

A021 838

REPORT

NCSL 242-75

OCTOBER 1975

THE ANALYSIS OF A LONGITUDINAL CONTROL SYSTEM FOR UNDERWATER VEHICLES

D. E. HUMPHREYS

R. W. MILLER

L. F. DEWBERRY

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED



NCSL

PANAMA CITY, FLORIDA 32401



NAVAL COASTAL SYSTEMS LABORATORY

PANAMA CITY, FLORIDA 32401

ADMINISTRATIVE INFORMATION

This effort was accomplished jointly under Task Area Number ZF 01-112-001, Work Unit Number 00112-50-1A, Vehicle Hydrodynamics and Task Area Number SF 34-371-208, Work Unit Number 16436, Analytical Investigation of Countermeasure Launchers.

APPROVED OCTOBER 1975

W. T. ODUM, Head
Diving and Salvage Department

CAPT ROBERT T. QUINN, USN
Commanding Officer

GERALD G. GOULD
Technical Director

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NCSL 242-75	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE ANALYSIS OF LONGITUDINAL CONTROL SYSTEM FOR UNDERWATER VEHICLES		5. TYPE OF REPORT & PERIOD COVERED Informal
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Douglas E. Humphreys Richard W. Miller Larry F. Dewberry		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Coastal Systems Laboratory Panama City, Florida 32401		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Tasks SF 34-371-208 and ZF 61-112-001 Work Units 00112-50-1A&16436
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE October 1975
		13. NUMBER OF PAGES 72
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
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)		
Underwater vehicles Automatic control Models Longitudinal stability Towed vehicles Feedback control Hydrodynamic coefficients Computer programs Hydromechanics Self-propelled vehicles Frequency		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
A general longitudinal feedback control system containing pitch, pitch rate, depth, and depth rate feedback is described. The Laplace domain transfer functions for each feedback loop are developed. A computer analysis program utilizing the root locus technique is developed for aiding in the design of the control system. An illustrative example design problem is included.		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

19.

Identifiers

- Multiloop control system
- Longitudinal feedback system
- Vehicle longitudinal feedback

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

	<u>Page No.</u>
INTRODUCTION.	1
LONGITUDINAL CONTROL SYSTEM ANALYSIS.	2
Background	2
First Loop	2
Second Loop.	5
Third Loop	6
Fourth Loop.	7
COMPUTER PROGRAM USERS GUIDE.	9
Basic Program Description.	9
Input Requirements	10
Example Problem.	10
First Loop Analysis.	14
Second, Third, and Fourth Loops.	16
CONCLUSIONS	17
APPENDIX A - SOME NOTES ON THE CONSTRUCTION AND INTERPRETATION OF ROOT LOCUS	A-1
APPENDIX B - EXPRESSIONS FOR THE LONGITUDINAL TRANSFER FUNCTION COEFFICIENTS.	B-1
APPENDIX C - LONGITUDINAL CONTROL SYSTEM ANALYSIS PROGRAM - CARD LIST.	C-1
APPENDIX D - TIME HISTORY PLOT DIAGRAM CARD LIST.	D-1

LIST OF ILLUSTRATIONS

<u>Figure No.</u>		<u>Page No.</u>
1	Block Diagram for Longitudinal Control System	3
2	Block Diagram for the First Loop	4
3	Block Diagram for the Second Loop	5
4	Block Diagram for the Third Loop	8
5	Block Diagram for the Fourth Loop	8
6	Example Data File	11
7	Example Problem	12
8	Example Problem: First Loop Analysis	14
9	Root Locus for the First Loop	15
10	Example Problem: Second, Third and Fourth Loop Analysis	18
11	Root Locus for the Second Loop	22
12	Root Locus for the Third Loop	22
13	Root Locus for the Fourth Loop	22

INTRODUCTION

The requirement for feedback control systems in underwater vehicles is well established. In its absolute form, feedback systems are used to control vehicle depth and yaw, while rate feedback (pitch or yaw rate) can be used to improve a vehicle's handling characteristics. The purpose of this report is to present a step-by-step procedure for the analysis of vehicle longitudinal feedback control systems.

The control system discussed is general in that it allows the designer to select any or all of four feedback loops (pitch rate, pitch, depth rate, and depth). Each loop is analyzed separately for this purpose. The root-locus technique is used in the analysis. To aid the engineer in the design process, a computer program has been written that will perform all the necessary computations. This program is suitable for analyzing both self-propelled and towed vehicles. Inputs to the program consists of vehicle length, speed, mass, moments of inertia, and the 30 linear hydrodynamic coefficients. Vehicle mass and moments of inertia are computed using the MIDCOHV computer program WTBAL reported in NCSL Report 220-74⁽¹⁾. The hydrodynamic coefficients are computed in the MIDCOHV computer program GEORGE. The details necessary for running the program are presented in the users guide section. An example design is included to illustrate the analysis of a longitudinal feedback control system. The analysis of lateral feedback control systems is discussed in an NCSL report⁽²⁾.

⁽¹⁾ Naval Coastal Systems Laboratory Report 220-74, *The MIDCOHV Weight and Balance Computer Program (WTBAL)*, by K. W. Watkinson, September 1974, Unclassified.

⁽²⁾ Naval Coastal Systems Laboratory Report, *The Analysis of Lateral Control Systems for Self-Propelled and Towed Submerged Vehicles*, by Douglas E. Humphreys, Richard W. Miller, and Larry F. Dewberry, (in publication), Unclassified

LONGITUDINAL CONTROL SYSTEM ANALYSIS

BACKGROUND

A general model of a longitudinal feedback system is shown in Figure 1. The feedback loops are pitch rate ($\dot{\theta}$), pitch angle (θ), depth rate ($\dot{Z}$), and depth (Z). The sensors are modeled as pure gains and are denoted as $K_{\dot{\theta}}$, K_{θ} , $K_{\dot{Z}}$, and K_Z . There are two command inputs: desired depth (Z_0) and desired pitch angle (θ_0).

The purpose of a feedback control system is to either stabilize an unstable system, improve the system response characteristics, to control a certain variable, such as depth, or a combination of these. The desired vehicle control is achieved by successively closing each loop and varying the loop gain until the desired system dynamics are achieved. The root-locus method is used here to aid in the analysis process. For additional details on the root-locus method and the mathematics of Laplace transforms, See References 3, 4, 5, and 6 and Appendix A.

FIRST LOOP

The inner-most loop (or first loop) is shown in Figure 2. The vehicle transfer function relating pitch rate response to stern plane input is

$$\frac{\dot{\theta}}{\delta_s} = \frac{s\theta}{\delta_s} = \frac{N_{\delta}^{\theta}}{D}.$$

-
- (3) Clark, R. N., *Introduction to Automatic Control Systems*, John Wiley and Sons, Inc., 1973.
- (4) Hale, F. J., *Introduction to Control System Analysis and Design*, Prentice-Hall, Inc., 1973.
- (5) Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, John Wiley and Sons, Inc., 1965.
- (6) Hildebrand, F. B., *Advanced Calculus for Applications*, Prentice-Hall, Inc., 1963.

(Text Continued on Page 4)

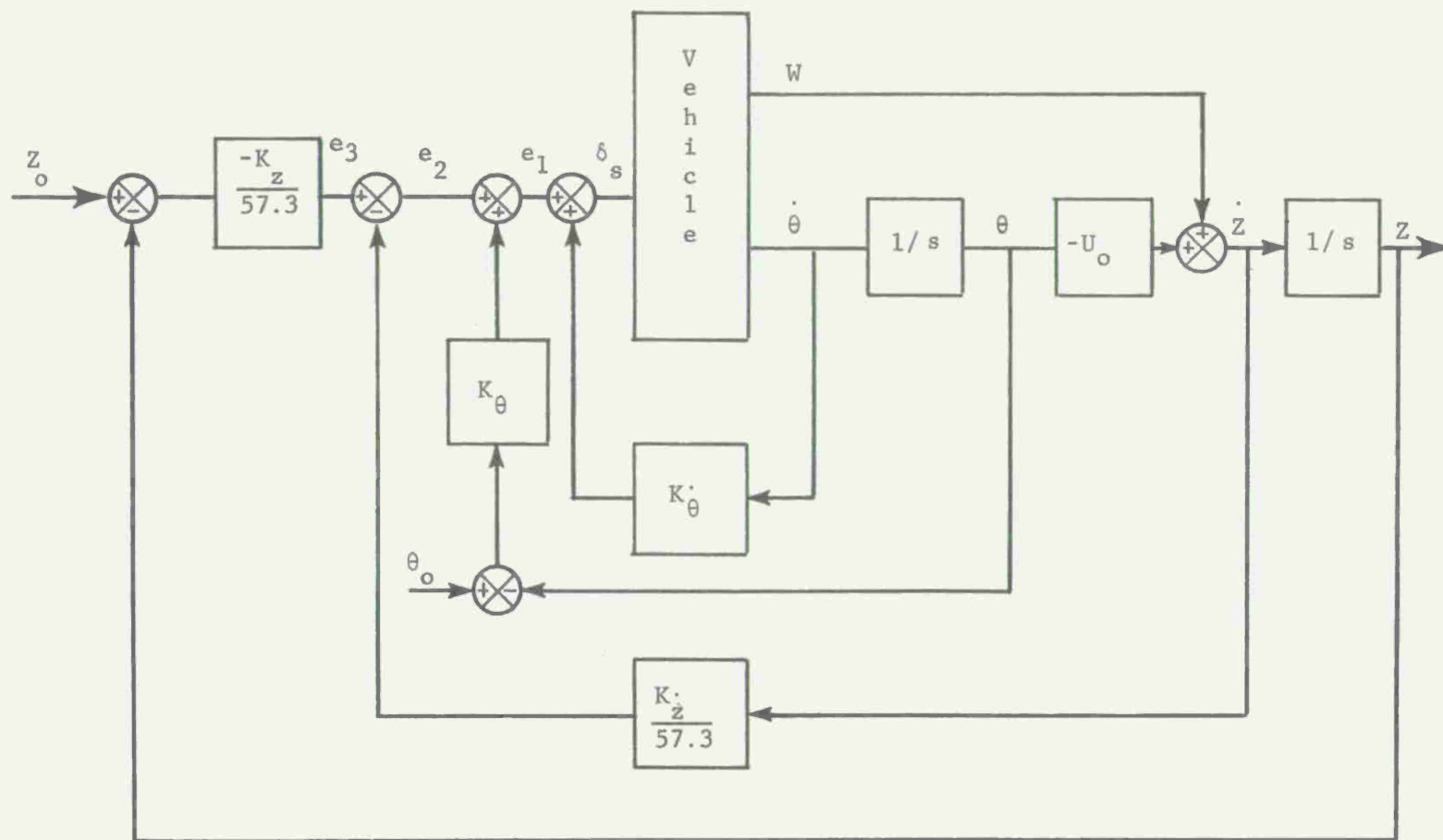


FIGURE 1. BLOCK DIAGRAM FOR LONGITUDINAL CONTROL SYSTEM

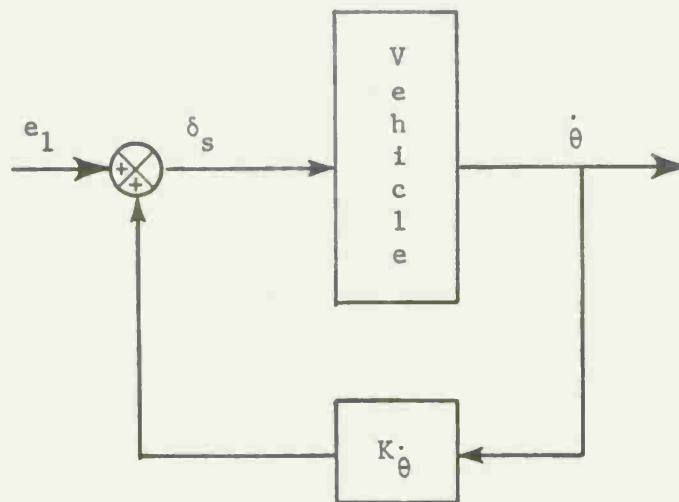


FIGURE 2. BLOCK DIAGRAM FOR THE FIRST LOOP

Where $N_{\delta_s}^{\theta}$ = Pitch angle/control deflection transfer function numerator

D = Denominator of vehicle transfer function.

The numerator and denominator are functions of the vehicle hydrodynamic coefficients. Appendix B gives the expanded form of each of the vehicle transfer functions.

By solving for the closed loop transfer function, $\dot{\theta}/e_1$, and varying the feedback gain, K_{θ} , the system dynamics can be adjusted to yield the desired performance. Solving for this closed loop transfer function yields

$$\frac{\dot{\theta}}{e_1} = \frac{\frac{s N_{\delta_s}^{\theta}}{D}}{1 - \frac{s N_{\delta_s}^{\theta}}{D} K_{\theta}} = \frac{s N_{\delta_s}^{\theta}}{D - s N_{\delta_s}^{\theta} K_{\theta}} \quad \checkmark$$

For stability: $K_{\theta} \geq 0$.

Note that although Figure 2 shows the feedback signal being added to the input signal, the system is actually a negative feedback system since the numerator, $N_{\delta_s}^{\theta}$, will always carry a negative sign.

Closing the first loop yields a new vehicle; i.e., a rate controlled vehicle, with a new characteristic equation

$$D' = D - sN_{\delta}^{\theta} K_{\theta}^{\cdot}.$$

The single prime indicates a system with one loop closure.

SECOND LOOP

Figure 3 shows the vehicle second loop after the first loop has been closed. The vehicle transfer function with one loop closed is denoted by $\dot{\theta}/e_1$. The vehicle transfer functions with two loops closed is

$$\frac{\theta}{e_2} = \frac{1/s \frac{\dot{\theta}}{e_1}}{1 + \frac{\dot{\theta}}{e_1} \frac{K_{\theta}}{s}} = \frac{N_{\delta}^{\theta} s}{D - sN_{\delta}^{\theta} K_{\theta}^{\cdot} + K_{\theta} N_{\delta}^{\theta} s}$$

For stability: $K_{\theta} \leq 0$.

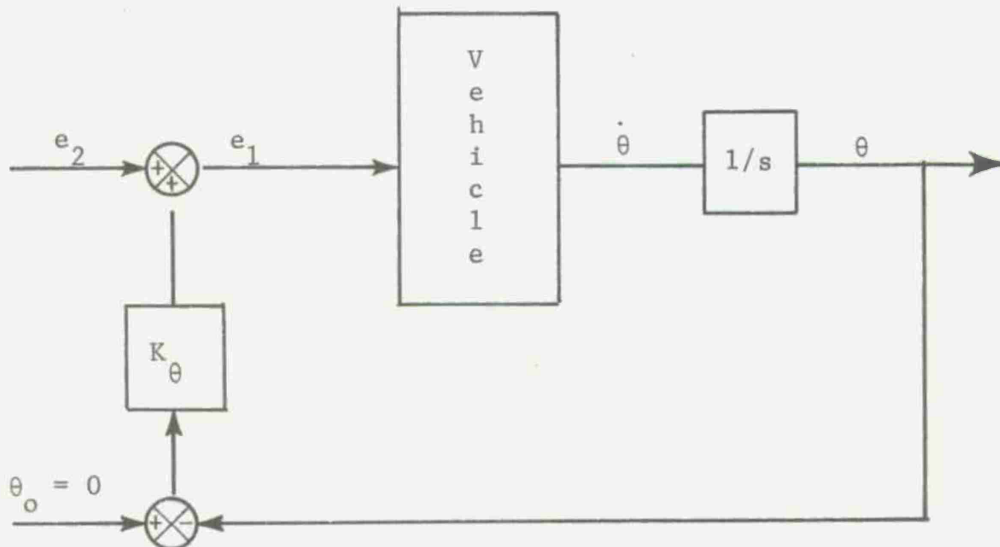


FIGURE 3. BLOCK DIAGRAM FOR THE SECOND LOOP

Note that in order to solve for the above transfer function, the pitch angle command was set to zero. Since the commanded pitch angle does not affect the vehicle characteristic response, this requirement in no way restricts the analysis capability. After the system response has been evaluated for $\theta_0 = 0$, trajectories for other pitch angle commands can be evaluated using time domain solutions such as the one shown in Reference 7.

THIRD LOOP

Figure 4 shows the block diagram of the third loop. The vehicle transfer function with two loops closed is denoted by θ/e_2 . To form the depth rate signal ($\dot{Z}$) requires the combination of the pitch signal and the vertical velocity according to the following equation

$$\dot{Z} = w - U_0 \theta ,$$

or

$$\dot{Z}/\theta = w/\theta - U_0 .$$

The w/θ transfer function is obtained by dividing the w/δ_s transfer function by the θ/δ_s transfer function which yields

$$\frac{w/\delta_s}{\theta/\delta_s} = \frac{w}{\theta} = \frac{\frac{N_{\delta_s}^w}{D}}{\frac{N_{\delta_s}^\theta}{D}} = \frac{N_{\delta_s}^w}{N_{\delta_s}^\theta} .$$

The theoretical basis for this operation can be found in an NSRDC report ⁽⁷⁾.

⁽⁷⁾Naval Ship Research and Development Center, Report No. P-4-3-H-01, *User's Guide NSRDC Digital Program for Simulating Submarine Motion ZZMN Revision 1.0*, by Ronald W. Richard, June 1971, Unclassified.

The vehicle transfer function with three loops closed is

$$\frac{\dot{Z}}{e_3} = \frac{\frac{\theta}{e_2} \left(\frac{w}{\theta} - U_o \right)}{1 + \frac{\theta}{e_2} \left(\frac{w}{\theta} - U_o \right) K_Z^*}$$

$$\frac{\dot{Z}}{e_3} = \frac{N_{\delta s}^w - N_{\delta s}^\theta U_o}{D - s N_{\delta s}^\theta K_\theta^* + K_\theta N_{\delta s}^\theta + K_Z^* (N_{\delta s}^w - N_{\delta s}^\theta U_o)}$$

For stability $K_Z^* \geq 0$.

Note that from Figure 4 this is a *positive feedback system*. This convention was chosen to conform with the Navy's standard motion simulation program in Hildebrand's textbook⁽⁶⁾. The reader should note the difference between a negative feedback and a positive feedback root locus. In a negative feedback system, the locus of roots on the real axis lies to the left of an odd number of poles or zeros. In a *positive feedback system*, the locus of roots on the real axis lies to the right of an odd number of poles or zeros. In both cases, the locus emanates from a pole and terminates at a zero.

Also note from Figure 4 that the value for K_Z is dimensionalized by dividing it by 57.3 ($\approx 4 \arctan 1$).

FOURTH LOOP

Figure 5 shows the model of the fourth and final loop. The vehicle transfer function with three loops closed is denoted by Z/e_3 . The vehicle transfer function with four loops closed is

$$\frac{Z}{Z_o} = \frac{-K_Z \frac{Z}{e_3} \frac{1}{s}}{1 - K_Z \frac{Z}{e_3} \frac{1}{s}}$$

$$\frac{Z}{Z_o} = \frac{-K_Z (N_{\delta s}^w - N_{\delta s}^\theta U_o)}{s [D - s N_{\delta s}^\theta K_\theta^* + K_\theta N_{\delta s}^\theta + K_Z^* (N_{\delta s}^w - N_{\delta s}^\theta U_o)] - K_Z (N_{\delta s}^w - N_{\delta s}^\theta U_o)}$$

⁽⁶⁾ ibid.

(Text Continued on Page 9)

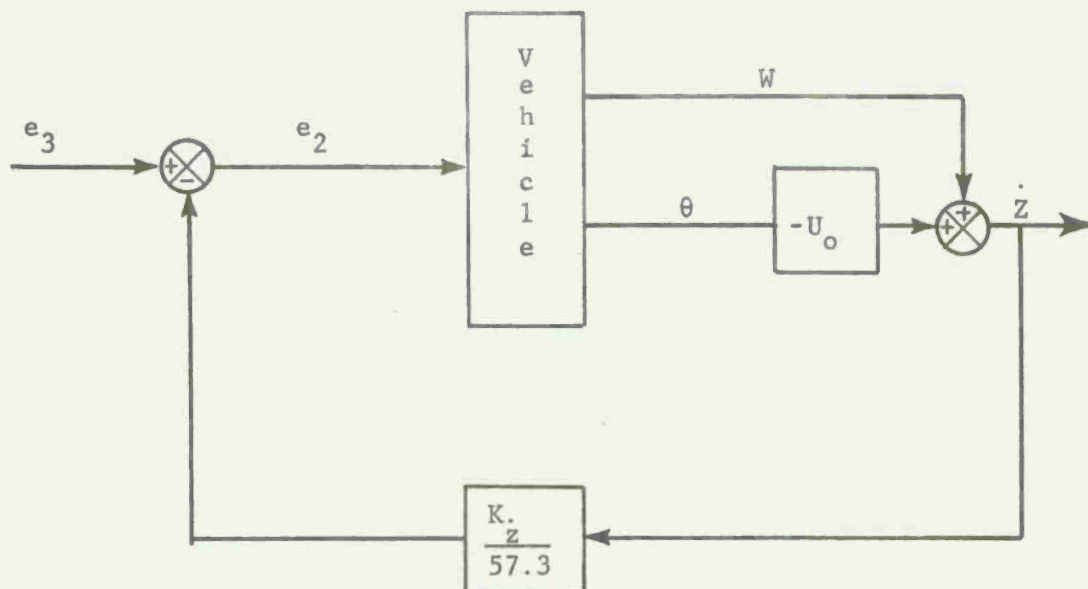


FIGURE 4. BLOCK DIAGRAM FOR THE THIRD LOOP

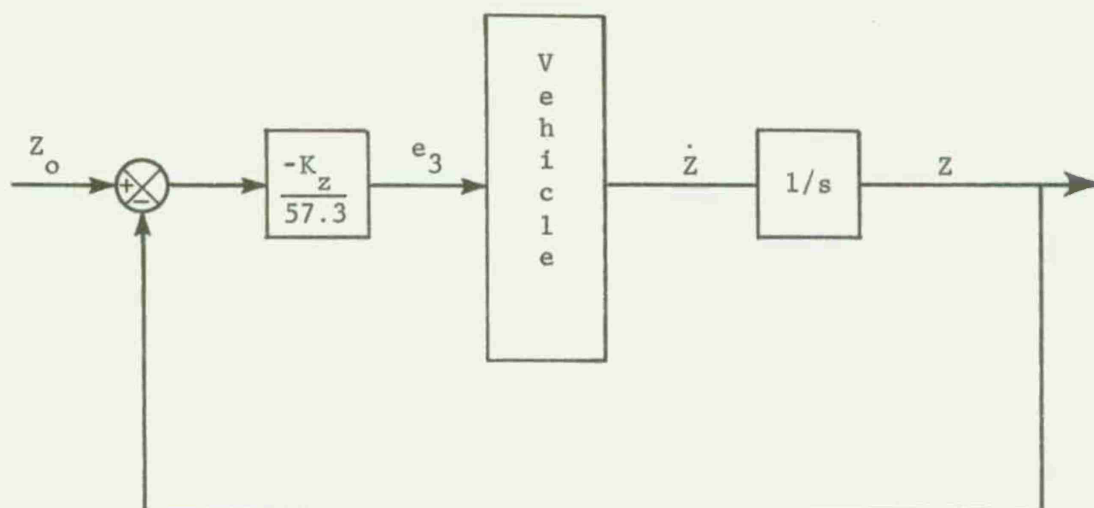


FIGURE 5. BLOCK DIAGRAM FOR THE FOURTH LOOP

For stability: $K_Z \leq 0$.

Note again that this is *positive feedback*. Again notice that the value of K_Z is dimensionalized by dividing it by 57.3.

The augmented vehicle response is achieved by the commanded deflection of the stern plane. The control law that determines the stern plane position as a function of time is seen from Figure 1 as

$$\delta_s = K_\theta \dot{\theta} = (\theta_o - \theta) K_\theta - K_Z \dot{Z} - (Z_o - Z) K_Z.$$

This conforms to the control law used in Reference 8.

COMPUTER PROGRAM USERS GUIDE

BASIC PROGRAM DESCRIPTION

This program computes the roots of the numerator and denominator for a submerged vehicle with the feedback control system described in the previous section and Appendix C. The basic inputs to the program are the vehicle nondimensional hydrodynamic coefficients, mass, and moments of inertia as defined in a SNAME publication⁽⁸⁾. The vehicle length and speed, whether the vehicle is a towed or self-propelled body (input by means of a disk data file), and the range of loop gains to be analyzed (input systematically from an interactive terminal).

The main program computes the coefficients of the numerator and denominator equations that are shown in Appendix B. This calculation is broken down into four basic segments; one segment for each loop analyzed. The resulting polynomial equations are solved for the roots of the system by using a polynomial root factoring routine.

At the completion of the main analysis program, a data file can be automatically written for a companion program known as TIMEPLT (Appendix D). TIMEPLT is another analysis program which takes the S-domain

⁽⁸⁾Air Force Flight Dynamics Laboratory Research and Technology Division, Wright-Patterson Air Force Base, *Analysis of Multiloop Vehicular Control Systems*, by D. T. McRuer, I. L. Ashkenas, and H. R. Pass, March 1964.

⁽⁹⁾Society of Naval Architects and Mechanical Engineers, *Nomenclature for Treating the Motion of a Submerged Body through a Fluid*, 1952.

analysis and transfers it back into the time domain, complete with plot diagrams of the system output response due to standard input signals.

INPUT REQUIREMENTS

Data input to the object program LØCSAP/ØBJECT is through the disk data file LØCSAP/DATA and from an interactive terminal. All inputs are in a free-field format. Refer to Figure 6 for an example of data file input.

INTERACTIVE TERMINAL DATA

As described earlier, the control system analysis proceeds by closing each successive feedback loop. This is accomplished by inputting a range of loop gains for the innermost loop and then deciding on a single value before proceeding to the next system loop. At any point in the analysis, the operator is allowed to change any previous loop gains until a full set of four loop gains have been chosen. Details of the interactive terminal data can be found in the example problem (Figure 7).

EXAMPLE PROBLEM*

The program is executed as follows:

```
??EX LØCSAP/ØBJECT * [charge number]
```

```
FILE FILE 1 = LØCSAP/DATA;END.
```

At the interactive terminal, the programer is now allowed to have outputted three different data sets. The program prints the statement:

```
PRINT STAB DERIV, DIMRTS, TFCPRT . . .
```

These data sets are the vehicle nondimensional stability derivatives (input to the program from the previously mentioned data file LØCSAP/DATA), the dimensional polynomial roots, and the dimensional polynomial coefficients, respectively. A "1" input for each of the variables allows the data to be printed out; an "0" means that the printout is not desired. Following these data sets, the numerator roots (zeros) for the four control system loops are printed.

*This example problem is for a self-propelled vehicle, consequently some additional terms are zeroed by the program.

(Text Continued on Page 14)

	Line Number	Data Description
Record 1	0000:	10.1270, 49.3330, 0,*
Record 2	0010:	-.15020E-01, -.57700E-03, .81000E-04,/
	0020:	.0, -.50138E-01, .95500E-02,/
	0030:	.0, .0, -.15709E 00,/
	0040:	.0, -.17455E-01, -.11310E-01,/
	0050:	-.16230E-02, .0, .0,/
	0060:	.0, -.31545E-01, -.14600E-03,/
	0070:	.0, -.13000E-03, -.15730E-02,/
	0080:	.0, .0, .0,/
	0090:	.0, .0, .0,/
	0100:	.0, -.27695E-01, -.12797E-01,/
	0110:	.36397E-01, .19170E-02,

Note: The following information identifies the input by record, column, and line.

		UQ, LB, [Type]
Record 1	0000:	U $\emptyset$ - Vehicle Velocity (ft/sec)
		LB - Vehicle length (ft)
		[Type] - 1 if a towed body,
		0 if self-propelled
Record 2*	0010:	XU, ZU, MU
	0020:	XW, ZW, MW
	0030:	XTHUSQ**, ZTHUSQ**, MTHUSQ**
	0040:	XQ, ZQ, MQ
	0050:	XUD, ZUD, MUD
	0060:	XWD, ZWD, MWD
	0070:	XQD, ZQD, MQD
	0080:	XX, ZX, MX
	0090:	XZ, ZZ, MZ
	0100:	XDELT, ZDELT, MDELT
	0110:	M, IY,

*Nondimensional Stability Derivatives (Appendix A)

**Program reads XTHUSQ(=X'₀ U²), ZTHUSQ, MTHUSQ and converts to XTH(= XTHUSQ/U²), ZTH, MTH

FIGURE 6. EXAMPLE DATA FILE

5:LOCSAP/ØBJECT = 1 BOJ 1302 10/01/74 FROM 01/06
 PRINT STAB DERIV. DIMRTS, TFCPRT...
 1,1,1←

S N A M E NON-DIMENSIONAL
 LONGITUDINAL STABILITY DERIVATIVES

XU	= -.15020E-01	ZU	= -.57700E-03	MU	= .81000E-04
XW	= .0	ZW	= -.50138E-01	MW	= .95500E-02
XTH	= .0	ZTH	= .0	MTH	= .15317E-02
XQ	= .0	ZQ	= -.17455E-01	MQ	= -.11310E-01
XUD	= -.16230E-02	ZUD	= .0	MUD	= .0
XWD	= .0	ZWD	= -.31545E-01	MWD	= -.14600E-03
XQD	= .0	ZQD	= -.13000E-03	MQD	= -.15730E-02
XX	= .0	ZX	= .0	MX	= .0
XZ	= .0	ZZ	= .0	MZ	= .0
XDELT	= .0	ZDELT	= -.27695E-01	MDELT	= -.12797E-01
M	.36397E-01	IY	= .19170E-02		

**** DENOMINATOR DS(J) ****

DIMENSIONAL COEFFICIENTS

J = 1	DS = .0
J = 2	DS = .0
J = 3	DS = .273739387358E-04
J = 4	DS = .118877286995E-02
J = 5	DS = .185102157604E-01
J = 6	DS = .108587460950E 00
J = 7	DS = .120470690538E 00

DIMENSIONAL ROOTS

J = 1	ROOTR = -.59231E-01	ROOTI = -.22004E-01
J = 2	ROOTR = -.59231E-01	ROOTI = .22004E-01
J = 3	ROOTR = -.81096E-01	ROOTI = .0
J = 4	ROOTR = -.70180E 00	ROOTI = .0
J = 5	ROOTR = .0	ROOTI = .0
J = 6	ROOTR = .0	ROOTI = .0

**** X NUMERATOR ****

***** XS(J) COEFFICIENTS ALL ZERO *****

FIGURE 7. EXAMPLE PROBLEM
 (Sheet 1 of 2)

**** Z NUMERATOR ****

DIMENSIONAL COEFFICIENTS

J =	1	ZS =	.0
J =	2	ZS =	-.153127243398E-03
J =	3	ZS =	-.116585064408E-01
J =	4	ZS =	-.128614635109E 00
J =	5	ZS =	-.100337496561E 00

DIMENSIONAL ROOTS

J =	1	ROOTR =	-.15883E-01	ROOTI =	.0
J =	2	ROOTR =	-.81096E-01	ROOTI =	.0
J =	3	ROOTR =	-.11848E 01	ROOTI =	.0
J =	4	ROOTR =	.0	ROOTI =	.0

**** I NUMERATOR ****

DIMENSIONAL COEFFICIENTS

J =	1	TS =	.0
J =	2	TS =	.0
J =	3	TS =	-.322969154535E-03
J =	4	TS =	-.548520303593E-02
J =	5	TS =	-.185293864643E-01

DIMENSIONAL ROOTS

J =	1	ROOTR =	-.81096E-01	ROOTI =	.0
J =	2	ROOTR =	-.21493E 00	ROOTI =	.0
J =	3	ROOTR =	.0	ROOTI =	.0
J =	4	ROOTR =	.0	ROOTI =	.0

ZEROS OF TD/E1

-0.0811	0.0000	-0.2149	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000						

ZEROS OF T/E2

-0.0811	0.0000	-0.2149	0.0000	0.0000	0.0000	0.0000	0.0000
---------	--------	---------	--------	--------	--------	--------	--------

ZEROS OF ZD/E3 AND Z/ZO

-0.0811	0.0000	-0.3690	0.0000	1.0384	0.0000	0.0000	0.0000
0.0000	0.0000						

FIGURE 7.
(Sheet 2 of 2)

FIRST LOOP ANALYSIS

To analyze the first loop (pitch rate), the initial value of K_0 must be entered along with the step increment, ΔK_0 , and the final value of K_0 . The program prints out the statement:

ENTER KTDORG, DELKTD, KTDFIN.

The programmer must then enter the desired values as follows:

0, .5, 5.

A root locus for the inner loop will then be generated for the gain values of 0 to 5 in steps of .5 (Figure 8).

ENTER KTDORG,DELKTD,KTDFIN

```
0,.5,5+
0.00
-0.0592 -0.0220 -0.0592 0.0220 -0.0811 0.0000 -0.7018 0.0000
0.0000 0.0000 0.0000 0.0000
0.50
-0.0407 0.0000 -0.0811 0.0000 -0.0897 0.0000 -0.7667 0.0000
0.0000 0.0000 0.0000 0.0000
1.00
-0.0307 0.0000 -0.0811 0.0000 -0.1094 0.0000 -0.8340 0.0000
0.0000 0.0000 0.0000 0.0000
1.50
-0.0253 0.0000 -0.0811 0.0000 -0.1227 0.0000 -0.9030 0.0000
0.0000 0.0000 0.0000 0.0000
2.00
-0.0217 0.0000 -0.0811 0.0000 -0.1328 0.0000 -0.9734 0.0000
0.0000 0.0000 0.0000 0.0000
2.50
-0.0190 0.0000 -0.0811 0.0000 -0.1409 0.0000 -1.0449 0.0000
0.0000 0.0000 0.0000 0.0000
3.00
-0.0170 0.0000 -0.0811 0.0000 -0.1475 0.0000 -1.1172 0.0000
0.0000 0.0000 0.0000 0.0000
3.50
-0.0154 0.0000 -0.0811 0.0000 -0.1531 0.0000 -1.1901 0.0000
0.0000 0.0000 0.0000 0.0000
4.00
-0.0141 0.0000 -0.0811 0.0000 -0.1578 0.0000 -1.2637 0.0000
0.0000 0.0000 0.0000 0.0000
4.50
-0.0129 0.0000 -0.0811 0.0000 -0.1618 0.0000 -1.3376 0.0000
0.0000 0.0000 0.0000 0.0000
5.00
-0.0120 0.0000 -0.0811 0.0000 -0.1654 0.0000 -1.4119 0.0000
0.0000 0.0000 0.0000 0.0000
```

FIGURE 8. EXAMPLE PROBLEM: FIRST LOOP ANALYSIS

Each root is listed as a real and imaginary pair, read from left to right, top and bottom. For example, at a gain value $K_0 = 0.00$ (first three digit number printed) the denominator roots (poles) are

<u>Real</u>	<u>Imaginary</u>
-.0592	-j .0220
-.0592	+j .0220
-.0811	j0
-.7018	j0

While the loop zeros (previously printed out) are

<u>Real</u>	<u>Imaginary</u>
-.0811	j0
-.2149	j0
0.0	j0.0

The two zero vlaue roots for each are not shown since this example is for a self-propelled vehicle; for a towed vehicle they would have a nonzero value.

Figure 9 is a plot of the root locus for this loop.

After printing the value for the roots over the gain range specified, the program will then print out the next statement:

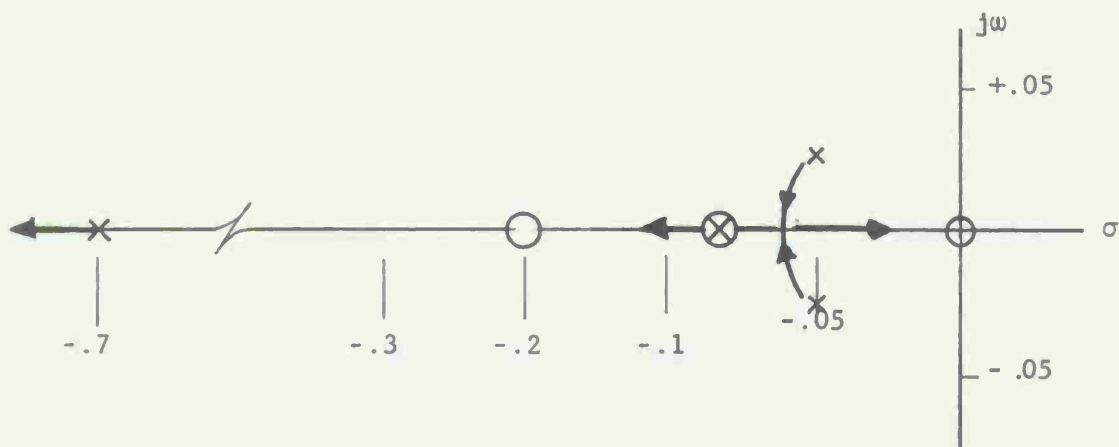


FIGURE 9. ROOT LOCUS FOR THE FIRST LOOP

DO YOU WANT TO CONTINUE KTD R. LOCUS.

The programmer must enter:

0 - if he desires to go on to the second loop.

1 - if he desires to continue the root locus in the first loop.

If 1 is entered, the program will again ask for values of KTDORG, DELKTD, KTDFIN.

If 0 is entered, the program will ask for the desired first loop gain by printing ENTER KTD. The programmer must enter the selected first loop gain.

SECOND, THIRD, AND FOURTH LOOPS

The above procedure is repeated for the next three loops. When new values are requested at the end of each root locus, three options are available. This is to allow the programmer to go back to any desired inside loop at any point in the program. For example, if after completing the root locus in the second loop, the programmer may wish to reanalyze the first loop based on what he learned from the second loop root locus. This is accomplished as follows. The program prints out the statement

DO / YOU WANT NEW KTD, 0 = NO, 1 = ENT, 2 = ENT & COMP, KTD NOW = 1
after the second loop root locus is complete. The programmer must enter

0 - If he desires to go to the third loop,

1 - If he desires to change the value of the first loop gain,

2 - If he desires to go back to the first loop and recompute the root locus.

This return option is available at the completion of each loop's locus.

For the fourth loop, two gain values are printed, the loop gain and the system gain. The system gain is defined as

$$\text{GAIN} = K_Z \frac{ZS(5)}{DS(5)}$$

for self-propelled vehicles (see Appendix B for definition of terms). The gain is the steady state value for depth, Z , for a unit step in depth command, Z_0 . Normally, it is desirable for this gain value to be equal to unity.

The remainder of the vehicle control system analysis follows as shown in an example problem (Figure 10), along with root locus plots for the second, third, and fourth loops (Figures 11, 12, and 13).

When the programmer is satisfied with the analysis and has selected gain values for all four loops, the program prints out

DØ YOU WANT A TIMEPLT.

By inputting a 1 (yes) to the LØCSAP program, a data file is automatically created which will later be utilized by the program TIMEPLT. The root values to be passed to the file are determined by the operators answer to the next two questions asked of him by the program.

ENTER KTD, KT, KZD, KZ, GAMP

TIMEPLT FØR WHICH LØØP .

The K values are the individual loop gains, and the variable GAMP is indicative of the type and magnitude of the standard test signal to be inputted to the system. Answering the question as to which loop, determines which loop output will be plotted by the companion program. This analysis can be repeated as many times as desired, until the operator answers NØ (0) to the question:

DØ YOU WANT ANØTHER TIMEPLT,

at which point the root-locus program is terminated.

CONCLUSIONS

A computer program was written utilizing the root-locus technique to analyze a longitudinal feedback control system. An example problem is included illustrating the use of this program.

To date, this analysis program has been utilized in the design of control systems for swimmer delivery vehicles, a submarine, and towed mine-hunting vehicles.

DO YOU WANT TO CONTINUE KTD R. LOCUS

0←

ENTER KTD

1←

ENTER KTORG,DELKT,KTFIN

0,-.5,-5←

0.00							
-0.0307	0.0000	-0.0811	0.0000	-0.1094	0.0000	-0.8340	0.0000
0.0000	0.0000	0.0000	0.0000				
-0.50							
-0.1150	-0.1130	-0.1150	0.1130	-0.0811	0.0000	-0.7441	0.0000
0.0000	0.0000	0.0000	0.0000				
-1.00							
-0.1724	-0.1651	-0.1724	0.1651	-0.0811	0.0000	-0.6292	0.0000
0.0000	0.0000	0.0000	0.0000				
-1.50							
-0.0811	0.0000	-0.2584	-0.2186	-0.2584	0.2186	-0.4572	0.0000
0.0000	0.0000	0.0000	0.0000				
-2.00							
-0.0811	0.0000	-0.3110	0.0000	-0.3315	-0.3342	-0.3315	0.3342
0.0000	0.0000	0.0000	0.0000				
-2.50							
-0.0811	0.0000	-0.2726	0.0000	-0.3508	-0.4364	-0.3508	0.4364
0.0000	0.0000	0.0000	0.0000				
-3.00							
-0.0811	0.0000	-0.2565	0.0000	-0.3588	-0.5185	-0.3588	0.5185
0.0000	0.0000	0.0000	0.0000				
-3.50							
-0.0811	0.0000	-0.2476	0.0000	-0.3632	-0.5888	-0.3632	0.5888
0.0000	0.0000	0.0000	0.0000				
-4.00							
0.0811	0.0000	-0.2410	0.0000	0.3661	0.6514	-0.3661	0.6514
0.0000	0.0000	0.0000	0.0000				
-4.50							
-0.0811	0.0000	-0.2379	0.0000	-0.3681	-0.7083	-0.3681	0.7083
0.0000	0.0000	0.0000	0.0000				
-5.00							
-0.0811	0.0000	-0.2349	0.0000	-0.3696	-0.7609	-0.3696	0.7609
0.0000	0.0000	0.0000	0.0000				

DO YOU WANT NEW KTD,0=NO,1=ENT,2=ENT&COMP,KTD NOW= 1.00

0←

DO YOU WANT TO CONTINUE KT R. LOCUS

0←

FIGURE 10. EXAMPLE PROBLEM: SECOND, THIRD AND FOURTH LOOP ANALYSIS
(Sheet 1 of 5)

ENTER KT

-1←

ENTER KZDORG, DELKZD, KZDFIN

0, .5, 5←

0.00

-0.1724	-0.1651	-0.1724	0.1651	-0.0811	0.0000	-0.6292	0.0000
0.0000	0.0000	0.0000	0.0000				

0.50

-0.0811	0.0000	-0.3992	0.0000	-0.0792	-0.6952	-0.0792	0.6952
0.0000	0.0000	0.0000	0.0000				

1.00

-0.0811	0.0000	-0.3839	0.0000	0.1213	-0.9539	0.1213	0.9539
0.0000	0.0000	0.0000	0.0000				

1.50

-0.0811	0.0000	-0.3788	0.0000	0.3270	-1.1186	0.3270	1.1186
0.0000	0.0000	0.0000	0.0000				

2.00

-0.0811	0.0000	-0.3764	0.0000	0.5340	-1.2271	0.5340	1.2271
0.0000	0.0000	0.0000	0.0000				

2.50

-0.0811	0.0000	-0.3749	0.0000	0.7415	-1.2938	0.7415	1.2938
0.0000	0.0000	0.0000	0.0000				

4.00

-0.0811	0.0000	-0.3739	0.0000	0.9492	-1.3249	0.9492	1.3249
0.0000	0.0000	0.0000	0.0000				

3.50

-0.0811	0.0000	-0.3732	0.0000	1.1571	-1.3229	1.1571	1.3229
0.0000	0.0000	0.0000	0.0000				

3.00

-0.0811	0.0000	-0.3726	0.0000	1.3650	-1.2877	1.3650	1.2877
0.0000	0.0000	0.0000	0.0000				

4.50

-0.0811	0.0000	-0.3722	0.0000	1.5731	-1.2164	1.5731	1.2164
0.0000	0.0000	0.0000	0.0000				

5.00

-0.0811	0.0000	-0.3719	0.0000	1.7811	-1.1019	1.7811	1.1019
0.0000	0.0000	0.0000	0.0000				

DO YOU WANT NEW KTD, KTD NOW = 1.00

0←

DO YOU WANT NEW KT, KT NOW = -1.00

0←

DO YOU WANT TO CONTINUE KZD R. LOCUS

1←

ENTER KZDORG, DELKZD, KZDFIN

0, -.5, -3←

FIGURE 10.
(Sheet 2 of 5)

0.00							
-0.1724	-0.1651	-0.1724	0.1651	-0.0811	0.0000	-0.6292	0.0000
0.0000	0.0000	0.0000	0.0000				
-0.50							
-0.0811	0.0000	0.2740	0.0000	-0.3410	0.0000	-1.3235	0.0000
0.0000	0.0000	0.0000	0.0000				
-1.00							
-0.0811	0.0000	-0.3547	0.0000	0.4253	0.0000	-1.8776	0.0000
0.0000	0.0000	0.0000	0.0000				
-1.50							
-0.3594	0.0000	0.5174	0.0000	-0.0811	0.0000	-2.3814	0.0000
0.0000	0.0000	0.0000	0.0000				
-2.00							
-0.0811	0.0000	-0.3617	0.0000	0.5822	0.0000	-2.8603	0.0000
0.0000	0.0000	0.0000	0.0000				
-2.50							
-0.0811	0.0000	-0.3632	0.0000	0.6311	0.0000	-3.3242	0.0000
0.0000	0.0000	0.0000	0.0000				
-3.00							
-0.0811	0.0000	-0.3641	0.0000	0.6698	0.0000	-3.7783	0.0000
0.0000	0.0000	0.0000	0.0000				
DO YOU WANT NEW KTD, KTD NOW = 1.00							
0←							
DO YOU WANT NEW KT, KT NOW = -1.00							
0←							
DO YOU WANT TO CONTINUE KZD R. LOCUS							
0←							
ENTER KZD							
0,.5,5←							
0.00							
GAIN = 0.000							
-0.1724	-0.1651	-0.1724	0.1651	-0.0811	0.0000	-0.6292	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		
0.50							
GAIN = -2.710							
-0.0811	0.0000	-0.3803	0.0000	0.4495	0.0000	-0.5216	-0.8133
-0.5216	0.8133	0.0000	0.0000	0.0000	0.0000		
1.00							
GAIN = -5.421							
-0.0811	0.0000	-0.3745	0.0000	0.5577	0.0000	-0.5787	-1.0922
-0.5787	1.0922	0.0000	0.0000	0.0000	0.0000		
1.50							
GAIN = -8.131							
-0.0811	0.0000	0.6230	0.0000	-0.3726	0.0000	-0.6122	-1.2989
-0.6122	1.2989	0.0000	0.0000	0.0000	0.0000		

FIGURE 10.
(Sheet 3 of 5)

2.00
GAIN = -10.841
-0.0811 0.0000 -0.3717 0.0000 0.6690 0.0000 -0.6357 -1.4705
-0.6357 1.4705 0.0000 0.0000 0.0000 0.0000
2.50
GAIN = -13.552
-0.0811 0.0000 -0.3711 0.0000 0.7041 0.0000 -0.6535 -1.6204
-0.6535 1.6204 0.0000 0.0000 0.0000 0.0000
3.00
GAIN = -16.262
-0.0811 0.0000 -0.3708 0.0000 0.7321 0.0000 -0.6677 -1.7552
-0.6677 1.7552 0.0000 0.0000 0.0000 0.0000
3.50
GAIN = -18.972
-0.0811 0.0000 -0.3705 0.0000 0.7552 0.0000 -0.6794 -1.8788
-0.6794 1.8788 0.0000 0.0000 0.0000 0.0000
4.00
GAIN = -21.683
-0.0811 0.0000 -0.3703 0.0000 0.7746 0.0000 -0.6892 -1.9937
-0.6892 1.9937 0.0000 0.0000 0.0000 0.0000
4.50
GAIN = -24.393
-0.0811 0.0000 -0.3702 0.0000 0.7913 0.0000 -0.6976 -2.1014
-0.6976 2.1014 0.0000 0.0000 0.0000 0.0000
5.00
GAIN = -27.103
-0.0811 0.0000 -0.3701 0.0000 0.8058 0.0000 -0.7049 -2.2032
-0.7049 2.2032 0.0000 0.0000 0.0000 0.0000
DO YOU WANT NEW KTD, KTD NOW = 1.00
0←
DO YOU WANT NEW KT, KT NOW = -1.00
0←
DO YOU WANT NEW KZD, KZD NOW = 0.00
0←
DO YOU WANT TO CONTINUE KZ R. LOCUS
1←
ENTER KZORG, DELKZ, KZFIN
0, -.5, -3←
0.00
GAIN = 0.000
-0.1724 -0.1651 -0.1724 0.1651 -0.0811 0.0000 -0.6292 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
-0.50
GAIN = 2.710
-0.0811 0.0000 -0.3586 0.0000 0.3000 -0.5254 0.3000 0.5254
-1.2154 0.0000 0.0000 0.0000 0.0000 0.0000
-1.00

FIGURE 10.
(Sheet 4 of 5)

GAIN = 5.421							
-0.0811	0.0000	-0.3637	0.0000	0.4504	-0.6146	0.4504	0.6146
-1.5113	0.0000	0.0000	0.0000	0.0000	0.0000		
-1.50							
GAIN = 8.131							
-0.0811	0.0000	0.5629	-0.6620	0.5629	0.6620	-0.3654	0.0000
-1.7345	0.0000	0.0000	0.0000	0.0000	0.0000		
-2.00							
GAIN = 10.841							
-0.0811	0.0000	0.6564	-0.6901	0.6564	0.6901	-0.3663	0.0000
-1.9206	0.0000	0.0000	0.0000	0.0000	0.0000		
-2.50							
GAIN = 13.552							
0.7380	-0.7066	0.7380	0.7066	-0.0811	0.0000	-0.3668	0.0000
-2.0832	0.0000	0.0000	0.0000	0.0000	0.0000		
-3.00							
GAIN = 16.262							
-0.0811	0.0000	-0.3672	0.0000	0.8112	-0.7151	0.8112	0.7151
-2.2293	0.0000	0.0000	0.0000	0.0000	0.0000		

DO YOU WANT NEW KTD,KTD NOW = 1.00

0←

DO YOU WANT NEW KT, KT NOW = -1.00

0←

DO YOU WANT NEW KZD,KZD NOW = 0.00

0←

DO YOU WANT TO CONTINUE KZ R. LOCUS

0←

DO YOU WANT A TIMEPLT

1←

ENTER KTD,KT,KZD,KZ,GAMP

1,-1,0,0,10←

TIMEPLT FOR WHICH LOOP; ENTER 1=1ST, 2=2ND, 3=3RD, 4=4TH

2←

DO YOU WANT ANOTHER TIMEPLT

0←

PROCESSOR TIME	=	15 SEC	\$ 0.60
I/O TIME	=	23 SEC	\$ 0.46
PRORATED TIME	=	249 SEC	\$ 2.49
		TOTAL COST	\$ 3.55

LOCSAP/MILLER = 1 EOJ 1326

FIGURE 10.
(Sheet 5 of 5)

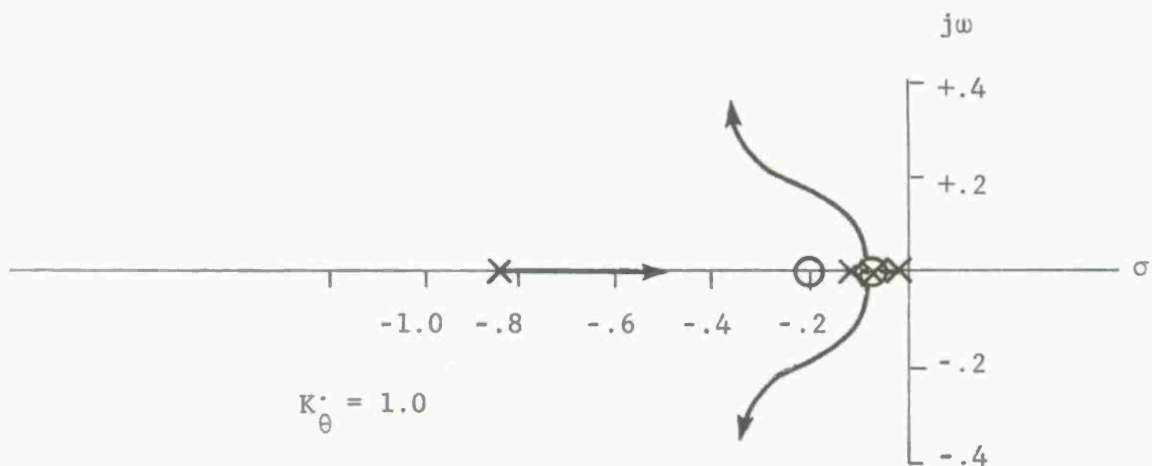


FIGURE 11 ROOT LOCUS FOR THE SECOND LOOP

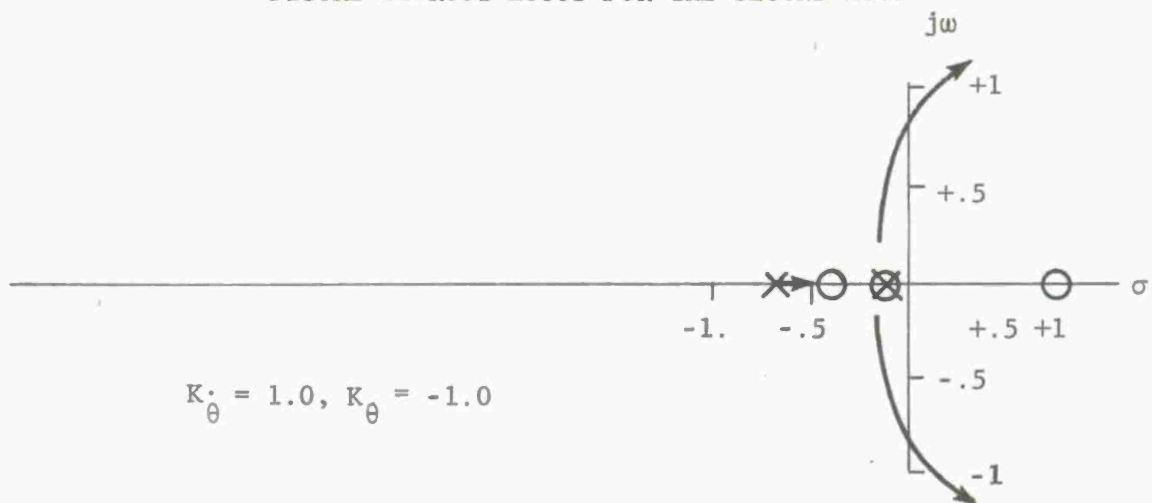


FIGURE 12. ROOT LOCUS FOR THE THIRD LOOP

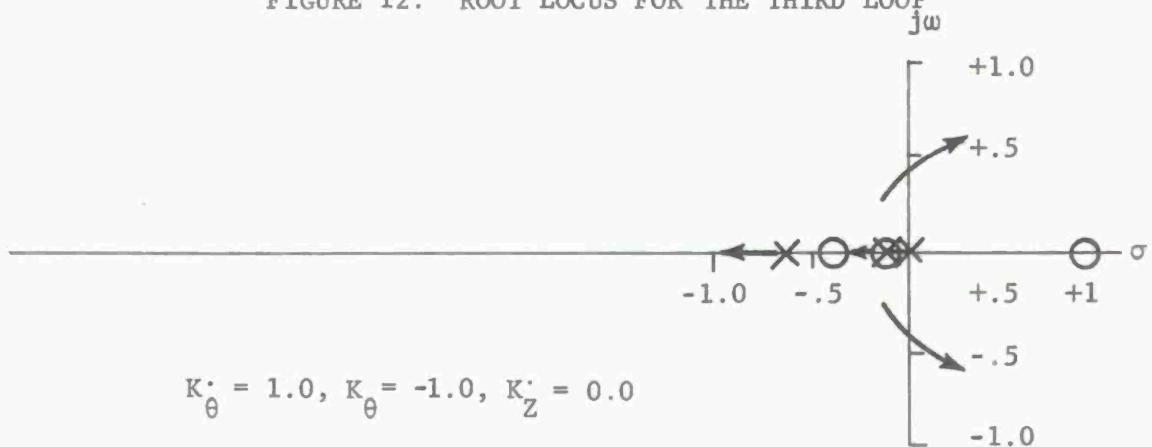
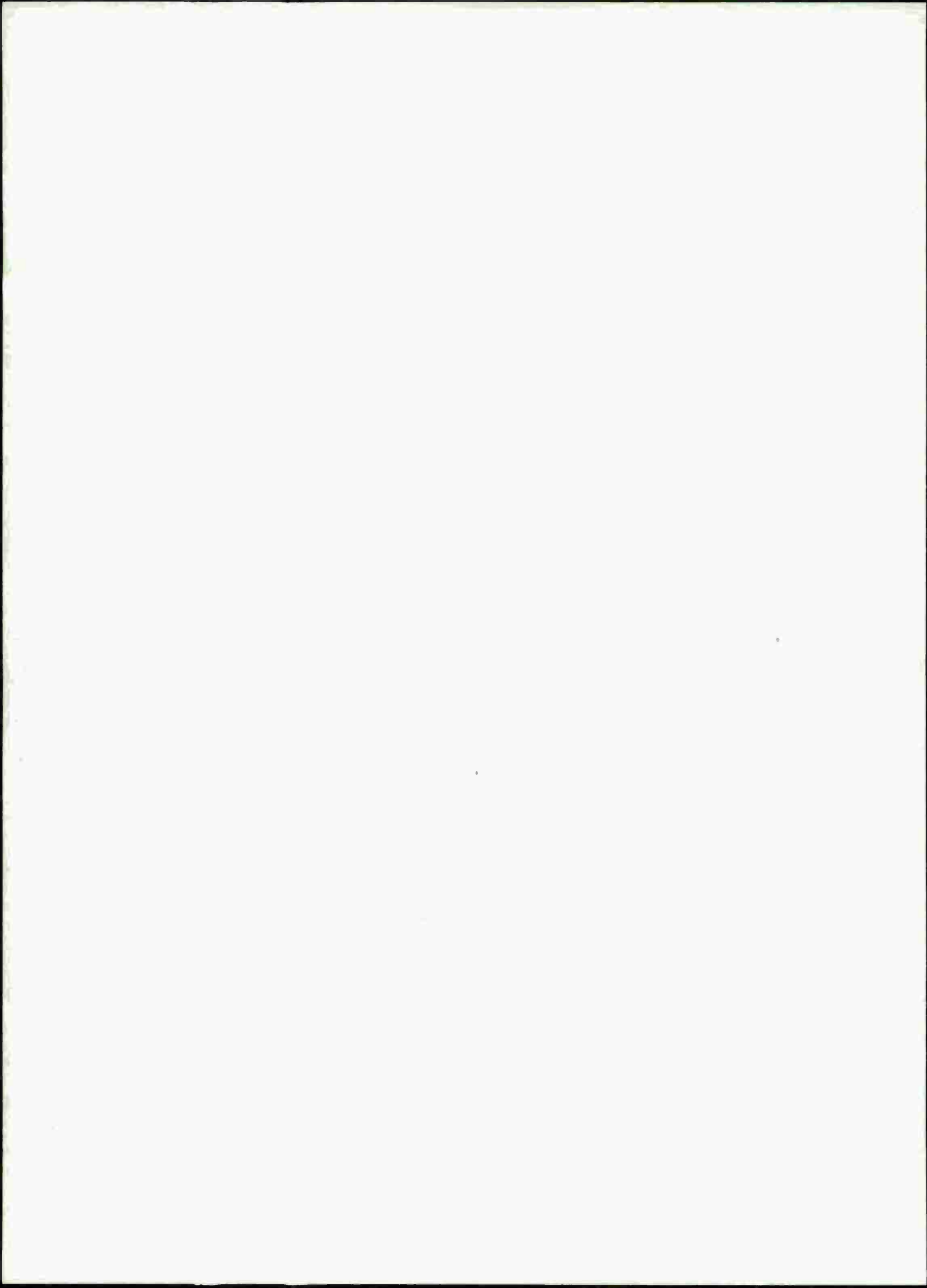


FIGURE 13. ROOT LOCUS FOR THE FOURTH LOOP



APPENDIX A

SOME NOTES ON THE CONSTRUCTION AND INTERPRETATION OF ROOT LOCUS

A physical system can be represented by a block diagram composed of individual blocks that represent the various components of the system as shown in Figure A1. Each block is described by one or more

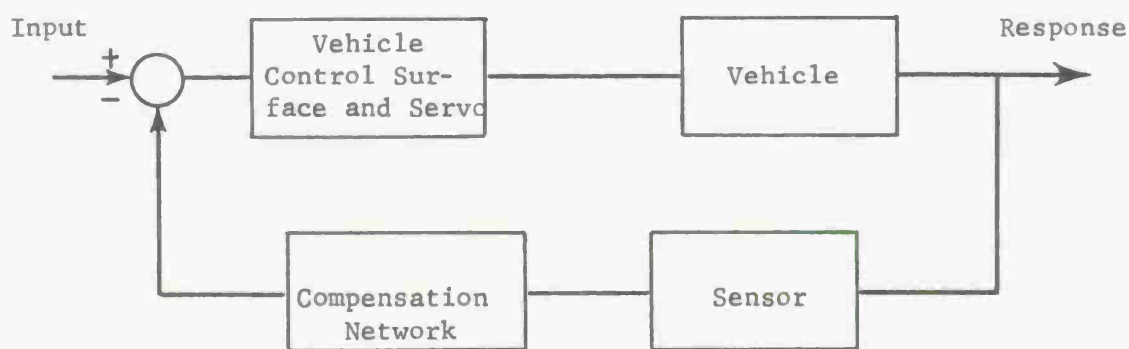


FIGURE A1. A TYPICAL BLOCK DIAGRAM

differential equations according to Newton's Second Law of Motion or its electrical equivalent. Combining the characteristics of each block to form the characteristics of the overall system is quite difficult because a signal is modified in both phase and amplitude in going through each block. By applying the Laplace transform

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt$$

to the describing differential equations, one obtains an algebraic representation for each block in the system. It is then convenient to arrange this representation in the form of a transfer function; i.e., as a ratio of block response to block excitation. These transfer functions can be multiplied together to yield system response to system excitation. It is a relatively straightforward procedure because each transfer function is merely a ratio of polynomials in the Laplace operator s .

Once the transfer functions for each of the blocks have been formed, it is then necessary to examine the effects of changing various unknown system parameters, such as feedback gain on the system dynamics. The root-locus diagram was developed to facilitate such an analysis. As the name implies, it shows on one figure the trajectory that the frequency and damping characteristic modes follow as system parameters are changed.

Consider the transfer function

$$\frac{O}{I} = \frac{K(s + a)(s^2 + bs + c)}{s(s + d)(s + e)(s^2 + fs + g)}.$$

The denominator of the transfer function represents the characteristic equation of the system; e.g., the equation describing the free motion of the system (the response independent of control input). It is responsible for the general solution of the system of differential equations. The particular solution comes from the numerator.

It will be observed that all values of s which make the denominator zero are solutions of the characteristic equation and therefore contribute a term of the e^{st} to the time response. Since for these roots the transfer function is undefined, denominator roots are called poles. Numerator roots are appropriately called zeros. It is customary to plot these poles and zeros on a graph whose abscissa is the real part of s and whose ordinate is the imaginary part. Poles are commonly depicted as x's and zeros are O's. A first order root; e.g., $(s + d)$, will always lie on the abscissa. A second order system has two roots. They may be real, in which case they lie on the abscissa, or they may be complex, in which case they are placed equidistant above and below the abscissa.

Any pole which lies in the right half s -plane represents an unstable motion. Zeros in the right half plane are significant in terms of the type motion only if the system depicted is a feedback system. In this case the zeros represent the location of the poles when the feedback gain is made infinite. For zeros in the right half plane then, the system will become unstable at some finite value of feedback gain. Knowledge of the location of the basic vehicle zeros is needed by designers to combine the control system characteristics with those of the vehicle to obtain the desired response without unexpected instabilities. Note also that a zero placed on top of a pole will eliminate the motion caused by that pole from the time history of the particular variable associated with the numerator (θ in θ/δ_r for example) but from no other time history.

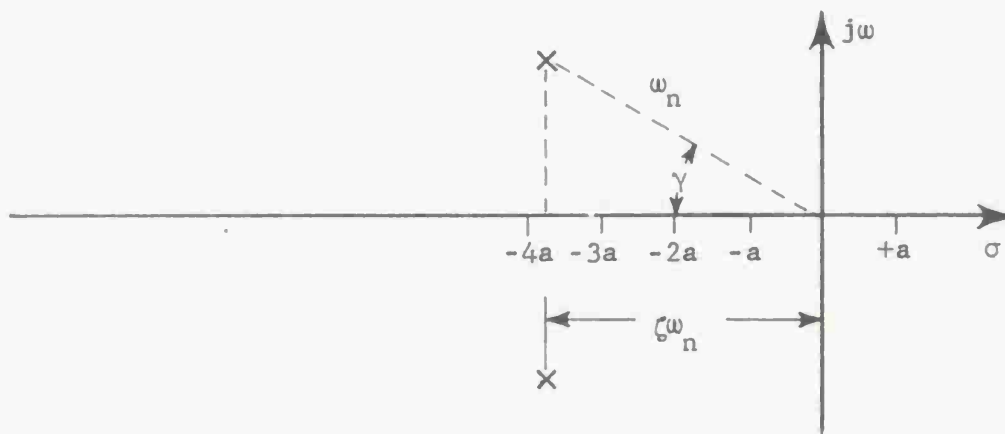
A pole located at $s = -3$, for example, means that there is contribution to the time history given by e^{-3t} . Thus, the further to the left the pole, the more rapid is the subsidence. Conversely, a pole at $s = 3$ means the motion has an unstable component described by e^{3t} .

Stable oscillatory modes, it will be recalled, have roots which can be expressed by

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$

Figure A2 indicates how varying either frequency or damping ratio separately moves the poles. It also shows that the product $\zeta\omega_n$ determines the time for an oscillation to decay to half amplitude. When $\zeta\omega_n = 0.591$, the oscillation will decay to half amplitude in 1 second. Smaller values of the product mean the time to damp to half amplitude is longer.

For further details on the construction and interpretation of root locus diagrams refer to *Introduction to Automatic Control Systems*, John Wiley and Son, 1962.



ζ = Damping Ratio = $\cos \gamma$

ω_n = Undamped Natural Frequency

$\zeta\omega_n$ = Total Damping

t_{ss} = Time to reach 0.95 steady state value = $3.0/\zeta\omega_n$

$t_{1/2}$ = Time to damp to half amplitude = $0.69/\zeta\omega_n$

t_2 = Time to double amplitude (unstable systems only) = $0.69/\zeta\omega_n$

FIGURE A2. IMPORTANT FEATURES OF ROOT-LOCUS DIAGRAM

(Reverse Page A-4 Blank)

APPENDIX B

EXPRESSIONS FOR THE LONGITUDINAL TRANSFER FUNCTION COEFFICIENTS

The longitudinal characteristic equation is

$$\Delta_{\text{Long}} = As'^4 + Bs'^3 + Cs'^2 + Ds' + E$$

where

$$\begin{aligned} A = & (m' - X'_{ii})(m' - Z'_{iw})(I'_y - M'_{\dot{q}}) - X'_{iw}M'_{ii}Z'_{\dot{q}} \\ & - Z'_{ii}M'_{iw}X'_{\dot{q}} - (m' - Z'_{iw})M'_{ii}X'_{\dot{q}} \\ & - Z'_{ii}X'_{iw}(I'_y - M'_{\dot{q}}) - M'_{iw}(m' - X'_{ii})Z'_{\dot{q}} . \end{aligned}$$

$$\begin{aligned} B = & -(m' - X'_{ii})(m' - Z'_{iw})M'_{\dot{q}} - Z'_{iw}(m' - X'_{ii})(I'_y - M'_{\dot{q}}) \\ & - X'_{ii}(m' - Z'_{iw})(I'_y - M'_{\dot{q}}) - X'_{iw}M'_{ii}(Z'_{\dot{q}} + m') \\ & - M'_{ii}X'_{iw}Z'_{\dot{q}} - M'_{ii}X'_{iw}Z'_{\dot{q}} - Z'_{ii}M'_{iw}X'_{\dot{q}} - Z'_{ii}M'_{iw}X'_{\dot{q}} \\ & - Z'_{ii}M'_{iw}X'_{\dot{q}} - (m' - Z'_{iw})M'_{ii}X'_{\dot{q}} - M'_{ii}(m' - Z'_{iw})X'_{\dot{q}} \\ & + M'_{ii}Z'_{iw}X'_{\dot{q}} + Z'_{ii}X'_{iw}M'_{\dot{q}} - Z'_{ii}X'_{iw}(I'_y - M'_{\dot{q}}) \\ & - Z'_{ii}X'_{iw}(I'_y - M'_{\dot{q}}) - M'_{iw}(m' - X'_{ii})(Z'_{\dot{q}} + m') \\ & - M'_{iw}(m' - X'_{ii})Z'_{\dot{q}} + M'_{iw}X'_{ii}Z'_{\dot{q}} . \end{aligned}$$

$$\begin{aligned}
C = & -(m' - X'_{ii})(m' - Z'_{iw})M'_{\theta} + Z'_{iw}(m' - X'_{ii})M'_{\theta} \\
& + X'_{ii}(m' - Z'_{iw})M'_{\theta} + Z'_{iw}X'_{ii}(I'_y - M'_{\theta}) - X'_{iw}M'_{ii}Z'_{\theta} \\
& - M'_{ii}X'_{iw}(Z'_{\theta} + m') - M'_{ii}X'_{iw}(Z'_{\theta} + m') - Z'_{ii}M'_{iw}X'_{\theta} \\
& - Z'_{ii}M'_{iw}X'_{\theta} - Z'_{ii}M'_{iw}X'_{\theta} - Z'_{ii}M'_{iw}X'_{\theta} \\
& - (m' - Z'_{iw})M'_{ii}X'_{\theta} - M'_{ii}(m' - Z'_{iw})X'_{\theta} + M'_{ii}Z'_{iw}X'_{\theta} \\
& + Z'_{iw}M'_{ii}X'_{\theta} + Z'_{ii}X'_{iw}M'_{\theta} + Z'_{ii}X'_{iw}M'_{\theta} + Z'_{ii}X'_{iw}M'_{\theta} \\
& - X'_{iw}Z'_{ii}(I'_y - M'_{\theta}) - M'_{iw}(m' - X'_{ii})Z'_{\theta} - M'_{iw}(m' - X'_{ii})(Z'_{\theta} + m') \\
& + M'_{iw}X'_{ii}(Z'_{\theta} + m') + X'_{ii}M'_{iw}Z'_{\theta} - M'_{ii}X'_{iw}Z'_{\theta} .
\end{aligned}$$

$$\begin{aligned}
D = & \mathcal{Z}'_w M'_\theta (m' - X'_u) + X'_u (m' - \mathcal{Z}'_{w'}) M'_\theta - \mathcal{Z}'_w X'_u M'_g \\
& - M'_u X'_{w'} \mathcal{Z}'_\theta - M'_u X'_w \mathcal{Z}'_\theta - M'_u X'_w (\mathcal{Z}'_g + m') \\
& - \mathcal{Z}'_u M'_w X'_\theta - \mathcal{Z}'_u M'_{w'} X'_\theta - \mathcal{Z}'_u M'_w X'_g \\
& - M'_u (m' - \mathcal{Z}'_{w'}) X'_\theta + M'_u \mathcal{Z}'_w X'_\theta + \mathcal{Z}'_w M'_u X'_g \\
& + \mathcal{Z}'_u X'_{w'} M'_\theta + \mathcal{Z}'_u X'_w M'_\theta + X'_w \mathcal{Z}'_u M'_g \\
& - M'_w (m' - X'_u) \mathcal{Z}'_\theta + M'_{w'} X'_u \mathcal{Z}'_\theta + M'_w X'_u (\mathcal{Z}'_g + m') .
\end{aligned}$$

$$\begin{aligned}
E = & - \mathcal{Z}'_w X'_u M'_\theta - M'_u X'_w \mathcal{Z}'_\theta - \mathcal{Z}'_u M'_w X'_\theta \\
& + \mathcal{Z}'_w M'_u X'_\theta + X'_w \mathcal{Z}'_u M'_\theta + X'_u M'_w \mathcal{Z}'_\theta .
\end{aligned}$$

The pitch response transfer function is

$$\frac{\theta}{\delta_s} = \frac{N_{\delta_s}^{\theta}}{\Delta_{\text{Long}}} = \frac{A_{\theta} s'^2 + B_{\theta} s' + C_{\theta}}{\Delta_{\text{Long}}}$$

where

$$A_{\theta} = M'_{\delta_e} (m' - X'_{iu}) (m' - Z'_{iw}) + Z'_{\delta_e} X'_{iw} M'_{iu} + X'_{\delta_e} Z'_{iu} M'_{iw} \\ - M'_{\delta_e} X'_{iw} Z'_{iu} + X'_{\delta_e} (m' - Z'_{iw}) M'_{iu} + Z'_{\delta_e} M'_{iw} (m' - X'_{iu}) .$$

$$B_{\theta} = -M'_{\delta_e} (m' - X'_{iu}) Z'_{iw} - M'_{\delta_e} X'_{iu} (m' - Z'_{iw}) + Z'_{\delta_e} X'_{iw} M'_{iu} \\ + Z'_{\delta_e} X'_{iw} M'_{iu} + X'_{\delta_e} Z'_{iu} M'_{iw} + X'_{\delta_e} Z'_{iu} M'_{iw} \\ - M'_{\delta_e} X'_{iw} Z'_{iu} - M'_{\delta_e} X'_{iw} Z'_{iu} + X'_{\delta_e} (m' - Z'_{iw}) M'_{iu} \\ - X'_{\delta_e} Z'_{iw} M'_{iu} - Z'_{\delta_e} M'_{iw} X'_{iu} + Z'_{\delta_e} M'_{iw} (m' - X'_{iu}) .$$

$$C_{\theta} = M'_{\delta_e} X'_{iu} Z'_{iw} + Z'_{\delta_e} X'_{iw} M'_{iu} + X'_{\delta_e} Z'_{iu} M'_{iw} \\ - M'_{\delta_e} X'_{iw} Z'_{iu} - X'_{\delta_e} Z'_{iw} M'_{iu} - Z'_{\delta_e} M'_{iw} X'_{iu} .$$

The vertical velocity transfer function is

$$\frac{W'}{\delta_s} = \frac{N_{\delta_s}^W}{\Delta_{\text{Long}}} = \frac{A_W s'^3 + B_W s'^2 + C_W s' + D_W}{\Delta_{\text{Long}}}$$

where

$$A_w = \mathcal{Z}'_{\delta_e} (m' - X'_{\dot{u}}) (I'_y - M'_{\dot{q}}) + X'_{\delta_e} M'_{\dot{u}} \mathcal{Z}'_{\dot{q}} + M'_{\delta_e} \mathcal{Z}'_{\dot{u}} X'_{\dot{q}} \\ - \mathcal{Z}'_{\delta_e} M'_{\dot{u}} X'_{\dot{q}} + X'_{\delta_e} \mathcal{Z}'_{\dot{u}} (I'_y - M'_{\dot{q}}) + M'_{\delta_e} (m' - X'_{\dot{u}}) \mathcal{Z}'_{\dot{q}} .$$

$$B_w = -\mathcal{Z}'_{\delta_e} (m' - X'_{\dot{u}}) M'_{\dot{q}} - \mathcal{Z}'_{\delta_e} X'_{\dot{u}} (I'_y - M'_{\dot{q}}) \\ + X'_{\delta_e} M'_{\dot{u}} \mathcal{Z}'_{\dot{q}} + X'_{\delta_e} M'_{\dot{u}} (\mathcal{Z}'_{\dot{q}} + m') \\ + M'_{\delta_e} \mathcal{Z}'_{\dot{u}} X'_{\dot{q}} + M'_{\delta_e} \mathcal{Z}'_{\dot{u}} X'_{\dot{q}} \\ - \mathcal{Z}'_{\delta_e} M'_{\dot{u}} X'_{\dot{q}} - \mathcal{Z}'_{\delta_e} M'_{\dot{u}} X'_{\dot{q}} \\ - X'_{\delta_e} \mathcal{Z}'_{\dot{u}} M'_{\dot{q}} + X'_{\delta_e} \mathcal{Z}'_{\dot{u}} (I'_y - M'_{\dot{q}}) \\ + M'_{\delta_e} (m' - X'_{\dot{u}}) (\mathcal{Z}'_{\dot{q}} + m') - M'_{\delta_e} X'_{\dot{u}} \mathcal{Z}'_{\dot{q}} .$$

$$\begin{aligned}
C_{\mu\nu} = & -\bar{Z}'_{\delta_e} (m' - X'_{\dot{\mu}}) M'_{\theta} + \bar{Z}'_{\delta_e} X'_{\mu} M'_{\theta} + X'_{\delta_e} M'_{\dot{\mu}} \bar{Z}'_{\theta} \\
& + X'_{\delta_e} M'_{\mu} (\bar{Z}'_{\theta} + m') + M'_{\delta_e} \bar{Z}'_{\dot{\mu}} X'_{\theta} + M'_{\delta_e} \bar{Z}'_{\mu} X'_{\theta} \\
& - \bar{Z}'_{\delta_e} M'_{\dot{\mu}} X'_{\theta} - \bar{Z}'_{\delta_e} M'_{\mu} X'_{\theta} - X'_{\delta_e} \bar{Z}'_{\dot{\mu}} M'_{\theta} - X'_{\delta_e} \bar{Z}'_{\mu} M'_{\theta} \\
& + M'_{\delta_e} (m' - X'_{\dot{\mu}}) \bar{Z}'_{\theta} - M'_{\delta_e} X'_{\mu} (\bar{Z}'_{\theta} + m') .
\end{aligned}$$

$$\begin{aligned}
D_{\mu\nu} = & \bar{Z}'_{\delta_e} X'_{\mu} M'_{\theta} + X'_{\delta_e} M'_{\mu} \bar{Z}'_{\theta} + M'_{\delta_e} \bar{Z}'_{\mu} X'_{\theta} \\
& - \bar{Z}'_{\delta_e} M'_{\mu} X'_{\theta} - X'_{\delta_e} \bar{Z}'_{\mu} M'_{\theta} - M'_{\delta_e} X'_{\mu} \bar{Z}'_{\theta} .
\end{aligned}$$

The forward speed transfer function is

$$\frac{U'}{\delta_s} = \frac{N_{\delta_s}^U}{\Delta_{\text{Long}}} = \frac{A_U s^3 + B_U s^2 + C_U s + D_U}{\Delta_{\text{Long}}}$$

where

$$A_{\mu} = X'_{\delta_e} (m' - Z'_{iw}) (I'_y - M'_{\dot{q}}) + M'_{\delta_e} X'_{iw} Z'_{\dot{q}} + Z'_{\delta_e} M'_{iw} X'_{\dot{q}} \\ + M'_{\delta_e} (m' - Z'_{iw}) X'_{\dot{q}} + Z'_{\delta_e} X'_{iw} (I'_y - M'_{\dot{q}}) - X'_{\delta_e} M'_{iw} Z'_{\dot{q}} .$$

$$B_{\mu} = -X'_{\delta_e} (m' - Z'_{iw}) M'_{\dot{q}} - X'_{\delta_e} (I'_y - M'_{\dot{q}}) Z'_{iw} \\ + M'_{\delta_e} X'_{iw} (Z'_{\dot{q}} + m') + M'_{\delta_e} X'_{iw} Z'_{\dot{q}} + Z'_{\delta_e} M'_{iw} X'_{\dot{q}} \\ + Z'_{\delta_e} M'_{iw} X'_{\dot{q}} + M'_{\delta_e} (m' - Z'_{iw}) X'_{\dot{q}} - M'_{\delta_e} Z'_{iw} X'_{\dot{q}} \\ - Z'_{\delta_e} X'_{iw} M'_{\dot{q}} + Z'_{\delta_e} X'_{iw} (I'_y - M'_{\dot{q}}) \\ - X'_{\delta_e} M'_{iw} (Z'_{\dot{q}} + m') - X'_{\delta_e} M'_{iw} Z'_{\dot{q}} .$$

$$\begin{aligned}
C_{\mu} = & -X'_{\delta_e} (m' - \bar{Z}'_{iw}) M'_{\theta} + X'_{\delta_e} \bar{Z}'_{iw} M'_{\theta} + M'_{\delta_e} X'_{iw} \bar{Z}'_{\theta} \\
& + M'_{\delta_e} X'_{iw} (\bar{Z}'_{\theta} + m') + \bar{Z}'_{\delta_e} M'_{iw} X'_{\theta} + \bar{Z}'_{\delta_e} M'_{iw} X'_{\theta} \\
& + M'_{\delta_e} (m' - \bar{Z}'_{iw}) X'_{\theta} - M'_{\delta_e} \bar{Z}'_{iw} X'_{\theta} - \bar{Z}'_{\delta_e} X'_{iw} M'_{\theta} \\
& - \bar{Z}'_{\delta_e} X'_{iw} M'_{\theta} - X'_{\delta_e} M'_{iw} \bar{Z}'_{\theta} - X'_{\delta_e} M'_{iw} (\bar{Z}'_{\theta} + m') .
\end{aligned}$$

$$\begin{aligned}
D_{\mu} = & X'_{\delta_e} \bar{Z}'_{iw} M'_{\theta} + M'_{\delta_e} X'_{iw} \bar{Z}'_{\theta} + \bar{Z}'_{\delta_e} M'_{iw} X'_{\theta} \\
& - M'_{\delta_e} \bar{Z}'_{iw} X'_{\theta} - \bar{Z}'_{\delta_e} X'_{iw} M'_{\theta} - X'_{\delta_e} M'_{iw} \bar{Z}'_{\theta} .
\end{aligned}$$

APPENDIX C
LONGITUDINAL CONTROL SYSTEM ANALYSIS
PROGRAM - CARD LIST

(Reverse Page C-2 Blank)


```

BSFT FRIFFORM SEQXFQ RESET IJST                                00000000
FILE 1=LCCSAP/DATA,UNIT=DISK,SAVE=30,BLOCKING=3,RECORD=10      00000010
FILE 3=N/S,UNIT=REMOTE,LOCK,RECORD=9                            00000020
FILE 4=LCCSAP/TIMEPLT,UNIT=DISK,SAVE=30,LOCK,AREA=200,         00000030
- BLOCKING=3,RECORD=10                                           00000040
C- ***** SOURCE FILE IS LCCSAP/JAN0275 *****              00000050
REAL KT,KTORG,KTFIN,KZD,KZDRG,KZDFIN,KZ,KZORG,KZFIN            00000060
REAL KTORG,KTD,KTDFIN,NVAL                                       00000070
DIMENSION RTDF1(7),RTF2(7),RZDE3(7),RZ70(8),RZDF3(6),RTNE1(6) 00000080
DIMENSION RTE2(6)                                                 00000090
DIMENSION OTDF1R(6),OTDF1I(6),OTE2R(6),OTE2I(6),OZDE3R(6),    00000100
- OZDE3I(6),TITLE(11)                                             00000110
RADIAN = 180/(4*ATAN(1))                                          00000120
READ(1,4000) (TITLE(I),I=1,11)                                   00000130
4000 FORMAT(11A6)                                                 00000140
WRITE(3,4001) (TITLE(I),I=1,11)                                   00000150
4001 FORMAT(/,11A6)                                               00000160
WRITE(3,3101)                                                     00000170
3101 FORMAT(/,"PRINT STAR DERIV, DIMRTS, TFCPRT...")           00000180
READ (3,/,END=99)NVAL,IDIMRTS,ITFCPRT; ISDPRT=NVAL              00000190
REAL IY,IB,M,MU,MW,MTH,MQ,MUD,MWD,MQD,MX,MZ,MDEL,MTHUSQ        00000200
READ(1,/,END=99) U,LB,ICARIF                                     00000210
104 FORMAT(////)                                                  00000220
301 READ(1,/,END=99)XU,ZU,MU,XW,ZW,MW,XTHUSQ,ZTHUSQ,MTHUSQ,XQ,ZQ,MQ, 00000230
- XUD,ZUD, MUD,XWD,ZWD,MWD,XQD,ZQD,MQD,XX,ZX,MX,XZ,ZZ,MZ,XDEL, 00000240
- ZDEL,MDEL,M,IY;XTH=XTHUSQ/(U*U);ZTH=ZTHUSQ/(U*U)             00000250
MTH=MTHUSQ/(U*U)                                                  00000260
311 IF (ISDPRT.EQ.0) GO TO 351                                    00000270
WRITE (3,321)                                                     00000280
321 FORMAT(/20X,"S N A M E NON-DIMENSIONAL")                  00000290
WRITE (3,331)                                                     00000300
331 FORMAT(16X,"LONGITUDINAL STABILITY DERIVATIVES"/)         00000310
WRITE(3,341)XU,ZU,MU,XW,ZW,MW,XTH,ZTH,MTH,XQ,ZQ,MQ,XUD,ZUD,MUD,XW 00000320
- ,ZWD,MWD,XQD,ZQD,MQD,XX,ZX,MX,XZ,ZZ,MZ,XDEL,ZDEL,MDEL,M,IY,U, 00000330
341 FORMAT(1X,"XU" =",F12.5,4X,"ZU" =",E12.5,4X,"MU" =",E12.5/,00000340
- 1X,"XW" =",F12.5,4X,"ZW" =",E12.5,4X,"MW" =",E12.5/, 00000350
- 1X,"XTH" =",F12.5,4X,"ZTH" =",F12.5,4X,"MTH" =",E12.5/, 00000360
- 1X,"XQ" =",F12.5,4X,"ZQ" =",E12.5,4X,"MQ" =",E12.5/, 00000370

```

-	1X,"XUD	=",E12.5,4X,"ZUD	=",E12.5,4X,"MUD	=",E12.5/,	00000380
-	1X,"XWD	=",E12.5,4X,"ZWD	=",E12.5,4X,"MWD	=",E12.5/,	00000390
-	1X,"XQD	=",E12.5,4X,"ZQD	=",E12.5,4X,"MQD	=",E12.5/,	00000400
-	1X,"XX	=",E12.5,4X,"ZX	=",E12.5,4X,"MX	=",E12.5/,	00000410
-	1X,"XZ	=",E12.5,4X,"ZZ	=",E12.5,4X,"MZ	=",E12.5/,	00000420
-	1X,"XDELT	=",E12.5,4X,"ZDELT	=",E12.5,4X,"MDELT	=",E12.5//	00000430
-	9X,"M	=",E12.5,8X,"IY	=",E12.5//,		00000440
-	9X,"UO	=",E12.5,8X,"LB	=",E12.5//)		00000450

351 CONTINUE

110 FORMAT(/,"J=",10X,"ECONV=",E12.5)

DIMENSION DS(8),XS(8),ZS(8),TS(8),ROOTR(8),ROOTI(8),ECONV(8)

CONTINUE

UO=U

N=6

NP1=N+1

ZQ=ZQ+M

DCST=LB

IF(ICABLE.EQ.0) DCST=UO

C=

C=

C=

NON-DIMENSIONAL D(S) COEFFICIENTS FOR TOWED VEHICLES

DS(7)=(M-XUD)*(M-ZWD)*(IY-MQD)-XWD*ZQD*MUD

-XQD*ZUD*MWD-(M-XUD)*ZQD*MWD

-XWD*ZUD*(IY-MQD)-XQD*(M-ZWD)*MUD

DS(6)=-XU*(M-ZWD)*(IY-MQD)-(M-XUD)*ZW*(IY-MQD)-(M-XUD)*(M-ZWD)*MQ

-XW*ZQD*MUD-XWD*ZQ*MUD-XWD*ZQD*MU

-XQ*ZUD*MWD-XQD*ZU*MWD-XQD*ZUD*MW

+XU*ZQD*MWD-(M-XUD)*ZQ*MWD-(M-XUD)*ZQD*MW

-XW*ZUD*(IY-MQD)-XWD*ZU*(IY-MQD)+XWD*ZUD*MQ

-XQ*(M-ZWD)*MUD+XQD*ZW*MUD-XQD*(M-ZWD)*MU

DS(5)=-XX*(M-ZWD)*(IY-MQD)+XU*ZW*(IY-MQD)-(M-XUD)*ZZ*(IY-MQD)

+XU*MQ*(M-ZWD)+(M-XUD)*ZW*MQ-(M-XUD)*(M-ZWD)*MTH

-XZ*ZQD*MUD-XW*ZQ*MUD-XWD*ZTH*MUD-XW*MU*ZQD-XWD*ZQ*MU-XWD*ZQD*MX

-XTH*ZUD*MWD-XQ*ZU*MWD-XQD*ZX*MWD-XQ*ZUD*MW-XQD*ZU*MW-XQD*ZUD*MZ

+XX*ZQD*MWD+XU*ZQ*MWD-(M-XUD)*ZTH*MWD+XU*ZQD*MW

-(M-XUD)*ZQ*MW-(M-XUD)*ZQD*MZ

-XZ*ZUD*(IY-MQD)-XW*ZU*(IY-MQD)-XWD*ZX*(IY-MQD)

+XW*ZUD*MQ+XWD*ZU*MQ+XWD*ZUD*MTH

```

- -XTH*(M-ZWD)*MU+XQ*ZW*MU+XQ*ZZ*MU-XQ*(M-ZWD)*MU      00000760
- +XQ*ZW*MU-XQ*(M-ZWD)*MX      00000770
DS(4)=XX*ZW*(IY-MQD)+XU*Z/(IY-MQD)+XX*(M-ZWD)*MQ-XU*ZW*MQ      00000780
- +(M-XUD)*ZZ*MQ+XU*(M-ZWD)*MTH+(M-XUD)*ZW*MTH      00000790
- -XZ*ZQ*MU-XW*ZTH*MU-XZ*ZQ*MU-XW*ZQ*MU      00000800
- -XWD*ZTH*MU-XW*ZQ*MU-XWD*ZQ*MU      00000810
- -XTH*ZU*MW-XQ*ZW*MW-XTH*ZUD*MW-XQ*ZU*MW      00000820
- -XQ*ZU*MW-XQ*ZUD*MZ-XQ*ZU*MZ      00000830
- +XX*ZQ*MW+XU*ZTH*MW+XX*ZQ*MW+XU*ZQ*MW=(M-XUD)*ZTH*MW      00000840
- +XU*ZQ*MZ=(M-XUD)*ZQ*MZ      00000850
- -XZ*ZU*(IY-MQD)-XW*ZX*(IY-MQD)+XZ*ZUD*MQ+XW*ZU*MQ      00000860
- +XWD*ZU*MQ+XW*ZUD*MTH+XWD*ZU*MTH      00000870
- +XTH*ZW*MU+XQ*ZZ*MU-XTH*(M-ZWD)*MU+XQ*ZW*MU      00000880
- +XQ*ZZ*MU-XQ*(M-ZWD)*MX+XQ*ZW*MU      00000890
DS(3)=XX*ZZ*(IY-MQD)-XX*ZW*MQ-XU*ZZ*MQ+XX*(M-ZWD)*MTH      00000900
- -XU*ZW*MTH+(M-XUD)*ZZ*MTH      00000910
- -XZ*ZTH*MU-XZ*ZQ*MU-XW*ZTH*MU-XZ*ZQ*MU      00000920
- -XW*ZQ*MU-XWD*ZTH*MU      00000930
- -XTH*ZX*MW-XTH*ZU*MW-XQ*ZW*MW-XTH*ZUD*MZ      00000940
- -XQ*ZU*MZ-XQ*ZU*MZ      00000950
- +XX*ZTH*MW+XX*ZQ*MW+XU*ZTH*MW+XX*ZQ*MZ      00000960
- +XU*ZQ*MZ=(M-XUD)*ZTH*MZ      00000970
- -XZ*ZX*(IY-MQD)+XZ*ZU*MQ+XW*ZU*MQ+XZ*ZUD*MTH      00000980
- +XW*ZU*MTH+XWD*ZU*MTH      00000990
- +XTH*ZZ*MU+XTH*ZW*MU+XQ*ZZ*MU-XTH*(M-ZWD)*MX      00001000
- +XQ*ZW*MU+XQ*ZZ*MU      00001010
DS(2)=-XX*ZZ*MQ-XX*ZW*MTH-XU*ZZ*MTH      00001020
- -XZ*ZTH*MU-XZ*ZQ*MU-XW*ZTH*MU      00001030
- -XTH*ZX*MW-XTH*ZU*MZ-XQ*ZU*MZ      00001040
- +XX*ZTH*MW+XX*ZQ*MZ+XU*ZTH*MZ      00001050
- +XZ*ZX*MQ+XZ*ZU*MTH+XW*ZU*MTH      00001060
- +XTH*ZZ*MU+XTH*ZW*MU+XQ*ZZ*MU      00001070
DS(1)=-XX*ZZ*MTH-XZ*ZTH*MU-XTH*ZX*MZ      00001080
- +XX*ZTH*MZ+XZ*ZX*MTH+XTH*ZZ*MU      00001090
WRITE (3,3)      00001100
3 FORMAT (//,20X,"***** DENOMINATOR DS(J) *****")      00001110
  DO 4 J=1,NP1      00001120
4 IF (DS(J).NE.0.0) GO TO 14      00001130

```

```

WRITE (3,5)                                00001140
5  FORMAT (/ ,16X,"***** DS(J) COEFFICIENTS ALL ZERO *****") 00001150
GO TO 20                                    00001160
8  FORMAT (/ ,12X,"J = ",I3,10X,"DS = ",E20.12)) 00001170
11 FORMAT (/ ,15X,"J = ",I3,10X,"FCONV = ",E15.8) 00001180
13 FORMAT (/ ,1X,"J = ",I3,9X,"ROOTR = ",E12.5,9X,"ROOTI = ",E12.5)) 00001190
C=                                           00001200
C= DIMENSIONALIZE THE COEFFICIENTS OF THE DENOMINATOR TRANSFER 00001210
C= FUNCTION                                00001220
C=                                           00001230
14 DO 15 I=1,NP1                            00001240
15 DS(I)=DS(I)*((LR/U)**(I-1))              00001250
IF (ITFCPRT.NE.1) GO TO 17                 00001260
WRITE (3,16)                               00001270
16 FORMAT (/ ,22X,"DIMENSIONAL COEFFICIENTS") 00001280
WRITE (3,8) ((J,DS(J)),J=1,NP1)           00001290
17 CALL PRNBM (N,DS,ROOTR,ROOTI,FCONV)     00001300
DO 18 J=1,N                                00001310
18 IF (ECONV(J).GT. .5E-09) WRITE(3,11) J,ECONV(J) 00001320
IF(IDIMRTS.EQ.0) GO TO 1                   00001330
WRITE (3,19)                               00001340
19 FORMAT (/ ,25X,"DIMENSIONAL ROOTS")     00001350
WRITE (3,13) ((J,ROOTR(J),ROOTI(J)),J=1,N) 00001360
20 WRITE (3,21)                             00001370
21 FORMAT (/)                               00001380
1  N=4                                       00001390
NP1=N+1                                    00001400
C=                                           00001410
C= NON-DIMENSIONAL X(S) COEFFICIENTS FOR TOWED VEHICLES 00001420
C=                                           00001430
XS(5)=(M-ZWD)*(IY-MQD)*XDFLT+XWD*ZQD*MDEL T+XQD*MWD*ZDEL T 00001440
- -ZQD*MWD*XDEL T+XWD*(IY-MQD)*ZDEL T+XQD*(M-ZWD)*MDEL T 00001450
XS(4)=-ZW*(IY-MQD)*XDEL T-(M-ZWD)*MQ*XDEL T 00001460
- +XW*ZQD*MDEL T+XWD*ZQ*MDEL T 00001470
- +XQ*MWD*ZDEL T+XQD*MW*ZDEL T 00001480
- -ZQ*MWD*XDEL T-ZQD*MW*XDEL T 00001490
- +XW*(IY-MQD)*ZDEL T-XWD*MQ*ZDEL T 00001500
- +XQ*(M-ZWD)*MDEL T-XQD*ZW*MDEL T 00001510

```


-	XS(3)=-Z*(IY-MQD)*XDEL T+ZW*MQ*XDEL T-(M-ZWD)*MTH*XDEL T	00001520
-	+XZ*7QD*MDEL T+XW*ZQ*MDEL T+XWD*ZTH*MDEL T	00001530
-	+XTH*MWD*ZDEL T+XQ*MW*ZDEL T+XQD*MZ*ZDEL T	00001540
-	-ZTH*MWD*XDEL T-7Q*MW*XDEL T-7QD*MZ*XDEL T	00001550
-	+XZ*(IY-MQD)*ZDEL T-XW*MQ*ZDEL T-XWD*MTH*ZDEL T	00001560
-	+XTH*(M-ZWD)*MDEL T-XQ*ZW*MDEL T-XQD*ZZ*MDEL T	00001570
-	XS(2)=ZZ*MQ*XDEL T+ZW*MTH*XDEL T+XZ*7Q*MDEL T+XW*ZTH*MDEL T	00001580
-	+XTH*MW*ZDEL T+XQ*MZ*ZDEL T-7TH*MW*XDEL T-ZQ*MZ*XDEL T	00001590
-	-XZ*MQ*ZDEL T-XW*MTH*ZDEL T-XTH*ZW*MDEL T-XQ*ZZ*MDEL T	00001600
-	XS(1)=ZZ*MTH*XDEL T+XZ*7TH*MDEL T+XTH*MZ*ZDEL T	00001610
-	-ZTH*MZ*XDEL T-XZ*MTH*ZDEL T-XTH*ZZ*MDEL T	00001620
	WRITE (3,2122)	00001630
	2122 FORMAT (//,23X,"***** X NUMERATOR *****")	00001640
	DO 25 J=1,NP1	00001650
	25 IF (XS(J).NE.0.0) GO TO 31	00001660
	WRITE (3,26)	00001670
	26 FORMAT (//,13X,"***** XS(J) COEFFICIENTS ALL ZERO *****")	00001680
	GO TO 35	00001690
	28 FORMAT (//,(12X,"J = ",I3,10X,"XS = ",E20.12))	00001700
C-		00001710
C-	DIMENSIONALIZE THE COEFFICIENTS OF THE X NUMERATOR TRANSFER	00001720
C-	FUNCTION	00001730
C-		00001740
	31 DO 32 I=1,NP1	00001750
	32 XS(I)=XS(I)+DCST*((LR/U)**(I-1))	00001760
	IF (ITFCPT.NE.1) GO TO 33	00001770
	WRITE (3,16)	00001780
	WRITE (3,28) ((J,XS(J)),J=1,NP1)	00001790
	33 CALL PRNRM (N,XS,ROOTR,ROOTI,ECONV)	00001800
	DO 34 I=1,N	00001810
	34 IF (ECONV(J).GT..5E-09) WRITE(3,11) J,ECONV(J)	00001820
	IF (IDIMRTS.FQ.0) GO TO 36	00001830
	WRITE (3,19)	00001840
	WRITE (3,13) ((J,ROOTR(J),ROOTI(J)),J=1,N)	00001850
	35 WRITE (3,21)	00001860
	36 N=4	00001870
	NP1=N+1	00001880
C-		00001890

C-	NON-DIMENSIONAL Z(S) COEFFICIENTS FOR TOWED VEHICLES	00001900
C-		00001910
-	$ZS(5) = (M-XUD) * (IY-MQD) * ZDFIT + 7QD * MUD * XDFLT + XQD * ZUD * MDFLT$	00001920
-	$+ (M-XUD) * ZQD * MDFLT + ZUD * (IY-MQD) * XDFLT - XQD * MUD * ZDELT$	00001930
-	$ZS(4) = -XU * (IY-MQD) * ZDELT - (M-XUD) * MQ * ZDELT$	00001940
-	$+ ZQ * MUD * XDFLT + ZQD * MU * XDFLT$	00001950
-	$+ XQ * ZUD * MDFLT + XQD * ZU * MDFLT - XU * ZQD * MDFLT + (M-XUD) * ZQ * MDFLT$	00001960
-	$+ ZU * (IY-MQD) * XDFLT - ZUD * MQ * XDFLT$	00001970
-	$- XQ * MUD * ZDELT - XQD * MU * ZDELT$	00001980
-	$ZS(3) = -XX * (IY-MQD) * ZDELT + XI * MQ * ZDELT - (M-XUD) * MTH * ZDELT$	00001990
-	$+ ZTH * MUD * XDFLT + 7Q * MU * XDFLT + 7QD * MX * XDFLT + XTH * 7UD * MDFLT$	00002000
-	$+ XQ * ZU * MDFLT + XQD * ZX * MDFLT$	00002010
-	$- XX * 7QD * MDFLT - XI * ZQ * MDFLT + (M-XUD) * ZTH * MDFLT$	00002020
-	$+ ZX * (IY-MQD) * XDFLT - ZU * MQ * XDFLT - ZUD * MTH * XDFLT$	00002030
-	$- XTH * MUD * ZDELT - XQ * MU * ZDELT - XQD * MX * 7DELT$	00002040
-	$ZS(2) = XX * MQ * ZDELT + XU * MTH * ZDELT + ZTH * MU * XDFLT + 7Q * MX * XDFLT$	00002050
-	$+ XTH * ZU * MDFLT + XQ * ZX * MDFLT - XX * 7Q * MDFLT - XU * ZTH * MDFLT$	00002060
-	$- ZX * MQ * XDFLT - ZU * MTH * XDFLT - XTH * MU * ZDELT - XQ * MX * ZDELT$	00002070
-	$ZS(1) = XX * MTH * ZDELT + ZTH * MX * XDFLT + XTH * ZX * MDFLT$	00002080
-	$- XX * ZTH * MDFLT - ZX * MTH * XDFLT - XTH * MX * 7DELT$	00002090
-	WRITE (3,350)	00002100
350	FORMAT (//,23X,"***** Z NUMFRATOR *****")	00002110
	DO 39 J=1,NP1	00002120
39	IF (ZS(J).NE.0.0) GO TO 45	00002130
	WRITE (3,40)	00002140
40	FORMAT (/,12X,"***** ZS(J) COEFFICIENTS ALL ZERO *****")	00002150
	GO TO 49	00002160
42	FORMAT (//,(12X,"J = ",I3,10X,"ZS = ",E20.12))	00002170
C-		00002180
C-	DIMENSIONALIZE THE COEFFICIENTS OF THE Z NUMFRATOR TRANSFER	00002190
C-	FUNCTION	00002200
C-		00002210
45	DO 46 I=1,NP1	00002220
46	$ZS(I) = ZS(I) * DCST * ((LB/U) ** (I-1))$	00002230
	IF (ITFCPRT.NE.1) GO TO 47	00002240
	WRITE (3,16)	00002250
	WRITE (3,42) ((J,ZS(J)),J=1,NP1)	00002260
47	CALL PRNBM (N,ZS,ROOTR,ROOTI,FCONV)	00002270

	DO 48 J=1,N	00002280
48	IF (ECONV(J).GT..5E-09) WRITE(3,11) J,ECONV(J)	00002290
	IF(IDIMRIS.FQ.0) GO TO 50	00002300
	WRITE (3,19)	00002310
	WRITE (3,13) ((1,ROOTK(J),ROOTI(J)),J=1,N)	00002320
49	WRITE (3,21)	00002330
50	N=4	00002340
	NP1=N+1	00002350
C-		00002360
C-	NON-DIMENSIONAL T(S) COEFFICIENTS FOR TOWED VEHICLES	00002370
C-		00002380
	TS(5)=(M-XUD)*(M-ZWD)*MDEL T+XWD*MUD*ZDEL T+ZUD*MWD*XDEL T	00002390
-	+ (M-XUD)*MWD*ZDEL T-XWD*ZUD*MDEL T+(M-ZWD)*MUD*XDEL T	00002400
	TS(4)=-XU*(M-ZWD)*MDEL T-(M-XUD)*ZW*MDEL T	00002410
-	+XW*MUD*ZDEL T+XWD*MU*ZDEL T+ZU*MWD*XDEL T+ZUD*MW*XDEL T	00002420
-	-XU*MWD*ZDEL T+(M-XUD)*MW*ZDEL T-XW*ZUD*MDEL T-XWD*ZU*MDEL T	00002430
-	-ZW*MUD*XDEL T+(M-ZWD)*MU*XDEL T	00002440
	TS(3)=-XX*(M-ZWD)*MDEL T+XU*ZW*MDEL T-(M-XUD)*ZZ*MDEL T	00002450
-	+XZ*MUD*ZDEL T+XW*MU*ZDEL T+XWD*MX*ZDEL T	00002460
-	+ZX*MWD*XDEL T+ZU*MW*XDEL T+ZUD*MZ*XDEL T	00002470
-	-XX*MWD*ZDEL T-XU*MW*ZDEL T+(M-XUD)*MZ*ZDEL T	00002480
-	-XZ*ZUD*MDEL T-XW*ZU*MDEL T-XWD*ZX*MDEL T	00002490
-	-ZZ*MUD*XDEL T-ZW*MU*XDEL T+(M-ZWD)*MX*XDEL T	00002500
	TS(2)=XX*ZW*MDEL T+XU*ZZ*MDEL T+XZ*MU*ZDEL T+XW*MX*ZDEL T	00002510
-	+ZX*MW*XDEL T+ZU*MZ*XDEL T-XX*MW*ZDEL T-XU*MZ*ZDEL T	00002520
-	-XZ*ZU*MDEL T-XW*ZX*MDEL T-ZZ*MU*XDEL T-ZW*MX*XDEL T	00002530
	TS(1)=XX*ZZ*MDEL T+XZ*MX*ZDEL T+ZX*MZ*XDEL T	00002540
-	-XX*MZ*ZDEL T-XZ*ZX*MDEL T-ZZ*MX*XDEL T	00002550
	WRITE (3,52)	00002560
52	FORMAT (//,23X,"***** 1 NUMFRATOR *****")	00002570
	DO 53 J=1,NP1	00002580
53	IF (TS(J).NE.0.0) GO TO 59	00002590
	WRITE (3,54)	00002600
54	FORMAT (/,13X,"***** TS(J) COEFFICIENTS ALL ZERO *****")	00002610
	GO TO 63	00002620
56	FORMAT (//(12X,"J = ",I3,10X,"TS = ",E20.12))	00002630
C-		00002640
C-	DIMENSIONALIZE THE COEFFICIENTS OF THE T NUMFRATOR TRANSFER	00002650

C-	FUNCTION	00002660
C-		00002670
59	DO 60 I=1,NP1	00002680
60	TS(I)=TS(I)*((LR/U)**(I-1))	00002690
	IF (ITFCPKT.NE.1) GO TO 61	00002700
	WRITE (3,16)	00002710
	WRITE (3,56) ((J,TS(J)),J=1,NP1)	00002720
61	CALL PRNBM (N,TS,ROOTR,ROOTI,ECONV)	00002730
	DO 62 I=1,N	00002740
62	IF (ECONV(J).GT..5E-09) WRITE(3,11) J,ECONV(J)	00002750
	IF (IDIMRTS.EQ.0) GO TO 63	00002760
	WRITE (3,19)	00002770
	WRITE (3,13)((J,ROOTR(J),ROOTI(J)),J=1,N)	00002780
63	CONTINUE	00002790
C-		00002800
C-		00002810
C-		00002820
	OTDE1(6)=TS(5)	00002830
	OTDE1(5)=TS(4)	00002840
	OTDE1(4)=TS(3)	00002850
	OTDE1(3)=TS(2)	00002860
	OTDE1(2)=TS(1)	00002870
	OTDE1(1)=0.0	00002880
	N=5;N10=N	00002890
	CALL PRNBM(N,OTDE1,OTDE1R,OTDF1I,ECONV)	00002900
	WRITE(3,3201)(OTDE1R(J10),OTDF1I(J10),J10=1,N)	00002910
3201	FORMAT(/,"ZEROS OF TD/E1",/,4(F9.4,F9.4))	00002920
	N=4;N12=N	00002930
	DO 2000 I1=1,5	00002940
	OTE2(I1)=TS(I1)	00002950
2000	CONTINUE	00002960
	CALL PRNBM(N,OTE2,OTE2R,OTF2I,ECONV)	00002970
	WRITE(3,3202)(OTE2R(J12),OTF2I(J12),J12=1,N)	00002980
3202	FORMAT(/,"ZEROS OF T/E2",/,4(F9.4,F9.4))	00002990
	OZDE3(6)=ZS(5)	00003000
	OZDE3(5)=ZS(4)-TS(5)*U0	00003010
	OZDE3(4)=ZS(3)-TS(4)*U0	00003020
	OZDE3(3)=ZS(2)-TS(3)*U0	00003030

	OZDE3(2)=ZS(1)*TS(2)*U0	00003040
	OZDF3(1)=-TS(1)*U0	00003050
	N=5;N14=N	00003060
	CALL PRNRM(N,OZDE3,OZDE3H,OZDF3I,ECONV)	00003070
	WRITE(3,3203)(OZDE3H(J14),OZDF3I(J14),J14=1,N)	00003080
3203	FORMAT(/,"ZEROS OF ZO/E3 AND Z/ZU",/4(F9.4,F9.4))	00003090
2997	WRITE(3,2998)	00003100
2998	FORMAT("ENTER KTDORG,DELKTD,KTDFIN",/)	00003110
	READ(3,/)KTDORG,DELKTD,KTDFIN	00003120
	IF(DELKTD.EQ.0) GO TO 99	00003130
	NGAIN=(KTDFIN-KTDORG)/DELKTD+1	00003140
	KTD=KTDORG	00003150
	DO 3001 JD=1,NGAIN	00003160
	RTDE1(7)=US(7)	00003170
	RTDE1(6)=DS(6)-KTD*TS(5)	00003180
	RTDE1(5)=DS(5)-KTD*TS(4)	00003190
	RTDE1(4)=DS(4)-KTD*TS(3)	00003200
	RTDE1(3)=DS(3)-KTD*TS(2)	00003210
	RTDE1(2)=DS(2)-KTD*TS(1)	00003220
	RTDE1(1)=DS(1)	00003230
	N=6	00003240
	CALL PRNRM(N,RTDE1,ROOTR,ROOTI,ECONV)	00003250
	DO 3006 J=1,N	00003260
	IF(ECONV(J).GT..5E-09) WRITE(3,110) J,ECONV(J)	00003270
3006	CONTINUE	00003280
	WRITE(3,3002)KTD	00003290
3002	FORMAT(F8.2)	00003300
	WRITE(3,3003) (ROOTR(J),ROOTI(J),J=1,N)	00003310
3003	FORMAT(4(F9.4,F0.4))	00003320
3001	KTD=KTD+DELKTD	00003330
	WRITE(3,3004)	00003340
3004	FORMAT("DO YOU WANT TO CONTINUE KTD R. LOCUS")	00003350
	READ(3,/)NVAL	00003360
	IF(NVAL.EQ.1) GO TO 2997	00003370
3016	WRITE(3,3005)	00003380
3005	FORMAT("ENTER KTD")	00003390
	READ(3,/)KTD	00003400
3007	WRITE(3,3008)	00003410

3008	FORMAT("ENTER KTORG,DELKT,KTFIN")	00003420
	READ(3,/)KTORG,DELKT,KTFIN	00003430
	IF(DELKT.EQ.0) GO TO 99	00003440
	NGAIN=(KTFIN-KTORG)/DELKT+1	00003450
	KT=KTORG	00003460
	DO 3012 JD=1,NGAIN	00003470
	RTE2(7)=DS(7)	00003480
	RTE2(6)=DS(6)-KTD*TS(5)	00003490
	RTE2(5)=DS(5)-KTD*TS(4)+KT*TS(5)	00003500
	RTE2(4)=DS(4)-KTD*TS(3)+KT*TS(4)	00003510
	RTE2(3)=DS(3)-KTD*TS(2)+KT*TS(3)	00003520
	RTE2(2)=DS(2)-KTD*TS(1)+KT*TS(2)	00003530
	RTE2(1)=DS(1)+KT*TS(1)	00003540
	N=6	00003550
	CALL PRNBM(N,RTE2,ROOTR,ROOTI,ECONV)	00003560
	DO 3009 J=1,N	00003570
	IF(ECONV(J).GT..5E-09) WRITE(3,110) J,ECONV(J)	00003580
3009	CONTINUE	00003590
	WRITE(3,3010)KT	00003600
3010	FORMAT(F8.2)	00003610
	WRITE(3,3011)((ROOTR(J),ROOTI(J),J=1,N))	00003620
3011	FORMAT(4(F9.4,F9.4))	00003630
3012	KT=KT+DELKT	00003640
		00003650
	WRITE(3,3013)KTD	00003660
3013	FORMAT("DO YOU WANT NEW KTD,0=NO,1=ENT,2=ENT&COMP,KTD NOW=",F8.2)	00003670
	READ(3,/)NVAL	00003680
	IF(NVAL.EQ.1)GO TO 3016; IF(NVAL.EQ.2) GO TO 2997	00003690
	WRITE(3,3014)	00003700
3014	FORMAT("DO YOU WANT TO CONTINUE KT R. LOCUS")	00003710
	READ(3,/)NVAL	00003720
	IF(NVAL.EQ.1) GO TO 3007	00003730
3017	WRITE(3,3015)	00003740
3015	FORMAT("ENTER KT")	00003750
	READ(3,/)KT	00003760
3018	WRITE(3,3019)	00003770
3019	FORMAT("ENTER KZDORG,DELKZD,KZDFIN")	00003780
	READ(3,/)KZDORG,DELKZD,KZDFIN	00003790

```

IF (DELK7D.EQ.0) GO TO 99
NGAIN=(KZDFIN-K7DORG)/DELK7D+1
KZD=KZDORG
DO 3022 JU=1,NGAIN
RZDE3(7)=DS(7)
RZDE3(6)=DS(6)-KTD*TS(5)+(K7D/RADIAN)*ZS(5)
RZDF3(5)=DS(5)-KTD*TS(4)+KT*TS(5)+(KZD/RADIAN)*(ZS(4)-TS(5)*UU)
RZDE3(4)=DS(4)-KTD*TS(3)+KT*TS(4)+(K7D/RADIAN)*(ZS(3)-TS(4)*UU)
RZDF3(3)=DS(3)-KTD*TS(2)+KT*TS(3)+(KZD/RADIAN)*(ZS(2)-TS(3)*UU)
RZDE3(2)=DS(2)-KTD*TS(1)+KT*TS(2)+(KZD/RADIAN)*(ZS(1)-TS(2)*UU)
RZDE3(1)=DS(1)+KT*TS(1)+(K7D/RADIAN)*(-TS(1)*UU)
N=6
CALL PRNRM(N,RZDF3,KOUTR,ROUTT,ECONV)
DO 3020 J=1,N
IF (ECONV(J).GT..5E-09) WRITE (3,110) J,ECONV(J)
3020 CONTINUE ;WRITE(3,110)K7D
WRITE (3,3021)((ROUTR(J),ROUTT(J),J=1,N))
3021 FORMAT (4(F9.4,F9.4))
3022 KZD=KZD+DELK7D
WRITE(3,3099)KTD
3099 FORMAT("DO YOU WANT NEW KTD. KTD NOW =",F8.2)
READ(3,/)NVAL
IF (NVAL.EQ.1) GO TO 3016; IF (NVAL.EQ.2) GO TO 2997
WRITE(3,3023)KT
3023 FORMAT("DO YOU WANT NEW KT. KT NOW =",F8.2)
READ(3,/)NVAL
IF (NVAL.EQ.1) GO TO 3017; IF (NVAL.EQ.2) GO TO 3007
WRITE(3,3103)
3103 FORMAT("DO YOU WANT TO CONTINUE KZD R. LOCUS")
READ(3,/)NVAL
IF (NVAL.EQ.1) GO TO 3018
3024 WRITE(3,3025)
3025 FORMAT("ENTER K7D")
READ(3,/)KZD
3026 WRITE(3,3098)
3098 FORMAT("ENTER KZORG,DELK7,KZFIN")
READ(3,/)KZORG,DELK7,KZFIN
IF (DELK7.EQ.0) GO TO 99

```

```

00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
00004000
00004010
00004020
00004030
00004040
00004050
00004060
00004070
00004080
00004090
00004100
00004110
00004120
00004130
00004140
00004150
00004160
00004170

```


	NGAIN=(KZFIN-KZORG)/DELKZ+1	00004180
	KZ=KZORG	00004190
	DO 3029 JD=1,NGAIN	00004200
	RZZO(8)=DS(7)	00004210
	RZZO(7)=DS(6)-KTD*TS(5)+KZN*ZS(5)	00004220
	RZZO(6)=DS(5)-KTD*TS(4)+KT*TS(5)+KZD*(ZS(4)-TS(5)*UO)-K7*ZS(5)	00004230
	RZZO(5)=DS(4)-KTD*TS(3)+KT*TS(4)+KZD*(ZS(3)-TS(4)*UO)-K7*(ZS(4)	00004240
-	-TS(5)*UO)	00004250
	RZZO(4)=DS(3)+KT*TS(3)+KZD*(ZS(2)-TS(3)*UO)-KZ*(ZS(3)-TS(4)*UO)	00004260
-	-KTD*TS(2)	00004270
	RZZO(3)=DS(2)+KT*TS(2)+KZD*(ZS(1)-TS(2)*UO)-KZ*(ZS(2)-TS(3)*UO)	00004280
-	-KTD*TS(1)	00004290
	RZZO(2)=DS(1)+KT*TS(1)-KZD*TS(1)*UO-KZ*(ZS(1)-TS(2)*UO)	00004300
	RZZO(1)=-KZ*(-TS(1)*UO)	00004310
	N=7	00004320
	CALL PRNBM(N,RZ70,ROOTR,ROOTI,ECONV)	00004330
	DO 3027 J=1,N	00004340
	IF (ECONV(J) .GT. .5E-09) WRITE (3,110) J,ECONV(J)	00004350
3027	CONTINUE ,WRITE(3,3110)KZ	00004360
	GAIN=KZ*ZS(5)/DS(5)	00004370
	WRITE(3,3250)GAIN	00004380
3250	FORMAT("GAIN =",F8.4)	00004390
	WRITE (3,3028)((ROOTR(J),ROOTI(J),J=1,N))	00004400
3028	FORMAT (4(F9.4,F9.4))	00004410
3029	KZ=KZ+DELKZ	00004420
	WRITE(3,3099)KTD	00004430
	READ(3,/)NVAL	00004440
	IF (NVAL.EQ.1) GO TO 3016;IF (NVAL.EQ.2.) GO TO 2997	00004450
	WRITE(3,3023)KT	00004460
	READ(3,/)NVAL	00004470
	IF (NVAL.EQ.1) GO TO 3017;IF (NVAL .EQ.2) GO TO 3007	00004480
	WRITE(3,3032)KZN	00004490
3032	FORMAT("DO YOU WANT NEW K7D,K7D NOW =",F8.4)	00004500
	READ(3,/)NVAL	00004510
	IF (NVAL .EQ.1) GO TO 3024; IF(NVAL.EQ.2)GO TO 3018	00004520
	WRITE(3,3033)	00004530
3033	FORMAT("DO YOU WANT TO CONTINUE KZ R. LOCUS")	00004540
	READ(3,/)NVAL	00004550

```

IF (NVAL.EQ.1) GO TO 3026

WRITE(3,9000)
9000 FORMAT("DO YOU WANT A TIMEPLT")
READ(3,/)NVAL
IF(NVAL.EQ.0) GO TO 99

C-
WRITE(4,4000) (TITLE(I),I=1,11)

C-
C- STANDARD TEST VALUES USED FOR THESE TIMEPLTS
IF=1
TDEL=0.1
TMAX=20.
RTIME=0.
PTIME=10.
W=0
IPLOT=1
IWRITE=0

WRITE(4,9003)IF,TDEL,TMAX,RTIME,PTIME,W,IPLOT,IWRITE
9003 FORMAT(I1,1H,,F5.3,1H,,F8.3,1H,,F10.5,1H,,F10.5,1H,,
- F10.5,1H,,I5,1H,,I5,2H,*)
88 WRITE(3,9001)
9001 FORMAT("ENTER KTD,KT,KZD,KZ,GAMP")
READ(3,/)KTD,KT,KZD,KZ,GAMP
WRITE(3,9002)
9002 FORMAT("TIMEPLT FOR WHICH LOOP?",
- ," ENTER 1=1ST, 2=2ND, 3=3RD, 4=4TH")
READ(3,/)NVAL
IF(NVAL.EQ.2) GO TO 9100
IF(NVAL.EQ.3) GO TO 9101
IF(NVAL.EQ.4) GO TO 9102
GAIN=GAMP*(TS(5)/DS(7))
WRITE(4,9004)N10,GAIN
9004 FORMAT(I1,1H,,F15.5,2H,*)
WRITE(4,9005)((OTDEF1(I(I)),OTDEF1(I(I))),I=1,N10)
9005 FORMAT(4(F12.5,1H,),1H/)

```

```

00004560
00004570
00004580
00004590
00004600
00004610
00004620
00004630
00004640
00004650
00004660
00004670
00004680
00004690
00004700
00004710
00004720
00004730
00004740
00004750
00004760
00004770
00004780
00004790
00004800
00004810
00004820
00004830
00004840
00004850
00004860
00004870
00004880
00004890
00004900
00004910
00004920
00004930

```


	GO TO 9103	00004940
9100	GAIN=GAMP*(TS(5)/DS(7))	00004950
	WRITE(4,9006)N12,GAIN	00004960
9006	FORMAT(I1,1H,,F15.5,2H,*)	00004970
	WRITE(4,9007)((NTE2R(I2),NTE2I(I2)),I2=1,N12)	00004980
9007	FORMAT(4(F12.5,1H,),1H/)	00004990
	GO TO 9103	00005000
9101	GAIN=GAMP*(ZS(5)/DS(7))	00005010
	WRITE(4,9008)N14,GAIN	00005020
9008	FORMAT(I1,1H,,F15.5,2H,*)	00005030
	WRITE(4,9009)((NZDE3R(I3),NZDF3I(I3)),I3=1,N14)	00005040
9009	FORMAT(4(F12.5,1H,),1H/)	00005050
9103	RZDE3(7)=DS(7)	00005060
	RZDE3(6)=DS(6)-KTD*TS(5)+K7D*7S(5)	00005070
	RZDE3(5)=DS(5)-KTD*TS(4)+KT*TS(5)+KZD*(ZS(4)-TS(5)*UO)	00005080
	RZDE3(4)=DS(4)-KTD*TS(3)+KT*TS(4)+KZD*(ZS(3)-TS(4)*UO)	00005090
	RZDE3(3)=DS(3)+KT*TS(3)+KZD*(7S(2)-TS(3)*UO)-KTD*TS(2)	00005100
	RZDE3(2)=DS(2)+KT*TS(2)+KZD*(7S(1)-TS(2)*UO)-KTD*TS(1)	00005110
	RZDE3(1)=DS(1)+KT*TS(1)-K7D*TS(1)*UO	00005120
	N=6	00005130
	CALL PRNBM(N,RZDE3,ROOTR,ROOTI,ECONV)	00005140
	DO 9010 J=1,N	00005150
	IF (ECONV(J) .GT. .5E-09) WRITE (3,110) J,ECONV(J)	00005160
9010	CONTINUE	00005170
	WRITE(4,9011)N	00005180
9011	FORMAT(I1,2H,*)	00005190
	WRITE(4,9012)((ROOTR(J6),ROOTI(J6)),J6=1,N)	00005200
9012	FORMAT(4(F12.5,1H,),1H/)	00005210
	GO TO 77	00005220
		00005230
9102	GAIN=GAMP*((-KZ+ZS(5))/DS(7))	00005240
	WRITE(4,9013)N14,GAIN	00005250
9013	FORMAT(I1,1H,,F15.5,2H,*)	00005260
	WRITE(4,9014)((NZDE3R(J7),NZDF3I(J7)),J7=1,N14)	00005270
9014	FORMAT(4(F12.4,1H,),1H/)	00005280
	RZZO(8)=DS(7)	00005290
	RZZO(7)=DS(6)-KTD*TS(5)+KZD*ZS(5)	00005300
	RZZO(6)=DS(5)-KTD*TS(4)+KT*TS(5)+K7D*(ZS(4)-TS(5)*UO)-K7*ZS(5)	00005310

EFIX=.5E-09
 CONV=1.E-35
 NC=N+1
 C-
 C- SEND COEFFICIENTS TO REDUCED COEFFICIENT STORAGE
 C-
 DO 1 I=1,NC
 ECONV(I)=0.0
 1 H(I)=A(I)
 C-
 C- INITIALIZE GUESSES AND SET REVERSAL INDICATOR NORMAL
 C-
 P=0.
 Q=0.0
 R=0.
 IREV=1
 C-
 C- SCALING TO BE DONE AT THIS POINT AND REMOVE ALL ZERO ROOTS
 C-
 2 IF(H(NC)) 4,3,4
 3 NC=NC-1
 V(NC)=0.0
 U(NC)=0.0
 GO TO 2
 4 IF(H(1)) 7,5,7
 5 NC=NC-1
 V(NC)=0.
 U(NC)=0.
 DO 6 I=1,NC
 6 H(I)=H(I+1)
 GO TO 4
 C-
 C- TEST FOR VARIOUS DEGREES
 C-
 7 IF(ICOUNT.LT.2) GO TO 8
 ECONV(ICOUNT-1)=E
 8 ICOUNT=ICOUNT+1
 9 IF(NC-1) 10,50,10

00005700
 00005710
 00005720
 00005730
 00005740
 00005750
 00005760
 00005770
 00005780
 00005790
 00005800
 00005810
 00005820
 00005830
 00005840
 00005850
 00005860
 00005870
 00005880
 00005890
 00005900
 00005910
 00005920
 00005930
 00005940
 00005950
 00005960
 00005970
 00005980
 00005990
 00006000
 00006010
 00006020
 00006030
 00006040
 00006050
 00006060
 00006070

10	IF(NC=2) 12,11,12	00006080
11	R=-H(1)/H(2)	00006090
	GO TO 37	00006100
12	IF(NC=3) 14,13,14	00006110
13	P=H(2)/H(3)	00006120
	Q=H(1)/H(3)	00006130
	GO TO 42	00006140
C-		00006150
C-	TEST TO REVERSE COEFFICIENTS AND DO SO IF TEST SUCCEEDS	00006160
C-		00006170
14	IF(ABS(H(NC-1)/H(NC))-ABS(H(2)/H(1))) 15,21,21	00006180
15	IREV=-IREV	00006190
	M=NC/2	00006200
	DO 16 I=1,M	00006210
	NL=NC+1-I	00006220
	F=H(NL)	00006230
	H(NL)=H(I)	00006240
16	H(I)=F	00006250
	IF(Q) 18,17,18	00006260
17	P=0.	00006270
	GO TO 19	00006280
18	P=P/Q	00006290
	Q=1./Q	00006300
19	IF(R) 20,21,20	00006310
20	R=1./R	00006320
C-		00006330
C-	NEWTON, CALCULATE F(R) AND TEST FOR ROOT	00006340
C-		00006350
21	E=EFIX	00006360
	R(NC)=H(NC)	00006370
	C(NC)=H(NC)	00006380
	R(NC+1)=0.	00006390
	C(NC+1)=0.	00006400
	NP=NC-1	00006410
22	DO 35 J=1,1000	00006420
	DO 23 I1=1,NP	00006430
	I=NC-I1	00006440
	B(I)=H(I)+R*B(I+1)	00006450

23	C(I)=B(I)+R*C(I+1)	00006460
	IF(ABS(B(I)/H(I))-E) 37,37,24	00006470
24	IF(C(2)) 26,25,26	00006480
25	R=R+1.	00006490
	GO TO 27	00006500
26	R=R-B(1)/C(2)	00006510
C=		00006520
C=	MAKE A BAIRSTOW REDUCTION AND CORRECT	00006530
C=		00006540
27	DO 28 I1=1,NP	00006550
	I=NC-I1	00006560
	B(I)=H(I)-P*B(I+1)-Q*B(I+2)	00006570
28	C(I)=B(I)-P*C(I+1)-Q*C(I+2)	00006580
C=		00006590
C=	TEST FOR CONVERGENCE OF BAIRSTOW PROCESS	00006600
C=		00006610
	IF(H(2)) 30,29,30	00006620
29	IF(ABS(B(2)/H(1))-E) 31,31,32	00006630
30	IF(ABS(B(2)/H(2))-E) 31,31,32	00006640
31	IF(ABS(B(1)/H(1))-E) 42,42,32	00006650
32	CBAR=C(2)-B(2)	00006660
	D=C(3)**2-CBAR*C(4)	00006670
	IF(D) 34,33,34	00006680
33	P=P-2.	00006690
	Q=Q*(Q+1.)	00006700
	GO TO 35	00006710
34	P=P+(B(2)*C(3)-R(1)*C(4))/D	00006720
	Q=Q+(-B(2)*CBAR+B(1)*C(3))/D	00006730
35	CONTINUE	00006740
	E=E*10.	00006750
	IF(E=CONV) 22,22,36	00006760
36	CONV=E	00006770
	GO TO 42	00006780
C=		00006790
C=	LINEAR. COMPUTE AND STORE LINEAR ROOTS	00006800
C=		00006810
37	NC=NC-1	00006820
	V(NC)=0.	00006830

	IF(IREV) 38,39,39	00006840
	38 U(NC)=1./R	00006850
	GO TO 40	00006860
	39 U(NC)=R	00006870
	40 DO 41 I=1,NC	00006880
	41 H(I)=H(I+1)	00006890
	GO TO 7	00006900
C-		00006910
C-	QUADRATIC. SOLVE QUADRATIC AND STORE ROOTS	00006920
C-		00006930
	42 NC=NC-2	00006940
	IF(IREV) 43,44,44	00006950
	43 QP=1./Q	00006960
	PP=P/(Q*2.0)	00006970
	GO TO 45	00006980
	44 QP=Q	00006990
	PP=P/2.0	00007000
	45 F=(PP)**2-QP	00007010
	IF(F) 46,47,47	00007020
C-		00007030
C-	CASE OF IMAGINARY ROOTS.	00007040
C-		00007050
	46 U(NC+1)=-PP	00007060
	U(NC)=-PP	00007070
	V(NC+1)=SQRT(-F)	00007080
	V(NC)=-V(NC+1)	00007090
	GO TO 48	00007100
C-		00007110
C-	CASE OF REAL ROOTS	00007120
C-		00007130
	47 U(NC+1)=-SIGN(ARS(PP)+SQRT(F),PP)	00007140
	V(NC+1)=0.	00007150
	U(NC)=QP/U(NC+1)	00007160
	V(NC)=0.	00007170
C-		00007180
C-	FORM NEW REDUCED COEFFICIENTS	00007190
C-		00007200
	48 DO 49 I=1,NC	00007210

49 H(I)=B(I+2)
GO TO 7
50 RETURN
END

00007220
00007230
00007240
00007250

APPENDIX D

TIME HISTORY PLOT DIAGRAM CARD LIST

(Reverse Page D-2 Blank)

SCARD	FREFFORM	00000000
FILE	1=TIMEPLT/DA,UNIT=DISK,SAVE=30,LOCK,BLOCKING=3,RECORD=10	00000010
FILE	2=D/H,UNIT=RMUTE,LOCK,RECORD=9	00000020
FILE	3=PRIASP,UNIT=PRINTER,RECORD=15	00000030
FILE	4=FORP/DSK,UNIT=DISK,SAVE=30,LOCK,AREA=5000,RECORD=30	00000040
FILE	5=APFR/DA,UNIT=DISK,SAVE=30,LOCK,BLOCKING=3,RECORD=10	00000050
	REAL NG	00000060
	DIMENSION XLABEL(2),YLABEL(3),CONTRL(8),TITLE(11)	00000070
	DIMENSION NG(14),DG(14),ROOTR(14),ROOTI(14),OUTPUT(1002),I(1002),	00000080
-	FORCE(6,2),NC(5),NT(15),ZEROR(10),ZEROI(10)	00000090
	COMMON TDEL,TMAX	00000100
	DATA FORCE/"IMPI", "STEP", "RAMP", "PULS", "RAMP", "SIN", "LcF",	00000110
-	" ", " ", "E", "STEP", "SOIN"/	00000120
	DATA(XLABEL(1),I=1,2)/12H TIME=(SEC) /	00000130
	DATA(YLABEL(1),I=1,3)/18H OUTPUT=RESPONSE /	00000140
C=	****TIMEPLT/HUMP** SOURCE IS TIMEPLT/SORSE DR. D.F. HUMPHREYS	00000150
	ITF=0 ; KK=-1	00000160
C=	99 FORMAT(/,"***** NOW, RUN THE PROGRAM PLOT/HUMP*****")	00000170
D-3	100 FORMAT(8F10.3)	00000180
	105 FORMAT(4F20.4)	00000190
	106 FORMAT(5F15.3)	00000200
	READ(1,4000) (TITLE(I),I=1,11)	00000210
4000	FORMAT(11A6)	00000220
	READ(1,/) IF,TDEL,TMAX,RTIME,PTIME,W,IPLT,IWRITE	00000230
1	READ(1,/,END=3) INGM1,GAIN	00000240
	IF(INGM1.GT.0) GO TO 2	00000250
	ZEROR(1)=-1. ; GO TO 215	00000260
2	READ(1,/)(ZEROR(I),ZEROI(I),I=1,INGM1)	00000270
215	READ(1,/) IDGM1	00000280
	READ(1,/)(ROOTR(I),ROOTI(I),I=1,IDGM1)	00000290
	IF(INGM1.GT.1) GO TO 212	00000300
	NG(1)=-ZEROR(1)	00000310
	NG(2)=INGM1*1.	00000320
	ING=INGM1 + 1	00000330
	GO TO 4	00000340
212	CALL CPMPY(ZEROR,ZEROI,INGM1,NG,ING)	00000350
4	CONTINUE	00000360
C=		

C=		00000370
C=		00000380
	KK=KK+1	00000390
	ITF=ITF + 1	00000400
107	FORMAT(1H1)	00000410
	AMP=GAIN	00000420
C=		00000430
C=	IF = FORCING FUNCTION INDICATOR	00000440
C=	IF=0 IMPLIES AN IMPULSE	00000450
C=	IF=1 IMPLIES A STEP	00000460
C=	IF=2 IMPLIES A RAMP	00000470
C=	IF=3 IMPLIES A PULSE	00000480
C=	IF=4 IMPLIES A RAMPSTEP	00000490
C=	IF=5 IMPLIES A SINUSOID	00000500
	CALL TIME(IF,AMP,PTIME,RTIME,W,OUTPUT,T,IO,NG,ING,ROOTR,	00000510
	ROOTI,IDGM1,ICODE)	00000520
	WRITE(3,208)ITF	00000530
208	FORMAT(/,1X,"*****TIME RESPONSE RUN NUMBER",I3,"*****")	00000540
	WRITE(3,98)	00000550
D-4 98	FORMAT(/ ,10X,"THE COEFFICIENTS OF THE NUMFRATOR"/)	00000560
	DO 70 I=1,ING	00000570
70	WRITE(3,101)I,NG(I)	00000580
101	FORMAT(10X,"NG(",I1,")=",F17.7)	00000590
	IF(INGM1.EQ.0)GO TO 216 ; WRITE(3,210)	00000600
210	FORMAT(///10X,"THE ZEROS OF THE NUMERATOR"/)	00000610
	DO 82 I=1,INGM1	00000620
82	WRITE(3,209)I,ZFROR(I),ZERON(I)	00000630
209	FORMAT(10X,"ZERON(",I1,")=",F17.7," + J ",F17.7)	00000640
216	WRITE(3,102)	00000650
102	FORMAT(///10X,"THE ROOTS OF THE DENOMINATOR"/)	00000660
	DO 80 I=1,IDGM1	00000670
80	WRITE(3,103)I,ROOTR(I),ROOTI(I)	00000680
103	FORMAT(10X,"ROOT(",I1,")=",F17.7," + J ",F17.7)	00000690
	WRITE(3,104) IF,(FORCE(IF+1,I),I=1,2),AMP	00000700
104	FORMAT(///1X,"THE FORCING FUNCTION INDICATOR (IF) =",I3,/,	00000710
	" THIS IMPLIES THAT A ",2A4," INPUT WAS USED.",/,	00000720
	" AMPLITUDE=",F17.7,///)	00000730
	IF(IF.EQ.3) WRITE(3,205) PTIME	00000740

205	FORMAT(//20X,"PTIME=",F17.7)	00000750
	IF(IF.EQ.4) WRITE(3,204) RTIME	00000760
204	FORMAT(//20X,"RTIME WAS ADJUSTED TO ",F10.4," SECONDS.")	00000770
	IF(IF.EQ.5) WRITE(3,206)W	00000780
206	FORMAT(//20X,"FREQUENCY=",F17.7)	00000790
	IF (ICODE.EQ.0) GO TO 302;WRITE(3,96)ICODE	00000800
96	FORMAT(//," ICODE =",I10)	00000810
	IF(ICODE.EQ.1) WRITE(3,201)	00000820
201	FORMAT(//10X,"THE COMPLEX PART OF THE OUTPUT VECTOR BECAME",	00000830
	"SIGNIFICANT")	00000840
	IF(ICODE.EQ.2)WRITE(3,202)	00000850
202	FORMAT(//10X,"MULTIPLE ROOTS ENCOUNTERED")	00000860
	IF(ICODE.EQ.3) WRITE(3,203)	00000870
203	FORMAT(//10X,"BAD ENTRY - CHECK POLYNOMIAL ORDERS OF T.F.")	00000880
	IF(ICODE.NE.0) GO TO 1	00000890
302	XMAX=T(1)	00000900
	XMIN=T(1)	00000910
	YMAX=OUTPUT(1)	00000920
	YMIN=OUTPUT(1)	00000930
	ELENUM=TMAX/TDEF	00000940
	IIH=ELENUM + 1.0	00000950
	DO 600 KI=1,IIH	00000960
	IF(T(KI).LT.XMIN)XMIN=T(KI)	00000970
	IF(T(KI).GT.XMAX)XMAX=T(KI)	00000980
	IF(OUTPUT(KI).GT.YMAX)YMAX=OUTPUT(KI)	00000990
	IF(OUTPUT(KI).LT.YMIN)YMIN=OUTPUT(KI)	00001000
600	CONTINUE	00001010
	WRITE(3,97)	00001020
97	FORMAT(1H1,////,8X,"TIME",13X,"OUTPUT",//)	00001030
	DO 10 I=1,10,2	00001040
10	WRITE(3,200)T(I),OUTPUT(I)	00001050
200	FORMAT(2F17.7)	00001060
	WRITE(3,107)	00001070
	IF(1WRITE.EQ.0)GO TO 11	00001080
	ITER=1	00001090
	WRITE(5,213)	00001100
	WRITE(5,214)ITER,TMAX,10GM1,IIH	00001110
213	FORMAT(30X)	00001120

```

214 FORMAT(15,F10.1,2I5)
C-
12 WRITE(5,207)(OUTPUT(I),I=1,10)
207 FORMAT(5(E12.5,1X))
11 CONTINUE
IF(IPLT.EQ.0)GO TO 1
DX=(XMAX-XMIN)/7. ; DY=(YMAX-YMIN)/5.
IF(ABS(DY).GT..00001)GOTO334 ; WRITE(3,301) ; GOTO1
334 IF(KK.GT.0)GO TO 700
CALL PLOT(0,-12,-3) ; CALL PLOT(1,1,-3) ; GO TO 800
700 CALL PLOT(12,0,-3)
800 CONTINUE
301 FORMAT(///," *****NO PLOT FOR LAST OUTPUT***** ")
A=0. ; IF(YMIN.I.T.O..AND.YMIN+DY*5.GT.O.)A=-YMIN/DY
CALL AXIS(0,A,XI ABEL,-11,7,0,XMIN,DX)
CALL AXIS(0,0,YI ABEL,15,5,0,YMIN,DY)
CALL LINE(T,OUTPUT,I1H,1,0,3,XMIN,DX,YMIN,DY)
GO TO 1
3 CONTINUE ; IF(IPLT.EQ.0)GO TO 217
CALL PLOT(0,0,999)
LOCK 4
C- WRITE(2,107) ;C= WRITE(5,99)
217 STOP
END
SUBROUTINE CPVAL(RES,ARG,X,IDIMX)
COMPLEX RES,ARG
DIMENSION X(20)
RES=(0.,0.)
J=IDIMX
1 IF(J)3,3,2
2 RES=RES+ARG+X(J)
J=J-1
GO TO 1
3 RETURN
END
C-*****
C-
C- PURPOSE"

```

```

00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500

```

C-		00001510
C-	THE PURPOSE OF THIS SUBROUTINE IS TO DETERMINE THE TIME	00001520
C-	RESPONSE OF AN INPUT TO A TRANSFER FUNCTION BY TAKING THE	00001530
C-	INVERSE LAPLACE TRANSFORM BY THE METHOD OF RESIDUES.	00001540
C-	INTRODUCTION TO AUTOMATIC CONTROL SYSTEMS - ROBERT N. CLARK	00001550
C-	(PP 70 - 77).	00001560
C-		00001570
C-		00001580
C-	VARIABLES"	00001590
C-	AMP = AMPLITUDE OF THE FORCING FUNCTION	00001600
C-	PTIME = PULSE TIME (SPECIFIED FOR IF=3)	00001610
C-	RTIME = RAMP TIME (SPECIFIED FOR IF=4)	00001620
C-	W = FREQUENCY OF THE SINUSOIDAL INPUT (SPECIFIED FOR IF=5)	00001630
C-	OUTPUT = VECTOR OF CALCULATED RESPONSE AMPLITUDE VALUES	00001640
C-	T = VECTOR OF SEQUENTIAL TIME VALUES DIRECTLY RELATED TO OUTPUT	00001650
C-	IO = NUMBER OF OUTPUT VALUES (CALCULATED)	00001660
C-	NG = VECTOR OF NUMERATOR COEFFICIENTS	00001670
C-	ING = DIMENSION OF THE NUMERATOR COEFFICIENTS	00001680
C-	ROOTR = VECTOR OF REAL PARTS OF THE ROOTS	00001690
C-	ROOTI = VECTOR OF IMAGINARY PARTS OF THE ROOTS	00001700
C-	IOGM1 = NUMBER OF ROOTS (ORDER OF THE DENOMINATOR)	00001710
C-	ICODE = RETURN CODE VARIABLE	00001720
C-	ICODE=0 IMPLIES NORMAL EXECUTION	00001730
C-	ICODE=1 IMPLIES THAT THE COMPLEX PART OF THE OUTPUT	00001740
C-	VECTOR BECAME SIGNIFICANT.	00001750
C-		00001760
C-		00001770
C-	ICODE=3 IMPLIES THAT THE ORDER OF THE DENOMINATOR WAS	00001780
C-	NOT GREATER THAN THE ORDER OF THE NUMERATOR.	00001790
C-	ICODE=4 IMPLIES THAT THE FORCING FUNCTION INDICATOR	00001800
C-	WAS SPECIFIED INCORRECTLY	00001810
C-		00001820
C-		00001830
C-	SUBROUTINES CALLED"	00001840
C-		00001850
C-	CPVAL = COMPLEX EVALUATION OF A POLYNOMIAL	00001860
C-	TIME = RESPONSE BY THE METHOD OF RESIDUES	00001870
C-		00001880

C=		00001890
C=	REMARKS"	00001900
C=		00001910
C=	THIS SUBROUTINE IS DESIGNED TO GENERATE THE TIME RESPONSE OF A	00001920
C=	GENERAL OUTPUT FUNCTION $XO(S) = X1(S)G(S)$. IN THIS	00001930
C=	EVALUATION TWO IMPORTANT ASSUMPTIONS ARE MADE.	00001940
C=	1) THE ORDER OF THE DENOMINATOR OF THE OUTPUT FUNCTION	00001950
C=	OF THE OUTPUT FUNCTION.	00001960
C=	MUST BE LARGER THAN THE ORDER OF THE NUMERATOR	00001970
C=	2) MULTIPLE ROOTS OF THE DENOMINATOR MAY NOT EXIST.	00001980
C=		00001990
C=	*****	00002000
C=		00002010
-	SUBROUTINE TIME(IF,AMP,PTIME,RTIME,W,OUTPUT,T,IO,NG,ING,ROOTR,	00002020
	ROOTI,IDGM1,ICODE)	00002030
	REAL NG	00002040
	DIMENSION P(14),NG(14),K(14),OUT(1002),T(1002),OUTPUT(1002),	00002050
	ROOTR(14),ROOTI(14),SAVE(1002)	00002060
	COMMON TDEL,TMAX	00002070
	ICODE=4	00002080
	IF(IF.GT.5.OR.IF.LT.0)RETURN	00002090
	GAINDG=1./AMP	00002100
C=		00002110
C=	DETERMINE TMAX	00002120
C=		00002130
C=	SMALL=1.E6	00002140
C=	DO 9 I=1,IDGM1	00002150
C=	ABSR=ABS(ROOTR(I))	00002160
C=	IF (ABSR.EQ.0.) GO TO 10	00002170
C=	IF(ABSR.LT.SMALL) SMALL=ABSR	00002180
C=		00002190
	IF(IF.EQ.0)GO TO 11	00002200
	GO TO (10,20,30,40,50) ,IF	00002210
11	CALL TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)	00002220
	RETURN	00002230
10	IDGM1=IDGM1+1	00002240
	ROOTR(IDGM1)=0.	00002250
	ROOTI(IDGM1)=0.	00002260

CALL TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)	00002270
RETURN	00002280
20 IDGM1=IDGM1+1	00002290
ROOTR(IDGM1)=.001	00002300
ROOTI(IDGM1)=0.	00002310
IDGM1=IDGM1+1	00002320
ROOTR(IDGM1)=-.001	00002330
ROOTI(IDGM1)=0.	00002340
CALL TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)	00002350
RETURN	00002360
30 IDGM1=IDGM1+1	00002370
ROOTR(IDGM1)=0.0	00002380
ROOTI(IDGM1)=0.0	00002390
CALL TYME(SAVE,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)	00002400
IF(ICODE.NE.0) GO TO 34	00002410
MTIME=PTIME/TDEL+0.5	00002420
IF(MTIME.EQ.0) GO TO 32	00002430
DO 31 I=1,MTIME	00002440
31 OUTPUT(I)=SAVE(T)	00002450
32 TP1=MTIME+1	00002460
DO 33 I=TP1,IO	00002470
33 OUTPUT(I)=SAVE(T)-SAVE(I-MTIME)	00002480
34 RETURN	00002490
40 I=RTIME/TDEL+.5	00002500
RTIME=I*TDEL	00002510
IDGM1=IDGM1+1	00002520
ROOTR(IDGM1)=.001	00002530
ROOTI(IDGM1)=0.0	00002540
IDGM1=IDGM1+1	00002550
ROOTR(IDGM1)=-.001	00002560
ROOTI(IDGM1)=0.0	00002570
GAINDG=GAINDG*RTIME	00002580
CALL TYME(SAVE,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)	00002590
IF(ICODE.NE.0) GO TO 44	00002600
MTIME=RTIME/TDEL+0.5	00002610
IF(MTIME.EQ.0) GO TO 42	00002620
DO 41 I=1,MTIME	00002630
41 OUTPUT(I)=SAVE(T)	00002640

42	IP1=MTIME+1	00002650
	DO 43 I=IP1,IO	00002660
43	OUTPUT(I)=SAVE(I)-SAVE(I-MTIME)	00002670
44	RETURN	00002680
50	IDGM1=IDGM1+1	00002690
	ROOTR(IDGM1)=0.	00002700
	ROOTI(IDGM1)=W	00002710
	IDGM1=IDGM1+1	00002720
	ROOTR(IDGM1)=0.	00002730
	ROOTI(IDGM1)=-W	00002740
	CALL TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)	00002750
	RETURN	00002760
C=	DEBUG SUBCHK	00002770
	END	00002780
	SUBROUTINE TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)	00002790
	COMPLEX P,S,OUT,OUT1,K,KJ,KNUM	00002800
	REAL NG	00002810
	DIMENSION P(14),NG(14),K(14),OUT(1002),OUTPUT(1002),T(1002),	00002820
	ROOTR(14),ROOTI(14),TTEST(16)	00002830
	COMMON TDEL,TMAX	00002840
		00002850
C=	CHECK FOR BAD ENTRY	00002860
C=		00002870
C=	IF(IDGM1.LT.ING) GO TO 55	00002880
		00002890
C=	CHECK FOR MULTIPLE ROOTS	00002900
C=		00002910
C=		00002920
	DO 5 I=1,IDGM1	00002930
	RRP1=ROOTR(I)+.0001	00002940
	RRM1=ROOTR(I)-.0001	00002950
	RIP1=ROOTI(I)+.0001	00002960
	RIM1=ROOTI(I)-.0001	00002970
	DO 5 J=1,IDGM1	00002980
	IF(I.EQ.J) GO TO 5	00002990
	RRJ=ROOTR(J)	00003000
	RIJ=ROOTI(J)	00003010
	IF(RRM1.LT.RRJ.AND.RRP1.GT.RR.I.AND.RIM1.LT.RIJ.AND.RIP1.GT.RIJ)	00003020
	GO TO 50	

	GO TO 5	00003030
50	RUOTR(J)=RUOTR(J) - 0.0002	00003040
5	CONTINUE	00003050
	ICODE=0	00003060
	DO 11 I=1, IDGM1	00003070
11	P(I)=CMPLX(RUOTR(I), RUOTI(I))	00003080
C-		00003090
C-	DETERMINE THE K'S	00003100
C-		00003110
	DO 15 J=1, IDGM1	00003120
	S=P(J)	00003130
	CALL CPVAL(KNUM, S, NG, ING)	00003140
	KJ=(1., 0.)	00003150
	DO 25 L=1, IDGM1	00003160
	IF(L.EQ.J) GO TO 25	00003170
	KJ=KJ/(S-P(L))	00003180
25	CONTINUE	00003190
	K(J)=KJ*KNUM/GAINDG	00003200
15	CONTINUE	00003210
C-		00003220
C-	DETERMINE THE TIME RESPONSE	00003230
C-		00003240
	IO=0	00003250
	T1=-TDEL	00003260
34	IO=IO+1	00003270
	OUT1=(0., 0.)	00003280
	T1=T1+TDEL	00003290
	DO 35 J=1, IDGM1	00003300
	IF(RUOTR(J)*T1.T.-100.) GO TO 35	00003310
	OUT1=OUT1+K(J)*CEXP(T1*P(J))	00003320
35	CONTINUE	00003330
	OUTPUT(IO)=REAL(OUT1)	00003340
	UNREAL=ATMAG(OUT1)	00003350
	IF(ABS(UNREAL).LT..01)GO TO 100	00003360
	WRITE(3,101)OUT1	00003370
101	FORMAT(2F20.7)	00003380
100	CONTINUE	00003390
	T(IO)=T1	00003400

```

      IF(T1.LT.TMAX) GO TO 34
      IF(ABS(UNREAL).GT..01)WRITE(3,102)
102  FORMAT(1H1)
      RETURN
55   ICODE=3
      RETURN
C=   DEBUG SUBCHK
      END
      SUBROUTINE PLOT(X,Y,IPEN)
      DIMENSION CONT(9)
      DATA CONT/51HCCFX PLOT/HUMP*577351002;FILE FILERAY=FORP/DSK;END./
C=
      BCD =(6H      P)
      WRITE(4)BCD,X,Y,IPEN;IF(IPEN.NE.999)RETURN;LOCK 4
      CALL ZIP(CONT);RETURN
      END
      SUBROUTINE SYMBOL(X,Y,SZ,BCD,ANG,NC)
      DIMENSION BCD(13)
      T=(6H      S)
               NW1=6
      IF(MOD(NC,6).EQ.0)NW1=0
      NW =(IABS(NC)+ NW1)/6
      WRITE(4)T,X,Y,SZ,ANG,NC,(BCD(I),I=1,NW)
      RETURN
      END
      SUBROUTINE AXIS(X,Y,BCD,NC,AXLEN,ANG,RMIN,DELT)
      DIMENSION BCD(13)
      T =(6H      A)
               NW1=6
      IF(MOD(NC,6).EQ.0)NW1=0
      NW =(IABS(NC)+ NW1)/6
      WRITE(4)T,X,Y,NC,AXLEN,ANG,RMIN,DELT,(BCD(I),I=1,NW)
      RETURN
      END
      SUBROUTINE LINE(PX,PY,NPT,INC,LTP,ISM,FIRSTX,DELT,XY,DELTY)
      DIMENSION PX(NPT),PY(NPT),BCD(1)
      INTEGER A,TEMP,R,C
      S =(6H      S)

```

```

00003410
00003420
00003430
00003440
00003450
00003460
00003470
00003480
00003490
00003500
00003510
00003520
00003530
00003540
00003550
00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780

```

P = (6H	P)	00003790
SZ = .0R		00003800
ANG = 0.0 ; IX = 1		00003810
20 A = 1		00003820
R = IABS(INC)		00003830
C = NPT		00003840
NA = 0		00003850
IC = 3		00003860
IS = -1		00003870
ICA = 2		00003880
ISA = -2		00003890
NT = 1		00003900
IF(LIYP) 30, 40, 50		00003910
30 ISA = -1		00003920
B = R * LIYP * ISA		00003930
GO TO 60		00003940
40 NT = -1		00003950
GO TO 60		00003960
50 NT = LIYP; NA = -1 + NT		00003970
60 IF(ISM=3) BCD(1) = (6H+00000)		00003980
IF(ISM=1) BCD(1) = (6H000000)		00003990
DO 100 I = A, C, R		00004000
XPT = (PX(I) - FIRSTX) / DELTX		00004010
YPT = (PY(I) - FIRSTY) / DELTY		00004020
NA = NA + 1		00004030
IF(NA=NT) GO TO 110		00004040
WRITE(4) P, XPT, YPT, IC		00004050
GO TO 105		00004060
110 WRITE(4) S, XPT, YPT, SZ, ANG, IX, BCD(1)		00004070
NA = 0		00004080
105 IC = ICA		00004090
IS = ISA		00004100
100 CONTINUE		00004110
RETURN		00004120
END		00004130
		00004140
		00004150
		00004160

SUBROUTINE CPMY(ZERUR, ZEROI, INGM1, A, IDIMZ)
COMPLEX X, Y, Z

DIMENSION X(28),Y(28),Z(28),ZEROR(28),ZEROI(28),A(28)

DO 11 I=1,INGM1

11 ZEROR(I)=-ZEROR(I)

X(1)=CMPLX(ZEROR(1),ZEROI(1))

X(2)=(1.,0.)

IDIMX=2

Y(2)=(1.,0.)

IDIMY=2

MAXDO=INGM1 - 1

DO 1 IO=1,MAXDO

J=IO + 1

Y(1)=CMPLX(ZEROR(J),ZEROI(J))

IDIMZ=IDIMX + IDIMY - 1

DO 30 I1=1,IDIMZ

30 Z(I1)=(0.,0.)

DO 40 I2=1,IDIMX

DO 40 J1=1,IDIMZ

K=I2 + J1 - 1

40 Z(K)=X(I2)*Y(J1) + Z(K)

DO 50 I3=1,IDIMZ

50 X(I3)=Z(I3)

1 IDIMX=IDIMZ

DO 60 IA=1,IDIMZ

60 A(IA)=REAL(Z(IA))

DO 20 J=1,INGM1

20 ZEROR(J)=-ZEROR(J)

RETURN

END

00004170

00004180

00004190

00004200

00004210

00004220

00004230

00004240

00004250

00004260

00004270

00004280

00004290

00004300

00004310

00004320

00004330

00004340

00004350

00004360

00004370

00004380

00004390

00004400

00004410

00004420

00004430

00004440

INITIAL DISTRIBUTION LIST
NCSL REPORT 242-75

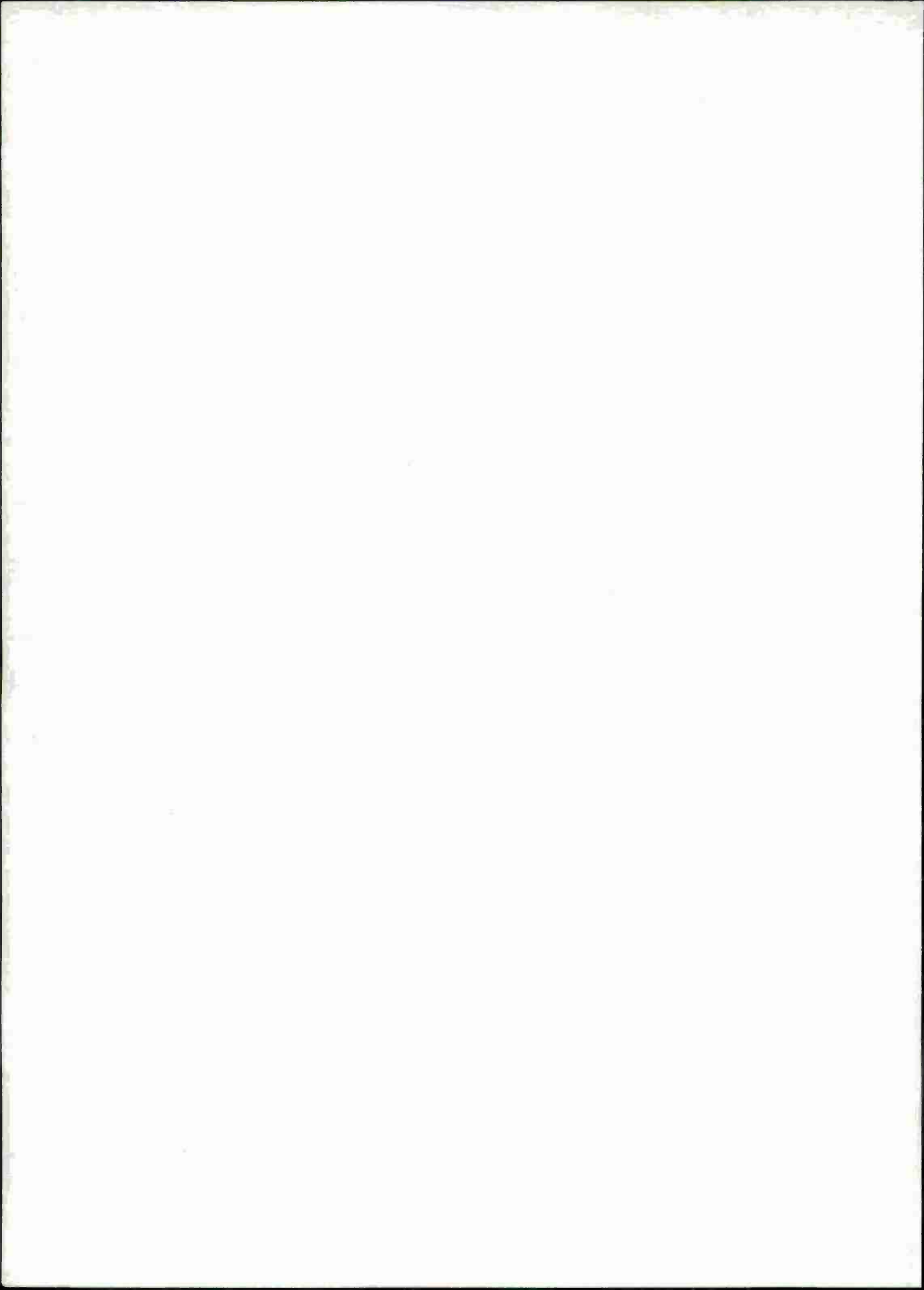
	Commander, Naval Coastal Systems Laboratory	
	(Code 112.2 - Technical Library)	(Copies 1-4)
	(Code 741 - D. W. Shepherd)	(Copy 5)
	(Code 731 - R. E. Kelly)	(Copy 6)
	(Code 731 - R. Mallory)	(Copy 7)
	(Code 731 - C. A. Dildy)	(Copy 8)
	(Code 731 - A. M. Thomas)	(Copy 9)
	(Code 722 - A. G. Holston)	(Copy 10)
	(Code 796 - K. W. Watkinson)	(Copy 11)
	(Code 732 - T. C. Watson)	(Copy 12)
	(Code 796 - D. E. Humphreys)	(Copies 13-32)
	(Code 796 - R. W. Miller)	(Copies 33-37)
	(Code 710.2 - L. F. Dewberry)	(Copies 38-42)
	(Code 724 - A. S. Piper)	(Copy 43)
427	Commander, Naval Sea Systems Command	
	(SEA 03C)	(Copy 44)
	(SEA 0322 - Benen, Sejd)	(Copies 45-46)
	(SEA 0342 - Romano)	(Copy 47)
	(SEA 03512 - Peirce)	(Copy 48)
	(SEA 03513 - Pasiuk)	(Copy 49)
	(SEA 0353 - Freund)	(Copy 50)
	(SEA 036 - Smith)	(Copy 51)
	(SEA 06H1)	(Copy 52)
	(SEA 09G32)	(Copy 53)
	(SEA 660E)	(Copy 54)
	(PMS 395)	(Copy 55)
210	Commander, Naval Surface Weapons Center, White Oak	
	(Technical Library)	(Copy 56)
223	Director, Naval Research Laboratory	
	(Ocean Engineering Dept. - R. Skop)	(Copy 57)
	(Ocean Engineering Dept. - H. Johnson)	(Copy 58)
	(Technical Library)	(Copy 59)
186	Officer-in-Charge, Civil Engineering Laboratory	
	(Code L31)	(Copy 60)
198	Commander, Naval Facilities Engineering Command	
	(Code 032C)	(Copy 61)
266	Commanding Officer, Naval Underwater Systems Center	
	(Technical Library)	(Copy 62)
265	Commander, Naval Undersea Center	
	(Technical Library)	(Copy 63)
	(Code 65 - I. Lemaire)	(Copy 64)
	(Code 6512 - R. A. Marrone)	(Copy 65)
268	Commander, Naval Weapons Center	
	(Technical Library)	(Copy 66)
54	Chief of Naval Research	
	(Code 438 - R. D. Cooper)	(Copy 67)
-	Director, Office of Naval Research, Branch Office, 495 Summer St., Boston, MA 02210	
	(Dr. Al Wood)	(Copy 68)

INITIAL DISTRIBUTION LIST (Cont'd.)
NCSL REPORT 242-75

1	Chief of Naval Material	
	(MAT 03P)	(Copy 69)
	(MAT 031)	(Copy 70)
	(MAT 032)	(Copy 71)
3	Chief of Naval Operations	
	(OP 098 - RDT&E - VADM Weymouth)	(Copy 72)
	(OP 02 - Submarine Warfare, VADM Beshamy)	(Copy 73)
	(OP 095 - Dir., ASW Programs, VADM Shear)	(Copy 74)
	(OP 21 - Strategic Sub. Div. & ULMS Prog. Coord., RADM Kaufman)	(Copy 75)
	(OP 22 - Attach Sub. Div. & SSN Prog. Coord.)	(Copy 76)
78	Director of Defense Research & Engineering	
	(Research & Technology)	(Copy 77)
28	Assistant Secretary of the Navy	
	(Research & Development)	(Copy 78)
208	Commander, Naval Oceanographic Office	
	(Code 1640 - Library)	(Copy 79)
154	Superintendent, Naval Academy	
	(Technical Library)	(Copy 80)
222	Superintendent, Naval Postgraduate School	
	(Code 2124 - Library)	(Copy 81)
236	Commander, Naval Ship Research & Development Center	
	(Code 15)	(Copy 82)
	(Code 1564)	(Copy 83)
	(Code 1576)	(Copy 84)
75	Director, Defense Documentation Center	(Copies 85-96)
-	Library of Congress, Science & Technology Division, Washington, DC 20540	(Copy 97)
-	Department of Ocean Engineering, Mass. Institute of Technology, Cambridge, MA 02138	
	(Department Library)	(Copy 98)
	(Prof. M. Abkowitz)	(Copy 99)
-	Davidson Laboratory, Stevens Institute of Technology, 711 Hudson St., Hoboken, NJ 07030	(Copy 100)
-	Iowa Institute of Hydraulic Research, Univ. of Iowa, Iowa City, IA 52240	
	(Prof. L. Landweber)	(Copy 101)
-	(Prof. V. C. Patel)	(Copy 102)
-	Penn. State University, Ordnance Research Laboratory, P. O. Box 30, State College, PA 16801	
	(Library)	(Copy 103)
340	Director, Woods Hole Oceanographic Institute	
	(Ocean Engineering Department)	(Copy 104)
-	Dept. of Mechanical Engineering, Catholic University, Washington, DC	
	(Prof. M. J. Casarella)	(Copy 105)

INITIAL DISTRIBUTION LIST (Cont'd.)
NCSL REPORT 242-75

- Oceanics, Inc., Technical Industrial Park, Plainview
L.I., New York 11803 (Copy 106)
- Society of Naval Architects & Marine Engineers,
74 Trinity Place, New York 10006
(Technical Library) (Copy 107)
- Hydronautics, Inc., Pindell School Road, Howard
County, Laurel, MD 20810
(Dr. B. L. Silverstein) (Copy 108)
(M. P. Tulin) (Copy 109)



DUDLEY KNOX LIBRARY - RESEARCH REPORTS



5 6853 01065452 8

U17203
still Sab u/