OPT03350
OPT03360

```

```

C-----OPT03370
358 IF(IRET.EQ.1) GO TO 370
CALL WRTMAT(BA,G,HO,CAM,FBGC,FBGE,AY,B,NS,NC,NOB,NG)
C-----OPT03380
OPT03390
OPT03400
370 WRITE (5,82C)
CALL RDCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 380
GO TO 390
380 WRITE (5,88C)
GO TO 370
390 CCNTINUE
C-----RESET OPGRAPH DATA FILE
IF(IANS.EQ.IZ) GO TO 395
END FILE 10
REWIND 10
END FILE 1
REWIND 1
END FILE 4
REWIND 4
C-----OPT03410
OPT03420
OPT03430
OPT03440
OPT03450
OPT03460
OPT03470
C-----OPT03480
OPT03490
OPT03500
OPT03510
OPT03520
OPT03530
OPT03540
OPT03550
C-----OPT03560
OPT03570
OPT03580
C-----ISAF
OPT03590
C-----OPT03600
OPT03610
OPT03620
OPT03630
OPT03640
OPT03650
OPT03660
OPT03670
OPT03680
OPT03690
OPT03700
OPT03710
C-----ISAH
OPT03720
OPT03730
OPT03740
OPT03750
OPT03760
OPT03770
OPT03780
OPT03790
OPT03800
OPT03810
OPT03820
OPT03830
OPT03840

```

470	CCNT INUE	ISAG	OPT03850
	IF (NG.EQ.C) GO TO 510		OPT03860
480	CALL FRTCMS ('CLRSCRN ')		OPT03870
	WRITE (5,88C)		OPT03880
	CALL RDCHAR (IANS)		OPT03890
490	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 490		OPT03900
	GO TO 500		OPT03910
	WRITE (5,88C)		OPT03920
500	GO TO 480		OPT03930
	CCNT INUE		OPT03940
	IF (IANS.EC.IY) ISAG=1		OPT03950
	IF (IANS.EC.IZ) ISAG=0		OPT03960
510	CONT INUE	IGAM	OPT03970
	IF (NG.EQ.C) GO TO 550		OPT03980
	CALL FRTCMS ('CLRSCRN ')		OPT03990
520	WRITE (5,87C)		OPT04000
	CALL RDCHAR (IANS)		OPT04010
	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 530		OPT04020
530	GO TO 540		OPT04030
	WRITE (5,88C)		OPT04040
540	GO TO 520		OPT04050
	CONT INUE		OPT04060
	IF (IANS.EC.IY) IGAM=1		OPT04070
	IF (IANS.EC.IZ) IGAM=0		OPT04080
550	CONT INUE	ISAA	OPT04090
	CALL FRTCMS ('CLRSCRN ')		OPT04100
551	WRITE (5,872)		OPT04110
	CALL RDCHAR (IANS)		OPT04120
	IF ((IANS.EC.IY).OR.(IANS.EQ.IZ)) GO TO 553		OPT04130
	WRITE (5,88C)		OPT04140
553	GO TO 551		OPT04150
	CONT INUE		OPT04160
	IF (IANS.EC.IY) ISAA=1		OPT04170
	IF (IANS.EC.IZ) ISAA=0		OPT04180
	CONT INUE	ISAB	OPT04190
555	CALL FRTCMS ('CLRSCRN ')		OPT04200
	WRITE (5,874)		OPT04210
	CALL RDCHAR (IANS)		OPT04220
	IF ((IANS.EC.IY).OR.(IANS.EQ.IZ)) GO TO 557		OPT04230
	WRITE (5,88C)		OPT04240
557	GO TO 555		OPT04250
	CONT INUE		OPT04260
	IF (IANS.EC.IY) ISAB=1		OPT04270
	IF (IANS.EC.IZ) ISAB=0		OPT04280
	GO TO 10		OPT04290
			OPT04300
			OPT04310
			OPT04320

```

C-----WRITE (5,92C)-----TERMINATE-----OPT04320
560 STOPOPT04340
C-----OPT04350
885 FORMAT (5X,22HIN CASE OF PROBLEMS CALL EXT 2826,///,10X,42H1 DEFIOPT04370
1PNET MATRIX EQUATIONS USED IN OPTSYS,///,15X,22H2 ALTERNATE DATA INOPT04380
2PUT METHODS,///,10X,26H2 START INTERACTIVE INPUT,///,15X,13HENTER OPT04390
32,///,5X,59H NORMALLY TWO SUCCESSIVE NULL ENTERS TERMINATES TOPT04400
4HE PROGRAM.OPT04410
890 FORMAT (5X,55HPROGRAM. IT WILL SOLVE NUMEROUS CONTROL PROBLEMS OPT04420
1CONTROL,///,8X,45HECLLCHING TYPES OF SYSTEMS CONTROL EQUATION: OPT04430
2N THE,///,8X,45HECLLCHING TYPES OF SYSTEMS CONTROL EQUATION: OPT04440
335HXDOT = F8#X + G8#U + GAM8*(W+W0),///,20X,22HMEASUREMENT EQUATION: OPT04450
4TION--///,15X,21H2 = H8#X + D8#U + V,///,20X,22HREGULATOR PERFORMOPT04460
5ANCE INDEX--///,15X,42H2 = 1/2 * INTEGRAL ((Y9# A8#Y + U9# B8#U)DT,OPT04470
6//,20X,32HSTATE FEEDBACK GAIN DEFINITION--///,25X,1CHU = - C8#X,///OPT04480
7,15X,45HDO YOU WISH TO CONTINUE? "YES" OR "NO".) OPT04490
890 FORMAT (5X,49HALTHOUGH OPTSYSX IS SPECIFICIALLY, SEVEOPT04500
1CALLY DESIGNED TO READ,///,5X,48HALL MATRIX DATA INTERACTIVELY, SEVEOPT04510
2RAT THE,///,5X,31HMETHODS ARE AVAILABLE TO USERS: ///,10X,43HMEOPT04520
3THOD AL--THE DATA FILES: ///,10X,50HMETHOD 2--THE "F", "G", AND "GAMMA" OPT04530
4M SEPARATES MAY BE,///,13X,45HEXPLICITLY DEFINED WITHIN A SUBROUTINE "SOP" OPT04540
5" MATRICES MAY BE,///,13X,45HEXPLICITLY DEFINED WITHIN A SUBROUTINE "SOP" OPT04550
6ETUP,///,10X,52HNO TESTS IN EITHER CASE, THE USER SHOULD OBTAIN A COPT04560
7OPY,///,17X,34HCF THE PROGRAM LISTING AND EXAMINE,///,17X,35HTHE EXAMPOPT04570
8LES, CONTAINED IN S/R "SETUP".6,///,10X,45HDO YOU WISH TO CONTINUE? OPT04580
9 TYPE "YES" OR "NO".) OPT04590
910 FORMAT (///,5X,46HDO YOU WISH TO INPUT THE "F", "G", AND "GAMMA",///OPT04600
110X,40HMATRICES FROM SUBROUTINE "SETUP" IAW THE,///,1CX,40HMETHOD DEOPT04610
2SCRIBED ON THE PREVIOUS SCREEN,///,15X,19HTYPE "YES" OR "NO".) OPT04620
57C FORMAT (25X,24HGENERAL CPTSYSX OPEN-LOOP WITH CPEN-LOOP,///,10X,25HCPCTION 1 -- SYSTOPT04630
1EM ANALYSIS WITHOUT,///,22X,35HOPEN-LOOP 3 -- CPEN-LOOP,///,10X,25HCPCTION 2 -- SYSTOPT04640
2//,10X,42HOPTIONS,///,10X,39HOPTION 3 -- CPEN-LOOP,///,10X,25HCPCTION 3 -- SYSTOPT04650
3SYSTEM CALCULATIONS,///,10X,39HOPTION 4 -- CPEN-LOOP,///,10X,25HCPCTION 4 -- SYSTOPT04660
4OUND,///,22X,23HAND PRGGRAM TERMINATES,///,10X,39HOPTION 4 -- CPEN-LOOP,///,10X,25HCPCTION 4 -- SYSTOPT04670
5OLLOWS IMMEDIATELY.6,///,10X,48HCPCTION 4 -- CPEN-LOOP,///,10X,25HCPCTION 4 -- SYSTOPT04680
6ICES COMPUTED,///,22X,37HWITHOUT FILTER OR REGULATOR SYNTHESIS,///,22XOPT04690
7,25HOR STEADY-STATE ANALYSIS,///,15X,30HSELECT AN OPTION: 1,2,3, OPT04700
8,4.) OPT04710
580 FORMAT (///,5X,46HDO YOU DESIRE RMS VALUES OF STATE AND CONTRL,///OPT04720
110X,19HTYPE "YES" OR "NO".) OPT04730
590 FORMAT (///,20X,30HCPTSYSX LQR/CLASSICAL OPTICS: ///,10X,43HCPCTION 1OPT04740
1FORMAT (///,20X,30HCPTSYSX LQR/CLASSICAL OPTICS: ///,10X,43HCPCTION 1OPT04750
2OPTIMAL CR "K",///,22X,13HMATRIX INPUT,///,10X,43HOPTION 2 -- OPTOPT04760
3OPTIMAL CR "K",///,22X,13HMATRIX INPUT,///,10X,43HOPTION 2 -- OPTOPT04770
4OPTIMAL CR "K",///,22X,13HMATRIX INPUT,///,10X,43HOPTION 2 -- OPTOPT04780
5OPTIMAL CR "K",///,22X,13HMATRIX INPUT,///,10X,43HOPTION 2 -- OPTOPT04790
6OPT,///,10X,43HOPTION 2 -- OPTOPT04800

```



```

2NTRLS OF THE CONTROLLED SYSTEM WHEN FORCED BY //,22X,45+PROCESS ANOPT05290
3C MEASUREMENT IDENT IN THE //,22X,46+PROGRAM TO USE THIS OPT05300
4R MUST BE RESIDUALS OF EACH TRANSFER FUNCTION //,22X,28HPRGAM TO USE THIS OPT05310
5,10X,53HRESIDUALS OF EACH TRANSFER FUNCTION //,22X,28HPRGAM TO USE THIS OPT05320
6X,34HRESIDUALS OF EACH TRANSFER FUNCTION //,22X,28HPRGAM TO USE THIS OPT05330
7COMPUTATION //,10X,24HOPTION 3 -- NOT DESIRED //,10X,29HSELECT A OPT05340
8N OPT05350
9FORMAT //,10X,24HOPTION 3 -- NOT DESIRED //,10X,29HSELECT A OPT05350
10FORMAT //,10X,24HOPTION 3 -- NOT DESIRED //,10X,29HSELECT A OPT05350
11OPT05360
12OPT05370
13OPT05380
14OPT05390
15OPT05400
16OPT05410
17OPT05420
18OPT05430
19OPT05440
20OPT05450
21OPT05460
22OPT05470
23OPT05480
24OPT05490
25OPT05500
26OPT05510
27OPT05520
28OPT05530
29OPT05540
30OPT05550
31OPT05560
32OPT05570
33OPT05580
34OPT05590
35OPT05600
36OPT05610
37OPT05620
38OPT05630
39OPT05640
40OPT05650
41OPT05660
42OPT05670
43OPT05680
44OPT05690
45OPT05700
46OPT05710
47OPT05720
48OPT05730
49OPT05740
50OPT05750
51OPT05760
52OPT05770
53OPT05780
54OPT05790
55OPT05800
56OPT05810
57OPT05820
58OPT05830
59OPT05840
60OPT05850
61OPT05860
62OPT05870
63OPT05880
64OPT05890
65OPT05900
66OPT05910
67OPT05920
68OPT05930
69OPT05940
70OPT05950
71OPT05960
72OPT05970
73OPT05980
74OPT05990
75OPT06000
76OPT06010
77OPT06020
78OPT06030
79OPT06040
80OPT06050
81OPT06060
82OPT06070
83OPT06080
84OPT06090
85OPT06100
86OPT06110
87OPT06120
88OPT06130
89OPT06140
90OPT06150
91OPT06160
92OPT06170
93OPT06180
94OPT06190
95OPT06200
96OPT06210
97OPT06220
98OPT06230
99OPT06240

```



```

CALL FRTCMS ('FILEDEF', '03', 'DISK', 'X29A82', '
1 DATA 'A
-----
THIS IS AN EXAMPLE OF A 83 X 84 DATA FILE X29A82 DATA A1 READ FROM
A USER'S DISK AND CONVERTED (FROM A "DUMMY" ARRAY NAMED 'DUM') TO A
SYMMETRIC ARRAY. THE FCRMAT STATEMENT MUST MATCH YOUR DISK DATA
FCRMAT OR THE PROGRAM WILL FAIL. NOTE: ALL PROGRAM DIMENSIONS
MUST BE ENLARGED ACCORDINGLY FOR A SYSTEM OF THIS SIZE.
-----
READ (3,100)
READ (3,100)
DO 20 I=1,NS
READ (3,50) (BA(I,J),J=1,NS)
CONTINUE
20
-----
THESE ARE EXAMPLES OF SEVERAL POSSIBLE METHODS OF ARRAY GENERATION
WITHIN SUBROUTINE SETUP. THE "GAM" ARRAY WAS SET TO ZERO SINCE NO
"NOISE" WAS PRESENT, AND THE NON-ZERO ELEMENTS OF THE "G" ARRAY WERE
EXPLICITLY DEFINED. THEY COULD ALSO BE READ FROM FILES AS ABOVE.
-----
DO 40 I=1,NS
DO 30 J=1,NC
GAM(I,J)=0.0D+00
G(I,J)=0.0D+00
CONTINUE
G(82,1)= 0.1000D+00
G(52,1)= 0.362D+07
G(77,1)= 0.1591D+02
G(78,1)= 0.2448D+00
G(79,1)= 0.2448D+00
G(81,1)= 0.1000D+00
-----
READ (3,100)
DO 70 I=1,NS
READ (3,50) (G(I,J),J=1,NC)
CONTINUE
70
-----
THIS IS AN EXAMPLE OF ONE POSSIBLE METHOD OF ARRAY GENERATION
WITHIN THE PROGRAM ITSELF. FOR VERY LARGE DATA ARRAYS, THIS METHOD
MAY BE PREFERABLE TO SOME USERS OVER INTERACTIVE ENTRY OF EACH
INDIVIDUAL ELEMENT.
-----
DO 2 I=1,11
DO 1 J=1,82
HO(I,J) = 0.0D+00
HO(1,1) = 0.11520D+00
HO(2,75) = 0.5730D+02

```

```

C-----HO(3,74) = C.1000D+01OPT06730
C-----HO(4,63) = C.5730D+02OPT06740
C-----HO(5,62) = C.1000D+01OPT06750
C-----HO(6,76) = C.5730D+02OPT06760
C-----HO(7,44) = C.5730D+02OPT06770
C-----HO(8,45) = C.5730D+02OPT06780
C-----HO(9,46) = C.5730D+02OPT06790
C-----HO(10,47) = C.5730D+02OPT06800
C-----HO(11,48) = C.5730D+02OPT06810
C1-----CONTINUEOPT06820
C2-----CONTINUEGC TO 90OPT06830
C3-----CONTINUEOPT06840
C-----READ(3,100)OPT06850
C-----DC 90 I=1,NCOPT06860
C-----READ(3,50) (HO(I,J),J=1,NS)OPT06870
C-----CONTINUEOPT06880
C-----RETURNOPT06890
C-----OPT06900
C-----FORMAT(5E12,4)OPT06910
C-----FORMAT(3(E22,15,3X))OPT06920
C-----ENDOPT06930
C-----OPT06940
C-----OPT06950
C-----OPT06960
C=====OPT06970
C-----SUBROUTINE CHECK(EPS,NC,NG,NO,IRET)OPT06980
C-----CHECKS THE CONSISTENCY OF REQUESTED OPTIONS.=OPT06990
C=====OPT07000
C-----DOUBLE PRECISION EPSOPT07010
C-----COMMON/PRCG/IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAE,IDEROPT07020
C-----1UG,1SET,IPEG,IPSD,IYU,INORMOPT07030
C-----SET MODAL ANALYSIS WHEN OL EIGENSYS OR CL TF REQUESTED-----OPT07040
C-----IF (IM.EQ.1 .AND. IOL.EQ.0) IOL=1OPT07050
C-----IF (IOL.EQ.3 .OR. ITF1.NE.0) IM=1OPT07060
C-----CHECK TO SEE IF H MATRIX INPUT-----OPT07070
C-----IF (NO.NE.0 .OR. ICL.GE.2) GO TO 10OPT07080
C-----WRITE(5,9C)OPT07090
C-----IRET=1OPT07100
C-----RETURNOPT07110
C-----CONTINUEOPT07120
C-----TRANSFER FUNCTION CHECKS-----OPT07130
C-----IF (IE.EQ.0) IE=6OPT07140
C-----EPS=10.*(-IE)OPT07150
C-----OPEN LOOP TF-----OPT07160
C-----IF (ITF1.EC.0 .OR. AC.NE.0) GO TO 20OPT07170
C-----WRITE(5,1CC)OPT07180
C-----IRET=1OPT07190
C-----RETURNOPT07200

```

```

C-----COMPENSATOR TF-----OPT07210
2C IF (ITF3.EC. 0) GO TO 30 OPT07220
   IF (IREG.EC. 0.AND. (NC.NE. 0.AND. NG.NE. 0)) GO TC 30 OPT07230
   WRITE(5,11C) OPT07240
   RETURN OPT07250
   CONTINUE OPT07260
C-----NCISE TF-----OPT07270
2C IF (ITF2.EC. 0) GO TO 40 OPT07280
   IF (NG.NE. 0.AND. NC.NE. 0) GO TO 40 OPT07290
   WRITE(5,12C) OPT07300
   RETURN OPT07310
C-----DESTABILIZATION RESTRICTIONS-----OPT07320
4C IF (IDSTAB.EC. 0) GO TO 50 OPT07340
   IF (NC.EC. 0) GO TC 50 OPT07350
   IF (NG.NE. 0) IREG=1 OPT07360
   WRITE(5,13C) OPT07370
   IF (IREG.EC. 1) GO TO 50 OPT07380
   RETURN OPT07390
   CONTINUE OPT07400
C-----PSD INPUT-----OPT07410
5C IF (IPSD.EC. 0) GO TO 80 OPT07420
   IF (IPSD.LT. 0.OR. IPSD.GT. 3) GO TO 60 OPT07430
   IF (IYU.LT. 0.OR. IYU.GT. 3) GO TO 60 OPT07440
   IF (INORM.LT. 0.OR. INORM.GT. NG+MD) GO TC 60 OPT07450
   GO TO 70 OPT07460
   WRITE(5,14C) OPT07470
   RETURN OPT07480
   CONTINUE OPT07490
7C IF (IREG.EC. 0.AND. NC.NE. 0) GO TO 80 OPT07510
   WRITE(5,15C) OPT07520
   RETURN OPT07530
   CONTINUE OPT07540
   RETURN OPT07550
C-----MATRIX MUST BE INPUT, I.E. "NC" MUST BE > 0.-----OPT07560
9C FORMAT (//,EX,49H H - MATRIX MUST BE INPUT, I.E. "NC" MUST BE > 0. OPT07570
   1,/) OPT07580
10C FORMAT (//,EX,46H(G) MATRIX MUST BE INPUT, I.E. NC MUST BE > 0.,/, OPT07590
   110X,26HTD CCMPUTE OPEN LOOP TO F.,/) OPT07600
110 FORMAT (//,EX,48HREGULATOR AND FILTER SYNTHESIS MUST BE REQUESTED, OPT07620
   1/,5X,44HIN THE SAME RUN TO COMPUTE COMPENSATOR T. F.,/,5X,47H I.E. OPT07630
   2IPEG MUST = 0.;"NC" AND "NG" MUST BE > 0.,/) OPT07640
120 FOPMAT (//,EX,51HNCISE T. F. CALCULATED ONLY WHEN REGULATOR DESIGN OPT07650
   1ED,/,5X,47H I.E. IREG MUST = 1.;"NC" AND "NG" MUST BE > 0.,/) OPT07660
130 FOPMAT (//,EX,47HDESTABILIZATION OPTION DESIGNED FOR A REGULATOR, OPT07670
   1,/) OPT07680

```



```

1C      CALL RCREAL (ANSR)
2C      CSTAB=ANSR
3C      DO 10 I=1,NS
4C      DESTAB(I)=CSTAB
5C      CCNTINUE
6C      GO TO 30
7C      CALL SETUP (BA,G,GAM,HO,NS,NC,NG,NO)
8C      CCNTINUE
9C      WRITE (6,1350)
10C      DO 40 I=1,NS
11C      WRITE (6,1350) (BA(I,J),J=1,NS)
12C      IF (IDSTAB.EQ.0) GO TO 50
13C      WRITE (6,1450)
14C      WRITE (6,1350) (DESTAB(I),I=1,NS)
15C      CCNTINUE
16C      -----EIGENSYSTEM OF THE OPEN LOOP DYNAMICS-----
17C      IF (IOL.EQ.0) GO TO 90
18C      IF (IOL.EQ.0) GO TO 90
19C      DO 60 I=1,NS
20C      DO 60 J=1,NS
21C      GN(I,J)=BAL(I,J)
22C      CALL BALANC (NS,NS,GN,LOW,HIGH,D1)
23C      CALL ORTHES (NS,NS,LCW,HIGH,GN,D2)
24C      CALL ORTRAN (NS,NS,LCW,HIGH,GN,D2,SC)
25C      CALL HQP2 (NS,NS,LOW,HIGH,GN,CWR,SC,IERR)
26C      IF (IERR.NE.0) CALL EEXIT (NS,GN,SC,IERR)
27C      CALL BALBAK (NS,NS,LCW,HIGH,D1,NS,SC)
28C      -----NORMALIZE AND PRINT OPEN LOOP EIGENSYSTEM-----
29C      IWRITE=1
30C      CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNCRMI,FC,CM,
31C      1NO,NS)
32C      -----CPTGRCL/CPTGRNO DATA-----
33C      DO 64 I = 1,NS
34C      WRITE (1,2000) CWR(I), CWI(I)
35C      WRITE (1,2000) CWR(I), CWI(I)
36C      CCNTINUE
37C      IF (IOL.EQ.2) RETURN
38C      IF (IOL.EQ.0) OR (NC.NE.0) OR (IDSTAB.GT.0) GO TO 50
39C      DO 70 I=1,NS
40C      IF (CWR(I).LT.0.) GO TO 70
41C      WRITE (5,1450)
42C      RETURN
43C      CCNTINUE
44C      GO TO 130
45C      DO 50 I=1,NS
46C      DO 80 J=1,NS
47C      W1(I,J)=SC(I,J)

```

OPT08170
OPT08180
OPT08190
OPT08200
OPT08210
OPT08220
OPT08230
OPT08240
OPT08250
OPT08260
OPT08270
OPT08280
OPT08290
OPT08300
OPT08310
OPT08320
OPT08330
OPT08340
OPT08350
OPT08360
OPT08370
OPT08380
OPT08390
OPT08400
OPT08410
OPT08420
OPT08430
OPT08440
OPT08450
OPT08460
OPT08470
OPT08480
OPT08490
OPT08500
OPT08510
OPT08520
OPT08530
OPT08540
OPT08550
OPT08560
OPT08570
OPT08580
OPT08590
OPT08600
OPT08610
OPT08620
OPT08630
OPT08640

```

          CALL MINV (NSQ,W11,NS,DDD,D1,D2)
          CCNT INUE
          IF (IDSTAB.EQ.0) GO TO 130
          FORM U * DIAG(DESTAB) * U-INV-----
100      DO 100 J=1,NS
          DO 100 I=1,NS
          AA(I,J)=WNCFM(I,J)*DESTAB(J)
          CC 120 I=1,NS
          DO 120 J=1,NS
          CDD=0.00
          DO 110 K=1,NS
          CDD=DDC+AA(I,K)*WNORMI(K,J)
          DSTORE(I,J)=DDD
          BA(I,J)=BA(I,J)+DDD
110      CCNT INUE
          IF (NO.EQ.0) GO TO 145
          IF (ISET.EC.1) GO TO 135
          CALL READH (NO,NS,ISAH,HD)
          CCNT INUE
          WRITE (6,1440)
          DO 1140 I=1,NC
          WRITE (6,1450) (HD(I,J),J=1,NS)
          IF (IM.NE.1) GO TO 150
          CALL IMODE (WNORM,HO,CM,NS,NO,NS,2)
125      CCNT INUE
          IF (IFCFW.EC.0) GO TO 170
          CALL READD (NO,NC,D)
          WRITE (6,1470)
          DO 1160 I=1,NC
          WRITE (6,1480) (D(I,J),J=1,NC)
          CCNT INUE
          NOB=0
          IF (INC.EQ.0) GO TO 550
          IF (IOL.EC.2) GO TO 270
          IF (IR.NE.1.AND.0IR.NE.3) GO TO 210
          IF (ISET.EC.1) GO TO 180
          CALL READG (NS,NC,ISAG,G)
          CCNT INUE
          CALL IRADFE (NC,NS,FBGC)
          WRITE (6,1400)
          DO 1190 I=1,NS
          WRITE (6,1490) (G(I,J),J=1,NC)
          IF (IM.NE.1) GO TO 200
          CALL IMODE (WNORMI,G,BM,NS,NS,NC,0)
          CCNT INUE
          GO TO 330
          DO 220 I=1,NS
          DO 220 J=1,NS

```

```

OPT08650
OPT08660
OPT08670
OPT08680
OPT08690
OPT08700
OPT08710
OPT08720
OPT08730
OPT08740
OPT08750
OPT08760
OPT08770
OPT08780
OPT08790
OPT08800
OPT08810
OPT08820
OPT08830
OPT08840
OPT08850
OPT08860
OPT08870
OPT08880
OPT08890
OPT08900
OPT08910
OPT08920
OPT08930
OPT08940
OPT08950
OPT08960
OPT08970
OPT08980
OPT08990
OPT09000
OPT09010
OPT09020
OPT09030
OPT09040
OPT09050
OPT09060
OPT09070
OPT09080
OPT09090
OPT09100
OPT09110
OPT09120

```



```

WRITE (6,1570)
CALL RAPRT (NS,NS,NS,9,SC,4,'(9(1X,1PD13.6))')
CCNT INUE
450 C-----RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-----
      IF (IDSTAB.EQ.0) GC TO 470
      CC 460 I=1,NS
      BA(I,1)=BA(I,1)-DESTAB(I)
      IR=1
460 C-----CALCULATION OF FEEDBACK GAIN-----
      CCNT INUE
      U = -(BINVERSE)*GT*GNE
      FEEDBACK GAINS-----
      CALCULATE GT-----
      DO 490 I=1,NC
      CC 490 J=1,NS
      PRO(I,J)=0.0
      DO 480 K=1,MH
      PRO(I,J)=PRO(I,J)+C(K,I)*GN(K,J)
480 C-----NORMALIZE AND PRINT OPT. REG. CLOSED LOOP EIGENSYSTEM-----
490 C-----
      IF (IDSTAB.EQ.1) GC TO 500
      IWRITE=2
      CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNCRMI,FBGC,
      1AA,NC,NS)
      THE OPTIMUM FEEDBACK CONTROL GAINS-----
500 C-----
      WRITE (6,1580)
      DO 510 I=1,NC
      WRITE (6,1590) (FBGC(I,J),J=1,NS)
510 C-----COMPUTE MICAL C MATRIX OPEN LOOP U-INVERSE SAVED IN WNORMI8-----
      IF (IM.NE.1) GO TO 530
      C IN COMPUTING MICAL C RECCOMPUTE U OPEN LOOP SINCE WNORM USED TO STCRE
      C U & U-INV FOR CLOSED LOOP SYSTEMS; WNCRMI USED TO SAVE U-INV OPEN LOOP
      DO 520 I=1,NS
      DO 520 J=1,NS
      WNCRMI(I,J)=WNORMI(I,J)
520 C-----
      CALL MINV (NSQ,WNORM,NS,DDD,D1,D2)
      CALL MCDE (WNORM,FBGC,AA,NS,NC,NS,3)
      CCNT INUE
      THE CLOSED LOOP DYNAMICS MATRIX-----
530 C-----
      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)
540 C-----
      ACL(I,J)=BA(I,J)+SUM
      WRITE (6,1600)
      CALL RAPRT (MH,MH,MH,5,ACL,4,'(5(1X,1PD13.6))')

```

```

560 IF (IR.NE.1.AND.IR.NE.3) GO TO 590
DO 560 I=1,NS
DO 560 J=1,NS
GN(I,J)=ACL(I,J)
CALL BALANC (NS,NS,GN,LOW,IHIGH,D1)
CALL ORTRAN (NS,NS,LCW,IHIGH,GN,D2,SC)
CALL HQR2 (NS,NS,LOW,IHIGH,GN,CWR,CW,SC,IERR)
IF (IERR.NE.0) CALL IEREXIT (NS,GN,IERR)
CALL BALBAK (NS,NS,LOW,IHIGH,D1,NS,SC)
C-----NORMALIZE AND PRINT CLOSED LOOP SUBOPT. REG. EIGENSYSTEM-----
IWRITE=3
IWRITE (CWR,CW,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNCRMI,FEGC,
1AA,NC,NS)
DO 570 I=1,NS
IF (CWR(I).LT.0.0) GC TO 570
WRITE (5,1410)
RETURN
570 CCNT INUE
IF (IO.NE.1) GO TO 590
DO 580 I=1,NS
DO 580 J=1,NS
W1(I,J)=S(I,J)
CALL MINV (NSQ,W1,NS,DDD,D1,D2)
NCB=NO
IF (NG.EC.0) RETURN
IF (ISET.EC.1) GO TO 610
CALL READG2 (NS,NG,IGAM,GAM)
580 CCNT INUE
IF (IOL.EC.3) GO TO 620
CALL PEADQ (NG,Q)
WRITE (6,1420)
DO 630 I=1,NS
WRITE (6,1430) (GAM(I,J),J=1,NG)
IF (IMCDE.1) GO TO 640
CALL MCDE (WNORMI,GAM,AA,NS,NS,NG,1)
590 CCNT INUE
IF (IOL.EC.3) RETURN
WRITE (6,1420)
DO 650 I=1,NG
WRITE (6,1430) (Q(I,J),J=1,NG)
IF ((IG.EC.1).AND.(NG.EC.0)) GO TO 1260
DO 660 I=1,NS
DO 660 J=0,LO
PRO(I,J)=PRC(I,J)+Q(I,K)*GAM(J,K)
660 DO 670 I=1,NS
DO 670 J=1,NS
OPT10570
OPT10580
OPT10590
OPT10600
OPT10610
OPT10620
OPT10630
OPT10640
OPT10650
OPT10660
OPT10670
OPT10680
OPT10690
OPT10700
OPT10710
OPT10720
OPT10730
OPT10740
OPT10750
OPT10760
OPT10770
OPT10780
OPT10790
OPT10800
OPT10810
OPT10820
OPT10830
OPT10840
OPT10850
OPT10860
OPT10870
OPT10880
OPT10890
OPT10900
OPT10910
OPT10920
OPT10930
OPT10940
OPT10950
OPT10960
OPT10970
OPT10980
OPT10990
OPT11000
OPT11010
OPT11020
OPT11030
OPT11040

```


AD-A152 148

DEVELOPMENT OF GRAPHICAL POLE-ZERO ROOT-LOCUS BODE
NYQUIST AND NICHOLS RESPONSES USING THE OPTSYSX PROGRAM
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M H LAPTAS

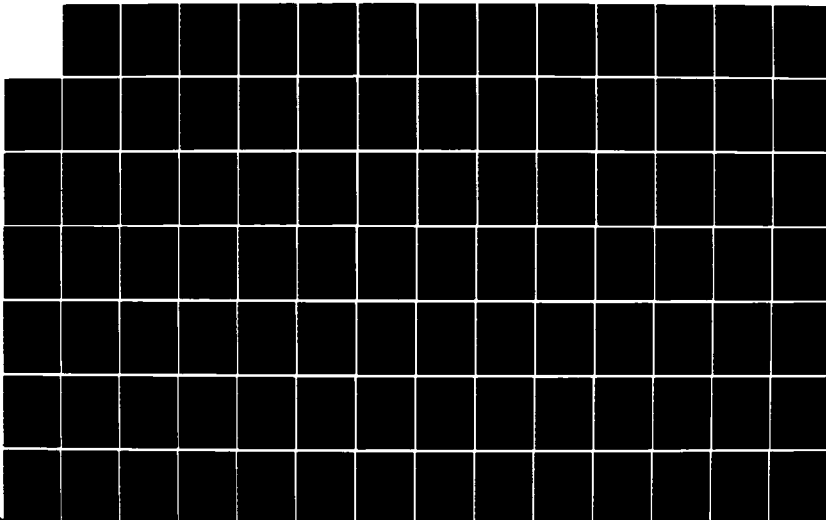
2/3

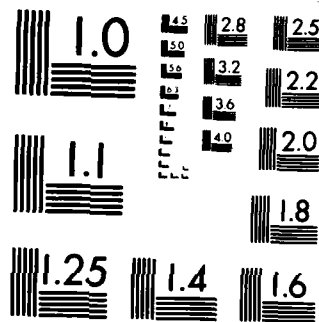
UNCLASSIFIED

SEP 84

F/G 9/2

NL





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

```

74C RM(I+MH,J+MF)=-BA(J,I)
    RM(I,J+NS)=CC(I,J)
    GC TO 400
C-60 BACK TO 450 TO SET UP THE FILTER HAMILTONIAN: CALC. THE FILTER GAINS
750 CALL RGAIN (M,NS,NC,NCB,WR,WI,X,GN,GM, RM,W21,D1,CR,CI,PRO,M+S,D2)
C-----CHECK EIGENVECTORS-----
    IF (IDEBUG.EQ. 0) GC TO 760
    WRITE (6,I=70)
    CALL RAPRNT (NS,NS,NS,9,PRO,4,'(9(1X,1PD13.6))')
760 CCNTINUE
    IF (IDSTAB.EQ. 1) GC TO 770
C-----NORMALIZE AND PRINT OPT. ESTIMATOR EIGENSYSTEM-----
    IWRITE=4
    CALL CNORM (CR,CI,PRC,NS,IWRITE,NSQ,DDD,D1,D2,WNCRM,WNCRMI,HC,AA,
77C 1NC,NS)
    CO 780 I=1,MH
    CO 780 J=1,NC
78C PRO(I,J)=+PC(J,I)/RC(J,J)
    DO 790 I=1,MH
    DO 790 J=1,NC
    FBGE(I,J)=C.DO
    DO 790 K=1,MH
    FEGE(I,J)=FEGE(I,J)+GN(I,K)*PRO(K,J)
790 IF (IDSTAB.EQ. 1) GC TO 810
    WRITE (6,I=70)
    CALL RAPRNT (MH,MH,MH,5,GN,4,'(5(1X,1PD13.6))')
    IWRITE=4
    CO 800 I=1,MH
    X(I,I)=DSCRT(GN(I,I))
80C WRITE (6,I=70) (X(I,I),I=1,MH)
810 CO 820 I=1,MH
    CO 820 J=1,NC
    FBGE(I,J)=1.40 (FBGE(I,J),J=1,NC)
82C WRITE (6,I=70) MODAL K MATRIX OPEN LOOP U-INV SAVED IN WNCRMIG-----
C-----COMPUTE MODE I GO TO 830
    IF (IM.NE. 1) GO TO 830
    CALL MODE (WNCRMI,FEGE,AA,MH,MH,NO,4)
830 CCNTINUE
C-----RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-----
    IF (IDSTAB.EQ. 0) GC TO 850
    DO 840 I=1,NS
    DO 840 J=1,NS
84C PA(I,J)=BA(I,J)-DSTORE(I,J)
    IR=2
850 CCNTINUE
    DO 870 I=1,NS
    DO 870 J=1,NS
    SUM=0.0
    DO 860 K=1,NC

```

```

840      SUM=SUM+FBGE(I,K)*HC(K,J)
850      PRO(I,J)=BA(I,J)-SUM
      WRITE(6,1450)
      CALL RAPRNT(NS,NS,NS,5,PRO,4,'(5(1X,1PD13.6))')
      IF(IR.LT.2) GO TO 890
      CALL BALANC(NS,NS,PRO,LOW,IGH,D1)
      CALL ORTHES(NS,NS,LCW,IGH,PRO,D2)
      CALL ORTRAN(NS,NS,LCW,IGH,PRO,D2,GM)
      CALL HQR2(NS,NS,LOW,IGH,PRO,CR,CI,GM,IEFR)
      IF(IEFR.NE.0) CALL EEXIT(NS,PRO,IEFR)
      CALL BALBAK(NS,NS,LCW,IGH,D1,NS,GM)
      WRITE(6,1460)
      C-----NORMALIZE AND PRINT SUBPT. ESTIMATOR EIGENSYSTEM-----
      IWRITE=5
      CALL CNORM(CR,CI,GM,NS,IWRITE,NSQ,DD,D1,C2,WNCRM,WNCRMI,HC,AA,
      1NO,NS)
      DO 880 I=1,NS
      IF(CR(I).LT.0.0) GO TO 880
      WRITE(5,1460)
      RETURN
      CCNT INUE
      GO TO 900
880      IF(IQ.EQ.C) GO TO 1260
890      DO 910 I=1,NH
      DO 910 J=1,NH
      PRO(I,J)=0.0
      DO 910 K=1,NC
      PRO(I,J)=PRC(I,J)+RC(I,K)*FBGE(J,K)
      DO 920 I=1,NH
      DO 920 J=1,NH
      CQ(I,J)=0.0
      DO 920 K=1,NC
      CQ(I,J)=CQ(I,J)-FBGE(I,K)*PRO(K,J)
      CCNT INUE
      C-----THE RMS STATE AND CONTROL RESPONSES-----
      IR=IR+1
      GO TO (1090,1090,940,940), IR
840      DO 950 I=1,NS
      DO 950 J=1,NG
      X(I,J)=0.0
      DO 950 K=1,NG
      X(I,J)=X(I,J)+GAM(I,K)*Q(K,J)
      DO 970 I=1,NS
      DO 970 J=1,NS
      SUM=0.0
      DO 960 K=1,NG
      SUM=SUM-X(I,K)*GAM(J,K)
      PRO(I,J)=SUM+CQ(I,J)

```

```

OPTI2010
OPTI2020
OPTI2030
OPTI2040
OPTI2050
OPTI2060
OPTI2070
OPTI2080
OPTI2090
OPTI2100
OPTI2110
OPTI2120
OPTI2130
OPTI2140
OPTI2150
OPTI2160
OPTI2170
OPTI2180
OPTI2190
OPTI2200
OPTI2210
OPTI2220
OPTI2230
OPTI2240
OPTI2250
OPTI2260
OPTI2270
OPTI2280
OPTI2290
OPTI2300
OPTI2310
OPTI2320
OPTI2330
OPTI2340
OPTI2350
OPTI2360
OPTI2370
OPTI2380
OPTI2390
OPTI2400
OPTI2410
OPTI2420
OPTI2430
OPTI2440
OPTI2450
OPTI2460
OPTI2470
OPTI2480

```

```

570      PRO(J,I)=FFC(I,J)
        CQ(I,J)=SUM
        CQ(J,I)=SUM
        W21(I,J)=GM(I,J)
        CALL MINV (NSQ,W21,NS,DDD,DI,D2)
        CALL SCOV (NS,GM,W21,CR,CI,NS,GM,W21,CR,CI,PRC,GN)
        WRITE (6,1670)
        CALL RAPRT (MH,MH,MH,5,GN,4,*(5(1X,1PD13.6)))
        WRITE (6,1680)
        DO 980 I=1,MH
          X(I)=DSQR(GN(I,I))
          IF (I=1) GO TO 1260
          IF (I=1000) GO TO 1260
          IF (I=1000) I=1,NS
          SUM=0.0
          DO 990 K=1,NS
            SUM=SUM+FBGC(I,K)*GN(K,I)
          X(I,J)=SUM
          DO 1020 I=1,NS
            SUM=0.0
            IF (NC=0) GO TO 1020
            DO 1010 K=1,NC
              SUM=SUM+G(I,K)*X(K,J)
            PRO(I,J)=CQ(I,J)+SUM
            CALL SCOV (NS,SC,W11,CWR,CWI,NS,GM,W21,CR,CI,PRO,BA)
            IF (NC=0) GO TO 1040
            DO 1030 I=1,NC
              DO 1030 J=1,NS
                W21(I,J)=0.0
                W21(I,J)=W21(I,J)+FBGC(I,K)*BA(J,K)
              DO 1060 I=1,NS
                SUM=0.0
                IF (NC=0) GO TO 1060
                DO 1050 K=1,NC
                  SUM=SUM+G(I,K)*W21(K,J)
                PRO(I,J)=SUM
                DO 1070 I=1,NS
                  PRO(I,J)=PFC(I,J)
                PRO(I,J)=CQ(I,J)+PRO(J,I)
                CALL SCOV (NS,SC,W11,CWR,CWI,NS,SC,W11,CWP,CWI,PRO,CQ)
                DO 1080 I=1,NS

```

```

OPT1125490
OPT1125500
OPT1125510
OPT1125520
OPT1125530
OPT1125540
OPT1125550
OPT1125560
OPT1125570
OPT1125580
OPT1125590
OPT1126000
OPT1126100
OPT1126200
OPT1126300
OPT1126400
OPT1126500
OPT1126600
OPT1126700
OPT1126800
OPT1126900
OPT1127000
OPT1127100
OPT1127200
OPT1127300
OPT1127400
OPT1127500
OPT1127600
OPT1127700
OPT1127800
OPT1127900
OPT1128000
OPT1128100
OPT1128200
OPT1128300
OPT1128400
OPT1128500
OPT1128600
OPT1128700
OPT1128800
OPT1128900
OPT1129000
OPT1129100
OPT1129200
OPT1129300
OPT1129400
OPT1129500
OPT1129600

```

```

1080 GM(I,J)=CQ(I,J)-BA(I,J)-BA(J,I)+GN(I,J)
1090 GM(J,I)=GM(I,J)
1100 GO TO 1100
1110 CALL SCOV (NS,SC,W1,CWR,CWI,NS,SC,W1,CHR,CWI,CC,GM)
1120 IF (NC.EQ.0) GO TO 1150
1130 DO 1120 I=1,NS
1140 DO 1120 J=1,NC
1150 PRO(I,J)=0
1160 DO 1110 K=1,NS
1170 PRO(I,J)=PFC(I,J)+GM(I,K)*FBGC(J,K)
1180 CCNT=INUE
1190 DO 1140 I=1,NC
1200 DO 1140 J=1,NC
1210 SC(I,J)=0
1220 DO 1130 K=1,NS
1230 SC(I,J)=SC(I,J)+FBGC(I,K)*PRO(K,J)
1240 CCNT=INUE
1250 IF (IREG .EQ. 0) GO TO 1170
1260 DO 1160 I=1,NS
1270 DO 1160 J=1,NS
1280 CQ(I,J)=GM(I,J)
1290 WRITE (6,170)
1300 CALL RAPRT (MH,MH,MH,5,GM,4,'(5(1X,1PD13.6))')
1310 IF (IR.GT.2) GO TO 1190
1320 DO 1180 I=1,MH
1330 DO 1180 J=1,MH
1340 CQ(I,J)=GN(I,J)+GM(I,J)
1350 CCNT=INUE
1360 WRITE (6,171)
1370 CALL RAPRT (MH,MH,MH,5,CQ,4,'(5(1X,1PD13.6))')
1380 IF (NC.EQ.0) GO TO 1210
1390 WRITE (6,172)
1400 DO 1200 I=1,NC
1410 DO 1200 J=1,NC
1420 SC(I,J)=SC(I,J)+CQ(I,I)
1430 IF (NC.EQ.0) GO TO 1240
1440 DO 1230 I=1,NC
1450 DO 1230 J=1,NC
1460 SC(I,J)=DSCT(SC(I,I))
1470 WRITE (6,174)
1480 DO 1250 I=1,NS
1490 IF (I.LE.NC) WRITE (6,175) CQ(I,I),SC(I,I)
1500 IF (I.GT.NC) WRITE (6,175) CQ(I,I)
1510 CCNT=INUE
1520 IF (ITF3 .EQ. 0) GC TC 1290
1530 IF (ITF3 .EQ. 0) FORM COMPENSATOR FROM MEAS TO INPUT AND COMPUTE TF
1540 DC 1280 I=1,NS

```

```

OPT112970
OPT112980
OPT112990
OPT113000
OPT113010
OPT113020
OPT113030
OPT113040
OPT113050
OPT113060
OPT113070
OPT113080
OPT113090
OPT113100
OPT113110
OPT113120
OPT113130
OPT113140
OPT113150
OPT113160
OPT113170
OPT113180
OPT113190
OPT113200
OPT113210
OPT113220
OPT113230
OPT113240
OPT113250
OPT113260
OPT113270
OPT113280
OPT113290
OPT113300
OPT113310
OPT113320
OPT113330
OPT113340
OPT113350
OPT113360
OPT113370
OPT113380
OPT113390
OPT113400
OPT113410
OPT113420
OPT113430
OPT113440

```

```

1270      DO 1280 J=1,NS
1280      SUM=0
      DO 1270 K=1,NC
      SUM=SUM+FBGE(I,K)*HC(K,J)
      CQ(I,J)=ACL(I,J)-SUM
      WRITE (6,176C)
      ITEX=3
      IZERO=0
      CALL TF (NS,NS,NSQ,CC,AA,NC,FBGE,BM,NC,FBGC,CM,IZERC,D,EB,CC,CP,
      1WR,WI,CWR,CWI,SC,JCF,RES,DI,D2,DDD,EPS,ITF2,ITFX)
1290      C----- COMPUTE PSD FUNCTIONS OF THE CONTROLLED SYSTEM -----
      IF (IPSD.EC.O) GO TO 1310
      IF (IYU.LT.3) GO TO 1300
      CALL PSDCAL (M,NS,RM,X,NC,GM,GV,FBGC,NO,HY,HU,HQ,FBGE,NG,
      1GAM,ACL,BA,WR,WI,DI,D2,JCF,RES,Q,RC,BB,CC,1,IPSD,INORM)
      CALL PSDCAL (M,NS,RM,X,NC,GM,GV,FBGC,NO,HY,HU,HQ,FBGE,NG,
      1GAM,ACL,BA,WR,WI,DI,D2,JCF,RES,Q,RC,BB,CC,2,IPSD,INORM)
      GO TO 1310
1300      CALL PSDCAL (M,NS,PM,X,NC,GM,GV,FBGC,NO,HY,HU,HQ,FBGE,NG,
      1GAM,ACL,BA,WR,WI,DI,D2,JCF,RES,Q,RC,BB,CC,IYU,IPSD,INORM)
      IF (ISC.EC.O) RETURN
      IF (INC.NE.O) GO TO 1350
      DO 1320 I=1,NS
      DO 1320 J=1,NS
      ACL(I,J)=BA(I,J)
      CCONTINUE
      CALL MINIV (NSQ,ACL,NS,DDD,D1,D2)
      CALL READW (NG,WR)
      WRITE (6,1770) (WR(I),I=1,NG)
      CO I340 I=1,NS
      WI(I)=0
      DO I340 J=1,NG
      WI(I)=WI(I)+GAM(I,J)*WR(J)
      DO I360 I=1,NS
      CR(I)=0
      DO I350 J=1,NS
      CR(I)=CR(I)-ACL(I,J)*WI(J)
      WRITE (6,1780) CR(I)
      CO I370 I=1,NC
      CI(I)=0
      DO I370 J=1,NS
      CI(I)=CI(I)+FBGC(I,J)*CR(J)
      WRITE (6,1790) (CI(I),I=1,NC)
      RETURN
C670      FORMAT (2X,1P6D14.6,/,2X,6D14.6)

```

```

OPT113450
OPT113460
OPT113470
OPT113480
OPT113490
OPT113500
OPT113510
OPT113520
OPT113530
OPT113540
OPT113550
OPT113560
OPT113570
OPT113580
OPT113590
OPT113600
OPT113610
OPT113620
OPT113630
OPT113640
OPT113650
OPT113660
OPT113670
OPT113680
OPT113690
OPT113700
OPT113710
OPT113720
OPT113730
OPT113740
OPT113750
OPT113760
OPT113770
OPT113780
OPT113790
OPT113800
OPT113810
OPT113820
OPT113830
OPT113840
OPT113850
OPT113860
OPT113870
OPT113880
OPT113890
OPT113900
OPT113910
OPT113920

```



```

177C  FORMAT (//,EX,46HSTEADY DISTURBANCE VECTOR.....W.....//OPTI14410
1,10(1X,1PD12.4//))OPTI14420
1780  FORMAT (//,EX,45HSTEADY STATE VALUES OF STATE VAR. ARE.....//)OPTI14430
1790  FORMAT (//,EX,47HSTEADY STATE CONTROL IS .....//)OPTI14440
1/10(1X,1PD12.4//))OPTI14450
1800  FORMAT (//,EX,49HENTER THE MAGNITUDE OF THE DESTABILIZATION VECTOROPTI14460
1,8X,47HTC BE ADDED DOWN THE DIAGONAL OF THE "F"-MATRIX,/,8X,18HTOPTI14470
2C DESTABILIZE IT.,//)OPTI14480
C-----CFGRAPH DATA-----OPTI14490
C000  FORMAT(5X,2C30.14)OPTI14500
C-----OPTI14510
C-----END-----OPTI14520
C=====OPTI14530
SUBROUTINE RAPRNT (NMWAX,M,N,L,A, IDIM,FMT)
REAL*8 A(NMAX,N)
DIMENSION FMT(IDIM)
NU=L
DO 20 NL=1,N,L
IF (NU.GT.N) NU=N
DC 10 I=1,M
WRITE (6,FMT) (A(I,J),J=R(L,NU)
1C
2C
3C
NU=NU+L
RETURN
FORMAT (1X)
END
C=====OPTI14600
SUBROUTINE RGAIN (M,NS,NC,NOB,WR,WI,VF,GN,W11,TCB,W21,LT,C,CI,CT,M
1HS,MT)OPTI14610
IMPLICIT REAL*8 (A-H,O-Z)OPTI14620
DIMENSION WR(M),WI(M),VF(M,M),GN(NS,NS)OPTI14630
DIMENSION W11(NS,NS),TCB(M,M),W21(NS,NS),LT(NS),MT(NS)OPTI14640
DIMENSION C(NS),CI(NS),CT(NS,NS)OPTI14650
K=1OPTI14660
KP=1OPTI14670
KN=1OPTI14680
NRZEV=0OPTI14690
NCPZEV=0OPTI14700
IF (K.GT.M) GO TO 21C OPTI14710
C-----OPTI14720
C CHECK FOR SIGNAL AT OR NEAR J-OMEGA AXIS TO INCLUDE IN E-L EIGSYS OPTI14730
C TURN FIRST ONE POSITIVE AND SECOND ONE NEGATIVE OPTI14740
C-----OPTI14750
SIGVR=CABS(WR(K))OPTI14760
IF (EIGVR.GE.1.D-10) GO TO 60 OPTI14770
IF (WI(K)) 40,20,40 OPTI14780
NRZEV=NRZEV+1 OPTI14790
IF (NRZEV.GT.1) GO TC 30 OPTI14800
2C OPTI14810
OPTI14820
OPTI14830
OPTI14840
OPTI14850
OPTI14860
OPTI14870
OPTI14880

```

```

3C      WP(K)=EIGVP
        GO TO 80
        WR(K)=-EIGVF
        WRITE (6,25C)
        GO TO 150
4C      NCPZEV=NCPZEV+1
        IF (NCPZEV.GT.1) GO TO 50
        WP(K)=EIGVP
        WR(K+1)=EIGVR
        GO TO 110
5C      WR(K)=-EIGVF
        WR(K+1)=-EIGVR
        WRITE (6,30C)
        GO TO 180
6C      IF (WR(K)) 140,70,70
7C      IF (WI(K)) 110,80,110
8C      IF (NOB.EQ.C) GO TO 100
        C-----EIGENVECTOR FOR REAL EIGENVALUE, POSITIVE-----
9C      DO 90 J=1,N
        TCB(J,KP)=VF(J,K)
        KP=KP+1
10C     K=K+1
        GO TO 10
        C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, POSITIVE REAL PART-----
11C     IF (NOB.EQ.C) GO TO 130
        DO 120 J=1,N
        FR=VF(J,K)
        FI=-VF(J,K+1)
        TCB(J,KP)=FR+FI
        TCB(J,KP+1)=FR-FI
        KP=KP+2
        K=K+2
        GO TO 10
        C-----EIGENVECTOR FOR REAL EIGENVALUE, NEGATIVE REAL PART-----
14C     IF (WI(K)) 180,150,120
        C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, NEGATIVE REAL PART-----
15C     C(KN)=WR(K)
        CI(KN)=WI(K)
        IF (NOB.NE.C) GO TO 170
        KNS=KN+NS
        DO 160 J=1,N
        TCB(J,KNS)=VF(J,K)
        KN=KN+1
        K=K+1
        GO TO 10
        C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, NEGATIVE REAL PART-----
18C     PR=WR(K)
        RI=WI(K)
        C(KN)=PR

```

```

OPT1 4890
OPT1 4900
OPT1 4910
OPT1 4920
OPT1 4930
OPT1 4940
OPT1 4950
OPT1 4960
OPT1 4970
OPT1 4980
OPT1 4990
OPT1 5000
OPT1 5010
OPT1 5020
OPT1 5030
OPT1 5040
OPT1 5050
OPT1 5060
OPT1 5070
OPT1 5080
OPT1 5090
OPT1 5100
OPT1 5110
OPT1 5120
OPT1 5130
OPT1 5140
OPT1 5150
OPT1 5160
OPT1 5170
OPT1 5180
OPT1 5190
OPT1 5200
OPT1 5210
OPT1 5220
OPT1 5230
OPT1 5240
OPT1 5250
OPT1 5260
OPT1 5270
OPT1 5280
OPT1 5290
OPT1 5300
OPT1 5310
OPT1 5320
OPT1 5330
OPT1 5340
OPT1 5350
OPT1 5360

```

```

C(KN+1)=RR
C I(KN)=RI
C I(KN+1)=-RI
IF (NOB.NE.C) GO TO 200
KNS=KN+NS
DO 190 J=1,N
FR=VF(J,K)
FI=-VF(J,K+1)
TCB(J,KNS)=FR+FI
TCB(J,KNS+1)=FR-FI
KN=KN+2
K=K+2
GO TO 10
CONTINUE
IF (NOB.NE.C) GO TO 240
-----FORMATION OF W1-----
DO 220 I=1,NS
DO 220 J=1,NS
W1(I,J)=TCB(I,J+NS)
CT(I,J)=W1(I,J)
-----FORMATION OF W2-----
DO 230 I=1,NS
DO 230 J=1,NS
W2(I,J)=TCB(I+NS,J+NS)
IF (NOB.EQ.C) GO TO 260
DO 250 I=1,NS
DO 250 J=1,NS
W2(I,J)=-TCB(I,J)
W1(I,J)=TCB(I+NS,J)
CCNT INUE
-----INVERT W1-----
NSO=NS*NS
CALL MINV (NSQ,W1,NS,DETC,LI,MT)
-----CALCULATE THE RGAIN MATRIX-----
DO 270 IL=1,NS
DO 270 JL=1,NS
GN(IL,JL)=0.DO
DO 270 KL=1,NS
GN(IL,JL)=GN(IL,JL)+W2(I,KL)*W1(KL,JL)
IF (NOB.EQ.C) RETURN
DO 280 I=1,NS
DO 280 J=1,NS
CT(I,J)=W1(J,I)
RETURN
-----
FORMAT (IX,5H EULER-LAGRANGE EQUATIONS HAVE A REAL EIGENVALUE AT,OPT15810
114H OR NEAR ZERO,/)
FORMAT (IX,49H EULER-LAGRANGE EQUATIONS HAVE A COMPLEX PAIR OF ,40OPT15840

```

```

1EIGENVALUES AT OR NEAR THE J-OMEGA AXIS.)
END
=====
SUBROUTINE MINV (NSQ,A,N,D,L,M)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(NSQ),L(N),M(N)
COMPILE PRECISION A,C,BIGA,HOLD
NV=N*N
C=1.0D0
AK=-N
DO 180 K=1,N
NK=NK+N
L(K)=K
M(K)=K
KK=NK+K
BIGA=A(KK)
DO 20 J=K,N
IZ=N*(J-1)
CC 20 I=K,N
IJ=IZ+I
IF (DABS(BIGA)-DABS(A(IJ))) 10,20,20
1C BIGA=A(IJ)
L(K)=I
M(K)=J
2C CONTINUE
-----INTERCHANGE ROWS-----
J=L(K)
IF (J-K) 5C,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
4C
5C I=M(K)
IF (I-K) 8C,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
7C
C-----DIVIDE COLUMN BY MINUS PIVOT-----
C-----CF PIVOT ELEMENT IS CONTAINED IN BIGA)-----
8C IF (BIGA) 1C0,9C,1C0
(VALUE
1C0
9C
1C1
1C2
1C3
1C4
1C5
1C6
1C7
1C8
1C9
1C10
1C11
1C12
1C13
1C14
1C15
1C16
1C17
1C18
1C19
1C20
1C21
1C22
1C23
1C24
1C25
1C26
1C27
1C28
1C29
1C30
1C31
1C32

```

```

SC      D=0.0D0
RETURN
100     CO 120 I=1,N IIC,120,110
110     IK=NK+I
120     A(IK)=A(IK)/(-BIGA)
C-----REDUCE MATRIX-----
      DC 150 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 150 J=1,N
      IJ=IJ+N
      IF (I-K) 120,150,130
      IF (J-K) 140,150,140
130     KJ=IJ-I+K
140     A(IJ)=HOLD*A(KJ)+A(IJ)
150     CONTINUE
C-----DIVIDE ROW BY PIVOT-----
      KJ=K-N
      DO 170 J=1,N
      KJ=KJ+N
      IF (J-K) 160,170,160
160     A(KJ)=A(KJ)/BIGA
170     CONTINUE
C-----PRODUCT OF PIVOTS-----
      C=D*BIGA
C-----REPLACE PIVOT BY RECIPROCAL-----
      A(KK)=(1.0CC)/BIGA
180     CONTINUE
C-----FINAL ROW AND COLUMN INTERCHANGE-----
      K=N
      K=(K-1)
      IF (K) 260,260,200
      I=L(K)
      IF (I-K) 230,230,210
      JC=N*(K-1)
      JR=N*(I-1)
      CC 220 J=1,N
      JK=JC+J
      FOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
      A(JI)=FOLD
      JM(K)
      IF (J-K) 150,190,240
      KI=K-N
      200
      210
      220
      230
      240

```

```

OPT116330
OPT116340
OPT116350
OPT116360
OPT116370
OPT116380
OPT116390
OPT116400
OPT116410
OPT116420
OPT116430
OPT116440
OPT116450
OPT116460
OPT116470
OPT116480
OPT116490
OPT116500
OPT116510
OPT116520
OPT116530
OPT116540
OPT116550
OPT116560
OPT116570
OPT116580
OPT116590
OPT116600
OPT116610
OPT116620
OPT116630
OPT116640
OPT116650
OPT116660
OPT116670
OPT116680
OPT116690
OPT116700
OPT116710
OPT116720
OPT116730
OPT116740
OPT116750
OPT116760
OPT116770
OPT116780
OPT116790
OPT116800

```

```

DC 250 I=1,N
KI=KI+N
HOLD=A(KI)
JI=KI-K+J
A(KI)=-A(JI)
A(JI)=HOLD
GC TO 190
K=0
RETURN
END
C=====
SUBROUTINE SCOV (NL,NL,WL,WLI,VLI,VL2,NR,WR,WRI,VRI,VR2,Q,X)
REAL*8 VLI(NL),VL2(NL),WL(NL,NL),WLI(NL,NL),X(NL,NR),Q(NL,NR),
1 VRI(NR),VR2(NR),WR(NR,NR),WRI(NR,NR)
1 REAL*8 A,B,C,D,K1,K2,K3,K4
DC 20 I=1,NL
DO 20 J=1,NR
X(I,J)=0.
DC 20 II=1,NL
X(I,II)=X(I,J)+WLI(I,II)*Q(II,J)
CC 40 I=1,NL
CC 40 J=1,NR
Q(I,J)=0.
CC 30 JJ=1,NR
Q(I,J)=Q(I,J)+X(I,JJ)*WRI(J,JJ)
CONTINUE
I=1
IF (VL2(I)) 60,110,60
J=1
IF (VR2(J)) 80,90,80
A=VLI(I)+VRI(J)
B=-2.*VL2(I)*VR2(J)
C=A**2+VL2(I)**2+VR2(J)**2
C=C**2-B**2
K1=A*C/D
K2=- (VR2(J)*C+VL2(I)*B)/D
K3=- (VR2(J)*B+VL2(I)*C)/D
I=I+1
JI=J+1
X(I,J)=+K1*C(I,J)+K2*Q(I,J)+K3*Q(I,J)+K4*Q(I,J,J1)
X(I,J1)=-K2*Q(I,J)+K1*Q(I,J1)-K4*Q(I,J,J1)
X(I,J1,J)=+K3*Q(I,J1)-K4*Q(I,J,J1)+K1*Q(I,J,J1)
X(I,J1,J1)=+K4*Q(I,J1)-K3*Q(I,J,J1)+K1*Q(I,J,J1)
J=J+2
GC TO 100
A=VRI(J)+VLI(I)
B=A**2+VL2(I)**2

```

```

100 K1=A/B
    K2=VL2(I)/E
    X(I,J)=K1*C(I,J)-K2*C(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

110 J=1
120 IF (VR2(J)) 120,140,130
130 A=VR1(J)+VL1(I)
    B=A**2+VR2(J)**2
    K1=A/B
    K2=VR2(J)/E
    X(I,J)=K1*C(I,J)-K2*C(I,J+1)
    X(I,J+1)=K2*Q(I,J)+K1*Q(I,J+1)
    J=J+2
    GO TO 150

140 X(I,J)=Q(I,J)/(VR1(J)+VL1(I))
    J=J+1
    IF (J.LE.NR) GO TO 120

150 I=I+1
    IF (I.LE.NL) GC TO 50
    DO 170 I=1,NL
    DO 170 J=1,NR
    Q(I,J)=0.
    DO 170 I=1,NL
    DO 170 J=1,NR
    Q(I,J)=Q(I,J)+WL(I,I)*X(I,J)
    DO 180 JJ=1,NR
    DO 180 JJ=1,NR
    X(I,J)=X(I,J)+Q(I,JJ)*WR(J,JJ)
    CCNT=CCNT+1
    RETURN
END
C=====
C SUBROUTINE MCDE (WNORM,G,GNORM,NS,N1,N2,ICCN)
C
C WNORM TRANSFORMATION MATRIX U OR U-INV
C NS NO. OF STATES
C NC NO. OF INPUTS CP OUTPUTS
C ICCN CONTROL FLAG TO INDICATE WHICH TRANSFORMATION
C 0 = MCEAL GAMMA
C 1 = MCEAL H
C 2 = MCEAL C
C 3 = MCEAL K
C 4 = MCEAL K

```

```

OPTI 17290
OPTI 17300
OPTI 17310
OPTI 17320
OPTI 17330
OPTI 17340
OPTI 17350
OPTI 17360
OPTI 17370
OPTI 17380
OPTI 17390
OPTI 17400
OPTI 17410
OPTI 17420
OPTI 17430
OPTI 17440
OPTI 17450
OPTI 17460
OPTI 17470
OPTI 17480
OPTI 17490
OPTI 17500
OPTI 17510
OPTI 17520
OPTI 17530
OPTI 17540
OPTI 17550
OPTI 17560
OPTI 17570
OPTI 17580
OPTI 17590
OPTI 17600
OPTI 17610
OPTI 17620
OPTI 17630
OPTI 17640
OPTI 17650
OPTI 17660
OPTI 17670
OPTI 17680
OPTI 17690
OPTI 17700
OPTI 17710
OPTI 17720
OPTI 17730
OPTI 17740
OPTI 17750
OPTI 17760

```



```

C=====
END
SUBROUTINE CNORM (WZ,WY,VEC,NS,IWRITE,NSQ,CDD,D1,D2,WNORM,WNCPMI,H
10,CM,N1,N2)
C-----
      WZ(I)      REAL PART OF I-TH EIGENVALUE
      WY(I)      COMPLEX PART OF I-TH EIGENVALUE
      VEC        MATRIX OF RIGHT EIGENVECTORS STORED IN REAL FORM
      NS         FROM PQR2
      NS         NO. OF STATES
      IWRITE     FLAG TO CONTROL FORMATS FOR DIFFERENT EIGENSYSTEMS
      WNORM      NORMALIZED MATRIX U OF RIGHT EIGENVECTORS STORED
      WNORMI     BY COLUMNS IN REAL FORM
      NSQ,DDC,D1,D2 - ARGUMENTS PASSED TO MINV
C=====
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 FIELDC,COMMA,SEMCOL,RIGHT,FMT
      DIMENSION WZ(2*NS),WY(NS),VEC(NS,NS),WNORM(NS,NS),WNORMI(NS,NS),STOR
1E(6),D1(NS),D2(NS),FMT(14),HO(N1,N2),CM(N1,N2)
      DATA FIELDC/5E12.5/,COMMA/5H,/,SEMCOL/5H,/,RIGHT/1H/,FMT/
16H(1X,1P,13#1H/,SEMCOL/4H,/,/
C-----NORMALIZE COMPLEX EIGENVECTORS BY LARGEST ELEMENT-----
      KK=0
      LR=0
      LC=0
      DO 50 K=1,NS
      IF (DABS(WY(K)).LT.1.0D-10) GO TO 50
      LC=LC+1
      FMAX=0
      DO 20 I=1,NS
      CMOD=VEC(I,K)**2+VEC(I,K+1)**2
      IF (CMOD-FMAX) 20,10,10
      FMAX=CMOD
      N=I
      GO TO 10
      10
      20
      CCNT INUE
      VMR=VEC(M,K)
      VMI=VEC(M,K+1)
      DO 30 I=1,NS
      VR=VEC(I,K)
      VI=VEC(I,K+1)
      VECRN=(VR*VMR+VI*VMI)/FMAX
      OPT118250
      OPT118260
      OPT118270
      OPT118280
      OPT118290
      OPT118300
      OPT118310
      OPT118320
      OPT118330
      OPT118340
      OPT118350
      OPT118360
      OPT118370
      OPT118380
      OPT118390
      OPT118400
      OPT118410
      OPT118420
      OPT118430
      OPT118440
      OPT118450
      OPT118460
      OPT118470
      OPT118480
      OPT118490
      OPT118500
      OPT118510
      OPT118520
      OPT118530
      OPT118540
      OPT118550
      OPT118560
      OPT118570
      OPT118580
      OPT118590
      OPT118600
      OPT118610
      OPT118620
      OPT118630
      OPT118640
      OPT118650
      OPT118660
      OPT118670
      OPT118680
      OPT118690
      OPT118700
      OPT118710
      OPT118720

```

```

      3C      VECIN=(-VR*VMI+VI*VMP)/EMAX
      4C      WNCNM(I,K)=VECRN
      5C      WNCNM(I,K+1)=VECIN
      6C      CCNT INUE
      7C      KK=1
      8C      GC TO 50
      9C      KK=0
      AC      CCNT INUE
      BC      NORMALIZE REAL EIGENVECTORS BY THE TOTAL LENGTH-----
      CC      DO 80 K=1,N
      DC      IF (DABS(WY(K)).GE.1.D-10) GO TO 80
      EC      LR=LR+1
      FC      REMOD=0
      GC      DO 60 I=1,N
      HC      REMOD=VEC(I,K)**2+REM0D
      IC      REMOD=DSQRT(REMOD)
      JC      DO 70 I=1,N
      KC      RVEC=VEC(I,K)/REM0D
      LC      WNCNM(I,K)=RVEC
      MC      CCNT INUE
      NC      CCNT INUE
      OC      GC TO (90,100,110,120,130), IWRITE
      PC      WRITE (6,32C)
      QC      GO TO 140
      RC      WRITE (6,33C)
      SC      GO TO 140
      TC      WRITE (6,34C)
      UC      GO TO 140
      VC      WRITE (6,35C)
      WC      GC TO 140
      XC      WRITE (6,36C)
      YC      KK=0
      ZC      NPRTW=0
      10C      NFMTW=1
      11C      IF (KK.EQ.1) GO TO 170
      12C      IF (DABS(WY(I)).GT.1.D-10) KK=1
      13C      PRINT OUT NC MORE THAN 6 WORDS, NOT SEPARATING CCMPLEX EIGVAL
      14C      IF (NPRTW.LT.5) OR (NPRTW.EQ.5) AND (KK.EQ.0) GC TO 150
      15C      FMT(NFMTW+1)=F1GHT
      16C      WRITE (6,FMT) (STORE(J),J=1,NPRTW)
      17C      NPRTW=0
      18C      NFMTW=1
      19C      NPRTW=NPRTW+1
      20C      NFMTW=NFMTW+1
      21C      IF (KK.EC.1) GO TO 160
      22C      STORE(NPRTW)=WZ(I)
      23C      FMT(NFMTW)=FIELD
      24C
      25C
      26C
      27C
      28C
      29C
      30C
      31C
      32C
      33C
      34C
      35C
      36C
      37C
      38C
      39C
      40C
      41C
      42C
      43C
      44C
      45C
      46C
      47C
      48C
      49C
      50C
      51C
      52C
      53C
      54C
      55C
      56C
      57C
      58C
      59C
      60C
      61C
      62C
      63C
      64C
      65C
      66C
      67C
      68C
      69C
      70C
      71C
      72C
      73C
      74C
      75C
      76C
      77C
      78C
      79C
      80C
      81C
      82C
      83C
      84C
      85C
      86C
      87C
      88C
      89C
      90C
      91C
      92C
      93C
      94C
      95C
      96C
      97C
      98C
      99C
      100C
      101C
      102C
      103C
      104C
      105C
      106C
      107C
      108C
      109C
      110C
      111C
      112C
      113C
      114C
      115C
      116C
      117C
      118C
      119C
      120C
      121C
      122C
      123C
      124C
      125C
      126C
      127C
      128C
      129C
      130C
      131C
      132C
      133C
      134C
      135C
      136C
      137C
      138C
      139C
      140C
      141C
      142C
      143C
      144C
      145C
      146C
      147C
      148C
      149C
      150C
      151C
      152C
      153C
      154C
      155C
      156C
      157C
      158C
      159C
      160C
      161C
      162C
      163C
      164C
      165C
      166C
      167C
      168C
      169C
      170C
      171C
      172C
      173C
      174C
      175C
      176C
      177C
      178C
      179C
      180C
      181C
      182C
      183C
      184C
      185C
      186C
      187C
      188C
      189C
      190C
      191C
      192C
      193C
      194C
      195C
      196C
      197C
      198C
      199C
      200C

```

```

160      NFMTW=NFMTH+1
      FMT(NFMTH)=SEMCOL
      GO TO 180
      STORE(NPRTW)=WZ(I)
      FMT(NFMTH+1)=FIELD
      FMT(NFMTH+1)=COMMA
      STORE(NPRTW+1)=WY(I)
      FMT(NFMTH+2)=FIELD
      FMT(NFMTH+3)=SEMCOL
      NPRTW=NFMTH+3
      NPRTW=NPRTH+1
      GO TO 180
      KK=0
      CCNT INUE
      FMT(NFMTH)=SEMENTD
      FMT(NFMTH+1)=RIGHT
      WRITE(6,FMT) (STORE(J) J=1,NPRTW)
      IF (IWRITE.EQ.1) GO TO 190
      WRITE(6,37C)
      GO TO 200
      WRITE(6,38C)
      CALL RAPRNT (NS,NS,NS,6,WNORM,4,(6(IX,1PC13.6)))
      GO TO 210
      CALL MCDE (WNORM,HD,CM,NS,N1,N2,5)
      GO TO 230
      CALL MCDE (WNORM,HD,CM,NS,N1,N2,6)
      GO TO 240
      WRITE(6,39C)
      GO TO 290
      WRITE(6,40C)
      GO TO 290
      WRITE(6,41C)
      GO TO 290
      WRITE(6,420)
      GO TO 290
      WRITE(6,43C)
      -----SAVE U-INVERSE OPEN LOOP IN WNORMI-----
25C      IF (IWRITE.GT.1) GC TC 310
      DO 300 I=1,NS
      DO 300 J=1,NS
      WNORMI(I,J)=WNORM(I,J)
      CALL MINV (NSQ,WNORMI,NS,DDD,D1,D2)
      CALL RAPRNT (NS,NS,NS,6,WNORMI,4,(6(IX,1PC13.6)))
      RETURN
21C      CALL MINV (NSQ,WNORM,NS,DDD,D1,D2)
      CALL PAPRNT (NS,NS,NS,6,WNORM,4,(6(IX,1PC13.6)))
      RETURN
      -----

```

```

OPTI 19210
OPTI 19220
OPTI 19230
OPTI 19240
OPTI 19250
OPTI 19260
OPTI 19270
OPTI 19280
OPTI 19290
OPTI 19300
OPTI 19310
OPTI 19320
OPTI 19330
OPTI 19340
OPTI 19350
OPTI 19360
OPTI 19370
OPTI 19380
OPTI 19390
OPTI 19400
OPTI 19410
OPTI 19420
OPTI 19430
OPTI 19440
OPTI 19450
OPTI 19460
OPTI 19470
OPTI 19480
OPTI 19490
OPTI 19500
OPTI 19510
OPTI 19520
OPTI 19530
OPTI 19540
OPTI 19550
OPTI 19560
OPTI 19570
OPTI 19580
OPTI 19590
OPTI 19600
OPTI 19610
OPTI 19620
OPTI 19630
OPTI 19640
OPTI 19650
OPTI 19660
OPTI 19670
OPTI 19680

```



```

C=====
END
SUBROUTINE POLES (N,NM,A,AA,M,B,L,C,EVR,EVI,D1,D2,JCF,SC)
IMPLICIT REAL*8(A-H,C-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,AA,LCM,IHIGH,D1)
CALL ORTHES (NM,N,LCM,IHIGH,AA,D2,SC)
CALL ORTRAN (NM,N,LCM,IHIGH,AA,D2,SC)
CALL HQR2 (NM,N,LOW,IHIGH,AA,EVR,EVI,SC,IERR)
IF (IERR.NE.0) GO TO 30
CALL BALBAK (NM,N,LCM,IHIGH,D1,N,SC)
WRITE (6,4C)
DO 20 I=1,N
C-----WRITE (4,2CC) EVR(I),EVI(I)-----OPTGRM DATA-----
C-----
C-----
2CWRITE (6,5C) EVR(I),EVI(I)
C-----
3CRETURN
C-----
4CFORMAT (//,28H TF DENOMINATOR EIGENVALUES:,//)
5CFORMAT (//,2X,2H (,F13.6,4H)+J(F13.6,1H))
6CFORMAT (35F,FAILURE IN HQR2, CALCULATING PCLES)
C-----OPTGRM DATA-----
2000FORMAT(5X,2(30.14)
C-----
C-----
END
SUBROUTINE ZEROS (K1,K2,IFD,F,N,NM,A,AA,M,E,L,C,D,BB,CC,CP,EVR,EVI
1,D1,D2,EPS,ITFX)
IMPLICIT REAL*8(A-H,C-Z)
DIMENSION A(N,N),AA(N,N),B(N,M),C(L,N),D(L,M),BE(N),CC(N),CP(N),EVR
1E(N),EVI(N),D1(N),D2(N)
DOUBLE PRECISION SCL,DABS
C-----OPGRAPH DATA-----
IF (ITFX.EQ.1) IFL = 10
IF (ITFX.EQ.2) IFL = 1
IF (ITFX.EQ.3) IFL = 4
C-----
DO 10 I=1,N
BB(I)=B(I,K1)
CC(I)=C(K2,I)
DO 10 J=1,N

```

```

1C  AA(I,J)=A(I,J)
    WRITE (6,9C) K1,K2
    IF (IFCFW.EQ.0) GO TO 20
    H=D(K2,K1)
    IF (DABS(H).LE.EPS) GO TO 20
    JJ=N
    GO TO 50
2C  AN=N-1
    DO 30 I=1,N
    H=SCL(N,BB,CC)
    CALL CCOMP(N,NM,AA,CC,CP)
    IF (DABS(H).GT.EPS) GO TO 40
    CONTINUE
    H=SCL(N,BB,CC)
    WRITE (6,10C) H
    KK=ITFX
    WRITE (IFL,300) KK,K2,K1
    ORDER = C.O
    WRITE (IFL,301) CRDER,H
3C  GO TO 70
4C  JJ=N-1
5C  WRITE (6,11C) JJ,H
    KK=ITFX
    WRITE (IFL,300) KK,K2,K1
    ORDER = FLOAT(JJ)
    WRITE (IFL,301) CRDER,H
6C  CALL ACOMP(N,NM,AA,BB,CC,H)
    CALL BALANC(NM,N,AA,LOW,HIGH,D1)
    CALL ORTHES(NM,N,LOW,HIGH,AA,D2)
    CALL HQR(N,N,LOW,HIGH,AA,EVR,EVI,IERR)
    IF (IERR.NE.0) GO TO 80
    WRITE (6,12C)
    DC 60 I=1,N
    WRITE (6,13C) EVR(I),EVI(I)
7C  WRITE (IFL,302) K2,K1
    DC 53 LL=1,N
    WRITE (IFL,301) EVR(LL),EVI(LL)
8C  RETURN
    WRITE (5,14C)
    RETURN
9C  FORMAT (///,17H TF FOR INPUT NO.,I3,15H AND CPUT NO.,I3,1H:)

```

```

100 FORMAT (//,5X,27HNO FINITE ZEROS, TF GAIN =,E12.4)
110 FORMAT (//,3X,20HORDER OF NUMERATOR =,I3,9X,9H TF GAIN =,E12.4)
120 FORMAT (//,3X,57HNUMERATOR EIGENVALUES (INCLUDING EXTRANEOUS ZERO V
121 VALUES) :
130 FCRMAT (//,4X,1H(,F13.6,4H)+J(,F13.6,1H))
140 FCRMAT (52H FAILURE IN HQR CALCULATING TRANSFER FUNCTION ZERCE)
C-----OPGRAPH DATA-----
300 FCRMAT(5X,215)
301 FCRMAT(5X,2130.14)
302 FCRMAT(5X,215)
C-----
C=====
SUBROUTINE ACOMP (N,NM,A,B,C,T)
REAL*8 A,B,C,T
DIMENSION A(NM,N),B(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
END
C=====
SUBROUTINE CCOMP (N,NM,A,C,CC)
REAL*8 A,C,CC
DIMENSION A(NM,N),C(N),CC(N)
DO 10 I=1,N
CC(I)=0
DO 10 J=1,N
CC(I)=CC(I)+C(J)*A(J,I)
10 CC(I)=CC(I)
20 RETURN
END
C=====
FUNCTION SCL (N,B,C)
REAL*8 B,C,SCL
DIMENSION B(N),C(N)
SCL=0
DO 10 I=1,N
SCL=SCL+C(I)*B(I)
10 RETURN
END
C=====
SUBROUTINE RESID (K1,K2,N,JCF,M,BM,L,CM,PR,PI,RES,BB,CC,IPT)
IMPLICIT REAL*8(A-H,I-Z)
DIMENSION JCF(N),BM(N,M),CM(L,N),PR(N),PI(N),RES(N),BB(N),CC(N),PRQ
1T(4)
DATA SN/8H*5IN(B*T/,R1/8H */,R2/8HEXF(A*T)/,5D/1H)/

```

```

C-----DATA ZERO/C,DO/,T1/4H*T**/,BLANK/8H-----OPT21610
TEMPORARY MCD TILL JCF IS CALCULATED-----OPT21620
1C-----DC 10 I=1,N-----OPT21630
JCF(I)=0-----OPT21640
C-----TEMPORARY MOD-----OPT21650
IF (IPT,EC,1) WRITE (6,170)OPT21660
DO 20 I=1,NOPT21670
BB(I)=BM(I,K1)OPT21680
CC(I)=CM(K2,I)OPT21690
C-----LOOP THROUGH THE POLES-----OPT21700
I=0OPT21710
I=I+1OPT21720
IF (I,GT,N) GO TO 160OPT21730
IF (JCF(I),EQ,1) GO TO 60OPT21740
IF (DABS(PI(I)),LT,1E-10) GO TO 50OPT21750
C-----COMPUTE SIMPLE COMPLEX POLE RESIDUES AND PRINT BOTH-----OPT21760
RES(I)=CC(I)*BB(I)+CC(I+1)*BB(I+1)OPT21770
IF (IPT,EC,0) GO TC 40OPT21780
PRT(I)=BLANKOPT21790
PRT(2)=R2OPT21800
IF (PI(I),EQ,0.00) PRT(2)=BLANKOPT21810
PRT(3)=CSOPT21820
PRT(4)=EDOPT21830
WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)OPT21840
I=I+1OPT21850
PRT(3)=SNOPT21860
WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)OPT21870
GO TO 30OPT21880
I=I+1OPT21890
GO TO 30OPT21900
C-----COMPUTE SIMPLE REAL POLE RESIDUE-----OPT21910
RES(I)=CC(I)*BB(I)OPT21920
IF (IPT,EC,0) GO TC 30OPT21930
PRT(1)=R1OPT21940
PRT(2)=R2OPT21950
PRT(3)=BLANKOPT21960
PRT(4)=BLANKOPT21970
WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)OPT21980
GO TO 30OPT21990
C-----LOOK AHEAD TO DETERMINE SIZE OF THE JCRCAN BLOCK-----OPT22000
K=1OPT22010
KT=N-I+1OPT22020
DO 70 J=I,KTOPT22030
IF (JCF(J),EQ,0) GO TO 80OPT22040
K=K+1OPT22050
C-----CCCONTINUEOPT22060
OPT22070
OPT22080

```

```

C-----
IF (DABS(PI(I)).LT. 1.D-10) GO TO 110
K=1
DO 120 J=1,K
  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+2)
  RES(I+2)=CC(I)*BB(I+3)+CC(I+1)*BB(I+2)
  RES(I+3)=CC(I)*BB(I+2)-CC(I+1)*BB(I+3)
  IF (IPT.EQ. 0) GO TO 100
  PRT(1)=R1
  PRT(2)=R2
  IF (DABS(PR(I)).GT. 1.D-10) GO TO 90
  PRT(1)=BLANK
  PRT(2)=BLANK
  PRT(3)=CS
  PRT(4)=ED
  WRITE(6,18C) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
  I=I+1
  WRITE(6,18C) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
  PRT(1)=T1
  PRT(2)=R2
  IF (DABS(PR(I)).LT. 1.D-10) PRT(2)=BLANK
  PRT(3)=CS
  I=I+1
  WRITE(6,15C) PR(I),PI(I),RES(I),PRT(1),K,(PRT(J),J=2,4)
  PRT(3)=SN
  I=I+1
  WRITE(6,15C) PR(I),PI(I),RES(I),PRT(1),K,(PRT(J),J=2,4)
  GO TO 30
  I=I+3
  GO TO 30
C-----
110 COMPUTE REPEATED REAL PCLE RESIDUE AND PRINT CUT ALL K OF THEM-----
CONTINUE
KT=I+K-1
NN=0
DO 130 J=I,KT
  NN=NN+1
  RES(J)=ZERC
  GO 120 JJ=J,KT
  RES(J)=RES(J)+BB(J)*CC(JJ-NN+1)
CONTINUE
IF (IPT.EQ. 0) GO TO 150
NN=0
PRT(1)=T1
PRT(2)=R2
PRT(3)=BLANK
PRT(4)=BLANK

```

```

OPT222090
OPT222100
OPT222110
OPT222120
OPT222130
OPT222140
OPT222150
OPT222160
OPT222170
OPT222180
OPT222190
OPT222200
OPT222210
OPT222220
OPT222230
OPT222240
OPT222250
OPT222260
OPT222270
OPT222280
OPT222290
OPT222300
OPT222310
OPT222320
OPT222330
OPT222340
OPT222350
OPT222360
OPT222370
OPT222380
OPT222390
OPT222400
OPT222410
OPT222420
OPT222430
OPT222440
OPT222450
OPT222460
OPT222470
OPT222480
OPT222490
OPT222500
OPT222510
OPT222520
OPT222530
OPT222540
OPT222550
OPT222560

```

```

DC 140 J=I,KT
WRITE (6,I50) PR(J),PI(J),RES(J),PRT(1),NN,(PRT(JJ),JJ=2,4)
NN=NN+1
GO TO 30
150 I=KT
GO TO 30
160 CONTINUE
RETURN
C-----
170 FORMAT (//,2X,22RESIDUES AT THE POLES:/,T16,5HF 0 L E S,T41,15HR
1E S I C U E S//T9,7HR REAL(A),T26,7HIMAG(B))
180 FORMAT (//,4X,1H(,F13.6,4H)+J(,F13.6,1H),4X,1H(,F13.6,1H),3A8,A1)
190 FORMAT (//,4X,1H(,F13.6,4H)+J(,F13.6,1H),4X,1H(,F13.6,1H),A4,I2,2X,
12A8,A1)
END
C=====
SUBROUTINE BALANC (NM,N,A,LOW,IGH,SCALE)
INTEGER I,J,K,L,M,N,JJ,NM,IGH,LOW,IEXC
REAL*8 A(NM,N),SCALE(N)
REAL*8 C,F,G,R,S,B2,RADIX
LOGICAL NOCCNV
DATA RADIX/2421000CCCC000000/
C-----
B2=RADIX*RADIX
K=1
L=N
GO TO 60
C-----
1C SCALE(M)=J
IF (J.EQ.0) 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(I,I)
A(J,I)=A(M,I)
A(M,I)=F
CONTINUE
GO TO (50,5C),IEXC
40 SEARCH FOR FOWS ISOLATING AN EIGENVALUE AND PUSH THEM DOWN-----
5C IF (L.EC.1) GO TO 230
L=L-1
DO 80 JJ=1,L
J=L+1-JJ
DO 70 I=1,L

```

```

7C      IF (I.EQ. J) GO TO 70
        IF (A(J,I).NE. 0.0D0) GO TO 80
        CONTINUE
        M=L
        IEXC=1
        GO TO 10
8C      CCNTINUE
        GO TO 100
        CCNTINUE
        SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE AND PUSH THEM LEFT
        K=K+1
        DO 120 J=K,L
          DO 110 I=K,L
            IF (I.EQ. J) GO TO 110
            IF (A(I,J).NE. 0.0D0) GO TO 120
          CCNTINUE
          M=K
          IEXC=2
          GO TO 10
12C     CONTINUE
        NOW BALANCE THE SUBMATRIX IN ROWS K TO L
        DO 130 I=K,L
          SCALE(I)=1.0D0
13C     CCNTINUE
        ITERATIVE LOOP FOR NORM REDUCTION
        NOCONV=.FALSE.
        DO 220 I=K,L
          C=0.0D0
          R=0.0D0
          DO 150 J=K,L
            IF (J.EQ. I) GO TO 150
            C=C+DABS(A(J,I))
            R=R+DABS(A(I,J))
          CCNTINUE
          GUARD AGAINST ZERO C OR R DUE TO UNDERFLOW
          IF (C.EQ. 0.0D0 .OR. R.EQ. 0.0D0) GO TO 220
          G=P/RADIX
          F=1.0D0
          S=C+R
          IF (C.GE. G) GO TO 170
          F=F*RADIX
          C=C*B2
          GO TO 160
17C     G=R*RADIX
18C     IF (C.LT. G) GO TO 190
          F=F/RADIX
          C=C/B2
          GO TO 180
        NOW BALANCE
        IF ((C + R) / F .GE. 0.95D0 / S) GO TO 220
19C     CONTINUE

```

```

G=1.0D0/F
SCALE(I)=SCALE(I)*F
NCONV=.TRUE.
DO 200 J=K,N
  A(I,J)=A(I,J)*G
DO 210 J=1,I
  A(J,I)=A(J,I)*F
CONTINUE
IF (NCONV) GO TO 140
LCW=K
IGH=L
RETURN
END
=====
SUBROUTINE CRTHES (NM,N,LOW,IGH,A,ORT)
  INTEGER I,J,M,N,I1,JJ,LA,MP,NM,IGH,KP1,LCW
  REAL*8 A(NM,N),CRT(IGH)
  REAL*8 F,G,F,SCALE
  REAL*8 DSQRT,CABS,DSIGN
  LA=IGH-1
  KP1=LOW+1
  IF (LA.LT. KP1) GO TO 100
  DO 90 M=KP1,LA
    H=0.0D0
    CRT(M)=0.0D0
    SCALE=0.0D0
    SCALE COLUMN (ALGCL TOL THEN NOT NEEDED)
    DO 10 I=M,IGH
      SCALE=SCALE+CABS(A(I,M-1))
      MP=M+IGH
      DO 20 II=M,IGH
        I=MP-II
        CRT(I)=A(I,M-1)/SCALE
        H=H+ORT(I)*CRT(I)
      CONTINUE
    G=-DSIGN(DSQRT(H),CRT(M))
    H=H-ORT(M)*G
    CRT(M)=ORT(M)-G
    -----FORM (I-(U*UT)/H) * A-----
    DO 50 J=M,N
      F=0.0D0
      DO 30 II=M,IGH
        I=MP-II
        F=F+ORT(I)*A(I,J)
      CONTINUE
      F=F/H
      DO 40 I=M,IGH

```

```

4C      A(I,J)=A(I,J)-F*ORT(I)
5C      CONTINUE
C-----FORM (I-(U*UT)/H)*A*(I-(U*UT)/H)-----
DO 80 I=1,IGH
F=0.0DC
DO 60 JJ=M,IGH
J=MP-JJ
F=F+ORT(J)*A(I,J)
CONTINUE
F=F/H
DO 70 J=M,IGH
A(I,J)=A(I,J)-F*ORT(J)
CONTINUE
CRT(M)=SCALE*ORT(M)
A(M,M-1)=SCALE*G
CONTINUE
RETURN
END
C=====
SUBROUTINE CRTRAN (NM,N,LOW,IGH,A,ORT,Z)
INTEGER I,J,N,KL,MM,MP,NM,IGH,LOW,MP1
REAL*8 A(NM,IGH),ORT(IGH),Z(NM,N)
REAL*8 G
C-----INITIALIZE Z TO IDENTITY MATRIX-----
DO 20 I=1,N
DO 10 J=1,N
Z(I,J)=0.0DC
Z(I,I)=1.0DC
CONTINUE
KL=IGH-LOW-1
IF (KL.LT.1) GO TO 80
DO 70 MM=1,KL
MP=IGH-MM
MP1=MP+1
IF (A(MP,MP-1) .EQ. 0.0DC) GO TO 70
MP1=MP+1
DO 30 I=MP1,IGH
CRT(I)=A(I,MP-1)
DO 60 J=MP,IGH
G=C*JDJ
DO 40 I=MP,IGH
G=G+ORT(I)*Z(I,J)
CONTINUE
G=(G / ORT(MP))/A(MP,MP-1)
Z(I,J)=Z(I,J)+G*ORT(I)
CONTINUE
DO 50 I=MP,IGH
CONTINUE
CONTINUE
7C
8C
9C
10C
C=====
SUBROUTINE CRTRAN (NM,N,LOW,IGH,A,ORT,Z)
INTEGER I,J,N,KL,MM,MP,NM,IGH,LOW,MP1
REAL*8 A(NM,IGH),ORT(IGH),Z(NM,N)
REAL*8 G
C-----INITIALIZE Z TO IDENTITY MATRIX-----
DO 20 I=1,N
DO 10 J=1,N
Z(I,J)=0.0DC
Z(I,I)=1.0DC
CONTINUE
KL=IGH-LOW-1
IF (KL.LT.1) GO TO 80
DO 70 MM=1,KL
MP=IGH-MM
MP1=MP+1
IF (A(MP,MP-1) .EQ. 0.0DC) GO TO 70
MP1=MP+1
DO 30 I=MP1,IGH
CRT(I)=A(I,MP-1)
DO 60 J=MP,IGH
G=C*JDJ
DO 40 I=MP,IGH
G=G+ORT(I)*Z(I,J)
CONTINUE
G=(G / ORT(MP))/A(MP,MP-1)
Z(I,J)=Z(I,J)+G*ORT(I)
CONTINUE
DO 50 I=MP,IGH
CONTINUE
CONTINUE
5C
6C
7C

```

```

8C      RETURN
C=====
      SUBROUTINE FOR2 (NM,N,LCW,IGH,H,WR,WI,Z,IERR)
      INTEGER I,J,K,L,M,N,EN,II,JJ,LL,MM,NA,NN,IGH,ITS,LOW,MP2,ENM2,
1EPR
      REAL*8 H(NM,N),WR(N),WI(N),Z(NM,N)
      REAL*8 P,Q,R,S,T,W,X,Y,RA,SA,VI,VR,ZZ,NORM,MACHEP
      INTEGER MINC
      LOGICAL NOTLAS
      COMPLEX*16Z3
      COMPLEX*16CCMPLX
      REAL*8 DREAL,DIMAG
      C----- STATEMENT CF DOUBLE PRECISION COMPLEX NUMBERS-----
      C----- PARTS CF DOUBLE PRECISION COMPLEX NUMBERS-----
      DREAL(Z3)=Z3
      DIMAG(Z3)=(C.ODD,-1.CDD)*Z3
      DATA MACHEP/2341000000000000000000/
      IERR=0
      NORM=0.ODD
      K=1
C----- STORE ROOTS ISOLATED BY BALANC AND COMPUTE MATRIX NORM-----
      DO 20 I=1,N
      DO 10 J=K,N
      NORM=NORM+ABS(H(I,J))
      K=I
      IF (I.GE. LOW .AND. I.LE. IGH) GO TO 20
      WR(I)=H(I,I)
      WI(I)=0.ODD
      CCCONTINUE
      EN=IGH
      T=C.ODD
C----- SEARCH FOR NEXT EIGENVALUES-----
      IF (EN.LT. LOW) GO TO 290
      ITS=0
      NA=EN-1
      ENM2=NA-1
C----- LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT-----
      DO 50 LL=LCH,EN
      L=EN+LCW-LL
      IF (LL.EQ. LOW) GO TO 60
      S=DABS(H(L-1,L-1))+DABS(H(L,L))
      IF (S.EQ. C.ODD) S=NORM
      IF (DABS(H(L,L-1)).LE. MACHEP * S) GO TO 60
      CCCONTINUE
      X=H(EN,EN)
      C----- FORM SHIFT-----

```

```

IF (L.EQ. EN) GO TC 220
Y=H(NA,NA)
W=H(EN,NA)*F(NA,EN)
IF (L.EQ. NA) GO TC 230
IF (ITS.EC. 30) GO TO 500
IF (ITS.NE. 10) AND ITS.NE. 20) GO TO 80
C-----FORM EXCEPTIONAL SHIFT-----
T=T+X
DO 70 I=LOW,EN
H(I,I)=H(I,I)-X
S=DABS(H(EN,NA))+DABS(H(NA,ENM2))
X=0.75DO*S
Y=X
W=-0.4375DC*S*S
ITS=ITS+1
C-----LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS.-----
DO 90 MM=L,ENM2
M=ENM2+L-MM
ZZ=H(M,M)
R=X-ZZ
S=Y-ZZ
P=(R#S-W)/H(M+1,M)+H(M,M+1)
Q=H(M+1,M+1)-ZZ-R-S
R=H(M+2,M+1)
S=DABS(P)+DABS(Q)+DABS(R)
P=P/S
Q=Q/S
R=R/S
IF (M.EQ. 1) 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
SC
100
C-----CONTINUE
MP2=M+2
DO 110 I=MF2,EN
P(I,I-2)=0.CD0
IF (I.EQ. MP2) GO TC 110
H(I,I-3)=0.CD0
C-----CONTINUE
110
C-----DOUBLE CR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN-----
DO 210 K=M,NA
NOTLAS=K.NE.NA
IF (K.EQ. N) GO TC 120
P=H(K,K-1)
Q=H(K+1,K-1)
R=0.CD0
IF (NOTLAS) R=H(K+2,K-1)
X=DABS(P)+DABS(Q)+DABS(R)
IF (X.EQ. 0.CD0) GO TO 210

```

OPT24970
OPT24980
OPT24990
OPT25000
OPT25010
OPT25020
OPT25030
OPT25040
OPT25050
OPT25060
OPT25070
OPT25080
OPT25090
OPT25100
OPT25110
OPT25120
OPT25130
OPT25140
OPT25150
OPT25160
OPT25170
OPT25180
OPT25190
OPT25200
OPT25210
OPT25220
OPT25230
OPT25240
OPT25250
OPT25260
OPT25270
OPT25280
OPT25290
OPT25300
OPT25310
OPT25320
OPT25330
OPT25340
OPT25350
OPT25360
OPT25370
OPT25380
OPT25390
OPT25400
OPT25410
OPT25420
OPT25430
OPT25440

```

P=P/X
Q=Q/X
R=R/X
S=DSIGN(DSCRT(P*P+Q*(R*R),P)
IF (K.EC.N) GC TC 130
H(K,K-1)=-S*X
GC TO 140
IF (L.NE.N) H(K,K-1)=-H(K,K-1)
P=P+S
X=P/S
Y=Q/S
ZZ=R/S
Q=Q/P
R=R/P
C-----ROW MODIFICATION-----
DO 160 J=K,N
P=H(K,J)+Q*(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
CCNT INUE
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. NCTLAS) GO TO 170
P=P+ZZ*H(I,K+2)
H(I,K+2)=H(I,K+2)-P*R
H(I,K+1)=H(I,K+1)-P*C
H(I,K)=H(I,K)-P
CCNT INUE
C-----ACCUMULATE TRANSFORMATIONS-----
DO 200 I=LCH,IGH
P=X*Z(I,K)+Y*Z(I,K+1)
IF (.NOT. NCTLAS) GC 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*C
Z(I,K)=Z(I,K)-P
CCNT INUE
GO TO 40
C-----ONE ROOT FOUND-----
220 H(EN,EN)=X+7
WR(EN)=H(EN,EN)
WI(EN)=2.0CC

```

```

EN=NA
GO TO 30
C-----TWO ROOTS FOUND-----
230  P=(Y - X)/2.0D0
      C=P*P+W
      ZZ=DSQRT(DABS(Q))
      H(EN,EN)=X+Y
      X=H(EN,EN)
      H(NA,NA)=Y+T
      IF (Q .LT. C.0D0) GO TO 270
C-----REAL PAIR-----
      ZZ=P+DSIGN(ZZ,P)
      WR(NA)=X+ZZ
      WR(EN)=WR(NA)
      IF (ZZ .NE. C.0D0) WR(EN)=X-W/ZZ
      WI(NA)=0.0CC
      WI(EN)=0.0CC
      X=H(EN,NA)
      S=DABS(X)+[ABS(ZZ)
      P=X/S
      Q=ZZ/S
      R=DSQRT(P*F+Q*Q)
      P=P/R
      C=Q/R
C-----ROW MODIFICATION-----
      DO 240 J=NA,N
      ZZ=H(NA,J)
      H(NA,J)=Q*ZZ+P*H(EN,J)
      H(EN,J)=Q*F(EN,J)-P*ZZ
      CONTINUE
240  C-----COLUMN MODIFICATION-----
      DO 250 I=1,EN
      ZZ=H(I,NA)
      H(I,NA)=Q*ZZ+P*H(I,EN)
      H(I,EN)=Q*F(I,EN)-P*ZZ
      CONTINUE
250  C-----ACCUMULATE TRANSFORMATIONS-----
      DO 260 I=LCh,IGH
      ZZ=Z(I,NA)
      Z(I,NA)=Q*ZZ+P*Z(I,EN)
      Z(I,EN)=Q*Z(I,EN)-P*ZZ
      CONTINUE
      GC TO 280
260  C-----COMPLEX PAIR-----
      WR(NA)=X+P
      WR(EN)=X+P
      WI(NA)=ZZ
      WI(EN)=-ZZ
270

```

OPT25930
OPT25940
OPT25950
OPT25960
OPT25970
OPT25980
OPT25990
OPT26000
OPT26010
OPT26020
OPT26030
OPT26040
OPT26050
OPT26060
OPT26070
OPT26080
OPT26090
OPT26100
OPT26110
OPT26120
OPT26130
OPT26140
OPT26150
OPT26160
OPT26170
OPT26180
OPT26190
OPT26200
OPT26210
OPT26220
OPT26230
OPT26240
OPT26250
OPT26260
OPT26270
OPT26280
OPT26290
OPT26300
OPT26310
OPT26320
OPT26330
OPT26340
OPT26350
OPT26360
OPT26370
OPT26380
OPT26390
OPT26400

```

28C      EN=ENM2
C      GO TO 30
C-----ALL ROOTS FOUND. BACKSUBSTITUTE TO FIND-----
C-----VECTORS OF UPPER TRIANGULAR FORM-----
29C      IF (NORM.EC.C.O.DD) GO TO 510
C      DO 450 NN=1,N
C      P=WR(EN)
C      Q=WI(EN)
C      NA=EN-1
C      IF (Q) 370,300,450
C-----REAL VECTOR-----
30C      M=EN
C      H(EN,EN)=1.CDD
C      IF (NA.EQ.Q) GO TO 450
C      DO 360 II=1,NA
C      I=EN-II
C      W=H(I,I)-P
C      R=H(I,EN)
C      IF (M.GT.NA) GO TO 320
C      DO 310 J=M,NA
C      R=P+H(I,J)*F(J,EN)
31C      IF (WI(I).GE.Q.O.DD) GO TO 330
32C      ZZ=W
C      S=R
C      GO TO 360
C      M=I
C      IF (WI(I).NE.Q.O.DD) GO TO 340
C      T=W
C      IF (W.EQ.C.O.DD) T=MACHEP*NORM
C      H(I,EN)=R/T
C      GO TO 360
C-----SOLVE REAL EQUATIONS-----
33C      X=H(I,I+1)
C      Y=H(I+1,I)
C      C=(WR(I)-P)+WI(I)*WI(I)
C      T=(X*S-ZZ*R)/Q
C      H(I,EN)=T
C      IF (DABS(X).LE.DABS(ZZ)) GO TO 350
C      H(I+1,EN)=(-R-W*T)/X
C      GO TO 360
35C      H(I+1,EN)=(-S-Y*T)/ZZ
36C      CONTINUE
C-----END REAL VECTOR-----
C      GO TO 450
C-----COMPLEX VECTOR-----
37C      M=NA
C-----LAST VECTOR COMPONENT CHOSEN IMAGINARY SC THAT-----
OPT26410
OPT26420
OPT26430
OPT26440
OPT26450
OPT26460
OPT26470
OPT26480
OPT26490
OPT26500
OPT26510
OPT26520
OPT26530
OPT26540
OPT26550
OPT26560
OPT26570
OPT26580
OPT26590
OPT26600
OPT26610
OPT26620
OPT26630
OPT26640
OPT26650
OPT26660
OPT26670
OPT26680
OPT26690
OPT26700
OPT26710
OPT26720
OPT26730
OPT26740
OPT26750
OPT26760
OPT26770
OPT26780
OPT26790
OPT26800
OPT26810
OPT26820
OPT26830
OPT26840
OPT26850
OPT26860
OPT26870
OPT26880

```

```

C-----EIGENVECTR MATRIX IS TRIANGULAR-----
IF (DABS(H(EN,NA))) .LE. DABS(H(NA,EN))) GC TC 380
H(NA,NA)=Q/F(EN,NA)
H(NA,EN)=-(F(EN,EN) - P)/H(EN,NA)
GO TO 390
380 Z3=DCMPLX(C.ODO,-H(NA,EN))/DCMPLX(H(NA,NA)-P,Q)
H(NA,NA)=DREAL(Z3)
H(NA,EN)=DIMAG(Z3)
390 H(EN,NA)=C.ODO
H(EN,EN)=1.CDO
ENM2=NA-1
IF (ENM2 .EQ. 0) GC TO 450
GO 440 I=1,ENM2
I=NA-I
P=H(I,I)-P
RA=O.ODO
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)
CNT INUE
400 IF (WI(I) .GE. O.ODO) GO TO 410
ZZ=W
R=RA
S=SA
GO TO 440
410 M=I
IF (WI(I) .NE. C.ODO) GO TO 420
Z3=DCMPLX(-RA,-SA)/DCMPLX(W,Q)
H(I,NA)=DREAL(Z3)
H(I,EN)=DIMAG(Z3)
GO TO 440
C-----SOLVE COMPLEX EQUATIONS-----
420 X=H(I,I+1)
Y=H(I+1,I)
VR=(WR(I) - P)*(WR(I) - P)+WI(I)*WI(I)-Q*Q
VI=(WR(I) - P)*2.ODO*Q
IF (VR .EQ. O.ODO .AND. VI .EQ. O.ODO) VR=MACFEP*NORM*(DABS(W) +
1ABS(Q) + DABS(X) + DABS(Y) + DABS(Z3))
Z3=DCMPLX(X*VR-Z3*RA+Q*SA,X*S-Z3*SA-Q*RA)/DCMPLX(VR,VI)
H(I,NA)=DREAL(Z3)
H(I,EN)=DIMAG(Z3)
IF (DABS(X) .LE. DABS(Z3) + DABS(Q)) GO TO 430
H(I+1,NA)=(-RA - W * H(I,NA) + Q * H(I,EN))/X
H(I+1,EN)=(-SA - W * H(I,EN) - Q * H(I,NA))/X
GO TO 440
430 Z3=DCMPLX(-R-Y*H(I,NA)-S-Y*H(I,EN))/DCMPLX(Z3,C)
H(I+1,NA)=DREAL(Z3)

```

OPT26890
 OPT26900
 OPT26910
 OPT26920
 OPT26930
 OPT26940
 OPT26950
 OPT26960
 OPT26970
 OPT26980
 OPT26990
 OPT27000
 OPT27010
 OPT27020
 OPT27030
 OPT27040
 OPT27050
 OPT27060
 OPT27070
 OPT27080
 OPT27090
 OPT27100
 OPT27110
 OPT27120
 OPT27130
 OPT27140
 OPT27150
 OPT27160
 OPT27170
 OPT27180
 OPT27190
 OPT27200
 OPT27210
 OPT27220
 OPT27230
 OPT27240
 OPT27250
 OPT27260
 OPT27270
 OPT27280
 OPT27290
 OPT27300
 OPT27310
 OPT27320
 OPT27330
 OPT27340
 OPT27350
 OPT27360

```

440 H(I+1,EN)=C*IMAG(Z3)
C CONTINUE
450 CONTINUE-----END COMPLEX VECTOR-----
C-----
C-----END BACK SUBSTITUTION. VECTORS OF ISOLATED ROOTS-----
DO 470 I=1,N
IF (I.EQ. LOW .AND. I .LE. IGH) GO TO 470
DC 460 J=I,N
Z(I,J)=H(I,J)
460 CONTINUE
470 CONTINUE-----MULTIPLY BY TRANSFORMATION MATRIX TO GIVE-----
C-----VECTORS OF ORIGINAL FULL MATRIX.-----
DO 490 JJ=LCH,N
J=N+LOW-JJ
M=MINO(J,I,IGH)
DC 490 I=LCH,IGH
ZZ=0.000
DO 480 K=LCH,M
ZZ=ZZ+Z(I,K)*H(K,J)
480 Z(I,J)=ZZ
490 CONTINUE
C-----GO TO 510
C-----SET ERROR --->NO CONVERGENCE TO AN-----
C-----EIGENVALUE AFTER 30 ITERATIONS-----
500 IERR=EN
510 RETURN
C-----END-----
C-----
C-----SUBROUTINE BALBAK (N,M,N,LOW,IGH,SCALE,M,Z)
INTEGER I,J,K,M,N,II,NM,IGH,LCW
REAL*8 SCALE(N),Z(NM,M),S
IF (M.EQ.0) GO TO 60
IF (IGH.EQ. LOW) GO TO 30
DC 20 I=LCH,IGH
S=SCALE(I)
C-----LEFT HAND EIGENVECTORS ARE BACK TRANSFORMED-----
C-----IF THE FORGECING STATEMENT IS REPLACED BY-----
C-----S=1.000/SCALE(I).-----
LC 10 J=1,M
Z(I,J)=Z(I,J)*S
C CONTINUE
DC 50 II=1,N
I=II
IF (I.EQ. LOW .AND. I .LE. IGH) GO TO 50
IF (I .LT. LOW) I=LCH-II
K=SCALE(I)
IF (K.EQ.0) GO TO 50
DC 40 J=1,M

```

```

S=Z(I,J)
Z(I,J)=Z(K,J)
Z(K,J)=S
CCCONTINUE
RETURN
END
=====
SUBROUTINE FOR (NM,N,LOW,IGH,H,WR,WI,IERR)
INTEGER I,J,K,L,M,N,EN,LL,MM,NA,NM,IGH,ITS,LCW,MP2,ENM2,IERR
REAL*8 H(NM,N),WR(N),WI(N)
REAL*8 P,Q,F,S,T,W,X,Y,ZZ,NORM,MACHEP
REAL*8 DSQRT,DABS,DSIGN
INTEGER MIND
LOGICAL NOTLAS
DATA MACHEP/Z3410000C00000000/
IERR=0
NORM=0.0D0
K=1
C-----STORE RCCTS ISOLATED BY BALANC AND COMPUTE MATRIX NORM-----
CC 20 I=1,N
CC 30 J=K,N
NORM=NORM+ABS(H(I,J))
K=I
IF (I.GE. LCW .AND. I.LE. IGH) GO TO 20
WR(I)=H(I,I)
WI(I)=0.0D0
CONTINUE
EN=IGH
T=0.0D0
C-----SEARCH FOR NEXT EIGENVALUES-----
IF (EN.LT. LOW) GO TO 250
ITS=0
NA=EN-1
ENM2=NA-1
C-----LOCK FOR SINGLE SMALL SUR-DIAGONAL ELEMENT-----
CC 50 LL=LCW,EN
L=EN+LCW-LL
IF (L.EQ. LCW) GO TO 60
S=DABS(H(L-1,L-1))+DABS(H(L,L))
IF (S.EQ. C.0D0) S=NORM
IF (DABS(H(L,L-1)).LE. MACHEP * S) GO TO 60
CONTINUE
C-----FORM SHIFT-----
X=H(EN,EN)
IF (L.EC. EN) GO TO 200
Y=H(NA,NA)
W=H(EN,NA)*F(NA,EN)

```

```

IF (L .EQ. NA) GO TO 210
IF (ITS .EQ. 30) GO TO 240
IF (ITS .NE. 10 .AND. ITS .NE. 20) GO TO 80
-----FORM EXCEPTIONAL SHIFT-----
T=T+X
DO 70 I=LQ,EN
  F(I,I)=H(I,I)-X
  S=DABS(H(EK,NA))+DABS(H(NA,ENM2))
  X=X+.75DO*S
  Y=X
  W=-O.4375DO*S*S
  ITS=ITS+1
-----LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS.-----
DO 90 MM=L,ENM2
  M=ENM2+L-MM
  ZZ=H(M,M)
  R=X-ZZ
  S=Y-ZZ
  P=(R+M+1,M+1)/H(M+1,M)+H(M,M+1)
  Q=H(M+1,M+1)-ZZ-R-S
  R=H(M+2,M+1)
  S=DABS(P)+DABS(Q)+DABS(R)
  P=P/S
  Q=Q/S
  R=R/S
IF (M .EQ. L) GO TO 100
IF (DABS(H(M,M-1)) * (DABS(Q) + DABS(R)) .LE. MACHEP * DABS(P))
  1 * (DABS(H(M-1,M-1)) + DABS(ZZ) + DABS(H(M+1,M+1))) GO TO 100
CONTINUE
MP2=M+2
DO 110 I=MP2,EN
  F(I,I-2)=O.(DO
  IF (I .EQ. MP2) GO TO 110
  F(I,I-3)=C.(DO
  CONTINUE
OR STEP INVOLVING ROWS L TC EN AND COLUMNS M TO EN-----
DO 190 K=M,NA
  NCTLAS=K.NE.NA
  IF (K .EQ. N) GO TO 120
  P=H(K,K-1)
  Q=H(K+1,K-1)
  R=O.ODO
  IF (NOTLAS) R=H(K+2,K-1)
  X=DABS(P)+DABS(Q)+DABS(R)
  IF (X .EQ. C.OOO) GO TO 190
  P=P/X
  Q=Q/X
  R=R/X

```

OPT28330
OPT28340
OPT28350
OPT28360
OPT28370
OPT28380
OPT28390
OPT28400
OPT28410
OPT28420
OPT28430
OPT28440
OPT28450
OPT28460
OPT28470
OPT28480
OPT28490
OPT28500
OPT28510
OPT28520
OPT28530
OPT28540
OPT28550
OPT28560
OPT28570
OPT28580
OPT28590
OPT28600
OPT28610
OPT28620
OPT28630
OPT28640
OPT28650
OPT28660
OPT28670
OPT28680
OPT28690
OPT28700
OPT28710
OPT28720
OPT28730
OPT28740
OPT28750
OPT28760
OPT28770
OPT28780
OPT28790
OPT28800

```

120 S=DSIGN(DSCRT(P*P+Q*C+R*R),P)
    IF (K.EQ.M) GO TO 130
    H(K,K-1)=-S*X
    GO TO 140
130 IF (L.NE.M) H(K,K-1)=-H(K,K-1)
140 P=P+S
    X=P/S
    Y=Q/S
    ZZ=R/S
    Q=Q/P
    R=R/P
C-----ROW MODIFICATION-----
DO 160 J=K,EN
P=H(K,J)+C*F(K+1,J)
IF (.NOT. NCTLAS) GO TO 150
P=P+R*H(K+2,J)
H(K+2,J)=H(K+2,J)-P*ZZ
H(K+1,J)=F(K+1,J)-P*Y
H(K,J)=H(K,J)-P*X
CCNT INUE
J=MINO(EN,K+3)
C-----COLUMN MODIFICATION-----
DO 180 I=L,J
P=X*H(I,K)+Y*H(I,K+1)
IF (.NOT. NCTLAS) GO TO 170
P=P+ZZ*H(I,K+2)
H(I,K+2)=H(I,K+2)-P*R
H(I,K+1)=H(I,K+1)-P*C
H(I,K)=H(I,K)-P
CCNT INUE
GO TO 40
170 WR(EN)=X+T
180 WI(EN)=O.OCC
190 EN=NA
    GO TO 30
C-----ONE ROOT FOUND-----
200 P=(Y - X)/2.OOO
    C=P*P+Q
    ZZ=DSQRT(DABS(Q))
    X=X+T
    IF (Q.LT.C.OOO) GO TO 220
C-----TWO ROOTS FOUND-----
210 ZZ=P+DSIGN(ZZ,P)
    WR(NA)=X+ZZ
    WR(EN)=WR(NA)
    IF (ZZ.NE.C.OOO) WR(EN)=X-W/ZZ
C-----REAL PAIR-----
220

```

OPT28810
 OPT28820
 OPT28830
 OPT28840
 OPT28850
 OPT28860
 OPT28870
 OPT28880
 OPT28890
 OPT28900
 OPT28910
 OPT28920
 OPT28930
 OPT28940
 OPT28950
 OPT28960
 OPT28970
 OPT28980
 OPT28990
 OPT29000
 OPT29010
 OPT29020
 OPT29030
 OPT29040
 OPT29050
 OPT29060
 OPT29070
 OPT29080
 OPT29090
 OPT29100
 OPT29110
 OPT29120
 OPT29130
 OPT29140
 OPT29150
 OPT29160
 OPT29170
 OPT29180
 OPT29190
 OPT29200
 OPT29210
 OPT29220
 OPT29230
 OPT29240
 OPT29250
 OPT29260
 OPT29270
 OPT29280

```

WI(NA)=0.0DC
WI(EN)=0.0DC
GC TO 230
-----COMPLEX PAIR-----
220 WR(NA)=X+P
    WR(EN)=X+P
    WI(NA)=ZZ
    WI(EN)=ZZ
    EN=ENM2
    GC TO 30
230
-----SET ERROR --- NO CONVERGENCE TO AN-----
    EIGENVALUE AFTER 30 ITERATIONS-----
240 IERR=EN
250 RETURN
    END
C=====
    SUBROUTINE PSDCAL (N2,NS,FA,X,NC,GW,GV,C,AC,HY,HU,H,
1 FBGE,NG,GAM,ACL,F,WR,WI,D1,D2,JCF,RES,G,R,EP,CC,IYU,
2 IPSD,INORM)
C=====
    PSDCAL COMPUTES THE PSD OF OUTPUTS OR CONTROLS OF
    A CONTROLLED SYSTEM
C=====
    IYU= 1      CUTPUT PSD
    = 2      CONTROL PSD
    = 3      BOTH OUTPUT AND CONTROL PSD
    IPSD=1      PSD AND TF RESIDUES
    =2
    INORM=      1,2,... NG NORMALIZED BY ITH PROCESS NOISE
    NG+1,... NG+NG NORMALIZED BY ITH MEAS NOISE
C=====
    DOUBLE PRECISION FA,X,GW,GV,C,HY,H,FBGE,GAM,ACL,F,WR,WI,D1,D2,RES,
1 RB,CC,C,R,PSD,W,DNORM,DNI,EMAX,ELOG,EMOD,CH,ST,CM,RE,AI,HU,DW1
    COMPLEX X*16D,ZN,ZZ
    DIMENSION FA(N2,N2),X(N2,N2),GW(N2,NG),C(NC,NS),HY(NC,NS),H(NC,NS),
1 FBGE(NS,NC),GAM(NS,NG),ACL(NS,NS),F(NS,NS),WF(N2),D1(N2),D2(N2),
2 RES(N2),RES(N2),QING,NG1,R(NC,NC),PSD(30),W(30),BE(N2),CC(N2),GV(N2),
3 NOI,HU(NC,N2),DW1(4)
    INTEGER JCF(N2)
    DATA DW1/1.0,2.0,5.0,10.0/
    IF (IYU.EQ.0) IYU=1
    IF (INORM.EQ.0) INORM=1
    IPT=0
    IF (IPSD.EQ.1) IPT=1
    IF (INORM.NG

```

```

IF (IX .GT. 0) WRITE (6,330) IX
IF (IX .LE. 0) WRITE (6,340) INORM
NSQ=N2#N2
C----- COMPUTE EIGENSYSTEM OF CONTROLLED SYSTEM; FCRM FA-----
DO 10 I=1,NS
DO 10 J=1,NS
FA(I,J)=ACL(I,J)
FA(NS+I,J)=C.D0
DC 30 I=1,NS
DC 30 J=1,NS
ST=0.D0
DO 20 K=1,NC
ST=ST+FBGE(I,K)*H(K,J)
FA(I,NS+J)=-ST
FA(NS+I,NS+J)=F(I,J)-ST
CALL RAPRNT (N2,N2,N2,9,FA,4,'(9(IX,1PD13.6))')
C----- DEBUG ABOVE-----
CALL BALANC (N2,N2,FA,LCW,IHIGH,D1)
CALL ORTHES (N2,N2,LCW,IHIGH,FA,D2)
CALL ORTRAN (N2,N2,LCW,IHIGH,FA,D2,X)
CALL HQR2 (N2,N2,LOW,IHIGH,FA,WR,WI,X,IERR)
IF (IERR .NE. 0) GO TO 320
CALL BALBAK (N2,N2,LCW,IHIGH,D1,N2,X)
CALL RAPRNT (N2,N2,N2,5,X,4,'(9(IX,1PD13.6))')
C----- DEBUG ABOVE; DETERMINE MODAL MATRICES-----
IF (IYU .EC. 1) GO TO 60
C----- HSUBU-----
DO 50 I=1,NC
DO 50 J=1,N2
ST=0.D0
DO 40 K=1,NS
ST=ST-C(I,K)*X(K,J)
HU(I,J)=ST
GO TO 90
C----- HSUBY-----
DO 80 I=1,NC
DO 80 J=1,N2
ST=0.D0
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,NC,N2,9,HY,4,'(9(IX,1PD13.6))')
C----- DEBUG ABOVE-----
CALL MINV (NSC,X,N2,ST,D1,D2)
CALL RAPRNT (N2,N2,N2,5,X,4,'(9(IX,1PD13.6))')
C----- DEBUG ABOVE-----
DC 110 I=1,N2

```

```

OPT29770
OPT29780
OPT29790
OPT29800
OPT29810
OPT29820
OPT29830
OPT29840
OPT29850
OPT29860
OPT29870
OPT29880
OPT29890
OPT29900
OPT29910
OPT29920
OPT29930
OPT29940
OPT29950
OPT29960
OPT29970
OPT29980
OPT30000
OPT30010
OPT30020
OPT30030
OPT30040
OPT30050
OPT30060
OPT30070
OPT30080
OPT30090
OPT30100
OPT30110
OPT30120
OPT30130
OPT30140
OPT30150
OPT30160
OPT30170
OPT30180
OPT30190
OPT30200
OPT30210
OPT30220
OPT30230
OPT30240

```

```

100 CC 110 J=1,NG
110 ST=0.000
    DO 100 K=1,NS
    GW(I,J)=ST
    CALL RAPRNT (N2,N2,NG,9,GW,4,'(9(IX,1PD13,6))')
    IF (INORM .LE. NG) DNORM=1.00/0.0(INORM,INORM)
    IF (INORM .GT. NG) DNORM=1.00/R(INORM-NG,INORM-NG)
    C-----DETERMINE BANDWIDTH OF CONTROLLED SYSTEM-----
    EMAX=0.00
    DO 120 I=1,N2
    EMOD=DABS(WF(I)*2 +WI(I)*2)
    IF (EMOD .GT. EMAX) EMAX=EMOD
    CCNT INUE
    EMOD=2*EMOD
120 C-----ROUND UP TO NEAREST 2,4,5,8,10-----
    ELOG=DLOG10(EMOD)
    IF (ELOG .LT. 0.00) IPOW=-IDINT(DABS(ELOG) + 1)
    IF (ELOG .GE. 0.00) IPOW=IDINT(ELOG)
    EMAX=EMOD*10**(-IPOW)
    IF (EMAX .GT. 2.00) EMOD=2.00
    IF (EMAX .GT. 4.00) EMOD=4.00
    IF (EMAX .GT. 5.00) EMOD=5.00
    IF (EMAX .GT. 8.00) EMOD=8.00
    IF (EMAX .GE. 10.00) EMOD=10.00
    EMAX=EMOD*10**IPOW
    CW=EMAX/20.00
    C-----ADD 10 POINTS 3 DECADES UP-----
    IF (EMOD .LT. 5.0) GC TO 130
    EMAX=1.001
    IK=3
    GC TO 140
    EMAX=5.00
    IK=2
130 CCNT INUE
140 C-----STORE 30 FREQUENCIES-----
    DO 150 I=1,20
    W(I)=DW*(I-1)
    DO 160 J=1,3
    IP=20+2*(I-1)
    CG 160 J=1,3
    IX=MOD(IK+J-1,2)+1
    JJ=0
    IF (IK .EQ. 2 .AND. J .GE. 2) JJ=1
    W(IP+JJ)=DW*(IX)*10**J*(IPOW+I-1+JJ+IK-2)
    CCNT INUE
150
160

```

```

IX=MOD(IK,2)+1
W(30)=DW1(I)*10**((I*POW+3 +IK-2)
C-----LARGE LOOP THRU OUTPUTS-----
IF (IYU.EQ.1) NL=NC
IF (IYU.EQ.2) NL=NC
DO 310 I=1,NL
DO 170 I=1,20
PSD(I)=0.D0
C-----LCCP THRU PROCESS NOISE-----
DO 220 I=1,NG
CNI=DNCRM*(I,I)
IF (IYU.EQ.1) AND. IPT.EQ.1) WRITE (6,350) I,L
IF (IYU.EQ.2) AND. IPT.EQ.1) WRITE (6,360) I,L
IF (IYU.EQ.1) CALL RESID (I,L,N2,JCF,NG,GW,NL,HY,WR,WI,
1RES,BB,CC,IPT)
IF (IYU.EQ.2) CALL RESID (I,L,N2,JCF,NG,GW,NL,HU,WR,WI,
1RES,BB,CC,IPT)
DO 210 K=1,20
ZZ=DCMPLX(0.D0,0.D0)
OW=W(K)
DO 200 I=1,N2
IF (WI(I)) 200,180,190
ZZ=DCMPLX(-WR(I),OW-WI(I))
ZZ=RES(I)/ZZ+ZZ
GO TO 200
RE=WR(I)
AI=WI(I)
ZZ=DCMPLX(RES**2 + AI**2 - OW**2 - 2.D0*RE*CM)
ZN=DCMPLX(RES(I+1)*AI-RES(I)*RE,RES(I)*CM)
ZZ=ZZ+ZN/ZZ
CCNT INUE
PSD(K)=PSD(K)+DN1*(ZZ*DCONJG(ZZ))
CCNT INUE
C-----GSUBV-----
DO 240 I=1,N2
DO 240 J=1,NC
ST=0.D0
DO 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,NC,9,GV,4,'(9(IX,IPD13.6))')
C-----DEBUG ABOVE, LOOP THRU MEAS NOISE-----
DO 300 I=1,NC
CNI=DNCRM*(I,I)
IF (IYU.EQ.1) AND. IPT.EQ.1) WRITE (6,370) I,L
IF (IYU.EQ.2) AND. IPT.EQ.1) WRITE (6,380) I,L
IF (IYU.EQ.1) CALL RESID (I,L,N2,JCF,NG,GV,NL,HY,WR,WI,RES,
1BB,CC,IPT)

```

```

      IF (IYU.EQ.2) CALL RESID (I,L,N2,JCF,NO,GV,NL,HU,WR,WI,RES,
1    BB,CC,IFT)
      DO 290 K=1,30
      ZZ=DCMPLX(C.DC,0.D0)
      CM=W(K)
      DO 270 I=1,N2
      IF (WI(I)) Z7C=25C,26C
      ZD=DCMPLX(-WR(I),OM-WI(I))
      ZZ=ZZ+RES(I)/ZD
      GO TO 270
      RE=WR(I)
      AI=WI(I)
      ZC=DCMPLX(RE**2 + AI**2 -OM**2,-2.D0*RE*OM)
      ZN=DCMPLX(RES(I+1)*AI-RES(I)*RE,RES(I)*CM)
      ZZ=ZZ+ZN/ZC
      CCNTINUE
      IF (IYU.EC.2.OR. I.NE. L) GO TO 280
      PSD(K)=PSD(K)+DN1
      CCNTINUE
      IF (IYU.EC.1) WRITE (6,390) L
      IF (IYU.EC.2) WRITE (6,400) L
      WRITE (6,410) (W(I),PSD(I),I=1,30)
      CCNTINUE
      RETURN
      CALL EXEXIT (N2,FA,IERR)
      RETURN
-----
      FORMAT (/,4F SUBSEQUENT PSD IS NORMALIZED BY MEAS NO.,I3,/)
      FORMAT (/,50F SUBSEQUENT PSD IS NORMALIZED BY PROCESS NOISE NC.,I3
1,/)
      FORMAT (/,38F TRANSFER FUNCTION FROM PROCESS NCISE ,I2,3F TC,13H ME
1ASUREMENT ,I2,/)
      FORMAT (/,36F TRANSFER FUNCTION FROM PROCESS NCISE ,I2,3F TC,9F CON
1TROL ,I2,/)
      FORMAT (/,36F TRANSFER FUNCTION FROM MEASUREMENT ,I2,16H TO MEASURE
1MENT ,I2,/)
      FORMAT (/,36F TRANSFER FUNCTION FROM MEASUREMENT ,I2,12H TO CCNTPOL
1,/)
      FORMAT (/,14F PSD OF OUTPUT,I3,32H FORCED BY ALL NCISE-(RAD FRFQ.,
115HNORMALIZED PSD)/)
      FORMAT (/,14F PSD OF CCNTROL,I3,32H FORCED BY ALL NCISE-(RAD FRFQ.,
115HNORMALIZED PSD)/)
      FORMAT (4(I,1H,E11.4,1H,E11.4,1H,E11.4,1H))
      END
=====

```

```

OPT31210
OPT31220
OPT31230
OPT31240
OPT31250
OPT31260
OPT31270
OPT31280
OPT31290
OPT31300
OPT31310
OPT31320
OPT31330
OPT31340
OPT31350
OPT31360
OPT31370
OPT31380
OPT31390
OPT31400
OPT31410
OPT31420
OPT31430
OPT31440
OPT31450
OPT31460
OPT31470
OPT31480
OPT31490
OPT31500
OPT31510
OPT31520
OPT31530
OPT31540
OPT31550
OPT31560
OPT31570
OPT31580
OPT31590
OPT31600
OPT31610
OPT31620
OPT31630
OPT31640
OPT31650
OPT31660
OPT31670
OPT31680

```



```

WRITE (5,12C) K,L
CALL RCRREAL (ANSR)
DUM=ANSR
DO 100 I=1,NS
DO 90 J=1,NS
IF ((I.EQ.K).AND.(J.EQ.L)) BA(I,J)=DUM
CCNTINUE
GO TO 30
CCNTINUE
CALL FRTCMS ('CLRSCRN ')
RETURN
C-----
120 FORMAT (5X,14H THE ELEMENT F(I2,1H,I2,2H)=)
130 FORMAT (/,EX,26H ENTER THE SYSTEM MATRIX "F"-MATRIX,/,10X,41HDIMENSIONAL MATRIX "F" STATES NS& X # STATES NS&)
140 FORMAT (//,EX,33H THE SYSTEM MATRIX "F"-MATRIX,/,10X,41HDIMENSIONAL MATRIX "F" STATES NS& X # STATES NS&)
150 FORMAT (//,EX,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT?//,10X,19H TYPE "YES" OR "NO".)
160 FORMAT (//,EX,54H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
170 FORMAT (5X,14H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
180 FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED.)
1,END
C=====
SUBROUTINE READH (NO,NS,ISAH,HO)
INTERACTIVELY INPUTS THE "H" MATRIX MEASUREMENT SCALING MATRIX.
C=====
REAL*8 HO(NO,NS),DUM,ANSR
INTEGER IANS,IJ,K,L,ISAH
DATA IV,Y,IZ,N/
C-----
IF (ISAH.EQ.1) GO TO 40
WRITE (5,12C)
DO 20 I=1,NC
DO 10 J=1,NS
WRITE (5,11C) I,J
CALL RCRREAL (ANSR)
H(I,J)=ANSR
CCNTINUE
CCNTINUE
C-----
CALL FRTCMS ('CLRSCRN ')
CCNTINUE
WRITE (5,12C)
CALL MATPRT (HO,NO,NS)
WRITE (5,14C)
CALL RDOCHAR (IANS)
C-----

```



```

100 CALL RCREAL (ANSR)
110 D(I,J)=ANSR
120 CCNT INUE
130 CCNT INUE
140 CALL FRICMS ('CLRSCRN ')
150 WRITE (5,12C)
160 CALL MATPR1 (D,NO,NC)
170 WRITE (5,13C)
180 CALL RDCCHAR (IANS)
190 IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
200 GO TO 60
210 WRITE (5,14C)
220 GO TO 40
230 CCNT INUE
240 IF (IANS.EQ.IZ) GO TO 90
250 WRITE (5,15C)
260 CALL RDCINT (IANS)
270 K=IANS
280 WRITE (5,16C)
290 CALL RDCINT (IANS)
300 L=IANS
310 WRITE (5,17C) K,L
320 CALL RDCREAL (ANSR)
330 CUM=ANSR
340 DO 80 I=1,NC
350 DO 70 J=1,NC
360 IF ((I.EQ.K).AND.(J.EQ.L)) D(I,J)=DUM
370 CCNT INUE
380 GO TO 30
390 CCNT INUE
400 CALL FRICMS ('CLRSCRN ')
410 RETURN
420
430 FCRMAT (5X,14HTHE ELEMENT D(,12,1H,12,2H)=)
440 FCRMAT (/,5X,54HENTER THE MEASUREMENT FEEDTHROUGH MATRIX / FEEDFOROPT33480
450 IWARD,/,5X,34H DISTRIBUTION MATRIX "D"-MATRIXE.,//.8X,4SHDIMENSIONOPT33490
460 2=#OBSERVATIONSC NCC&X# CONTROLS NCC&)
470 FCRMAT (//,5X,50HTHE FEEDFORWARD DISTRIBUTION MATRIX "C"-MATRIXE.OPT33510
480 1,/)
490 FCRMAT (//,5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENTOPT33520
500 1ENT?//,10X,1SHTYPE "YES" OR "NO".)
510 FCRMAT (1X,15HWARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)OPT33530
520 FCRMAT (5X,5CHENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.OPT33540
530 FCRMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED.OPT33550
540 1,.)
550 END
OPT33130
OPT33140
OPT33150
OPT33160
OPT33170
OPT33180
OPT33190
OPT33200
OPT33210
OPT33220
OPT33230
OPT33240
OPT33250
OPT33260
OPT33270
OPT33280
OPT33290
OPT33300
OPT33310
OPT33320
OPT33330
OPT33340
OPT33350
OPT33360
OPT33370
OPT33380
OPT33390
OPT33400
OPT33410
OPT33420
OPT33430
OPT33440
OPT33450
OPT33460
OPT33470
OPT33480
OPT33490
OPT33500
OPT33510
OPT33520
OPT33530
OPT33540
OPT33550
OPT33560
OPT33570
OPT33580
OPT33590
OPT33600

```

```

C=====OPT333610
C SUBROUTINE READG (NS,NC,ISAG,G)OPT333620
C INTERACTIVELY INPUTS THE "G" MATRIX CONTROL DISTRIBUTION MATRIXE=OPT333630
C=====OPT333640
REAL*8 G(NS,NC),DUM,ANSR
INTEGER IANS,IJ,K,L,ISAG
DATA IY//,IY//,IZ//,N//
IF (ISAG.EQ.1) GO TO 40
WRITE (5,12C)
DO 20 I=1,NS
DO 10 J=1,NC
WRITE (5,11C) I,J
CALL RCREAL (ANSR)
G(I,J)=ANSF
CCNT INUE
CCNT INUE
1C-----
2C-----
3C-----
4C-----
5C-----
6C-----
7C-----
80-----
9C-----
10C-----
C-----
CALL FRTCMS ('CLRSCRN ')
CCNT INUE
WRITE (5,13C)
CALL MATPR1 (G,NS,NC)
WRITE (5,14C)
CALL RCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
GO TO 70
WRITE (5,15C)
GO TO 50
CCNT INUE
IF (IANS.EQ.IZ) GO TO 100
WRITE (5,16C)
CALL RCINT (IANS)
K=IANS
WRITE (5,17C)
CALL RDINT (IANS)
L=IANS
WRITE (5,11C) K,L
CALL RCREAL (ANSR)
DUM=ANSR
DO 30 I=1,NS
DO 30 J=1,NC
IF ((I.EQ.K).AND.(J.EQ.L)) G(I,J)=DUM
CCNT INUE
GO TO 30
CCNT INUE
CALL FRTCMS ('CLRSCRN ')
RETURN
C-----
OPT333650
OPT333660
OPT333670
OPT333680
OPT333690
OPT333700
OPT333710
OPT333720
OPT333730
OPT333740
OPT333750
OPT333760
OPT333770
OPT333780
OPT333790
OPT333800
OPT333810
OPT333820
OPT333830
OPT333840
OPT333850
OPT333860
OPT333870
OPT333880
OPT333890
OPT333900
OPT333910
OPT333920
OPT333930
OPT333940
OPT333950
OPT333960
OPT333970
OPT333980
OPT333990
OPT340000
OPT340010
OPT340020
OPT340030
OPT340040
OPT340050
OPT340060
OPT340070
OPT340080

```



```

100 7C DO 80 I=1,NC
110 8C IF ((I.EQ.K).AND.(J.EQ.L)) FBGC(I,J)=DUM
120 9C CCNT INUE
130 9C CCNT INUE
140 9C CCNT INUE
150 9C CCNT INUE
160 9C CCNT INUE
170 9C CALL FRTCMS ('CLRSCRN ')
180 9C RETURN
190 9C
200 9C FORMAT (5X,14HTPE ELEMENT C(I2,IH,I2,2H)=)
210 9C FORMAT (//,5X,52HTPE THE FEEDBACK GAIN CCNTRCL MATRIX "C"-MATRIX
220 9C 18,//,10X,24HTPE NCS X # STATES "NSE.")
230 9C FORMAT (//,10X,45HTPE FEEDBACK GAIN CONTROL MATRIX "C"-MATRIX&, //
240 9C 1)
250 9C FORMAT (//,5X,54HDO YCU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT
260 9C 1ENT?//,10X,19HTPE "YES" OR "NO".)
270 9C FCRMAT (//,1X,11HTPE IMPROPER DATA ENTRY ENTER "YES" CR "NAC".)
280 9C FORMAT (5X,52HTPE THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.
290 9C 1)
300 9C FORMAT (5X,53HTPE THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
310 9C 1)
320 9C END
330 9C
340 9C SUBROUTINE READAY (NC,ISAA,AY)
350 9C INPUTS THE "A" MATRIX DIAGONAL OUTPUT COST MATRIX&.
360 9C
370 9C REAL*8 AY(NC,NC),DUM,ANSR
380 9C INTEGER IANS,IJ,K,L
390 9C DATA IY,Y,Y,Y,I2,N,N,
400 9C IF (ISAA.EQ.1) GO TO 30
410 9C WRITE (5,11C)
420 9C DO 20 I=1,NC
430 9C DO 10 J=1,NC
440 9C WRITE (5,11C) I,J
450 9C CALL RDRREAL (ANSR)
460 9C AY(I,J)=ANSR
470 9C CCNT INUE
480 9C
490 9C CALL FRTCMS ('CLRSCRN ')
500 9C WRITE (5,12C)
510 9C CALL MATPRT (AY,NO,NC)
520 9C WRITE (5,13C)
530 9C CALL RDRCHAP (IANS)
540 9C IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
550 9C GO TO 60
560 9C WRITE (5,14C)
570 9C GO TO 40
580 9C
590 9C
600 9C
610 9C
620 9C
630 9C
640 9C
650 9C
660 9C
670 9C
680 9C
690 9C
700 9C
710 9C
720 9C
730 9C
740 9C
750 9C
760 9C
770 9C
780 9C
790 9C
800 9C
810 9C
820 9C
830 9C
840 9C
850 9C
860 9C
870 9C
880 9C
890 9C
900 9C
910 9C
920 9C
930 9C
940 9C
950 9C
960 9C
970 9C
980 9C
990 9C
1000 9C

```



```

10C IF (IGAM.EQ.1) GO TO 40
11C WRITE (5,12C)
12C DO 20 I=1,NS
13C DO 10 J=1,NG
14C WRITE (5,11C) I,J
15C CALL RCREAL (ANSR)
16C GAM(I,J)=ANSR
17C CONTINUE
18C CONTINUE
19C -----
20C CALL FRTCMS ('CLRSCRN ')
21C CONTINUE
22C WRITE (5,13C)
23C CALL MATPRT (GAM,NS,NG)
24C WRITE (5,14C)
25C CALL RCHAR (IANS)
26C IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
27C GO TO 70
28C WRITE (5,15C)
29C GO TO 50
30C CONTINUE
31C IF (IANS.EQ.IZ) GO TO 100
32C WRITE (5,16C)
33C CALL RDINT (IANS)
34C K=IANS
35C WRITE (5,17C)
36C CALL RDINT (IANS)
37C L=IANS
38C WRITE (5,11C) K,L
39C CALL RCREAL (ANSR)
40C DUM=ANSR
41C DO 90 I=1,NS
42C DO 80 J=1,NG
43C IF ((I.EQ.K).AND.(J.EQ.L)) GAM(I,J)=DUM
44C CONTINUE
45C CONTINUE
46C GO TO 30
47C CONTINUE
48C CALL FRTCMS ('CLRSCRN ')
49C RETURN
50C -----
51C FORMAT (5X,16HTPE ELEMENT GAM(I2,1H,I2,2H)=)
52C FORMAT (//1X,26HTENTER THE PROCESS NOISE DISTRIBUTION,/,5X,24HMAIRIN
53C 1X "GAMMA"-MATRIX,/,2X,56HDIMENSION = # STATES NS X # PROCESS
54C 2X NOISE SOURCES NG X)
55C FORMAT (//1X,10X,37HTHE PROCESS NOISE DISTRIBUTION MATRIX,/,10X,19H
56C 1 "GAMMA"-MATRIX,/,//)
57C FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATPIX FLEM

```



```

SC      CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----
100     FORMAT (5X,14H"THE ELEMENT Q(I,I2,1H,I2,2H)=)
110     FORMAT (//,5X,44H"ENTER THE PROCESS NOISE PSD WEIGHTING MATRIX,/,5X,42H"PROCESS NOISE SOURCES NG6)
120     1,12H"Q"MATRIX,/,/,5X,42H"PROCESS NOISE SOURCES NG6)
130     2 X,/,17X,27H"PROCESS NOISE SOURCES NG6)
140     FORMAT (//,5X,42H"THE PROCESS NOISE WEIGHTING MATRIX,/,/,5X,42H"PROCESS NOISE SOURCES NG6)
150     1ENT2,/,10X,19H"TYPE "YES" OR "NO".)
160     FORMAT (1X,11H"WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
      FORMAT (5X,53H"ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.")
      FORMAT (5X,53H"ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED.")
1.    )
      END
C=====
C      SUBROUTINE READR (NC,RC)
C      INTERACTIVELY INPUTS THE "R" MATRIX=
C      MEASUREMENT NOISE DISTRIBUTION MATRIX.8
C=====
      REAL*8 RC(NC,NO),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY,Y,Y,I2,N:/
      WRITE (5,90)
      DO 10 I=1,NC
      DO 10 J=1,NC
      WRITE (5,8C) I,J
      CALL RDCREAL (ANSR)
      RCTI,J)=ANSR
1C-----
2C     CALL FRTCMS ('CLRSCRN ')
      WRITE (5,10C)
      CALL MATPRT (RC,NO,NC)
      WRITE (5,11C)
      CALL RDCCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.I2)) GO TO 40
      GO TO 50
      WRITE (5,12C)
      GO TO 30
      CONTINUE
      IF (IANS.EC.I2) GO TO 70
      WRITE (5,13C)
      CALL RDCINT (IANS)
      K=IANS
      WRITE (5,14C)
      CALL RDCINT (IANS)
      L=IANS

```



```

60      GO TO 40
      CONTINUE
      IF (IANS.EQ.1Z) GO TO 90
      WRITE (5,15C)
      CALL RDCINT (IANS)
      K=IANS
      WRITE (5,16C)
      CALL RDCINT (IANS)
      L=IANS
      WRITE (5,10C) K,L
      CALL RDRREAL (ANSR)
      DUM=ANSR
      DO 80 I=1,NS
      DO 70 J=1,NC
      IF ((I.EQ.K).AND.(J.EQ.L)) FBGE(I,J)=DUM
      CONTINUE
      GC TO 30
      CONTINUE
      GC TO 30
      CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
-----
100  FORMAT (5X,14HTHE ELEMENT K(I2,1H,I2,2H)=)
110  FORMAT (//5X,54HTHE FEEDBACK GAIN ESTIMATOR MATRIX "K"-MATRIX
      11X&://,10X,48HDIMENSION = # STATES' NS& X # CBSEVATICS' NC&.)
120  FORMAT (//,15X,47HTHE FEEDBACK GAIN ESTIMATOR MATRIX "K"-MATRIX&,
      1//)
130  FORMAT (//,5X,54HDC YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELE
      1MENT?//,10X,19HTYPE "YES" OR "NO".)
140  FORMAT (//,1X,51HWARNING: IMPROPER DATA ENTRY ENTER "YES" CR "NO".)
150  FORMAT (5X,52HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160  FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
      1)
      END
C=====
      SUBROUTINE READW (NG,WR)
      INTERACTIVELY INPUTS THE "W0" MATRIX STEADY DISTURBANCE VECTOR =
      MATRIX& ELEMENT BY ELEMENT.
C=====
      REAL*8 WR(NG),DUM,ANSR
      INTEGER IANS,I,K
      DATA IY,Y,Z,I2/'N'/'
      WRITE (5,10C)
      DO 10 I=1,NC
      DO 10 J=1,NS
      WRITE (5,80) I
      CALL RDRREAL (ANSR)
      WR(I)=ANSR
      CONTINUE
      1C

```

```

C-----
2C  CALL FRTCMS ('CLRSCRN ')
    WRITE (5,13C) (WR(I),I=1,NG)
3C  WRITE (5,12C)
    CALL RCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 40
4C  GO TO 50
    WRITE (5,13C)
5C  GO TO 30
    CCNTINUE
    IF (IANS.EC.IZ) GO TO 70
    WRITE (5,14C)
    CALL RDINT (IANS)
    K=IANS (5,8C) K
    CALL RREAL (ANSR)
    CUM=ANSR
    DO 60 I=1,NG
6C  IF (I.EQ.K) WR(I)=DUM
    CCNTINUE
7C  GO TO 20
    CCNTINUE
    CALL FRTCMS ('CLRSCRN ')
    RETURN
C-----
8C  FORMAT (5X,15HTHE ELEMENT W0(,I2,2H)=)
9C  FORMAT (F12.5)
10C  FORMAT (/,5X,57HENTER THE STEADY DISTURBANCE VECTOR MATRIX "W0"-M
    1ATRX6. //,10X,44HDIMENSION = # PROCESS NOISE SOURCES NGE X 1)
110  FORMAT (//,15X,53HTHE STEADY DISTURBANCE VECTOR MATRIX "W0"-MATR
    1X6. //)
120  FORMAT (//,5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELE
    1ENT? //,10X,19HTYPE "YES" OR "NO".)
130  FORMAT (1X,11HWARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
14C  FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
    END
C=====
C  SUBROUTINE RREAL --- 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 RREAL (ANSR)
    REAL*8 ANSR
    INTEGER COUNT
C-----
1C  COUNT=0
    CCNTINUE
    OPT38410
    OPT38420
    OPT38430
    OPT38440
    OPT38450
    OPT38460
    OPT38470
    OPT38480
    OPT38490
    OPT38500
    OPT38510
    OPT38520
    OPT38530
    OPT38540
    OPT38550
    OPT38560
    OPT38570
    OPT38580
    OPT38590
    OPT38600
    OPT38610
    OPT38620
    OPT38630
    OPT38640
    OPT38650
    OPT38660
    OPT38670
    OPT38680
    OPT38690
    OPT38700
    OPT38710
    OPT38720
    OPT38730
    OPT38740
    OPT38750
    OPT38760
    OPT38770
    OPT38780
    OPT38790
    OPT38800
    OPT38810
    OPT38820
    OPT38830
    OPT38840
    OPT38850
    OPT38860
    OPT38870
    OPT38880

```

```

CCOUNT=COUNT+1
IF (COUNT.LT.3) GO TC 20
WRITE (5,6C)
GO TO 40
CCNTINUE
2C READ (5,*,END=30,ERR=30) ANSR
RETURN
3C REWIND 5
WRITE (5,50)
GO TO 10
4C CCNTINUE
STOP
C-----
5C FORMAT (1X,64HWARNING: NULL STRINGS ARE NOT ALLOWED, ENTER A NUMERICAL VALUE.)
6C FORMAT (//,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED)
END
C=====
C SUBROUTINE RDINT --- INTERACTIVELY READS AN INTEGER REPLY
C INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS AN IMPROPER
C DATA CHARACTER THE S/R ISSUES A WARNING AND ALLWS A RECOVERY.
C=====
SUBROUTINE RDINT (IANS)
INTEGER CCLNT,IANS
C-----
1C CCLNT=0
CCNTINUE
C CCLNT=COUNT+1
IF (COUNT.LT.3) GO TC 20
WRITE (5,6C)
GO TO 50
2C CCNTINUE
READ (5,*,END=40,ERR=40) IANS
3C IF (IANS) 4C,40,30
4C RETURN
5C REWIND 5
WRITE (5,70)
GO TO 10
6C CCNTINUE
STOP
C-----
6C FORMAT (//,5X,49HPROGRAM TERMINATION - TWO IMPROPER DATA ENTRIES)
7C FORMAT (1X,56HWARNING: IMPROPER DATA ENTRY ENTER A POSITIVE INTEGER.)
END
C=====

```



```

IF (NCOL.EC.11) WRITE (5,110) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCOL.EC.12) WRITE (5,120) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCOL.EC.13) WRITE (5,130) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCOL.EC.14) WRITE (5,140) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCOL.EC.15) WRITE (5,150) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
IF (NCOL.EC.16) WRITE (5,160) ((PRIT(I,J),J=1,NCOL),I=1,NROW)
RETURN
-----
FORMAT (F12.5)
FORMAT (2F12.5)
FORMAT (3F12.5)
FORMAT (4F12.5)
FORMAT (5F12.5)
FORMAT (6F12.5)
FORMAT (6F12.5,/,F12.5,/,/)
FORMAT (6F12.5,/,2F12.5,/,/)
FORMAT (6F12.5,/,3F12.5,/,/)
FORMAT (6F12.5,/,4F12.5,/,/)
FORMAT (6F12.5,/,5F12.5,/,/)
FORMAT (6F12.5,/,6F12.5,/,/,F12.5,/,/)
FORMAT (6F12.5,/,6F12.5,/,/,2F12.5,/,/)
FORMAT (6F12.5,/,6F12.5,/,/,3F12.5,/,/)
FORMAT (6F12.5,/,6F12.5,/,/,4F12.5,/,/)
END
=====
C SUBROUTINE RDMATF -- READS THE FLAGS AND MATRIX SIZES FROM
C THE DATA FILE CN FILEDEF 9. ASKS IF YOU WANT TO USE THE MATRICES.
C=====
SUBROUTINE RDMATF (NS,NC,NOB,NG,ISAF,ISAG,ISAM,ISAA,ISAB,IRDMAT,INO,IAN,S,K
1AT)
DATA IYES/'Y',IND/'N'/
INTEGER NS,NC,NOB,NG,ISAF,ISAG,ISAH,ISAM,IRDMAT,INO,IAN,S,K
REWIND 9
READ (5,24C,END=30,EPR=30) K,IAN
IF (IAN.EC.1) GO TO 10
GO TO 30
READ (9,25C) NS,NC,NOB,NG
WRITE (5,255)
CALL FRTCM5 ('CLRSCRN ')
CALL RDCINT (IAN)
IF (IAN.GT.3) GO TO 2C
IF (IAN.EC.3) GO TO 3C
IRDMAT=1
IF (IAN.EC.2) GO TO 4C
ISAF=1
ISAG=1

```



```

1//,5X, 'RUN TO BE USED IN THIS RUN?'
2MATRIX, 'WILL BE REDISPLAYED AT //,5X,34HTHE PROPER INPUT SEQUENCE INTOPT41290
3SERVAL, //,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING, //,5X,27HINDOPT41300
4INDIVIDUAL MATRIX ELEMENTS, //,15X,19HTYPE "YES" OR "NO" OPT41310
FORMAT (//,5X,52HDO YOU WISH TO SAVE THE "GAMMA"-MATRIX FROM THE OPT41320
LAST //,5X, 'RUN TO BE USED IN THIS RUN?' OPT41330
1THE MATRIX, 'WILL BE REDISPLAYED AT //,5X,34HTHE PROPER INPUT SEQUENCE INTOPT41340
2SERVAL, //,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING, //,5X,27HINDOPT41350
4INDIVIDUAL MATRIX ELEMENTS, //,15X,19HTYPE "YES" OR "NO" OPT41360
FORMAT (//,5X,52HDO YOU WISH TO SAVE THE "A"-MATRIX FROM THE LASTOPT41380
//,5X,39HNOTE: THE LASTOPT41390
1MATRIX, 'WILL BE REDISPLAYED AT //,5X,34HTHE PROPER INPUT SEQUENCE INTOPT41400
2SERVAL, //,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING, //,5X,27HINDOPT41410
4INDIVIDUAL MATRIX ELEMENTS, //,15X,19HTYPE "YES" OR "NO" OPT41420
FORMAT (//,5X,52HDO YOU WISH TO SAVE THE "B"-MATRIX FROM THE LASTOPT41430
//,5X,39HNOTE: THE LASTOPT41440
1MATRIX, 'WILL BE REDISPLAYED AT //,5X,34HTHE PROPER INPUT SEQUENCE INTOPT41450
2SERVAL, //,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING, //,5X,27HINDOPT41460
4INDIVIDUAL MATRIX ELEMENTS, //,15X,19HTYPE "YES" OR "NO" OPT41470
FORMAT (IX,FIHWARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".) OPT41480
END OPT41490
C===== OPT41500
C SUBROUTINE RDMAT -- READS THE F, G, H, GAM, A AND B MATRICES FROM =OPT41510
C MATRICES FROM THE DATA FILE OPTMAT ON FILEDEF 9. =OPT41520
C===== OPT41530
SUBROUTINE RDMAT(BA,G,HO,GAM,FBGC,FBGE,AY,B,NS,NC,NG,IRDMAT) OPT41540
IMPLICIT REAL*8(A-H,C-Z) OPT41550
DIMENSION BA(NS,NS),G(NS,NC),HO(NO,NS),GAM(NS,NG),FBGC(AC,NS), OPT41560
LAY(NO,NO),B(NC,NC),FBGE(NS,NO) OPT41570
IF(IRDMAT.EC.0) RETURN OPT41580
IF(IND 9 OPT41590
READ (9,20) K,IANS OPT41600
READ (9,20) NSI,NCI,NOI,NGI I=1,NSI) OPT41610
READ (9,10) ((BA(I,J),J=1,NCI),I=1,NSI) I=1,NSI) OPT41620
READ (9,10) ((G(I,J),J=1,NCI),I=1,NSI) I=1,NSI) OPT41630
READ (9,10) ((HO(I,J),J=1,NGI),I=1,NSI) I=1,NSI) OPT41640
READ (9,10) ((GAM(I,J),J=1,NCI),I=1,NSI) I=1,NSI) OPT41650
READ (9,10) ((FBGC(I,J),J=1,NCI),I=1,NSI) I=1,NSI) OPT41660
READ (9,10) ((FBGE(I,J),J=1,NCI),I=1,NCI) I=1,NCI) OPT41670
READ (9,10) ((LAY(I,J),J=1,NCI),I=1,NCI) I=1,NCI) OPT41680
READ (9,10) ((B(I,J),J=1,NCI),I=1,NCI) I=1,NCI) OPT41690
RETURN OPT41700
C----- OPT41710
10 FORMAT(4(D2C.13)) OPT41720
20 FCRMAT(415) OPT41730
END OPT41740
C----- OPT41750
C SUBROUTINE WRDMAT -- WRITES THE F, G, HO & GAM MATRICES TO =OPT41760

```


APPENDIX C
OPTGRAPH PROGRAM LISTING

This portion of the thesis contains the OPTGRAPH FORTRAN program (54 pages).


```

3C      CALL FRTCMS('CLRSCRN ')
        WRITE(6,92C)
        WRITE(6,750)
        WRITE(6,76C)
        CALL RCINT(IANS)
        IF(IANS.GT.4000.IANS.LT.1) GO TO 34
        GO TO 36
34      WRITE(6,505)
        GO TO 35
36      CCNTINUE
        IF(IANS.EQ.1) IFL = 10
        IF(IANS.EQ.2) IFL = 1
        IF(IANS.EQ.3) IFL = 4
        IF(IANS.EQ.4) GO TO 95
C-----READ(IFL,10C) IA,IB,NS,NC,NOB,IE,ITFX
C-----CHECK DATA FILE FOR PROPER DATA
        IF(IA.NE.0) GO TO 41
        IF(ITFX.EQ.1) WRITE(6,510)
        IF(ITFX.EQ.2) WRITE(6,511)
        IF(ITFX.EQ.3) WRITE(6,512)
        GO TO 35
41      IF(NS.NE.0) GO TO 42
        WRITE(6,513)
        GO TO 35
42      IF(NC.NE.0) GO TO 43
        IF(ITFX.EQ.1) WRITE(6,515)
        IF(ITFX.EQ.2) WRITE(6,517)
        IF(ITFX.EQ.3) WRITE(6,516)
        GO TO 35
43      IF(NOB.NE.0) GO TO 44
        IF(ITFX.EQ.1) WRITE(6,516)
        IF(ITFX.EQ.2) WRITE(6,516)
        IF(ITFX.EQ.3) WRITE(6,515)
        GO TO 35
C-----MARKCV CRITERIA FOR DETERMINING EXTRANEOUS ZEPES
44      IF(IE.EQ.0) IE = 6
        CC = 10.**(-IE)
C-----DENOMINATOR
        CC 50 I = 1,NS
        REAC(IFL,120) OLD(I,1),OLD(I,2)
5C      CCNTINUE
C-----NUMERATOR / GAIN / ORDER
        NN = NOB*NC
        DO 90 I = 1,NN
        READ(IFL,101) KK,K2,K1
        READ(IFL,120) GAN(1,K2,K1),GAN(2,K2,K1)
        IF(GAN(1,K2,K1).NE.0.0) GO TO 72
OPT000490
OPT000500
OPT000510
OPT000520
OPT000530
OPT000540
OPT000550
OPT000560
OPT000570
OPT000580
OPT000590
OPT000600
OPT000610
OPT000620
OPT000630
OPT000640
OPT000650
OPT000660
OPT000670
OPT000680
OPT000690
OPT000700
OPT000710
OPT000720
OPT000730
OPT000740
OPT000750
OPT000760
OPT000770
OPT000780
OPT000790
OPT000800
OPT000810
OPT000820
OPT000830
OPT000840
OPT000850
OPT000860
OPT000870
OPT000880
OPT000890
OPT000900
OPT000910
OPT000920
OPT000930
OPT000940
OPT000950
OPT000960

```

```

DC 70 LL = 1,NS
CLN(LL,1,K2,K1) = 0.0C
CLN(LL,2,K2,K1) = 0.0C
CCNTINUE
GC TO 75
CCNTINUE
DO 75 LL = 1,NS
  READ(IFL,120) OLN(LL,1,K2,K1),OLN(LL,2,K2,K1)
CCNTINUE
CALL GRAPH(NS,NC,NOB,IE,ITFX)
GC TO 30
CALL DCNEPL
STOP
-----
100 FORMAT(11,I4,5I5)
101 FORMAT(5X,2I3)
102 FORMAT(5X,2I3,14)
109 FORMAT(5X,2I5)
110 FORMAT(/,5X,***** ERROR *****//,10X,'IMPROPER DATA ENTRY',/)
111 *'PROPER DATA FILE IS NOT AVAILABLE FOR OPEN LOOP TF ANALYSIS')
112 *'PROPER DATA FILE IS NOT AVAILABLE FOR NOISE TF ANALYSIS')
113 *'PROPER DATA FILE IS NOT AVAILABLE FOR CCNFSATOR TF ANALYSIS')
114 *'MUST INPUT "F" MATRIX IN OPTSYSX PROGRAM')
115 *'MUST INPUT "G" MATRIX IN OPTSYSX PROGRAM')
116 *'MUST INPUT "H" MATRIX IN OPTSYSX PROGRAM')
117 *'MUST INPUT "GAMMA" MATRIX IN OPTSYSX PROGRAM')
118 *'MUST INPUT "CO YOU DESIRE TO ANALYZE:',/,
119 *'10X,1. OPEN LOOP TRANSFER FUNCTION',/,
120 *'10X,2. NCNFSATOR TRANSFER FUNCTION',/,
121 *'10X,3. CCNFSATOR TRANSFER FUNCTION',/,
122 *'10X,4. EXIT OPTGRAPH',/
123 FORMAT(5X,ENTER OPTION NUMBER,')
124 FORMAT(/,20X,'OPTGRAPH',/)
-----
END
=====
SUBROUTINE GRAPH(NS,NC,NOB,IE,ITFX)
C CALLS GRAPHICS ROUTINES FOR POLE-ZERO PLOTS, ROOT LOCUS, AND

```



```

C-----POLE-ZERO-----OPT01930
110 CALL PZERO(NS,JJ,II,ITFX)OPT01940
   GO TO 100OPT01950
C-----ROOT LOCUS-----OPT01960
120 CALL RTLC(NS,JJ,II,ITFX)OPT01970
   GO TO 100OPT01980
C-----BODE-----OPT01990
130 CALL BCDE(NS,JJ,II,ITFX)OPT02000
   GO TO 100OPT02010
C-----NYQUIST-----OPT02020
140 CALL NYQST(NS,JJ,II,ITFX)OPT02030
   GO TO 100OPT02040
C-----NICHOLS-----OPT02050
150 CALL NICHOL(NS,JJ,II,ITFX)OPT02060
   GO TO 100OPT02070
C-----EXIT OPGRAPH-----OPT02080
160 CONTINUEOPT02090
   RETURNOPT02100
C-----***** WARNING *****//,5X,'IMPRPER DATA ENTRY ',OPT02110
*3X,'ENTER "YES" OR "NO".',//OPT02120
*3X,'***** ERROR *****//,10X,'IMPROPER DATA ENTRY',//OPT02130
FORMAT(1X,'WARNING: THE TRANSFER FUNCTION INPUT# MUST NOT BE LESSOPT02140
*THAN 1 NOR GREATER THAN 13',//)OPT02150
FORMAT(1X,'WARNING: THE TRANSFER FUNCTION OUTPUT# MUST NOT BE LESSOPT02160
*THAN 1 NOR GREATER THAN 13',//)OPT02170
FORMAT(//,5X,'AN OPEN LOOP POLE-ZERO, ROOT LCCLS, BODE, NYQUIST,OPT02180
*AND/OR NICHOLS',//,5X,'PLOT IS DESIRED FOR:',//)OPT02190
FORMAT(10X,'INPUT #?',//)OPT02200
FORMAT(10X,'INPUT #?',//)OPT02210
FORMAT(5X,'OPEN LCCF TRANSFER FUNCTION',//)OPT02220
FORMAT(5X,'NOISE TRANSFER FUNCTION',//)OPT02230
FORMAT(5X,'COMPENSATOR TRANSFER FUNCTION',//)OPT02240
FORMAT(10X,'INPUT # =',13)OPT02250
FORMAT(10X,'OUTPUT # =',13)OPT02260
FORMAT(//,5X,'DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR DATAOPT02270
*FOR:',//,PCLE-ZERO MAP',//,OPT02280
*10X,'1. RECT-LOCUS',//,OPT02290
*10X,'2. BODE',//,OPT02300
*10X,'3. NYQUIST (POLAR PLOT)',//,OPT02310
*10X,'4. NICHOLS',//,OPT02320
*10X,'5. ANOTHER INPUT/OUTPUT COMBINATION (SAME TRANSFER FUNCTION)OPT02330
*10X,'6. ANOTHER TYPE TRANSFER FUNCTION /EXIT CPTGRAPH')OPT02340
*10X,'7. ANOTHER OPTION NUMBER.',//,OPT02350
FORMAT(5X,'ENTER OPTION NUMBER.',//)OPT02360
FORMAT(//,20X,'OPGRAPH',//)OPT02370
FORMAT(5X,'THE OPEN LOOP TRANSFER SYSTEM OBTAINED FROM CPTSYS CONTOPT02380
   OPT02390
   OPT02400

```

```

902 #AINS:','//)
903 #FORMAT(5X,'THE NOISE TRANSFER SYSTEM OBTAINED FROM OPTSYS CONTAINS
904 #NTAINS:','//)
905 #FORMAT(5X,'THE COMPENSATOR TRANSFER SYSTEM OBTAINED FROM OPTSYS CO
906 #NTAINS:','//)
907 #FOR4AT(10X,'1. NUMBER OF STATES = ',I3,/,
908 #10X,'2. NUMBER OF MEASUREMENTS (OUTPUTS) = ',I3,/,
909 #10X,'3. NUMBER OF MARKOV PARAMETERS = 10.0*-',I2,/,
910 #15X,'CLEAR SCREEN TO CONTINUE',//////////)
911
912 C-----
913 END
914
915 C=====
916 SUBROUTINE FZERO (NS,K2,K1,ITFX)
917 INTERACTIVELY PLOTS A POLE-ZERO PLOT USING DISSPLA GRAPHICS GIVEN
918 THE ZERO AND POLE LOCATIONS
919
920 C=====
921 REAL*8 OLN,CLD,ORNUM,GAN,CO,ZERO
922 REAL*4 XDC,YDC,XNO,YNO,XMIN,XMAX,YMIN,YMAX,XAXIS,YAXIS,PAGEX,
923 PAGEY,XX,YY,GOL,XL,YL,TS,TH,TH1,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10
924 INTEGER NS,IPTR,K1,K2,NN,NL,LINES,IANS,ICHG,LINE,J1,ORD,LL,
925 IPT,IPP,IZ,R,ITFX
926 COMMON /GRAFF/ OLN(55,2,12,12),OLD(99,2),GAN(2,12,12),CC
927 COMMON /SGRAF/ XDO(500),YDO(500),XNO(500),YNO(500)
928 COMMON /SHEAD/ LINES(4,18),LINE(9)
929 COMMON /SCALE/ TS,TH,TH1,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10
930 DATA IV,Y/,I2,N/,
931
932 C-----
933 CALL FRTCMS('CLRSCRN')
934 WRITE(6,12C)
935 IF(ITFX.EQ.1) WRITE(6,710)
936 IF(ITFX.EQ.2) WRITE(6,711)
937 IF(ITFX.EQ.3) WRITE(6,712)
938 CALL PRTPLOT(IPP)
939 GO TO (201,202,80),IPP
940
941 C-----
942 J1 = 0
943 CC 5 LL = 1,NS
944 XNO(LL) = C.O
945 YNO(LL) = C.O
946 XDO(LL) = SAGL(OLD(LL,1))
947 YDO(LL) = SAGL(OLD(LL,2))
948 IF(DABS(OLN(LL,1,K2,K1)).LE.CO).AND.(DABS(OLN(LL,2,K2,K1)).LE.CO)
949 *) GO TO 6
950 J1 = J1+1
951 XNO(J1) = SAGL(OLN(LL,1,K2,K1))
952 YNO(J1) = SAGL(OLN(LL,2,K2,K1))

```

```

CONTINUE
ORD = DINT (GAN(1,K2,K1))
IF (ORD.GT.J1) J1 = ORD
GOL = SNGL (GAN(2,K2,K1))

C-----PLOT SETUP-----
4C
ICHG = 0
CALL FRTCMS('CLRSCRN')
IF (ICHG.EQ.1) GO TO 41
WRITE(6,120)
IF (ITFX.EQ.1) WRITE(6,710)
IF (ITFX.EQ.2) WRITE(6,711)
IF (ITFX.EQ.3) WRITE(6,712)
CALL GSETUP(IPTR,PAGEX,PAGEY)
IF (ICHG.EQ.1) GC TO 50
5C
CALL FRTCMS('CLRSCRN')
CALL GGRAF(XMIN,XMAX,YMIN,YMAX)
6C
IF (ICHG.EQ.1) GO TO 50
CALL FRTCMS('CLRSCRN')
IF (ICHG.EQ.1) GO TO 90
7C
CONTINUE
IF (IPTR.EQ.2) GO TO 10
C-----TEK 618-----
CALL TEK618
CALL PAGE(PAGEX,PAGEY)
CALL HWRGT('AUTO')
CALL HWSCL('SCREEN')
CALL NCBDR
GC TO 20
C-----COMPRS-----
1C
CONTINUE
CALL COMPRS
CALL PAGE(PAGEX,PAGEY)
2C
CONTINUE
C-----SCALING CONSTANTS-----
CALL PSCL(PAGEX)
C-----PLOT EXECUTION-----
CALL NOCHEK
CALL GRACE(C)
CALL PHYSOR(T1,T1)
XAXIS = PAGEX - 2.*T1
YAXIS = PAGEY - T1 - T2
CALL AREA2C(XAXIS, YAXIS)
CALL TRIPLX
CALL HEIGHT(TH)
CALL XNAME(' $',100)
CALL HEIGHT(TH)

```

OPT02890
OPT02900
OPT02910
OPT02920
OPT02930
OPT02940
OPT02950
OPT02960
OPT02970
OPT02980
OPT02990
OPT03000
OPT03010
OPT03020
OPT03030
OPT03040
OPT03050
OPT03060
OPT03070
OPT03080
OPT03090
OPT03100
OPT03110
OPT03120
OPT03130
OPT03140
OPT03150
OPT03160
OPT03170
OPT03180
OPT03190
OPT03200
OPT03210
OPT03220
OPT03230
OPT03240
OPT03250
OPT03260
OPT03270
OPT03280
OPT03290
OPT03300
OPT03310
OPT03320
OPT03330
OPT03340
OPT03350
OPT03360

```

CALL YNAME(' $',100)
CALL GRAF(XMIN,SCALE',XMAX,YMIN,SCALE',YMAX)
XL = (XMAX-XMIN)*0.1
YL = (YMAX-YMIN)*0.1
CALL RLVEC(XMAX+XL,0.0,XMIN-XL,0.0,0000)
CALL RLVEC(C.0,YMAX+YL,0.0,YMIN-YL,0000)
HEADING
C-----
NN = NL + 1
IF(NL.EQ.0) GO TO 35
DC 35 I=1,NL
CALL CLINE(I)
CALL HEADIN(LINE,100,T3,NN)
35 CONTINUE
IF(ITFX.EQ.1)
* CALL HEADIN('OPEN LOOP TRANSFER POLE-ZERO MAP$',100,T3,NN)
* IF(ITFX.EQ.2)
* CALL HEADIN('NOISE TRANSFER POLE-ZERO MAP$',100,T3,NN)
* IF(ITFX.EQ.3)
* CALL HEADIN('COMPENSATOR TRANSFER POLE-ZERO MAP$',100,T3,NN)
C-----
CALL MARKER(4)
CALL SCLPIC(2)
CALL CURVE(XDO,YDO,NS,-1)
C-----
IF(J1.EQ.0) GO TO 36
CALL MARKER(16)
CALL SCLPIC(2)
CALL CURVE(XNO,YNO,J1,-1)
36 CONTINUE
C-----
XX = XAXIS - T7
YY = YAXIS + T8
CALL HEIGHT(TH1)
CALL MESSAG('INPUT # = $',100,XX+T8,YY+T10)
CALL HEIGHT(TH1)
CALL INTNO(K1,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAG('OUTPUT # = $',100,XX+T8,YY+T9)
CALL HEIGHT(TH1)
CALL INTNO(K2,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAG('DC GAIN = $',100,XX+T8,YY+T8)
CALL HEIGHT(TH1)
CALL REALNC(GOL,-3,'ABUT','ABUT')
CALL BLREC(XX,YY,T5,T6,3.32)
C-----
CALL DCT
C-----
GRIC
C-----
OPT03270
OPT03280
OPT03290
OPT03300
OPT03310
OPT03320
OPT03330
OPT03340
OPT03350
OPT03360
OPT03370
OPT03380
OPT03390
OPT03400
OPT03410
OPT03420
OPT03430
OPT03440
OPT03450
OPT03460
OPT03470
OPT03480
OPT03490
OPT03500
OPT03510
OPT03520
OPT03530
OPT03540
OPT03550
OPT03560
OPT03570
OPT03580
OPT03590
OPT03600
OPT03610
OPT03620
OPT03630
OPT03640
OPT03650
OPT03660
OPT03670
OPT03680
OPT03690
OPT03700
OPT03710
OPT03720
OPT03730
OPT03740
OPT03750
OPT03760
OPT03770
OPT03780
OPT03790
OPT03800
OPT03810
OPT03820
OPT03830
OPT03840

```

```

CALL GRID(1,1)
CALL RESET('DOT')
CALL ENDPL(0)
IF(IPTR.NE.2) GO TO 50
CALL FRTCMS('CLRSCRN ')
WRITE(6,130)
-----CHANGES TO FLCT-----
5C
CCNT INUE
CALL FRTCMS('CLRSCRN ')
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
IF(ITFX.EQ.3) WRITE(6,712)
IF(ITE(6,100))
WRITE(6,110)
CALL RDINT(IANS)
IF(IANS.GT.500.IANS.LT.1) GO TO 15
GO TO 25
WRITE(6,510)
GO TO 30
CCNT INUE
ICHG = 1
GO TO (40,50,60,70,200),IANS
-----TABULAR DATA-----
202
CCNT INUE
J1 = 0
CRD = 0
DINT(GAN(1,K2,K1))
ZERO = 0
CALL FRTCMS('CLRSCRN ')
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
IF(ITFX.EQ.3) WRITE(6,712)
CALL PRTR(IPT)
CALL FRTCMS('CLRSCRN ')
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(IPT,710)
IF(ITFX.EQ.2) WRITE(IPT,711)
IF(ITFX.EQ.3) WRITE(IPT,712)
K1
K2
NS
WRITE(IPT,300)
WRITE(IPT,301)
WRITE(IPT,302)
WRITE(IPT,303)
WRITE(IPT,304)
WRITE(IPT,305)
WRITE(IPT,306)
WRITE(IPT,307)
WRITE(IPT,308)
WRITE(IPT,309)
WRITE(IPT,310)
WRITE(IPT,311)
WRITE(IPT,312)
WRITE(IPT,313)
WRITE(IPT,314)
WRITE(IPT,315)
WRITE(IPT,316)
WRITE(IPT,317)
WRITE(IPT,318)
WRITE(IPT,319)
WRITE(IPT,320)
WRITE(IPT,321)
WRITE(IPT,322)
WRITE(IPT,323)
WRITE(IPT,324)
WRITE(IPT,325)
WRITE(IPT,326)
WRITE(IPT,327)
WRITE(IPT,328)
WRITE(IPT,329)
WRITE(IPT,330)
WRITE(IPT,331)
WRITE(IPT,332)
WRITE(IPT,333)
WRITE(IPT,334)
WRITE(IPT,335)
WRITE(IPT,336)
WRITE(IPT,337)
WRITE(IPT,338)
WRITE(IPT,339)
WRITE(IPT,340)
DO 210 I = 1,NS
WRITE(IPT,350) OLD(I,1),OLD(I,2)
IF(ORD.EQ.0) GO TO 220
21C

```

```

110 WRITE(IPT,310) ORD
120 WRITE(IPT,350)
130 WRITE(IPT,340)
140 DO 220 I=1,INS
150 IF((DABS(OLN(I,1,K2,K1)).LE.CG).AND.(DABS(OLN(I,2,K2,K1)).LE.CO)
160 *) GO TO 220
170 J1=J1+1
180 WRITE(IPT,320) CLN(I,1,K2,K1),OLN(I,2,K2,K1)
190 CONTINUE
200 IZR=ORD - J1
210 IF (IZR.LE.C) GO TO 240
220 DO 225 I=1,IZR
230 WRITE(IPT,330) ZERO, ZERO
240 GO TO 240
250 WRITE(IPT,350)
260 WRITE(IPT,360)
270 -----CHANGES FOR TABULAP DATA-----
280 IF (IPT.EQ.2) WRITE (6,401)
290 IF (IPT.EQ.2) WRITE (6,402)
300 WRITE (6,400)
310 CALL RCHAR (IANS)
320 IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 245
330 GO TO 250
340 WRITE (6,500)
350 GO TO 240
360 CONTINUE
370 IF (IANS.EC.IY) GO TO 202
380 IF (IANS.EC.IZ) GO TO 200
390 RETURN
400 -----
410 FORMAT (//,5X,'DO YOU DESIRE TO MAKE ANY CHANGES TO:',//,10X,'1.
420 *LOTTER / PAGE SIZE',//,10X,'2. GRAPH LIMITS',//,10X,'3. HEADING',//,10X,'4.
430 *CHANGES - EXIT POLE-ZERO PLOTTING ROUTINE',//,10X,'5.
440 *POLE-ZERO MAP',//,10X,'6. DISPLA METAFILE HAS BEEN CREATED',//,
450 *CLEAR SCREEN TO CONTINUE',//,10X,'7. INPUT NUMBER =',15)
460 FORMAT (//,5X,'ENTER OPTION NUMBER.',//)
470 FORMAT (//,10X,'POLE-ZERO MAP',//,10X,'1. DISPLA METAFILE HAS BEEN CREATED',//,
480 *CLEAR SCREEN TO CONTINUE',//,10X,'2. INPUT NUMBER =',15)
490 FORMAT (//,10X,'NUMERATOR ORDER =',15)
500 FORMAT (//,10X,'DENOMINATOR ORDER =',15)
510 FORMAT (//,10X,'TRANSFER FUNCTION (DC) GAIN =',D12.4,/)
520 FORMAT (//,10X,'REAL LOCATIONS',//)
530 FORMAT (//,10X,'IMAGINARY PART',//)
540 FORMAT (//,5X,'POLE LOCATIONS',//)
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```

```

3EQ  FORMAT(/,15X,'NO REAL ZEROS',/) CHANGE OUTPUT DEVICE?
400  *//,10X,TYPE "YES" OR "NO",/)
401  FORMAT(///,5X,'OUTPUT SENT TO PRINTER',/)
402  FORMAT(///,5X,'OUTPUT SENT TO LISTING FILE',/)
500  *3X,ENTER "YES" OR "NO",/)
510  *3X,ENTER "YES" OR "NO",/)
710  FORMAT(5X,'OPEN LOCP TRANSFER FUNCTION',/)
711  FORMAT(5X,'NOISE TRANSFER FUNCTION',/)
712  FORMAT(5X,'COMPENSATOR TRANSFER FUNCTION',/)
C-----
C= END
C=====
C SLBROUTINE FTLO (NS,K2,K1,ITFX)
C INTERACTIVELY PLOTS A ROOT-LOCUS PLOT USING CISSPLA GRAPHICS GIVEN
C THE ZERO AND POLE LOCATIONS
C=====
C REAL*8 OLD,CLN,MAG,QI,QR,QPSR,KMAX,KMIN,ZERO,GAIN,
C *DELTF,GAN,CC,XR,YI,CRNUM,K,KIC
C *REAL*4 XDC,YDC,XNO,YNO,XMIN,XMAX,YMIN,YMAX,XAXIS,YAXIS,PAGEX,
C *PAGEY,XX,YY,GOL,XL,YL,TS,TH,T1,T2,T3,T4,T5,T6,T7,T8,TH1,T9,T10
C *CMPLX*16 RTN,PLN,RTCL,PLD,CPOLN,MN,CPOLD,CPOL
C *INTEGER NS,IPTR,II,JJ,NN,NL,LINES,IANS,ICPG,LINE,J1,ORD,ITFX,
C *IL,N,N1,N2,K2,K1,NN1,NN2,ND1,ND2,JLL,IER,PTS,IP,T,IZR
C *COMMON /GRAFF/ XDO(500),YDO(500),XNO(500),YNO(500)
C *COMMON /SHEAD/ LINES(4,18),LINE(9)
C *COMMON /SHEEQ/ RTN(99),PLN(99),PLD(99),WN(500),QI(500),
C *GR(500),QPSR(500),MAG(500)
C *COMMON /SCALE/ TS,TH,TH1,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10
C DATA IY,Y,I,Z,N,/,
C-----
C 200  CALL FRTCMS('CLRSCRN',)
C WRITE(6,12C)
C IF(ITFX.EQ.1) WRITE(6,710)
C IF(ITFX.EQ.2) WRITE(6,711)
C IF(ITFX.EQ.3) WRITE(6,712)
C CALL PRIPLOT(IPP)
C GO TO (201,202,80),IPP
C-----
C 201  CCNTINUE
C ICHG = 0
C CALL FRTCMS('CLRSCRN',)
C CALL ICHG.EQ.1) GC TC 41
C 40  WRITE(6,12C)
C IF(ITFX.EQ.1) WRITE(6,710)

```

```

41 IF(ITFX.EQ.2) WRITE(6,711)
   IF(ITFX.EQ.3) WRITE(6,712)
   CALL GSETUP(IPTR,PAGEX,PAGEY)
5C IF(ICHG.EQ.1) GO TO 90
   CALL FRTCMS(,CLRSCRN, ,)
55 CALL GGRAF(XMIN,XMAX,YMIN,YMAX)
   IF(ICHG.EQ.1) GO TO 90
   CALL FRTCMS(,CLRSCRN, ,)
6C CALL RANGK(KMIN,KMAX)
   IF(ICHG.EQ.1) GO TO 90
   CALL FRTCMS(,CLRSCRN, ,)
7C CALL GHEAD(NL)
   IF(ICHG.EQ.1) GO TO 90
   CONTINUE
   IF(IPTR.EQ.2) GO TO 10
   -----TEK 618-----
   CALL TEK61E
   CALL PAGE(PAGEX,PAGEY)
   CALL HWRGT(,AUTO, ,)
   CALL HWSCL(,SCREEN, ,)
   CALL NCBRDF
   GO TO 20
   -----COMFRS-----
1C CONTINUE
   CALL COMPRS
   CALL PAGE(PAGEX,PAGEY)
2C CONTINUE
   -----SCALING CONSTANTS-----
   CALL PSCALE(PAGEX)
   -----PLOT EXECUTION-----
   CALL NCHEK
   CALL GRACE(C, ,)
   CALL PHYSOR(T1,T1)
   CALL XAXIS = PAGEY - T1 - T2
   CALL YAXIS = PAGEY - T1 - T2
   CALL AREA2C(XAXIS, YAXIS)
   CALL TRIPLX
   CALL HEIGHT(TH)
   CALL XNAME(, $, ,100)
   CALL YNAME(, $, ,100)
   CALL GGRAF(XMIN,SCALE',XMAX,YMIN,SCALE',YMAX)
   XL = (XMAX-XMIN)*0.1
   YL = (YMAX-YMIN)*0.1
   CALL RLVEC(XMAX+XL,0.0,XMIN-XL,0.0,0000)
   CALL PLVEC(C.0,YMAX+YL,0.0,YMIN-YL,0000)
   -----HEADING-----
   NN = NL + 1

```

```

OPT05290
OPT05300
OPT05310
OPT05320
OPT05330
OPT05340
OPT05350
OPT05360
OPT05370
OPT05380
OPT05390
OPT05400
OPT05410
OPT05420
OPT05430
OPT05440
OPT05450
OPT05460
OPT05470
OPT05480
OPT05490
OPT05500
OPT05510
OPT05520
OPT05530
OPT05540
OPT05550
OPT05560
OPT05570
OPT05580
OPT05590
OPT05600
OPT05610
OPT05620
OPT05630
OPT05640
OPT05650
OPT05660
OPT05670
OPT05680
OPT05690
OPT05700
OPT05710
OPT05720
OPT05730
OPT05740
OPT05750
OPT05760

```

```

IF(NL.EQ.0) GO TO 35
DC 35 I=1,NL
CALL CLINE(I)
CALL HEADIN(LINE,100,T3,NN)
35 CONTINUE
IF(ITFX.EQ.1)
* CALL HEADIN('ROOT-LOCUS PLOT (OPEN LOOP TF)',100,T3,NN)
* IF(ITFX.EQ.2)
* CALL HEADIN('ROOT-LOCUS PLOT (NCISE TF)',100,T3,NN)
* IF(ITFX.EQ.3)
* CALL HEADIN('ROOT-LOCUS PLOT (COMPENSATOR TF)',100,T3,NN)
C-----INPUT ARRAYS-----
ZERO = 0.0
J1 = 0
ND1 = NS + 1
ND2 = NS + 2
DC 5 LL=1,NS
XNO(LL) = C.O
YNO(LL) = C.O
QT(LL) = 0.0
GR(LL) = 0.0
XDO(LL) = SGL(OLD(LL,1))
YDO(LL) = SGL(OLD(LL,2))
RTD(LL) = ECPLX(OLD(LL,1),OLD(LL,2))
RTN(LL) = ECPLX(ZERC,ZERO)
IF((DABS(OLN(LL,1,K2,K1)).LE.CO).AND.(DABS(OLN(LL,2,K2,K1)).LE.CO))
* GO TO 6
J1 = J1+1
XNO(J1) = SGL(OLN(LL,1,K2,K1))
YNO(J1) = SGL(OLN(LL,2,K2,K1))
RTN(J1) = ECPLX(OLN(LL,1,K2,K1),OLN(LL,2,K2,K1))
CCNTINUE
ORD = DINT(GAN(1,K2,K1))
IF(ORD.GT.J1) J1 = ORD
GOL = SGL(GAN(2,K2,K1))
GAIN = GAN(2,K2,K1)
C-----POLE-----
CALL MARKER(4)
CALL SCLPIC(2,1)
CALL CURVE(XDO,YDO,NS,-1)
C-----ZERO-----
IF(J1.EQ.0) GO TO 36
CALL MARKER(16)
CALL SCLPIC(2,1)
CALL CURVE(XNO,YNO,J1,-1)
36 CONTINUE
C-----POLYNOMIALS-----

```

OPT05770
OPT05780
OPT05790
OPT05800
OPT05810
OPT05820
OPT05830
OPT05840
OPT05850
OPT05860
OPT05870
OPT05880
OPT05890
OPT05900
OPT05910
OPT05920
OPT05930
OPT05940
OPT05950
OPT05960
OPT05970
OPT05980
OPT05990
OPT06000
OPT06010
OPT06020
OPT06030
OPT06040
OPT06050
OPT06060
OPT06070
OPT06080
OPT06090
OPT06100
OPT06110
OPT06120
OPT06130
OPT06140
OPT06150
OPT06160
OPT06170
OPT06180
OPT06190
OPT06200
OPT06210
OPT06220
OPT06230
OPT06240

C	CALL MAKPOL(NS,RTD,PLD)	DENOMENATOR	OPT06250
	CI(1) = 1.0		OPT06260
	DO 7 I = 2,ND1		OPT06270
7	QI(I) = CREAL(PLD(ND2-I))		OPT06280
C		NUMERATOR	OPT06290
	IF(J1.EQ.0) GO TO 9		OPT06300
	CALL MAKPOL(J1,RTN,PLN)		OPT06310
	JJ = NS - J1 + 2		OPT06320
	QR(JJ-1) = 1.0		OPT06330
	DO 8 I = JJ,ND1		OPT06340
8	QR(I) = CREAL(PLN(ND2-I))		OPT06350
	GO TO 11		OPT06360
9	QR(ND1) = 1.0		OPT06370
	QR(1) = 0.0		OPT06380
C		ROOT FINDING ROUTINE	OPT06390
11	KIC = (KMAX-KMIN)/2000.		OPT06400
	K = KMIN		OPT06410
	DC 52 J = 1,2000		OPT06420
	DO 37 II = 1,ND1		OPT06430
	MAG(II) = CI(II) + K*GAIN*QR(II)		OPT06440
27	CONTINUE		OPT06450
	CALL ZRPOLY(MAG,NS,MN,IER)		OPT06460
	DO 51 II = 1,NS		OPT06470
	XR = DREAL(MN(II))		OPT06480
	XND(II) = SNGL(XR)		OPT06490
	YI = DIMAG(MN(II))		OPT06500
	YND(II) = SNGL(YI)		OPT06510
51	CONTINUE		OPT06520
	CALL MARKER(4)		OPT06530
	CALL SCLPIC(1.0)		OPT06540
	CALL CURVE(XND,YND,NS,-1)		OPT06550
	K=K+KIC		OPT06560
52	CONTINUE	LEGEND	OPT06570
	XX = XAXIS - T7		OPT06580
	YY = YAXIS + T8		OPT06590
	CALL HEIGHT(TH1)		OPT06600
	CALL MESSAG('INPUT # = \$',100,XX+T8,YY+T10)		OPT06610
	CALL HEIGHT(TH1)		OPT06620
	CALL INTNO(K1,'ABUT','ABUT')		OPT06630
	CALL HEIGHT(TH1)		OPT06640
	CALL MESSAG('OUTPUT # = \$',100,XX+T8,YY+T5)		OPT06650
	CALL HEIGHT(TH1)		OPT06660
	CALL INTNO(K2,'ABUT','ABUT')		OPT06670
	CALL HEIGHT(TH1)		OPT06680
	CALL MESSAG('DC GAIN = \$',100,XX+T8,YY+T8)		OPT06690
	CALL HEIGHT(TH1)		OPT06700
	CALL MESSAG('DC GAIN = \$',100,XX+T8,YY+T8)		OPT06710
	CALL HEIGHT(TH1)		OPT06720

```

C-----GRID-----
CALL REALNC(GOL,-3,'ABUT','ABUT')
CALL BLREC(XX,YY,15,16,0.02)
CALL DOT
CALL GRID(1,1)
CALL RESET('DOT')
CALL ENOPL(C)
IF(IPTR.NE.2) GO TO 90
CALL FRTCMS('CLRSCRN')
WRITE(6,13C)
C-----CHANGES TO PLOT-----
3C CONTINUE
CALL FRTCMS('CLRSCRN')
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
IF(ITFX.EQ.3) WRITE(6,712)
WRITE(6,110C)
CALL PCINT(IANS)
IF(IANS.GT.60R.IANS.LT.1) GO TO 15
15 GO TO 25
25 CONTINUE
ICHG = 1
GO TO (40,5C,60,55,7C,200),IANS
C-----TABULAR DATA-----
2C2 CONTINUE
ICHG = 0
CALL FRTCMS('CLRSCRN')
IF(ICHG.EQ.1) GO TO 249
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
IF(ITFX.EQ.3) WRITE(6,712)
CALL PTR(IPT)
IF(ICHG.EQ.1) GO TO 29C
CALL FRTCMS('CLRSCRN')
CALL RANGK(KMIN,KMAX)
IF(ICHG.EQ.1) GO TO 29D
252 CALL FRTCMS('CLRSCRN')
253 CALL NPTS(PTS)
IF(ICHG.EQ.1) GO TO 29D
C-----TABULATE DATA-----
254 CONTINUE
ZERO = 0.0
J1 = 0

```

```

ND1 = NS + 1
ND2 = NS + 2
ORD = DINT (CAN(1,K2,K1))
CALL FRTCM5(,CLRSCRN *)
WRITE(IPT,120) WRITE(IPT,710)
IF(ITFX.EQ.1) WRITE(IPT,711)
IF(ITFX.EQ.2) WRITE(IPT,712)
WRITE(IPT,300) K1
WRITE(IPT,301) K2
WRITE(IPT,320) CAN(2,K2,K1)
IF(ITFX.EQ.1) WRITE(IPT,121)
IF(ITFX.EQ.2) WRITE(IPT,122)
IF(ITFX.EQ.3) WRITE(IPT,123)
WRITE(IPT,311) NS
WRITE(IPT,340)
WC 210 I = 1, NS
DO 220 I = 1, NS
  RTD(I) = DCPLX(OLD(I,1),OLD(I,2))
  IF(ORD.EQ.0) GO TO 230
  WRITE(IPT,310) ORD
  WRITE(IPT,320)
  WRITE(IPT,340)
  DO 220 I = 1, NS
    RTD(I) = DCPLX(OLD(I,1),OLD(I,2))
    IF((DABS(OLD(I,1,K2,K1)).LE.CC).AND.(DABS(OLD(I,2,K2,K1)).LE.CC)
      * ) GO TO 22C
    J1 = J1 + 1
    RTN(J1) = CCPLX(OLD(I,1,K2,K1),OLD(I,2,K2,K1))
    CCNTINUE
    IZR = ORD - J1
    IF (IZR.LE.C) GO TO 240
    DO 225 I = 1, IZR
      CCNTINUE
      WRITE(IPT,330) ZERO, ZERO
      GO TO 240
    WRITE(IPT,350)
    WRITE(IPT,360)
  C-----POLYNOMIALS-----
  C-----DENOMINATOR-----
  C-----NUMERATOR-----
240 CONTINUE
CALL MAKPOL(NS,RTD,PLD)
QI(1) = 1.C
CC 207 I = 2,ND1
CC 207 I = 2,ND1
  C-----POLYNOMIALS-----
  C-----DENOMINATOR-----
  C-----NUMERATOR-----

```

```

IF(J1.EQ.0) GO TO 205
CALL MAKPOL(J1,RTN,PLN)
JJ = NS - J1 + 2
QP(JJ-1) = 1.0
DO 208 I = JJ,ND1
  QR(I) = CREAL(PLN(ND2-I))
GO TO 211
208 QR(ND1) = 1.0
209 QR(1) = 0.0
C-----ROOT FINDING ROUTINE-----
211 WRITE(IPT,220)
    KIC = (KMAX-KMIN)/DFLCAT(PTS)
    K = KMIN
    DO 262 J = 1,PTS
      DO 237 II = 1,ND1
        MAG(II) = CI(II) + K*GAIN*QR(II)
      CCNT INUE
      CALL ZRPOLY(MAG,NS,WA,IER)
      WRITE(IPT,221) K
      DO 261 II = 1,NS
        XR = DREAL(WN(II))
        YI = DIMAG(WN(II))
      WRITE(IPT,220) XR,YI
      CCNT INUE
      K=K+KIC
      CONTINUE
261
262 C-----CHANGES TABULAR DATA-----
241 IF(IPT.EQ.2) WRITE (6,401)
    IF(IPT.EQ.2) WRITE (6,402)
    WRITE(6,400)
    CALL RDCHAR (IANS)
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 245
    GO TO 250
245 WRITE (6,500)
    GO TO 241
250 CCNT INUE
    IF (IANS.EQ.IY) GO TO 292
    IF (IANS.EC.IZ) GO TO 200
C-----CHANGES-----
290 CCNT INUE
    CALL FRTCMS('CLRSCRN ')
    WRITE(6,120)
    IF(ITFX.EQ.01) WRITE(6,710)
    IF(ITFX.EQ.02) WRITE(6,711)
    IF(ITFX.EQ.03) WRITE(6,712)
    WRITE(6,410)
291

```



```

6C      CALL FRTCMS('CLRSCRN ')
      CALL GHEAD(NL)
      IF(I,CHG,EQ,1) GO TO 90
      CONTINUE
      -----SCALING CONSTANTS-----
      OPT09120
      OPT09140
      OPT09150
      OPT09160
      OPT09170
      OPT09180
      OPT09190
      OPT09200
      OPT09210
      OPT09220
      OPT09230
      OPT09240
      OPT09250
      OPT09260
      OPT09270
      OPT09280
      OPT09290
      OPT09300
      OPT09310
      OPT09320
      OPT09330
      OPT09340
      OPT09350
      OPT09360
      OPT09370
      OPT09380
      OPT09390
      OPT09400
      OPT09410
      OPT09420
      OPT09430
      OPT09440
      OPT09450
      OPT09460
      OPT09470
      OPT09480
      OPT09490
      OPT09500
      OPT09510
      OPT09520
      OPT09530
      OPT09540
      OPT09550
      OPT09560
      OPT09570
      OPT09580
      OPT09590
      OPT09600

C      CALL PSCALE(PAGEX)
      -----INPUT AFRAYS-----
      OPT09120
      OPT09140
      OPT09150
      OPT09160
      OPT09170
      OPT09180
      OPT09190
      OPT09200
      OPT09210
      OPT09220
      OPT09230
      OPT09240
      OPT09250
      OPT09260
      OPT09270
      OPT09280
      OPT09290
      OPT09300
      OPT09310
      OPT09320
      OPT09330
      OPT09340
      OPT09350
      OPT09360
      OPT09370
      OPT09380
      OPT09390
      OPT09400
      OPT09410
      OPT09420
      OPT09430
      OPT09440
      OPT09450
      OPT09460
      OPT09470
      OPT09480
      OPT09490
      OPT09500
      OPT09510
      OPT09520
      OPT09530
      OPT09540
      OPT09550
      OPT09560
      OPT09570
      OPT09580
      OPT09590
      OPT09600

      GOL = SNGL(GAN(2,K2,K1))
      FL = FMIN*0.1
      FU = FMAX*0.1
      XAXIS = PAGEX - 2.*T1
      YAXIS = PAGEX - T1 - T2
      PCYCLE = XAXIS/(ALOG10(FMAX)-ALOG10(FMIN))
      CALL FREQ(NS,K2,K1,FMIN,FMAX,N)
      DO 5 I = 1,N
      CMGA = 1/MAG(WN(I))
      XND(I) = SNGL(OMEGA)
      PA = 20.0*LOG10(MAG(I))
      YND(I) = SNGL(MA)
      YDO(I) = SNGL(QPSR(I))
      CONTINUE
      CALL MINMAX(N,YNO,MMIN,MMAX)
      MMIN = AINT(MMIN) - 1.0
      PCYCLE = (PMAX-MMIN)/YAXIS
      MCYCLE = AINT(MCYCLE) + 1.0
      CALL MINMAX(N,YDO,PMIN,PMAX)
      PMIN = AINT(PMIN) - 1.0
      PCYCLE = (FMAX-PMIN)/YAXIS
      PCYCLE = AINT(PCYCLE) + 1.0
      -----MAGNITUDE PLOT-----
      OPT09120
      OPT09140
      OPT09150
      OPT09160
      OPT09170
      OPT09180
      OPT09190
      OPT09200
      OPT09210
      OPT09220
      OPT09230
      OPT09240
      OPT09250
      OPT09260
      OPT09270
      OPT09280
      OPT09290
      OPT09300
      OPT09310
      OPT09320
      OPT09330
      OPT09340
      OPT09350
      OPT09360
      OPT09370
      OPT09380
      OPT09390
      OPT09400
      OPT09410
      OPT09420
      OPT09430
      OPT09440
      OPT09450
      OPT09460
      OPT09470
      OPT09480
      OPT09490
      OPT09500
      OPT09510
      OPT09520
      OPT09530
      OPT09540
      OPT09550
      OPT09560
      OPT09570
      OPT09580
      OPT09590
      OPT09600

      IF(I,PTR,EQ,2) GO TO 10
      CALL TEK61E
      CALL PAGE(PAGEX,PAGEY)
      CALL HWRDT('AUTO')
      CALL HNSCAL('SCREEN')
      CALL NOBRDR
      GO TO 20
      -----COMPRS-----
      OPT09120
      OPT09140
      OPT09150
      OPT09160
      OPT09170
      OPT09180
      OPT09190
      OPT09200
      OPT09210
      OPT09220
      OPT09230
      OPT09240
      OPT09250
      OPT09260
      OPT09270
      OPT09280
      OPT09290
      OPT09300
      OPT09310
      OPT09320
      OPT09330
      OPT09340
      OPT09350
      OPT09360
      OPT09370
      OPT09380
      OPT09390
      OPT09400
      OPT09410
      OPT09420
      OPT09430
      OPT09440
      OPT09450
      OPT09460
      OPT09470
      OPT09480
      OPT09490
      OPT09500
      OPT09510
      OPT09520
      OPT09530
      OPT09540
      OPT09550
      OPT09560
      OPT09570
      OPT09580
      OPT09590
      OPT09600

      CCNT INUE
      CALL CCMPRS
      CALL PAGE(PAGEX,PAGEY)
      CCNT INUE
      -----PLOT EXECUTION-----
      OPT09120
      OPT09140
      OPT09150
      OPT09160
      OPT09170
      OPT09180
      OPT09190
      OPT09200
      OPT09210
      OPT09220
      OPT09230
      OPT09240
      OPT09250
      OPT09260
      OPT09270
      OPT09280
      OPT09290
      OPT09300
      OPT09310
      OPT09320
      OPT09330
      OPT09340
      OPT09350
      OPT09360
      OPT09370
      OPT09380
      OPT09390
      OPT09400
      OPT09410
      OPT09420
      OPT09430
      OPT09440
      OPT09450
      OPT09460
      OPT09470
      OPT09480
      OPT09490
      OPT09500
      OPT09510
      OPT09520
      OPT09530
      OPT09540
      OPT09550
      OPT09560
      OPT09570
      OPT09580
      OPT09590
      OPT09600

      CALL NCCHK
      CALL GRACE(C)
      CALL PHYSOR(T1,T1)
      CALL AREA2C(XAXIS, YAXIS)

```

```

CALL TRIPLX
CALL HEIGHT(TH)
CALL XNAME('FREQUENCY - RADIANS PER SECOND $',100)
CALL HEIGHT(TH)
CALL YNAME('MAGNITUDE - DECIBELS $',100)
CALL XLOG(FMIN,FCYCLE,MMIN,MCYCLE)
CALL RLVEC(FMAX+FU,0.0,FMIN-FL,0.0,0000)
C-----HEADING-----
NN = NL + 1
IF(NL.EQ.0) GO TO 35
DO 35 I=1,NL
CALL CLINE(I)
CALL HEADIN(LINE,100,T3,NN)
35 CONTINUE
IF(ITFX.EQ.1)
* CALL HEADIN('OPEN LOOP TF BODE MAGNITUDE$',100,T3,NN)
* IF(ITFX.EQ.2)
* CALL HEADIN('NOISE TF BODE MAGNITUDE$',100,T3,NN)
* IF(ITFX.EQ.3)
* CALL HEADIN('COMPENSATOR TF BODE MAGNITUDE$',100,T3,NN)
C-----MAGNITUDE-----
CALL CURVE(XNO,YNO,N,0)
C-----LEGEND-----
XX = XAXIS - T7
YY = YAXIS + T8
CALL HEIGHT(TH1)
CALL MESSAG('INPUT # = $',100,XX+T8,YY+T10)
CALL HEIGHT(TH1)
CALL INTNO(K1,'ABUT','ABUT')
CALL MESSAG('OUTPUT # = $',100,XX+T8,YY+T9)
CALL HEIGHT(TH1)
CALL INTNO(K2,'ABUT','ABUT')
CALL MESSAG('DC GAIN = $',100,XX+T8,YY+T8)
CALL HEIGHT(TH1)
CALL RFALNC(GOL,-3,'ABUT','ABUT')
CALL BLREC(XX,YY,T5,T6,0.02)
C-----GRID-----
CALL DOT
CALL GPID(1,1)
CALL RESET('DOT')
CALL ENDPL(C)
C-----PHASE PLOT-----
IF(IPTR.EQ.2) GO TO 11
CALL TEK618
CALL PAGE(PAGEX,PAGEY)

```

OPT09610
OPT09620
OPT09630
OPT09640
OPT09650
OPT09660
OPT09670
OPT09680
OPT09690
OPT09700
OPT09710
OPT09720
OPT09730
OPT09740
OPT09750
OPT09760
OPT09770
OPT09780
OPT09790
OPT09800
OPT09810
OPT09820
OPT09830
OPT09840
OPT09850
OPT09860
OPT09870
OPT09880
OPT09890
OPT09900
OPT09910
OPT09920
OPT09930
OPT09940
OPT09950
OPT09960
OPT09970
OPT09980
OPT09990
OPT10000
OPT10010
OPT10020
OPT10030
OPT10040
OPT10050
OPT10060
OPT10070
OPT10080

```

CALL HWROT('AUTO')
CALL HWSAL('SCREEN')
CALL NCBDRP
GO TO 21
-----COMFRS-----
11  CCNT INUE
    CALL CCMPRS
    CALL PAGE(PAGEX,PAGEY)
    CCNT INUE
-----PLOT EXECUTION-----
21  CALL NCHECK
    CALL GRACE(C)
    CALL PHYSOP(T1,T1)
    CALL AREA2C(XAXIS, YAXIS)
    CALL TRIPLX
    CALL HEIGHT(TH)
    CALL XNAME('FREQUENCY - RADIANS PER SECOND $',100)
    CALL YNAME('PHASE - DEGREES $',100)
    CALL XLOG(FMIN,FCYCLE,PMIN,PCYCLE)
    CALL RLVEC(FMAX+FU,180.0,FMIN-FL,180.0,0000)
    CALL RLVEC(FMAX+FU,-180.0,FMIN-FL,-180.0,0000)
    NN = NL + 1
    IF(NL.EQ.0) GO TO 36
    DO 36 I=1,NL
      CALL CLINE(I)
      CALL HEADIN(LINE,100,T3,NN)
26  CCNT INUE
    IF(ITFX.EQ.1)
      * CALL HEADIN('OPEN LOOP TF BODE PHASE$',100,T3,NN)
      * IF(ITFX.EQ.2)
      * CALL HEADIN('NOISE TF BODE PHASE$',100,T3,NN)
      * IF(ITFX.EQ.3)
      * CALL HEADIN('COMPENSATOR TF BODE PHASE$',100,T3,NN)
    CALL CURVE(XNC,YDO,N,0)
-----LEGEND-----
36  XX = XAXIS - T7
    YY = YAXIS + T8
    CALL HEISSAG('INPUT # = $',100,XX+T8,YY+T10)
    CALL HEIGHT(TH1)
    CALL INTND(K1,'ABUT')
    CALL HEISSAG('OUTPUT # = $',100,XX+T8,YY+T9)
    CALL HEIGHT(TH1)
    CALL INTND(K2,'ABUT')

```

```

OPT10090
OPT10100
OPT10110
OPT10120
OPT10130
OPT10140
OPT10150
OPT10160
OPT10170
OPT10180
OPT10190
OPT10200
OPT10210
OPT10220
OPT10230
OPT10240
OPT10250
OPT10260
OPT10270
OPT10280
OPT10290
OPT10300
OPT10310
OPT10320
OPT10330
OPT10340
OPT10350
OPT10360
OPT10370
OPT10380
OPT10390
OPT10400
OPT10410
OPT10420
OPT10430
OPT10440
OPT10450
OPT10460
OPT10470
OPT10480
OPT10490
OPT10500
OPT10510
OPT10520
OPT10530
OPT10540
OPT10550
OPT10560

```

```

CALL HEIGHT(TH1)
CALL MESSAG('DC GAIN =',100,XX+T8,YY+T8)
CALL HEIGHT(TH1)
CALL REALNC(GOL,-3,'ABUT','ABUT')
CALL BLREC(XX,YY,T5,T6,0.02)
C-----GRID-----
CALL DOT
CALL GRID(1,1)
CALL RESET('DOT')
CALL ENDPL(C)
IF(IPTR.NE.2) GO TO 75
CALL FRTCMS('CLRSCRN')
WRITE(6,13C)
C-----GAIN AND PHASE MARGIN-----
75 CALL FRTCMS('CLRSCRN')
WRITE(6,13E)
C-----GAIN MARGIN-----
DC 71 I = 2,N
IF(ABS(YDO(I-1)).LT.180.0.AND.ABS(YDO(I)).GE.180.0) GO TO 72
IF(ABS(YDO(I-1)).GT.180.0.AND.ABS(YDO(I)).LE.180.0) GO TO 72
GO TO 71
GM = 0.0 - YNO(I)
WRITE(6,14C) XNO(I), GM
CCNTINUE
72
71 C-----PHASE MARGIN-----
DC 76 I = 2,N
IF(YNO(I-1).GT.0.0.AND.YNO(I).LE.0.0) GO TO 77
GO TO 76
PM = 180.0 + YDO(I)
WRITE(6,15C) XNO(I), PM
CCNTINUE
76
C-----CHANGES TO PLOT-----
5C CCNTINUE
CALL FRTCMS('CLRSCRN')
CALL F(6,12C)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
IF(ITFX.EQ.3) WRITE(6,712)
WRITE(6,10C)
WRITE(6,11C)
CALL RLIN(1,1ANS)
IF(1ANS.GT.50R.1ANS.LT.1) GO TO 15
GC TO 25
WRITE(6,51C)
GO TO 30
CCNTINUE
15
25 ICHG = 1

```


245	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 245	OPTI 1530
	GO TO 250	OPTI 1540
250	WRITE (6,500)	OPTI 1550
	GO TO 240	OPTI 1560
	CONTINUE	OPTI 1570
	IF (IANS.EC.IY) GO TO 290	OPTI 1580
	IF (IANS.EC.IZ) GO TO 200	OPTI 1590
250	CONTINUE	OPTI 1600
	CALL FRICMS('CLRSCRN')	OPTI 1610
	WRITE(6,120)	OPTI 1620
	IF(ITFX.EQ.1) WRITE(6,710)	OPTI 1630
	IF(ITFX.EQ.2) WRITE(6,711)	OPTI 1640
	IF(ITFX.EQ.3) WRITE(6,712)	OPTI 1650
	WRITE(6,410)	OPTI 1660
251	WRITE(6,110)	OPTI 1670
	CALL RCINT(IANS)	OPTI 1680
	IF(IANS.GT.5.OR.IANS.LT.1) GO TO 292	OPTI 1690
252	GO TO 293	OPTI 1700
	WRITE(6,510)	OPTI 1710
253	GO TO 291	OPTI 1720
	CONTINUE	OPTI 1730
	ICHG = 1	OPTI 1740
	GO TO (251,252,253,254,200),IANS	OPTI 1750
250	RETURN	OPTI 1760
251	WRITE(6,520)	OPTI 1770
	RETURN	OPTI 1780
		OPTI 1790
100	FORMAT(//,5X,'DO YOU DESIRE TO MAKE ANY CHANGES TO:',//10X,'1',	OPTI 1800
	*LCTTER / PAGE SIZE',//10X,'2. GRAPH LIMITS (FREQUENCY RANGE)',	OPTI 1810
	*//10X,'3. HEADINGS',	OPTI 1820
	*//10X,'4. NC ADDITIONAL CHANGES - PLOT BCDE PLOTS',//10X,'5. NC	OPTI 1830
	*FANGES - EXIT BCDE SUBROUTINE',/)	OPTI 1840
	FORMAT(//,5X,'ENTER CPTICN NUMBER.',/)	OPTI 1850
110	FORMAT(//,15X,'BODE PLOT',)	OPTI 1860
120	FORMAT(//,15X,'15X',A DISSPLA METAFILE HAS BEEN CREATED',/,	OPTI 1870
130	*15X,'CLEAR SCREEN TO CONTINUE',//,15X,'A CONTINUE',//,15X,'15X',	OPTI 1880
	FORMAT(//,15X,'15X',GAIN MARGIN AND PHASE MARGIN',/)	OPTI 1890
135	FORMAT(//,15X,'15X',GAIN MARGIN AND PHASE MARGIN',/)	OPTI 1900
140	*GAIN MARGIN IS',FIC.4,' DB',	OPTI 1910
	FORMAT(//,3X,'PHASE CROSSEVER FREQUENCY IS',FIC.4,' RAD/SEC',1X,	OPTI 1920
150	*GAIN MARGIN IS',FIC.4,' DB',	OPTI 1930
	FORMAT(//,3X,'GAIN',FIC.4,' DEG',	OPTI 1940
160	*PHASE MARGIN IS',FIC.4,' DEG',	OPTI 1950
170	FORMAT(//,15X,'CLEAR SCREEN TO CONTINUE',//,15X,'15X',	OPTI 1960
180	FORMAT(//,15X,'15X',INPUT NUMBER =',15)	OPTI 1970
190	FORMAT(10X,'OUTPUT NUMBER =',15)	OPTI 1980
200	FORMAT(10X,'NUMERATOR ORDER =',15)	OPTI 1990
210	FORMAT(10X,'DENOMINATOR ORDER =',15)	OPTI 2000


```

C-----GRAPHIC OR TABULAR OUTPUT-----OPTI 2490
200 CALL FRTCMS('CLRSCRN ')OPTI 2500
    WRITE(6,120)OPTI 2510
    IF(IITFX.EQ.1) WRITE(6,710)OPTI 2520
    IF(IITFX.EQ.2) WRITE(6,711)OPTI 2530
    IF(IITFX.EQ.3) WRITE(6,712)OPTI 2540
    CALL PRTPLOT(IPP)OPTI 2550
    GO TO (201,202,80),IPPOPTI 2560
C-----PLOT SETUP-----OPTI 2570
201 CONTINUEOPTI 2580
    N = 500OPTI 2590
    ICHG = 0OPTI 2600
    IF(GAN(2,K2,K1).EQ.0.0) GO TO 95OPTI 2610
    CALL FRTCMS('CLRSCRN ')OPTI 2620
    IF(IITFX.EQ.1) GO TO 41OPTI 2630
    WRITE(6,120)OPTI 2640
    IF(IITFX.EQ.1) WRITE(6,710)OPTI 2650
    IF(IITFX.EQ.2) WRITE(6,711)OPTI 2660
    IF(IITFX.EQ.3) WRITE(6,712)OPTI 2670
    CALL GSETUP(IPT,PAGE,PAGEY)OPTI 2680
    IF(IITFX.EQ.1) GO TO 90OPTI 2690
    IF(IITFX.EQ.2) GO TO 90OPTI 2700
    CALL FRTCMS('CLRSCRN ')OPTI 2710
    IF(IITFX.EQ.1) GO TO 90OPTI 2720
    CALL FRTCMS('CLRSCRN ')OPTI 2730
    IF(IITFX.EQ.1) GO TO 90OPTI 2740
    IF(IITFX.EQ.2) GO TO 90OPTI 2750
    IF(IITFX.EQ.3) GO TO 90OPTI 2760
    CONTINUEOPTI 2770
C-----SCALING CONSTANTS-----OPTI 2780
C-----INPUT ARRAYS-----OPTI 2790
    CALL PSCALE(PAGEX)OPTI 2800
    GOL = SNGL(GAN(2,K2,K1))OPTI 2810
    CALL FREQ(NS,K2,K1,FMIN,FMAX,N)OPTI 2820
    DO 5 I = 1,NOPTI 2830
        XNO(I) = SNGL(QR(I))OPTI 2840
        YNO(I) = SNGL(QI(I))OPTI 2850
    CONTINUEOPTI 2860
    CALL MINMAX(N,XNO,QRMIN,QRMAX)OPTI 2870
    CALL MINMAX(N,YNO,QIMIN,QIMAX)OPTI 2880
C-----PLOT LIMITS-----OPTI 2890
    XAXIS = PAGEX - 2.*T1OPTI 2900
    YAXIS = PAGEY - T1 - T2OPTI 2910
    IF(QRMIN.GT.-1.) QRMIN = -1.0OPTI 2920
    IF(QRMAX.LT.0.) QRMAX = 0.0OPTI 2930
    IF(QIMAX.LT.0.) QIMAX = 0.0OPTI 2940
    IF(QIMIN.GT.0.) QIMIN = 0.0OPTI 2950
    IF((QRMAX-QRMIN).GE.(QIMAX-QIMIN)) GO TO 6OPTI 2960
    RL = QIMAX - QIMINOPTI 2970

```

```

6 7
GO TO 7 QRMX - QRMN
RL = 2 * XAXIS
B1 = 2 * XAXIS
B2 = 2 * XAXIS
B3 = 2 * XAXIS
C2 = 2 * YAXIS
C3 = 2 * YAXIS
KK = 1
IF (RL.GT.B1) KK=2
IF (RL.GT.B2) KK=3
IF (RL.GT.B3) KK=4
GO TO (11,12,13,14),KK
RL = B1
GC = B1
RL = B2
GC = B2
RL = B3
GC = B3
II = 1
N1 = 1
IF (J.EQ.0) GO TO 21
IF (J.EQ.1) GO TO 17
GO TO 17 (FLCAT(II))/(FLCAT(N1))
RSTEP = 18
GO TO 18 RL/B3
RSTEP = RL/B3
IF ((QRMX-CRMN)/RSTEP).GE.B2) GO TO 22
XDIST = B2
GC TO 23
XDIST = -QMIN/RSTEP
IF ((QIMAX-CIMIN)/RSTEP).GE.C2) GO TO 24
YDIST = C2
GO TO 26
YDIST = -QIMIN/RSTEP
IF (XDIST.GT.B3) XDIST = B3
IF (YDIST.GT.C3) YDIST = C3
THEFAC = 3.1415927/180.
IF (IPTR.EQ.2) GO TO 10
CALL TEKDIS
CALL PAGE(PAGEX,PAGEY)
CALL HWROT(AUTO)
CALL HWSCL(SCREEN)
CALL NCBDRF
GC TO 20
C-----TEK 618-----
C-----COMPRS-----
C-----CONTINUE-----
10

```

```

20 C-----PLOT EXECUTION-----
CALL CCMPRS
CALL PAGE(PAGEX,PAGEY)
CCNTINUE
CALL NOCHECK
CALL GRACE(C)
CALL PHYSOR(T1,T1)
CALL AREA2C(XAXIS, YAXIS)
CALL TRIPLX
CALL HEIGHT(TH)
CALL XNAME(' $',100)
CALL HEIGHT(TH)
CALL YNAME(' $',100)
C-----POLAR PLOT-----
CALL POLAR(THFAC,RSTEP,XDIST,YDIST)
C-----HEADING-----
NN = NL + 1
IF(NL.EQ.0) GO TO 35
DC 35 I=1,NL
CALL CLINE(I)
CALL HEADIN(LINE,100,T3,NN)
35 CONTINUE
IF(ITFX.EQ.1)
* CALL HEADIN('OPEN LOOP TRANSFER NYQUIST$',100,T3,NN)
* IF(ITFX.EQ.2)
* CALL HEADIN('NOISE TRANSFER NYQUIST$',100,T3,NN)
* IF(ITFX.EQ.3)
* CALL HEADIN('COMPENSATOR TRANSFER NYQUIST$',100,T3,NN)
C-----AXIS PLOT-----
XDO(1) = 0.C
XDO(2) = RL*2.0
YDO(1) = 0.C
YDO(2) = 0.C
DO 27 I=1,4
CALL CURVE(YCC,XDO,2.0)
YDO(1) = YCC(I) + 90.
CCNTINUE
27 C-----UNIT CIRCLE-----
CALL DCT
CALL GRID(1,1)
XDO(1) = 0.C
YDO(1) = 1.C
DC 28 I=1,360
XDO(I) = XCC(I-1) + 1.0
YDO(I) = 1.C
CCNTINUE
CALL CURVE(XDO,YDO,360,0)
CALL RESET('DCT')
28

```

```

C-----CURVE PLCT-----OPTI 3937
DO 29 I = 1,N
XNO(I) = SNGL(QPSR(I))
YNO(I) = SNGL(MAG(I))
25 CCNT INUE
CALL CURVE(XNO,YNO,N,O)
CALL FRAME
C-----LEGEND-----OPTI 3940
XX = XAXIS - T7
YY = YAXIS + T8
CALL HEIGHT(TH1)
CALL MESSAGE(,INPUT # = $,100,XX+T8,YY+T10)
CALL HEIGHT(TH1)
CALL HEIGHT(K1,ABUT,ABUT)
CALL INTNO(K1,ABUT)
CALL HEIGHT(TH1)
CALL MESSAGE(,OUTPUT # = $,100,XX+T8,YY+T9)
CALL HEIGHT(TH1)
CALL INTNO(K2,ABUT,ABUT)
CALL HEIGHT(TH1)
CALL HEIGHT(TH1)
CALL MESSAGE(,DC GAIN = $,100,XX+T8,YY+T8)
CALL HEIGHT(TH1)
CALL HEIGHT(TH1)
CALL REALNC(GOL,-3,ABUT,ABUT)
CALL BLREC(XX,YY,T5,T6,0.02)
C-----OPTI 4000
CALL ENOPL(C)
IF(IPT,NE,2) GO TO 75
CALL FRTCMS(,CLRSCRN)
WRITE(6,13C)
C-----GAIN AND PHASE MARGIN-----OPTI 4010
DC 73 I = 1,N
OMEGA = DIMAG(WN(I))
XNO(I) = SNGL(OMEGA)
MA = 20.0*DLG10(MAG(I))
YNO(I) = SNGL(MA)
YDO(I) = SNGL(QPSR(I))
73 CCNT INUE
CALL FRTCMS(,CLRSCRN)
WRITE(6,13E)
C-----GAIN MARGIN-----OPTI 4020
DC 71 I = 2,N
IF(ABS(YDO(I-1)).LT.180.0.AND.ABS(YDO(I)).GE.180.0) GO TO 72
IF(ABS(YDO(I-1)).GT.180.0.AND.ABS(YDO(I)).LE.180.0) GO TO 72
GC TO 71
GM = 0.0 - YNO(I)
72 WRITE(6,14C) XNO(I), GM
CONTINUE
71 C-----PHASE MARGIN-----OPTI 4030
DO 76 I = 2,N
OPTI 4040

```

```

77 IF(YNO(I-1).GT.0.0.AND.YNO(I).LE.0.0) GO TC 77
GC TO 76
PM = 180.0 + YDO(I)
WRITE(6,150) XNO(I), PM
76 CCNT INUE
WRITE(6,160)
-----CHANGES TO PLOT-----
5C CCNT INUE
CALL FRTCMS('CLRSCRN ')
WRITE(6,120)
IF(IITFX.EQ.1) WRITE(6,710)
IF(IITFX.EQ.2) WRITE(6,711)
IF(IITFX.EQ.3) WRITE(6,712)
WRITE(6,100)
WRITE(6,110)
30 CALL RCINT(IANS)
IF(IANS.GT.5.OR.IANS.LT.1) GO TO 15
15 GO TO 25
WRITE(6,510)
25 CCNT INUE
ICHG = 1
GC TO (40,50,60,70,200),IANS
-----TABULAR DATA-----
202 CCNT INUE
IF(GAN(2,K2,K1).EQ.0.0) GO TO 95
ICHG = 0
CALL FRTCMS('CLRSCRN ')
251 WRITE(6,120)
GO TO 249
IF(IITFX.EQ.1) WRITE(6,710)
IF(IITFX.EQ.2) WRITE(6,711)
IF(IITFX.EQ.3) WRITE(6,712)
249 CALL PTR(IPT)
GO TC 290
IF(IICHC.EQ.1) GO TC 290
252 CALL FRTCMS('CLRSCRN ')
CALL FGRAF(FMIN,FMAX)
IF(IICHC.EQ.1) GO TC 290
253 CALL FRTCMS('CLRSCRN ')
CALL NPTS(PTS)
IF(IICHC.EQ.1) GO TC 290
-----TABULATE DATA-----
254 CCNT INUE
ZERO = 0.0
J1 = 0
CRD = DINT(GAN(1,K2,K1))
GAIN = GAN(2,K2,K1)
CALL FRTCMS('CLRSCRN ')

```

AD-A152 148

DEVELOPMENT OF GRAPHICAL POLE-ZERO ROOT-LOCUS BODE
NYQUIST AND NICHOLS RESPONSES USING THE OPTSVSX PROGRAM
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M H LAPTAS
SEP 84 F/G 9/2

3/3

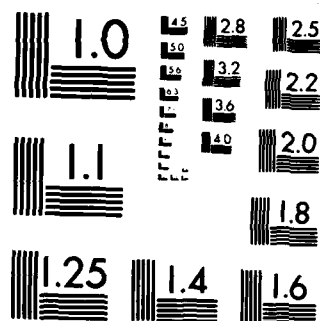
UNCLASSIFIED

NL

END

FILED

DATE



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963 A

```

WRITE(IPT,120) WRITE(IPT,710)
IF(ITFX.EQ.1) WRITE(IPT,711)
IF(ITFX.EQ.2) WRITE(IPT,712)
WRITE(IPT,301) K1
WRITE(IPT,302) K2
WRITE(IPT,311) NS
WRITE(IPT,310) CRD
WRITE(IPT,320) GAN(2,K2,K1)
WRITE(IPT,350)
N = PIS
CALL FREQ(NS,K2,K1,FMIN,FMAX,N)
DO 220 I = 1,N
OMEGA = DIMAG(WNII)
WRITE(IPT,330) OMEGA,MAG(I),QPSR(I),QR(I),CI(I)
CCNTINUE
220-----CHANGES TABULAR DATA-----
IF(IPT.EQ.2) WRITE(6,401)
IF(IPT.EQ.2) WRITE(6,402)
WRITE(6,400)
CALL RCHAR(IANS)
IF((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 245
GO TO 250
WRITE(6,500)
GO TO 240
CCNTINUE
IF(IANS.EC.IY) GO TO 290
IF(IANS.EC.IZ) GO TO 200
245-----CHANGES-----
CCNTINUE
CALL FRTCMS('CLRSCRN ')
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
WRITE(6,712)
WRITE(6,410)
WRITE(6,411)
CALL PCINT(IANS)
IF(IANS.GT.500) IANS.LT.1) GO TO 292
GO TO 293
WRITE(6,510)
GO TO 291
CCNTINUE
ICHG = I
GO TO (251,252,253,254,200),IANS
291-----
292-----
293-----
CCNTINUE
RETURN
WRITE(6,520)
295-----

```



```

C-----SCALING CONSTANTS-----OPTI 6330
C-----INPUT ARRAYS-----OPTI 6340
OPTI 6350
OPTI 6360
OPTI 6370
OPTI 6380
OPTI 6390
OPTI 6400
OPTI 6410
OPTI 6420
OPTI 6430
OPTI 6440
OPTI 6450
OPTI 6460
OPTI 6470
OPTI 6480
OPTI 6490
OPTI 6500
OPTI 6510
OPTI 6520
OPTI 6530
OPTI 6540
OPTI 6550
OPTI 6560
OPTI 6570
OPTI 6580
OPTI 6590
OPTI 6600
OPTI 6610
OPTI 6620
OPTI 6630
OPTI 6640
OPTI 6650
OPTI 6660
OPTI 6670
OPTI 6680
OPTI 6690
OPTI 6700
OPTI 6710
OPTI 6720
OPTI 6730
OPTI 6740
OPTI 6750
OPTI 6760
OPTI 6770
OPTI 6780
OPTI 6790
OPTI 6800

C-----PLOT LIMITS-----
C-----TEK 618-----
C-----COMPRS-----
C-----PLOT EXECUTION-----
C-----HEADING-----

CALL PSCALE(PAGEX)
CALL FREQ(NS,K2,K1,FMIN,FMAX,N)
DO 5 I=1,N
  WA = 20.0 * DLOG10(MAG(I))
  XDO(I) = SNGL(WA)
  YDO(I) = SNGL(QPSR(I))
  CCNT INUE
  CALL MINMAX(N,YDO,MMIN,MMAX)
  CALL MINMAX(N,XDO,PMIN,PMAX)
  XAXIS = PAGEX - 2 * T1
  YAXIS = PAGEY - T1 - T2
  IF(MMAX.LT.C.O) MMAX = 0.0
  IF(MMIN.GT.C.O) MMIN = 0.0
  IF(PMAX.LT.C.O) PMAX = 0.0
  IF(PMIN.GT.C.O) PMIN = 0.0
  IF(IPTR.EQ.2) GO TO 10
  CALL TEK618
  CALL PAGE(PAGEX,PAGEY)
  CALL HPROT('AUTO')
  CALL HWSCL('SCREEN')
  CALL NCBRDR
  GO TO 20
  CCNT INUE
  CALL CCMPRS
  CALL PAGE(PAGEX,PAGEY)
  CCNT INUE
  CALL NCCHCK
  CALL GRACE(C)
  CALL PHYSOR(T1,T2)
  CALL AREA2C(XAXIS, YAXIS)
  CALL TRIPLX
  CALL HEIGHT(TH)
  CALL XNAME('PHASE - DEGREES $',100)
  CALL HEIGHT(TH)
  CALL YNAME('MAGNITUDE - DECIBELS $',100)
  CALL GRAF(FMIN,'SCALE',PMAX,MMIN,'SCALE',MMAX)
  NN = NL + 1
  IF(NL.EQ.0) GO TO 35
  DC 35 I=1,NL
  CALL CLINE(I)

```

```

35 CALL HEADIN(LINE,100,T3,NN)
CONTINUE EQ.1)
IF(ITFX.EQ.1)
* CALL HEADIN('OPEN LOOP TRANSFER NICHOLS$',100,T3,NN)
* IF(ITFX.EQ.2)
* CALL HEADIN('NOISE TRANSFER NICHOLS$',100,T3,NN)
* IF(ITFX.EQ.3)
* CALL HEADIN('COMPENSATOR TRANSFER NICHOLS$',100,T3,NN)
C-----CURVE PLOT-----
CALL CURVE(XCO,YDO,N,C)
C-----LEGEND-----
XX = XAXIS - T7
YY = YAXIS + T8
CALL HEIGHT(TH1)
CALL MESSAGE('INPUT # = $',100,XX+T8,YY+T10)
CALL HEIGHT(TH1)
CALL INTNO(K1,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAGE('OUTPUT # = $',100,XX+T8,YY+T9)
CALL HEIGHT(TH1)
CALL INTNO(K2,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAGE('DC GAIN = $',100,XX+T8,YY+T8)
CALL HEIGHT(TH1)
CALL REALNC(GOL,-3,'ABUT','ABUT')
CALL BLREC(XX,YY,T3,T6,0.02)
C-----GRID-----
CALL DOT
CALL GRID(1,1)
CALL RESET('DOT')
CALL ENDPL(0)
IF(IPTR.NE.2) GO TO 75
CALL FRTCMS('CLRSCRN ')
WRITE(6,130)
C-----GAIN AND PHASE MARGIN-----
75 DO 73 I = 1,N
CMEGA = DIMAG(WN(I))
XNO(I) = SNGL(OMEGA)
PA = 20.0 * CLOG10(MAG(I))
YNO(I) = SNGL(PA)
YDO(I) = SNGL(QPSR(I))
CONTINUE
CALL FRTCMS('CLRSCRN ')
WRITE(6,135)
73 DO 71 I = 2,N
IF(ABS(YDO(I-1)).LT.180.0.AND.ABS(YDO(I)).GE.180.0) GO TO 72
C-----GAIN MARGIN-----

```

```

72 IF(ABS(YDO(I-1)).GT.180.0.AND.ABS(YDO(I)).LE.180.0) GO TO 72
   GM = 0.0 - YNO(I)
71 WRITE (6,14C) XNO(I), GM
   CONTINUE
   -----PHASE MARGIN-----
DO 76 I = 1,N
77 IF(YNO(I-1).GT.0.0.AND.YNO(I).LE.0.0) GO TO 77
   GO TO 76
   PM = 180.0 + YDO(I)
76 WRITE (6,150) XNO(I), PM
   CONTINUE
   -----CHANGES TO PLOT-----
50 CCNT INUE
   CALL FRTCMS('CLRSCRN ')
   WRITE(6,12C)
   IF(IITFX.EQ.1) WRITE(6,710)
   IF(IITFX.EQ.2) WRITE(6,711)
   IF(IITFX.EQ.3) WRITE(6,712)
   WRITE(6,100)
   WRITE(6,11C)
30 CALL RDINT(IANS)
   IF(IANS.GT.5.OR.IANS.LT.1) GO TO 15
15 GO TO 25
25 WRITE(6,51C)
   GO TO 30
   CONTINUE
   ICPG = 1
   GO TO (40,50,60,70,200),IANS
   -----TABULAR DATA-----
202 CCNT INUE
   IF(GAN(2,K2,K1).EQ.0.0) GO TO 95
   ICHG = 0
251 CALL FRTCMS('CLRSCRN ')
   WRITE(6,120)
   IF(IITFX.EQ.1) WRITE(6,710)
   IF(IITFX.EQ.2) WRITE(6,711)
   IF(IITFX.EQ.3) WRITE(6,712)
   CALL PRTF(FTS)
245 CALL CHG(EQ.1) GO TO 290
   IF(IITFX.EQ.1) GO TO 290
252 CALL FRTCMS('CLRSCRN ')
   CALL FGRAF(FMIN,FMAX)
   IF(IITFX.EQ.1) GO TO 290
253 CALL FRTCMS('CLRSCRN ')
   CALL NPTS(FTS)
   IF(ICHG.EQ.1) GO TO 290

```

```

C-----TABULATE DATA-----
254  CCNT INUE
      ZERO = 0.0
      J1 = 0
      ORD = CINT (GAN(1,K2,K1))
      GAIN = GAN(2,K2,K1)
      CALL FRTCMS('CLRSCRN ')
      WRITE(IPT,120) WRITE(IPT,710)
      IF(IIFX.EQ.1) WRITE(IPT,711)
      IF(IIFX.EQ.2) WRITE(IPT,712)
      IF(IIFX.EQ.3) K1
      WRITE(IPT,300) K2
      WRITE(IPT,301) K3
      WRITE(IPT,311) NS
      WRITE(IPT,310) ORD
      WRITE(IPT,320) GAN(2,K2,K1)
      WRITE(IPT,330)
      N = PTS
      CALL FREQ(NS,K2,K1,FMIN,FMAX,N)
      DO 220 I = 1,N
      CMGA = DIMAG(WN(I))
      MA = 20.0 * DLOG10(MAG(I))
      WRITE(IPT,330) CMGA,MA,QPSR(I),QR(I),QI(I)
      CONTINUE
      -----CHANGES TABULAR DATA-----
220  C-----
      IF(IPT.EQ.3) WRITE (6,401)
      IF(IPT.EQ.2) WRITE (6,402)
      WRITE(6,400)
      CALL RDCMAP (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 245
      GO TO 250
      WRITE (6,500)
      GO TO 240
240  CCNT INUE
      IF (IANS.EC.IY) GO TC 290
      IF (IANS.EC.IZ) GO TC 200
      -----CHANGES-----
250  C-----
      CCNT INUE
      CALL FRTCMS('CLRSCRN ')
      WRITE(6,120)
      IF(IIFX.EQ.1) WRITE(6,710)
      IF(IIFX.EQ.2) WRITE(6,711)
      IF(IIFX.EQ.3) WRITE(6,712)
      WRITE(6,410)
      WRITE(6,110)
      CALL RDCINT(IANS)
      IF(IANS.GT.5.OR.IANS.LT.1) GO TO 292
      GO TO 293
251  C-----
      WRITE(6,110)
      WRITE(6,120)
      WRITE(6,130)
      WRITE(6,140)
      WRITE(6,150)
      WRITE(6,160)
      WRITE(6,170)
      WRITE(6,180)
      WRITE(6,190)
      WRITE(6,200)
      WRITE(6,210)
      WRITE(6,220)
      WRITE(6,230)
      WRITE(6,240)

```

199


```

DELTF = 10.**CYCLE
DO 80 I = 1, NN
IF (I.GT.1) GO TO 5
WN(I) = DCMPLX(ZERO, FRMIN)
GO TO 6
WN(I) = WN(I-1) * DELTF
-----POLYNOMIAL-----
CPOL = 1.0
IF (J1.EQ.0) GO TO 7
DO 30 K = 1, J1
CPOL = CPOL*(WN(I)-RTN(K))/(WN(I)-RTD(K))
IF (J1.EQ.N) GO TO 8
JJ = J1 + 1
DO 40 K = JJ, NS
CPOL = CPOL*(WN(I)-RTD(K))
CPOL = CPOL*GAIN
-----MAGNITUDE-----
MAG(I) = CCABS(CPOL)
-----PHASE-----
QT(I) = CMAG(CPOL)
IF (I) = CREAL(CPOL)
IF (QI(I)) 601, 602, 603
IF (QR(I)) 620, 621, 622
IF (QI(I)) 623, 751, 625
IF (QR(I)) 626, 627, 628
QPSR(I) = -180. + 57.29577951*DATAN(QI(I)/QR(I))
GO TO 555
QPSR(I) = -50.
GO TO 555
QPSR(I) = - 57.29577951*DATAN(-QI(I)/QR(I))
GO TO 555
QPSR(I) = 180.
GO TO 555
QPSR(I) = 0.0
GO TO 555
QPSR(I) = 180. - 57.29577951*DATAN(-QI(I)/QR(I))
GO TO 555
QPSR(I) = 50.
GO TO 555
QPSR(I) = 57.29577951*DATAN(QI(I)/QR(I))
GO TO 555
IF (IPHS.EQ.1) GO TO 640
IF ((QPSR(I-1).GT.90.) .AND. (QPSR(I).LT.-90.)) GO TO 630
IF ((QPSR(I-1).LT.-90.) .AND. (QPSR(I).GT.90.)) IPHS = 1
GC TO 640
N = I - 1
DO 650 L = 1, N
QPSR(L) = QPSR(L) - 360.

```



```

65 GC TO 70
   WRITE(6,510)
70 GO TO 60
   CCNTINUE
   PAGEX = ANSF
   RETURN
C-----VRSTEC PAGE SIZE-----
8C CCNTINUE
   PAGEX = 11.0
   PAGEY = 8.5
   RETURN
C-----
100 FORMAT(/,5X,'PLOTTER SELECTION AND PAGE SIZE',/,5X,
*,'(NOTE: PAGE IS 11.0 INCHES BY 8.5 INCHES WITH SELECTION OF TEK618
*),/,,)
110 FORMAT(/,10X,'WHICH PLOTTER DO YOU DESIRE:',/,15X,'1. TEK618',/,
*15X,'2. VERSATEC',/,10X,'CHOCSE OPTION 1 OR 2',)
130 FCFORMAT(/,10X,'PAGE SIZE (MAXIMUM = 21.0 INCHES BY 21.0 INCHES)',
*,,)
140 FFORMAT(/,15X,'HEIGHT = ',/,,)
150 FFORMAT(/,15X,'WIDTH = ',/,,)
500 FFORMAT(/,5X,'***** ERROR *****',/,,)
510 FFORMAT(/,5X,'***** ERROR *****',/,,)
   END
C=====
C SUBROUTINE GGRAF(XMIN,XMAX,YMIN,YMAX)
C SUBROUTINE GGRAF INTERACTIVELY ALLOWS THE USER TO SELECT PLOTTING
C LIMITS FOR USE WITH DISSPLA GRAPHICS
C=====
   REAL XMIN,XMAX,YMIN,YMAX,ANSR
   DATA IY,'Y',IZ,'N',/
C-----
4C CCNTINUE
   WRITE(6,100)
   WRITE(6,105)
C-----
   WRITE(6,110)
   CALL RCREAL(ANSR)
   XMIN = ANSR
C-----
   WRITE(6,120)
   CALL RCREAL(ANSR)
   XMAX = ANSR
C-----
   WRITE(6,125)
   CALL RCREAL(ANSR)
   YMIN = ANSR
C-----
   WRITE(6,130)

```



```

C-----DATA IY/Y',IZ/N'-----
4C CCNTINUE
   WRITE(6,10C)
   WRITE(6,10E)
C-----FMIN-----
5C WRITE(6,11C)
   CALL RCREAL(ANSR)
   FMIN = ANSR
   IF (FMIN.GT.C.0) GO TC 60
   WRITE(6,110)
   GO TO 5C
C-----FMAX-----
6C WRITE(6,120)
   CALL RCREAL(ANSR)
   FMAX = ANSR
   IF (FMAX.GT.C.0) GO TC 70
   WRITE(6,120)
   GO TO 60
C-----CORRECTIONS-----
7C CALL FRTCMS('CLRSRN ')
   WRITE(6,10C)
   WRITE(6,10E)
   WRITE(6,150) FMIN
   WRITE(6,160) FMAX
   WRITE(6,19C)
   CALL RCHAR(ANS)
   IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 10
   GO TO 20
1C WRITE(5,50C)
   GO TO 30
20 CCNTINUE
   IF (IANS.EQ.IY) GO TC 40
   RETURN
C-----PLOTTING LIMITS FOR GRAPH'-----
100 FORMAT(/,5X,'PLOTTING LIMITS FOR GRAPH',/)
110 FORMAT(10X,'FREQUENCY RANGE = ',/,/)
120 FORMAT(15X,'MINIMUM FREQUENCY = ',/,/)
130 FORMAT(15X,'MAXIMUM FREQUENCY = ',/,/)
140 FORMAT(15X,'MINIMUM FREQUENCY = ',/,/)
150 FORMAT(15X,'MAXIMUM FREQUENCY = ',/,/)
160 FORMAT(15X,'DO YOU DESIRE TO MAKE ANY CHANGES?',/,10X,
*TYPE 'YES',CP 'NO'.//)
170 FORMAT(15X,'WARNING',/,5X,'IMPROPER DATA ENTRY',
*ENTER 'YES' OR 'NO'.//)
180 FORMAT(15X,'WARNING',/,5X,'IMPROPER DATA ENTRY',
*FREQUENCY MUST BE GREATER THAN ZERO.//)
190
500
510
END

```

```

C=====OPT21610
C=====OPT21620
C=====OPT21630
C=====OPT21640
C=====OPT21650
C=====OPT21660
C=====OPT21670
C=====OPT21680
C=====OPT21690
C=====OPT21700
C=====OPT21710
C=====OPT21720
C=====OPT21730
C=====OPT21740
C=====OPT21750
C=====OPT21760
C=====OPT21770
C=====OPT21780
C=====OPT21790
C=====OPT21800
C=====OPT21810
C=====OPT21820
C=====OPT21830
C=====OPT21840
C=====OPT21850
C=====OPT21860
C=====OPT21870
C=====OPT21880
C=====OPT21890
C=====OPT21900
C=====OPT21910
C=====OPT21920
C=====OPT21930
C=====OPT21940
C=====OPT21950
C=====OPT21960
C=====OPT21970
C=====OPT21980
C=====OPT21990
C=====OPT22000
C=====OPT22010
C=====OPT22020
C=====OPT22030
C=====OPT22040
C=====OPT22050
C=====OPT22060
C=====OPT22070
C=====OPT22080

SUBROUTINE RANGK(KMIN,KMAX)
SUBROUTINE RANGK INTERACTIVELY ALLOWS THE USER TO SELECT GAIN
LIMITS FOR USE WITH ROOT-LOCUS PLOTS
REAL*8 KMIN,KMAX
DATA IY,Y,IZ,N/
C-----
4C CCNTINUE
   WRITE(6,100)
C-----
5C   WRITE(6,110)
   CALL RCREAL(ANSR)
   KMIN = ANSR
   KMAX = ANSR
C-----
6C   WRITE(6,120)
   CALL RCREAL(ANSR)
   KMAX = ANSR
C-----
7C   CALL FRTCMS('CLRSCRN ')
   WRITE(6,100)
   WRITE(6,150) KMIN
   WRITE(6,160) KMAX
   WRITE(6,190)
   CALL RDCHEAR(IANS)
   IF(IANS.NE.IY).AND.(IANS.NE.IZ) GO TO 10
   GO TO 20
1C   WRITE(6,500)
   GO TO 30
2C   CCNTINUE
   IF(IANS.EQ.IY) GO TO 40
   RETURN
C-----
100  FORMAT(/,5X,'GAIN RANGE FOR ROOT-LOCUS PLCT',/)
110  FORMAT(15X,'MINIMUM GAIN = ',/)
120  FORMAT(15X,'MAXIMUM GAIN = ',/)
130  FORMAT(15X,'MINIMUM GAIN = ',E15.3,/)
140  FORMAT(15X,'MAXIMUM GAIN = ',E15.3,/)
150  FORMAT(/,5X,'DO YOU DESIRE TO MAKE ANY CHANGES ?',/,10X,
      *,'TYPE "YES" OR "NO".',/)
500  FORMAT(/,5X,'*****WARNING *****',/,5X,'IMPROPER DATA ENTRY ',/,
      *3X,'ENTER "YES" OR "NO".',/)
      END
C-----
C     SUBROUTINE GHEAD(NL)
C     SUBROUTINE GHEAD INTERACTIVELY ALLOWS THE USER TO INPUT THREE

```



```

C-----STOP
5C  FORMAT(IX,'WARNING: NULL STRINGS ARE NOT ALLOWED, ENTER',
6C  *IX,'CHARACTER VALUES.')
```

```

7C  FORMAT(///,2X,42HPROGRAM KILLED - TWO NULL STRINGS ENTERED ,/)
      FORMAT(9A4)
      END
C=====
C  SUBROUTINE CLINE(ILN)
C  SUBROUTINE PLACES A DCLLAR SIGN AT THE END OF A CHARACTER STRING
C  ENTERED FROM THE KEYBOARD AND PLACES THE RESULTING STRING INTO
C  A 1-D ARRAY TC BE USED BY DISPLA
C=====
C  INTEGER I,ILN,LINES,J,LINE,DOLLAR,BLANK
C  COMMON /SHEAD/ LINES(4,18),LINE(9)
C-----
C  DATA DOLLAR/'$ ',BLANK/' '
C  DO 10 I=1,5
C  J=1
C  LINE(I)=LINES(ILN,I)
C  IF(LINE(I).EQ.BLANK) GO TO 20
1C  CONTINUE
2C  LINE(J)=DCLLAR
      RETURN
      END
C=====
C  SUBROUTINE RDREAL -- INTERACTIVELY READS A REAL NUMBER REPLY
C  INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS A NULL
C  STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
C  SUBROUTINE RDREAL (ANSR)
C  REAL*8 ANSR
C  INTEGER COUNT
```

```

C-----
1C  COUNT=0
      CONTINUE+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (6,6C)
      GO TO 40
      CONTINUE=
      READ (5,*,END=30,ERR=30) ANSR
      RETURN
2C  REWIND 5
      WRITE (6,5C)
      GO TO 10

```

```

4C      CCNTINUE
      STOP
5C      FORMAT (1X,64HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUMERICAL VALUE.)
6C      FORMAT (///,2X,42HPROGRAM KILLED - TWO NULL STRINGS ENTERED ,/)
      END
C=====
C      SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY
C      INTCA FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS A NULL
C      STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
      SUBROUTINE RDINT (IANS)
      INTEGER COUNT,IANS
C=====
      COUNT=0
      CCNTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (6,6C)
      GO TO 40
      CCNTINUE
      READ (5,*,END=30,ERR=30) IANS
      RETURN
      REWIND 5
      WRITE (6,5C)
      GO TO 10
      CCNTINUE
      STOP
C=====
5C      FORMAT (1X,64HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUMERICAL VALUE.)
6C      FORMAT (///,2X,42HPROGRAM KILLED - TWO NULL STRINGS ENTERED ,/)
      END
C=====
C      SUBROUTINE RDCCHAR -- 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 RDCCHAR (IANS)
      INTEGER COUNT,IANS
C=====
      COUNT=0
      CCNTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20

```

```

2C      WRITE (6,6C)
        GO TO 40
        CCNTINUE
3C      READ (5,70,END=30,ERR=30) IANS
        RETURN 5
        WRITE (6,5C)
        GC TO 10
4C      CCNTINUE
        STOP
C-----
5C      FORMAT (1X,60HWARNING:  NULL STRINGS ARE NOT ALLCWD, ENTER "YES"
        1CR "NO")
6C      FORMAT (//,2X,42HPROGRAM KILLED - TWO NULL STRINGS ENTERED ,/)
7C      FORMAT (A1)
        END
C=====
C      SUBROUTINE MINMAX(N,AA,MIN,MAX)
C      SUBROUTINE MINMAX FINDS THE MINIMUM AND MAXIMUM VALUES FOR A ONE
C      DIMENSIONAL ARRAY
C=====
C      REAL*4 MIN,MAX,AA
C      INTEGER N
C      DIMENSION AA(N)
C-----
        MAX = AA(1)
        MIN = AA(1)
        DO 10 I = 1,N
            IF (AA(I) .GT. MAX) GC TO 5
            IF (AA(I) .LT. MIN) GC TO 6
        GO TO 10
        MAX = AA(1)
        MIN = AA(1)
        CCNTINUE
        RETURN
        END
C=====
C      SUBROUTINE MAKPOL(N,R,C)
C      SUBROUTINE MAKPOL COMPUTES THE COMPLEX COEFFICIENTS OF AN N-TH
C      DEGREE POLYNOMIAL GIVEN N COMPLEX ROOTS OF THE POLYNOMIAL
C=====
C      COMPLEX*16 R(N),C(N)
C      IF (N .LE. 0) RETURN
C      DO 10 I=1,N

```



```

15 WRITE(6,510)
25 GC TO 30
CCNT INUE
PTS = IANS
C-----
C-----
100 FORMAT(/,5X,'HOW MANY POINTS DO YOU WANT TO TABULATE?',/,
*10X,'(50 IS THE MAXIMUM)',/)
110 FORMAT(/,5X,'ENTER NUMBER OF POINTS',/)
510 FORMAT(/,5X,'***** ERROR ***** MUST ENTER BETWEEN 1 AND 50
*0 POINTS',/)
C-----
C-----
END
C=====
C=====
SUBROUTINE FSCALE(PAGEX)
SUBROUTINE PSCALE AUTOMATICALLY SCALES DISSPLA FLCTS
C=====
C=====
REAL*4 PAGEX,TS,TH,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10
COMMON /SCALE/ TS,TH,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10
C-----
C-----
TS = PAGEX/11.0
TH = 0.14*TS
T1 = 0.10*TS
T2 = 0.60*TS
T3 = 1.60*TS
T4 = 1.20
T5 = 0.30*TS
T6 = 2.10*TS
T7 = 0.9*TS
T8 = 2.00*TS
T9 = 0.20*TS
T10 = 3.0*TS
IF(PAGEX.LT.6.0) T3 =1.0
RETURN
C-----
C-----
END

```

APPENDIX D
OPGRAPH LISTING

This portion of the thesis contains a sample of tabular
output sent to a disk (OPGRAPH LISTING)

BCCE PLOT COMPENSATOR TRANSFER FUNCTION

```

INPUT NUMBER = 1
OUTPUT NUMBER = 1
DENOMINATOR ORDER = 2
NUMERATOR ORDER = 1
TRANSFER FUNCTION (DC) GAIN = -0.1135D+06

```

FREQUENCY (RAD/SEC)	MAGNITUDE (DECIBELS)	PHASE (DEGREES)	REAL PART	IMAGINARY PART
0.1000D+02	0.4815D+02	-0.1482D+03	-0.2171D+03	-0.1346D+03
0.1778D+02	0.5106D+02	-0.1401D+03	-0.2740D+03	-0.2292D+03
0.3162D+02	0.5504D+02	-0.1423D+03	-0.4466D+03	-0.3456D+03
0.5623D+02	0.5891D+02	-0.1629D+03	-0.8431D+03	-0.2598D+03
1.000D+02	0.5956D+02	-0.2025D+03	-0.8786D+03	-0.3637D+03

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. 1977.
2. Walker, R. A., User's Manual for OPTSYS 4 at SCIP, Stanford Univ., Aero/Astro Dept., Dec. 1979.
3. Liu, G., User's Manual for OPTSYS 5 at CIT, Stanford Univ., Aero/Astro Dept., Aug. 1982.
4. Hoden, J. G., Interactive Implementation of the Optimal Systems Control Design Program (OPTSYSX) on the IBM/3033, MS Thesis, Naval Postgraduate School, Monterey CA.
5. Diel, H. A., Development of Graphical Time Response using the OPTSYSX Program, MS Thesis, Naval Postgraduate School, Monterey CA.
6. Bryson, A. E. and Ho, Y. C., Applied Optimal Control, Hemisphere Pub. Co., 1969, (2nd Printing, 1975).
7. Research and Educational Association, Problem Solver in Automatic Control Systems/Robotics, 1982.
8. Kwakernaak, H. and Sivan, R., Linear Optimal Control Systems, Wiley-Interscience, 1972.

BIBLIOGRAPHY

D'Azzo, J.J. and Houpis, C.H., Linear Control System Analysis and Design: Conventional and Modern, McGraw-Hill, 1981.

DiStefano III, J.J., Stubberud, A.R., and Williams, I.J., Feedback and Control Systems, Schaum's Outline Series, McGraw-Hill, 1967.

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. CDR V.C. Gordon 315 Alameda Blvd. Coronado, Ca 92118		1
6. Professor A.E. Bryson Department of Aeronautics and Astronautics Stanford University Stanford, Ca 94305		1
7. LT M.H. Laptas 39 Valley Heights Holyoke, Ma 01040		5

END

FILMED

5-85

DTIC