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
This effort was accomplished jointly under Task Area Number ZF 0112-001, 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
THE ANALYSIS OF LONGITUDINAL CONTROL SYSTEM FOR UNDERWATER VEHICLES

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.
Identifiers

Multiloop control system
Longitudinal feedback system
Vehicle longitudinal feedback
# TABLE OF CONTENTS

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

<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Description</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Block Diagram for Longitudinal Control System</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Block Diagram for the First Loop</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Block Diagram for the Second Loop</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Block Diagram for the Third Loop</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Block Diagram for the Fourth Loop</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>Example Data File</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>Example Problem</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>Example Problem: First Loop Analysis</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>Root Locus for the First Loop</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>Example Problem: Second, Third and Fourth Loop Analysis</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>Root Locus for the Second Loop</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>Root Locus for the Third Loop</td>
<td>22</td>
</tr>
<tr>
<td>13</td>
<td>Root Locus for the Fourth Loop</td>
<td>22</td>
</tr>
</tbody>
</table>
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\(^2\).


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

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

$$\frac{\theta}{s} = s\theta_o = \frac{N_\theta}{D}.$$
FIGURE 1. BLOCK DIAGRAM FOR LONGITUDINAL CONTROL SYSTEM
Where $N_\delta^\theta$ = 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, $\theta/e_1$, and varying the feedback gain, $K_\theta$, the system dynamics can be adjusted to yield the desired performance. Solving for this closed loop transfer function yields

$$\frac{\theta}{e_1} = \frac{sN_\delta^\theta}{D - sN_\delta^\theta K_\theta} \cdot \sqrt{1 - \frac{D}{sN_\delta^\theta K_\theta}}$$

For stability: $K_\theta > 0$.

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, $N_\delta^\theta$, 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

\[ D' = D - sN_\delta^\theta K_\theta. \]

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

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

For stability: \( K_\theta < 0 \).

![Block diagram for the second loop](image)

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 $\theta/e_2$. To form the depth rate signal ($Z$) requires the combination of the pitch signal and the vertical velocity according to the following equation

$$Z = w - U_0 \theta,$$

or

$$Z/\theta = w/\theta - U_0.$$

The $w/\theta$ transfer function is obtained by dividing the $w/\delta_s$ transfer function by the $\theta/\delta_s$ transfer function which yields

$$\frac{w/\delta_s}{\theta/\delta_s} = \frac{w}{\theta} = \frac{\frac{N^W_\delta_s}{D}}{\frac{N^\theta_\delta_s}{D}} = \frac{N^W_\delta_s}{N^\theta_\delta_s}.$$

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

The vehicle transfer function with three loops closed is

\[
\frac{Z}{e_2} = \frac{\frac{\theta}{e_2} (\frac{W}{\theta} - U_o)}{1 + \frac{\theta}{e_2} (\frac{W}{\theta} - U_o) K_Z^*}
\]

\[
\frac{Z}{e_3} = \frac{N_0^w - N_0^\theta U_o}{s D - s N_0^\theta K_z^* + K_0 N_0^\theta + K_Z^* (N_0^w - N_0^\theta U_o)}
\]

For stability \(K_Z^* > 0\).

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\(^{(e)}\). 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 (= 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

\[
\frac{Z}{Z_0} = \frac{-K_Z^* \frac{2}{e_3} s + 1}{1 - K_Z^* \frac{2}{e_3} \frac{1}{s}}
\]

\[
\frac{Z}{Z_0} = \frac{-K_Z^* (N_0^w - N_0^\theta U_o)}{s \left[D - s N_0^\theta K_z^* + K_0 N_0^\theta + K_Z^* (N_0^w - N_0^\theta U_o)\right] - K_Z^* (N_0^w - N_0^\theta U_o)}
\]

\(^{(e)}\)ibid.
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

$$\dot{\delta}_s = K_\phi \dot{\theta} = (\theta_0 - \theta) K_\phi - K_Z Z - (Z_0 - Z) K_Z .$$

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\(^{(9)}\). The vehicle length and speed, whether the vehicle is a towed or self-propelled body (inputted by means of a disk data file), and the range of loop gains to be analyzed (inputted 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


\(^{(9)}\)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 outputed 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.
<table>
<thead>
<tr>
<th>Line Number</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000:</td>
<td>10.1270, 49.3330, 0,*</td>
</tr>
<tr>
<td>0010:</td>
<td>-.15020E-01, -.57700E-03, .81000E-04,/</td>
</tr>
<tr>
<td>0020:</td>
<td>.0, -.50138E-01, .95500E-02,/</td>
</tr>
<tr>
<td>0030:</td>
<td>.0, .0, -.15709E 00,/</td>
</tr>
<tr>
<td>0040:</td>
<td>.0, -.17455E-01, -.11310E-01,/</td>
</tr>
<tr>
<td>0050:</td>
<td>-.16230E-02, .0, .0,/</td>
</tr>
<tr>
<td>0060:</td>
<td>.0, -.31545E-01, -.14600E-03,/</td>
</tr>
<tr>
<td>0070:</td>
<td>.0, -.13000E-03, -.15730E-02,/</td>
</tr>
<tr>
<td>0080:</td>
<td>.0, .0, .0,/</td>
</tr>
<tr>
<td>0090:</td>
<td>.0, .0, .0,/</td>
</tr>
<tr>
<td>0100:</td>
<td>.0, -.27695E-01, -.12797E-01,/</td>
</tr>
<tr>
<td>0110:</td>
<td>.36397E-01, .19170E-02,</td>
</tr>
</tbody>
</table>

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

Record 1 0000:

<table>
<thead>
<tr>
<th>UQ, LB, [Type]</th>
</tr>
</thead>
<tbody>
<tr>
<td>UØ - Vehicle Velocity (ft/sec)</td>
</tr>
<tr>
<td>LB - Vehicle length (ft)</td>
</tr>
<tr>
<td>[Type] - 1 if a towed body,</td>
</tr>
<tr>
<td>0 if self-propelled</td>
</tr>
</tbody>
</table>

Record 2* 0010:

<table>
<thead>
<tr>
<th>XU, ZU, MU</th>
</tr>
</thead>
<tbody>
<tr>
<td>XW, ZW, MW</td>
</tr>
<tr>
<td>XTHUSQ**, ZTHUSQ**, MTHUSQ**</td>
</tr>
<tr>
<td>XQ, ZQ, MQ</td>
</tr>
<tr>
<td>XUD, ZUD, MUD</td>
</tr>
<tr>
<td>XWD, ZWD, MWD</td>
</tr>
<tr>
<td>XQD, ZQD, MQD</td>
</tr>
<tr>
<td>XX, ZX, MX</td>
</tr>
<tr>
<td>XZ, ZZ, MZ</td>
</tr>
<tr>
<td>XDELT, ZDELT, MDELT</td>
</tr>
<tr>
<td>M, IY</td>
</tr>
</tbody>
</table>

*Nondimensional Stability Derivatives (Appendix A)

**Program reads XTHUSQ(=X' U^2), ZTHUSQ, MTHUSQ and converts to XTH( = XTHUSQ/U^2), ZTH, MTH

FIGURE 6. EXAMPLE DATA FILE
S N A M E  NON-DIMENSIONAL
LONGITUDINAL STABILITY DERIVATIVES

\[
\begin{align*}
XU &= -0.15020 \times 10^{-1} \\
XW &= 0 \\
XTH &= 0 \\
XQ &= 0 \\
XUD &= -0.16230 \times 10^{-2} \\
XWD &= 0 \\
XQD &= 0 \\
XX &= 0 \\
XZ &= 0 \\
XDELT &= 0 \\
XQD &= 0 \\
XX &= 0 \\
XZ &= 0 \\
XDELT &= 0
\end{align*}
\]

\[
\begin{align*}
ZU &= -0.57700 \times 10^{-3} \\
ZW &= -0.50138 \times 10^{-1} \\
ZTH &= 0 \\
ZQ &= -0.17455 \times 10^{-1} \\
ZUD &= 0 \\
ZWD &= -0.31545 \times 10^{-1} \\
ZQD &= -0.13000 \times 10^{-3} \\
ZX &= 0 \\
ZZ &= 0 \\
ZDELT &= -0.27695 \times 10^{-1} \\
ZQD &= 0 \\
ZX &= 0 \\
ZZ &= 0 \\
ZDELT &= 0
\end{align*}
\]

\[
\begin{align*}
MU &= 0.81000 \times 10^{-4} \\
MW &= 0.95500 \times 10^{-2} \\
MTH &= 0.15317 \times 10^{-2} \\
MQ &= -0.11310 \times 10^{-1} \\
MUD &= 0 \\
MWD &= -0.14600 \times 10^{-3} \\
MQD &= -0.15730 \times 10^{-2} \\
MX &= 0 \\
MZ &= 0 \\
MDELT &= -0.12797 \times 10^{-1} \\
MQD &= -0.15730 \times 10^{-2} \\
MX &= 0 \\
MZ &= 0 \\
MDELT &= 0
\end{align*}
\]

\[
\begin{align*}
M &= 0.36397 \times 10^{-1} \\
IY &= 0.19170 \times 10^{-2}
\end{align*}
\]

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

DIMENSIONAL COEFFICIENTS

\[
\begin{align*}
J &= 1 & DS &= 0 \\
J &= 2 & DS &= 0 \\
J &= 3 & DS &= 0.273739387358 \times 10^{-4} \\
J &= 4 & DS &= 0.118877286995 \times 10^{-2} \\
J &= 5 & DS &= 0.185102157604 \times 10^{-1} \\
J &= 6 & DS &= 0.108587460950 \times 10^{0} \\
J &= 7 & DS &= 0.120470690538 \times 10^{0}
\end{align*}
\]

DIMENSIONAL ROOTS

\[
\begin{align*}
J &= 1 & ROOTR &= -0.59231 \times 10^{-1} & ROOTI &= -0.22004 \times 10^{-1} \\
J &= 2 & ROOTR &= -0.59231 \times 10^{-1} & ROOTI &= 0.22004 \times 10^{-1} \\
J &= 3 & ROOTR &= -0.10966 \times 10^{-1} & ROOTI &= 0 \\
J &= 4 & ROOTR &= -0.70180 \times 10^{0} & ROOTI &= 0 \\
J &= 5 & ROOTR &= 0.0 & ROOTI &= 0 \\
J &= 6 & ROOTR &= 0.0 & ROOTI &= 0
\end{align*}
\]

**** X NUMERATOR ****

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

FIGURE 7. EXAMPLE PROBLEM
(Sheet 1 of 2)

12
**** Z NUMERATOR ****

DIMENSIONAL COEFFICIENTS

<table>
<thead>
<tr>
<th>J</th>
<th>ZS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>-0.0153</td>
</tr>
<tr>
<td>3</td>
<td>-0.0011658</td>
</tr>
<tr>
<td>4</td>
<td>-0.0012861</td>
</tr>
<tr>
<td>5</td>
<td>-0.00110033</td>
</tr>
</tbody>
</table>

DIMENSIONAL ROOTS

<table>
<thead>
<tr>
<th>J</th>
<th>ROOTR</th>
<th>ROOTI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.015883</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>-0.000811</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>-0.00011848</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**** I NUMERATOR ****

DIMENSIONAL COEFFICIENTS

<table>
<thead>
<tr>
<th>J</th>
<th>TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>-0.000323</td>
</tr>
<tr>
<td>3</td>
<td>-0.00005485</td>
</tr>
<tr>
<td>4</td>
<td>-0.0001853</td>
</tr>
<tr>
<td>5</td>
<td>-0.00018529</td>
</tr>
</tbody>
</table>

DIMENSIONAL ROOTS

<table>
<thead>
<tr>
<th>J</th>
<th>ROOTR</th>
<th>ROOTI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.000811</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>-0.0002149</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

ZEROS OF TD/E1

-0.0811  0.0000  -0.2149  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/Z0

-0.0811  0.0000  -0.3690  1.0384  0.0000  0.0000  0.0000  0.0000

FIGURE 7.
FIRST LOOP ANALYSIS

To analyze the first loop (pitch rate), the initial value of $K_0^*$ must be entered along with the step increment, $\Delta K_0^*$, 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).

<table>
<thead>
<tr>
<th>Enter Ktdorg, Delktd, Ktdfin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 5, 5</td>
</tr>
<tr>
<td>0.00</td>
</tr>
<tr>
<td>0.0592 -0.220 -0.0592 0.0220 -0.0811 0.0000 -0.7018 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>0.50</td>
</tr>
<tr>
<td>-0.0407 0.0000 -0.0811 0.0000 -0.0897 0.0000 -0.7667 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>-0.0307 0.0000 -0.0811 0.0000 -0.1094 0.0000 -0.8340 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>1.50</td>
</tr>
<tr>
<td>-0.0253 0.0000 -0.0811 0.0000 -0.1227 0.0000 -0.9030 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>2.00</td>
</tr>
<tr>
<td>-0.0217 0.0000 -0.0811 0.0000 -0.1328 0.0000 -0.9734 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>2.50</td>
</tr>
<tr>
<td>-0.0190 0.0000 -0.0811 0.0000 -0.1409 0.0000 -1.0449 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>3.00</td>
</tr>
<tr>
<td>-0.0170 0.0000 -0.0811 0.0000 -0.1475 0.0000 -1.1172 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>3.50</td>
</tr>
<tr>
<td>-0.0154 0.0000 -0.0811 0.0000 -0.1531 0.0000 -1.1901 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>4.00</td>
</tr>
<tr>
<td>-0.0141 0.0000 -0.0811 0.0000 -0.1578 0.0000 -1.2637 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>4.50</td>
</tr>
<tr>
<td>-0.0129 0.0000 -0.0811 0.0000 -0.1618 0.0000 -1.3376 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>5.00</td>
</tr>
<tr>
<td>-0.0120 0.0000 -0.0811 0.0000 -0.1654 0.0000 -1.4119 0.0000</td>
</tr>
<tr>
<td>0.0000 0.0000 0.0000 0.0000</td>
</tr>
</tbody>
</table>

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
Real          Imaginary
-.0592          -j .0220
-.0592          +j .0220
-.0811          j0
-.7018          j0

While the loop zeros (previously printed out) are
Real          Imaginary
-.0811          j0
-.2149          j0
0.0            j0.0

The two zero value 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 ROOT 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 KTD\_\_\_\_\_\_\_\_\_\_, DELKTD, KTD\_\_\_\_\_\_\_\_\_\_, DELKTD, KTD\_\_\_\_\_\_\_\_\_.

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

$$GAIN = K \frac{ZS(5)}{Z} \frac{DS(5)}{DS(5)}$$
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

\[ \text{DØ YØU WANT A TIMEPLT.} \]

By inputing 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.

\[ \text{ENTER KTD, KT, KZD, KZ, GAMP} \]

\[ \text{TIMEPLT FOR WHICH LØØF} \]

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 inputed 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:

\[ \text{DØ YØU WANT ANØTER TIMEPLT,} \]

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+

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>0.00</td>
<td>0.50</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>-0.1724</td>
<td>-0.1651</td>
<td>-0.1724</td>
<td>0.1651</td>
<td>-0.0811</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3992</td>
<td>0.0000</td>
<td>-0.0792</td>
<td>-0.6952</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3839</td>
<td>0.0000</td>
<td>0.1213</td>
<td>-0.9539</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3788</td>
<td>0.0000</td>
<td>0.3270</td>
<td>-1.1186</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3764</td>
<td>0.0000</td>
<td>0.5340</td>
<td>-1.2271</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3749</td>
<td>0.0000</td>
<td>0.7415</td>
<td>-1.2938</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3739</td>
<td>0.0000</td>
<td>0.9492</td>
<td>-1.3249</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3732</td>
<td>0.0000</td>
<td>1.1571</td>
<td>-1.3229</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3726</td>
<td>0.0000</td>
<td>1.3650</td>
<td>-1.2877</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3722</td>
<td>0.0000</td>
<td>1.5731</td>
<td>-1.2164</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3719</td>
<td>0.0000</td>
<td>1.7811</td>
<td>-1.1019</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

DO YOU WANT NEW KTD, KTD NOW = 1.00

DO YOU WANT NEW KT, KT NOW = -1.00

DO YOU WANT TO CONTINUE KZD R. LOCUS

ENTER KZDORG, DELKZD, KZDFIN

FIGURE 10.

(Sheet 2 of 5)
<table>
<thead>
<tr>
<th>Value</th>
<th>-0.1724</th>
<th>-0.1651</th>
<th>-0.1724</th>
<th>0.1651</th>
<th>-0.0811</th>
<th>0.0000</th>
<th>-0.6292</th>
<th>0.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.50</td>
<td>-0.0811</td>
<td>0.0000</td>
<td>0.2740</td>
<td>0.0000</td>
<td>-0.3410</td>
<td>0.0000</td>
<td>-1.3235</td>
<td>0.0000</td>
</tr>
<tr>
<td>-1.00</td>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3547</td>
<td>0.0000</td>
<td>0.4253</td>
<td>0.0000</td>
<td>-1.8776</td>
<td>0.0000</td>
</tr>
<tr>
<td>-1.50</td>
<td>-0.3594</td>
<td>0.0000</td>
<td>0.5174</td>
<td>0.0000</td>
<td>-0.0811</td>
<td>0.0000</td>
<td>-2.3814</td>
<td>0.0000</td>
</tr>
<tr>
<td>-2.00</td>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3617</td>
<td>0.0000</td>
<td>0.5822</td>
<td>0.0000</td>
<td>-2.8603</td>
<td>0.0000</td>
</tr>
<tr>
<td>-2.50</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-3.00</td>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3632</td>
<td>0.0000</td>
<td>0.6311</td>
<td>0.0000</td>
<td>-3.3242</td>
<td>0.0000</td>
</tr>
<tr>
<td>-3.50</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-4.00</td>
<td>-0.0811</td>
<td>0.0000</td>
<td>-0.3641</td>
<td>0.0000</td>
<td>0.6698</td>
<td>0.0000</td>
<td>-3.7783</td>
<td>0.0000</td>
</tr>
<tr>
<td>-4.50</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DO YOU WANT NEW KTD, KTD NOW = ** 1.00

**DO YOU WANT NEW KT, KT NOW = ** -1.00

**DO YOU WANT TO CONTINUE KZD R. LOCUS**

**ENTER KZD**

<table>
<thead>
<tr>
<th>Value</th>
<th>0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>GAIN =</td>
<td>0.000</td>
</tr>
<tr>
<td>-0.1724</td>
<td>-0.1651</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.5216</td>
<td>0.8133</td>
</tr>
<tr>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>GAIN =</td>
<td>-5.421</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.5787</td>
<td>1.0922</td>
</tr>
<tr>
<td>1.50</td>
<td></td>
</tr>
<tr>
<td>GAIN =</td>
<td>-8.131</td>
</tr>
<tr>
<td>-0.0811</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.6122</td>
<td>1.2989</td>
</tr>
</tbody>
</table>

**FIGURE 10.**

(Sheet 3 of 5)
2.00
GAIN = \(-10.841\)
\[-0.0811 \quad 0.0000 \quad -0.3717 \quad 0.0000 \quad 0.6690 \quad 0.0000 \quad -0.6357 \quad -1.4705\]
\[-0.6357 \quad 1.4705 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \\]

2.50
GAIN = \(-13.552\)
\[-0.0811 \quad 0.0000 \quad -0.3711 \quad 0.0000 \quad 0.7041 \quad 0.0000 \quad -0.6535 \quad -1.6204\]
\[-0.6535 \quad 1.6204 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \\]

3.00
GAIN = \(-16.262\)
\[-0.0811 \quad 0.0000 \quad -0.3708 \quad 0.0000 \quad 0.7321 \quad 0.0000 \quad -0.6677 \quad -1.7552\]
\[-0.6677 \quad 1.7552 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \\]

3.50
GAIN = \(-18.972\)
\[-0.0811 \quad 0.0000 \quad -0.3705 \quad 0.0000 \quad 0.7552 \quad 0.0000 \quad -0.6794 \quad -1.8788\]
\[-0.6794 \quad 1.8788 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \\]

4.00
GAIN = \(-21.683\)
\[-0.0811 \quad 0.0000 \quad -0.3703 \quad 0.0000 \quad 0.7746 \quad 0.0000 \quad -0.6892 \quad -1.9937\]
\[-0.6892 \quad 1.9937 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \\]

4.50
GAIN = \(-24.393\)
\[-0.0811 \quad 0.0000 \quad -0.3702 \quad 0.0000 \quad 0.7913 \quad 0.0000 \quad -0.6976 \quad -2.1014\]
\[-0.6976 \quad 2.1014 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \\]

5.00
GAIN = \(-27.103\)
\[-0.0811 \quad 0.0000 \quad -0.3701 \quad 0.0000 \quad 0.8058 \quad 0.0000 \quad -0.7049 \quad -2.2032\]
\[-0.7049 \quad 2.2032 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 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
\[-0.1724 \quad -0.1651 \quad -0.1724 \quad 0.1651 \quad -0.0811 \quad 0.0000 \quad -0.6292 \quad 0.0000\]
\[0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad -0.50\]

GAIN = 2.710
\[-0.0811 \quad 0.0000 \quad -0.3586 \quad 0.0000 \quad 0.3000 \quad -0.5254 \quad 0.3000 \quad 0.5254\]
\[-1.2154 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad -1.00\]

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  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  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  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  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  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

$K_\theta = 1.0$

FIGURE 12.  ROOT LOCUS FOR THE THIRD LOOP

$K_\theta = 1.0, K_\theta = -1.0$

FIGURE 13.  ROOT LOCUS FOR THE FOURTH LOOP

$K_\theta = 1.0, K_\theta = -1.0, K_z = 0.0$
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 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 \( s \).

FIGURE A1. A TYPICAL BLOCK DIAGRAM
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

\[
\frac{0}{1} = \frac{K(s + a)(s^2 + bs + c)}{s(s + d)(s + e)(s^2 + fs + g)}.
\]

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 of the characteristic equation zero contribute a term of the form \( e^{\lambda t} \) 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 \( 0 \)'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 (\( \theta \) in \( \theta/\delta_x \) 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

\[ s_1, s_2 = -\zeta \omega_n - j\omega_n \sqrt{1 - \zeta^2} \]

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.

\[ \zeta = \text{Damping Ratio} = \cos \gamma \]
\[ \omega_n = \text{Undamped Natural Frequency} \]
\[ \zeta \omega_n = \text{Total Damping} \]
\[ t_{ss} = \text{Time to reach 0.95 steady state value} = 3.0/\zeta \omega_n \]
\[ t_{1/2} = \text{Time to damp to half amplitude} = 0.69/\zeta \omega_n \]
\[ t_2 = \text{Time to double amplitude (unstable systems only)} = 0.69/\zeta \omega_n \]

**FIGURE A2. IMPORTANT FEATURES OF ROOT-LOCUS DIAGRAM**
APPENDIX B

EXPRESSIONS FOR THE LONGITUDINAL TRANSFER FUNCTION COEFFICIENTS

The longitudinal characteristic equation is

\[ \Delta_{\text{Long}} = A s^4 + B s^3 + C s^2 + D s + E \]

where

\[ A = (m' - X_{\dot{u}})(m' - Z_{\dot{u}})(I_y' - M_{\dot{u}}) - X_{\dot{u}} M_{\dot{u}} Z_{\dot{q}} \]

\[ - Z_{\dot{u}} M_{\dot{u}} X_{\dot{q}} - (m' - Z_{\dot{u}}) M_{\dot{u}} X_{\dot{q}} \]

\[ - Z_{\dot{u}} X_{\dot{w}} (I_{\dot{y}} - M_{\dot{q}}) - M_{\dot{w}} (m' - X_{\dot{u}}) Z_{\dot{q}} \]

\[ B = -(m' - X_{\dot{u}})(m' - Z_{\dot{u}}) M_{\dot{q}} - Z_{\dot{w}} (m' - X_{\dot{u}})(I_{\dot{y}} - M_{\dot{q}}) \]

\[ - X_{\dot{u}} (m' - Z_{\dot{u}})(I_{\dot{y}} - M_{\dot{q}}) - X_{\dot{w}} M_{\dot{u}} (Z_{\dot{q}} + m') \]

\[ - M_{\dot{u}} X_{\dot{w}} Z_{\dot{q}} - M_{\dot{u}} X_{\dot{w}} Z_{\dot{q}} - Z_{\dot{u}} M_{\dot{w}} X_{\dot{q}} - Z_{\dot{u}} M_{\dot{w}} X_{\dot{q}} \]

\[ - Z_{\dot{u}} M_{\dot{w}} X_{\dot{q}} - (m' - Z_{\dot{u}}) M_{\dot{u}} X_{\dot{q}} - M_{\dot{w}} (m' - Z_{\dot{u}}) X_{\dot{q}} \]

\[ + M_{\dot{u}} Z_{\dot{w}} X_{\dot{q}} + Z_{\dot{u}} M_{\dot{w}} X_{\dot{q}} - Z_{\dot{u}} X_{\dot{w}} (I_{\dot{y}} - M_{\dot{q}}) \]

\[ - Z_{\dot{u}} X_{\dot{w}} (I_{\dot{y}} - M_{\dot{q}}) - M_{\dot{w}} (m' - X_{\dot{u}})(Z_{\dot{q}} + m') \]

\[ - M_{\dot{w}} (m' - X_{\dot{u}}) Z_{\dot{q}} + M_{\dot{w}} X_{\dot{u}} Z_{\dot{q}} \]
\[ C = - (m' - X'_w) (m' - Z'_w) M'_q + Z'_w (m' - X'_w) M'_q \\
+ X'_w (m' - Z'_w) M'_q + Z'_w X'_w (I'_q - M'_w) - X'_w M'_w Z'_w \\
- M'_w X'_w (Z'_w + m') - M'_w X'_w (Z'_w + m') - Z'_w M'_w X'_w \\
- Z'_w M'_w X'_w - Z'_w M'_w X'_w - Z'_w M'_w X'_w \\
- (m' - Z'_w) M'_w X'_w - M'_w (m' - Z'_w) X'_w + M'_w Z'_w X'_w \\
+ Z'_w M'_w X'_w + Z'_w X'_w M'_w + Z'_w X'_w M'_w + Z'_w X'_w M'_w \\
- X'_w Z'_w (I'_q - M'_w) - M'_w (m' - X'_w) Z'_w - M'_w (m' - X'_w) (Z'_w + m') \\
+ M'_w X'_w (Z'_w + m') + X'_w M'_w Z'_w - M'_w X'_w Z'_w . \]
\[ D = \frac{1}{2} \left( \mathcal{Z}' M_\theta (m' - X'_\mu) + X'_\mu (m' - \mathcal{Z}'_w) M'_\theta - \mathcal{Z}'_w X'_\mu M'_\theta \right) \\
- M'_\mu X'_w \mathcal{Z}'_\theta - M'_\mu X'_w \mathcal{Z}'_w - M'_\mu X'_w (\mathcal{Z}'_w + m') \\
- \mathcal{Z}'_w M'_w X'_\theta - \mathcal{Z}'_w M'_w X'_w - \mathcal{Z}'_w M'_w X'_w \\
- M'_w (m' - \mathcal{Z}'_w) X'_\theta + M'_w \mathcal{Z}'_w X'_\theta + \mathcal{Z}'_w M'_w X'_w \\
+ \mathcal{Z}'_w X'_w M'_\theta + \mathcal{Z}'_w X'_w M'_\theta + X'_w \mathcal{Z}'_w M'_\theta \\
- M'_w (m' - X'_\mu) \mathcal{Z}'_\theta + M'_w X'_w \mathcal{Z}'_\theta + M'_w X'_w (\mathcal{Z}'_w + m') \]  

\[ E = - \mathcal{Z}'_w X'_\mu M'_\theta - M'_\mu X'_w \mathcal{Z}'_\theta - \mathcal{Z}'_w M'_w X'_\theta \\
+ \mathcal{Z}'_w M'_w X'_\theta + X'_w \mathcal{Z}'_w M'_\theta + X'_w M'_w \mathcal{Z}'_\theta \]
The pitch response transfer function is

$$\frac{\theta}{\delta_s} = \frac{N^\theta_s}{\Delta_{\text{Long}}} = \frac{A^\theta s^2 + B^\theta s^1 + C^\theta}{\Delta_{\text{Long}}}$$

where

$$A^\phi = M'_e (m' - X'_e) (m' - Z'_w) + Z'_e X'_e M'_e + M'_e X'_e Z'_e M'_w$$

$$- M'_e X'_e Z'_e + X'_e (m' - Z'_w) M'_e + Z'_e M'_w (m' - X'_e).$$

$$B^\phi = - M'_e (m' - X'_e) Z'_w - M'_e X'_e (m' - Z'_w) + Z'_e X'_e M'_w$$

$$+ Z'_e X'_e M'_e + X'_e Z'_e M'_w + X'_e Z'_e M'_w$$

$$- M'_e X'_e Z'_e - M'_e X'_e Z'_e + X'_e (m' - Z'_w) M'_e$$

$$- X'_e Z'_e M'_e - Z'_e M'_e X'_e + Z'_e M'_w (m' - X'_e).$$

$$C^\phi = M'_e X'_e Z'_w + Z'_e X'_e M'_w + X'_e Z'_e M'_w$$

$$- M'_e X'_e Z'_w - X'_e Z'_w M'_e - Z'_e M'_e X'_e.$$

B-4
The vertical velocity transfer function is

\[
\frac{\delta s}{\Delta_{\text{Long}}} = \frac{N^W_s}{A_s s^3 + B_s s^2 + C_s s + D_s}
\]

where

\[
A_w = \frac{\delta e}{\delta e} z'(m' - \dot{x}' \dot{x}') (I' - M') + \frac{\delta e}{\delta e} \dot{z}' y' q' + \frac{\delta e}{\delta e} \dot{z}' \dot{z}' y' q' + \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q'
\]

\[
+ \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q' + \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q' + \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q'
\]

\[
- \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
- \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q' + \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q'
\]

\[
- \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q' - \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q'
\]

\[
- \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q' + \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]

\[
+ \frac{\delta e}{\delta e} \ddot{z}' \dot{x}' q' - \frac{\delta e}{\delta e} \dot{z}' \ddot{z}' y' q'
\]
\[ C_{\mu \nu} = -Z_\mu^\prime (m' - X_\mu') M_\alpha + Z_\mu^\prime X_\mu' M_\alpha' + X_\mu' M_\nu' \, Z_\nu' \]
\[ + X_{\delta e} M_\mu (Z_\mu' + m') + M_\mu' Z_\mu X_\phi' + M_\mu' Z_\mu' X_\phi' \]
\[ - Z_\mu^\prime M_\mu' X_\phi' - Z_\mu^\prime M_\mu' X_\phi' - X_\mu' Z_\mu' M_\phi - X_\mu' Z_\mu' M_\mu' \]
\[ + M_\mu' (m' - X_\mu') Z_\phi' - M_\mu' X_\mu' (Z_\mu' + m') . \]

\[ D_{\mu \nu} = Z_\mu^\prime X_\mu' M_\alpha + M_\mu' Z_\mu' X_\phi' \]
\[ + X_{\delta e} M_\mu M_\phi + M_\mu' Z_\mu' X_\phi' \]
\[ - Z_\mu^\prime M_\mu' X_\phi' - X_{\delta e} M_\mu' M_\phi - M_\mu' X_\mu' Z_\phi' . \]
The forward speed transfer function is

\[
\frac{\Delta U_s}{\delta_s} = \frac{N_s^{U}}{\Delta_{\text{Long}}} = \frac{A_s U_s^3 + B_s U_s^2 + C_s U_s + D_s}{\Delta_{\text{Long}}}
\]

where

\[
A_s = X_e' (m' - \bar{Z}_w) (I' - \bar{M}^e) + M_e' X_e' \bar{Z}_w^e + \bar{Z}_w^e M_e' X_e^i
\]

\[
+ M_e' (m' - \bar{Z}_w) X_e^i + \bar{Z}_w^e X_e' (I' - \bar{M}^e) - X_e' M_e' \bar{Z}_w^e
\]

\[
B_s = -X_e' (m' - \bar{Z}_w) M_e' - X_e' (I' - \bar{M}^e) \bar{Z}_w^e
\]

\[
+ M_e' X_e' (\bar{Z}_w + m') + M_e' X_e' \bar{Z}_w^e + \bar{Z}_w^e M_e' X_e^i
\]

\[
+ \bar{Z}_w^e M_e' X_e^i + M_e' (m' - \bar{Z}_w) X_e^i - M_e' \bar{Z}_w X_e^i
\]

\[-\bar{Z}_w^e X_e' M_e' + \bar{Z}_w^e X_e' (I' - \bar{M}^e)
\]

\[-X_e' M_e' (\bar{Z}_w + m') - X_e' M_e' \bar{Z}_w^e \]
\[ C_\mu = -X'_\delta (m' - \mathbb{Z}'_\theta) M' + X'_\delta \mathbb{Z}'_\theta M' + M'_\delta X'_\delta \mathbb{Z}'_\theta \]
\[ + M'_\delta X'_\theta (\mathbb{Z}'_\theta + m') + \mathbb{Z}'_\delta M'_\theta X'_\theta + \mathbb{Z}'_\delta M'_\theta X'_\theta \]
\[ + M'_\delta (m' - \mathbb{Z}'_\theta) X'_\theta - M'_\delta \mathbb{Z}'_\theta X'_\theta - \mathbb{Z}'_\delta X'_\delta M'_\theta \]
\[ - \mathbb{Z}'_\delta X'_\delta M'_\theta - X'_\delta M'_\theta \mathbb{Z}'_\theta - X'_\delta M'_\theta (\mathbb{Z}'_\theta + m') \]

\[ D_\mu = X'_\delta \mathbb{Z}'_\theta M'_\theta + M'_\delta X'_\theta \mathbb{Z}'_\theta + \mathbb{Z}'_\delta M'_\theta X'_\theta \]
\[ - M'_\delta \mathbb{Z}'_\theta X'_\theta - \mathbb{Z}'_\delta X'_\theta M'_\theta - X'_\delta M'_\theta \mathbb{Z}'_\theta \]

B-8
APPENDIX C

LONGITUDINAL CONTROL SYSTEM ANALYSIS
PROGRAM - CARD LIST
351 CONTINUE
110 FORMAT(/"J=",'INX','ECONV='",E12.5)
      DIMENSION DS(A),XS(8),ZS(A),TS(8),ROOTR(8),ROTTI(8),ECONV(A)
      CONTINUE
      UD=U
      N=6
      NPI=N+1
      ZU=ZQ+M
      DCST=LB
      IF(ICALAGE.EQ.0) DCST=UD

C = NON-DIMENSIONAL COEFFICIENTS FOR TOWED VEHICLES

C

| X, M, XUD = E12.5, 4X, MZUD = E12.5, 4X, MUD = E12.5, / |
| X, M, XWD = E12.5, 4X, MWD = E12.5, 4X, MQD = E12.5, /  |
| X, M, QDN = E12.5, 4X, ZQD = E12.5, 4X, MQD = E12.5, /  |
| X, M, XZ = E12.5, 4X, ZX = E12.5, 4X, MZ = E12.5, /    |
| X, M, XDELT = E12.5, 4X, ZDELT = E12.5, /              |
| X, M, M = E12.5, 4X, IM = E12.5, /                    |
| 9X, M, UO = E12.5, 4X, LB = E12.5, /                  |
|                                                        |
| 351 CONTINUE                                           |
| 110 FORMAT(/"J=",'INX','ECONV='",E12.5)               |
| DIMENSION DS(A),XS(8),ZS(A),TS(8),ROOTR(8),ROTTI(8),ECONV(A) |
| CONTINUE                                              |
| UD=U                                                 |
| N=6                                                  |
| NPI=N+1                                              |
| ZU=ZQ+M                                              |
| DCST=LB                                              |
| IF(ICALAGE.EQ.0) DCST=UD                            |

C = NON-DIMENSIONAL COEFFICIENTS FOR TOWED VEHICLES

C

| DS(7) = (M=XUD)*(M=ZWD)*(IY=MQD) = XWD*ZQU*MUD       |
| XQU=ZUD*MUD = (M=XUD)*ZQU*MUD                        |
| XWD=ZUD*(IY=MQD) = XWD*(M=ZWD) = MUD                |
| XWD=ZUD*(IY=MQN) = XWD*(M=ZWN) = MUD                |
| DS(6) = (M=ZWN)*(IY=MQD) = (M=XUD)*ZWD*(IY=MQN) = (M=XUD)*(M=ZWD)*MQ |
| XW*ZQD*MUD = XWD*ZQ*MUD = XWD*ZQ*MU                  |
| XQ*ZUD*MWD = XQD*ZU*MWD = XQD*ZU*MW                  |
| +XU*ZQD*MWD = XU*(M=ZWD)*ZQ*MWN = XU*(M=ZWD)*ZQ*MW  |
| =XW*ZUD*(IY=MQD) = XWD*ZU*(IY=MQD) = XWD*ZUD*MQ     |
| =XQ*(M=ZWD)*MUD+XU*ZQ*MWN*(M=ZWD)*MU = XQ*(M=ZWD)*MU |
| +XQ*(M=ZWD)*MUD+XU*ZQ*MWN*(M=ZWD)*MU = XQ*(M=ZWD)*MU |
| DS(5) = (M=ZWN)*(IY=MQD) = XU*ZW*(IY=MQD) = (M=XUD)*ZZ*(IY=MQD) |
| +XU*MQ*ZWD = (M=XUD)*ZQ*MW = (M=XUD)*ZQ*MW = (M=XUD)*ZQ*MW |
| +XW*ZQD*MUD = XW*ZQ*MWN = XW*ZQ*MU = XW*ZQ*MWD = XW*ZQ*MWD |
| +XW*ZQD*MWD = XW*ZQ*MWN = XW*ZQ*MU = XW*ZQ*MWD = XW*ZQ*MWD |
| +XW*ZQD*MWD = XW*ZQ*MWN = XW*ZQ*MU = XW*ZQ*MWD = XW*ZQ*MWD |
| +XW*ZQD*MWD = XW*ZQ*MWN = XW*ZQ*MU = XW*ZQ*MWD = XW*ZQ*MWD |
| =C*(M=XUD)*ZQ*MW = (M=XUD)*ZQ*MW = (M=XUD)*ZQ*MW |
| =XZ*ZUD*(IY=MQD) = XW*ZU*(IY=MQN) = XU*ZM*(IY=MQD) |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
| =XW*ZUD*MQ+XW*ZU*MQ+XW*ZM*MTH                        |
DIMENSIONALIZE THE COEFFICIENTS OF THE DENOMINATOR TRANSFER FUNCTION

DO 15 I=1, NP1
15 DS(I)=DS(I)*(LR/U)**(I=1))
   IF (ITFCPRT.NE.1) GO TO 17
   WRITE (3, 16)
16 FORMAT ("DIMENSIONAL COEFFICIENTS")
   WRITE (3, 16) (J, DS(J)), J=1, NP1)
   CALL PRNBM (N, DS, ROOTR, ROOTS, FCONV)
   DO 18 J=1, N
18 IF (ECONV(J).GT.5E-09) WRITE(3, 11) J, ECONV(J)
   IF (IDIMRTS.EQ.0) GO TO 1
   WRITE (3, 19)
19 FORMAT ("DIMENSIONAL ROOTS")
   WRITE (3, 19) ((J, ROOTS(J), ROOTI(J)), J=1, N)
   WRITE (3, 21)
21 FORMAT ("")
1 N=4
   NP1=N+1

NON-DIMENSIONAL X(S) COEFFICIENTS FOR TOWED VEHICLES

XS(5)=(M-ZWD)*(IY-MQD)*XDF1+YWD*ZQD*MDELT+XQD*MWD*ZDELT
- ZQD*MWD*XDELT+YWD*(IY-MQD)*ZnELT+XQD*(M-ZWD)*MDELT
XS(4)=ZW*(IY-MQD)*XDELT=(M-ZWD)*MQ*XDELT
+XM*ZQD*MDELT+XWD*QZ*MDELT
+XQ*MWD*ZDELT+XQD*MW*ZDELT
=ZQ*MWD*XDELT+ZQD*MW*XDELT
+XM*(IY-MQD)*ZQFLT+XWD*MQ*ZDELT
+XQ*(M-ZWD)*MDELT+XQD*ZM*MDELT

WRITE (3, 5)
5 FORMAT (" **** DS(J) COEFFICIENTS ALL ZERO ****")
GO TO 20
8 FORMAT ("(12X,,J = "/(I3, 10X)*DS = "/*E20.12))
11 FORMAT ("15X,,J = "/(I3, 10X)*FCONV = "/*E15.8))
13 FORMAT ("1X,,J = "/(I3, 9X)*ROTR = "/*E12.5)*9X,*ROOTI = "/*E12.5))
XS(3) = -ZZ*(IY=MN(I))*XDL T +ZW*MQ*XDF I T *(M=ZHN)*MT* XDEL T
+XZ*ZNQ*MH FLT +XW*ZU*WDF I T +XW*ZM*MT* ZDEL T
+XTH*MWN*ZDF LT +XQ*MW2*ZDEL T +XQ*MN*ZDF LT
+XZ*(IY=MN(I))*ZDF LT +XW*MQ*ZDF LT +XW*MT*ZDF LT
+XTH*MWN*ZDF LT +XQ*MN*ZDF LT +XW*ZM*ZDF LT

XS(2) = ZZ*MQ*XNE I T +XW*MT*ZDF LT +XZ*ZQ*ZDF LT +XZ*ZM*ZDF LT
+XTH*MWN*ZDF LT +XQ*MN*ZDF LT +XW*ZM*ZDF LT

WRITE (3,212)

2122 FORMAT ('//3a3,***, X NUMFRACTOR ****")
DO 25 J=1,NP1
25 IF (XS(J),NE,n,n) GU TO 31
WRITE (3,26)
26 FORMAT ('//13x,**** XS(J) COEFFICIENTS ALL ZERO ****")
GO TO 35
28 FORMAT ('//(12x,** J = M,I3,10x,** XS = M,E20,12))
C = DIMENSIONALIZE THE COEFFICIENTS OF THE X NUMFRACTOR TRANSFER FUNCTION
C =
31 DO 32 I=1,NP1
32 XS(I) = XS(I)*DST* ((LA/U)**(I-1))
IF (ITFCPR T,NE,1) GO TO 33
WRITE (3,16)
WRITE (3,2A) ((I,XS(J)),J=1,NP1)
33 CALL PRNAM (N,XS,R0NT H,R0NT T,FCONV)
DO 34 I=1,N
34 TF (ECO NV(J),ST.,SE=09) WRITE (3,11) J,ECO NV(J)
TF(10IMNTS,F9.0) GO TO 34
WRITE (3,19)
WRITE (3,13) ((I,R0NT H(J),R0NT T(J)),J=1,N)
35 WRITE (3,21)
36 N=4
NP1=N+1
NON-DIMENSIONAL Z(S) COEFFICIENTS FOR TOWED VEHICLES

\[ ZS(S) = (M \cdot XD) \cdot (Y \cdot MQ) \cdot ZDFLT + ZQ \cdot MU \cdot XDFLT + XU \cdot ZUD \cdot MD FLT \]

\[ + (M \cdot XD) \cdot ZQD \cdot MD FLT + ZUD \cdot (Y \cdot MQ) \cdot XD FLT + ZQ \cdot MU \cdot ZDELT \]

\[ ZS(4) = XU \cdot (Y \cdot MQ) \cdot ZDELT - (M \cdot XI) \cdot MU \cdot ZDELT \]

\[ + ZQ \cdot MU \cdot XD FLT + ZQD \cdot MU \cdot XD FLT + XU \cdot ZUD \cdot MD FLT \]

\[ + XU \cdot ZUD \cdot MDELT + XQD \cdot ZU \cdot MDELT = XU \cdot ZUD \cdot MDELT + (M \cdot XI) \cdot ZG \cdot MDFLT \]

\[ + ZG \cdot ZMDFLT + XU \cdot ZMDFLT + ZQ \cdot ZMDFLT \]

\[ + ZU \cdot ZMDFLT = ZU \cdot ZMDFLT + XU \cdot ZMDFLT + ZQ \cdot ZMDFLT \]

\[ + ZG \cdot ZMDFLT + XU \cdot ZMF LT = ZU \cdot ZMDFLT + ZQ \cdot ZMDFLT \]

\[ + ZG \cdot ZMDFLT + XU \cdot ZMDFLT + ZQ \cdot ZMDFLT \]

\[ + ZG \cdot ZMDFLT + XU \cdot ZMDFLT + ZQ \cdot ZMDFLT \]

\[ + ZG \cdot ZMDFLT + XU \cdot ZMDFLT + ZQ \cdot ZMDFLT \]

\[ + ZG \cdot ZMDFLT + XU \cdot ZMDFLT + ZQ \cdot ZMDFLT \]

\[ + ZG \cdot ZMDFLT + XU \cdot ZMDFLT + ZQ \cdot ZMDFLT \]

WRITE (3*350)

350 FORMAT (/,,2X,***,** Z NUMFRATOR ****")

DO 39 J=1,NP1

39 IF (ZS(J),NE,0,N) GO TO 45

WRITE (3*40)

40 FORMAT (/,,17X,***,*** ZS(J) COEFFICIENTS ALL ZER0 ****")

GO TO 49

42 FORMAT (/,,(12X,**,J = **I3,0X,**ZS = **E20.12))

DIMENSIONALIZE THE COEFFICIENTS OF THE Z NUMFRATOR TRANSFER FUNCTION

45 DO 46 I=1,NP1

46 ZS(I)=ZS(I)*DCST*(((LR/U)**(1**1))

IF (ITFCPRT.NE,1) GO TO 47

WRITE (3*16)

WRITE (3*42) ((J,ZS(J)),J=1,NP1)

CALL PRNAM (N,ZS,ROOTR,ROOTT,FCNV)
C= DIMENSIONAL T(S) COEFFICIENTS FOR TOWED VEHICLES

C=

TS(S) = (M*XUN)*(M*ZWD)*MDLT + MWD*MUN*ZDFT + ZUN*MWD*XDEL

TS(4) = MUX*(M*ZWD)*MDLT + MWD*MUN*ZDFT + ZUN*MWD*XDEL

TS(3) = MUX*(M*ZWD)*MDLT + MWD*MUN*ZDFT + ZUN*MWD*XDEL

TS(2) = MUX*(M*ZWD)*MDLT + MWD*MUN*ZDFT + ZUN*MWD*XDEL

WRITE (3,52)

52 FORMAT (15,15E12)

C= NUMERATOR TRANSFER COEFFICIENTS FOR TOWED VEHICLES

C=

WRITE (3,54)

54 FORMAT (15,15E12)

GO TO 63

C= NUMERATOR TRANSFER COEFFICIENTS FOR TOWED VEHICLES

C=
C- FUNCTION

DO 60 I=1,NP1
TS(I)=TS(I)*((LR/U)**(I-1))
IF (ITFCPR,NE,1) GO TO 61
WRITE (3,16)
WRITE (3,56) ((J,TS(J)),J=1,NP1)
61 CALL PRNAM (N,TS,Rooth,ROOT1,ECONV)
DO 62 I=1,N
IF (ECONV(J).GT.,5E-09) WRITE (3,11) J,ECONV(J)
WRITE (3,19)
WRITE (3,13) ((J,Ro0TR(J),ROOT1(J)),J=1,N)
62 CONTINUE

C=

OTDE1(6)=TS(5)
OTDE1(5)=TS(4)
OTDE1(4)=TS(3)
OTDE1(3)=TS(2)
OTDE1(2)=TS(1)
OTDE1(1)=0.0
N=5*NP1=N
CALL PRNAM (N,OTDE1,OTDE1R,OTDF11,ECONV)
WRITE (3,3201) (OTDE1R(J10),OTDF11(J10),J10=1,N)
3201 FORMAT(/"ZEROS OF TDE1",/4(F9.4,F9.4))
N=4*NP1=N
DO 2000 I=1,5
OTDE2(I)=TS(I1)
2000 CONTINUE
CALL PRNAM (N,OTF2,OTDE2R,OTF2I,ECONV)
WRITE (3,3202) (OTDE2R(J12),OTDE21(J12),J12=1,N)
3202 FORMAT(/"ZEROS OF TDE2",/4(F9.4,F9.4))
OZDE3(6)=ZS(5)
OZDE3(5)=ZS(4)-TS(5)*U0
OZDE3(4)=ZS(3)-TS(4)*U0
OZDE3(3)=ZS(2)-TS(3)*U0
OZDE3(2)=TZ(1)*TS(2)*U(1)
OZDF3(1)=TS(1)*U(1)
N=5

CALL PRNAM(N,OZDE3,OZDF3,N/0E3,N/0F3,ECONV)
WRITE(3,3203)(OZDE3(J),OZDF3(J),J=1,N)
3203 FORMAT(5ZE9S OF 70/E3 AND 7/20s,*4(F9.4,F9.4))

CALL PRNAM(N,RTDE1,RTDF1,ROOTR,ROOT1,ECONV)
3006 CONTINUE
WRITE(3,3006) J
IF(ECONV(J),GT.,5E-09) WRITE(3,110) J,ECONV(J)

FORMAT(*DO YOU WANT TO CONTINUE KTN R, LOCUS*)
READ(3/)NVAL
IF(NVAL.EQ.1) GO TO 2997
WRITE(3,3004) NVAL
3004 FORMAT(1X,NVT,W5,5F9.4)

WRITE(3,3008)
3008 FORMAT(1X,NVT,W5,5F9.4)

3008 FORMAT("ENTER KTO RG, DEL KT, KTF IN")
READ(3,/)KTO RG, DEL KT, KTF IN
IF(DEL KT, EQ, 0) GO TO 99
NGAIN=(KTF IN - KTO RG)/DEL KT + 1
K T = KTO RG
DO 3012 JD = 1, NGAIN
RTE2(7) = DS(7)
RTE2(6) = DS(6) - KTD * TS(5)
RTE2(5) = DS(5) - KTD * TS(4) + KT * TS(5)
RTE2(4) = DS(4) - KTD * TS(3) + KT * TS(4)
RTE2(3) = DS(3) - KTD * TS(2) + KT * TS(3)
RTE2(2) = DS(2) - KTD * TS(1) + KT * TS(2)
RTE2(1) = DS(1) + KT * TS(1)
N = 6
CALL PRNBM(N, RTF2, ROOTR, RROOT1, ECONV)
DO 3009 J = 1, N
IF(ECONV(J) .GT. .5E-09) WRITE(3, 310) J, ECONV(J)
CONTINUE
WRITE(3, 3010) KT
FORMAT(F8.2)
WRITE(3, 3011)((RRO TR(J), RRO TI(J), J = 1, N))
FORMAT(4(F9.4, F9.4))
KT = KT + DEL KT
WRITE(3, 3013) KT
FORMAT("DO YOU WANT NEW KT? O=NO, 1=ENT, 2=ENT&COMP, KTD NOW=", F8.2)
READ(3,/) NVAL
IF (NVAL .EQ. 1) GO TO 3016; IF (NVAL .EQ. 2) GO TO 2997
WRITE(3, 3014)
FORMAT("DO YOU WANT TO CONTINUE KT & LOCUS")
READ(3,/) NVAL
IF (NVAL .EQ. 1) GO TO 3007
WRITE(3, 3015)
FORMAT("ENTER KT")
READ(3,/) KT
WRITE(3, 3019)
FORMAT("ENTER KZD ORG, DEL KZD, KZD FIN")
READ(3,/) KZD ORG, DEL KZD, KZD FIN
IF (NFLKZD.F9,4) GO TO 99
NGAIN = (K2I*FTN-K7D*UK(J)/NFLKZD + 1
K2D = K2DONG
DO 3022 JV = 1, NGA1N

RZDE(7) = DS(7)
RZDE(6) = DS(6) + KTU*TS(5) + (KZI/RADIAN)*ZS(5)
RZDF(5) = DS(5) + KTU*TS(4) + KT*TS(5) + (K2I/RADIAN)*ZS(4) + TS(5)*UU
RZDE(4) = DS(4) + KTU*TS(3) + KT*TS(4) + (K2D/RADIAN)*ZS(3) + TS(4)*UU
RZDF(3) = DS(3) + KTU*TS(2) + KT*TS(3) + (K2D/RADIAN)*ZS(2) + TS(3)*UU
RZDF(2) = DS(2) + KTU*TS(1) + KT*TS(2) + (K2D/RADIAN)*ZS(1) + TS(2)*UU
RZDE(1) = DS(1) + KTU*TS(1) + (K2D/RADIAN)*ZS(1) + TS(1)*UU

N = 6
CALL PRNAM (N, RZDF, K2D, TR, K2D, ECONV)
DO 3020 J = 1, N
IF (ECONV(J) .GE. 5E-09) WRITE (3, 110) J, ECONV(J)
3020 CONTINUE WRITE (3, 3110) K2D
WRITE (3, 3011) (K2D(J) + K2D(J), J = 1, N)
3021 FORMAT (4(F9.4, F9.4))
3022 K2D = K2D + NFLKZD
WRITE (3, 39090) K2D
3099 FORMAT ("DO YOU WANT NEW K2D? KTU NOW =", F8.2)
READ (3, /) NVAL
IF (NVAL .EQ. 1) GO TO 3016; IF (NVAL .EQ. 2) GO TO 2997
WRITE (3, 3023) K2D
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 K2D R. LOCUS")
READ (3, /) NVAL
IF (NVAL .EQ. 1) GO TO 3018
WRITE (3, 3025)
3025 FORMAT ("ENTER K2D")
READ (3, /) K2D
WRITE (3, 3098)
3098 FORMAT ("ENTER K2D, NFLKZ, KZFIN")
READ (3, /) K2D, NFLKZ, KZFIN
IF (NFLKZ .EQ. 0) GO TO 99
NGAIN = (KZFIN - KZORG) / DELKZ + 1
KZ = KZORG

DO 3029 JD = 1, NGAIN
RZZ0(8) = DS(7)
RZZ0(7) = DS(6) + KTD * TS(5) + K7ς * TS(5)
RZZ0(6) = DS(5) + KTD * TS(4) + K7ς * TS(4) + KZD * (ZS(4)) * TS(5) * U0 = K7 * ZS(5)
RZZ0(5) = DS(4) * KTD * TS(3) + K7ς * TS(3) + KZD * (ZS(3)) = TS(4) * U0 = K7 * ZS(4)

RZZ0(4) = DS(3) * KTD * TS(3) + K7ς * TS(3) * ZS(2) * TS(2) + ZS(2) * TS(2) * U0 = K7 * ZS(3)
RZZ0(3) = DS(2) * KTD * TS(2) + K7ς * TS(2) * ZS(1) * TS(2) * U0 = K7 * ZS(2)
RZZ0(2) = DS(1) * KTD * TS(1) = K7ς * TS(1) * U0 = K7 * ZS(1)
RZZ0(1) = K7ς * TS(1) * U0 = K7 * ZS(1)

CALL PRNBM(N, RZ70, ROOTR, ROOTI, ECONV)
DO 3027 J = 1, N
IF (ECONV(J) .GT. .5E-09) WRITE (3, 110) J, ECONV(J)
CONTINUE J WRITE (3, 3250) GAIN
WRITE (3, 3028) (ROOTR(J), ROOTI(J), J = 1, N)
FORMAT ("GAIN =", F8.4)
WRITE (3, 3032) ("DO YOU WANT NEW KZ, K7D NOW =", F8.4)
READ (3, /) NVAL
IF (NVAL .EQ. 1) GO TO 3017
WRITE (3, 3033) "DO YOU WANT TO CONTINUE KZ R. LOCUS"
READ (3, /) NVAL
IF (IVAL.EQ.1) GO TO 3024

WHITE(3,9000)
9000 FORMAT("YOU WANT A TIMEPLT")
READ(3/)IVAL
IF(IVAL.EQ.0) GO TO 94

C=
WHITE(4,4000) (TITLE(1), I=1,11)

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

WHITE(4,9003)IF,TDEL,TMAX,RTIME,PTIME,W,IPLLOT,WRITE
9003 FORMAT(T1,1H*,F5.3*1H*,F8.3*1H*,F10.5,1H*,F10.5,1H*,)
= F10.5,1H*,F10.5,1H*,)
88 WRITE(3*9001)
9001 FORMAT("ENTER KTD,KZONE,K7,GAMP")
READ(3/)KTD,KZONE,K7,GAMP
WHITE(3,9002)
9002 FORMAT("TIMEPLT FOR WHICH 100P"
= " ENTER 1=1ST, 2=2ND, 3=3RD, 4=4TH")
READ(3/)IVAL
IF(IVAL.EQ.2) GO TO 9100
IF(IVAL.EQ.3) GO TO 9101
IF(IVAL.EQ.4) GO TO 9102
GAIN=GAMP*FITS(50/US(7))
WHITE(4,9004)N10,GAIN
9004 FORMAT(T1,1H*,F15.5,1H*)
WHITE(4,9005) (NTUE1(N1),NTUF1(I1),I1=1,N10)
9005 FORMAT(4F12.5,1H*)

GO TO 9103
9100 GAIN=GAMP*(TS(5)/DS(7))
WRITE(4,*9006) N12,GAIN
9006 FORMAT(I1,H,F15.5,2H,*)
WRITE(4,*9007)((NTE2R(I2),NTE2T(I2)),I2=1,N12)
9007 FORMAT(4(F12.5,1H,*),1H/)
GO TO 9103
9101 GAIN=GAMP*(ZS(5)/DS(7))
WRITE(4,*9008) N14,GAIN
9008 FORMAT(I1,H,F15.5,2H,*)
WRITE(4,*9009)((NZDE3R(I3),NZDF3I(I3)),I3=1,N14)
9009 FORMAT(4(F12.5,1H,*),1H/)
9103 RZDE3(7)=DS(7)
RZDE3(6)=DS(6)*KTD*TS(5)+K7D*7S(5)
RZDE3(5)=DS(5)*KTD*TS(4)+KT*TS(5)+K7D*(ZS(4)=TS(5)*U0)
RZDE3(4)=DS(4)*KTD*TS(3)+KT*TS(4)+K7D*(ZS(3)=TS(4)*U0)
RZDE3(3)=DS(3)*KT*TS(3)+K7D*(7S(2)=TS(3)*U0)=KTD*TS(2)
RZDE3(2)=DS(2)*KT*TS(2)+K7D*(7S(1)=TS(2)*U0)=KTD*TS(1)
RZDE3(1)=DS(1)+KT*TS(1)=K7D*TS(1)*U0
N=6
CALL PRBM(N,RZDE3,RZDT*RZDN,*ECONV)
GO 9010 J=1,N
IF(ECONV(J),GT.,5E-09) WRITE(3,110) J,ECONV(J)
9010 CONTINUE
WRITE(4,*9011) N
9011 FORMAT(I1,2H,*)
WRITE(4,*9012)((RZDT(R6),RZDT(J6)),J6=1,N)
9012 FORMAT(4(F12.5,1H,*),1H/)
GO TO 77
9102 GAIN=GAMP*(-KZ*ZS(5))/DS(7))
WRITE(4,*9013) N14,GAIN
9013 FORMAT(I1,H,F15.5,2H,*)
WRITE(4,*9014)((NZDE3R(J7),NZDF3I(J7)),J7=1,N14)
9014 FORMAT(4(F12.5,1H,*),1H/)
RZZ0C8=DS(7)
RZZ0(7)=DS(6)=KTD*TS(5)+K7D*ZS(5)
RZZ0(6)=DS(5)=KTD*TS(4)+KT*TS(5)+K7D*(ZS(4)=TS(5)*U0)=K7*ZS(5)
- RZ70(5) = N(S(4) - K(T) * T(S(4)) - K(T) * T(S(4)) + K(T) * T(S(4)) + K(T) * T(S(4)) + T(S(4)) - U(U)) = K(T) * T(S(4)) - T(S(4)) * U(U)
- RZ70(4) = N(S(3) + K(T) * T(S(3)) + K(T) * T(S(3)) + T(S(3)) * U(U)) = K(T) * T(S(3)) + T(S(3)) * U(U)
- RZ70(3) = N(S(2) + K(T) * T(S(2)) + K(T) * T(S(2)) + T(S(2)) * U(U)) = K(T) * T(S(2)) + T(S(2)) * U(U)
- RZ70(2) = N(S(1) + K(T) * T(S(1)) + K(T) * T(S(1)) + T(S(1)) * U(U)) = K(T) * T(S(1)) + T(S(1)) * U(U)
- N = 7

CALL PRNBA(N, RZ70, RONTR, RINTI, ECONV)
DO 9015 J = 1, N
IF (ECONV(J) .GT. .5E-09) WRITE (3, 110) J, ECONV(J)
9015 CONTINUE
WRITE(4, 9016) N
9016 FORMAT (11, 2H, *)
WRITE(4, 9017) (PRONTR(J), RINTI(J)), J = 1, N
9017 FORMAT (4(F12.5, 1H, 1H)/)
77 WRITE(3, 9018)
9018 FORMAT (3, 51) ITI, XM1, XM2, IT3, XM3, XM4
WRITE(3, 51) ITI, XM1, IT2, XM2, IT3, XM3, XM4
51 FORMAT (5S, "I/O TIME = " ; I5, " sec", 5X, "m", 5X, "F6, 2, ")
   "I/O TIME = " ; I5, " sec", 5X, "m", 5X, "F6, 2, ")
   "PRORATED TIME = " ; I5, " sec", 5X, "m", 5X, "F6, 2, ")
   "TOTAL COST = " ; I5, "m", 5X, "F6, 2")
STOP; END

SUBROUTINE PRNBA (N, A, U, V, FCUMV)
DIMENSION A(8), U(8), V(8), FCUMV(8), H(8), H(8), C(8)
ICOUNT = 1
EFIX=.5E-09
CONV=1.E-35
NC=N+1

SEND COEFFICIENTS TO REDUCED COEFFICIENT STORAGE

DO 1 I=1,NC
    ECONV(I)=0.0
    H(I)=A(I)
1

INITIALIZE GUESSES AND SET REVERSAL INDICATOR NORMAL

P=0.
Q=0.0
R=0.
IREV=1

SCALING TO BE DONE AT THIS POINT AND REMOVE ALL ZERO ROUTS

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

TEST FOR VARIOUS DEGREES

7 IF(1COUNT.LT.2) GO TO 8
    ECONV(1COUNT-1)=E
8 1COUNT=1COUNT+1
9 IF(NC=1) 10,50,10
10 IF(NC=2) 12,11,12
11 N=H(1)/H(2)
20 GO TO 37
12 IF(NC=3) 10,13,14
13 P=H(2)/H(3)
14 Q=H(1)/H(3)
20 GO TO 42
C= TEST TO REVERSE COEFFICIENTS AND DO SO IF TEST SUCCEEDS
C=
14 IF(ABS(H(NC=1)/H(NC))=ABS(H(2)/H(1))) 15,21,21
15 IREFV=IREFV
16 M=NC/2
17 DO 16 I=1,M
18 NL=NC+1=1
19 F=H(NL)
20 H(NL)=H(I)
21 H(I)=F
17 IF(Q) 18,17,18
18 P=0.
20 GO TO 19
19 IF(R) 20,21,20
20 R=1./R
C= NEWTON, CALCULATE F(R) AND TEST FOR ROOT
C=
21 E=EFIX
22 R(NC)=H(NC)
23 C(NC)=H(NC)
24 R(NC+1)=0.
25 C(NC+1)=0.
26 NP=NC+1
27 DO 35 J=1,1000
28 DU 23 I1=1,NP
29 I=NC+1
30 B(I)=H(I)+R*B(I+1)
31 35 CONTINUE
32 C= NEWTON, CALCULATE F(R) AND TEST FOR ROOT
C=
33 E=EFIX
34 R(NC)=H(NC)
35 C(NC)=H(NC)
36 R(NC+1)=0.
37 C(NC+1)=0.
38 NP=NC+1
39 DO 45 J=1,1000
40 DU 21 I1=1,NP
41 I=NC+1
42 B(I)=H(I)+R*B(I+1)
43 45 CONTINUE
23 C(I)=B(I)+R*C(I+1)
    IF(ABS(B(I)/H(I))=E) 37,37,24
24 IF(C(I)) 26,25,26
25 R=R+1,
    GO TO 27
26 R=R*B(1)/C(2)
C= MAKE A BAIRSTOW REDUCTION AND CORRECT
C=
27 DO 28 I1=1,NP
    I=NC-I1
    B(I)=H(I)=P*B(I+1)-Q*R(I+2)
28 C(I)=B(I)=P*C(I+1)-Q*C(I+2)
C= TEST FOR CONVERGENCE OF BAIRSTOW PROCESS
C=
29 IF(H(2)) 30,29,30
30 IF(ABS(B(2)/H(1))=E) 31,31,32
31 IF(ABS(B(1)/H(1))=E) 42,42,32
32 CBAR=C(2)*B(2)
    D=C(3)**2*CBAR*C(4)
    IF(D) 34,33,34
33 P=P+2,
    Q=Q*(Q+1.)
    GO TO 35
34 P=P+(B(2)*C(3)=R(1)*C(4))/D
    Q=Q+=(B(2)*CBAR*B(1)*C(3))/D
35 CONTINUE
    E=E*10,
    IF(E=CONV) 22,27,36
36 CONV=E
    GO TO 42
C= LINEAR, COMPUTE AND STORE LINEAR ROOTS
C=
37 NC=NC-1
    V(NC)=0.
IF(IREV) 38, 39, 39
38 U(NC) = 1 / H
GO TO 40
39 U(NC) = R
40 DO 41 I = 1, NC
41 H(I) = H(I + 1)
GO TO 7
C=
C= QUADRATIC. SOLVE QUADRATIC AND STORE ROOTS
C=
42 NC = NC - 2
IF(IREV) 43, 44, 44
43 QP = 1 / Q
PP = P / (Q * 2,0)
GO TO 45
44 QP = Q
PP = P / 2, 0
45 F = (PP) ** 2 - QP
IF(F) 46, 47, 47
C=
C= CASE OF IMAGINARY ROOTS
C=
46 U(NC + 1) = -PP
U(NC) = = PP
V(NC + 1) = SQRT(-F)
V(NC) = = V(NC + 1)
GO TO 48
C=
C= CASE OF REAL ROOTS
C=
47 U(NC + 1) = = SIGN(ARS(PP) * SQRT(F), PP)
V(NC + 1) = 0
U(NC) = = QP / U(NC + 1)
V(NC) = = 0.
C=
C= FORM NEW REDUCED COEFFICIENTS
C=
48 DO 49 I = 1, NC
49  H(I)=B(I+2)
    GO TO 7
50  RETURN
   END
**SCARD**

FILE 1 = TIMPFLIT, UNIT = DISK, SAVE = 30, LCK, BLCKING = 3, RECORD = 10
FILE 2 = DHI, UNIT = REMUTE, LCK, RECORD = 9
FILE 3 = PRTASC, UNIT = PRINTER, RECORD = 15
FILE 4 = FORP/DSK, UNIT = DISK, SAVE = 30, LCK, AREA = 5000, RECORD = 30
FILE 5 = APER/DSK, UNIT = DISK, SAVE = 30, LCK, BLCKING = 3, RECORD = 10

**REAL**

DIMENSION XLARE(2), YLABEL(3), CONTROL(8), TITLE(11)

DIMENSION NG(14), MM(14), ROUTE(14), ROUTI(14), INPUT(100), I(100)

**COMMON**

TD(EL, TMAY)
DATA FORCE/*MPLM*", "STLF", "RAMP", "PULS", "RAMP", "SINU", "LSE", "*", "", "", "", "" STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*/"STEP*
C=
C=
KK=KK+1
ITF=ITF + 1
107 FORMAT(1H1)
AMP=GAIN
C=
IF = FORCING FUNCTION INDICATOR
IF=0 IMPLIES AN IMPULSE
IF=1 IMPLIES A STEP
IF=2 IMPLIES A RAMP
IF=3 IMPLIES A PULSE
IF=4 IMPLIES A RAMPSTEP
IF=5 IMPLIES A SINUSOID
CALL
WRITE(3,208)ITF
WRITE(3,98)
98 FORMAT(/,1X,"*****TIME RESPONSE RUN NUMBER",I3,"*****")
WRITE(3,101)I,NG(I)
101 FORMAT(10X,"NG(",I1,")="",F17.7)
IF(NGM1.EQ.0)GO TO 216
WRITE(3,210)
210 FORMAT(//,10X,"THE ZEROS OF THE NUMERATOR")
DO 82 I=1,INGM1
82 WRITE(3,209)I,ZERO(I)
WRITE(3,102)
102 FORMAT(//,10X,"THE ROOTS OF THE DENOMINATOR")
DO 80 I=1,IDGM1
80 WRITE(3,103)I,ROOT(I)
103 FORMAT(10X,"ROOT(",I1,")="",F17.7," + J ",F17.7)
WRITE(3,104)IF,FORCE(IF+1,1),AMP
104 FORMAT(//,1X,"THE FORCING FUNCTION INDICATOR (IF) =",I3,"/
A "TVME=",F17.7,")
IF(IF.EQ.3) WRITE(3,205) PTMF
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
205 FORMAT(//20X,"PTMNE=".F17.7) 00000756
IF(IF.EQ.4) WRITE(3,204) RTMNE
204 FORMAT(//20X,"TIME HAS ADJUSTED TO ",F10.4," SECUNDS.") 00000766
IF(IF.EQ.5) WRITE(3,206) W
206 FORMAT(//20X,"FREQUENCY=".F17.7) 00000776
IF (ICODE.EQ.0) GO TO 3024 WRITE(3,96) ICOD,
96 FORMAT(//" CODE =",110) 00000786
IF (ICODE.EQ.1) WRITE(3,201) 
201 FORMAT(//10X,"THE COMPLEX PART OF THE OUTPUT VECTOR BECAME", 00000796
= "SIGNIFICANT")
IF (ICODE.EQ.2) WRITE(3,202) 00000806
202 FORMAT(//10X,"MULTIPLE ROOTS ENCOUNTERED")
IF (ICODE.EQ.3) WRITE(3,201) 00000816
203 FORMAT(//10X,"BAD ENTRY - CHECK POLYNOMIAL ORDERS OF T.F.")
IF (ICODE.NE.0) GO TO 1 00000826
100 CONTINUE 00000836
WRITE(3,97) 00000846
97 FORMAT(1H1)//,HX,"TIME","13X,"OUTPUT"/) 00000856
DO 10 I=1,10,2 00000866
10 WRITE(3,200) TC11,OUTPUT(I) 00000876
200 FORMAT(2F17.7) 00000886
WRITE(3,107) 00000896
IF (WRITE.EQ.0) GO TO 11 00000906
ITER=1 00000916
WRITE(5,213) 00000926
WRITE(5,214) ITER,TMAX,IDGM1,ITH 00000936
213 FORMAT(30X) 00000946
D-5
214 FORMAT (15, F10.1, 215)
C-
12 WRITE (5, 207) (OUTPUT(I), I = 1, 10)
207 FORMAT (5 (E12.5, 1X))
11 CONTINUE
   IF (IPLOT .EQ. 0) GO TO 1
   DX = (XMAX - XMIN) / 7.
   DY = (YMAX - YMIN) / 5.
   IF (ABS (DY) .GT. 0.00001) GOTO 334
   WRITE (3, 301) 
   GOTO 1
334 IF (KK, GT, 0) GO TO 700
   CALL PLOT (0, -12, -3) 
   CALL PLOT (1, 12, -3)
700 CALL PLOT (12, 0, -3)
800 CONTINUE
301 FORMAT (///, "**** NO PLOT FOR LAST OUTPUT ****")
A = 0.
IF (YMIN, LT, 0.) AND, YMIN .GT. DY = YMIN / DY
   CALL AXIS (0, A, X), A, 0, 7, 0, XMIN, DX
   CALL AXIS (0, 0, YI, AREL, 15, 5, 90, YMIN, DY)
   CALL LINE (T, OUTPUT, IIH, 1, 0, 3, XMIN, DX, YMIN, DY)
   GO TO 1
3 CONTINUE
   IF (IPLOT .EQ. 0) GO TO 217
   CALL PLOT (0, 0, 999)
   LOCK 4
C-
217 STOP
END
SUBROUTINE CPVA1 (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-****S************************************************************
C=
C= PURPOSE
THE PURPOSE OF THIS SUBROUTINE IS TO DETERMINE THE TIME RESPONSE OF AN INPUT TO A TRANSFER FUNCTION BY TAKING THE INVERSE LAPLACE TRANSFORM BY THE METHOD OF RESIDUES.

INTRODUCTION TO AUTOMATIC CONTROL SYSTEMS - ROBERT N. CLARK (PP 70 - 77).

VARIABLES

AMP = AMPLITUDE OF THE FORCING FUNCTION

PTIME = PULSE TIME (SPECIFIED FOR IF=3)

RTIME = RAMP TIME (SPECIFIED FOR IF=4)

W = FREQUENCY OF THE SINEOIDAL INPUT (SPECIFIED FOR IF=5)

OUTPUT = VECTOR OF CALCULATED RESPONSE AMPLITUDE VALUES

T = VECTOR OF SEQUENTIAL TIME VALUES DIRECTLY RELATED TO OUTPUT

NO = NUMBER OF OUTPUT VALUES (CALCULATED)

NG = VECTOR OF NUMERATOR COEFFICIENTS

NG = DIMENSION OF THE NUMERATOR COEFFICIENTS

ROOTR = VECTOR OF REAL PARTS OF THE ROOTS

ROOTI = VECTOR OF IMAGINARY PARTS OF THE ROOTS

NOG = NUMBER OF ROOTS (ORDER OF THE DENOMINATOR)

ICODE = RETURN CODE VARIABLE

ICODE=0 IMPLIES NORMAL EXECUTION

ICODE=1 IMPLIES THAT THE COMPLEX PART OF THE OUTPUT VECTOR BECAME SIGNIFICANT.

ICODE=3 IMPLIES THAT THE ORDER OF THE DENOMINATOR WAS NOT GREATER THAN THE ORDER OF THE NUMERATOR.

ICODE=4 IMPLIES THAT THE FORCING FUNCTION INDICATOR WAS SPECIFIED INCORRECTLY

SUBROUTINES CALLED

CPVAL = COMPLEX EVALUATION OF A POLYNOMIAL

RTME = RESPONSE BY THE METHOD OF RESIDUES
REMARKS

THIS SUBROUTINE IS DESIGNED TO GENERATE THE TIME RESPONSE OF A GENERAL OUTPUT FUNCTION $X_0(s) = X_1(s)G(s)$. IN THIS EVALUATION TWO IMPORTANT ASSUMPTIONS ARE MADE.


2) MULTIPLE ROOTS OF THE DENOMINATOR MAY NOT EXIST.

SUBROUTINE TIME(AMP,PTIME,RTIME,W,OUTPUT,T,IDNG,ICODE)

REAL NG
DIMENSION P(14),NG(14),K(14),OUT(1002),T(1002),OUTPUT(1002),
ROOTR(14),ROOTI(14),SAVE(1002)
COMMON TDEL,TMAX
ICODE=A
IF(ICODE.LT.0)RETURN
GAINDG=1./AMP

DETERMINE TMAX

SMALL=1.0E-6
DO 9 IDNG=1,IDNG1
  ABSR=ABS(ROOTR(1))
  IF (ABSR.EQ.0) GO TO 10
  IF(ABSRLT.SMALL) SMALL=ABSR
9   IDNG1=IDNG1+1

IF(IDNG.EQ.0)GO TO 11
GO TO (10,20,30,40,50),IF
10 CALL Tyme(OUTPUT,T,IDNG,ICODE)
RETURN
11 IDNG1=IDNG1+1
   ROOTR(IDNG1)=0.
   ROOTI(IDNG1)=0.
RETURN
RETURN
0
10 I, DG = I, DG + 1
20 ROOT(R, I, DG) = .01
30 ROOT(I, I, DG) = .01
RETURN
30 I, DG = I, DG + 1
40 ROOT(R, I, DG) = .01
50 ROOT(I, I, DG) = .01
RETURN
60 OUTPUT(T) = SAVE(T)
70 TP1 = MT1ME + 1
80 DO 33 I = 1, TP1, 1
90 OUTPUT(T) = SAVE(T) - SAVE(I - MT1ME)
100 RETURN
110 I = RTIME / TOE + .5
120 RTIME = I * TOE
130 I, DG = I, DG + 1
140 ROOT(R, I, DG) = .01
150 ROOT(I, I, DG) = .01
160 I, DG = I, DG + 1
170 ROOT(R, I, DG) = .01
180 ROOT(I, I, DG) = .01
190 GAING = GAING * RTIME
210 IF(I, CODE, NF, 0) GOTO 44
220 MT1ME = RTIME / TOE + .5
230 IF(MT1ME, EQ, 0) GOTO 42
240 DO 41 I = 1, MT1ME
250 OUTPUT(T) = SAVE(T)
260 RETURN
42 IP1=MTIME+I
43 DO 43 I=IP1,IO
44 OUTPUT(I)=SAVE(I)=SAVE(I-MTIME)
45 RETURN
46 IDGM1=IDGM1+1
47 ROOTR(IDGM1)=0.
48 ROOTI(IDGM1)=W
49 IDGM1=IDGM1+1
50 ROOTR(IDGM1)=0.
51 RETURN
52 IDGM1=IDGM1+1
53 ROOTR(IDGM1)=0.
54 ROOTI(IDGM1)=W
55 INTEGER OUTPUT(1002), T(1002), RT(1002), R00TR(1002), ROOTI(1002), IDGM1, ICODE
56 REAL NG
57 COMMON TDEL'TMAX
58 CHECK FOR BAD ENTRY
59 IF(IDGM1.LT.IG) GO TO 55
60 CHECK FOR MULTIPLE ROOTS
61 DO 5 I=1,IDGM1
62 RRP1=ROOTR(I)+.0001
63 RRM1=ROOTR(I)-.0001
64 RIP1=ROOTI(I)+.0001
65 RIM1=ROOTI(I)-.0001
66 DO 5 J=1,IDGM1
67 IF(I.EQ.J) GO TO 5
68 RRJ=ROOTR(J)
69 RIJ=ROOTI(J)
70 IF(RRM1.LT.RRJ.AND.RRP1.GT.RRJ.AND.RIM1.LT.RIJ.AND.RIP1.GT.RIJ) GO TO 50
GO TO 5
50 RUOTR(J) = RUOTR(J) - 0.0001
S CONTINUE
ICONF = 0
10 DO 11 I = 1, IDGM1
11 P(I) = CMPLX(RUOTR(I), RUOTR(I))
C = DETERMINE THE K'S
C =
DO 15 J = 1, IDGM1
S = P(J)
CALL CPVAL(KNUM, S, NG, ING)
KJ = (1., 0., 0.)
DO 25 L = 1, IDGM1
IF(L .EQ. J) GO TO 25
KJ = KJ/(S .EQ. P(L))
25 CONTINUE
K(J) = KJ * KNUM / GATNOG
15 CONTINUE
C = DETERMINE THE TIME RESPONSE
C =
40 IO = 0
T1 = TDEL
34 40 IO = IO + 1
OUT1 = (0., 0., 0.)
T1 = T1 + TDEL
DO 35 J = 1, IDGM1
IF(RUOTR(J) * T1 + T .. 100.) GO TO 35
OUT1 = OUT1 + K(J) * EXP(T1 * P(J))
35 CONTINUE
OUTPUT(IO) = REAL(OUT1)
UNREAL = ATAN(OUT1)
IF(ABS(UNREAL) + T .. 01) GO TO 100
WRITE(3 + 101) OUT1
101 FORMAT(2F20.7)
100 CONTINUE
T(IO) = T1
IF(T1.LT.TMAX) GO TO 34
IF(ABS(SUNREAL).GT..01) WRITE(3,102)
102 FORMAT(1H1)
RETURN
55 ICQDE=3
RETURN
C= DEBUG SUBCHK
END
SUBROUTINE PLOT(X,Y,IPEN)
DIMENSION CONT(9)
DATA CONT/51HCCFX, PLOT/HUMP, 577351002J, FILE = FILERAY=FOP/DSK\END.
C=
BCD = (6H P)
WRITE(4)BCD,X,Y,IPEN;IF(IPF.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,AXIEN,ANG,RMIN,DELT)
DIMENSION BCD(13)
T=(6H A)
NW1=6
IF(MOD(NC,6).EQ.0)NW1=0
NW = (IARS(NC) + NW1)/6
WRITE(4)T,X,Y,NC,AXIEN,ANG,RMIN,DELT,(BCD(I),I=1,NW)
RETURN
END
SUBROUTINE LINE(PX,PY,NPT,TNC,LTYP,ISM,FIRSTX,DELTX,FIRSTY,DELTY)
DIMENSION PX(NPT),PY(NPT),BCD(1)
INTEGER A,TEMP,R,C
S=(6H S)
P = (6H, P)
SZ = 0, R
ANG = 0, 0; JY = 1
20 = 1
R = IAHS(INP)
C = NPT
NA = 0
IC = 3
IS = -1
ICA = 2
ISA = 2
NT = 1
IF (LTP) 30, 40, 50
30 ISA = -1
B = = R + LTP + ISA
GO TO 60
40 NT = -1
GO TO 60
50 NT = LTP; NA = 1 + NT
60 IF (ISH = 3) HCD(1) = (6H + 000000)
   IF (ISH = 1) HCD(1) = (6H + 000000)
   DO 100 IC = A, C, R
   XPT = (PX(I) - FIRSTX) / DELTX
   YPT = (PY(I) - FIRSTY) / DELTY
   NA = NA + 1
   IF (NA = NT) GO TO 110
   WRITE(4) P, XPT, YPT, IC
   GO TO 105
110 WRITE(4) S, XPT, YPT, SZ, ANG, JY, HCD(1)
   NA = 0
105 IC = ICA
   IS = ISA
100 CONTINUE
RETURN
FND

SUBROUTINE COMPY(ZERUP, ZHOD, INGHT, DM, D1, DIMZ)
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 10=1,MAXDO
10 J=IO + 1
Y(1)=CMPLX(ZEROR(J),ZEROI(J))
IDIMZ=IDIMX + INIMY - 1

DO 30 I=1,IDIMZ
30 Z(I)=(0.,0.)

DO 40 I2=1,IDIMX
40 DO 40 J1=1,IDIMZ

40 K=T2 + 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 I4=1,IDIMZ
60 A(I4)=REAL(Z(I4))

DO 20 J=1,INGM1
20 ZEROR(J)=-ZEROR(J)
RETURN
END
INITIAL DISTRIBUTION LIST
NCSL REPORT 242-75

Commander, Naval Coastal Systems Laboratory
(Code 112.2 - Technical Library)
(Code 741 - D. W. Shepherd)
(Code 731 - R. E. Kelly)
(Code 731 - R. Mallory)
(Code 731 - C. A. Dildy)
(Code 731 - A. M. Thomas)
(Code 722 - A. G. Holston)
(Code 796 - K. W. Watkinson)
(Code 732 - T. C. Watson)
(Code 796 - D. E. Humphreys)
(Code 796 - R. W. Miller)
(Code 710.2 - L. F. Dewberry)
(Code 724 - A. S. Piper)

427 Commander, Naval Sea Systems Command
(Sea 03C)
(Sea 0322 - Benen, Sejd)
(Sea 0342 - Romano)
(Sea 03512 - Peirce)
(Sea 03513 - Pasiuk)
(Sea 0353 - Freund)
(Sea 036 - Smith)
(Sea 06H1)
(Sea 09G32)
(Sea 660E)
(PMS 395)

210 Commander, Naval Surface Weapons Center, White Oak
(technical Library)

223 Director, Naval Research Laboratory
(Ocean Engineering Dept. - R. Skop)
(Ocean Engineering Dept. - H. Johnson)
(Technical Library)

186 Officer-in-Charge, Civil Engineering Laboratory
(Code L31)

198 Commander, Naval Facilities Engineering Command
(Code 032C)

266 Commanding Officer, Naval Underwater Systems Center
(Technical Library)

265 Commander, Naval Undersea Center
(Technical Library)
(Code 65 - I. Lemaire)
(Code 6512 - R. A. Marrone)

268 Commander, Naval Weapons Center
(Technical Library)

54 Chief of Naval Research
(Code 438 - R. D. Cooper)

- Director, Office of Naval Research, Branch Office, 495 Summer St., Boston, MA 02210
(Dr. Al Wood)
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
   - 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)
- 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)