DESIGN METHODS FOR SPECIFYING HANDLING QUALITIES FOR CONTROL CONFIGURED VEHICLES


D.W. VOGT

Approved for public release; distribution unlimited.
This report was prepared for the United States Air Force by the McDonnell Aircraft Company (MCAIR), a division of the McDonnell Douglas Corporation, P.O. Box 516, St. Louis, Missouri 63166 and covers the work performed under Air Force Contract F33615-73-C-3064, Project 8219, Task 821903. The work was performed from 15 January to 15 November 1973.

The investigation reported here was performed under the sponsorship of the Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio. The Air Force Project Engineers were Mr. David Mayhew (AFFML/FGl) and Capt. David Ullman.

Mr. Stanley A. LaFavor was the MCAIR program manager for this study. The principal investigator was Robert V. Brulle, from MCAIR Aerodynamics Department, and was assisted by Daniel C. (Chuck) Anderson from the Guidance and Control Department. The McPilot computer program was written and assembled by Donald W. Vogt and R. Dale Turner. The criticism and suggestions of John Hodgkinson are acknowledged.

This report was submitted by the authors in September 1973 and consists of two volumes. Volume I, titled Technical Discussion, presents the technique and rationale for the development of the McPilot computer program and the pilot rating correlations achieved. Volume II, titled McPilot Program Users Manual, describes the McPilot program, shows how to use it, and presents a complete program listing.

This technical report has been reviewed and is approved.

[Signature]
C. D. Weidman, Chief
Control Criteria Branch
AF Flight Control Division
AF Flight Dynamics Laboratory

Approved for Public Release
A method for predicting aircraft pilot ratings which is applicable to control configured vehicles (CCV's) is developed based on aircraft performance and pilot workload. A discussion of this method of handling quality analysis, and a description of four random tracking pilot rating correlations developed using the method are presented in Volume I of the report. A literature survey classifying the most recent reports applicable to flying qualities is tabulated. A user's manual for McPilot, a general purpose digital computer program developed to implement the method and correlate pilot rating data, is presented in Volume II. The correlations that were developed are used to evaluate the pilot ratings of two conventional contemporary fighter aircraft configurations. The correlations are also applied to two CCV fighter configurations for which limited amounts of handling quality data had been obtained through manned aircraft simulations.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. PROGRAM INPUT</td>
<td>2</td>
</tr>
<tr>
<td>2.1 Symbolic Coded Input</td>
<td>2</td>
</tr>
<tr>
<td>2.1.1 Card Format</td>
<td>2</td>
</tr>
<tr>
<td>2.1.2 Data Input</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2.1 Guide</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2.2 GROUP #1 General Input and Indicators</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2.3 GROUP #2 Input for Aerodynamic Data</td>
<td>6</td>
</tr>
<tr>
<td>Longitudinal Mode</td>
<td></td>
</tr>
<tr>
<td>2.1.2.4 GROUP #3 Input for Aerodynamic Data</td>
<td>8</td>
</tr>
<tr>
<td>Lateral Mode</td>
<td></td>
</tr>
<tr>
<td>2.1.2.5 GROUP #4 Equation and Optimization Input</td>
<td>9</td>
</tr>
<tr>
<td>2.1.2.6 GROUP #5 Input for Time Response and Frequency Response</td>
<td>11</td>
</tr>
<tr>
<td>2.1.2.7 GROUP #6 Input for Root Locate Auxiliary Computations</td>
<td>12</td>
</tr>
<tr>
<td>2.1.3 Model Transfer Function Input Clarification</td>
<td>12</td>
</tr>
<tr>
<td>2.1.4 Matrix Equations input</td>
<td>13</td>
</tr>
<tr>
<td>2.2 Fortran Coded Input</td>
<td>15</td>
</tr>
<tr>
<td>2.2.1 Card Format</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2 BLANK Input</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2.1 BLANKI</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2.2 COSTI</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2.3 FRATEI</td>
<td>20</td>
</tr>
<tr>
<td>2.2.2.4 Subroutine BLANK Required Cards</td>
<td>20</td>
</tr>
<tr>
<td>2.3 McPilot Deck Structure</td>
<td>22</td>
</tr>
<tr>
<td>3. PROGRAM OUTPUT</td>
<td>23</td>
</tr>
<tr>
<td>3.1 Output Sequence</td>
<td>23</td>
</tr>
<tr>
<td>3.1.1 Symbolic Input Listing</td>
<td>23</td>
</tr>
<tr>
<td>3.1.2 Airframe Characteristics</td>
<td>23</td>
</tr>
<tr>
<td>3.1.3 Out Calculations</td>
<td>24</td>
</tr>
<tr>
<td>3.1.4 Initial Optimization Data</td>
<td>24</td>
</tr>
<tr>
<td>3.1.5 Initial Stabilization</td>
<td>24</td>
</tr>
<tr>
<td>3.1.6 Initial Optimization Output</td>
<td>25</td>
</tr>
<tr>
<td>3.1.7 Optimization Output</td>
<td>25</td>
</tr>
<tr>
<td>3.1.8 Pilot Rating</td>
<td>25</td>
</tr>
</tbody>
</table>

Approved for Public Release
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.9 Frequency Response</td>
<td>25</td>
</tr>
<tr>
<td>3.1.10 Time Response Calculations</td>
<td>29</td>
</tr>
<tr>
<td>3.1.11 Root Locus Calculations</td>
<td>26</td>
</tr>
<tr>
<td>3.2 Example Problem</td>
<td></td>
</tr>
<tr>
<td>4. RUNNING HINTS</td>
<td>62</td>
</tr>
<tr>
<td>APPENDIX A - CORRELATION EQUATIONS</td>
<td>65</td>
</tr>
<tr>
<td>APPENDIX B - PROGRAM LISTING</td>
<td>67</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>Symbolic Input Data Grouping</td>
</tr>
<tr>
<td>2</td>
<td>Matrix Array Subscript Guide</td>
</tr>
<tr>
<td>3</td>
<td>McPilot Deck Structure</td>
</tr>
<tr>
<td>4</td>
<td>P-4 Longitudinal Axis Random Tracking Model</td>
</tr>
</tbody>
</table>
The purpose of Volume II of this report is to provide a User’s Manual for the MCpilot computer program which was developed to automate handling qualities calculations for conventional and CAV (Control Configured Vehicle) aircraft. The theoretical developments and technical discussion covering the material on which McPilot is based is presented in Volume I—Technical Discussion.

McPilot is a very flexible, general purpose computer program composed of a number of subprograms which perform the calculations required to evaluate aircraft handling qualities using the approach presented in Volume I. It is structured so that aircraft and pilot models, expressed in the form of linear differential equations, may be evaluated in terms of system performance functions and pilot workload, and pilot rating predictions can be calculated using pilot rating correlations developed in this study. Alternatively, the program can be used to develop new pilot rating correlations from available data. The program also performs many of the auxiliary computations necessary for engineering analysis of the pilot/aircraft system under study.

System equation inputs to the program are in matrix form:

\[ A \dot{x} = B u \]

where \( x \) is a vector of important pilot/aircraft system states, \( z \) is an input vector composed of positive or negative powers of the complex Laplace operator \( s \) and \( A \) and \( B \) are matrices whose elements are polynomials composed of positive powers of the Laplace variable \( s \). Program input, output, and computational features and options are fully described in this volume.

For reference purposes, the equations describing the system inputs, cost functions, and pilot rating formulas are tabulated in Appendix A. A complete McPilot program listing is presented in Appendix B. For a complete description of the study and McPilot program see Volume I.

1

Approved for Public Release
This section contains information regarding all cards required, and their placement to create a deck ready to be read into a CDC 6600 computer. This assumes that the McPilot program has been loaded onto a device (tape or disk pack) which is accessible by said computer.

2.1 Symbolic Coded Input

Symbolic coded input is used for the bulk of the input data. This includes all data that remains fixed for the entire job, initial starting values for those parameters that may be varied during the optimization process, indicators, title card and classification as required.

2.1.1 Card Format

The program input routine (READA) expects the following format.

<table>
<thead>
<tr>
<th>Card Columns</th>
<th>1-6</th>
<th>7</th>
<th>8-10</th>
<th>11</th>
<th>12-66</th>
<th>67-72</th>
<th>73-80</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>I</td>
<td>II</td>
<td>III</td>
<td>IV</td>
<td>V</td>
<td>VI</td>
<td>VII</td>
</tr>
</tbody>
</table>

Card Field I: Contains the symbolic name of the variable into which data contained in Field V begins loading.

Example: Card Column 1

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>BCD</td>
</tr>
<tr>
<td>TITLE</td>
<td>2A_{bbb}</td>
</tr>
</tbody>
</table>

Card Field II: Not used.

Card Field III: Contains the words DEC, BCD, TRA, INT, or is blank depending on the type of data to be loaded. The word BCD specifies that N binary coded decimal words of six characters each (N punched in column 12) beginning in column 13 are to be loaded. The word TRA denotes to the input routine that all data has been input and to return control to the calling program. The word DEC and blank are equivalent and specify that data loaded is decimal data.

BCD Example:

Card Column 1

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>BCD</td>
</tr>
<tr>
<td>TITLE</td>
<td>2A_{bbb}</td>
</tr>
</tbody>
</table>
The 2 in column 12 specifies 2 words where each word is considered to be 6 characters including blanks. The largest number of 6 character words that can be loaded from one card is 9. The user should be very careful to see that the BCD information does not get punched into Field VI. This will cause an input error.

**DEC Example:**

Card Column 1 8 12

```
CSTHX DEC 1000,.5,.10.
```

If the card is punched as above the numbers will be loaded into the machine as binary floating point numbers. If anything other than BCD, INT, TRA or blank appears in Field II then the word DEC is assumed.

**INT Example:**

Card Column 1 8 12

```
MYJOIN INT 1
MYJOIN INT 1
MYJOIN INT 1,1,1,1
```

When the word INT is used it is assumed that all numbers on the card will be loaded as integers. If only one integer is punched per card the INT may be punched or omitted.

**Card Field IV** Not used.

**Card Field V** The actual input data to the program is punched in the field V. DEC and INT data must always be left adjusted; that is it must start in column 12 on the input card. All numbers are separated by a "comma" and the field terminates with the first blank. Since field V ends with the first blank, the user may punch any comments in the remainder of the field.

**Card Field VI** This field specifies the relocation subscript for the first piece of data in Field V. If this field is blank, an initial subscript of 1 is implied. The subscript may appear anywhere within the field.
Example:
Card Column 1  12  67
    AINPUT  4,,0,,20,,10,,18.  1 or blank
    AINPUT  20,,17,,30,,15.  6

In the example above the number 4 is loaded into the first cell of the array AINPUT. On the second card 20. is loaded in the sixth cell of the array. The one and six punched in Field VI indicate the subscript for the array AINPUT.

Card Field VII Not used as far as the input routine is concerned. This may be used as a sequence number for the card.

2.1.2 Data Input - Some general notes concerning data input.
1. Nominal value for all input is zero unless noted differently.
2. If the nominal value(s) is satisfactory, no card is required. Do not insert blank cards.
3. If two or more cards are entered with the same variable name and contain the same relocation subscript, only the last will be used.
4. All variable names beginning with the letters I thru N must be whole numbers and contain no decimal unless noted. Likewise all others must contain a decimal unless noted.
5. Cards may be input in any order followed by a 'TEA' card and a END OF FILE control card.

2.1.2.1 Guide - For ease in setting up a case, the data has been grouped according to the McPilot option selected. This grouping is shown in Figure 1, and is detailed in the following sections.

2.1.2.2 GROUP 01 General Input and Indicators

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>-</td>
<td>May be omitted or if desired insert 'BCD' in columns 8, 9, 10 and a '9' in column 12 followed by up to 54 alphanumeric characters (include blanks)</td>
</tr>
<tr>
<td>ENDAIR</td>
<td>0</td>
<td>Indicator for aero. characteristics 0 - preset equations used 1 - all matrix data will be input</td>
</tr>
</tbody>
</table>

4
FIGURE 1
SYMBOLIC INPUT DATA GROUPING
<table>
<thead>
<tr>
<th>Name</th>
<th>Nominal Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INUMID</td>
<td>0</td>
<td>Indicator for mode 0 - longitudinal mode 1 - lateral directional mode (This indicator not used if INDAIR = 1)</td>
</tr>
<tr>
<td>INDCLS</td>
<td>0</td>
<td>Indicator for classification 0 - unclassified 1 - confidential 2 - secret 3 - top secret 4 - private 5 - proprietary</td>
</tr>
<tr>
<td>INDGST</td>
<td>0</td>
<td>Indicator for gusts 0 - no gusts 1 - include gust equations (This indicator not used if INDAIR = 1)</td>
</tr>
<tr>
<td>INBSTT</td>
<td>0</td>
<td>Indicator for thunderstorm condition 0 - no thunderstorm 1 - thunderstorm condition (This indicator not used if INDGST = 0)</td>
</tr>
<tr>
<td>NISEAP</td>
<td>0</td>
<td>Number of ISE terms for which auxiliary program calculations are desired. (Namely, time response and frequency response). 0 - skip</td>
</tr>
<tr>
<td>JLJUS</td>
<td>0</td>
<td>Which pilot function P(N) term will be varied (N=1,2,3,4, or 5) in root locus auxiliary program calculations. 0 - skip root locus</td>
</tr>
<tr>
<td>KTIME</td>
<td>0</td>
<td>Print time in seconds. Nominally (KTIME = 0) all intermediate optimization printout will be suppressed - unless elapsed time is within 10 sec. of total time estimated for the job at which time all intermediate results are printed out. This 10 seconds is a built in safety and any value of KTIME is an addition to the 10 seconds already built in.</td>
</tr>
</tbody>
</table>

2.1.2.3 GROUP #2 Input for Aerodynamic Data for Longitudinal Mode

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALT</td>
<td>H</td>
<td>Altitude (ft). Not required except if INDGST = 1 and for printout display</td>
</tr>
<tr>
<td>Name</td>
<td>Formulation</td>
<td>Symbol</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(\gamma)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(\delta)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(\chi)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(b)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(W)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(\rho)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(V)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(\Gamma_0)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(\alpha_A)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>(\text{INDRAF})</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{C}_{\text{LA}} & = C_{X_{\alpha}} \\
\text{C}_{\text{GAD}} & = C_{X_{\delta}} \\
\text{C}_{\text{GXQ}} & = C_{X_{\xi}} \\
\text{C}_{\text{CXU}} & = C_{X_{\eta}} \\
\text{C}_{\text{CQH}} & = C_{X_{\phi}} \\
\text{C}_{\text{CXD}} & = C_{X_{\psi}} \\
\text{C}_{\text{CRA}} & = C_{X_{\theta}} \\
\text{C}_{\text{GIAD}} & = C_{X_{\alpha}} \\
\text{C}_{\text{GQ}} & = C_{X_{\delta}} \\
\text{C}_{\text{GRU}} & = C_{X_{\xi}} \\
\text{C}_{\text{GHR}} & = C_{X_{\eta}} \\
\text{C}_{\text{GHD}} & = C_{X_{\phi}} \\
\text{C}_{\text{CMA}} & = C_{X_{\psi}} \\
\text{C}_{\text{CSAD}} & = C_{X_{\theta}} \\
\text{C}_{\text{CQ}} & = C_{X_{\delta}} \\
\text{C}_{\text{CMU}} & = C_{X_{\xi}} \\
\text{C}_{\text{CMH}} & = C_{X_{\eta}} \\
\text{C}_{\text{CSD}} & = C_{X_{\psi}}
\end{align*}
\]

Aerodynamic Characteristic data (non-dimensional/radian).

\[\text{C}_{\text{LA}}\]
<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALT</td>
<td>$H$</td>
<td>Altitude (ft). Not required except if $RDOCST = 1$ and for printout display.</td>
</tr>
<tr>
<td>AREFF</td>
<td>$S$</td>
<td>Reference wing area ($ft^2$).</td>
</tr>
<tr>
<td>SPANF</td>
<td>$b$</td>
<td>Span (ft).</td>
</tr>
<tr>
<td>WTRP</td>
<td>$W$</td>
<td>Airplane gross weight (lb).</td>
</tr>
<tr>
<td>RNH</td>
<td>$\rho$</td>
<td>Flight Air Density ($slug/ft^3$).</td>
</tr>
<tr>
<td>V</td>
<td>$V$</td>
<td>Flight Velocity (ft/sec).</td>
</tr>
<tr>
<td>GAMMA</td>
<td>$\gamma_0$</td>
<td>Initial flight path angle (deg).</td>
</tr>
<tr>
<td>AA</td>
<td>$\alpha_A$</td>
<td>Angle of attack of axis system about which aerodynamic data are given (deg).</td>
</tr>
<tr>
<td>ALFRA</td>
<td>$\alpha_f$</td>
<td>Angle of attack of axis system about which inertia data are given (deg).</td>
</tr>
<tr>
<td>MXXH</td>
<td>$I_x$</td>
<td>Moment of inertia about the x-axis (slug $ft^2$).</td>
</tr>
<tr>
<td>MZZH</td>
<td>$I_z$</td>
<td>Moment of inertia about the z-axis (slug $ft^2$).</td>
</tr>
<tr>
<td>MXXZH</td>
<td>$I_{xz}$</td>
<td>Moment of inertia about the $x_z$-axes (slug $ft^2$).</td>
</tr>
<tr>
<td>CY8</td>
<td>$c_y$</td>
<td>Aerodynamic characteristic data (non-dimensional/radian).</td>
</tr>
<tr>
<td>CY9D</td>
<td>$c_{y_d}$</td>
<td></td>
</tr>
<tr>
<td>CYP</td>
<td>$c_p$</td>
<td></td>
</tr>
<tr>
<td>CYN</td>
<td>$c_{y_n}$</td>
<td></td>
</tr>
<tr>
<td>CVDR</td>
<td>$c_{y_d}$</td>
<td></td>
</tr>
<tr>
<td>CYD1</td>
<td>$c_{y_d1}$</td>
<td></td>
</tr>
<tr>
<td>CL8</td>
<td>$c_{l_d}$</td>
<td></td>
</tr>
<tr>
<td>CL9D</td>
<td>$c_{l_d}$</td>
<td></td>
</tr>
<tr>
<td>CLP</td>
<td>$c_p$</td>
<td></td>
</tr>
<tr>
<td>CLR</td>
<td>$c_{l_p}$</td>
<td></td>
</tr>
<tr>
<td>CLD1</td>
<td>$c_{l_d1}$</td>
<td></td>
</tr>
<tr>
<td>CNB</td>
<td>$c_{n_b}$</td>
<td></td>
</tr>
<tr>
<td>CNBD</td>
<td>$c_{n_b2}$</td>
<td></td>
</tr>
<tr>
<td>CNP</td>
<td>$c_{n_p}$</td>
<td></td>
</tr>
</tbody>
</table>

8

Approved for Public Release
### Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRR</td>
<td>( C_r )</td>
<td>Aerodynamics characteristic data (non-dimensional/radian).</td>
</tr>
<tr>
<td>CKDR</td>
<td>( C_{d r} )</td>
<td></td>
</tr>
<tr>
<td>CN1</td>
<td>( C_{n1} )</td>
<td></td>
</tr>
<tr>
<td>CY1</td>
<td>( C_{y1} )</td>
<td></td>
</tr>
<tr>
<td>CY2</td>
<td>( C_{y2} )</td>
<td></td>
</tr>
<tr>
<td>CY3</td>
<td>( C_{y3} )</td>
<td></td>
</tr>
<tr>
<td>GLD1</td>
<td>( C_{ld1} )</td>
<td></td>
</tr>
<tr>
<td>GLD2</td>
<td>( C_{ld2} )</td>
<td></td>
</tr>
<tr>
<td>GLD3</td>
<td>( C_{ld3} )</td>
<td></td>
</tr>
<tr>
<td>CH1</td>
<td>( C_{h1} )</td>
<td></td>
</tr>
<tr>
<td>CH2</td>
<td>( C_{h2} )</td>
<td></td>
</tr>
<tr>
<td>CH3</td>
<td>( C_{h3} )</td>
<td></td>
</tr>
<tr>
<td>DA</td>
<td>( \delta_a )</td>
<td>Lateral control deflection, positive when producing right wing down rolling moment, must be nonzero (deg).</td>
</tr>
<tr>
<td>D1</td>
<td>( \delta_1 )</td>
<td>Control deflection (designated as ( \delta_1 ), ( \delta_2 ), or ( \delta_3 ) for aileron and other integral surfaces in lateral control system).</td>
</tr>
<tr>
<td>D2</td>
<td>( \delta_2 )</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>( \delta_3 )</td>
<td></td>
</tr>
</tbody>
</table>

**INOBAF** - Basic airframe handling qualities (MIL-8785 type). 0 - yes, compute; 1 - no, do not compute.

#### 2.1.2.5 GROUP 84 Equation and Optimization Input

<table>
<thead>
<tr>
<th>Name</th>
<th>Nominal Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLTP1</td>
<td>0.</td>
<td>Initial value of the Ith variable parameter (P's). The search routine must start at a stable point (P's can be stabilized by MINI).</td>
</tr>
<tr>
<td>CNSTMN</td>
<td>0.</td>
<td>Minimum permissible values of P's.</td>
</tr>
<tr>
<td>CNSTMK</td>
<td>0.</td>
<td>Maximum permissible values of P's.</td>
</tr>
<tr>
<td>CUTOFF</td>
<td>0.</td>
<td>The tolerances to be placed on P's to end search. (This card not required if ISETN = -1).</td>
</tr>
<tr>
<td>DEL</td>
<td>0.</td>
<td>Discrete increment of P's to be used in determining the gradient.</td>
</tr>
<tr>
<td>ISETRM</td>
<td>0.</td>
<td>The number of cost terms to be used in the cost function (maximum of 8).</td>
</tr>
</tbody>
</table>

Approved for Public Release
<table>
<thead>
<tr>
<th>Name</th>
<th>Nominal Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KPAMAM</td>
<td>0</td>
<td>The number of system parameters (P's) that will be optimized (maximum of 5).</td>
</tr>
<tr>
<td>ISETWL</td>
<td>0</td>
<td>To have program stop for tolerance on cost use -1; on P's use 0; on both cost and P's use 1.</td>
</tr>
<tr>
<td>TULISE</td>
<td>0</td>
<td>The tolerance to be placed on cost to end search. (This card not required if ISETWL = 0).</td>
</tr>
<tr>
<td>JUMP</td>
<td>0</td>
<td>Initial value of step size used in method to find minimum of a gradient line. Use 1. if unknown. (decimal point required).</td>
</tr>
<tr>
<td>NIN</td>
<td>0</td>
<td>The element numbers in y which defines the system transfer function input for each ISE term.</td>
</tr>
<tr>
<td>NMOD</td>
<td>0</td>
<td>The number of the models to be used in each ISE term (maximum of 8).</td>
</tr>
<tr>
<td>NOUT</td>
<td>0</td>
<td>The element numbers in x which defines the system transfer function output for each ISE term.</td>
</tr>
<tr>
<td>NTYPE</td>
<td>0</td>
<td>The orders of S in the input denominators (uIN = (1/S**NTYPE)); NTYPE can equal 0, 1, 2, ....</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>0</td>
<td>The weighing factors for each cost term.</td>
</tr>
<tr>
<td>NUNOOR</td>
<td>-1</td>
<td>Model numerator orders of up to 8 models.</td>
</tr>
<tr>
<td>NDDOOR</td>
<td>0</td>
<td>Model denominator orders of up to 8 models.</td>
</tr>
<tr>
<td>POLYN</td>
<td>0</td>
<td>Model numerator polynomial coefficients (Maximum number of coefficients is 40).</td>
</tr>
<tr>
<td>POLYD</td>
<td>1</td>
<td>Model denominator polynomial coefficients (Maximum number of coefficients is 60).</td>
</tr>
</tbody>
</table>

(See Section 2.1.3 for clarification of how the model data is inserted.)
<table>
<thead>
<tr>
<th>Name</th>
<th>Nominal Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICOST</td>
<td>0</td>
<td>Indicator to denote method to be used to determine cost function. Maximum of 8 terms. 0 - user supplied cost equations; 1 - Mean Square or integral squared error; 2 - root mean square or square root of ISE.</td>
</tr>
<tr>
<td>AORDER</td>
<td>-1</td>
<td>Order of polynomials to be input into the A matrix (no decimal).</td>
</tr>
<tr>
<td>BORDER</td>
<td>-1</td>
<td>Order of polynomials to be input into the B matrix (no decimal).</td>
</tr>
<tr>
<td>AINPUT</td>
<td>0</td>
<td>Coefficients of polynomials to be input into the A matrix.</td>
</tr>
<tr>
<td>BINPUT</td>
<td>0</td>
<td>Coefficients of polynomials to be input into B matrix (maximum number of coefficients is 290). (See Section 2.1.4 for clarification of the order and matrix polynomials input.)</td>
</tr>
<tr>
<td>MSKIP</td>
<td>0</td>
<td>Nominally the initial control parameters will be stabilized; # 0 - skip stabilization.</td>
</tr>
<tr>
<td>NUMEQ</td>
<td>0</td>
<td>Number of equations (Maximum of 12).</td>
</tr>
</tbody>
</table>

2.1.2.6 GROUP 35 Input for Time Response & Frequency Response

**Auxiliary Computations**

<table>
<thead>
<tr>
<th>Name</th>
<th>Nominal Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISEAX</td>
<td>0</td>
<td>The ISE term(s) (up to 8 max.) for which you desire these auxiliary calculations.</td>
</tr>
<tr>
<td>NFRQ</td>
<td>21</td>
<td>The number of omega's to use for frequency response calculations (maximum of 21).</td>
</tr>
<tr>
<td>OMEGAS</td>
<td></td>
<td>Calculation frequencies. The 21 nominal values are .01, .05, .1, .15, .2, .3, .5, .7, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 8, 10.</td>
</tr>
<tr>
<td>INDTRN</td>
<td>0</td>
<td>Time response option for input. 0 - unit step input; 1 - unit impulse input.</td>
</tr>
</tbody>
</table>

11

Approved for Public Release
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFI</td>
<td>0</td>
<td>Method of integration. 0 - predictor corrector; 1 - Runge-Kutta.</td>
</tr>
<tr>
<td>IMAX</td>
<td>.1</td>
<td>Delta time for print interval.</td>
</tr>
<tr>
<td>NTIME</td>
<td>2</td>
<td>The number of input time values (maximum of 60).</td>
</tr>
<tr>
<td>RTIMES</td>
<td>0, 3</td>
<td>Input time values.</td>
</tr>
<tr>
<td>RINPUT</td>
<td>1, 1</td>
<td>Corresponding input values at RTIMES listed.</td>
</tr>
</tbody>
</table>

2.1.2.7 **GROUP #6 Input for Root Locus Auxiliary Computations**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCASE</td>
<td>1</td>
<td>The number of root loci to be run (maximum of 4).</td>
</tr>
<tr>
<td>KGA1N</td>
<td>2</td>
<td>Gain mode. 2 180° locus; 1 0° locus.</td>
</tr>
<tr>
<td>DLT</td>
<td>.125</td>
<td>Step size between points.</td>
</tr>
<tr>
<td>TOL</td>
<td>.0001</td>
<td>Error tolerance.</td>
</tr>
<tr>
<td>NDLY</td>
<td>-12, 4, -2, 8</td>
<td>Computation area boundary for root locus plot. 4 values must be input (if nominal values aren't satisfactory) for each case, and in the following order - left, right, bottom and top respectively.</td>
</tr>
</tbody>
</table>

2.1.3 **Model Transfer Function Input Clarification** - Entering the desired model(s) numerator and denominator order is done by entering an integer subscript anywhere in the field from card column 67 to 72. For example, for model number 4, use a subscript of 4 on the cards MNOMIN and MNOMIN. Enter the coefficients using POLYN and POLYD starting with the constant term. Each succeeding term is the coefficient of the next highest power of S. If more than one model is entered, start with lowest numbered model. For a zero model, no input is required for this section. It should be noted that a subscript value entered on a card starting with POLYN or POLYD is needed only if more than one card is required (or desired) to input the coefficients. The following example should explain the above more clearly.
Example: Transfer function desired on model 2.

Model 2 numerator order is 2, denominator order is 4.

Card Column 1 8 12

NUMER  INT  2

PÔLYN 331., 183., -21.6

DENOM  INT  4

PÔLYD 331., 673., 104.6, 19.72, 1.

Note: INT was not required above since only 1 integer value was entered per card. See 2.1.1 for more information on use of INT and the subscript field, columns 67-72.

2.1.4 Matrix Equations Input - To input equations into the matrix it is necessary to input the order and the coefficients into their respective arrays correctly. This can be done by entering to integer subscripts anywhere in the field from card column 57 to 72. The matrix subscript guide is shown in Figure 2. This is for a 12 x 12 matrix, and the number in each square is the subscript to be used for the array AORDER (or BORDER). Enter the order for all equations to be input in this fashion. Then enter the coefficients using AINPUT (or BINPUT) starting with the constant term. Each succeeding term is the coefficient of the next higher power of S. Start with the polynomial whose order has the lowest subscript value, then proceed to the next polynomial, etc. It should be noted that a subscript value entered on a card starting with AINPUT (or BINPUT) is needed only if more than one card is required (or desired) to input the coefficients.

The following example should explain the above more clearly. (Note - when BLANK sub-routine is used to insert matrix array data, the row, column designation is used which is discussed in Section 2.2.2.1.)

Example: A second order term having ascending order coefficients of \(-1.6\), \(-0.08\), and \(0.024\) is to be entered into the 5th column, 6th row. Also a first order term with ascending order coefficients of \(-1.0\) and \(-0.05\) is to be entered into the 8th column, 8th row. Both forms shown in the following example are acceptable. The proper subscripts from Figure 2, would be 54 and 92 respectively.
<table>
<thead>
<tr>
<th>Row</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>13</td>
<td>25</td>
<td>37</td>
<td>49</td>
<td>61</td>
<td>73</td>
<td>85</td>
<td>97</td>
<td>109</td>
<td>121</td>
<td>133</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>14</td>
<td>26</td>
<td>38</td>
<td>50</td>
<td>62</td>
<td>74</td>
<td>86</td>
<td>98</td>
<td>110</td>
<td>122</td>
<td>134</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>15</td>
<td>27</td>
<td>39</td>
<td>51</td>
<td>63</td>
<td>75</td>
<td>87</td>
<td>99</td>
<td>111</td>
<td>123</td>
<td>135</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>16</td>
<td>28</td>
<td>40</td>
<td>52</td>
<td>64</td>
<td>76</td>
<td>88</td>
<td>100</td>
<td>112</td>
<td>124</td>
<td>136</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>17</td>
<td>29</td>
<td>41</td>
<td>53</td>
<td>65</td>
<td>77</td>
<td>89</td>
<td>101</td>
<td>113</td>
<td>125</td>
<td>137</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>18</td>
<td>30</td>
<td>42</td>
<td>54</td>
<td>66</td>
<td>78</td>
<td>90</td>
<td>102</td>
<td>114</td>
<td>126</td>
<td>138</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>19</td>
<td>31</td>
<td>43</td>
<td>55</td>
<td>67</td>
<td>79</td>
<td>91</td>
<td>103</td>
<td>115</td>
<td>127</td>
<td>139</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>20</td>
<td>32</td>
<td>44</td>
<td>56</td>
<td>68</td>
<td>80</td>
<td>92</td>
<td>104</td>
<td>116</td>
<td>128</td>
<td>140</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>21</td>
<td>33</td>
<td>45</td>
<td>57</td>
<td>69</td>
<td>81</td>
<td>93</td>
<td>105</td>
<td>117</td>
<td>129</td>
<td>141</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>22</td>
<td>34</td>
<td>46</td>
<td>58</td>
<td>70</td>
<td>82</td>
<td>94</td>
<td>106</td>
<td>118</td>
<td>130</td>
<td>142</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>23</td>
<td>35</td>
<td>47</td>
<td>59</td>
<td>/1</td>
<td>83</td>
<td>95</td>
<td>107</td>
<td>119</td>
<td>131</td>
<td>143</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>24</td>
<td>36</td>
<td>48</td>
<td>60</td>
<td>72</td>
<td>84</td>
<td>96</td>
<td>108</td>
<td>120</td>
<td>132</td>
<td>144</td>
</tr>
</tbody>
</table>

FIGURE 2
MATRIX ARRAY SUBSCRIPT GUIDE
### 2.2 Fortran Coded Input

Fortran coded input is also used for data input. It is of the form of a user-written subroutine called BLANK, and is divided into three sections called BLANKI, COSTI, and PRATI.

BLANKI is used to insert the equations that contain the optimization parameters which are varied to minimize the cost function \((P')s\) into the \(A\) & \(B\) matrices. BLANKI can also be used to insert other \(A\) & \(B\) matrix array terms in place of the symbolic data input cards AINPUT & BINPUT. This feature can be used to great advantage in analyses of the effects of control system gains by generalizing the equations and letting BLANKI perform the computations. In this manner the effects of control variations on pilot ratings can be easily made by changing a card or two in BLANKI.

COSTI allows the user to insert non-ISE cost function terms or equations into the program. COSTI Fortran input is required for programming cost functions that combine ISE terms in a non-linear manner and/or use optimization parameters. If a user input cost function is programmed, the corresponding symbolic data input card I/COST must have a zero for that term.

PRATI allows the user to enter the pilot rating equation. This equation can consist of optimization parameters and cost functions in any programmable arithmetical forms.
2.2.1 Card Format (Basic Information)

| Card Field I | Comment information is designated by a c, *, or $ in column 1. Comments may appear anywhere in columns 2-80. A comment statement has no effect upon the program but does appear on the listing. |
| Card Field II | Used for statement label (unsigned integer) which can be referred to from other sections of the program. The integer may appear anywhere in the field and must be a unique statement label. |
| Card Field III | A Fortran statement may be continued for 19 lines. Continuation of a Fortran statement is denoted by any character in column 6 other than zero. A blank or a zero denotes the first line of a statement. |
| Card Field IV | Used for the Fortran statement. Any type of Fortran statement is allowed as long as the computation remains within the BLANK subroutine and does not change the optimization parameters. |
| Card Field V | Positions beyond 72 may be used for identification codes or sequencing but are not processed by the compiler. |

2.2.2 BLANK Input - BLANK is a user written subroutine, and consists of three sections: BLANKI, COSTI, and PRANKI.

2.2.2.1 BLANKI - The BLANKI section is provided to input those system A and B matrix elements that are functions of the parameters PON which are varied to minimize the cost function. In addition, BLANKI can be used to input other A and B matrix elements if desired. The user coded Fortran statements are inserted after the card 'ENTRY BLANKI'. The variable names available and description follow along with an example.

ABorder \((i,j)\) is used to input the order, an integer value, for the \(i\)th row and \(j\)th column entry in the A matrix.

Border \((i,j)\) similarly used for B matrix.
A(i,j,k) is the name for the ith row, jth column and kth coefficient where k = 1 is used for the constant term coefficient, k = 2 for the first order term coefficient, etc. of the entering polynomial.

B(i,j,k) similarly used for B matrix.
P(n) is the array name of the parameters which are varied to minimize the cost function and N = 1, 2, 3, 4 or 1.

KNTISE is a counter which is zero only once -- the first time BLANK is called. It can be used to save computation time on inputs that do not vary (see below).

Card Column

```
ENTRY BLANK
   IF(KNTISE.NE.0)GO TO 30
   DO 20 I = 1, 12
       AORDER (1,1) = -1
       AORDER (1,3) = 1
       AORDER (4,1) = -1
       AORDER (1,4) = -1
       AORDER (1,5) = 0
       AORDER (4,4) = 0
   L(1,1,1) = 1.
   L(4,4,1) = 1.
   AORDER (6,8) = 2
   AORDER (6,6) = 2
   TAU = .2
   30 A(6,8,1) = -2. *P(1)
   A(6,8,2) = P(1) * (TAU-2.*P(2))
   A(6,8,3) = P(1) * TAU * P(2)
   A(6,6,1) = 2.
   A(6,6,2) = TAU * P(3)
   A(6,6,3) = TAU * P(3)
   RETURN
```
In the above example, the first time BLANK is called, the polynomial orders of rows 1 and 4 and columns 1 and 4 are set to -1, except for array locations 1,1 & 4,4. The order of these locations are set to zero, and 1.0 inserted in the element. The array locations having an order of -1 are ignored by the computer. This technique is used to eliminate the automatically programmed u & h equations in the longitudinal mode (eliminating the phugoid). The values themselves are still in the matrix locations but are not used. The other Fortran statement then insert the following expressions in array locations 6,8 & 6,6.

\[ P(1)[rP(2)S^2 + (r - 2P(2))S - 2] \text{ in array location 6,8} \]
\[ [-P(3)S^2 + (r + 2P(3))S + 2] \text{ in array location 6,6} \]

2.2.2.2 COSTI - In the BLANK subroutine after card 'ENTRY COSTI' the user supplied cost function terms are entered. This is required whenever a cost function is not just a weighted linear combination of ISE (or \( V^2 \)) terms. If a user supplied cost term is entered, the symbolic input data card ICOST must have a zero entered for the corresponding cost function term (see example below). The variable names and description follow along with the previous example.

ACOST(I) is the variable name that must be set equal to the supplied cost function. The subscript (I) equals 1 to 8 and depends on the cost function term being supplied.

P(N) is the array name of the parameters which are varied to minimize the cost function (N = 1, 2, 3, 4, or 5).

In this example the user is supplying the second cost term of a total cost function defined as:

\[ J = 75.0 \times \text{e}^{-0.777I_1} \]

\( c_0 = \sqrt{\text{ISE}} \), numerically equal to the r.m.s. roll angle. \( c_0 \) is computed directly and no user supplied cost term is required.

The user supplied portion of the second term (1 - \( \text{e}^{-0.777I_1} \)) uses \( I_1 \) (the pilot lead) and for this example \( I_1 \) is designated as P(2). To input this cost function term requires user supplied Fortran statements as follows:

18

Approved for Public Release
The other symbolic data required to complete the cost function input would be:

IWORDS 2, 0

The 2 indicates a standard ISO (c) to be computed for the first cost term, and the 0 indicates that a user written cost function term has been input for the second term.

WEIGHT 75., 3.25

This defines the coefficient weight on the two cost function terms to provide the complete cost function.

ISETRM 2

This indicates that two cost function terms are being computed.

NJN 1, J

NIOUT K, L

I and K are the matrix element number in u and X respectively which define the system transfer function input and output for the first (c) cost term. These data would depend on how the equations are input in the A and B matrices. Usually I = 1 and K = 2 for the a/b transfer function. J and L are the input and output element numbers for the second cost term. The transfer function for this second term is not used in the cost term calculation. J and L are dummy numbers which are always required whenever a user cost term is specified. Any dummy system transfer function can be specified. This dummy transfer function is computed and printed but not used in any other computation.
2.2.2.3 **PRATEI** - In the BLANK subroutine after card ENTRY PRATEI, the equation to calculate the pilot rating is entered if desired. The variable names and description follow along with an example.

PR is the variable name that must be set equal to the supplied pilot rating equation.

\[ \text{ACOST}(1) \] - cost function for 1st ISE term.

\[ P(N) \] - pilot parameter, Nth value.

In this example the pilot rating is:

\[ PR = 1 + 71 \times e^{-77.1} \]

where \( e^{-77.1} \) is \( \text{ACOST}(1) \) (the first cost term)

and \( (1 - e^{-77.1}) \) is \( \text{ACOST}(2) \) (see example in previous section 2.2.2.2)

Card Column 7

**ENTRY PRATEI**

\[ PR = 1 + 71 \times \text{ACOST}(1) + 3.1 \times \text{ACOST}(2) \]

**RETURN**

2.2.7.4 **Subroutine BLANK Required Cards** - The listing below shows the required cards of the BLANK subroutine, except for comment cards (C in column 1) which are optional.
CONTRAILS

SUBROUTINE BLANK

THIS SUBROUTINE IS FILLED IN BY USER, WHO MAKES THE A AND/OR B MATRICES
FUNCTIONS OF THE P(*)$S
SIMULTANEOUSLY THE USER MAKES THE AORDER AND BORDER MATRICES COMPATIBLE
WITH THE A AND B MATRICES RESPECTIVELY

BLANK COMPUTES THOSE ELEMENTS OF THE A MATRIX THAT
DEPEN ON THE PILOT PARAMETERS, P(I).  BKFIJ COMPUTES
THE DETERMINANT OF THE A MATRIX (AS A HOMOUNGOU POLYNOMIAL
IN S).  POLYRT FINDS ROOTS OF POLYNOMIALS.

COMMON COMON(4400)
COMMON / S / A(12,12,11),DUM(15),B(12,12,11),DUM1(16)
INTEGER AORDER(12,12),BORDER(12,12)
DIMENSION P(5),ACOST(8)
EQUIVALENCE (AORDER,COMON(  1))
EQUIVALENCE (BORDER,COMON( 142))
EQUIVALENCE (KNTISE,COMON ( 514))
EQUIVALENCE (P ,COMON ( 780))
EQUIVALENCE (TR ,COMON(4345))
EQUIVALENCE (ACOST ,COMON( 695))
ENTRY BLANKI

USER SUPPLIED BLANKI INPUT FOLLOWS THIS CARD
ENTRY COSTI

USER SUPPLIED COST FUNCTION EQUATIONS COSTI FOLLOW THIS CARD, IF DESIRED
ENTRY FRATEI

USER SUPPLIED PILOT RATING EQUATION FRATEI FOLLOWS THIS CARD, IF DESIRED
RETURN
RETURN
END
2.3 McPilot Deck Structure

Figure 3 shows the make-up of the deck to run a case on the McPilot program. The control cards, which vary from installation to installation are not detailed. The End-of-Record and End-of-Information control cards, consist of the digits 7, 8, 9 and 6, 7, 8, 9 respectively multi-punched in card column 1. All cards shown are required to run a case.
3. PROGRAM OUTPUT

This section contains information regarding the output generated by the McPilot program. Depending on the control cards used, a FORTRAN listing of the blank subroutine is usually listed first along with compilation information where applicable. This is followed by output as per example program output in Section 3.2 and detailed in the following sections. Portions of the output are optional per user input indicator cards.

3.1 Output Sequence

Output may be divided into the eleven groups listed below and explained further in the following subsections.

(a) Listing of symbolic input cards.
(b) Basic airframe characteristics data about the longitudinal or lateral axis (INIAIR = 0).
(c) Pryden relationship gust calculations.
(d) Initial check of input data with the A and B matrices evaluated at the initial parameter values.
(e) Transfer function initial stabilization (MSHIP = 0).
(f) Output of the optimization data at the initial parameter values.
(g) Output of the optimization data at the minimum cost function values along each gradient line of the search.
(h) Calculated pilot rating.
(i) Frequency response calculations.
(j) Time response calculations.
(k) Root locus calculations (JLOCUS > 0).

3.1.1 Symbolic Input Listing – All of the symbolic input inserted by the user is listed right after the compilation print. This makes it convenient for the user to quickly scan the data input for errors and to find data to be changed for additional runs.

3.1.2 Airframe Characteristics – If INIAIR = 0, either the longitudinal (ININD = 0) or lateral (ININD = 1) aerodynamic characteristics are printed. This consists of:

(a) Aerodynamic characteristic input data.
(b) Dimensional stability derivatives.

If in addition the user wants the airframe characteristics (ININDAP = 0) the following is computed and printed:
3.1.3 Gust Calculations - If INDGST = 1, the following results of the Dryden relationship gust calculations are listed:

(a) Scale length \( l_w \), \( l_u \), \( l_v \).

(b) Root mean square of the gust intensity \( u_w \), \( u_u \), \( u_v \).

3.1.4 Initial Optimization Data - This is a printout of input data, but arranged in a more understandable manner. It consists of:

(a) Expanded A and B matrices printed with 4 rows and 12 columns per page.

(b) Expanded model numerator and denominator transfer functions for each model.

(c) Number of ISE terms in the cost function.

(d) Data values for each ISE term.

(e) Number of variable parameters.

(f) Data values for each parameter including their initial input values.

(g) Number of equations.

(h) Method of ending optimization (tolerances on parameters, ISE, or both).

3.1.5 Initial Stabilization - When MSKIP = 0, McPilot program goes to the MIN subroutine. This subroutine checks on the stability of the transfer function with the initial values of the optimization parameters. If the system is not stable, MINI changes the optimization parameters in such a manner as to drive any positive real roots, or positive real parts of a complex root, negative. The printout consists of the values of the most positive root and the optimization parameters for each iteration until stable starting values are obtained, 200 iterations have been completed, or creep step (change in optimization parameters) is less than the original value/100. This last condition implies that the predominate unstable root is unchanged for a local change in the values of the optimization parameters. This is usually indicative of an error in the system set up.
3.1.6 Initial Optimization Output – This is output of the NISER sub-
program of McPilot. The quantities printed are:
(a) KOGNT: \( (i-j) \) where \( i \) is the number of times the gradient of the
cost function has been determined, and \( j \) is the total number of
cost function terms which have been calculated. Initial point
is 0-1.
(b) JUMP: The initial value of JUMP,
(c) The value of each cost term and the scale factor used to make
model and system steady states equal. Scale factors are intern-
ationally calculated.
(d) TOTISE: The total cost function value.
(e) ALP: The interval size along the gradient line at the last step
in the optimization procedure. Initial value is 0.
(f) P: The initial values of the optimization parameters.
(g) DICOSS: The direction cosines of the gradient line. Initial
values 0.
(u) The system transfer function for each ISE cost term and the roots
of each system transfer function.

3.1.7 Optimization Output – The output of this section is identical to
the previous section, except that it is the values found at the minimum point
along the gradient search. This portion can be suppressed by use of KTIME
(see Section 2.1.2.2).

3.1.8 Pilot Rating – The pilot rating, calculated using the equation(s)
supplied by the user in blank subroutine (PRATE), is printed.

3.1.9 Frequency Response – If NISEAP \( \geq 1 \), the frequency response is
calculated and printed. The output consists of:
(a) ISE term, requested per input card ISEAUX (see 2.1.2.6), and the
numerator and denominator order and coefficients.
(b) For each input frequency (rad/sec) the program prints real part,
imaginary part, magnitude, magnitude (dB), phase angle.

3.1.10 Time Response Calculations – If NISEAP \( \geq 1 \), the time response
is calculated using unit step input (INDSTR = 0), or unit impulse input
(INDSTR = 1), and the following is printed out:
(a) ISE term requested, type of input (unit step or unit impulse),
numerator and denominator orders and coefficients.
(b) For each time (sec) the program prints the input variable and output variable.
(c) Printer-plot of input versus time.
(d) Printer-plot of output versus time.

3.1.1 Root Locus Calculations - if JLOCUS ≥ 1, the auxiliary root locus calculations are performed and the results printed as follows:
(a) Contents of the A matrix for the open loop poles polynomial — JLOCUS set to zero. JLOCUS is the root locus parameter.
(b) Contents of the A matrix for the characteristic equation — JLOCUS set to one.
(c) Open loop pole and zero polynomials, and characteristic equation coefficients.
(d) Heading which lists the system order, number of open loop poles and zeros, location of open poles and zeros, and root locus computation date that was input.
(e) Root locus points and corresponding gains, listed 3 branches per page. Each branch begins on an open loop pole within the computation area or at the boundary, and ends on an open loop zero or at the boundary. If there are more open loop zeros than poles, branches begin on zeros and end on poles.
(f) Plot of the root locus over the computation area. Symbols used on the plot correspond to branch numbers as follows: A - Branch No. 1, B - Branch No. 2, ...., J - Branch No. 10. For systems with more than 10 branches, A is used again for Branch No. 11, etc.

3.2 Example Problem
An example problem, exercising the still air random tracking longitudinal mode with airframe characteristics computed, is detailed in this section. The block diagram, equations, and the computer printout are all shown. This example is for the F-4 flying at \( M = 1.3 \) at an altitude of 35,000 ft. The block diagram of the system is presented in Figure 4. The pilot parameters \( K_p, T_1 \) and \( T_2 \) were assigned the computer optimization symbols \( P(1), P(2), \) and \( P(3) \) respectively. The equations and subscript matrix array location for the control system and acceleration are detailed below. The airframe equations of motion, including the elevator deflection terms, are automatically inserted into the first 4 rows and 5 columns. The other equations
starting in the fifth row along with their numerical index location in the
A matrix (in parentheses under the equations) are:

\[ \begin{align*}
  (5) \quad 0_a + 0 &= 0_c \\
  (89) \quad (7) \quad \text{(3-B Matrix)}
\end{align*} \]

\[ \begin{align*}
  (6) \quad F(1) \quad [0.2P(2) S^2 + (0.2 - 2P(2))S - 2] \quad 0_e
  \\
  \quad (90 - \text{Inserted by FLANKE})
  \\
  + [0.2P(3) S^2 + (0.2 + 2P(3))S + 2] A = 0
  \\
  \quad (66 - \text{Inserted by FLANKE})
\end{align*} \]

\[ \begin{align*}
  (7) \quad 0.155^2 0 - (S + 1) A - (S + 1) B = 0 \\
  \quad (19) \quad (67) \quad (70)
\end{align*} \]

\[ \begin{align*}
  (8) \quad B - (0.05 S + 1) \delta e &= 0 \\
  \quad (80) \quad (56)
\end{align*} \]

\[ \begin{align*}
  (9) \quad -\frac{V_1^2}{S} \delta + \frac{V_2^2}{S} + (10^{-5} S + 1) N_3 &= 0
  \\
  \quad (21) \quad (33) \quad (105)
\end{align*} \]

There are nine equations with the variables being, in column order, \( U, \delta, a, b, \delta_e, A, B, \theta_e, \) and \( N_3. \) \( A \) and \( B \) are arbitrarily set ad variables to keep the equations simple. Equation 7 represents SAS on. The \( 10^{-5}S \) term is added to the \( N_3 \) term of equation 9 to increase the order of the transfer function denominator (see 4.0 Running Hints). A computer printout of this example follows.

---

**FIGURE 4**

F-4 Longitudinal Axis Random Tracking Model

---

Approved for Public Release
<table>
<thead>
<tr>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Model 5</th>
<th>Model 6</th>
<th>Model 7</th>
<th>Model 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
</tr>
<tr>
<td>2</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
</tr>
<tr>
<td>3</td>
<td>o</td>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td>u</td>
</tr>
<tr>
<td>4</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
</tr>
<tr>
<td>6</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
<td>P</td>
</tr>
<tr>
<td>7</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

2 The terms will be calculated for each set.

<table>
<thead>
<tr>
<th>Set</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.005</td>
<td>0.006</td>
<td>0.007</td>
<td>0.008</td>
</tr>
<tr>
<td>2</td>
<td>0.009</td>
<td>0.010</td>
<td>0.011</td>
<td>0.012</td>
</tr>
<tr>
<td>3</td>
<td>0.013</td>
<td>0.014</td>
<td>0.015</td>
<td>0.016</td>
</tr>
</tbody>
</table>

3 The result will be updated in operating set.

Initial values: 0.005, 0.006, 0.007, 0.008

Updated values: 0.009, 0.010, 0.011, 0.012

Initial values: 0.013, 0.014, 0.015, 0.016

Updated values: 0.017, 0.018, 0.019, 0.020
<table>
<thead>
<tr>
<th>TERM</th>
<th>SCALE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7777</td>
<td>2.5788</td>
</tr>
<tr>
<td>2.1675</td>
<td>2.5788</td>
</tr>
<tr>
<td>3.92345678-01</td>
<td>2.167588-01</td>
</tr>
<tr>
<td>3.23456789-01</td>
<td>2.167588-01</td>
</tr>
</tbody>
</table>

**FOR THE TERM 1**

<table>
<thead>
<tr>
<th>SYSTEM NUMBER</th>
<th>SYSTEM DESIGNATION</th>
<th>NUMERATOR ROOTS</th>
<th>DENOMINATOR ROOTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6479632567-01</td>
<td>2.57889012345</td>
<td>2.6479632567-01</td>
<td>3.56789012345</td>
</tr>
<tr>
<td>2.3456789012-02</td>
<td>3.6789012345</td>
<td>2.3456789012-02</td>
<td>3.6789012345</td>
</tr>
<tr>
<td>3.56789012345</td>
<td>4.6789012345</td>
<td>3.56789012345</td>
<td>4.6789012345</td>
</tr>
</tbody>
</table>

**FOR THE TERM 2**

<table>
<thead>
<tr>
<th>SYSTEM NUMBER</th>
<th>SYSTEM DESIGNATION</th>
<th>NUMERATOR ROOTS</th>
<th>DENOMINATOR ROOTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6489012345</td>
<td>3.789012345</td>
<td>2.6489012345</td>
<td>3.789012345</td>
</tr>
<tr>
<td>3.56789012345</td>
<td>4.6789012345</td>
<td>3.56789012345</td>
<td>4.6789012345</td>
</tr>
<tr>
<td>4.6789012345</td>
<td>5.789012345</td>
<td>4.6789012345</td>
<td>5.789012345</td>
</tr>
<tr>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
</tr>
<tr>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
</tr>
<tr>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
</tr>
<tr>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
</tr>
</tbody>
</table>

*Table for characteristic equation: Please refer to Table 1.*
<table>
<thead>
<tr>
<th>YEAR</th>
<th>REAL</th>
<th>LOAN</th>
<th>GAIN</th>
<th>LOCUM-REAL</th>
<th>LOAN</th>
<th>GAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12000</td>
<td>0</td>
<td>0</td>
<td>12000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>13000</td>
<td>0</td>
<td>0</td>
<td>13000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>14000</td>
<td>0</td>
<td>0</td>
<td>14000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>15000</td>
<td>0</td>
<td>0</td>
<td>15000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>16000</td>
<td>0</td>
<td>0</td>
<td>16000</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Incl. Value: 12,0000
Gain: 2,000
In the course of running McPilot to get the pilot rating correlations, several program characteristics were discovered that affect the program operation. These are outlined here to aid the McPilot user and caution him on the applicability of some answers obtained.

A slowly changing or shallow minimum cost function, or local minimum can be encountered which makes it very difficult to find the actual minimum. It is sometimes necessary to experiment with different starting values of the pilot parameters to assure arriving at the true cost function minimum. Usual starting values were a small pilot gain \(10^{-4}\) and zero lead and lag. If the system optimized quickly in the region of the initial pilot parameter values, it was usually advisable to start at another point, say with a gain of 1 or 10, to determine if the program optimized to the same minimum value. If it does not, there may be local minimums in the cost function, or a shallow cost region exists. It sometimes helped to tighten the tolerances on \(P(N)\) and \(I(SE)\). A rule of thumb is that the tolerance should be about \(10^{-3}\) of the parameter value. Tolerance values nominally used for the still air random tracking task were 0.05 for gain, 0.005 for lead and lag, and 0.01 for the cost function. Time response, frequency response, and root locus analyses were used to assure being in the minimum region. If a cost function describes a trough shaped surface, it can take a long time to reach the minimum using the steepest descent technique, and very close tolerances must be specified.

For a given system, it is better to have a small matrix array with high order terms, than a large matrix array with low order terms. This reduces the running time for a case. When continuing a case that ran out of time before optimizing, reduce DEL to about 0.1 (usually start a case with DEL = 1.0).

High frequency terms in the control system did not affect the results of McPilot. Eliminating them did, however, reduce the running time. Arbitrarily, all terms having a frequency of 20 rad/sec or greater and first order time constant terms of 0.025 seconds or less were eliminated.

It is advisable, when analyzing a complex system, to start with a simple version and progress to the complex system. In this way, the region where the complex system optimum occurs can be predicted, and the complex system may be checked against the simplified one.
Occasionally one of the optimization parameters is driven across its boundary value. If at the same time the system is driven unstable, it could take a long time for McPilot to drive the parameter back into the desired region. The most common occurrence of this phenomenon is when a gain boundary is set at zero, and negative gains are unstable. The situation may be alleviated by inserting a statement in BLANKI that if P(N) exceeds a boundary, set the parameter equal to the boundary. This statement must occur in the initial segment of BLANKI prior to the point where the parameter is first used to compute a matrix array value. This disrupts the slope computation in the optimization loop; however, the overall effect is often to reduce the run time for a case.

The method of Krasovskii requires that the numerator of the Laplace transform of the response must be of less order than the denominator. If this condition is not met, McPilot will exit on an error and print a message to that effect. This situation occurred frequently for the \( N_2 \) terms of the cost functions used in the study. To overcome this problem, a high frequency filter \( \frac{1}{10^6 \cdot s + 1} \) term was introduced in the \( N_2 \) equation. This term introduces an extraneous root which is of no consequence in the pilot rating correlation because the root magnitude is so large. The example problem (Section 3.2) used this technique.
The following sets of equations were developed in this study by correlating
conventional aircraft handling qualities data based on system models for specified
flight tasks. For a complete description of the correlations, the tasks per-
formed, and the data used see Volume I. Unless otherwise specified, quasi-linear
pilot models are required for these correlations.

Task: Still Air Random Tracking - Longitudinal Axis
Input Transient Analog (Laplace transform):
\[ \Theta_C = 1 \] (an impulse)
Cost Function:
\[ J = \sigma_{error}^2 + 10^{-4} \sigma_y^2 \]
Pilot Rating Formula:
\[ PR = 3.9 \sigma_{error} + 0.3 \sigma_y + 8.5 \frac{1}{\sigma_y} + 1.6 |\dot{\epsilon}_1| \]
Note: \( \sigma_{error} \) = r.m.s. pitch error between system and model where
\[ \frac{\sigma_{error}}{\sigma_C} = \frac{-11.68^2 + 1835 + 331}{5^2 + 19.72^2 + 104.62^2 + 4735 + 111} \]

Task: Rough Air Tracking - Longitudinal Axis
Input Transient Analog (Laplace transform):
\[ \frac{\omega}{\omega_C} \frac{12.25\psi}{561}{1000} = \frac{5^2}{1000} \frac{\psi}{\psi_C} \]
Cost Function:
\[ J = \sigma_{error}^2 + \sigma_y^2 \]
Pilot Rating Formula:
\[ PR = 1 + 2.5 |\dot{\epsilon}_1| + 7.8 \frac{\sigma}{\sigma_y} + 11.5 \sigma_y \]

Task: Rough Air Tracking - Lateral Axis
Input Transient Analog (Laplace transform):
\[ \frac{\psi}{\omega_C} \frac{(s + 20.32)}{(s + 34.2)} \frac{3^2 + 3/13.8 s + .348}{s^2 + 1.138 s + .348} \]
Cost Function:
\[ J = 75 \sigma_y + 3.25 (1-e^{-7.7T}) \]
Pilot Rating Formula:
\[ PR = 1 + 7.1 \sigma_y + 3.1 (1-e^{-7.7T}) \]
Task: Still Air Random Tracking – Lateral Axis
Input Transient Analog (Laplace transform):
\[ \Phi_s = \frac{1.06}{S + 1} \]
Cost Function:
\[ J = 2\phi_s + (1-e^{-0.77T_l}) \]
Pilot Rating Formula:
\[ \Phi_t = 1 + 20X + 4.68(1-e^{-0.77T_l}) \]
\[ X = \phi_s - 0.6, \quad \phi_s > 0.6 \]
\[ = 0, \quad \phi_s < 0.6 \]
Task: Step Input - Longitudinal Tracking
Note: The Dual Mode Pilot developed in Volume 1 was used as the pilot model for this correlation. Care should be exercised using this pilot model as it is not substantiated for this task.
Input (Laplace transform):
\[ \theta_s = \frac{1}{S} \text{ (a step)} \]
Cost Function:
\[ J = 15E_0 + 10^{-4} ISN_{N_2} \]
Pilot Rating Formula:
\[ \Phi_t = 3.2 \sqrt{ISE_0} + 0.076 \sqrt{ISN_{N_2}} + 4.5 \frac{1}{\sqrt{ISE_0}} + 6.8 |\delta_1| \]
Note: ISE_0 is the ISE of the difference between the system pitch response and a model response where
\[ \theta_{model} = \frac{1.23x10^5}{S^5} - 17.05S^4 - 1526S^3 - 1.82x10^5S^2 - 2.8x10^3S - 3.43x10^4 - 3.4x10^4 - 1.54x10^3 \]
-.0329S^6 - 1.42S^5 - 15.4S^4 - 261S^3 - 871S^2 - 2.8x10^4S - 5.21x10^5S - 4.79x10^4S - 1.54x10^5
The McPilot program is divided into five primary level overlays to reduce the required memory. During execution, the overlays are called into memory and executed as requested. Each overlay is numbered with an ordered pair of numbers (1,1). 1 denotes a primary level overlay and J the secondary level. The initial or main overlay which always remains in memory has levels (0,0). Synopsis of subprograms in the main overlay is shown, and is followed by a complete listing of the McPilot program.

**MCPILOT**
Initializes labeled common blocks to zero.

**OVLAY**
Calls in the data reading overlay or the various execution overlays.

**CLASS**
Writes classifications on output pages.

**BLOCK DATA**
Is the directory of input symbolic names.

**BLOCK DATA**
Is an analogue of the input directory to permit filling the A and B matrices directly.

**FUNCTION**
Used to place data in four-dimensional array.

**BLANK**
Is a subroutine, partially written by the user, that makes any A or B element of the system equation \( Ax = By \) a function of the parameters to be optimized.

**Synopsis of subprograms in the primary level overlay (1,0):**

**INPUT**
Drives the reading routine; gives non-zero nominal values to selected constants; relocates data from input arrays to the A and B matrices.

**DIPLAC**
Right justifies a number so that displacement (col. 67-72) on an input card may be punched anywhere in the field.

**READA**
Provides a general method of reading a variable field data card. Data may be read into symbolic locations in memory.

**PACKBD**
Packs BCD words into six character words

**PACKR**
Packs BCD characters into words.

**READXI**
Converts free BCD to octal, floating point, and integer numbers.

**DSEEKH**
Provides a method of searching the directory to find the common subscript corresponding to a BCD argument.

**SVI**
Provides a method of placing a number of variables and other arrays into one composite array.

---

67

Approved for Public Release
Synopsis of subprograms in the primary level overlay (2,0)

AERO
Is the main program of the AERO overlay; it calls the subroutines demanded by the user-input control indicators.

IVXIV
Calculates the dimensional stability derivatives for longitudinal motions.

STFIX
Drives the analysis of the stick-fixed derivatives of the subject airplane/control system.

FILLIT
Initializes a 4 x 4 matrix with polynomial terms and fills in the variables. Longitudinal mode.

DEVAL
Evaluates the determinant of a 4 x 4 matrix with polynomial terms.

DMELR
Determines the roots of a polynomial by an iterative method.

IIKJ
Calculates the dimensional stability derivatives for lateral motions and performs an analysis of the behavior of the subject airplane/control system.

GUST
Calculates and inserts into the matrix the equations for gust for either lateral or longitudinal mode.

FILX3
Initializes a 3 x 3 matrix with polynomial terms and fills in the variables.

Synopsis of subprograms in the primary level overlay (3,0)

MISER
Is the controlling program of the MISER overlay.

WRITE1
Prints all output except diagnostic messages.

MINIMI
Finds by golden section the point at which ISE is a minimum along the gradient.

GRAD1
Calculates the gradient at a point P.

ISE
Calculates the total ISE at a point P.

SKRUJ
Calculates the determinant of a matrix whose elements are polynomials in S.

POLYRT
Calculates the real and complex roots of a real polynomial.

MPBY
Multiplies two polynomials together.

PSUB
Subtracts one polynomial from another.

POLFEO
Multiplies any number of polynomials together.

SIMQ
Solves a set of simultaneous equations.

MINI
Used to find a set of pilot parameters for which the system is stable.

COST
Determines the value of the largest positive part of the system Eigenvalues.

68
Synopsis of subprograms in primary level overlay (4,0)

AUXIL  Calculates the steady-state frequency response of a transfer function input as a numerator/denominator polynomial ratio.
TIMRP  Calculates time response for a transfer function input.
POUCUT AND INITUP  Determines numerically the solution of an arbitrary set of simultaneous ordinary differential equations.
FLOTR  A point plot routine which plots a graph of one or more curves from given sets of rectangular coordinates.
PACK  Used to exchange bits or bytes from one word to another.
DISCOPT  Provides for linear interpolation of functions of one or two independent variables.

Synopsis of subprograms in primary level overlay (5,0)

MATLOC  Used in conjunction with SKBJI with logic to set specified elements to zero and one thus finding the open loop poles and zeroes which are passed to RTLCSV.
SKBJI  Calculates the determinant of a matrix whose elements are polynomials in \( s \).
PSUB  Subtracts one polynomial from another.
FLIPPO  Flips over an array of \( N \) values such that \( \text{ARRAY}(N) = \text{ARRAY}(1) \), etc.
RPLLOT  Generates a root locus printer plot.
RTLLOC  Does bulk of computation work to generate root locus, finds gain, etc.
BLOCK DATA  Initializes variable names to given constants.
ARCTAN  Used to calculate arctangent.
POLYRT  Calculates the real and complex roots of a real polynomial.
RTLCSV  Control subroutine for generating the plots, lists of points, etc.

Note: Many variables are passed from one subroutine to another by blank common and to save storage many variables are equivalenced on top of others that are no longer needed.
OVERLAY (MCPILOT,0,0)

PROGRAM MCPILOT (INPLT,CLTPLT,TAPS=INPUT,TAP6=OUTPUT)
COMMUN CMNTM(44CC)
COMMUN / 4 / LOCUTT,NAME(20S),LOC(200)
COMMUN / 5 / TBSLL(320C)
COMMUN / 6 / DUMMY(31C)
00 14, I=1,44CC
14 CMNTM(I)=C
01 15 T=I,320C
19 TBSLL(I)=C
03 16 T=I,31C
10 DUMMY(I)=O.
CALL OVERLAY (7LMCPILOT,1,0,6HRECALL)
END
SUBROUTINE OVERLAY
COMMON DUM(747), LLINK
GO TO (1, 2, 3, 4, 5), LLINK
1 CALL OVERLAY (7LMCPILOT, 1, 0.6, RECALL)
2 CALL OVERLAY (7LMCPILOT, 2, 0.5, SHRECALL)
3 CALL OVERLAY (7LMCPILOT, 3, 0.6, SHRECALL)
4 CALL OVERLAY (7LMCPILOT, 4, 0.6, SHRECALL)
5 CALL OVERLAY (7LMCPILOT, 5, 0.6, SHRECALL)
END
SUBROUTINE CLASS(I)

CLASSIFICATION AT RHYTHM OF PAGE AND AT TOP OF NEXT PAGE

8774 FORMAT(1H2,3I5,X,GARAGE),15X/3(15X,*GARAGE),15X/)
8780 FORMAT(' ')

8781 FORMAT(1H2,3I5,X,CONFIDENTIAL),15X/3(15X,*CONFIDENTIAL),15X/)
4282 FORMAT(1H2,3I5,X,SECRET),15X/3(15X,*SECRET),15X/)
4283 FORMAT(1H2,3I5,X,TOP SECRET),15X/3(15X,*TOP SECRET),15X/)
3244 FORMAT(1H2,3I5,X,PRIVATE),15X/3(15X,*PRIVATE),15X/)
3285 FORMAT(1H2,3I5,X,PROPRIETARY),15X/3(15X,*PROPRIETARY),15X/)

7779 FORMAT(*1,15X,X,WRITE(6,6279))
7811 FORMAT(*1,315X,X,WRITE(6,6279))
7828 FORMAT(*1,315X,X,WRITE(6,6279))
7833 FORMAT(*1,315X,X,WRITE(6,6279))
7844 FORMAT(*1,315X,X,WRITE(6,6279))
7855 FORMAT(*1,315X,X,WRITE(6,6279))

RETURN

ENTRY CLASS

CLASSIFICATION AT TOP OF PAGE ONLY

7779 FORMAT(*1,315X,X,WRITE(6,6279))
7811 FORMAT(*1,315X,X,WRITE(6,6279))
7828 FORMAT(*1,315X,X,WRITE(6,6279))
7833 FORMAT(*1,315X,X,WRITE(6,6279))
7844 FORMAT(*1,315X,X,WRITE(6,6279))
7855 FORMAT(*1,315X,X,WRITE(6,6279))

RETURN

ENTRY CLASSB

CLASSIFICATION AT BOTTOM OF PAGE ONLY

7779 FORMAT(*1,315X,X,WRITE(6,6279))
7811 FORMAT(*1,315X,X,WRITE(6,6279))
7828 FORMAT(*1,315X,X,WRITE(6,6279))
7833 FORMAT(*1,315X,X,WRITE(6,6279))
7844 FORMAT(*1,315X,X,WRITE(6,6279))
7855 FORMAT(*1,315X,X,WRITE(6,6279))

RETURN

END

72

Approved for Public Release
BLOCK DATA CATA
COMMON / 3 / MAXT, STABLE(2), LUCS(2)
DATA MAXT / 2 /
DATA (STABLE(1), I=1, 2) /
DATA (LUCS(1), I=1, 2) /
*1, 1600 /
END
FUNCTION NEW(I,J,K,L)
DATA IDIM, JDIM, KDIM / 12, 10, 11 /
NEW = 1+10*K+J+1+JDIM*(K-1)+KDIM*(L-1))
RETURN
END
CONTRAILS

SUBROUTINE BLANK

C THIS SUBROUTINE IS FILLED IN BY USER, WHO MAKES THE A AND/or B
C MATRICES FUNCTION OF THE P(i)*F
C SIMULTANEOUSLY THE USER MAKES THE AORDER AND BORDER MATRICES
C COMPATIBLE WITH THE A AND B MATRICES RESPECTIVELY
C
C BLANK COMPUTES THOSE ELEMENTS OF THE A MATRIX THAT
C DEPENDING ON THE PILOT PARAMETERS, P(i), SIGMAJ COMPUTES
C THE DETERMINANT OF THE A MATRIX IAS A HOMOGENEOUS POLYNOMIAL
C IN 51. POLYAT FINDS ROOTS OF POLYNOMIALS.
C
COMMON CC(DC(DC)
COMMON /1/ , / A(12,1,11), DUM(15,1), DUM(12,11), DUM(1,16)
INTEGER AORDER(12,12), BORDER(12,12)
COMMON /2/ , / P(5), ACOST(11)
COMMON /3/ , / EQUVAL(ACORDER,COM(1))
COMMON /4/ , / EQUVAL(EKNTISE,COM(1))
COMMON /5/ , / EQUVAL(EQPR,COM(1))
COMMON /6/ , / EQUVAL(EKONT,COM(1))
ENTRY BLANK)

C USER SUPPLIED BLANK INPLT FollowS THIS CARD

INPUT KNTISE. NEQ.0.0 GO TO 20

D 12 1, 1, 1
AORDER(1,1)=1
AORDER(1,2)=1
AORDER(1,3)=1
AORDER(1,4)=1
AORDER(1,5)=1
AORDER(1,6)=1
AORDER(1,7)=2
AORDER(1,8)=2
TAU=2

3j A5,3,1=-2.0P(1)
A6,3,2=-1.0P(1)
A6,4,2=TAU(2),P(2)
A6,5,2=TAU(2),P(2)
A6,6,2=TAU(2),P(2)
A6,7,2=TAU(2),P(2)
A6,8,2=TAU(2),P(2)
RETURN
END
ENTRY CONT

C USER SUPPLIED COST FUNCTION EQUATIONS FOLLOW THIS CARD, IF DESIRED

C USER SUPPLIED PILOT RATING EQUATION FOLLOWS THIS CARD, IF DESIRED
C SIGMA = ACOST(1)*p, 5
SIGMA=ACOST(2)*P, 5
PHE1=ATANP(2)-ATANP(3)

76
PHII = ABS(PHI(1))
IF (P(1) .LT. 0.0) PHI1 = 0.
PHI1 = 3.4 * SIG1 + 0.2 * SIGNZ + 4.6 / SIGNZ + 1.6 * PHI1
RETURN
END
OVERLAY(MPLOT. 150)
PROGRAM INPLT
COMMON CUMEN(4600)
COMMON CUMEN(4600)
COMMON / SA(I399J, A8(1601))
EQUIVALENCE (A MIPFR, COMMON 11)
EQUIVALENCE (ORDER, COMMON 1451)
EQUIVALENCE (ORDER, COMMON 289)
EQUIVALENCE (ORDER, COMMON 297)
EQUIVALENCE (HYOY, COMMON 449)
EQUIVALENCE (MPSY, COMMON 409)
EQUIVALENCE (MMAX, COMMON 507)
EQUIVALENCE (MODELU, COMMON 519)
EQUIVALENCE (MODELU, COMMON 671)
EQUIVALENCE (INCA, COMMON 743)
EQUIVALENCE (ILIAK, COMMON 748)
EQUIVALENCE (RINPUT; COMMON 1101)
EQUIVALENCE (ATINPUT, COMMON 1711)
EQUIVALENCE (RTIME, COMMON 4176)
EQUIVALENCE (RINPUT, COMMON 4236)
EQUIVALENCE (OEGAS, COMMON 4296)
EQUIVALENCE (NTIME, COMMON 4321)
EQUIVALENCE (ANREA, COMMON 4331)
EQUIVALENCE (ACASE, COMMON 4344)
EQUIVALENCE (KGAH, COMMON 450)
EQUIVALENCE (DELTL, COMMON 455)
EQUIVALENCE (TOL, COMMON 4362)
EQUIVALENCE (ICLUSE, COMMON 4366)
EQUIVALENCE (CUST, COMMON 4371)
EQUIVALENCE (H, H41, K(4, A4, A4)
INTEGER BORDER
DIMENSION (12, 12), BORDER (12, 12), RINPUT (11)
DIMENSION (RINPUT, 12), MODELU (8, COMMON 11)
DIMENSION (12, 12), BORDER (12, 12), RINPUT (11)
DIMENSION (12, 12, 211), RTIMES (201), RINPUT (60)
DIMENSION (RINPUT, 12), ICURST (4), TOL (4), DELTA (4), KGAH (4)
DIMENSION (N6150) UNSET (4)
DIMENSION ACOST (1)
REAL MODLU, MODELU
INTEGER BORDER
DO 14 I = 1, 12
DO 14 J = 1, 12
ABORD(I, J) = -1
14 ABORD(I, J) = -1
DO 46 N = 1, 8
46 MODLU(N) = -1
46 N = 8
46 MODLU(N) = -1
N = 8
TPFR = 0
LMEGAS = 21.521
LMEGAS = 31.528
LMEGAS = 31.51
MEGAST(4)=15
MEGAST(5)=2
MEGAST(6)=4
MEGAST(7)=5
MEGAST(8)=7
MEGAST(9)=4
MEGAST(10)=4.5
MEGAST(11)=2
MEGAST(12)=7.5
MEGAST(13)=2
MEGAST(14)=4.5
MEGAST(15)=4
MEGAST(16)=4.5
MEGAST(17)=3.
MEGAST(18)=5.5
MEGAST(19)=6
MEGAST(20)=8
MEGAST(21)=10
NHEO=21
MAX=1
NITE=2
RTII(1)=C.
RTII(2)=2.
RTII(3)=5.
RINPUT(1)=1.
RINPUT(2)=1.
RINPUT(3)=1.
JG 10 M=1.4
KATT=NH=2
TL(N)=CC1
NLT(N)=125
NNDY(4)=1=12.
NNDY(4)=3=1.
NNDY(4)=4=1.
10 CALL READ(INPERR)
IF (INPERR .LT. C) GO TO 15
IF (INPERR .GT. C) INPERR
15 FORMAT(A16.)
IF (INPERR .LT. 1.42E-26) STOP
17 CONTINUE
I=11
C4M(N)=CUMMIN(7251)
DJ 5:J=1,12
DJ 5:J=1,12
IF (BORDER(I,J)=0) GO TO 11
DJ 6:K=I,10
4J=I,J
g(4J) = 3INPUT(IJ)
11 I=I+1
CONTINUE
H = 1
DO 2 J = 1, 12
DO 2 I = 1, 12
IF (A(I,J) .EQ. -1) GO TO 2
M(K) = A(I,J) + 1
2 CONTINUE
H = 1300
GO TO 100
100 DO 1 E = 1, 8
IF (M(J,E) .EQ. -1) GO TO 100
NE = M(J,E) + 1
GO TO 100
1 CONTINUE
H = 1340
DO 3 J = 1, 8
IF (G(J,E) .EQ. -1) GO TO 202
NE = G(J,E) + 1
GO TO 202
3 CONTINUE
IF (INSEK .EQ. 0) GO TO 4
CALL USER PROGRAM
I = 3
GO TO 905
4 I = 2
CALL AERO PROGRAM
C = CALL DISPLAY
H = THE NUMBER OF EQUATIONS IN THE SYSTEM—MAXIMUM 12
C = SYSTEMS ARE OF THE Form AX = B
C = ALL J, K TO A SYSTEM MATRIX WHERE T = 1—MAXIMUM 12
C = J = COLUMN—MAXIMUM 12
C = K = COEFFICIENT OF 5**((K-1)—
C = MAXIMUM 11
C = A(I,J,K) TELLS ORDER OF A(I,J,K) POLYNOMIAL ELEMENT
C = I = K—LEFT ORDER(I,J,K) = I FOR ALL K
C = ORDER(I,J,K) IS A SYSTEM MATRIX WHERE I,J,K ARE DEFINED AS ABOVE
C = ORDER(I,J,K) TELLS ORDER OF B(I,J,K) (ANALOGOUS TO A(I,J,K))
C = MOOELN(K) IS MODEL NUMERATOR WHERE K IS DEFINED AS ABOVE AND N
C = ORDER(I,K) IS THE MODEL NUMBER—MAXIMUM 5
C = MOOELN(K) IS MODEL DENUMERATOR WHERE K AND N ARE DEFINED AS ABOVE

80

Approved for Public Release
SUBROUTINE DPLAC(RAI,INC,BLANK)
DIMENSION RAI(1)
DIMENSION RC(1)
C
******************************************************************************
CPICK UP COLUMN 67 TO 72 AND
CRIGHT JUSTIFY FOR USE IN THE INPUT ROUTINE
C
C
******************************************************************************
DO 1 N=1,6
IF (RAI(N) .NE. BLANK) GO TO 2
IF (1.EQ.6) GO TO 7
DO 1 N=1,6
N=7-N
X(J)=X(N)
1 RAI(J)=RAI(N)
LO(1)=BLANK
INC=0
C
C
IF EXIT THROUGH GO STATEMENT 2
THEN A VALID NUMBER EXIST.
C
GO TO 6
2 CALL PACKR (RAI,RC ,6)
   CALL READ(3+RC,INC,1)
6 RETURN
END
CONTRAINTS

SUBROUTINE RECA(HPERR)
DIMENSION FI(16)
DIMENSION MSG(59) RA(59) *FU(2,28)
DIMENSION PA(51)
EQUIVALENCE (MSG(1),SYM) *(MSG(2),UP)
EQUIVALENCE (MSG(3),WM) *(MSG(50),INC)

C DATA SLTSYM /6H/
DATA RC(6),INT,OCI,TRA,CCMP,BLANK,POINT,EE/ "HASC",INT,HIIC,INTA,HH,HH,IM,IF/
C CARD NO. Z1 D 1764
C DATA SIP,MADSYM /6H/STOP
4 FORMAT (AC1,EXA3,1X,55A1,6A1)
5 FORMAT (AC1,EXA3,1X,55A1,1A)
6 FORMAT (EXA3,1X,55A1,1A)
7 FORMAT (E16.14) (55A1) 1000.00000000000.0
13) READ(5,4)SYM,IP,PA,RA1
13) IF (SYM.NE.SLP) GO TO 103
STOP
103 CONTINUE
CALL CIPLACE(RA,INC,BLAN)
IF (MSG(59).LE.0) J=57
WTR(16,14) (MSG(I))=I,J)
IF (IP.NE.RA) TRA GO TO 212
ID=1
RETURN
212 IF (RA(1).LE.BLANK) GO TO 803
IF (SYM.EQ.MADSYM) GO TO 100
IF (SYM.EQ.SLTSYM) GO TO 215
CALL XERCH (SYM,IND,ITARLE)
ITARLE.EQ.0,11 GU M .500
C CALL MACRO SECTION
C 213 IF (UP .LE. ACGO) GO TO 30C
CALL CALL(PACACD (RA,F1,JJ)
CJ TO 50C
C0MERIC TYPF
JF 1=C
C0MIAL TYPF
IF (UP .EQ.,).CT) IFI=1
C INTEGEL TYPF
IF (IP .EQ.,AINT) IFI=9
JU=0
JU=0
JU+1
31) JU=J+2
41) JU=J+1
51) JU=J+1
IF (I0 .LE. CT, 551 GO TO 218
IF (XAILQL.IE.C0MMA) GO TO 320
IF (PA(0).NE.BLANK) GO TO 315
82

Approved for Public Release
IF ( IFI .NE. 0 ) GO TO 340
JU = JU - 1
IF ( JU .LE. MQ ) GO TO 335
IF ( JU .LE. 0 ) GO TO 340
IF ( IFI .EQ. 2 ) GO TO 40
GO TO 40

IF ( IJU-MQ ) .LT. 345, 345
IFI = IFI - 1
CALL PACKR ( RA(4) , FO(I+JI) , M2+KQ+1 )
IF ( IFIQ .NE. 0 ) GO TO 42
JJ = JJ + 1
GO TO 310
CALL READ31 ( IFI , FI , IFI , JJ )
LURO = 0
IFI ( ING , GT , 0 ) HUR = -1
IIC = IIC + 1 NINC = IIC + 1
CALL SY ( ITAB , JJ , II , IFI )
CONTINUE
SLTSY = SY
GO TO 100
CONTINUE
WRITE (6,93)
6 FORMAT (26+ ER, COLUMNS 12 IS ALAUK)
WHELP = INPERP + 1
IF ( IFPK = 1 ) GO TO 100
"Approved for Public Release"
SUBROUTINE PACKRD (RA, F, JJ)
C DIMENSION RA(25), F(55)
C DECE 1, 5, RA) JJ
      FORMAT(III)
      KK = 7
      DO 10 I=1, JJ
      ENCOD 1 (F(I), I=1, JJ) (RA(I), L=KL, KK)
      KL = KL + 6
10    KK = KK + 6
      FORMAT(99)
      RETURN
      END

Approved for Public Release
SUBROUTINE PACK (I1, I2, N)

DIMENSION I1(N)

1 FORMAT (/1H1,'12.2H1','12.3H1')

K = 20 - N
END
SUBROUTINE DSEARCH(SYM,LOC,LCOM)
COMMUN CUMON(400)
EQUIVALENCE (DIRECT,CUMON(709))
EQUIVALENCE (LOCATE,CUMON(714))
COMMUN / 7 / MAXT,STARE(2),LOCST(2)
COMMUN / 4 / LCOUNT,ANAMB(200),LOCF(200)
DIMENSION DIRECT(5),LOCATE(5)
DO 50 I=1,LCOUNT
IF (SYM.NE. ANAMB(I)) GC TO 50
ICUM=0
LOC=LOCF(I)
RETURN
50 CONTINUE
DO 10 I=1,MAXT
IF (SYM.NE. STARE(I)) GC TO 100
ICUM=1
LOC=LOCST(I)
RETURN
100 CONTINUE
DO 200 I=1,5
IF (SYM.NE. DIRECT(I)) GC TO 200
ICUM = 0
LOC=LOCATE(I)
RETURN
200 CONTINUE
ICUM=1
LOC = 0
WRITE (6,6) SYM
RETURN
6 FORMAT(2HC=, A10, ISH= NOT IN DIRECTORY)
END
SUBROUTINE SW(ICM1,ICM2,ICM3)
COMMON DATA(4,CC)
COMMON ICM / 5 / TAILF(1)
DIMENSION A(I)
J=ISTART
K = IAMS(ICC)+1
GOTO (ICM*20+30,1,K
1) 100 I=1,N
   DATA(I)=A(I)
   J = J+1
   RETURN
2) 200 J = J+1
   TAILF(J)=A(J)
200 J = J+1
   RETURN
10 CONTINUE
   RETURN
END
PROGRAM AERO
C ACTS - ROUTS OF A/C LONITUDINAL TRANSFER FUNCTIONS
C
COMMUN / 5 / (132CO)
COMMUN COMMEN (44CO)
EQUIVALENCE (INDCLS, COMON (725))
EQUIVALENCE (INDAIR, COMON (743))
EQUIVALENCE (INDGST, COMON (726))
EQUIVALENCE (INDMOD, COMON (727))
EQUIVALENCE (TITLE, COMON (748))
EQUIVALENCE (TITLE, COMON (795))
DIMENSION TITLE(10)
C
IF(INDAIR .NE. O) GO TO 595
CALL CLASS (INDCLS)
WRITE(*,600C) TITLE
900 FORMAT (1H20X,10A6)
IF(INDMOD .NE. C) GO TO 100
CALL TVXIV
5) CONTINUE
IF(INDGST .GE. C) CALL GSL1
GO TO 995
100 CONTINUE
CALL ITIN3
GO TO 50
995 CONTINUE
ITINK = 2
CALL OVERLAY
END
COMMON COMM(144CC)
COMM / 5 / A(L12,12,1)

EQUIVALENCE (ARORDER, COMM(1 11))
EQUIVALENCE (RHM, COMM(754))
EQUIVALENCE (XS, COMM(755))
EQUIVALENCE (S, COMM(728))
EQUIVALENCE (CM, COMM(113))
EQUIVALENCE (MV, COMM(756))
EQUIVALENCE (CMC, COMM(736))
EQUIVALENCE (CMC, COMM(179))
EQUIVALENCE (CMC, COMM(731))
EQUIVALENCE (CM, COMM(1400))
EQUIVALENCE (CM, COMM(1401))
EQUIVALENCE (CM, COMM(1402))
EQUIVALENCE (CM, COMM(1403))

EQUIVALENCE (CM, COMM(1404))
EQUIVALENCE (CM, COMM(1405))
EQUIVALENCE (CM, COMM(1406))
EQUIVALENCE (CMC, COMM(1407))
EQUIVALENCE (CM, COMM(1408))
EQUIVALENCE (CM, COMM(1409))
EQUIVALENCE (CM, COMM(1410))
EQUIVALENCE (CM, COMM(1411))
EQUIVALENCE (CM, COMM(1412))
EQUIVALENCE (CM, COMM(1413))

EQUIVALENCE (CM, COMM(1414))
EQUIVALENCE (CM, COMM(1415))
EQUIVALENCE (CM, COMM(1416))
EQUIVALENCE (CM, COMM (1417))
EQUIVALENCE (CM, COMM(1476))
EQUIVALENCE (CM, COMM(1475))
EQUIVALENCE (CM, COMM(1474))
EQUIVALENCE (CM, COMM (1473))
EQUIVALENCE (CM, COMM(1480))

EQUIVALENCE (CM, COMM(1469))
EQUIVALENCE (CM, COMM(1468))
EQUIVALENCE (CM, COMM(1467))
EQUIVALENCE (CM, COMM(1466))
EQUIVALENCE (CM, COMM(1465))
EQUIVALENCE (CM, COMM(1464))
EQUIVALENCE (CM, COMM(1463))
EQUIVALENCE (CM, COMM(1462))
EQUIVALENCE (CM, COMM(1461))
EQUIVALENCE (CM, COMM(1460))

EQUIVALENCE (CM, COMM(1459))
EQUIVALENCE (CM, COMM(1458))
EQUIVALENCE (CM, COMM(1457))
EQUIVALENCE (CM, COMM(1456))
EQUIVALENCE (CM, COMM(1455))
EQUIVALENCE (CM, COMM(1454))
EQUIVALENCE (CM, COMM(1453))
EQUIVALENCE (CM, COMM(1452))
EQUIVALENCE (CM, COMM(1451))
EQUIVALENCE (CM, COMM(1450))
C THE CORE LOCATIONS USED FOR THE AERODYNAMIC
C COEFFICIENTS ARE ALSO IN THE MODULE OF ASKBJJ.
C A WORKING ARRAY IN THE MAIN Miser PROGRAM. THIS
C IS ALSO THE TEMP ARRAY OF SUBROUTINE SKRJ1.
C
WRITE(6,411C0)
411C0 WRITE(6,450C0)CX,A,CXAD,CXO,CXI,CXN,CXM,CZL,CZM,CZLH,CZAD,CZLH,
1 C2H,C2H1,CMA,CMAU,CMG,CMO,CMU,CMH,VS,LY,AA,ALT,G,MA,
2 HMI,WM,CMAP
25 CZERO=CXH+CXL+CMU+CHM+CMU+CMH
II1CEPD.GEQ.0.100 TO 57
M=SS**G
VS=H*ML**S
VSC=V**S**CMAP
G=5.9873**2
C CALCULATE DIMENSIONAL DERIVATIVES UNPRIMED
C
MU=VSC/IVCMU
MM=VSC/L.TY*CMH
MA=VSC/#.L.TY*CMA
MD=VSC*CMH/4.*.L.TY*CMO
MH=VSC/1.2*TY*CMO
XHV=VSC/1.2*MASS*CH
XH=VSC/1.2*MASS*CH
XDA=VSC/(1.2*MASS*CMR)
XAD=VSC/(1.4*MASS*CMR)
X=VSC/1.4*MASS*CMR
XD=VSC/(1.2*MASS*CMR)
ZU=VSC/MASS*CE
ZH=VSC/(1.2*MASS*CE)
ZD=VSC/(1.2*MASS*CE)
C =VSC/(1.4*MASS*CE)
ZL=VSC/(1.4*MASS*CE)
ZL=VSC/(1.2*MASS*CE)
C
A=AX-AA
SINA=SIN(AC/DIR)
COSA=COS(AC/DIR)
C
IF(AUON.EQ.0.)GO TO 40

92
WRITE DIMENSIONAL DERIVATIVES UNPRIMED

X=x/X
Z=z/Z
A=a/A

WRITE DIMENSIONAL DERIVATIVES PRIMED

X=x/X
Z=z/Z
A=a/A

CALCULATE DIMENSIONAL DERIVATIVES PRIMED

U=U+M*A+M*V*SINA
V=AH+ML+M/S
X=AP+Q*X+V*ML*SINA
M=AM+P*M
U=U+X*SINA
X=x+SIN
X=x+SIN
X=x+SIN
X=x+SIN
X=x+SIN
X=x+SIN
X=x+SIN
X=x+SIN
X=x+SIN
X=x+SIN

WRITE DIMENSIONAL DERIVATIVES PRIMED

CALCULATE AND PRINT LAW,DELTA/NZ, AND NZA

CONTINUE

Approved for Public Release
**Syntax Error**: The provided text contains syntactical errors and is not properly formatted. It seems to be a part of a programming or computational document, possibly related to control systems or data processing. However, without proper formatting or context, it's challenging to provide a meaningful translation or analysis.
A[(s, 1) = MC
RETURN
END

95

Approved for Public Release
COMMON COMCPN(40)
EQUIVALENCE (ZA ,COMCPN(1469))
EQUIVALENCE (ZU ,COMCPN(1473))
EQUIVALENCE (ANG ,COMCPN(1477))
EQUIVALENCE (M ,COMCPN(1478))
EQUIVALENCE (MU ,COMCPN(1479))
EQUIVALENCE (NT ,COMCPN(255))
EQUIVALENCE (INDCLS,COMCPN(725))

REAL MA,MO,MS,LA,LAM,LAN
DIMENSION RGTDRP(101),RODT(101)

DIMENSION RGDTN(101),RGTVD(101),DEN(11)
DATA 622.174 /
DATA 229.12320/

COMPLETE EQUATIONS

EVALUATE DENOMINATOR

CALL FILL111
CALL SETVALN,DEN
L=1/FA/V
N=1

CALCULATE ROOTS

CALL PHLDOPN,NK(DPREM),RODT(1)
WRITE(*,11)
DJ LG 1=1+K
IFABS(RGDTN(111)).LT.1.E-6)GO TO 0
WRITE(*,2)CGM,RGDTN(111), RODT(11)
DJ TO 9

3
WRITE(*,3)CGM,RGDTN(111)
DJ LG JT=1
NJT=1+RODT(111)
NGM=0
IFABS(RGDT(111)).LT.1.E-6)CGM1 GO TO 12
NGM=NGM1
J=1
K=1
CJ TO 13

12
IFABS(RGDT(111)).LT.1.E-6)CGM1 GO TO 13
CONTROLS

J=2
K=2
GU TO 14
15 IF ANS(R,GT1(2),LT,C,CC01) GO TO 14
WNC=WNC+1
J=3
L=3
GU TO 15
15 IF ANS(R,GT1(2),LT,C,CC01) GO TO 15
WNC=WNC+1
J=4
L=4
15 IF WNC=1140,16,26
C
C 1 PAIR OF COMPLEX ROUTS
C
15 WN=SQR(T(R, DCTA(J)++2*RGT1(J+2))
WNC=WNC/2
Z=RGT1(J)/WN
P=W(NP)/ABS(RGT1(J))
P=WNP/P
Y=-0.5275/PMT1(J)
T0=-5.3325/PMT1(J)
C2=12/PB
C10=T0/PB
OC2=1,1/C2
UC10=L1/C10
CN=Z/A/V*M=M+MA
WNPD=SQR(T(CON)
Y=WNL=Z/1(PA+2,2,LT-ZA1/CON
LSP=A5S(WNP-WN)
IF(DSP,GT,DP) IGT1 TO 18
C
C WRITE SHORT PERIOD
C
LWNL=L/L/WN
X=F16.2902/3,WN+WN
IF(LF16.2902/3,LT,C,CC01) GO TO 70
WN/L/WN**2/WN2
WRITE(F6.3CC01)WN2
GU TO 70
C
C WRITE LONG PERIOD
C
14 WRITE(F6.3CC01)WN+WN
22 IF(LT,C,CC) GU TO 22
X=F16.45CC01,P,T7,T15,PD,C2,C10,OC2,OC10
GU TO 40
22 X=F16.45CC01,P,T7,T15,PD,C2,C10,OC2,OC10
GU TO 40

97
Approved for Public Release
2. CALCULATE SHORT PERIOD

\[ \text{wNSP} = \text{wNSP} / \text{wNSP} \]
\[ \text{zSP} = \text{zSP} / \text{wNSP} \]
\[ \text{zSP} = \text{zSP} / \text{wNSP} \]
\[ \text{wNSP} = \text{wNSP} / \text{wNSP} \]
\[ \text{wNSP} = \text{wNSP} / \text{wNSP} \]

3. CALCULATE LONG PERIOD

\[ \text{wSP} = \text{wSP} / \text{wSP} \]
\[ \text{zSP} = \text{zSP} / \text{wSP} \]
\[ \text{wSP} = \text{wSP} / \text{wSP} \]
\[ \text{wSP} = \text{wSP} / \text{wSP} \]
\[ \text{wSP} = \text{wSP} / \text{wSP} \]

4. WRITE COEFFICIENTS

Approved for Public Release
WRITE(*,50) *SCECIDENT*1,1*1,M1

190 FORMAT(1I11,**ROOTS (COMPLEX FORM*)

200 FORMAT(1I11,1X,**SHORT PERIOD MODE**,/T5,**ETA** =*,E12.5,T5*,,**N** =*,

1 E17.5,1X,**RAD/SEC*//T5,** =*,E12.5,1X,**CYCLES*/SEC*)

300 FORMAT(**,T10.2,2X,**SQUARED**/**N**A* =*,E12.5)

500 FORMAT(1I11,1X,**LONG PERIOD MODE**,/T5,**ZETA** =*,E12.5,T5*,,**N** =*,

1 E17.5,1X,**RAD/SEC*//T5,** =*,E12.5,1X,**CYCLES*/SEC*)

400 FORMAT(1I11,**UNDAMPED PERIOD** =*,E12.5,16X,**TIME TO HALF AMP.** =*,

1 E12.5,14X,**TIME TO ONE TENTH AMP.** =*,E12.5,17X,**DAMPED PERIOD** =*,

2 E12.5,14X,**CYCLES TO HALF AMP.** =*,E12.5,12X,**CYCLES TO ONE TENTH AMP.** =*,

3 E12.5,14X,**TIME TO ONE TENTH AMP.** =*,E12.5,14X,**CYCLES TO ONE TENTH AMP.** =*,E12.5,14X,**CYCLES TO ONE TENTH AMP.** =*,E12.5)

401 FORMAT(1I11,**UNDAMPED PERIOD** =*,E12.5,16X,**TIME TO DOUBLE AMP.** =*,

1 E12.5,14X,**TIME TO TEN TIMES AMP.** =*,E12.5,17X,**DAMPED PERIOD** =*,

2 E12.5,12X,**CYCLES TO DOUBLE AMP.** =*,E12.5,12X,**CYCLES TO TEN TIMES AMP.** =*,

3 E12.5,14X,**TIME TO TEN TIMES AMP.** =*,E12.5,14X,**TIME TO TEN TIMES AMP.** =*,E12.5)

501 FORMAT(1I11,**CHARACTERISTIC EQUATION COEFFICIENTS**/*T5,3E15.6*//T5,*A
e =*,1PE12.5,5H S**5 B =*,1PE12.5,5H S**4 C =*,1PE12.5,5H S**3 D =*,

1 PE12.5,5H S**2 E =*,1PE12.5,5H S**1 F =*,1PE12.5,5H S**0)

949 RETURN

END

Approved for Public Release
SUBROUTINE FILL444C
C
C TO INITIALIZE A 4X4 DETERMINANT WITH POLYNOMIAL TERMS
C
IRC = POSITION OF FIRST TERM ON DIAGONAL
C
NORD = ARRAY OF ORDERS OF POLYNOMIAL TERMS
C
G = ARRAY OF COEFFICIENTS OF POLYNOMIAL TERMS
C
C COMMON GCOMN(444C)
C COMMON / S / D(12,12,11)
C EQUIVALENCE (IRC,GCMCN)
C EQUIVALENCE (X,GCMCN)
C EQUIVALENCE (XA,GCMCN)
C EQUIVALENCE (XSD,GCMCN)
C EQUIVALENCE (X,GCMCN)
C EQUIVALENCE (XL,GCMCN)
C EQUIVALENCE (ZG,GCMCN)
C EQUIVALENCE (ZA,GCMCN)
C EQUIVALENCE (ZAD,GCMCN)
C EQUIVALENCE (Z,GCMCN)
C EQUIVALENCE (ZL,GCMCN)
C EQUIVALENCE (ZM,GCMCN)
C EQUIVALENCE (MA,GCMCN)
C EQUIVALENCE (MAD,GCMCN)
C EQUIVALENCE (MD,GCMCN)
C EQUIVALENCE (M2,GCMCN)
C EQUIVALENCE (VL,GCMCN)
C EQUIVALENCE (V,GCMCN)
C EQUIVALENCE (GAM,GCMCN)
C EQUIVALENCE (GAM,NORD)
C EQUIVALENCE (LSA,GCMCN)
C EQUIVALENCE (LCDA,GCMCN)
C EQUIVALENCE (LCDS,GCMCN)
C EQUIVALENCE (LCDS,GCMCN)
C EQUIVALENCE (LGS,GCMCN)
C EQUIVALENCE (LGS,GCMCN)
C EQUIVALENCE (L, NORD)
C
C REN MH, MA, MAD, MD, 1.0
C INTEGER NORD(12,12,100)
C DATA IRC/57,255784/G/22,1274/
C DATA ORD/1,369,29,21,20,2,38,14,0,1/
C
C ZERO DETERMINANT
C
C DO 10 I=1,4
C DO 10 J=1,4
C 10 MOD(I,J)=1
C DO 10 K=1,11
C 10 L=1,K-1=0
C SET DIAGONAL TO 1.0
C
C DO 20 I=1,4
C 20 NORD(I,1)=C

100

Approved for Public Release
C
C FILL ARRAY WITH ORDERS OF POLYNOMIALS OF 4x4 DETERMINANT
C
1K=1RC+1
2K 30 1=1,4
14=1K+1
IC=1K-C-1
C1 30 1=1,4
IC=1K+C
IC=1K+C

C JKNH (P,IC)=JRD(I,J)
C
C FILL ARRAY WITH COEFFICIENTS OF POLYNOMIALS OF 4x4 DETERMINANT
C
SINC=SIN(GAMA/DIR)
COSG=COS(GAMA/DIR)
IR=IC
IC=1RC
IJ=IJ
ICI=ICI+1,C
ICI=ICI
I1=I1*COSG
I1=I1*SINA-XQ
ICI=ICI
I1=I1-XA
I1=I1+XQ
I1=I1+111
III=III
IC=ICI
ICI=ICI+1
ICI=ICI+1
I1=I1+SING
I1=I1+COSSA-ZO
ICI=ICI+1
I1=I1-ZA
I1=I1-ZA
I1=I1+111
III=III
IC=ICI
ICI=ICI+1
ICI=ICI
I1=I1-XU
ICI=ICI+1
III=III
I1=I1+111
ICI=ICI+1
I1=I1-MA
I1=I1-MA
I1=I1+111
III=III
IC=ICI
ICI=ICI+1
ICI=ICI

101

Approved for Public Release
C(1R, IC+2, 1) = V*C(150)
IC = IC+1
C(1R, IC, 1) = C.
C(1R, IC, 2) = L.C
RETURN
ENG
SUBROUTINE DETVAL(N,AN,DANS)
COMMON / 5 / A(12,12,11)
EQUIVALENCE (NUMO,GMCM(11))
COMMON GMCM(1400)

******************************************************************************
EVALUATION OF DETERMINANTS WITH POLYNOMIAL TERMS
******************************************************************************

NILC(I,J,K) = I X J X K DETERMINANT
NILC(I,J) = ORDER OF EACH TERM -- MAX. = 10TH ORDER
******************************************************************************

DIMENSION RESULT(81),ANS(81),R2(21),R3(31)
DIMENSION CANS(11)
DIMENSION NO(111,12)

******************************************************************************
ZERO THE ORDER OF THE ANSWER AND THE ANSWER
******************************************************************************

NA=0
GO TO 10 I=1,E1
1) ANS(I)=0.0

******************************************************************************
THE FOLLOWING FINDS THE PROPER TERMS OF THE DETERMINANT TO
MULTIPLY AND Keeps A RECORD OF THE SIGN USING ISGN1,ISGN2,ISGN3,ISGN4,
ISGN5,ISGN6,ISGN7,AND KSIGN.
******************************************************************************

!0) 900 J=1,4
KSIGN=0
IF(NILC(J,J,J)>0)GO TO 500
D1=MOD J=1,4
ISGN1=0
IF(NILC(2,J,J,2,J,J,J)>1)GO TO 300
IF(J=I)I2=I,CC,14
ISGN1=ISGN1+1
1* IA=NILC(J,J,J,NILC(2,J,J,J)+1
GO TO 400
1* R2(I,J)=Ec,C
NJ=NL(N1,1,1,1)+1
NJ=NL(N1,2,J,J)+1
CJ 18 1X=1,JX=1
DJ 18 JX=1,MO2
IND=1X+JX-1
1* R2(IND)=R2(IN1,1,1,1)+A(2,J,J,J)+R2(IND)
GO TO 700
K=1,4
IF(NILC(I,K,1,1,1))GO TO 700
ISGN2=0
IF(K=125,700,22
2) ISGN2=ISGN2+1

103
C++ 1.124, 750, 26
1.124N+1<550N+1
1.124N+1<550N+1
UD 7B 1U=1,15
1.124N+1<550N+1
UD YC IX=1,110
UD XJ=1,113
INC=1,JX=1
1.124N+1<550N+1
P(1,K)*A(3,K,JX)*R31N+1
UD 605 L=1,4
IF(NORD(4,L,E0,-1)) GOTO 160
ISGN=1
IF(=1135,40C,34)
ISGN=ISGN+1
26 IF(=1136,40C,36)
36 ISGN=ISGN+1
56 IF(=1137,40C,42)
76 ISGN=ISGN+1
57 IF(NORD(4,L)+1)
UD 44 L=1,17
RESULT(3)+C
401 NORD(4,L)+1
NL=NL+1
IF(J=3,1,110)
JF 42 IX=1,110
UD XJ=1,115
INC=1,JX=1
1.115 RESULT(3)(1X)*41(4,L,JX)*RESULT(1)
C
**** calculus the sign of the term using nulz 7.
C

C
K<1G<115G<115G<25G<115G<115
1.115 155 M=1,117
1.115 RESULT(3)+RESULT(1)
1.115 155 10=1,117
1.115 A=15G<1ANS(15G<1)+RESULT(1)
1.115 155 114 DELETED
1.115 CONTINUE
1.115 CONTINUE
1.115 CONTINUE
1.115 CONTINUE
C
C.
C
C
C
DO 910 I = 1, M
   J = N(I) + 1
910  DANS(I) = ANS(J)
RETURN
END
SUBROUTINE CMULRICOEF(N1,ROOT,NROOT)
C******************************************************************************
C POLYNOMIAL ROOT FINDER SUBROUTINE....
C ITERATIVE METHOD FOR POLYNOMIAL EQUATIONS....
C THIS METHOD APPROXIMATES THE FUNCTION F(X) BY A QUADRATIC
C WHICH MAY, IN GENERAL, HAVE COMPLEX COEFFICIENTS AND DOES NOT
C REQUIRE A KNOWLEDGE OF THE DERIVATIVE OF F(X) THOUGH
C THE FUNCTION F(X) MUST BE EVALUATED ONCE PER ITERATION....
C THIS SUBROUTINE FINDS REAL AND COMPLEX ROOTS OF A POLYNOMIAL
C WITH REAL COEFFICIENTS....
C USE OF MULER SUBROUTINE....
C 1. CALL CMULR (COEF,N1,ROOT,NROOT)....
C 2. COEF IS THE TAG OF THE ARRAY OF COEFFICIENTS.
C THE COEFFICIENTS MUST BE ORDERED FROM HIGHEST DEGREE
C TO LOWEST DEGREE.
C 3. N1 IS DEGREE OF THE POLYNOMIAL.
C 4. NROOT IS THE TAG OF THE ARRAY WHERE THE REAL PARTS
C OF THE COMPLEX ROOTS ARE STORED.
C 5. NROOT IS THE TAG OF THE ARRAY WHERE THE IMAGINARY
C PARTS OF THE COMPLEX ROOTS ARE STORED....
C ALL ARITHMETIC IS IN THE COMPLEX MODE....
C THEREFORE UNDERFLOW IS NORMAL FOR REAL ROOTS....
C MULTIPLE ROOTS DECREASES ACCURACY OF THIS SUBROUTINE....
C WHEN MULTIPLICITY IS FULK THE ACURACY DECREASES TO
C ABOUT TWO PLACES....
C DIMENSION (DEF111),ROOT(11),ROOT(11),COE(121)
N=0
I=N+1
1 I=I+1
2 IF (I>DEF111) GOTO 3
3 I=I+1
4 IF (DEF111) GOTO 5
5 I=I-1
6 GOTO 4
7 GOTO 1
8 I=I+1
9 N=N-1
10 GOTO 2
11 CMULRICOEF(I-1,N1)
12 AX=0.0
13

106

Approved for Public Release
IF (AB6ERD0711(N4)) 1, CE=5110.10, 51

GO TO 10

A6X=ALP4I
ALP4I=ALP4I
M=5
GO TO 99

G6P=TEM6
REM=TEM6
AX6=ALP2I
ALP2I=ALP2I
M=6
GO TO 95

REM=TEM6
REM=REM
AX6=ALP3I
ALP3I=ALP3I
L=9
M=3

TEM6=COE11
TEM6=0.0
DO 1 = 1, M1
TEM6=TEM6+AX6-TEM6*AX6
TEM6=TEM6+AX6*TEM6*AX6
TEM6=COE11
REM=TEM6
REM=REM

IF (N4+IC2, IC2, IC2)

IC2=10 I=1, N4
TEM6=REM-REM(11)
TEM6=REM+TEM6+TEM6*TEM6
TEM6=REM+TEM6+TEM6*TEM6
GO TO 11, 12, 13, 15, 23, 34, 1, M

CONTINUE
RETURN
END
SUBROUTINE IIIX3
COMMON CMCN1(44,62)
COMMON / S / KM12,12,111
EQUIVALENCE (AORDER,CMCN1.11)
EQUIVALENCE (SIMO,CMCN1.754)
EQUIVALENCE (S,CMCN1.755)
EQUIVALENCE (S,CMCN1.7281)
EQUIVALENCE (SNX,CMCN1.7301)
EQUIVALENCE (ASPAK,CMCN1.7321)
EQUIVALENCE (SIXX,CMCN1.14561)
EQUIVALENCE (IIIK8,CMCN1.14571)
EQUIVALENCE (SIXX2,CMCN1.14581)
EQUIVALENCE (ALPHA,CMCN1.14591)
EQUIVALENCE (S,CMCN1.7531)
EQUIVALENCE (ALITX,CMCN1.7551)
EQUIVALENCE (GANA,CMCN1.7571)
EQUIVALENCE (CIXB,CMCN1.14251)
EQUIVALENCE (CIXD,CMCN1.14261)
EQUIVALENCE (CIXP,CMCN1.14271)
EQUIVALENCE (CIXY,CMCN1.14281)
EQUIVALENCE (CIXYD,CMCN1.14291)
EQUIVALENCE (CIXYD2,CMCN1.14301)
EQUIVALENCE (CIXB,CMCN1.14311)
EQUIVALENCE (CIXBD,CMCN1.14321)
EQUIVALENCE (CIXDP,CMCN1.14331)
EQUIVALENCE (CIXDP2,CMCN1.14341)
EQUIVALENCE (CIXDR,CMCN1.14351)
EQUIVALENCE (CIXDL1,CMCN1.14361)
EQUIVALENCE (CIXDL2,CMCN1.14371)
EQUIVALENCE (CIXDL3,CMCN1.14381)
EQUIVALENCE (CIXDL4,CMCN1.14391)
EQUIVALENCE (CIXDL5,CMCN1.14401)
EQUIVALENCE (CMN10,CMCN1.14411)
EQUIVALENCE (CMN102,CMCN1.14421)
EQUIVALENCE (CMN103,CMCN1.14431)
EQUIVALENCE (CMN104,CMCN1.14441)
EQUIVALENCE (IGA,CMCN1.14451)
EQUIVALENCE (OD,CMCN1.14461)
EQUIVALENCE (OD2,CMCN1.14471)
EQUIVALENCE (OD3,CMCN1.14481)
EQUIVALENCE (ICLX,CMCN1.14491)
EQUIVALENCE (ICLU1,CMCN1.14501)
EQUIVALENCE (ICLU2,CMCN1.14511)
EQUIVALENCE (ILT,CMCN1.7561)
EQUIVALENCE (ICMU3,CMCN1.14541)
EQUIVALENCE (ICMU3,CMCN1.14551)
EQUIVALENCE (ICMU3,CMCN1.7251)
EQUIVALENCE (ICMU3,CMCN1.741)
EQUIVALENCE (ICMU3,CMCN1.726)
EQUIVALENCE (ICMU3,CMCN1.7951)
EQUIVALENCE (ICMU3,CMCN1.14411)
EQUIVALENCE (ICMU3,CMCN1.14591)
EQUIVALENCE (ICMU3,CMCN1.14589)
176 FORMATT(17,14) LATERAL DIRECTIONAL DENOMINATOR

C = ALP*ALP + ALP*ANP + ALP + ANP + (1.0 - Y/F)*Y + (1.0 - Y/F)*Y + Y/F

2.0 + ALP*ANP + ALP*ANP + (1.0 - Y/F)*Y + (1.0 - Y/F)*Y + Y/F

Y = (1.0 - Y/F)*Y + ALP*ANP + ALP*ANP + (1.0 - Y/F)*Y + (1.0 - Y/F)*Y + Y/F

E = (1.0 - Y/F)*Y + ALP*ANP + ALP*ANP + (1.0 - Y/F)*Y + (1.0 - Y/F)*Y + Y/F

178 CONTINUE

STOP

END
227 PER = XON / (40 * 5) RETURN (1, -ABS(S[Z]) * 21)
FOR = XON / (41 * 4)
TO1 = .692111 / (ABS(S[Z]) * 9)
TO2 = .703225 / (ABS(S[Z]) * 8)
C01 = TO1 / PER
C02 = TO2 / PER
C03 = 1.0 / C01
C04 = 1.0 / C02
IF01 = 122.5, 123.224
1144 WRITE (6, 114) RETR, TO1, TO2, PER, C01, C02, C03, C04
1154 FORMAT (/T2, 901, 901, 901, 901) RETR = *, E13.5,
13K1940 * 1000 = 1.3K1940 * 1013 = 1.3K1940 * 1013 = 1.3K1940 * 1013 =
SUBROUTINE FIL3X3(IRC)
COMMON COMMN(44CD)
COMMON / S / D(12,12,11)
EQUIVALENCE (NORD , COMMN(11))
EQUIVALENCE (ANRP , COMMN(1491))
EQUIVALENCE (ANRP , COMMN(1490))
EQUIVALENCE (ANDBP , COMMN(1489))
EQUIVALENCE (ANRP , COMMN(1488))
EQUIVALENCE (ALRP , COMMN(1487))
EQUIVALENCE (ALRP , COMMN(1486))
EQUIVALENCE (ALRP , COMMN(1485))
EQUIVALENCE (YR , COMMN(1484))
EQUIVALENCE (YR , COMMN(1483))
EQUIVALENCE (YR , COMMN(1482))
EQUIVALENCE (YR , COMMN(1481))
EQUIVALENCE (GAMA , COMMN(757))
EQUIVALENCE (U , COMMN(755))

C DO INITIALIZE A 12X12 DETERMINANT WITH POLYNOMIAL TERMS AND 
C FILL IN A VARIABLE 3X3 DETERMINANT AT THE INDICATED 
C POSITION ON THE DIAGONAL 
C NORD - POSITION OF FIRST TERM ON DIAGONAL 
C ANRP - ARRAY OF ORDERS OF POLYNOMIAL TERMS 
C D - ARRAY OF COEFFICIENTS OF POLYNOMIAL TERMS 
C
C 1405 DELETED 
DIMENSION NORD(12,12)
INTEGER ORC(3,3)

DATA NRD / 4*1,2,3*1,2 /
DATA DTR , C / 57.29378 , 32.174 /

C C C
C ZERO DETERMINANT 
C DD 10 I=1,2 
C DD 10 J=1,2 
C NORD(I,J)=1 
10 DD 10 K=1,11 
C ORC(I,J,K)=0,C,C 

C C C
C SET DIAGONAL TO 1.0 
C DD 20 I=1,3 
C NORD(I,1,1)=C 

20 DD(I,1,1)=1.0 
C C C
C FILL ARRAY WITH ORDERS OF POLYNOMIALS OF 3X3 DETERMINANT 
C IRC=IRC-1 

118
Contrails

C

FILL ARRAY WITH COEFFICIENTS OF POLYNOMIALS OF 3X3 DETERMINANT

C

CDE=GAMA/D

C

CG=G*SIN(TDB)

C

CGC=G*COS(TDB)

C

IK=IRC

C

OR(IK,IC,1)=YS/U

C

OR(IK,IC,2)=YBD/U

C

IC=IC+1

C

OR(IK,IC,1)=GCG/U

C

OR(IK,IC,2)=YQ/U

C

IC=IC+1

C

OR(IK,IC,1)=GSC/U

C

OR(IK,IC,2)=YH/U

C

IC=IC+1

C

OR(IK,IC,1)=ALBP

C

OR(IK,IC,2)=ALBDP

C

IC=IC+1

C

OR(IK,IC,1)=C

C

OR(IK,IC,2)=ALRDP

C

IC=IC+1

C

OR(IK,IC,1)=D

C

OR(IK,IC,2)=ALRBP

C

IC=IC+1

C

OR(IK,IC,1)=AMBP

C

OR(IK,IC,2)=AMRBP

C

IC=IC+1

C

OR(IK,IC,1)=C

C

OR(IK,IC,2)=AMRBP

C

IC=IC+1

C

OR(IK,IC,1)=C

C

OR(IK,IC,2)=AMRBP

C

IC=IC+1

C

RETURN

END

Approved for Public Release
SUBROUTINE GLST
COMMON CMCN(444C)
COMMON ylf(12, 12, 11)$DMU(15), 0(12, 12, 11), 0UM(16)
EQUIVALENCE (ALG, CMCN(1))
EQUIVALENCE (BORDER, CMCN(145))
EQUIVALENCE (INDST, CMCN(726))
EQUIVALENCE (INDST MOD, CMCN(727))
EQUIVALENCE (SPANF, CMCN(732))
EQUIVALENCE (INDST, CMCN(746))
EQUIVALENCE (V, CMCN(755))
EQUIVALENCE (ALT, CMCN(756))
EQUIVALENCE (LU, CMCN(760))
EQUIVALENCE (LV, CMCN(761))
EQUIVALENCE (SIGHI, CMCN(762))
EQUIVALENCE (SIGHI, CMCN(763))
EQUIVALENCE (YA, CMCN(1463))
EQUIVALENCE (XAD, CMCN(1464))
EQUIVALENCE (KX, CMCN(1465))
EQUIVALENCE (KX, CMCN(1466))
EQUIVALENCE (KX, CMCN(1467))
EQUIVALENCE (ZA, CMCN(1468))
EQUIVALENCE (ZAD, CMCN(1470))
EQUIVALENCE (ZG, CMCN(1471))
EQUIVALENCE (ZU, CMCN(1473))
EQUIVALENCE (MA, CMCN(1475))
EQUIVALENCE (MAN, CMCN(1476))
EQUIVALENCE (Mo, CMCN(1477))
EQUIVALENCE (Mo, CMCN(1478))
INTEGER AOMOD(12, 12), BORDER(12, 12)
REAL LU, LV, XL, LVV, XLV
REAL MAX, MOD, LV, XL, LVV
IF (INDST, EQ, 110) GO TO 140
KH = ALT
IF (KH, EQ, 110) X = 10.
IF (KH, GT, .179), .NR. INDST, NE, 0) GO TO 100
LM = KH
LVV = 1.0, .MX + 0.13
LXYV
IF (KH, GT, .110), .NO. GO TO 6
$G = 1.25 - .35*ALOIGIC(XH)
SIGHT = SIGHT + $G*SIGN + 2/LK
SIGHT = SIGHT
GU TO 120
 SIGHT = L$G, .LT, .125*ALOIGIC(XH)
GU TO 60
11) LW = 1195.
14) LWU
LWV
11 IF (INDST, EQ, 110) GO TO 6
6) $G = 21.
SIGHT = $G
SIGHT = $G
12) CONTINUE

120

Approved for Public Release
150 BODKERTL1=l-l=1-6
P1=1,14155
PIV=PIV
FGBDV=V*SPANFJ/PIV
WRITEE6,10CCUV=LU,SIGM,SIGL
C = AKHw(COLUMN, N) WHERE N=1 IS USED FOR CONSTANT TERM
C = N=2 IS USED FOR LAMBDA TERM
C = N=3 IS USED FOR LAMBDA SQUARED TERM ETC.
\[ A(8, 2, 2) = 2.4 \cdot L \cdot V \cdot V \cdot T \cdot O \cdot P \cdot V \]
\[ A(8, 3, 3) = 4.4 \cdot L \cdot V \cdot V \cdot 2 \cdot 2 \cdot L \cdot V \cdot V \cdot T \cdot O \cdot P \cdot V \]
\[ A(8, 4, 3) = 4.4 \cdot L \cdot V \cdot V \cdot 2 \cdot T \cdot O \cdot P \cdot V \]
\[ \text{PART} = \text{SORT}(\{ (P \cdot L \cdot V) / (4 \cdot 4 \cdot \text{SPAN}) \} \cdot \cdot \cdot , 3333) \]
\[ b(6, 1, 1) = (\text{SIG} / 1) \cdot 1 \cdot 2 \cdot 5 \cdot (\text{SORT}(L \cdot V \cdot V)) \cdot \text{PART} \]
\[ b(7, 2, 1) = (\text{SIG} / V) \cdot \text{SORT}(L \cdot V \cdot V) \]
\[ b(7, 2, 1) = b(7, 2, 1) + 1.73205 \cdot L \cdot V \cdot V \]
\[ b(7, 2, 2) = b(7, 2, 1) \]
\[ z(0, 3, 3) = b(7, 2, 2) \]

RETURN

END
MAIN FOR MISER—MULTITERM ISE REDUCTION

EXTERNAL KLOCK,KLOCK1
COMMON COMON(4400)
COMMON / S / A(1559),B(1601)
INTEGER AORDER(12,12,1),BORDER(12,12,1)
INTEGER RUNLINK
DIMENSION AP(5),ASKBJJ(12,12,1),CNSTM(5)
DIMENSION CNSTM(5),CLTOFF(5),DICOSS(5),KKR(5),LLL(5)
DIMENSION MORD(12,12),NIXX(B),NOLT(8),P(5)
DIMENSION TITLE(10)
DIMENSION FFLEN(37),TFLEN(3)
EQUIVALENCE (CNSTM,CMON(704))
EQUIVALENCE (CNSTM,CMON(712))
EQUIVALENCE (CLTOFF,CMON(734))
EQUIVALENCE (TIKJCC,CMON(743))
EQUIVALENCE (DICOSS,CMON(764))
EQUIVALENCE (P,CMON(780))
EQUIVALENCE (P,CMON(781))
EQUIVALENCE (PIKJCC,CMON(801))
EQUIVALENCE (TFLEN,CMON(823))
EQUIVALENCE (FFLEN,CMON(504))
EQUIVALENCE (ASKBJJ,CMON(1007))
EQUIVALENCE (AORDER,CMON(1))
EQUIVALENCE (AORDER,CMON(1))
EQUIVALENCE (BORDER,CMON(1))
EQUIVALENCE (BORDER,CMON(1))
EQUIVALENCE (IN,CMON(469))
EQUIVALENCE (NOLT,CMON(481))
EQUIVALENCE (TITLE,CMON(799))
EQUIVALENCE (NUMEQ,CMON(509))
EQUIVALENCE (TED,CMON(508))
EQUIVALENCE (ISETOL,CMON(509))
EQUIVALENCE (ISETOL,CMON(510))
EQUIVALENCE (ISTASS,CMON(511))
EQUIVALENCE (NMIIE,CMON(521))
EQUIVALENCE (AMOUNT,CMON(512))
EQUIVALENCE (KNTIE,CMON(514))
EQUIVALENCE (KAPAM,CMON(515))
EQUIVALENCE (NUMNC,CMON(517))
EQUIVALENCE (NUMRC,CMON(518))
EQUIVALENCE (IKTIM,CMON(431))

MI8M=0

CALL RUNLINK(PLUCK1,-18)

6.6 FORMAT(///,*(YOU STARTING POINT LEAVES SOMETHING TO BE DESIRED AS
THE ARIVE STATEMENT INDICATES*))
6.7 FORMAT(///,*(ATTENTION, YOU HAVE CNSTM(*,12,*)=,0PE10.4, *= GREATER TH
AN CNSTM(*,12,*)=,0PE10.4))
1.3 KNTIE=0

123

Approved for Public Release
CONTAINS

DO 10 I=1,NPARAM
     IF(CNSTMNI.LT.LE.CNSTMX(I)) GO TO 70
     WRITE(6,ECST).CNSTMN(I),CNSTMX(I)
10    CONTINUE
     CALL BLANK
     CALL WRITE
C CALL MINI TO CHECK FOR AND CREATE A STABLE STARTING CONFIGURATION.
     CALL MINIFX(F,P,KPARAM)
C IF MINI FAILS, THE FAILURE WILL BE DISCOVERED IN ISE.
C CALL ISE TO INITIALIZE PISE
C CALL ISE
C GO TO 320 TO WRITE DLT DATA FOR FIRST ISE CALCULATIONS
C IF(JSTAB.GT.CIGN) GO TO 320
     WRITE(6,ECIG)
     STOP 1
C CHECK TO SEE IF CNSTMN.LE.P.LE.CNSTMX
C DO 70 I=1,NPARAM
     KKK(I)=10
     LLL(I)=10
     IF(P(I).LT.CIGN) GO TO 175
     P(I)=CNSTMN(I)
     KKK(I)=10
     GO TO 200
175 IF(P(I).LT.CNSTMX(I)) GO TO 200
     P(I)=CNSTMX(I)
     LLL(I)=10
70 CONTINUE
C IF P WAS CONSTRAINED, CALL ISE BEFORE CALLING GRADI
C DO 225 I=1,NPARAM
     IF(KKK(I).LT.C).AND.(LLL(I).LT.0)) GO TO 225
     CALL ISE
     GO TO 250
225 CONTINUE
C CALL GRADI
C IF P WAS CONSTRAINED, SET APPROPRIATE DIRCOS(*)S TO 0.00
C DO 300 I=1,NPARAM
     IF(KKK(I).LT.0)ANU.(JIRCOS(I),GT.0.00)DIRCOS(I)=0.00
     IF(LLL(I).LT.0)ANU.DIRCOS(I)=0.00
300 AP(I)=P(I)

124

Approved for Public Release
AISE=PISE
AI 799 I=M, KPARA
IF(UNICHST(I).NE.C).GO TO 310
CALL GRADII
GO TO 310
CONTINUE
C
C MINIMIZE ALONG GRADIENT BY CALLING MINIM
C
310 CALL MINIM
C
410 NTIM=RUNLIN(KLOCK.DUM)
IIFLT(I.GT.0.AND.0.AG.T.0.)GO TO 321
IIF(KNTISE.EQ.1).GO TO 321
IIF(NTIM.CE.1T.0).GO TO 352
C
411 CALL WRITE2
GO 350 NNNISE=1.ISETIM
MATCH=1
IEO=NUMEO
DI 325 I=1.E0
DI 325 J=1.E0
NUORD(I,J)=AIRDER(I,J,MATCH)
GO 325 K=1.11
NOW=NUMEI(J.K,MATCH)

325 ASKJII(I,J,K)=R(NOW)
CALL SKBJI
AKDREN=NORDUN
AKDUN=MORDNUM+1
DI 350 I=1,NORDUN
C
330 TFDA(I)=TFNUMI(I)
GO 350 I=1,EQ0
NOUT=NOUT(NNNISE)
NINT=NINT(NNNISE)
NUORD=NUMEI(J,K,MATCH)
GO 335 K=1.11
NOW=NUMEI(J,K,MATCH)

335 ASKJII(I,K,1)=R(NOW)
CALL SKBJI
C
330 CALL WRITE2
C
352 CONTINUE
LT=0
IF(KNTIF.EQ.1).GO TO 15C
C
C IF PROGRAM STUCK GO TO 300C
C
C IF(AISE.EQ.PISE).GO TO 300C
C
C DECIDE WHETHER TO STOP DEPENDING ON ISETIM
C
C IF(ISETIM.LT.0).GO TO 450
DI 402 I=1.KPARA
IIF(ANSEP(A)-PI(I)).GT.0.CLOFF(I).GO TO 150
IIF(PI(I).LT.CNSTPM(A)).GO TO 150

125
Approved for Public Release
1F \( P(1) \), GT, CNSTXM * X1500 TO 1500
4 CONTINUE
1F IE150LMG.X150 TO +75
4 IE1 ABS(AISE-PISE) , GT, I0ISE150 TO 150
4 CONTINUE
1LT (TIM=TIM+1
1IF (1M.T.1250 TO 321
CALL CLASS | ENDSL |
WRITE (4, 600)
600 FORMAT (1X, 2X, 1X, 15X, 'NAME BROTHER, YOU HAVE ARRIVED', 
* 50X-------YOU HAVE HIT A NEW LOW------*)
ILNK = 4
CALL OVLAY
C       IF PROGRAM STUCK, MOVE P A SMALL DISTANCE
C       C       IA IS THE CUMULATIVE SUM OF TIMES THE PROGRAM IS STUCK
C 1050 IF (IA.T.50) TO 3002
WRITE (6, 350)
350 FORMAT (2X, 15X, 'THE SEARCH FOR THIS CASE HAS STOPPED SINCE THE PR
OGRAM HAS RETURNED TO THE SAME POINT FIVE TIMES (CUMULATIVE).', 
* 15X, 'NEITHER ACCEPT THE LAST POINT AS THE MINIMUM', 
* 15X, 'OR ACT THE SEARCH IN A DIFFERENT REGION TO SEE IF IT APPROACHES THE S
AME POINT', 
* 15X, 'OR ELSE', 
* 15X, 'EITHER TIGHTEN THE TOLERANCE', 
* 15X, ' OR CHANGE DEL(1).')
ILNK = 4
CALL OVLAY
C 1350 MATCH = 0
LT = 1
J = C
K = C
160 IF (3010 = 1, \( I=1 \), K, PARAM
MATCH = MATCH + 1
IF (X11(1), 15X, 'I = J + 1
IF (X11(1), 15X, 'X = K + 1
IF (J, NF, MATCH, AND, K, NF, MATCH) TO 3015
310 CONTINUE
WRITE (6, 402)
402 FORMAT (11, 1X, 1X, 15X, 'THE CONSTRAINTS HAVE STOPPED FURTHER SEARCHING', 
* 15X, 'Стations')
0 = 1
STJP = 2
330 IF (0D 3025 = 1, K, PARAM
330 FORMAT (1D) (15X, 'COS (I) = 0.90',
* 15X, 'KITE (0.15))
WRITE (6, 404)
404 FORMAT (2X, 15X, 'NEW ISE, PROGRAM STUCK, NEW I ATTEMPTED')
KOUNT = (0) (15X, 'CALL ISC
IF (ISTARE.LT.0) TO 2100

126
CALL WRITE
LD=IA+1
G) TO 150
3101 FORMAT('ERROR ENDING SIMULATION BECAUSE THE NEW POINT IT CAL-
CULATED IS IN AN UNSTABLE REGION')
3102 WRITE(6,3101)
STOP 3101
END
SUBROUTINE WRITE1

THIS SUBROUTINE PRINTS ALL OUTPUT EXCEPT DIAGNOSTIC MESSAGES

COMMON COMDNI(4400)
COMMON / 6 / SAVE(310)
COMMON / 5 / A(1540),A(1600)
INTEGER AGORDER(12,12,11),BORDER(12,12,11),NOI(15)
DIMENSION CMXXY(12)
REAL JUMP,NUMF(I8,11),MODELM(11)
DIMENSION AISE(I8),ANDPR(137),ANDPRK(37)
DIMENSION CNSTMX(5),CNSTM(5),COLNF(5),DEL(5),DICEOS(5)
DIMENSION MINDON(8),MINDONP(8),NIN(8),MADX(8),MNOT(8),NSYST(6)
DIMENSION TTYPE(18),P(5),SCLSY(5)
DIMENSION TDOE(37),TNUM(37),TITLE(10),WEIGHT(8),WORK(115)
DIMENSION ROOTR(36),ROOTN(36),RCOTR(36),RCOTON(36),RMTL(37)

EQUIVALENCE (MODELD,COMCN(1,579))
EQUIVALENCE (MODELD,COMCN(607))
EQUIVALENCE (FAISE,COMCN(699))
EQUIVALENCE (JUMP,COMCN(703))
EQUIVALENCE (CNSTM,COMCN(764))
EQUIVALENCE (CNSTMX,COMCN(791))
EQUIVALENCE (TDFINS,COMCN(4317))
EQUIVALENCE (SAVEN,SAVE(115))
EQUIVALENCE (TDFINS,SAVE(11))
EQUIVALENCE (JTDFINS,COMCN(734))
EQUIVALENCE (QFL,COMCN(749))
EQUIVALENCE (QDF,COMCN(764))
EQUIVALENCE (JUMP,COMCN(779))
EQUIVALENCE (CMCN(780))
EQUIVALENCE (CMCN(795))
EQUIVALENCE (TDFINS,COMCN(805))
EQUIVALENCE (JUMP,COMCN(806))
EQUIVALENCE (TDFINS,COMCN(815))
EQUIVALENCE (TDFINS,COMCN(823))
EQUIVALENCE (TDFINS,COMCN(904))
EQUIVALENCE (ADPR2,COMCN(959))
EQUIVALENCE (ANDPR2,COMCN(1022))
EQUIVALENCE (RCOTR,COMCN(1059))
EQUIVALENCE (RCOTR,COMCN(1095))
EQUIVALENCE (RCOTR,COMCN(1131))
EQUIVALENCE (RCOTR,COMCN(1167))
EQUIVALENCE (MDSTK,COMCN(1203))
EQUIVALENCE (MDSTK,COMCN(145))
EQUIVALENCE (MDSTK,COMCN(297))
EQUIVALENCE (MADX,COMCN(465))
EQUIVALENCE (MADX,COMCN(475))

128

Approved for Public Release
EQUIVALENCE (NUMFA, COM1, 405)
EQUIVALENCE (NUMFA, COM2, 405)
EQUIVALENCE (NUMFA, COM3, 405)
EQUIVALENCE (NUMFA, COM4, 405)

DATA NW, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

CALL CLASS1(NINDCLS)
WRITE(6, ECCCT) TITLE
WRITE(6, ECCCT)

IF (L.EQ.1) THEN
  CALL CLASS1(NINDCLS)
  WRITE(6, ECCCT) TITLE
  WRITE(6, ECCCT)
END IF

L = 1

CALL CLASS1(NINDCLS)
WRITE(6, ECCCT) TITLE
WRITE(6, ECCCT)

END IF

DO 3 1 = 1, 12
  IF (L.EQ.1) THEN
    CALL CLASS1(NINDCLS)
    WRITE(6, ECCCT) TITLE
    WRITE(6, ECCCT)
  END IF
  L = L + 1
3 CONTINUE

DO 3 1 = 1, 12
  WRITE(6, ECCCT) TITLE
  WRITE(6, ECCCT)
3 CONTINUE

END IF

DO 3 1 = 1, 12
  IF (L.EQ.1) THEN
    CALL CLASS1(NINDCLS)
    WRITE(6, ECCCT) TITLE
    WRITE(6, ECCCT)
  END IF
  L = L + 1
3 CONTINUE

END IF
DO 31 K=1,11
  M=1
  DO 30 J=1,12
    N=NEW(I,J,K,L)
  30 SUMMY(J) = NDK
  CALL CLASS(INDCLS)
  WRITE(*,674) (SUMY(I,J),J=1,12)

  0.11 FORMAT(10,D8.0,MODEL 1*,7X,MODEL 2*,7X,MODEL 3*,7X,MODEL 4*,7X,
         MODEL 5*,7X,MODEL 6*,7X,MODEL 7*,7X,MODEL 8*,7X,MODEL 9*,7X,NUMFRAT
         ,15X,4X))
  WRITE(*,6011) MCD(N),N=1,8

  0.12 FORMAT(6X,(12X,ORDER = *,12X,2X))
  J=2
  K=1,11
  M=1
  WRITE(*,4010)M,(MODEL(N,K),N=1,8)
  WRITE(*,6012)

  0.13 FORMAT(6X,25HMODLIN(N,K),N=1,8)
  DO 55 K=1,11
    D(K)=-1

  WRITE(*,6013)M,(MODEL(N,K),N=1,8)
  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRITE(*,6013)

  WRIT
SUBROUTINE MINIMI

THIS SUBROUTINE FINDS BY GOLDEN SECTION THE POINT P AT WHICH ISF I
MINIMUM ALONG A RAY WHICH EMANATES FROM THE POINT P-START AND WHI
FOLLOWS THE DIRECTION DEFINED BY WLNS DIRCOS(I)

THE INPUTS TO THIS SUBROUTINE ARE DISTIC, DROGS, ISTABLE, JUMP, KOUN
KPARAM, P, PISE

THE OUTPUTS OF THIS SUBROUTINE ARE AJP, DWHICH, JUMP, KOUNT, P, PISE

DEFINITIONS
KOUNT IS A COUNTER, IT IS INCREASED BY 1 BY THIS SUBPROGR

COMMON CUMCN(44CC)
REAL JUMP
DIMENSION A(5), C(5), CSTMX(5), CUTOFF(5), DIRCOS(5)
DIMENSION ISTABLE(5), JUMP(5), STEP(5)
DIMENSION CNSTMX(5)
EQUIVALENCE (AJP, CUMCN 7651)
EQUIVALENCE (CSTMX, CUMCN 7641)
EQUIVALENCE (CNSTMX, CUMCN 7691)
EQUIVALENCE (CUTOFF, CUMCN 7341)
EQUIVALENCE (DIRCOS, CUMCN 7641)
EQUIVALENCE (JUMP, CUMCN 7791)
EQUIVALENCE (P, CUMCN 7801)
EQUIVALENCE (PISE, CUMCN 8061)
EQUIVALENCE (ISETOL, CUMCN 5091)
EQUIVALENCE (ISTABLE, CUMCN 5131)
EQUIVALENCE (KOUNT, CUMCN 5151)
EQUIVALENCE (KPARAM, CUMCN 5152)
KOUNT = KOUNT + 1

BEGIN DETERMINING END POINTS -- (A, 0) OF INTERVAL

C ) 25 I=1, KPARAM
7 ) A(I)=P(I)
15 ) A(1)=PISE
C ) L1S=L+1, EIC
C ) IC=100
C ) IF=100
C ) JJ=0
30 ) D: J5 I=1, KPARAM
17 ) S(I)=S(I)+DIRCos(I)*JUMP
L1 ) DU L2S = I+1, KPARAM
L15 ) P(I)+A(I)*STEP(I)
CALL ISF
IF ISTABLE, GFI, GICO TO 15C

133

Approved for Public Release
JUMP=JUMP+.500
GO TO 50

C CHECK IF G II WITHIN CONSIDEREDS
C
150 DO 155 I=1,KPARAM
   IF(P(I).GE.CNSTMXI(I).AND.P(I).LE.CNSTMXII(I))GO TO 155
   RETURN
155 CONTINUE
   LLL=LLL+1
   IF(PISET.GT.AIASE)GO TO 175
   GO 160 I=1,KPARAM
   C(I)=M(I)
160 A(I)=P(I)
   CISE=AIASE
   AIASE=PISET
   SAVE=JUMP
   GO TO 100
175 IF(LLL.LT.2)GO TO 190
   GO 180 I=1,KPARAM
   A(I)=C(I)
190 G(I)=P(I)
   AISET=CISE
   CISE=PISET
   AJP=SAVE+JUMP
   JUMP=AJP
   GO TO 200
200 JUMP=JUMP
   CD 195 I=1,KPARAM
210 G(I)=P(I)
   CISE=PISET
   JUMP=JUMP+.500
C END DETERMINING END POINTS--{A,G} OF INTERVAL
C
200 IF(SETOLT.LT.C)GO TO 206
   GO 205 I=1,KPARAM
   IF ANS(A(I)-G(I)).GT.CLIFFI(i))GO TO 208
205 CONTINUE
   IF(SETOLT.GT.C)GO TO 206
   IF ANS(AISE-GISE).GT.JOLISE(I)GO TO 208
   GO TO 205
C RECIR GOLDEN SECTION
C
210 AJP=AJP+.618033F877CC
   CD 210 I=1,KPARAM
220 STEPI=GEQST(I)*AJP
   IF(L.I.E.225)GO TO 225
   GO 215 I=1,KPARAM
225 P(I)=G(I)*STEPI
   CALL 156

134
QU 720 I=1,KPARAM
221) G(I)=P(I)
    CISE=RISE
222) IF(I.EQ.0) GO TO 230
QU 726 I=1,KPARAM
223) *I(I)+=I(E)+ISTEP(I)
    CALL ISE
QU 728 I=1,KPARAM
224) E(I)=P(I)
    EISE=RISE
225) IF(I.EQ.EISE) GO TO 250
QU 740 I=1,KPARAM
    A(I)=C(I)
226) C(I)=E(I)
    AISE=CISE
    CISE=EISE
    IC=100
    IE=100
QU TU 200
227) QU 760 I=1,KPARAM
    G(I)=C(I)
228) E(I)=C(I)
    EISE=EISE
    CISE=EISE
    IC=100
    IE=-100
QU TU 200
C
C END GOLDEN SECTION
C
C GET MAXIMUM SMALLEST ISE BEFORE RETURNING
C
310) IF(I.EQ.EISE) RETURN
    IF(I.EQ.0) G(I)=I GU TO 350
QU 713 I=1,KPARAM
311) P(I)=G(I)
    PISE=GISE
    RETURN
313) QU 760 I=1,KPARAM
314) P(I)=A(I)
    PISE=AISE
    RETURN
END

Approved for Public Release
SUBROUTINE GRADI

THIS PROGRAM CALCULATES THE GRADIENT OF A POINT P

THE GRADIENT IS COMPUTED ONLY ON ONE SIDE OF P. (IF DEL(I) IS TOO
TRUE GRADIENT AT P BUT ALSO COMPUTE A PICK APPROXIMATION T
NEARER TO MINIMUM THAN DEL(I)). FOR THIS REASON, KEEP DEL(I)
THE INPUTS TO THIS SUBROUTINE ARE DEL,I(1),KPARAM,P,ISE
THE OUTPUTS OF THIS SUBROUTINE ARE DIRCOS
DEFINITIONS

DEL(I) IS THE DISTANCE P WILL BE INCREMENTED IN THE I-TH
DIMENSION TO CALCULATE DIRCOS IN THE I-TH DIMENSION
PISE IS THE VALUE OF ISE AT P I, THE I-TH DIMENSION
P(I) IS THE VALUE OF P IN THE I-TH COORDINATE OF P

COMMON COMON(4400)
DIMENSION A(5),DEL(5),DIRCOS(5),P(5)
EQUIVALENCE (DEL,COMON (749))
EQUIVALENCE (DIRCOS,COMON (764))
EQUIVALENCE (P,COMON (780))
EQUIVALENCE (PISE,COMON (806))
EQUIVALENCE (I(1),COMON (513))
EQUIVALENCE (KPARAM,COMON (515))
EQUIVALENCE (I(1),DENOM)
DO 5 I=1,KPARAM

A(I)=P(I)
AISE=PISE
CO 15 I=1,KPARAM
P(I)=A(I)*DEL(I)
CALL ISE
IF(I(1),.GT.,10)
P(I)=A(I)-DEL(I)
CALL ISE
LIRCOS(I)=(AISE-PISE)/DEL(I)
GO TO 15
10 DIRCOS(I)=(PISE-AISE)/DEL(I)
15 P(I)=A(I)
PISE=AISE
ENTRY GRADI

DENOM=0.C0
DO 25 I=1,KPARAM
DENOM=DENOM+DIRCOS(I)*DIRCOS(I)
DENOM=SQR(DENOM)
IF(DENOM.LE.5.E-20)WRITE(6,50)
FORMAT(1X,2X,CONSTRAINED MINIMUM (DIRECTICA COSINES EQUAL ZER
I00*)
DENOM=1.CC/DENOM

136
DO 30 I=1,KPARAM
30 CIRCUS1 = CIRCOS1/1+DENCM
RETURN
END
THIS SUBROUTINE FINDS THE DETERMINANT OF A MATRIX WHOSE ELEMENTS ARE POLYNOMIALS IN S

COMMON COMOM(44CC)
EQUIVALENCE (ANS ,COMO(N, C04))
EQUIVALENCE (TEMP ,COMO(1G07))
EQUIVALENCE (A ,COMO(2591))
EQUIVALENCE (MURD ,COMO(N, 30S))
EQUIVALENCE (N ,COMO(3, 50B))
EQUIVALENCE (NA ,COMO(N, 51B))

! DIMENSION IDO(24)
! DIMENSION SUBPR2(21), SUBPR3(31), SUBPR4(41), SUBPR5(51), SUBPR6(61)
! * SUBPR7(71), SUBPR8(81), SUBPR9(91), SUBP10.1011, SUBP11.1111
! * SUBP12.1(121)
! DIMENSION MURD(12, 12), A(12, 12, 11), ANS(37)
! DIMENSION MURD(12, 12), TEMP(12, 12, 11)
EQUIVALENCE (ID0(1), J00), (ID0(2), J1STOP)
* (ID0(3), J0G0), (ID0(4), J1STOP)
* (ID0(5), K00), (IDC(6), K1STOP)
* (ID0(7), L00), (IDC(8), L1STOP)
* (ID0(9), M00), (IDC(10), M1STOP)
* (ID0(11), J2G0), (ID0(12), J2STOP)
* (ID0(13), J2G0), (ID0(14), J2STOP)
* (ID0(15), J2G0), (IDC(16), J2STOP)
* (ID0(17), L2G0), (ID0(18), L2STOP)
* (ID0(19), M2G0), (ID0(20), M2STOP)
* (ID0(21), J3G0), (ID0(22), J3STOP)
* (ID0(23), J3G0), (IDC(24), J3STOP)

IT IS REALIZED THAT PROGRAMMING CAN BE CONDENSLED BY USING SUBSCRIPT VARIABLES IN THIS PROGRAM, BUT USE HAS SHOWN THE NON-ELEGANT SCHEME HERE IS FASTER—AND IN THIS PROGRAM TIME IS THE CRITERION

NX = 14 - N
DO 10 I = 1, 12
PD 10 I = 1, 12
MURD(I, J) = 1
DO 12 K = 1, 11
11 A(I, J, K) = 0
DO 12 I = 1, 12
MURD(1, J) = C
12 A(I, J, 1) = 1
JJ = 0
DO 13 JJ = NX, 12
13 JJ = JJ + 1

138

Approved for Public Release
NORD(I,J) = MORD(I,J,J)
DO 13 I=1,11
13 K(I,J,K) = TEMP(I,J,K)
C
C SET UP LIMITS ON DU LOOPS
C
DO 44 I=1,12
DO 43 J=1,12
IF(NORD(I,J,J),EQ.,-1) GO TO 43
40 DO(I,J=1,12)
43 CONTINUE
44 CONTINUE
DO 44 I=1,12
DO 47 J=1,12
46 INJ=INJ+1
IF(NORD(I,J,J),EQ.,-1) GO TO 47
45 DO(I,J=1,12)
47 CONTINUE
46 CONTINUE
C
C ******************************
C *
C N=0
DO 60 I=1,12
60 ANSI(I) = C
C
C FIND A ROW AND A COLUMN COMBINATION
C
KFLAG=2
DO 156 I=1,12,1 stop
IF(NORD(I,1,1),EQ.,-1) GO TO 156
DO 155 J=1,12,1 stop
IF(NORD(I,J,1),EQ.,-1) GO TO 155
IF(JEQ(1,1),EQ.,-1) GO TO 155
DO 154 K=1,12,1 stop
IF(NORD(I,J,K),EQ.,-1) GO TO 154
IF(KEQ(1,1),EQ.,-1) GO TO 154
DO 153 L=1,12,1 stop
IF(NORD(I,J,K,L),EQ.,-1) GO TO 153
IF(I,1,1),EQ.,-1) GO TO 153
DO 152 M=1,12,1 stop
IF(NORD(I,J,K,L,M),EQ.,-1) GO TO 152
IF(MEQ(1,1),EQ.,-1) GO TO 152
DO 151 N=1,12,1 stop
IF(NORD(I,J,K,L,M,N),EQ.,-1) GO TO 151
IF(NEQ(I,J,K,L,M,N),EQ.,-1) GO TO 151
DO 150 J=1,12
150 ANSI(I) = ANSI(I) + 1
C
C ******************************
IF NORD T J21 EQ -11GC TO 14G

J2.EQ.12G0 TO 15C

DU 149 K2.EQ 2G0.K2STOP

IF NORD T J30M J21 EQ-11GC TO 14G

IF K2.EQ J1.OR.K2.EQ J1.OR.K2.EQ J1.OR.K2.EQ J1.OR.
K2.EQ.J2.OR.K2.EQ.J2.OR.K2.EQ.J2.OR.

DU 148 L2.EQ 2G0.L2STOP

IF NORD T J31M J21 EQ-11GC TO 14G

IF L2.EQ J1.OR.L2.EQ J1.OR.L2.EQ J1.OR.L2.EQ J1.OR.
L2.EQ.J2.OR.L2.EQ.J2.OR.L2.EQ.12G0 TO 14G

UD 147 M2.EQ 2G0.M2STOP

IF NORD T J32M J21 EQ-11GC TO 14G

IF M2.EQ J1.OR.M2.EQ J1.OR.M2.EQ J1.OR.M2.EQ J1.OR.
M2.EQ J2.OR.M2.EQ J2.OR.M2.EQ J2.OR.M2.EQ J2.OR.

UD 146 L2.EQ 13GC.L2STOP

IF NORD T J33M J21 EQ-11GC TO 14G

IF I2.EQ J1.OR.I2.EQ J1.OR.I2.EQ J1.OR.I2.EQ J1.OR.
I2.EQ J2.OR.I2.EQ J2.OR.I2.EQ J2.OR.I2.EQ J2.OR.

UD 145 J3.EQ 4G0.J3STOP

IF NORD T J34M J21 EQ-11GC TO 14G


UD 144 J3.EQ 13GU TO 14G

C

C CALCULATE SUBORDERS IN JORD* AND SUBPRODUCTS IN SUPPR*

C KFLAG SIGNALS WHERE TO START CALCULATING SUBPRODUCTS

C

GO TO 185,65,64,93,57,1C1,1C5,1C9,113,1L7,121,125,J.KFLAG

83 KKK=NORD T J11+1

LLL=NORD T J11+1

JORD*=KKK+LLL-2

INC=JORD+1

UD = INC,1,INUT

86 SUBPR2 (IND) = C

GO 8T KKKK=L,1KKK

GO 87 LLLL=L,1LLL

IND=KKKK,LLL-1

87 SUBPR2 (IND)=SUBPR2 (IND)+A(11111111K)

JISIGN=1

IF J11+1=11111111SIGN=J1SIGN

89 KKK=JORD+1

LLL=NORD T J11+1

JORD*=KKK+LLL-2

IND=JORD+1

UD 90 INC=1,INUT

90 SUBPR3 (IND)=C

GO 9I1=1KKK

DU 91 LLLL=L,1LLL

IND=KKK,LLL-1

160

Approved for Public Release
SUPPR3(IND) = SUPPR3(IND) + SLBPR2(KKKK) * A(3, Ki, LLLL) * AISIGN = JISIGN
IF (K1. L1. I1) JISIGN = M1SIGN
IF (K1. L1. J1) JISIGN = M1SIGN

93 KKK = JORD4 + 1
LLL = NORD5, L1 + 1
JORD4 = KKK * LLLL - 2
INC = JORD4 + 1
DJ 99 INE = 1, INOT

95 SUPPR4(IND) = SLBPR4(IND) + SLBPR3(KKKK) * A(4, Li, LLLL) * LISIGN = K1SIGN
IF (L1. L1. L1) LISIGN = M1SIGN
IF (L1. L1. J1) LISIGN = M1SIGN
IF (L1. L1. K1) LISIGN = M1SIGN

97 KKK = JORD5 + 1
LLL = NORD5, L1 + 1
JORD5 = KKK * LLLL - 2
INC = JORD5 + 1
DJ 58 INE = 1, INOT

91 SUPPR5(IND) = 0
DJ 94 KKKK = 1, KKK
DJ 94 LLLL = 1, LLL
IND = KKKK + LLLL - 1

99 SUPPR5(IND) = SUPPR5(IND) + SUPPR4(KKKK) * A(5, M1, LLLL) * M1SIGN = L1SIGN
IF (M1. L1. J1) M1SIGN = M1SIGN
IF (M1. L1. K1) M1SIGN = M1SIGN
IF (M1. L1. L1) M1SIGN = M1SIGN

101 KKK = JORD5 + 1
LLL = NORD5, L1 + 1
JORD5 = KKK + LLLL - 2
INC = JORD5 + 1
DJ 102 INE = 1, INOT

132 SUPPR6(IND) = 0
DJ 103 KKKK = 1, KKK
DJ 103 LLLL = 1, LLL
IND = KKKK + LLLL - 1

133 SUPPR6(IND) = SUPPR6(IND) + SLBPR5(KKKK) * A(6, I2, LLLL) * I2SIGN = M1SIGN
IF (I1. L1. I1) I2SIGN = M1SIGN
IF (I1. L1. J2) I2SIGN = M1SIGN
IF (I1. L1. L2) I2SIGN = M1SIGN

105 KKK = JORD6 + 1
LLL = NORD6, L2 + 1
JORD6 = KKK + LLLL - 2

141

Approved for Public Release
[Unreadable text]
LLL = NORDIC(LLL, LLL+1)
JORD = JORD+1
INDT = IND+1
CJ 118 INC+1, INDT
118 SUPPINDING = C
DJ 119 KKKK = KKK
DD 119 LLLL = LLLL
IN = KKKK, LLLL, LLLL, LLLL, LLLL, LLLL, LLLL, LLLL
119 SUPPIND = SUPPIND(IND) * SLAPR9(KKKK) * A(10, M2, LLLL)
M2SIGN = L2SIGN
IF(M2LT + M1) M2SIGN = M2SIGN
IF(M2LT + M2) M2SIGN = M2SIGN
IF(M2LT + M3) M2SIGN = M2SIGN
IF(M2LT + M4) M2SIGN = M2SIGN
IF(M2LT + M5) M2SIGN = M2SIGN
IF(M2LT + M6) M2SIGN = M2SIGN
IF(M2LT + M7) M2SIGN = M2SIGN
IF(M2LT + M8) M2SIGN = M2SIGN
IF(M2LT + M9) M2SIGN = M2SIGN
IF(M2LT + M10) M2SIGN = M2SIGN
IF(M2LT - 1) M2SIGN = M2SIGN
IF(M2LT - 2) M2SIGN = M2SIGN
L2SIGN = M2SIGN
IF(M2LT - M1) L2SIGN = 15SIGN
IF(M2LT - M2) L2SIGN = 15SIGN
IF(M2LT - M3) L2SIGN = 15SIGN
IF(M2LT - M4) L2SIGN = 15SIGN
IF(M2LT - M5) L2SIGN = 15SIGN
IF(M2LT - M6) L2SIGN = 15SIGN
IF(M2LT - M7) L2SIGN = 15SIGN
IF(M2LT - M8) L2SIGN = 15SIGN
IF(M2LT - M9) L2SIGN = 15SIGN
IF(M2LT - M10) L2SIGN = 15SIGN
121 KKKK = JORD+1
LLL = KORD(LLL, LLL+1)
JORD = JORD+1
DD 123 KKKK = 1, KKK
DD 123 LLLL = LLLL
IN = KKKK, LLLL, LLLL, LLLL
123 SUPPIND = SUPPIND(IND) * SLAPR9(KKKK) * A(11, M3, LLLL)
M3SIGN = L3SIGN
IF(M3LT + M2) M3SIGN = M3SIGN
IF(M3LT + M3) M3SIGN = M3SIGN
IF(M3LT + M4) M3SIGN = M3SIGN
IF(M3LT + M5) M3SIGN = M3SIGN
IF(M3LT + M6) M3SIGN = M3SIGN
IF(M3LT + M7) M3SIGN = M3SIGN
IF(M3LT + M8) M3SIGN = M3SIGN
IF(M3LT + M9) M3SIGN = M3SIGN
IF(M3LT + M10) M3SIGN = M3SIGN
IF(M3LT - 1) M3SIGN = M3SIGN
IF(M3LT - 2) M3SIGN = M3SIGN
L3SIGN = M3SIGN
IF(M3LT - M3) L3SIGN = 15SIGN
IF(M3LT - M4) L3SIGN = 15SIGN
IF(M3LT - M5) L3SIGN = 15SIGN
IF(M3LT - M6) L3SIGN = 15SIGN
IF(M3LT - M7) L3SIGN = 15SIGN
IF(M3LT - M8) L3SIGN = 15SIGN
IF(M3LT - M9) L3SIGN = 15SIGN
IF(M3LT - M10) L3SIGN = 15SIGN

143 Approved for Public Release
Contrails

LWL=NRD(12,L3)+1
JORD12=KKK+LLL-2
INDD=JNRC12+1
NU 126 IND=1, INDI

SUBPIZ CONT = C
GO 127 KKK=1,KKK
GO 127 LLLL=1,LLL
IND=KKK+LLL+1

SUBPIZ INDI=SUBPIZ(IND)+SLBP11(KKKK)+A(12,J3+LLL)
IND=JORD12+1
IF(JSTON,JL=1)GO TO 140
GO 139 LITTLE=+1,IND

ANSF(LITTLE)=ANSF(LITTLE)+SLBP1Z(LITTLE)
GU TO 142
GO 143 IND=LITTLE=+1,IND
GO 141 ANSF(LITTLE)=ANSF(LITTLE)+SLBP1Z(LITTLE)
GO 142 NA=MAX(NA, JORD12)
KFLAG=12
GO CONTINUE

IF(KFLAG.GT.11)KFLAG=11
IF(KFLAG.GT.10)KFLAG=10
IF(KFLAG.GT.9)KFLAG=9
IF(KFLAG.GT.8)KFLAG=8
IF(KFLAG.GT.7)KFLAG=7
IF(KFLAG.GT.6)KFLAG=6
IF(KFLAG.GT.5)KFLAG=5
IF(KFLAG.GT.4)KFLAG=4
IF(KFLAG.GT.3)KFLAG=3
KFLAG=2
GO CONTINUE

IF(NA.1)ZANS=ANSF(NA)
IF(ABS(ZANS).GT.1.E-6)GO TO 400
NA=NA+1
IF(NA.NOT.0)GO TO 500
GO TO 402

RETURN

WRITE(*,501)
501 FORMAT(62MH SOMETHING IS WRONG. SKRM HAS GENERATED A ZERO POLYN
OMIAL )
STOP 500
END
THIS SUBROUTINE CALCULATES THE TOTAL ISE BY KRASOVSKII'S METHOD AT
HAS DETERMINED THE SYSTEM TRANSFER FUNCTION AND TESTED FOR STABILITY.

COMMON COMCN(4400)
COMCN / 2 / AI15595,AMAT(1600)
INTEGER AZERDER(12,12),AHORDER(12,12,1),GRNMO,GRNDO
REAL MODELDE(11),MODELDE(8,11)
DIMENSION ISE(1),AMO(11),AM1(147),AN(47)
DIMENSION AS1(37),ASKB1(12,12,1),AN1(37),A2(47)
DIMENSION ICOST(8)
DIMENSION ELT(47),MDADO(8),MDNCK(8),MORD(12,12)
DIMENSION NOUT,NSYSTEM,AETYPE(4),NUMEC(11),P(15),R(36,18)
DIMENSION SCLSY(35),TEMPD(11),TNM(11),TFD(37),TFN(37)
DIMENSION L(147),V(47),XM(2209),Y(47),Z(47),WEIGHT(3)
DIMENSION XNIE(1),ANIMO(8)
EQUIVALENCE (MMN,COMCN(519))
EQUIVALENCE (MMN,COMCN(607))
EQUIVALENCE (FAISE,COMCN(655))
EQUIVALENCE (INDN,COMCN(727))
EQUIVALENCE (TCST,COMCN(4331))
EQUIVALENCE (P,COMCN(7801))
EQUIVALENCE (TUT,COMCN(606))
EQUIVALENCE (WEIGHT,COMCN(807))
EQUIVALENCE (SCLSY,COMCN(815))
EQUIVALENCE (LAMAT,TFD,ASN,COMCN(823))
EQUIVALENCE (TFN,ASN,COMCN(904))
EQUIVALENCE (TNM,AMN,COMCN(985))
EQUIVALENCE (TENSION,ASN,COMCN(996))
EQUIVALENCE (ASKB1,AN1,COMCN(1007))
EQUIVALENCE (AN2,COMCN(154))
EQUIVALENCE (P,COMCN(1101))
EQUIVALENCE (AMDER,COMCN(1))
EQUIVALENCE (AHORDER,COMCN(145))
EQUIVALENCE (MORDER,COMCN(299))
EQUIVALENCE (MOLEF,COMCN(297))
EQUIVALENCE (MORD,COMCN(305))
EQUIVALENCE (NIN,COMCN(465))
EQUIVALENCE (NMOD,COMCN(473))
EQUIVALENCE (NOUT,COMCN(481))
EQUIVALENCE (NSYSTEM,COMCN(489))
EQUIVALENCE (NTYPE,COMCN(497))
EQUIVALENCE (NUE,COMCN(505))
EQUIVALENCE (IEO,COMCN(508))
EQUIVALENCE (IEETR,COMCN(510))
EQUIVALENCE (IESTAB,COMCN(511))
EQUIVALENCE (NISE,COMCN(512))
EQUIVALENCE (KNIST,COMCN(514))
EQUIVALENCE (KPARAM,COMCN(515))
EQUIVALENCE (NORD,NND,IMAX,COMCN(517))
EQUIVALENCE (NUMAL,N,COMCN(518))

145
THE BLANK SUBROUTINE CALCULATES THOSE ELEMENTS OF THE A AND THE RH MATRICES THAT ARE FUNCTIONS OF P(*)

CALL BLANK
CALL COST
KNTISE=KNTISE+1

BEGIN ITERATION OF ISE TERMS

T0T1SE=0.0CC
DI 99999 NANNISE=1,ISETRM
IF(ICOST(NANNISE).LT.1.OR.ICOST(NANNISE).GT.2)GO TO 99999
K1=N1(NANNISE)
K2=NMOD(NANNISE)
K3=NFUT(NANNISE)
K4=NSYST(NANNISE)
K5=NTYPE(NANNISE)

FINAL ITH-SYSTEM CALCULATIONS (DETERMINATION OF TRANSFER FUNCTION-TFNUM/TFDEN)

TFNUM(I)=TFNUM(I)
DI 51 I=I,IF
MURD(I,K3)=MURDER(I,K1,K4)
DI 51 K=K,11

TFDEN(I)=TFDEN(I)
DI 51 I=I,IF
MURD(I,K3)=MURDER(I,K1,K4)
DI 51 K=K,11

ADJUST FOR INPI (BOTH SYSTEM AND MODEL)

THE CALCULATION OF ISE IS SET UP FOR A STEP INPUT
THE INPUT IS NOT A STEP, BOTH THE SYSTEM AND THE
MISTE MANIPULATED BY MULTIPLYING EITHER NUMER
DENOMINATOR OF BOTH 5 OR 5**2 OR ...
CONTRAILS

93 TFDEN(I)=TFDEN(J)
   J=K-1
   DO 96 I=1,J
96   TFDE(1)=0.0C0
   UR(DIM)=UF2DOM*(K+1)
   KM=ORDU(J+1)
   UJ=K+1
   I=J+I+1
97   TEMPO(I)=TEMPO(J)
   J=K-1
   DO 98 I=1,J
98   TEMPO(I)=C.C0
   UJ=J+UJ
9J   WRITE(6,6560)(K,K)
6560 FORMAT(30HERROR AT 6C--FOR INPUL 1/1S**12,*), DENOMINATOR IS ALR.
   EASY TO RIG--SYSTEM*12)
   STOP 60
657 FWRITE(6,6561)(K,K2)
6561 FORMAT(30HERROR AT 67--FOR INPUL 1/1S**12,*), DENOMINATOR IS ALR.
   EASY TO RIG--MUDEL*12)
   STOP 67
658 WRITE(6,6562)(K,K2)
6562 FORMAT(3C0EA,84,NC,ND)
6563 FORMAT(1I10,MODEL NUMERATOR ORDER = *12,
   *11'C*MODEL DENOMINATOR ORDER = *12,
   *11'C*SYSTEM NUMERATOR ORDER = *12,
   *11'C*SYSTEM DENOMINATOR ORDER = *12,
   *11'C*AT LEAST ONE OF THE ABOVE IS TCC RIG*1)
   STOP 46
6571 IF ((NA.GT.10).OR.(NB.GT.10).OR.(NC.GT.36).OR.(ND.GT.36)) GO TO 98
130 NA=KOMND(1)
   NB=KOMND(2)
   NC=KOMND(3)
   ND=KOMND(4)
   C=CKECK FOR FREE S'S IN NUMERATOR AND DENOMINATOR AND CANCELL I- POS
   C
133 IF (A(XTFCDE)) GT 1.0 .E-014U TO 110
   IF (A(XTFAK)) GT 1.0 .E-014U TO 1C0
   DD 106 L=2,NC
   J=I-1
136 IF (N(JM)) TCNUM(I)
   GO 107 L=2,ND
137 J=I-1
139 TFDE(1)=TFDEN(1)
   NC=NC-1
   ND=ND-1

148
6109 FORMAT(1X,'CHECK AT 106-- THERE IS A FREE S IN THE DENOMINATOR AND NO
FREE S IN THE NUMERATOR OF THE SYSTEM-- INPUT ACCOUNTED FOR*)
STOP 106
110 IF (ABS(TEMPOM(I)),GT,1,E-6) GO TO 120
IF (ABS(TEMPOM(I)),GT,1,E-6) GO TO 119
GO TO 116
115 J+=+1
116 TEMPOM(I)=TEMPOM(I)
GO TO 117
117 TEMPO(M,J)=TEMPOM(I)
NA=NA+1
GO TO 116
119 WRITE(3,E115)
119 FORMAT(* CHECK AT 116-- THERE IS A FREE S IN THE DENOMINATOR AND NO
FREE S IN THE NUMERATOR OF THE MODEL-- INPUT ACCOUNTED FOR*)
STOP 116
C
C BEGIN ROUTE TEST
C
C TEST THE ODD OR EVEN NUMBER OF COEFFICIENTS & DEFINE JMAX
C
120 JMAX=JMAX+1
IF((-1)**JMAX,LT,6) JMAX=JMAX+1
JMAX=JMAX/2
C
C LOAD FIRST TWO ROWS OF ROUTE ARRAY
C
70 306 J+=1,JMAX
K1=JMAX-2-J+2
K2=K1-1
K1,J1)=ASD(K1)
IF(K2,E,GT,6) GO TO 305
R(2,J1)=C,CC
GO TO 306
305 R(1,J1)=ASD(K1)
310 CONTINUE
C
C COMPUTE ROUTE ARRAY & CHECK STABILITY AFTER EACH ROW IS ESTABLISHED
C
125 IF (ABS(R(1,1,1)),GT,1,E-6) GO TO 325
WRITE(6,E325)
2325 FORMAT(* THE 1ST ENTRY IN THE ROUTE ARRAY IS ZERO*)
GO TO 325
325 KK=K(2,1,J+2(1,1)
GO TO 330
325 R(I,J,MAX)=C,CC
R(I,J,MAX)=C,CC
JMAX=JMAX-1
IF (KK,LE,6.0) GO TO 325

Approved for Public Release
CONTROLS

K3=1-2
K4=J+1
K5=I-1

R(X,Y,Z)=R((X,Z,K4)-R(K3,Z)*R(K5,Z))/R(K5,Z)
R=0.0111/R(K5,Z)

CONTINUE

IF (Y.GT.0.1G0) GO TO 350

WRITE(6,632)

FORMAT(*TEST FAILED. SYSTEM IS UNSTABLE*)

WRITE(6,6096)(1,1:I=1,10)

IF (Y.GT.0.1G0) GO TO 350

WRITE(6,632)

FORMAT(*THE FOLLOWING TRANSFER FUNCTION(S) HAVE BEEN ADJUSTED

END OUTPUT TEST

FINAL SYSTEM CALCULATIONS (ISE POLYNOMIAL COEFFICIENTS)

COMPLETE COST FUNCTION

AISP(N,NP)=C.CC

BEGIN EVALUATION OF THE ERROR EXPRESSION

SCALE SYSTEM STEADY STATE TO MODEL STEADY STATE

IF (ABS(SAMX(1:1:11).GT.1.E-4)GO TO 350

IF (ABS(SAPX(1:1:11).LT.1.E-4)GO TO 350

WRITE(6,632)

FORMAT(*THE MODEL AND THE SYSTEM CANNOT BE MADE TO HAVE EQUAL STE

ADY STATE RESPONSES TO A STEP*)

GO TO 335

IF (ABS(SAMX(11).GT.1.E-4)GO TO 364

50 IF (ABS(SAPX(11).LT.1.E-4)GO TO 362

WRITE(6,632)

FORMAT(*THE FORWARD LOOP GAIN IS ZERO*)

GO TO 335

IF (ABS(SAMX(11).GT.1.E-4)GO TO 352

GO TO 375
CALL PMPYI(N1,N1,AMK,NA,ASD,ND)
CALL PMPYI(N2,N2,ASK,NC,ASD,NR)
CALL PSUBL(*U,AN1,A1,AN2,A2)
EU 406 J=1,NU
NU1 = NU + 1 - J
IF (ABST(U(NU1)) .LT. 1.0E-4) GO TO 465
JORDEN=NU1
GO TO 467
405 JORDEN=NU-J
406 CONTINUE
407 CONTINUE
NU=JORDEN
CALL PMPYIV,NU,AMD,NR,ASD,NO)
U(I) MUST BE ZERO
U(I)=0.0CC
END EVALUATION OF THE ERROR EXPRESSION
C SCALE V TO PREVENT VMAT INVERSION OVERFLOW
SCALE=1.0CC
DO 488 I=1,NU
488 SCALE=SCALE*ABS(V(I))
489 SCALE=SCALE/NU
SCALE=EXP(-SCALE)
GO TO 492 J=1,NU
492 V(I,J)=V(I,J)/SCALE
C ESTABLISH *V* MATRIX (VMAT)
C NV1+NV-1
DO 500 J=1,NV1
500 DO 504 I=1,NV1
I=NV1(J-1)
K=NV1-J+1
IF (I .GE. NV1) GO TO 502
IF (J .GE. NV1) GO TO 503
502 AMAT(I,J)=0.00
GO TO 504
503 AMAT(I,J)=V(I,K)*(-1.0)**(I+J)
504 CONTINUE
C ESTABLISH *V* COLUMN VECTOR (VV)
C K1=NV1-2
DO 905 I=1,K1
905 VV(I)=0.00
K2=NV1-1
DO 620 J=1,K
   K3=K+1-J
   IF(K3.LE.0.OR.K3.GT.NL) GO TO 620
   A*K=ZD**2**R*(U(KA)+TERM2*UGV*V(KJ))*(L(KC)-TERM2*UGV*V(KJ))
   V(KA)+A*K
   CONTINUE
   A*K=K0*(V(V1)***V(V1))
   AISE(NNNISE)=AISE(NNNISE)+A*K
   IF(GUIST(NNNISE).LE.2) GO TO 99999
   AISE1=AISE1+AISE(NNNISE)
   AISE=AISE1/NNUIS
   AISE1=0
   AISE=NNUIS=AISE1
   VV=VV+TUTI*TUTI*AISE(NNNISE)**WEIGHT(NNNISE)
   ISTAH=1
   RETURN
   END
CONTRAITS

Q(K) = Q(K-1)/FCTR
IF(NRF(K-1)=7) 2126,2128,2131
2126 RF(K) = RF(K-1)/FCTR
2128 NRF(K) = NRF(K) + 1
GU TO 2122
2130 IF(ARS-1.E-20*) 2132,2165,2165
2132 IF(NRF(K-1)) 2135,2165,2134
2134 P(K) = P(K)*FCTR
Q(K) = Q(K)*FCTR
IF(NRF(K-1)=7) 2136,2138,2138
2136 RF(K) = RF(K-1)*FCTR
2138 NRF(K) = NRF(K) - 1
ARS = ARS(P(K)) + ARS(Q(K))
GO TO 2136
2140 CONTINUE
2140 IF(AR*21) 2142,215C,2190
2142 DIV = P(2)*P(2) + Q(2)*Q(2)
DIV = DIV/2
2190 NXT = NRF(K-1) - NRF(K-1)
IF(NXT) 2192,220C,220C
2192 NXT = NXT + 1
T2 = 1./FCTR
GU TO 2202
2200 T2 = FCTR
2202 GU 2204 X = 1.,NRT
DX = DX/T2
2204 VY = DX/T2
2206 IF(ABS(CX) + ABS(DY)) - ACHGJ 2207,2207,2207
2207 DX = SIGN(ACHGJ/CCOS,.DX)
UY = SIGN(ACHG/JCCOS,DY)
IVSW = -2
220C XA = XR + CX
YR = YR + CY
ARS = ARS(CX) + ARS(DY)
IF(AI*E-61) 2216,2216,2218
2216 IF(AI*AH.I) 2217,221F,2218
2218 IWS = IWS+1
IF(IWS) 2227,2227,2227
2227 IWS = IWS+2
2229 IF(NFI) 2226,2226,2226
2226 NFI = 1
X1 = XR
Y1 = YR
K1 = 2
GO TO 2114
2114 ARL = ARS
2118 IF(NFI) 2230,223C,223C
2230 N1 = N1 + 1
LI = 0
IF(N1-500) 2232,2232,2232
2232 IF(N5G-81) 2110,2234,2234

155

Approved for Public Release
IF(ABS(RX-XT)+ABS(YH-YT)<+.1) 2205, 2205, 2205

RI(1) = X*R*FCTR
DY 2205 T=1, NY

IF(ABS(RX(1)-RX(1))+ABS(RY(1)-RY(1))-TOL) 2280, 2280, 2280

CONTINUE
GO TO 2205

IF(NSIG=4) 2110, 2290, 2290

X = XT
Y = YT
GO TO 233C

Z = RX
Y = RY

IF(ABS(Y-1.E-4) 2305, 2315, 2315

C = 0.
GO TO 2310 T=1, NT

A(1,1) = A(1,1)+X*A(1-1,1)
GO TO 2325

T = T+X

IF(1) = -(X*X) + Y*Y

A(1,1) = A(1,1) + T*A(1,1)

GO TO 2320 T=3, NT

A(1,1) = A(1,1) + T*A(1-1,1) + O1V*A(1-2,1)

X = X*R*FCTR
Y = Y*R*FCTR

IF(1) = RX(2) 2340, 2340, 2340

(I1) = [RX(1) + 1

IF(I1) = RX(1) = [RX(1) + 1C4E576

IF(I1) = Y(2) 2345, 2355, 2355

(I1) = TRY(1) + 1

CONTINUE, 2355, 2355, 2355

ISY(1) = Y(1) + C4E576

RTX(1) = X

RY(1) = Y

IF(Y) 2365, 2365, 2365

NT = 1
GO TO 240C

NT = 2

RX(2) = X

RTY(2) = Y

N = N-NT

GO 2470 T=1, NT

J = 1

JFL = 0

Approved for Public Release
2435 IF(JFL) 2440,2440,2465
2436 IF[(RY(J) RY[(1)]) (RX(J) RX[(1)]) (RTY(J) RTY[(1)])]
2437 IF(0) 2444,2444,2444
2438 RX(J) = RX[(1)]
2439 RY(J) = RY[(1)]
2440 RMUL(J) = PMUL(J)
2441 K = K1
2442 J = JFL
2443 K = NR+1
2444 IF(JFL) 2445,2445,2445
2445 RX(J) = RX[(1)]
2446 RY(J) = RY[(1)]
2447 J = J + 1
2448 IF(JFL) 2447,2447,2460
2449 J = JFL
2450 RMUL(J) = 1.
2451 K = NR+1
2452 J = J + 1
2453 RX(J) = RX[(1)]
2454 RY(J) = RY[(1)]
2455 J = JFL
2456 RMUL(J) = 1.
2457 J = J + 1
2458 RX(J) = RX[(1)]
2459 RY(J) = RY[(1)]
2460 J = J + 1
2461 RX(J) = RX[(1)]
2462 RY(J) = RY[(1)]
2463 J = J + 1
SUBROUTINE MNPY (X, IDIMX, Y, IDIMY)

DIMENSION Z(IDIMZ)

IF (IDIMX*IDIMY .GT. 0) GOTO 20

IDIMZ = 0
GOTO 30

20 IDIMZ = IDIMX*IDIMY
GOTO 30

30 Z(1) = 0.

40 I = 1, IDIMX
50 J = 1, IDIMY
K = I + J - 1
60 Z(K) = Z(K) + Y(J)*X(I)
70 RETURN

END
SUBROUTINE POLPRO

THIS SUBROUTINE MULTIPLYS NNN POLYNOMIALS TOGETHER

POLYNOMIALS CAN ALL BE IN ASCENDING ORDER OR CAN ALL BE IN DESCENDING ORDER. THE NNN1*1 PRODUCT POLYNOMIAL WILL BE IN THE SAME ORDER AS FACTOR POLYNOMIALS.

GLOBAL VARIABLES ARE PASSED BY THE PAN COMMON BLOCK

COMMON COMON(4400)
DIMENSION ANZEI,ATERM(47,47),NNORD(10),TEMP(81)
EQUIVALENCE (ANZ,COMON(704))
EQUIVALENCE (ATERM,COMON(205))
EQUIVALENCE (TEMP,COMON(677))
EQUIVALENCE (NNORD,COMON(477))
EQUIVALENCE (NNORD,COMON(508))
EQUIVALENCE (NNN,COMON(518))

DJ 5 I=1,NNN
IF(NNORD(I,GE,C))GO TO 5
ANZ(I)=0.CC
ICMR=0
GO TO 73
5 CONTINUE
ICMR=NNORD(I)+1
UU 1C J=1,ICMR
10 ANZ(I,J)=ATERM(I,J)
IF(JNORD(J,GT,0))GO TO 7C
UU 60 L=2,NNN
13 J=NNORD(I)+1
UU 20 I=1,ICMR
20 TEMP(I)=ANZ(I,1)
ICMR=ICMR
IF(I TEMP,I,FMI.GT.0)GO TO 40
ICMR=0
GO TO 70
40 I M R=I M R+1
UU 5C I=1,ICMR
50 ANZ(I)=0.CC
UU 60 I=1,ICMR
UU 60 J=1,ICMR
K=I+1
60 ANZ(K)=TEMP(I)*ATERM(L,J)*ANZ(K)
70 RETURN
ENC

159

Approved for Public Release
SUBROUTINE PSUB (Z, IDIMZ, X, IDIMX, Y, IDIMY)

C THIS I RM-3SP SUBROUTINE SUBTRACTS ONE POLYNOMIAL FROM ANOTHER

C

DIMENSION Z(I), X(I), Y(I)

IDIMX = IDIMX
IDIMY = IDIMY

DO 10 I = 1, IDIMX
   IF (I.LE.IDIMY) GO TO 20
   10 CONTINUE

RETURN

END
SUBROUTINE SIMQ(N, N+K5)
DIMENSION A(J), B(J)

C FORWARD SOLUTION
C
IUL=0.0
K5=0
JY+1
J0=J-JY+1
SICA=0
1T=J-J0
C03D(1)=J-Y+N

C SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN
C
1J=IT+1
20 IF(ABSB(1J)>ABS(A(1J))>&10.32,30
30 PICA=A(1J)
40 IMAX=1
50 CONTINUE
C
C TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)
C
60 IF(ABSB(1J)>TOL)155,35,40
70 KS=1
80 RETURN
C
C INTERCHANGE ROWS IF NECESSARY
C
1J=J+M1(J-2)
2T=IMAX-J
30 S5OK=J-Y+N
40 I1=I1+1
50 SAVF=A(11)
60 A(11)=A(12)
70 A(12)=SAVE
C
C DIVIDE EQUATION BY LEADING COEFFICIENT
C
90 A(11)=A(11)/B1GA
100 SAVE=SAVMAX(J)
110 B1J=SAVE/B1GA
C
C ELIMINATE NEXT VARIABLE
C
120 IF(1J=N155,75,95
130 IQS=N1(J-1)
140 C065IX=J-Y+N

161
\begin{align*}
\text{IXJ} & = [\text{IS}+1]X \\
\text{IT} & = Ix \\
\text{D} & = 60.3X + JY + 0.068 \\
\text{IJK} & = \text{IXJ} + 1 \\
\text{AIJX} & = \{\text{IXJ} \times 1\} + 1X \\
\text{JJK} & = \{\text{IXJ} \times 1\} \\
\text{A} \{(\text{X} + 1)J + \{\text{X} \times 1\} \} & = \{\text{X} \times 1\} - \{\text{X} \times 1\} \\
\text{B} & = \{\text{IX} \times 1\} - \{\text{X} \times 1\} \\
\text{C} & = \text{BACK SOLUTION} \\
\text{C} & = \text{BACK SOLUTION} \\
\text{NY} & = N - 1 \\
\text{N} & = \text{N} \\
\text{A} & = A + 1 \\
\text{B} & = B + 1 \\
\text{C} & = C + 1 \\
\text{AI} & = I - A + \{\text{A} \times 1\} + B + \{\text{C} \times 1\} \\
\text{A} & = A - \text{N} \\
\text{B} & = B - \text{C} - 1 \\
\text{RETURN} & \\
\text{END} & \\
\end{align*}

162
SUBROUTINE KIN(F,X,N)
C DESCRIPTION OF PARAMETERS
C F = SINGLE VARIABLE CONTAINING THE FUNCTION VALUE ON
C RETURN, I.E., FP(X)
C X = VECTOR OF DIMENSION N CONTAINING THE INITIAL
C ARGUMENTS WHERE THE ITERATION STARTS. ON RETURN,
C X HOLDS THE ARGUMENT CORRESPONDING TO THE COMPUTED
C MINIMUM FUNCTION VALUE.
C N = NUMBER OF VARIABLES.
C REMARKS--
C THIS SUBROUTINE WAS ORIGINALLY PROGRAMMED BY MIKE
C JUGERMAN TO NUMERICALLY SOLVE A GENERAL CONSTRAINED
C MINIMIZATION PROBLEM VIA MEANS OF A PATTERN SEARCH
C TECHNIQUE (REF. G. J. WILDE, OPTIMUM SEEKING METHODS).
C THE ROUTINE HAS BEEN MODIFIED FOR PARIKE PILOT AND IS
C USED TO FIND A SET OF PILOT PARAMETERS FOR WHICH THE
C SYSTEM IS STABLE, INSTEAD OF SEARCHING FOR A MINIMUM
C VALUE OF THE COST FUNCTION, F, THE SUBROUTINE IS EXITED
C WHEN F IS LESS THAN C. THE COST FUNCTION USED IS
C THE MAXIMUM OF THE REAL PARTS OF THE EIGENVALUES OF
C THE A MATRIX. THIS WHEN THE COST FUNCTION GOES STRICTLY
C NEGATIVE, THE A MATRIX IS ASYMPTOTICALLY STABLE.
C SUBROUTINE CALLED
C COST (F,X,N)
C C Cost IS A USER WRITTEN SUBROUTINE. IT MUST BE
C WRITTEN SO THAT FOR A GIVEN VALUE OF THE ARGUMENTS, X,
C THE FUNCTION VALUE, F, AND THE VECTOR CONSTRAINT FUNCTION
C VALUES ARE RETURNED. N AND M MUST NOT BE MODIFIED
C BUT CAN BE USED TO INDICATE THE DIMENSION OF X.
C COMMON DLME(75), MSKIP
C DIMENSION X(N),TX(I),STEPX(I),X(I),A(I),DELX(I),LNG(I)
C IF(MSKIP.EQ.0)RETURN
C IF(N.GT.5) GO TO 100
C WRITE(*,6,999)
C 9999 FORMAT(1X,10LINNER MIN)
10 DO 1 I=1,N
UDELX(I) = XP5
TX(I) = X(I)
STEPX(I) = 0
LNG(I) = CEX(I)
2 CONTINUE
CENVM = 5,
EPS = 0.0001
IFER = 0
FTEST=1.0/(2.0**UENC)
KOUNT=0
FACTOH=1.0
NMIN=N
CALL COST(F,X,NMIN)
3 WRITE(6,4421)

163
FORMAT(1X,16HINITIAL POINT IS)
WRITE(6,25)F,F(X(I),I=1,N)
TF=TF
75 GO TO 1
500 TX(I)=X(I)+DELX(I)
CALL COST(TPMA,IX,NMIN)
IF(TPMA.GT.TF) GO TO 473
8 A(I)=1.
10 X(I)=TX(I)
TF=TPMA
IF(TF.LT.0.) GO TO 437
GU=GU+2C
4 TX(I)=X(I)-DELX(I)
CALL COST(TPMA,IX,NMIN)
IF(TPMA.GT.TF) GO TO 555
10 A(I)=1.
GO TO 18
555 TX(I)=X(I)
50 CONTINUE
WRITE(6,800)
400 FORMAT(1X,7HINNER SEARCH MOVE)
WRITE(6,25)TF,F(X(I),I=1,N)
ITER = ITER + 1
ATF=TF=1CCCCCC.
AF=AF=10000CC.
FX=ABS(AF-AF)
IF(FIX.LE.E,C) GO TO 22
IF(AF+GE,AF) GO TO 22
VAL=ABS(TF-F)
IF(VAL.EPS) 66,66,24
66 F=TF
WRITE(6,26)
26 FORMAT(1X,4HDELTA COST LESS THAN E'S FINAL VALUES ARE )
READ 88 WRITE(6,25)F,(X(I),I=1,N)
25 FORMAT(1X,7E12.3)
RETURN
417 F=TF
WRITE(6,555B)
9418 FORMAT(1X,2E=STABLE STARTING PARAMETERS)
WRITE(6,25)F,(X(I),I=1,N)
RETURN
24 DO 27 L=1,N
STEP(L)=X(L)-TX(L)+STEPX(L)
TX(L)=X(L)+STEPX(L)
X(L)=TX(L)
A(L)=0
TX(L)=X(L)
27 CONTINUE
F=TF
KOUNT=0
CALL COST(TF,X,NMIN)
GO TO 513

364
512 IF = 3,
513 WRITE(6,475)
885 FORMAT(1X,12hPATTERN MOVE)
WRITE(6,25) TF, (X(I), I=1, N)
IF(TF.LT.-1) GO TO 437
IF(TF .GT. 2.5) RETURN
GO TO 75
2 KOUNT = KOUNT + 1
GO 30 1 = 1, N
X(I) = X(I) - STEP * (1) - A(I) * DELX(I)
50 CONTINUE
WRITE(6,488)
888 FORMAT(1X,37hREVERT TO LAST GOOD POINT AND RESTART)
45 IF(KOUNT .LT. 2) 40, 41, 41
40 TF = F
DO 42 IJ = 1, N
STEPX(IJ) = C
X(IJ) = X(IJ)
MODX(IJ) = X(IJ)
42 CONTINUE
GO TO 75
41 IF(FACOR .LT. 100, 100, 100, 100)
47 FORMAT(1X,50hCREEP STEP LESS THAN ORIGINAL/100.FINAL VALUES ARE)
GO TO 88
52 FACTOR = FACTOR / 2.
WRITE(6,477)
489 FORMAT(1X,21hREDUCING STEP SIZE TO)
DO 111 KM = 1, N
DELX(KM) = 1.0 * KM + FACTOR
STEPX(KM) = C
X(KM) = X(KM)
MODX(KM) = X(KM)
111 CONTINUE
TF = F
GO TO 75
100 WRITE(6,477)
RETURN
111 FORMAT(1X,71h**SURROGATE MINI - N GREATER THAN 5 - CORRECT DIMEN
SION STATEMENT** )
END
SUBROUTINE COST(F,X,N)
COMMON COMDEN(4,CC), COMX(5,16CC)
COMMON /S/A(1599),B(16CC)
EQUIVALENCE (NUMDEN,COMDEN(518))
EQUIVALENCE (NUMFG,COMCN(505))
EQUIVALENCE (ILO,COMCN(508))
EQUIVALENCE (MIND,COMCN(505))
EQUIVALENCE (ACORDER,COMCN(1))
EQUIVALENCE (TFUN,COMCN(504))
EQUIVALENCE (ROTOR,COMCN(1059))
EQUIVALENCE (MTPD,COMCN(1055))
EQUIVALENCE (ASKRI,COMCN(1007))
EQUIVALENCE (PO,COMCN(700))
DIMENSION ADDP2(37)
DIMENSION ASKRIJ(12,12,11)
DIMENSION MTPD(12,12),MTPD1(12,12,11,15)
DIMENSION RMULT(37),MTPD1(36),ROTOR(36),TDEN(81),X(15)
INTEGER ACORDER
DATA IOL,IC1/.CCCN/
DJ 1=1,N
49 PI(J) = XI(J)
40 D0 I=1,3E
2 ADTFUN(I) = -1.
CD 6 I=1,12
D0 J=1,12
4 MIKAI(I,J) = -1
CALL HLANK
160 IQ=NUMFG
DO 49 I=1,160
DO 49 J=1,160
MURD(I,J) = ACORDER(I,J)
49 X=1,11
49 X=1,11
49 ASKRIJ(I,J,K) = A(NUM)
CALL SKRJ
DO 43 I=1,11
43 D1 = 11
1 CONTINUE
RETURN
END

Approved for Public Release
C THIS PROGRAM CALCULATES THE SS FREQUENCY RESPONSE OF A TRANSFER
C FUNCTION INPUT AS A NUMERATOR/DEMONINATOR POLYNOMIAL RATIO.
C USEM INPLS ORDERS NUM & DEN; EQUFS & EQPS IN DESCENDING ORDER.
C
C*******************************************************************************
COMMON / 6 / SAVE(310)
DIMENSION STNLM(0,37), OMEGA(21), TFDEM(37), ISEAUX(8), IORDER(5)
DIMENSION CUFN(37), CCEF(37), TITLE(10)
EQUIVALENCE (STNLM(1), SAVE(1))
EQUIVALENCE (IORDER(1), SAVE(1))
EQUIVALENCE (NORDEN,COMON(517))
EQUIVALENCE (INDLCS, COMON(725))
EQUIVALENCE (NISEAP, COMON(4320))
EQUIVALENCE (TFDEM, COMON(4320))
EQUIVALENCE (CUFN, COMON(7483))
EQUIVALENCE (ISAP, COMON(1321))
EQUIVALENCE (TITLE, COMON(7953))
EQUIVALENCE (TFDEM, COMON(233))
EQUIVALENCE (CUFN, COMON(4296))
EQUIVALENCE (ISAP, COMON(4345))
EQUIVALENCE (CCEF, COMON(4311))
REN MAGN, MAGNLR
COPPLFX GNUM, GCEN, GTF. S
COMON(725)=COMON(4311)
CALL PRAIE)
WHITIE6, S0(01)
3000 FORMAT(1, 5(1, 42X, 47(*)) / 42X, *10X, THE CALCUL
ATES PILOT RATING IS = Fe. 3/4X, *10X, 42X, 47(*))
IF(NISEAP. 0.4J4.4D0 TO 995
NORDEN=NORDEN
MENDEN=MENDEN+1
DO 4 K=1, NCMER
4 C=(CMER-K+1)=TFDEM(K)
M+1
C FINISH OUT WHICH ISE TERM
DO 700 L=1, 8
IF(L. NE. ISEAUX(M)) GO TO 699
CALL CLASS(INOLCS)
WHITIE6, ECCC(TITLE)
6000 FORMAT(1, 20X, 706)
WHITIE6, ECOFL
632 FORMAT(1, 30X, 5F4.0, FREQUENCY RESPONSE FOR ISE NO. *1, 2)
1 CONTINUE
NUM=IORDER(L)
HUM=HUM+1
WHITIE6, CINUM
410 FORMAT(2X, THE INFLUENT NUMERATOR POLYNOMIAL IS OF ORDER *1, 2)
DO 3 K=1, MLM
3 CUFLM(1, NUM-K+11)=STNLM(L, K)
WHITIE6, 411(1, CUFLM(11), 71)=MLM
}

Approved for Public Release
FORMAT(2X,10HITS COEFFICIENTS IN DESCENDING ORDER ARE=,6(3X,1PE15.
+.7))

WRITE(6,412)INDEN

FORMAT(2X,THE INPUT DENOMINATOR POLYNOMIAL IS CF ORDER =,12)
WRITE(6,411)COEFD(JJ),JJ=1,MDEN
WRITE(6,66C)
LI 132 IF=1,FREQ
S=COMPLEX(C*,OMEGA(I))
GNUM=COEFD(JJ)
IF(GNUM.LT.1.0)GO TO 6C
UD 90 J=2,MDM
50 GNUM=COEFD(JJ)*GNUM*S
50 CONTINUE
GDDN=COEFD(JJ)*GDDN*S
70 CONTINUE
CTF=GNUM/GDEN
MAGN=CARC(CTF)
MAGNB=20.,*LOGIC(MAGN)
PHASE=57.295782*ATAN2(1.0,MAGN)*REAL(CTF)
100 WRITE(6,6C)JOMEGA(I),JTF,MAGN,MAGNB,PHASE
132 CONTINUE
530 FORMAT(2X,F10.6X,*OMEGA*,6X,*REAL PART*,5X,*IMAG PART*,5X,*MAGN*,9X,*IMAGN*)
531 FORMAT(2X,F10.4,1X,*5(3X,1PE11.4))
AX=1
IF(AX.GTNIS=AP)GO TO 701
699 CONTINUE
700 CONTINUE
701 CONTINUE
CALL TIMRP
999 CONTINUE
ILNK=5
C CALL MATLOC
CALL NVRAY
STOP
END
VARIABLE NAME DICTIONARY

COEFN IS THE COEFFICIENTS OF THE NUMERATOR IN DESCENDING ORDER
KORDN IS THE ORDER OF THE NUMERATOR
COEFD IS THE COEFFICIENTS OF THE DENOMINATOR IN DESCENDING ORDER
KORDE IS THE ORDER OF THE DENOMINATOR
R IS THE REDUCTION FACTOR
NOTE THAT Y IS YDOT(1), THE FIRST DERIVATIVE OF Y IS YDOT(1), ETC.

COMMON / 6 / SAVE(31)
COMMON / CINT / 1, HMAX, HMIN, EMIN, EMAX, X5(8), HCUTERR, IP, IV,
! J, HRON, IFIN, IVAL, IAD(8), IND(8), J
DIMENSION TIME(66), RR(66)
DIMENSION COEFN(37), COEFD(37), TITLE(110)
DIMENSION SYNUM(32), ISEND(8), IORDER(8)
DIMENSION IPLOT(400,11), MPLOT(400,11), CPLOT(400,11), MPAY(11), AARRAY(11,1
1 TITLE(10)
EQUIVALENCE (STNUM,1,17,SAVE(15))
EQUIVALENCE (IORDER(1),SAVE(I))
EQUIVALENCE (DCOMN(556))
EQUIVALENCE (MAXI,SCOMN(507))
EQUIVALENCE (TIME,SCOMN(517))
EQUIVALENCE (INDEL,SCOMN(729))
EQUIVALENCE (INTEAP,SCOMN(4320))
EQUIVALENCE (NNAME,SCOMN(4329))
EQUIVALENCE (ISAVE,SCOMN(4321))
EQUIVALENCE (TITLE,SCOMN(759))
EQUIVALENCE (TUFEN,SCOMN(823))
EQUIVALENCE (TIME,SCOMN(4176))
EQUIVALENCE (RRR,SCOMN(4236))
EQUIVALENCE (EMON,SCOMN(4379))
DATA (TITLE,1,10)/5*10H, 10HVS TIME
1 4*10H /
DATA TINO/10C INPLT, 10H OUTPUT /
DATA AARRAY(1)/1M/A/
HMAX=HMAX1
NDE=NDEN
N=NDEN+1
NCOFFC=NDEN+1
& I=1, NCOEFF
4 COEFFU=NCOEFF*(1)=TFD(1)(IK)
1 IF(HMAX+LFC.*HC+HAMAX+G1+CC).HMAX++.02
HMIN=E-6
EMAX=1.0-4
IP=5
IV=0
C IV=0 IMPLIES A VARIABLE INTERVAL SIZE CONTROLLED BY INTEG AND
CONTINUE
YDOTT(I+1)=YDOTT(I)+YDOTT(AA)
NN=NN+1

120 CONTINUE
YDOTT(I+1)=YDOTT(I)
NN=NN-1

121 CALL INTEGYDOTT(NN),YDOTT(NN-1)

122 CONTINUE
IF(KUNNT. NE. E) GO TO 121
IF(PK. NE. C) GO TO 150
c=CCP(I)(KUNNT)*YDOTT(I)
IF(KUNNT. EQ. C) GO TO 170
KK=K
DD=1.0E4K
C=C+CCP(I)*YDOTT(KK)
KK=KK+1

120 CONTINUE
IF(KUNNT. EQ. ACC) GO TO 140
KUNNT=KUNNT+1
CPLOT(KUNNT,1)=C
PLOT(KUNNT,1)=F
RM=RM(KUNNT,1)+K

140 CONTINUE

150 CONTINUE
DD=1.25E6+K

155 FMAT(1)=5X,9=F6.1,3X,*R0,F12.5,3X,*C0,F12.7

150 CONTINUE
CALL UPDATE(DUMMY,DUMMY)
IF(TI. GT. TIME1) MC TC 190
GO TO 90

190 CONTINUE
CALL ARRAY(I)=KUNNT
TITLE(I)=TI
CALL PLOT1(I,PL,ARRAY1,7,0,10,TITLE,400,61)
TITLE(I)=TI
CALL PLOT1(I,PL,ARRAY1,7,0,10,TITLE,400,61)
CONTINUE

190 CONTINUE
IF(MM. GT. Wi) RETURN
RETURN
SUBROUTINE PCLT ( YD, Y )
DIMENSION XMN1(f),XM(f),
COMMON / IXVAM / COMMTU1
COMMON / IZ23RF / IPTAIL, IPTOL, LJCKAP(50)
COMMON / CINT / T, VMAX, VMIN, EMIN, EMAX, XS(X), HZ, CUTF1ER,
1 IP, IVARH, IMTH, IPTAT, IFIN, IVAL, IMAX1, J
2
do j
C CI=55/14 C2=9/14 C3=1/6 C4=59/55 C5=37/59
C C6=-54/17 C7=15/5 C8=-5/19 C9=19/270
C DATA C1/2.2511666666666677/C2/3.9357/C3/1.6666666666666677/
C DATA C4/-1.67127272727272/C5/-0.271118440477977/
C DATA C6/-1.2432432432432432C7/2.1111111111111111/
C DATA C8/-1.2631519447436842/C9/0.00370370370370370
ENTRY SETUP
IFPTAIL = C
IFPOTYL = C
IERROR = C
ISTEP = 1
F = 0
IVAL = 0
MC = VMAX2.*(-IP)
IF2=2**IP
IF1=0
IFPNN=1
IFV=0
INC=1
d5=1
101=1
LALP=4
L51=0
NUMP=0
IFINTH=110,10,100
1) IU2=1
JG=7
IU3=2
1SCNTC=0
1BLTA=2
1GAM=1
1FIFIWARM) IC,IC,20
20 IBA=1
IBE=2
G1 NTU 4G
30 IBA=2
IAG=1
40 )1H7=1
G1 NTU 110
100 IB2=2
JG=6
IB3=1
1B7=7
110 H02 = HZ / 2.
H0=02
172
Approved for Public Release
AI=C1*H2
AZ=C2*H2
RCTM=1
RETURN
ENTRY INT3C
GO TO 121,122,26C,24C,185
120 GO TO (J25..125,..46C,125),1 STEP
129 Y(11) = Y(1) +* Y(10)
RETURN
130 Y(11) = Y(2) +* Y(10)
RETURN
133 Y(11) = Y(3) +* Y(11)
RETURN
140 Y(11) = Y(4) + C2*H*YD(4)*2,00*YD(1) + YD(2) + YD(11)
RETURN
160 Y(11) = Y(11) + A1*YD(1) + C4*YD(2) + C6*YD(4))
RETURN
200 CN = Y(2) + A2*YD(1) + C7*YD(2) + C8*YD(3) - YD(4) + 5.0*YD(11))
XM = ABS(Y(1)) - CN1
Y(1) = CN1 + CN1*CN1)
IF THEN = 2 2C6C*2C6C
253 RETURN
260 IF-CN1(12C,2C+,2B0)
280 XM(1) = EPSI XM / CN1)
IF THEN = XM(1) 22C6C,29C,2E5
285 XM(1) = XM1
290 IF THEN = XM(3) 210,3C+,3CC
300 IF THEN = XM(4) 2 MIN 1 VAL = - 93CC
IVAL = IVAL + 1
RETURN
310 IF THEN = XM(2) + 12C3..320,320
320 IVAL = IVAL + 1
3 30 RETURN
ENTRY UPDAT
IF = 1
IF THEN = 1350.24C,39C
400 IF THEN = 2P12
350 IF THEN = 11C4..6C,36C
400 IF THEN = 11D20..37C,7CC
470 I = 1
IPT = 1IPT-1
GJ THEN = 14C,38C,1M2
475 IF THEN = 1B2F4+1
480 IJPF = 1
ISTRP = 1
N = H / 2.
IF THEN = 0
IHE = 1
GJ THEN = 14C,38C,1M3
483 GJ THEN = 531 [MVK = + 1,IP2T] ]
484 [MVK = LUNC, IMV = 1 ]
485 [MVK = ID, ]
173

Approved for Public Release
COMINT( KMVER + 2 ) = COMINT( KMVER + 4 )
COMINT( KMVER + 3 ) = COMINT( KMVER + 5 )
COMINT( KMVER + 4 ) = COMINT( KMVER + 5 )
COMINT( KMVER ) = COMINT( KMVER + 6 )
GJ T0 455

GJ 455 IMVER = 1, IPTOTL
KMVER = LCGNAM( IMVER )
COMINT( KMVER ) = COMINT( KMVER + 5 )
COMINT( KMVER + 1 ) = COMINT( KMVER + 3 )

GJ 455 IPTOTL

400 IF(IPLS1A-1)375.46C,375
400 IA=2
IR5=3
IF(IN=0)
IF(IFVARH14F=35C,465
405 IA=2
GJ T0 390
700 IF(ALP=1ALP-1)
IF(ALP-1)4C,2C,71C
710 ISTEP=1STEP+1
GJ TO 73C
720 IR5=2
730 IF(ALP-1)4C,75C,74C
740 IF(IH=1)
GJ TO 76C
760 H=H
760 GJ TO(77C,78C),IR7
770 GJ 771 IMVER = 1, IPTOTL
KMVER = LCGNAM( IMVER )
COMINT( KMVER + 5 ) = COMINT( KMVER + 4 )
COMINT( KMVER + 4 ) = COMINT( KMVER + 3 )
COMINT( KMVER + 3 ) = COMINT( KMVER + 2 )
COMINT( KMVER + 2 ) = COMINT( KMVER + 1 )
COMINT( KMVER + 1 ) = COMINT( KMVER )
771 COMINT( KMVER ) = COMINT( KMVER + 6 )
GJ TO 455
780 GJ 781 IMVER = 1, IPTOTL
KMVER = LCGNAM( IMVER )
COMINT( KMVER + 3 ) = COMINT( KMVER + 2 )
COMINT( KMVER + 2 ) = COMINT( KMVER + 1 )
COMINT( KMVER + 1 ) = COMINT( KMVER )
781 COMINT( KMVER ) = COMINT( KMVER + 9 )
GJ T0 455
400 IGAMA=IGAM
IF(IGAMA=162C,82C,610
410 IR5=4
T=F+H
GJ T0 78C
420 IR5=3
GJ T0(IPC,140),IR4
430 IPT1=IPT1-1

124
IFN = 0
G0 T0 856
840 IF(IVAL) = 645,665,685
845 IGCN = 0
850 IFCVL = 1,5CC,6CC,855
855 [NDRH=1]
860 [IPT1 = IPT1-1;]
865 [I=I+1]
870 IFN = 0
875 IVAL = 0
880 G0 T0 855
890 IF (ISCNT = 2) 6655,665,8IC
900 ISCNT = ISCNT + 1
905 G0 T0 855
910 IF (2*IP11/2) = 0, IPT1 = GO TO 855
915 IFN = 1 MAXE(860,645,685)
920 IP1 = IP1/2
925 ISCNT = 0
930 IR = IA-1
935 IR = 1
940 ISIP = 1
945 NDRH = 0
950 IPT2 = IPT2 / 2
955 [NUMET = T]
960 IP = IP + 1
965 [IFN = 0]
970 IC
975 A1 = A1 + 1
980 A2 = A2 + HZ
990 G0 T0 855
995 IF (IP11) = 855,6C5,505
1000 IMETA = 3
1005 IAIP = 4
1010 IP = IP + 1
1015 IHM = 1
1020 IF(3) = 1C,82C,510
1025 IF(1H) = 1
1030 RTKIP1 = RTKIP1 + 1
1035 [IPT1 = IPT1]
1040 GO 911 IMVER = 1, IPTOTL
1045 KMKER = LCCNAM(IWVER)
1050 G0 T0 93C
1055 CMMINT (KMKER) = CMMINT (KMKER + 1)
1060 GO T0 93C
1065 CMMINT (KMKER + 1) = CMMINT (KMKER + 1)
Contrails

930 HZ = H/2.
IF H .LT. HMIN ) HZ = HMIN
HZ = H / 2.
HZHZ = 2 * HZ
INKH = -1
G0 T0 F05
ENTRY CUT
IF ( FMIN .SLT. .95C, 940
490 JXG
ERRORW = 1
RETURN
750 K = 1
555 IF K .LE. IPIATL ) G0 T0 980
980 IF (K SLT. 1.565, .940, 915
565 I6 = K - 1
G0 970 I = 1, IK
KMVER = [AI(I)]
XMIN(I) = XMIN(I) KMVER
CONTINUE
G0 T0 940
940 KMVER = [II(IK)]
X(IK) = CMNMT KMVER
AJ = X(IK) + (ABS(X(IK)) + 1 )*CUTERR
XL = X(IK) - (ABS(X(IK)) + 1 )*CUTERR
IF (INDIK) .LE. IIC5, 555
105 IF (X(IK) .LT. 104C, 104C, 1035
105 IF (X(IK) .LT. 101C, 101C, 102C
105 IF (K SLT. 5 ) G0 T0 955
G0 T0 945
1220 IF ( X(IK) .LT. 103C, 104C, 1040
1300 RETURN
1340 IF ( IIRAND .NE. 0 ) G0 T0 1054
WRITE(6,IC3)
1391 FORMAT(1X,26=******** CUTOFF PASSED IN .50TH CUTOFF VARIABLE
ON WHICH THE INITIAL CALL TO CLT *********)
STOP
1054 CONTINUE
T = 1 - HZ
HZ = 1 - HZ
HZ = HZ / 2.
HZ = HZ / 2.
M = HZ
A1 = C2 * HZ
A2 = C2 * HZ
ISOCNT = 0

176
LPRINT = 1

377
SUBROUTINE INITUP ( IYD, CLVAL, IDIR )
COMMON / IZERM / IPTAIL, IPTOL, LCCNAM(50)
COMMON / CINT / I, PXAX, HMIN, EMIN, EPAX, XS(I), MZ, CUTFRR,
IN, INVH, FINT, IPRT, IPDA, TVL, IACIS),
2 IND(81, 1
ENTRY LOC
IPTAIL = IPTAIL + 1
IF ( IPTAIL .LE. 8 ) GO TO 3
WRITE (6,1)
1 FORMAT (5X) ------JOB TERMINATED, MORE THAN EIGHT CALLS TO LOC
C(--)(--)
STOP
3 CONTINUE
1 ACCEPTPTAIL = IYD
K = IPTAIL + 1
ACCEPTPTAIL = IDIR
RETURN
ENTRY INLP
IPTOL = IPTOL + 1
IF ( IPTOL .LE. 50 ) GO TO 4
WRITE (6,2)
2 FORMAT(1X) -------JOB TERMINATED, MORE THAN FIFTY CALLS TO INLP
C(--)(--)
STOP
4 LCCNAM(IPTOL) = IYD
RETURN
END
SUBROUTINE PLOT3 (X,Y,LM,IC,PM,LL,XDEL,TITLE,IDM,TAPEN)
C PLOT3, A POINT PLOT ROUTINE ADAPTED FROM 360 PLOT
C PLOTS A GRAPH OF ONE OR MORE CURVES FROM GIVEN SETS OF RECTANGULAR COO
C
C CALL PLOT3 (X,Y,LM,IC,PM,LL,XDEL,TITLE,IDM)
C
C X   NAME OF A 2 DIMENSIONAL ARRAY CONTAINING THE X COORDINATES OF
C           ALL THE CURVES TO BE PLOTTED. THE X COORDINATES OF THE N-TH
C           CURVE, FOR EXAMPLE, ARE STORED FROM X(I1,N) TO X(I1+N) WHERE N IS
C           THE NUMBER OF POINTS IN THE CURVE.
C
C Y   HAS THE SAME DIMENSIONS AS X AND CONTAINS THE Y COORDINATES.
C
C LM  NAME OF ONE DIMENSIONAL ARRAY SET UP BY THE USER.  
C      LM(I) IS THE NUMBER OF POINTS IN THE N-TH CURVE, WHOSE FIRST 
C      POINT IS AT X(I1,N) AND Y(I1,N).
C
C IC  NAME OF ONE DIMENSIONAL ARRAY SET UP BY THE USER.  
C      IC(I) IS THE CHARACTER—LEFT ADJUSTED IN A 4 BYTE WORD—TO BE 
C      USED IN PLOTTING THE N-TH CURVE.
C
C PM  THE NUMBER OF CURVES TO BE PLOTTED. IT MUST BE LESS THAN OR 
C      EQUAL TO THE SECOND DIMENSION SPECIFIED FOR THE X AND Y ARRAYS 
C      IN THEIR DIMENSION STATEMENT.
C
C LL  1 NO BORDER, NO AXIS
C      2 BORDER, NO AXIS
C      3 NO BORDER, X AXIS ONLY
C      4 BORDER, X AXIS ONLY
C      5 NO BORDER, Y AXIS ONLY
C      6 BORDER, Y AXIS ONLY
C      7 NO BORDER, BOTH AXES
C      8 BORDER, BOTH AXES
C
C XDEL 0 SINGLE PAGE PLOT DESIRED
C      1 MULTIPLE PAGE OR FRACTION OF PAGE PLOT DESIRED
C
C TIT  1 PLOT GIVEN POINTS
C      2 SEMILOG (PLOT LOGS OF Y COORDINATES)
C      3 LOG-LOG (PLOT LOGS OF X AND Y COORDINATES)
C
C XDEL 0 INDICATES DELTA X IS TO BE CALCULATED
C      OTHERWISE SPECIFY DELTA X IN FLOATING POINT.
C
C TITLE NAME OF THE ARRAY IN WHICH THE TITLE TO HEAD EACH PAGE IS 
C      STORED.  15 TO 50 CHARACTERS.  IDM IS ALWAY PRINTED.
C
C IDM  THE FIRST DIMENSION SPECIFIED FOR THE X AND Y ARRAYS IN THEIR 
C      DIMENSION STATEMENT. IT MUST BE AT LEAST EQUAL TO THE NUMBER 
C      OF POINTS IN THE CURVE WHICH HAS THE GREATEST NUMBER OF POINTS 
C      OF THE CURVES TO BE PLOTTED.
C
C

179

Approved for Public Release
C COMMON COMC,(440)
EQUIVALENCE ((NDCLS,COMC1), (725))
EQUIVALENCE ((TITLE,COMC1), (795))
DIMENSION TITLE(11)
DIMENSION X{11}, Y{11}, L{11}, A{11}, PLINE{11}, XAX{6}, TITLE{11}
INTEGER A{5}, PLINE{5}, ALNK{PLUS,TAPEN
DATA PLUS,MINL{5}, T{11}, MINL{PLUS, TAPEN, MINL{PLUS, TAPEN}
11 FORMAT(F4.2,E6X,4A10)
12 FORMAT(1H7 SCALE/INCH X=1.1,E14.7,5H Y=1,E14.7,10X,25H+OK TO
DIERANCE/POINT X=-1,E14.7,5H Y=-1,E14.7)
13 FORMAT(11X,11A10)
14 FORMAT(1X,2E5.2,1X,1A15)
15 FORMAT(11X,5F1H,19X), I=7I7, X{5I7(4.E9.2,1I1),69.2)
16 FORMAT(1L, MINL{5}, X=1,E14.7,5H Y=-1,E14.7,10X,23X,12HMAX{MUM,
MUM{5}, X=1,E14.7,5H Y=-1,E14.7)
MAX{5X(11), YMAX{5X(11), MIN{5X(11)}
XDEL{5X(11), YDEL{5X(11), XDEL{5X(11), YDEL{5X(11)
MP{5X(11), IF{MP{5X(11), XDEL{5X(11), YDEL{5X(11), XDEL{5X(11), YDEL{5X(11)
4 INITIAL IGN IS NOW COMPLETE
C WH CURVES WILL BE SEARCHED FOR XMAX, XMIN, YMAX, YMIN, AND XDEL{MINIMUM
C DISTANCE BETWEEN ANY TWO POINTS ON ANY CURVE
GO TO (3,2,1,1)
1 XMAX=ABSIXMAX{5)
2 YMAX=ABS{YMAX{5)
3 XMIN=XMAX{5)
4 YMIN=YMAX{5)
5 IF{XDEL{5X(11), YDEL{5X(11), C1) GO TO 5
6 L=2
7 DO 400 J=1,11C
N=1+11M{J-11
X{CMAX{5X(N)
Y{CMAX{5X(N)
GO TO 223+21,2C)LL
20 X{CMAX{5X{CMAX{5)
21 Y{CMAX{5X{Y{CMAX{5)
22 XMIN=X{CMAX{5)
23 YMIN=4CMAX{5)
24 GO TO (23,27,7)
25 XCEL{T=0
26 DO 400 N=1,111M{N=1+11M{J-11
J=4TM{5X(N)
Y{5X(N)
GO TO (30,25,28,7)
28 AT{EMP=ABS{X{TEMP{5)
180
29 YTEMP=ABS(YTEMP)
30 XMAX=MAX1(XTEMP,XMAX)
XMIN=MIN1(XTEMP,XMIN)
YMAX=MAX1(YTEMP,YMAX)
YMIN=MIN1(YTEMP,YMIN)
GO TO 21,30C
11 IFIX .GE. NFI1 GO TO 30C
11 NL=NL+1
GO 100

31 IFIX+1
GO 100
NL=NL+1
GO 100

32 KDIFl=XTEMP-XINV)
33 XIff=XTEMP/ABS(VNI)
34 IF(DIFF .EQ. 0.) GO TO 100
35 XDIFF=ACLID0(XDIFF)
36 IF(DIFF .EQ. 0.) GO TO 100
37 IF(XDELl=0.,0.) GO TO 338
38 XCDEN+XDIFF
39 KDIF+1
GO TO 100

32 XCDEN+YMIN1(DIff.*XDELl)
30 CONTINUE
30 CONTINUE
XMAX=MAX1(XMAX,XMAX)
XMIN=MIN1(XMIN,XMIN)
YMAX=MAX1(YMAX,YMAX)
YMIN=MIN1(YMIN,YMIN)
GO TO 37,40C
34 XDEL-l=XCDEN
GO TO 400

40 CONTINUE
40 IF(XDEL=0.,0.) GO TO 365

41 IF(XMAX=100.,0.)
42 XMIN=100.,0.
GO TO 340

43 XMAX=XMAX=MAX1
44 XMIN=XMIN=MAX1
GO TO 340

45 XMIN=MIN1
GO TO 340

46 YMAX=XMAX=15.
GO TO 3364

47 NIP=0

181

Approved for Public Release
C IF THE CURVES ARE PARALLEL TO THE Y AXIS, A SIMILAR PROCEDURE IS
C FOLLOWED FOR YMAX, YMIN, AND YDELT
165 IF (YMAX >.0, YMIN) GO TO 366
164 IF (YMAX >.0, YMIN) GO TO 364
163 YMAX=.10, YMAX
YMIN=10, YMIN-10.
G0 TO 751
162 YMAX=2, YMAX
YMIN=YMIN-YMAX
GO TO 751
161 YMIN=2, YMIN
YMAX=YMAX-YMIN
160 MP=0
159 G0 T(I37), (35), LL
158 IF (MAX >.0, C.) GO TO 356
157 YMAX=ALOGIC (YMAX)
156 IF (XMIN >.0, C.) GO TO 358
155 YMIN=ALOGIC (YMIN)
154 IF (YMAX >.0, C.) GO TO 355
153 YMAX=ALOGIC (YMAX)
152 YMIN=ALOGIC (YMIN)
151 X RANGE=XMAX-XMIN
PRANGE=DXDEL*ICE.
C IF THE SINGLE PAGE OPTION IS DESIRED BUT XDELT IS TOO LARGE FOR THE
C RANGE OF X VALUES TO FIT ON ONE PAGE, THEN A NEW XDELT MUST BE FOUND
148 IF (MP <.0, C.) GO TO 376
147 IF (X RANGE >.0, PRANGE) GO TO 376
146 XDELT=XRANGE/ICE.
PRANGE=X RANGE
145 WAGE=XRANGE/PRANGE Y1.
YDELT=Y RANGE/EC.
Y2=F Y RANGE/.33
X3=F PRANGE/10.8
YTP=YDELT/2.
YTP=YDELT/.
XMAX=XMIN+4XTP
C WHEN THE PLOT IS FORMED A LINE AT A TIME, SEARCHING EACH CURVE ONCE
C FOR EVERY LINE ON EVERY PAGE AND PRINTING EACH LINE AS SOON AS
C IT IS FORMED
39 GO T(-X, Y, WAGE
135 IF (XMAX -XMAX >X, XTP) GO TO 73
134 IF (XMAX -XMAX >X, XTP) GO TO 401
133 IF (XMAX -XMAX >X, XTP) GO TO 403
132 IF (XMAX -XMAX >X, XTP) GO TO 402
131 (XMAX -XMAX >X, XTP)
402 G0 T04, 42, 41, LL
41 PXMIN=10, **+PXMIN
PXMAX=10, **+PXMAX

182
NEL = (J)
GO 600 I=1,NEL
M=1
I(JM(J-1))
GO TO(54,52,52)LL
57 YTEMP=ABS(YM(I))
IF(YTEMP-YNB)GO TO 56,53
54 IF(YM(I)-YMAX)GO TO 56,55
53 IF((XJ(XC)YUB) GO TO 6CC
51 XTEMP=X(I)
50 IF(XTEMP-XMIN)GO TO 52,57
57 IF(XTEMP-XMAX)GO TO 62,60,6CC
44 GU TOE56,55,56,LL
34 XTEMP=ABS(XM(I))
IF(XTEMP-XMAX)GO TO 61,5C
63 IF(XTEMP-C1)GO TO 63
31 IF(XTEMP-XC)GO TO 62
31 XTEMP=ALGOID(XTEMP)
22 JJ=XTEMP-XMIN/XDEL1
GJ FUB=J+4,61,LL
31 IF(X(J)MLEC)GO TO 65
34 IF(YJ(XMGEIC)GO TO 66
32 SS=JNN
GU TO 67
60 SS=A(J)
60 IF(KAP==MOD(JJ+1C)
N=J(J)+41
61 CALL PACK(PLINEI1,I.S,NCHAR)
630 CONTINUE
60 IF(M<DELNE,16,NE,1) GO TC 69
931 IF(INL,NE.1,1) GO TO 71
917 WRITE(TAPEN14)YM(1),PLINE
GU TO 71
64 IF(LINE-LC,51) GO TO 71
71 WRITE(TAPEN17)PLINE
GU TO 715
71 YTEMP=YMAX+X
715 WRITE(TAPEN14)YTEMP,PLINE
715 YUB=YUB-YDEL
YDEL=YMAX-YDEL
700 CONTINUE
77 GO 800 I=I,6
J=I-1
XA(I)=XMIN+XP+XDEL*FLOC(I)Z0.
800 CONTINUE
WRITE(TAPEN15)XMAX
700 CONTINUE
73 RETURN
END
SUBROUTINE PACK(WORD, CHAR, IPOD)
DIMENSION SPLIT(10)
DATA SPLIT/10*10/
CICODE(10,2,WORD) (SPLIT(1),I=1,10)
SCODE(10,*1,CHAR) (SPLIT(1),I=1,10)
RETURN
END
FORMAT(ICA1)
ENC
SUBROUTINE DISC02(X,Y,TABX,TABZ,TABY,NS,NZ,NY,Z,NDIM)
C
DISC02 PROVIDES FOR LINEARLY INTERPOLATION OF FUNCTIONS
OF ONE OR TWO INDEPENDENT VARIABLES.
CALL DISC02(X,Y,TABX,TABZ,TABY,NS,NZ,NY,Z,NDIM)
C
X FIRST INDEPENDENT VARIABLE
Y SECOND INDEPENDENT VARIABLE
DUMMY IF THERE IS ONLY ONE INDEPENDENT VARIABLE
TABX ARRAY OF VALUES FOR THE FIRST INDEPENDENT VARIABLE
MUST BE IN ASCENDING ORDER
TABZ ARRAY OF VALUES FOR THE DEPENDENT VARIABLE
TABY ARRAY OF VALUES FOR THE SECOND INDEPENDENT VARIABLE
MUST BE IN ASCENDING ORDER
DUMMY IF THERE IS ONLY ONE INDEPENDENT VARIABLE
NS NUMBER OF INDEPENDENT VARIABLES
NS=1 ONE INDEPENDENT VARIABLE
NS=2 TWO INDEPENDENT VARIABLES
NZ NUMBER OF ELEMENTS IN THE ARRAY OF
DEPENDENT VARIABLES (TABZ)
NY NUMBER OF ELEMENTS IN THE Y ARRAY, (TARY)
NOT USED IF THERE IS ONLY ONE INDEPENDENT VARIABLE
Z INTERPOLATED DEPENDENT VARIABLE
NDIM DIMENSION OF I IN Z(I,J) AND X(I,J)
THE FIRST SUBSCRIPT OF THE ARRAYS TABX
AND TABZ WHEN THEY ARE TWO DIMENSION ARRAYS
C
DIMENSION TABX(1),TABZ(1), TARY(1)
IX=NS
IF(NS.GE.2) NX=NX/NY
IF(IX.LE.TABX(1)) GO TO 4
IF(IX.GE.TABX(NX)) GO TO 5
GO TO 6
IX=1
CONTINUE 7
IXZ=IX-1
GO TO P
4 IX=1
IXZ=2
CONTROLS

IF (NS GT 1) GO TO 5

Dx = TABX(I)

ZCA = TABZ(1) - TABZ(II) / TABX(I)

RETURN

IF (Y LE TAYV(I)) GO TO 11

IF (Y LE TAYV(II)) GO TO 12

D0 (3 1 = 1) AY

IV1 = Y

IF (Y LE TAYV(II)) GO TO 14

CCCONTINUE

IV2 = IV1 - 1

GO TO 15

IV1 = IV2

GO TO 15

IF (Y = NY)

IV2 = NV - 1

CONTINUE

I = N0[I*(IV1 - 1)

I1 = I + 1

I2 = I + I2

I3 = I + I1

I4 = I + I2

Z = TABX(I) - TABX(I)

ZCA1 = CTabZ(II) - TABZ(II) / Z

ZCA2 = CTabZ(14) - TABZ(14) / Z

Z = TAYV(I) * TAYV(II)

ZBY = CTabZ(III) - TABZ(III) / Z

ZB = - TAYV(IV)

ZBY = TAYV(IV)

Z = TAYV(III) + TAYV*DSX + TAYV*DSY + DSYDSY*DSY

RETURN

ENC

187
OVERLAY MCPLOT,5,5

PROGRAM MALOC
COMMON CCMDNC(4400)
COMMON / 5 / A(I12,12,11),B(I16)
COMMON P1R,LBLK/TEMP,ANS,AN,NA
COMMON /ETLUCS/DUM1(I26),PR1(I20),DLW2(2001),JF(100),DUM3(1207)

* NDPROP,NDORDU,DUM4(95),GAFCTR,
R KOSTORE(2,1501)
DIMENSION NH(10),TITLE(10),P(15)
DIMENSION NHY(11),NBNDY(16),SAVE(5),ANS(121),ANSWER(11),TEMP(10,5)
DIMENSION TEMPO(10C),TINAR(100)
DIMENSION KGA(4),JLOCUS(4),DELTI(4),TOLI(4),ICROS(14)
EQUIVALENCE (ACORD,CCMMN(11))
EQUIVALENCE (NCASE,CCEEDM(4344))
EQUIVALENCE (ENDL5,CCMMN(7259))
EQUIVALENCE (JLOCUS,CCMMN(4354))
EQUIVALENCE (P,CCMMN(7801))
EQUIVALENCE (TITLE,CCMMN(7951))
EQUIVALENCE (KGAIN,CCMMN(4350))
EQUIVALENCE (KGAIN,CCMMN(4350))
EQUIVALENCE (DELTI,CCMMN(4358))
EQUIVALENCE (DELTI,CCMMN(4358))
EQUIVALENCE (TOLI,CCMMN(4362))
EQUIVALENCE (TOLI,CCMMN(4362))
EQUIVALENCE (ICROS,CCMMN(4361))
EQUIVALENCE (ICROSS,CCMMN(4333))
EQUIVALENCE (NBNDY,CCMMN(4491))
EQUIVALENCE (NNREY,CCMMN(43461))
INTEGER ACROM(12,12)
DATA (FERO/C/
SAVE(1)=P(1)
SAVE(2)=P(2)
SAVE(3)=P(3)
SAVE(4)=P(4)
SAVE(5)=P(5)
JSAV=0
DO 500 L=1,NCASE
IF(JLOCUS(L).EQ.0)GO TO 500
P(1)=SAVE(1)
P(2)=SAVE(2)
P(3)=SAVE(3)
P(4)=SAVE(4)
P(5)=SAVE(5)
KGAIN=KGAIN(I)
DELTI=DELTI(I)
TOLI=TOLI(I)
ICROSS=ICROSS(I)
DO 10 KLK=1,4
JSAV=JSAV+1
10 NBNDY(KL)=NBNDY(KL)+JSAV
KLK=JLOCUS(I)
PLUCUS=0
PI(1K)=PLOCULS
CALL BLANK
CALL CLASS(INDCOLS)
WRITE(6,60C1)
630 FORMAT(1X,10X,1C6)
631 FORMAT(1X,28X,*A MATRIX FOR OPEN LOOP POLES POLYNOMIAL=1)
WRITE(6,6C1)
633 FORMAT(1X,1D12(4X,12,5X))
DO 30 K=1,12
30 WRITE(6,6C1) (A(I,J,K),J=1,12)
634 FORMAT(1X,12((IPE11,4),)
CALL SKBJ
NORDP=NA
NORDP=1
TEMPUR(I)=ANS(I)
43 PR(I)=ANS(I)
PLOCUS=1
PI(1K)=PLOCULS
CALL BLANK
CALL CLASS(INDCOLS)
WRITE(6,60C1)
WRITE(6,6C1)
632 FORMAT(1X,20X,*A MATRIX FOR CHARACTERISTIC EQUATION=PLOCUS = 1*)
DO 60 K=1,12
60 WRITE(6,6C1)
CALL PSLUE TEMPUR,NORDP1,ANS,NA,TEMPER,NORDPL
DO 70 I=1,NORDZ
70 WRITE(6,6C1)
IPABS(2N(NORDZ)+1,GT,1,E-10) GO TO 60
NORDZ=NORDZ+1
IF(NORDZ.EQ.0) GO TO 250
GO TO 80
93 NORDZ=NORDZ-1
CALL CLASS(INDCOLS)
WRITE(6,60C1)
WRITE(6,6C1)
611 FORMAT(1X,28X,*OPEN LOOP POLES POLYNOMIAL IN ASCENDING CHDF*)
WRITE(6,6C1)
612 FORMAT(1X,28X,*OPEN LOOP POLES POLYNOMIAL IN ASCENDING CHDF*)
WRITE(6,6C1)
613 FORMAT(1X,28X,*OPEN LOOP POLES POLYNOMIAL IN ASCENDING CHDF*)
WRITE(6,6C1)
189

Approved for Public Release
**Character Equation in Ascending Order**—PLOCS

\[ 9 = 17/1 \]

\[ NA=NA-1 \]

\[ WRITE(6,121)ZERO,ANS(1),(T,ANS(I+1),I=1,NA) \]

\[ WRITE(6,144) \]

**Open Loop Zeros Polynomial in Ascending Order**

\[ WRITE(6,121)ZERO,ZR(I+1),I=1,NZERO \]

\[ NZERO=0 \]

\[ IF(ABS(ZR(I)) GT 1.E-60) .OR. (ABS(ZR(I)) GT 1.E-60) \] goto 107

\[ KCGDFM=NZERO+1 \]

\[ D1 100 1=2,KCGDFM \]

\[ J=1-1 \]

\[ PRI(I)=ZR(I) \]

\[ KCGDFM=NZERO+1 \]

\[ C0 105 1=2,KCGDFM \]

\[ J=I-1 \]

\[ Z(RJ)=ZR(I) \]

\[ NZERO=NZERO-1 \]

\[ NZERO=NZERO+1 \]

\[ G1 105 \]

**Write**

\[ IF(NQUI.R.EQ.N) WRITE(6,637)INDLR \]

**Write**

\[ IF(NQUI.R.EQ.N) WRITE(6,637)INDLR \]

**Write**

\[ CALL FLIPPCF(PR,NORDP) \]

**Write**

\[ CALL FLIPPCF(PR,NORDZ) \]

**Write**

\[ IF(IAHST(KGAIN.1),EQ.1)GO TO 120 \]

**Write**

\[ IND=1 \]

**Write**

\[ KGOFF=1 \]

**Write**

\[ KGOFF=1 \]

**Write**

\[ GO TO 120 \]

**Write**

\[ IND=1 \]

**Write**

\[ KGOFF=NZERO+1 \]

**Write**

\[ KGOFF=NZERO+1 \]

**Write**

\[ IF(IFPR(KCGDFM).NE.0) GO TO 132 \]

**Write**

\[ KCGFPC=KCGDFM+INC \]

**Write**

\[ GO TO 125 \]

**Write**

\[ IF(IFPR(KCGFPC).NE.0) GO TO 135 \]

**Write**

\[ KCGFPC=KCGFPC+INC \]

**Write**

\[ GO TO 132 \]

**Write**

\[ CFACT=2*PK(KCGFPC)/PK(KCGFPC) \]

**Write**

\[ CALL ATLCSV \]

**Write**

\[ CONTINUE \]

**Write**

\[ CALL CLASS(INDCLS) \]

**Write**

\[ CONTINUE \]

**Write**

\[ IF(AQUI.R.EQ.N) GO TO 90 \]

\[ EV0 \]
SUBROUTINE SKBJ1
COMMON COMCN(44,CC)
COMCN / 5 / TEMP(12,12,11,8116)
COMMON /PLABK/CUMP(12,5),ANSW(11),ANS39110,AN
DIMENSION MD1(12,12,144),ANSW(11),ANS(11,12),APR(213),AP(311),1411,1
+ R5(11),KRF(11),KRF(11),R9(51),R9(101),R11(111)
* RESULT(111),MEMO(12,12)
EQUIVALENCE (A,C,COMCN2591)
EQUIVALENCE (EN,C,COMCN 5991)
EQUIVALENCE (MORD,C,COMCN 111)
C
NA=13-N
DJ 1 1=1,12
DJ 11 J=-2,12
ND(II,JJ)=J-1
DO 11 K=1,11
II(l,J,J)=C
DO 12 I=1,12
ND(I,J)=C
12 11(I,J,J)=I
II(I,J)=NA+12
11=I
II=I
JJ=0
DO 13 J=NA+12
JJ=JJ+1
13 11=II,11
I=NGE(I11,111,11)
DO 12 K=1,11
11=II,3+K
II=II
C * ZEROTES THE ANSWER AND THE ORDER OF THE ANSWER.
C ************************* ________________________________
C
C
NA=0
II=60
1 1=1,121
61 AN(N)=0.
200 DJ 154 I=1,12
K=SIGN=0
11=NGNE(1,11,111,111,15<n,61
C
C *** ..............................................................
C * THE FOLLOWING FINDS THE PROPER TERMS OF THE DETERMINANT T T *
C * THE MULTIPLY AND KEEPS A RICHUD OF THE SIGN USING TSIGN,1, TSIGN,
C * TSGN4, TSGN5, TSGN6, TSGN7, TSGN8, TSGN9, TSGN10, TSGN11, AND *
C * TSGN.
C * KSIGN.
C ******************* ................................................
C
61 DG 155 J=1,12
155=SIGN
1F(K=0)
91  ISGN=ISGN+1
92  IF(M=1) PRINT 152,94
93  ISGN=ISGN+1
94  S=S+SIGN(I,2,M)
   DO 95 E=1,S
95  K5(E)=0.
   N0=N0+M+1
   D0=1+M
   DO 96 E=1,N0
96 =G  E=1,N0
   DO 96 E=1,N0
   IND=E+F-1
   R5(IND)=RA(E)*A(S,M,F)+K5(IND)
   DO 151 P=1,12
   IF(NORD(6,P))152,151,96
98  ISGN=0
   IFP=1106,151,1CC
99  ISGN=ISGN+1
100  IFP=I)101,151,1C2
101  ISGN=ISGN+1
102  IFP=K1113,151,1C4
103  ISGN=ISGN+1
104  IFP=L1105,151,1C6
105  ISGN=ISGN+1
106  IFP=M1117,151,1C8
107  ISGN=ISGN+1
108  T=S+NORD(6,F)
   DO 109 E=1,S
109  R5(F)=0.
   N0=NORD(6,F)+1
   DO 110 E=1,N0
   IND=E+F-1
   R5(IND)=RA(E)*A(S,F)+F
110  DO 112 E=1,112
   IF(NORD(4,E))112,111,112
112  ISGN+C
   IFQ=1112,15C,114
113  ISGN=ISGN+1
114  IFQ=J1115,15C,116
115  ISGN=ISGN+1
116  IFQ=K1117,15C,118
117  ISGN=ISGN+1
118  IFQ=L1119,15C,120
119  ISGN=ISGN+1
120  IFQ=M1121,15C,122
121  ISGN=ISGN+1
122  IFQ=P1123,15C,124
123  ISGN=ISGN+1
124  M=M+NORD(7,G)
   DO 125 F=1,M
125  RTIE=0.

193
SUBROUTINE PSUB (X, IDIMX, Y, IDIMY)
C
C THIS IOM-SP SUBROUTINE SUBTRACTS ONE POLYNOMIAL FROM ANOTHER

DIMENSION Z(IDIMX, IDIMY)
IDIM = IDIMX
IF (IDIMX .GT. IDIMY) GO TO 20
NDIM = IDIMY
20 IF (IDIM .LE. 0) GO TO 90
IF (I .GT. NDIM) GO TO 60
IF (I .GT. IDIMX) GO TO 40
Z(I) = X(I) - Y(I)
GO TO 90
60 Z(I) = Y(I)
GO TO 40
70 Z(I) = X(I)
80 CONTINUE
90 IDIM = NDIM
RETURN
END
SUBROUTINE FLIPRO(A,M)
DIMENSION A(1)
NUM=N+1
NUM2=NUM/2
DO 10 I=1,NUM2
J=NUM+1-I
DUMMY = A(J)
A(J)=A(I)
A(I)=DUMMY
10 CONTINUE
RETURN
END
830 IF (JSW) MIC, 1IC, 820
840 WRITE (10, F) (GRAPH(J, J+1, 131))
850 GO TO 830
860 WRITE(J, 5) XVAR (GRAPH(J, 131), J=1, 131)
870 IF (NP-NAXIS) 35, 55, 63
880 IF (NP-NAXIS) J=1, NMK
890 J1 = MARK(J1)
900 GRAPH(J1, NSW) = BLANK
910 GO TO 900
920 DO 100 J=1, 131
930 DO 100 J=1, 131
940 IF (NP-NAXIS) NAX, 50, 50
950 JSW = 1
960 IF (NGRID) 52C, 52C, 904
970 KSW = 2
980 KSW = 2
990 IF (NVIEW) 51C, 51C, 9C5
990 WRITE (10, 12) NVIEW
990 WRITE (10, 10)
990 RETURN
1000 ENC
SUBROUTINE RTOCSINF, NP, X, Y, ANG, STEP, TCOL, CHG, NAI, OUT, ISX, ASX, XDY, YDY, ANGB, C(TLD)

CJUMP /RTLOC/, 1, 2, 3, 4, 5, 6
1 P11, P12, I10, D10, T12, T10, I10, D10, X10, ANG, NAG, ANGB, C(TLD), MODE,
2 SGNLT(3), DELT2, TOLHL, NPL, N2L, AKAIL, TAU, DELT, TCOL, ANDY(4),
3 P110, P120, I100, D100, ZK100, Z100, X100, Y100, XDY, YDY, ANGB, C(TLD),
4 LUMI(3), NPREU, DUMB2191, T(14), DUM3212, TCLANG, DUM4, C(NFCT),
5 KOSTOR(2, 1501)
1 JUMP /IUKA/, 2, XEP, 16, 16, 15
25 DUM TO 156, 55, 65, 55, 40, 505, 160, 40, 54, NF
5) NRT = 0 STEP2 = STEP*STEP IF(NP) = C(55, 55, 55)
C SINGLE PRECISION ROUTINE LINKAGES
5) IF(NP) = C(55, 55, 65)
5) NRT = -1 STEP2 = STEP*STEP
61 TPX = X TPY = Y IF(NP) = C(55, 55, 65)
62 IF(NP) = C(100) 226, 226, 200
63 IF(TOL = C(41) = C(100)
C MULT LOCUS ADVANCE - SINGLE PRECISION
100 ANGT = ANG NIA = 1
105 CX = STEP*COS(ANGT) CY = STEP*SIN(ANGT)
TPX = X + CX TPY = Y + CY IF(AHS(TPY) = TOLL 110, 110, 115
110 TPY = 0.
115 NPOINT = 1 GO TO 300
120 IF(AHS(SANGT) = TOLL 110, 110, 125
125 IF(NIA = NAI) 125, 125, 125
128 ISX = 5 RETURN
140 NIA = NIA + 1 C ANGLE CORRECTION ROUTINE (NEWTON-RAPHSON)
130 NRT = 1 GO TO 200
175 IF(NP) = C(105, 105, 110)
140 WRITE(10), 11 SGNLT(3), TPX, TPY, A, SANGT, TANG, STEP, NIA, NID, NF
185 XS = TPX YS = TPY NRT = 0 C TOTAL ANGLE COMPUTATION AND ANGLE CORRECTION - SINGLE PRECISION
200 FX = 0.
PY = -YAL NS = 0

205

Approved for Public Release
IF(NPL) 224,224,224
204 EU 220 +1,NPL
XT = TP-2(1)
YT = TP-Y(1)
DIST = XT*XT + YT*YT
IF(NRFI) 216,266,266
206 IF(EEMODE) 216,216,216
208 IF(DIST-STEP2) 212,212,216
210 IF(DIST) 212,212,216
212 ISW = 2
V$ = 1
IF(DIST) 255,259,216
216 FX = FX + YT*XM(1)/DIST
FY = FY - XT*XM(1)/DIST
CONTINUE
219 IF(NRF-10) 226,247,247
221 IF(NQFL) 246,246,226
223 DU 244 1=N21
XT = TX-2(1)
YT = TY-2(1)
DIST = XT*XT + YT*YT
IF(NRFI) 232,232,230
234 IF(EEMODE) 232,232,234
236 IF(DIST-STEP2) 236,236,240
238 IF(DIST) 236,236,240
240 ISW = 2
NS = 1
IF(DIST) 255,259,240
243 FX = FX - YXMZ(1)/DIST
FY = FY + XZMZ(1)/DIST
CONTINUE
245 IF(NRFI) 246,246,247
247 IF(NRF-10) 246,246,247
250 IF(NS) 1030,1030,247
247 ISW = 1
RETURN
249 IF(EEMODE) 252,252,250
259 CALL ARCTAN(1,FY-FX,TANG,0,0,0)
GU TO 253
252 CALL ARCTAN(1-FX,FY,TANG,0,0,0)
253 IF(NS) 255,255,254
254 IF(NRF-41) 255,256,255
256 IF(NRF-12) 256,256,256
258 IF(ABS(ABS(TANG)-PI2)-TCL) 256,256,258
259 TANG = SIGN(PI2,TANG)
261 IF(DIST) 264,264,264
262 WRITE(10,1) SGNL(1),FPX,TPY,TANG,FX,FY,STEP,NIA,NID,INSTEP,NF
262 A = ABS(TANG-ANG)
264 IF(ABS(A-TWPI) > CMG) 264,264,270
266 IF(ABS(FX) > ABS(FY)-1.0E-51) 270,268,268
268 ISW = 1

206
IF(INF=2) 265,269,272

A = ANS(TAN=ANGT)

36A-CHG) 269,269,2691

IF(IN=2) 266,266,27C

IF(INF=2) 266,254,256

15M = 4

CUT = TANG

RETURN

1N IF(N=1) 275,275,2742

242 IF(INPREC) 275,275,299

213 A = FY#EX-FX#CY

1N(A) 276,270,27E

36A TANG = -SANG/A

36A+TANG)-CHG) 276,276,277

1T7 TANG = SIGN(A/NIA×TANG)

178 ANGT = ANGT + TANG

G0 TO 15C

C CHECK THAT A ZERO OF SANG IS WITHIN TOL' PERPENDICULAR DISTANCE

248 DX = TOLL#DX

DY = TOLL#DY

FPA = X5 + D51

FPT = Y5 + D51

NPQINT = 2

GQ TO 30C

24B DIF = X5 - D51

FPT = Y5 + D51

NPQINT = 3

GQ TO 30C

703 AD = X

YG = Y

AANG = ANG

AANCT = ANGCT

FITAL2 = TOL+ANG

NDT = 1

GQ TO 30C

24B AD = X

Y = YS

AN = TANG

GQ = SANG

RETURN0

C ANGLE SUMMATION ROUTINE (EVALUATE SANG+1 - SINGLE PRECISION

313 F=TAU) 35C,254,252

342 A = -TAL#IPT

A = A - IPXIA/TWOP) #TWOP1

GQ TO 30C

344 A = 0

343 IF(NPL) 311,215,3C6

370 GO 310 11E,NPL

CALL ARCTAN((1,T0#-PR(1) ,TPY-PI(1) ,ANGLC,DS,D,D)

370 A = A - ANGLEX#XMP(1)

373 IF(NR) 33C,33C,320

320 GO 325 1=I,NR

207
CALL ARCTAN(1,TPX-2R(1),TPY-Z(T1),ANGLE,D,0,0)
129 A = A + ANGLE*628318
325 IF(INGAIN) = 340,335,335
329 A = A + P1
340 IF(A<P1) = 345,250,350
344 A = A + TWOP
GO TO 34C
350 IF(A<P1) = 360,360,355
354 A = A-TWOP
GO TO 35C
359 OUT = A
415 IF(INF=5) = 365,259,1020
425 IF(POINT=1) = 370,370,335
430 SANG = A
GO TO 12C
525 IF(A=SANG) = 380,346,3EC
530 IF(NPNT-NP-21) = 388,296,54C
C DOUBLE PRECISION ROUTINE LINKAGES
400 XD = X
YC = Y
ANG = ANG
405 IF(INF=-2) = 666,666,41C
410 TPX= XC
TPY= YC
TPY= YC
415 IF(INF=-4) = 666,666,7CC
C ROUTE LIGUS ADVANCE - DOUBLE PRECISION
510 ANGTS= ANGC
TOL2 = TOL1
NIA = 1
NIO = C
525 DX1 = STEP*COS(ANGT0)
DYC = STEP*SIN(ANGT0)
TPX= XC + DX1
TPY= YC + DYC
530 IF(AABS(TPYE-TOL1)) = 51C,510,515
510 TPY= C.
515 NPOINT = 1
GO TO 70C
520 IF(AABS(SMAGD)-TOL2) = 565,565,525
524 IF(AAIA-NAA1) = 524,524,528
524 ISW = 5
RETURN
515 NIA = NIA+1
C ANGLE CORRECTION ROUTINE (NEWTON-RAPHSON)
525 NMT = 1
GO TO 40C
579 IF(INIT=2) = 505,505,56C
594 WRITE(ION,11) SGNLT(3),TPX,TPY,AD,SANG,ANGIDO,TANDO,STEP,NIA,NIO,1
GO TO 50E
585 XSC = TPX
YSC = TPY
C

TRIAL MLCFL COMPUTATION AND ANGLE CORRECTION - DOUBLE PRECISION

030  FXC = C

031  FYC = -TAU

036  NS = 0

037  IF(NPL) e24,4,604

038  DO 20 I=1,NPL

20   XT0 = TPXD-PR(I)

21   YT0 = TPYD-PI(I)

22   EISTC = XT0*XTC + YTD*YTD

23   ENRNT = CCC,C,606

24   IF(MODE) E1C,C,6C8

25   IF(DISTC-STEP) E12,E12,E16

26   IF(DISTC) E13,E73,E16

27   ISW = 3

28   NS = I

29   IF(DISTC) E55,E99,E16

30   FX0 = FXD + YTD*XTMXM(II/DISTC)

31   FYD = FYD - XTD*XM(II/DISTC)

32   IF(NPL) E44,E46,E28

33   CJ E44 T=1,NPL

34   EISTC = TPXD - ZIK(I)

35   EISTC = TPYD - ZIK(I)

36   DISTC = EISTC*XT0 + YTD*YTD

37   ENRNT = E3,E3,E3C

38   IF(MODE) E12,E12,E3A

39   IF(DISTC=51) E36,E36,E4C

40   IF(DISTC) E73,E73,E40

41   ISW = 2

42   NS = 1

43   IF(DISTC) E55,E99,E4C

44   FX0 = FXD - YTD*XM(II/DISTC)

45   FYD = FYD + XTD*XM(II/DISTC)

46   IF(NPL) E44,E46,E74

47   IF(NPL) E1C,C,247

48   IF(MODE) E55,E52,E5C

49   CALL ARC4ANL(I,I,I,FYD,-FXD,TANGD)

50   CG TO 45

51   CALL ARC4ANL(I,I,I,-FYD,FXD,TANGD)

52   IF(NS) 651,E55,E54

53   IF(NF) 651,E72,E72

54   IF(NPR) E55,E55,E55

55   IF(AH(ARSTAN) - 1.D0) - ECL E55,E56,E58

56   TANGD = 5.12,E12,TANGD)

57   IF (TANGD) E22,E22,E6C

58   NSTEP = 0,1

59   INITION(I,II) GONL(T1,TPXD,TPYD,TANGD,FXD,FYD,STEP,NIA,NC,1)

60   ADSTAN=ANGD

61   IF(AH(AD-TWPO) - CHG) E66,E66,E66

62   IF(AH(AD) - CHG) E66,E66,E70

63   IF(AH(AD) - WAP>FXD) - E14,5E,668

64   ISW = 1

209

Approved for Public Release
C CHECK THAT A ZERO OF 'SANGD' IS WITHIN 'TOL' PERPENDICULAR DISTANCE
6001 AX = TCOL**DXD
6003 CY = TCOL**DYD
6005 TPXD = XSD + DXD
6006 TPYD = YSD + DTYD
6008 NPINT = 2
6009 GU TO 700
6010 TPXD = XSD - CYD
6012 TPYD = YSD + DIXD
6013 NPINT = 3
6014 GU TO 700
6015 IF (NPINT .EQ. 3) YS2, 646, 656
6016 XID = NID + 1
6018 TPXD = XSD
6019 TPYD = YSD
6020 TOL2 = TOLANG
6021 GU TO 932
6022 IF (NPINT .LT. 4) 654, 657, 697
6023 X = XSD
6024 Y = YSD
6025 OUT = SANGD
6026 ANG = TANGD
6027 OUT = TANGD
6028 NID = NID + 1
6029 TPXD = XSD
6030 TPYD = YSD
6031 ANG = TANGD
6032 OUT = SANGD
6033 GU TO 919
6034 RETURN
C ANGLE SUMMATION ROUTINE (EVALUATE 'SANGD') = DOUBLE PRECISION
710 IF (TANGD) TC2 = TC4, 702
720 AC = -TAP*TPYD
730 AU = AC-IF (SXISNL (AU/TWOPID)) TWOPID
740 GU TO TCS

210

Approved for Public Release
C THE VARIABLE NA1 IS USED TO DETECT IF THIS IS A RUN TO FIND THE
C NORMALIZING GAIN CONSTANT. IF SU. NA1=30.AND THE PLANT GAIN IS NOT
C DIVIDED OUT OF THE GAIN.

GAIN*GAIN
IF(NA1.0.EQ.30)GO TO 522
GAIN=GAIN/[1+INFCTR*GNFCTR]
GO TO 522

X=M*Z(1)
UST=(X-2R(1))*(X-2R(1))*(Y-2I(1))*(Y-2I(1))

XNP=XNP*500,500,500,500

GO TO 952 J=1,N

GAIN=1.251,548,548,548

GAIN=GAIN/1,E1C

GO TO 544

GAIN=1,E-26) 55C,152,952

GAIN=GAIN*1,E1C

GO TO 522

GAIN=GAIN*1,E1C

GO TO 522

GAIN=GAIN/1,E1C

GO TO 566

213
977 IF (GAIN < 1.0 = 3) 972, 974, 976
972 GAIN = GAIN * 10.
NS = NS - 1
GOTO 970
974 IF (NS) 550, 555, 576
975 IF (GAIN < 1.0 = 3) 576, 555, 590
978 GAIN = GAIN * 10.
NS = NS - 1
GOTO 974
980 IF (GAIN = 1.0 = 21) 585, 582, 582
982 GAIN = GAIN / 10.
NS = NS + 1
GOTO 974
984 MODE = GAIN
NS = NS + 1
GOTO 984
986 IF (GAIN < 1.0 = 1) 591, 588, 588
987 GAIN = GAIN / 10.
NS = NS - 1
GOTO 982
988 OUT = SQRT (GAIN)
RETURN
989 GAIN = GAIN * 1000.
NS = NS - 1
GOTO 982
990 OUT = GAIN
RETURN
992 IF (NP) 998, 554, 558
994 MODE = 1.
OUT = 1.
GOTO 997
996 OUT = 0.
997 NS = 0.
RETURN
998 OUT = 1.0000000000
NS = 995
RETURN

C. NEXT-STEP SUM ROUTINE TO FIND ZEROS CF*SGN* WITH FIXED X OR Y
1000 N/A = C
1001 ART = 1
STEP2 = STEP * STEP
N/E = NF - E
TANG = 1.0 * CHK
1002 PFX = PFX - 5598(4) 1048, 1002, 1002
1003 TYPX = X
1004 TYPY = Y
GOTO 300
1100 IF (ABS (S) > TOLL) 1046, 1046, 200
1130 IF (TYPX) 1032, 1032, 1032
1132 IF (TYPY) 1032, 247, 1034

214

Approved for Public Release
1:134  DX = -A/EX
      IF(ABS(DX)-TANG) 1036,1038,247
1036  TPX = TPX+EX
      GU TO 1044
1038  IF(FY) 1044,247,104C
1040  OY = -A/FY
      IF(ABS(DY)-TANG) 1042,1044,247
1042  TPY = TPY+CY
1044  NIA = NIA + 1
      IF(NIA=NAI) 3CC,247,247
1046  ISW = 0
      X = TPX
      Y = TPY
      RETURN
1048  FPXO = X
      ;TPYD = Y
      (u) TO 75C
1050  IF(ABS(EC)-TOL) 1076,1076,600
1060  IF(NTR) 1CE2,1CC2,1CE6
1062  EFFXO) 1CE4,247,1064
1064  DKE = -AC/FXO
      IF(ABS(DKD)-TANG) 1064,1066,247
1066  FPXO = TPXE * DKD
      GU TO 1074
1074  IF(FYD) 13C,247,13TC
1076  OYE = -AC/FYD
      IF(ABS(DYD)-TANG) 1072,1072,247
1077  TPYD = TPYE + UYD
1077  NIA = NIA + 1
      IF(NIA=NAI) 7CC,247,247
1079  X = 1PXD
      Y = TPYD
      GU TO 888
     END

Approved for Public Release
SUBROUTINE ARC TAN(IFX,Y,X,YD,XD,YD,OUTD)
  GO TO 110,200,110
110  IFX = Y/X
  IFX = 140,200,130
130  OUT = ATAN(IFX)
  RETURN
140  IFY = 16C,15C,15C
150  OUT = ATAN(IFX) + 3,141593
  RETURN
160  OUT = ATAN(IFX) - 3,141593
  RETURN
210  IFY = 21C,22C,23C
213  OUT = -1,570796
  RETURN
220  OUT = 0.
  RETURN
230  OUT = 1,570796
  RETURN
300  IFXU = 210,400,210
310  ARGU = YC/XE
  IFXU = 34C,40C,230
340  OUTU = ATAN(ARGU)
  RETURN
341  IFYD = 360,76C,35C
350  OUTD = ATAN(ARGU) + 3,14159265358979
  RETURN
360  OUTD = ATAN(ARGU) - 3,14159265358979
  RETURN
400  IFYD = 410,42C,430
410  OUTD = -1,5707963267545
  RETURN
470  OUTD = 0.
  RETURN
470  OUTD = 1,5707963267549
  RETURN
END

217
SUBROUTINE POLYRTM(C,TCL1,RR,RY,RMULT,MR,ISW)
C
THIS SUBROUTINE CALCULATES THE REAL AND COMPLEX ROOTS
OF A REAL POLYNOMIAL
C
DESCRIPTION OF PARAMETERS
C
M - ORDER OF POLYNOMIAL (MAXIMUM ORDER = 99)
C
C - VECTOR OF M+1 COEFFICIENTS OF POLYNOMIAL
C
ORDERED FROM LARGEST TO SMALLEST
C
TOL1 - TOLERANCE OF ROOTS
C
RX - REAL ROOTS
C
RY - RESPECTIVE IMAGINARY ROOTS
C
RMULT - MULTICITY OF ROOTS
C
MR - NUMBER OF UNIQUE ROOTS
C
ISW - ERROR CODE WHERE
     ISW=1 NO ERROR
     ISW=2 UNABLE TO DETERMINE ROOT
     ISW=3 M GREATER THAN 99
C
DIMENSION (C(11C),RX(11C),RY(100),RMULT(100),PTX(2),RY(2),IRX(2),
     )
     IRY(2),FRX(2),FRY(2),L(4),ARF(4)
DIMENSION A(11C),PF(4),P(2),U(2)
EQUIVALENCE (X,FRX(1)),(Y,FRY(1)),(1,RY(1))
FXCF = .4294967296.
TOLM = TOL1*TOL1
N = M + 1
NK = 0
CU 2000 I=1,N
IF(C(I)) 2CC2,2CCC,2CC2
2000 CONTINUE
2012 N = N-I+1
IF(N-100) 201C,201C,2CC5
2015 ISW = 0
2050 NR = -1
RETURN
2010 L(I) =N
L(I) =N
K= I-I;
00 2015 I=1,N
2015 J = I+K
2015 A(I,J) = C(I)
2015 NZ = 0
2120 IF(N-I) 202C,2C44,2C22
2222 IF(A(N+1)) 2C24,2C23,2224
2223 NZP = NZP +1
2250 GO TO 202C
2224 T1 = ARS(A(N+1))
2224 T2 = 16.#ERS(A(N+1))
2224 I = 0
2228 IF(I=12)11C22,2012,203C
2230 T1 = T1/T2.
218
2J32 NF = f(x+1)(x+1)
2J34 EFCTR = 1.
2J34 EFTR = 2044, 2C44, 2034
2J34 EFTR = 2035
2J34 RFCTR = RFCTR+2.
2J34 RFCTR = 1.
2J34 K = 0
2J34 T1 = 1.
2J34 T = T1+1.
2J34 N = N+1.
2J34 M = M+1.
2J34 L = L+1.
2J34 J = J+1.
2J34 I = I+1.
2J34 H = H+1.
2J34 G = G+1.
2J34 F = F+1.
2J34 E = E+1.
2J34 D = D+1.
2J34 C = C+1.
2J34 B = B+1.
2J34 A = A+1.
2J34 Z = Z+1.
2J34 Y = Y+1.
2J34 X = X+1.
2J34 W = W+1.
2J34 V = V+1.
2J34 U = U+1.
2J34 T = T+1.
2J34 S = S+1.
2J34 R = R+1.
2J34 Q = Q+1.
2J34 P = P+1.
2J34 O = O+1.
2J34 N = N+1.
2J34 M = M+1.
2J34 L = L+1.
2J34 K = K+1.
2J34 J = J+1.
2J34 I = I+1.
2J34 H = H+1.
2J34 G = G+1.
2J34 F = F+1.
2J34 E = E+1.
2J34 D = D+1.
2J34 C = C+1.
2J34 B = B+1.
2J34 A = A+1.
2J34 Z = Z+1.
2J34 Y = Y+1.
2J34 X = X+1.
2J34 W = W+1.
2J34 V = V+1.
2J34 U = U+1.
2J34 T = T+1.
2J34 S = S+1.
2J34 R = R+1.
2J34 Q = Q+1.
2J34 P = P+1.
2J34 O = O+1.
2J34 N = N+1.
2J34 M = M+1.
2J34 L = L+1.
2J34 K = K+1.
YR = YR + CY
AAS = AAS(X) + AAS(Y)
  IF(AAS = AAS(X) + AAS(Y)) = 1, E=6) 2216, 2216, 2228
2210 IF(AAR = AAR) 2221, 2216, 2218
2219 TVSW = TVSW + 1
2220 IF(TVSW) 2277, 2277, 2224
2224 IF(IF1) 2246, 2246, 225C
2226 NFI = 1
XT = X
YF = YR
K1 = 2
GU TO 2114
2227 ARL = ARS
2228 IF(IF2) 222C, 223C, 224C
2230 N1 = N1 + 1
K1 = 0
2232 IF(N1=50C) 211B, 211B, 2232
2232 IF(N1=50B) 2110, 2274, 2234
2234 SW = 0
GU TO 20CB
2236 NFI = NFI + 1
K1 = 2
2236 IF(NFI=100) 211B, 211B, 225C
2236 IF(NFI=50XJ) 222C, 223C, 224C
2236 IF(IF3) 2262, 2262, 226C
2236 IF(IF4=1, E=6) 2264, 2264, 226C
2236 IF(AAR) = XAR*RECTR
RV11) = YSR*RECTR
GU 2265 1=1, NR
2236 IF(IF5=KTY11) = RV111) + AAR(RV111-RV111-TOL11) 2280, 2280, 2265
2236 CONTINUE
2236 GU TO 2119
2236 IF(IF6=4) 211C, 2290, 229C
2236 X = XT
Y = YT
GU TO 2300
2236 X = XR
Y = YR
+400 IF(AAR(V)=1, E=6) 2305, 2315, 2315
2255 Y = C.
GU 2310 E=1, NT
2310 A1(1,1) = A1(1,1) + X*A1(1,1)
GU 2320 E=1, NT
2320 A1(1,1) = A1(1,1) + X*A1(1,1) + DIV*A1(1,1)
2325 X = X*RECTR
Y = Y*RECTR
2236 IF(IF1(X=2)) 233C, 234C, 2340
2330 IRA(1) = IRA(1) + 1

221
2430 J = J+1
2431 GO TO 2445
2445 {IFJFL = 2446, 2448, 2450
2446 IFK(J) = RX(J) - TOL1
2447 J = JFL
2448 K = NR+1
2449 /K(J) = PX(K)
2450 KY(K) = KY(K)
2451 KMULT(K) = KMULT(K)
2452 K = K
2453 GO TO 2445
2455 {IFJFL = 2456, 2458, 2460
2460 RMULT(J) = 1.
2462 {RA(J) = RTX(I)
2463 {NR = NR1 + 1
2470 RY(I) = RTY(I)
2470 GO TO 2050
END
3 "ANY POINT#T35.4B = BREAKAWAY POINT#T35.4S - PROGRAM STUCK\n4 ." - BRANCH TERMINATED*/1
54 FORMAT (1X,4I1,1PE23.13,1PE15.6,1E13.3X,31X)
55 FORMAT (1H1*T35,"BINARY SEARCH FOR ADDITIONAL STARTING POINTS -
1 ** RESULTS#T39.20X,781,1RUN NUMBER#T13//17X=POINT NO.#T4X-REAL* 1
2 1X\*IMAGINARY#15X=ANGLE ERROR#8X=STARTING ANGLE#7X=N1*1/)
56 FORMAT (15X13,1PE4E2C.2E212)
57 FORMAT (2X1PE4E2C.2E212,1X5(T=NOT START P1.1))
58 FORMAT (///17X=NO ADDITIONAL STARTING POINTS FOUND*)
59 FORMAT (1HC,T44\*IMAGINARY AXIS CROSSING POINTS - RUN NUMBER#T13//
1 /T44\*IMAG. VALUE#27X\*GAIN#1/)
60 FORMAT (14C1,1PE21.15)
61 FORMAT (T35,F20.6)
62 FORMAT (T44\*POINT OUT LOCS DOES NOT CROSS IMAGINARY AXIS*/T40\*HETWE* 1
1 *EN\*F11.4,3X\*AND\*F11.4)
63 FORMAT (1H4\*T12\*PAGE#13)
64 FORMAT (1H6\*RUN NUMBER#13\* NOT EXECUTED - *)
65 FORMAT (1P\*T31\*HOLE POLYNOMIAL UNSOLVABLE*)
66 FORMAT (1H,1T31\*POLE POLYNOMIAL ORDER EXCEEDS 99*)
67 FORMAT (1H,1T31\*ZERO POLYNOMIAL UNSOLVABLE*)
68 FORMAT (1H,1T31\*ZERO POLYNOMIAL ORDER EXCEEDS 99*)
69 FORMAT (1H,55\*ON PREVIOUS RUN - NO NEW INFORMATION GIVEN*)
70 FORMAT (1P\*T31\*ZERO ORDER SYSTEM - NPL-NZL = 0*)
71 FORMAT (1H1\*T15\*F13.2X\*F13)
72 FORMAT (1H4\*T15.6)
73 FORMAT (1X11,1PE(5EX14.6,4X),22X314)
74 FORMAT (1X11,1PE23.13,22X314)
75 FORMAT (//6X\*BRANCH STARTING POINTS ON BOUNDARY*/11X-REAL#6X-IMAG\* 1 \*NARY#//6X\*T12.5,2FX13.5))
76 FORMAT (///1// COMPARISON PLOT CANNOT BE MADE-NO CURVES ACCUMULATED*)
77 FORMAT (6X\*PASSES CAN - LOCUS DISTANCE CHECK*)
78 FORMAT (1H121\*LOCUS DISTANCE CHECK*)
79 FORMAT (1H6\*XPLOIT GAIN#T21,1-1PE13.5)

C SETUP COMPLAINTS

C C ENSURE SUBROUTINE STOPS TRACEBACK MESSAGES FOR UNDERFLOWS WHICH
C COMMONLY OCCUR WHEN FINDING ROOTS OF HIGH ORDER POLYNOMIALS
C CALL ERSETG2(126,256,-1.1,C,208)
DUM2(3)=IFNY(3)
DUM2(4)=IFNY(4)
DUM2(5)=IFNY(5)
DUM2(6)=IFNY(6)
DUM1=KAIN
DUM2(1)=DELT
DUM2(2)=TOL
IP1=KUN1-1C
C TO 51CI
15X=10
TO=6
J1=#6
L[CHK]=1
P410=P11057

225

Approved for Public Release
CONTRAILS

1MUPID = P10 # 2
P12 = P11/2
CANG(1) = P12
CANG(2) = .01
GO TO 5, 10
9)
CANG(I) = P11*(1-I.1/I)
C
 INITIAL ROUTINE
1,10
NPRINT = 1C
ISW = 15
GO TO 992
L11 IF(NRUN.NEF.C) RETURN
5091
NRL = NRUN
NRL = NRUN + 1
C
CALL INPUT(NLIG, NTYPE, KNL)
IF(KNL.EQ.1) RETURN
IF(NKLSW).LT.103.9011.101
C
ROUTINE FOR COMPARISON PLOT ONLY
6111
NRL = NRL
IF(NKLSW).EQ.71C, 71C, 2101
7101
KITE(:M, 76)
GO TO 960
8111
ISWX = 2
ISWX = + 2
GO TO 970
9101
TOLH = TOL/DELT
DELT2 = (DELT+TOL)*(DELT+TOL)
IF(NPREC).GT.2.103
1,2
RCGH = 1.
RCGH = 1.
RCGH = 1.
DELT = DELT
TOL = DELT
GO TO 104
103
ALFG = C+C
BCFG = + 2
DELT = DELT/1CC.
DTM = 2.*DELT
104
GO TO 105 T=1.3
1,35
TOL(I) = C
GO TO 110 T=1.3, 2
IF(SCALE(I+1).EQ.SCALE(I)) 106, 105, 108
176
SCC(I) = RADY(I1)
SCC(I+1) = RADY(I1+1)
GO TO 11C
109
SCC(I) = SCALE(I)
SCC(I+1) = SCALE(I+1)
110
CONTINUE
IF(NSYM).LT.115, 115, 122
115
GO TO 120 T=1.3, 2
IF(SCALE(I+1).EQ.SCALE(I)) 116, 116, 113
118
SCC(I) = RADY(I)
SCC(I+1) = RADY(I+1)
GO TO 12C

226
116 $GCI() = SCALE(GI)
117 $GCI() = SCALE(GI+1)
120 CONTINUE
122 $R = 0
125 $D = 129
127 $R = $SYM(I)
128 IF(INUT(I)) 134, 136, 138
130 IF(INOL(I), E=1) 132, 136, 138
132 $R = -1
134 IF(INW = 0) 138
136 $R = IT(I) + IT(I+1); IABS(11(11(I) + IABS(11((11)) + IABS(11)))) 140, 140, 136
140 $R = 1
142 $R = 1
144 IF(NURP(I)) 142, 146, 146
146 IF(INPL(I)) 144, 151, 158
148 CONTINUE
150 CALL CLASS(INCLASS)
152 WRITE(6, 06)
154 FORMAT(FT, 10X, 10AE)
156 WRITE(6, 05)
158 WRITE(6, 05)
160 GO TO 184
162 CALL POLYRT(INHID, PR, TOL, SX, SY, XR, YR, HIL, ISW)
164 NPOINT = E
166 GO TO 992
168 IF(INW = 2) 174, 148, 150
170 CONTINUE
172 CALL CLASS(INCLASS)
174 WRITE(6, 06)
176 WRITE(6, 05)
178 WRITE(6, 05)
180 GO TO 184
182 IF(INPL(I)) 151, 155
184 GO TO 158
186 NURP(I) = 1
188 IF(INHID(I)) 162, 160, 166
190 IF(INPL(I)) 164, 171, 178
192 CONTINUE
194 CALL CLASS(INCLASS)
196 WRITE(6, 06)
198 WRITE(6, 06)
200 WRITE(6, 06)
202 GO TO 184
204 CALL POLYRT(NURP, ZR, TOL, SX, SY, XR, YR, ISW)
CONT example

CASE <class>
WRITE(6,ECITLE)
WRITE(6,ETITLE)
GO TO 172
CONTINUE
CASE <class>
WRITE(6,ECITLE)
WRITE(6,ETITLE)
GO TO 184
IF(NZL) 178 175
GO 176 E=I,NZL
WRITE (1) 5X(T(1))
WRITE (1) ZU(I) = SY(I)
WRITE (1) N=0Z = 0
WRITE (1) IF(NPL) 210,211,212
WRITE (1) 214 NP = NP + 3*NPL(I)
WRITE (1) NZ = NZ + NP
WRITE (1) IF(NP) 215,235,236,238
WRITE (1) 2110 N=0Z = NZ
GO TO 240
WRITE (1) N=NP + NZ
WRITE (1) IF(NP) 215,265,266
WRITE (1) 2100 N=0
WRITE (1) IF(NP) 215,265,266
WRITE (1) 2100 N=0
WRITE (1) GO TO 215
WRITE (1) 2105 N=0
WRITE (1) IF(IA*SIGMA(I-1)) 212,285,282
WRITE (1) 2300 GAIN = 1
WRITE (1) CALL RTLOC((G,0.0,0.0,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E)
WRITE (1) GO TO 285
WRITE (1) 2400 GAIN = 6
WRITE (1) GAIN(I) = 1
WRITE (1) NEII = 0
WRITE (1) IF(1) 295,297,298
WRITE (1) NLC = 1

END
GU TO 380
NL = 2
NSP = 0
IF(NCUTS) 360, 380, 310
NPOINT = +6
GO TO 952
511 GO 312 I = I + 1
512 DELTA11 = (RANDY(2*I+1)-BNDY(2*I+1)/NCUTS
513 ALI = RANDY(1) / SGN
514 CONTINUE
515 WRITE(6, E0111) "CLASS INDICS"
516 WRITE(6, 55) NPLN
517 A(I) = RL(I)
518 K(I) = RL(I)
519 RLI = RL(I) - SIGN(DELTA11, SGN)
520 CALL RTLC(G, C, RLI, RL(1), RL(2), E, E, E, P, SANGL, M, M, C, D, D, D)
521 IF(SANGL) 280, 250, 320
522 CALL ATLC(E, T, RL(1), RL(2), E, E, E, P, SANGL, M, M, C, D, D, D)
523 IF(SANGL*SANGL) 350, 350, 320
524 SANGL = SANG
525 IF(SGN*(RL(1) - MPX)) 320, 230, 330
526 I = I - 1
527 IF(1) 240, 140, 325
528 RL(2) = BNDY(K+2)
529 MPX = BNDY(K)
530 K = K - 1
GU TO 315
531 IF(SGN) 376, 376, 345
532 SGN = -1
533 I = 1 - 1
534 IF(I) 340, 340, 325
535 RL(2) = BNDY(K+2)
536 MPX = BNDY(K)
537 GU TO 315
538 CALL ATLC(G, T, OR(1), OR(2), E, DELTA1, E-12, DELTA11, M, TANG, ISK, 1
539 M, D, D, D)
540 IF(ISK) 354, 354, 325
541 IF(AUSER(RL) = TL) 325, 225, 355
542 IF(SGN*(RL(1) - MPX)) 350, 330, 330
543 RL(1) = RL(1) + SIGN (DELTA, SGN)
544 RL(1) = RL(1)
545 CALL RTLC(E, T, RL(1), RL(2), C, DELTA1, E-10., M, ANGLE, ISW, M, C, D, D, D)
546 IF(ISW = 1) 255, 255, 325
547 IF(I = 1) 240, 240, 365
548 IF(SGN*ANGLE) 370, 367, 367

Approved for Public Release
303 IF(NSN(*ANS(ANGLE)-P1)) 370,367,367  
307 IF(1(4)) 366,355,355  
309 WRITE((10w,5f) ) RB(1),RB(2),TANG,ANGLE,NA  
GO TO 315  
310 NSP = NSP + 1  
311 SNSP = *RP(1)  
315 IF(NSN(*TUL)) 371,371,372  
317 SYN(SNP) = C.  
GO TO 372  
372 SYN(SNP) = RB(2)  
374 IF(1(4)) 374,375,374  
375 WRITE((10w,56) ) NSP,RB(1),RB(2),TANG,ANGLE,NA  
377 IF(NSP-100) 315,371,377  
379 IF(NSP) 372,377,379  
377 IF(1(4)) 374,375,378  
375 WRITE((10w,56) )  
379 NPQINT = 1  
GO TO 992  
391 N4 = NBR + NSP  
C WRITE MEASING  
CALL CLASS(FINCLS)  
403 WRITE((1w,1)) NRUN,TITLE  
WRITE((10w,2) ) NORD,NP,NU,TAU  
IF(NP-NZ) 4C5,425,43C  
405 IF(NP-L) 411,415,41C  
410 JSW = 1  
GO TO 42C  
415 WRITE((10w,5f) )  
420 WRITE((1w,5f) )  
421 WRITE((10w,4) ) ZR(I),ZI(I),I  
GO TO 460  
420 ABL =NPL  
422 = NPL+1  
423 = NPL  
GO TO 450  
425 JSW = 1  
427 = NZL  
GO TO 450  
430 IF(NZL) 44C,44C,435  
432 JSW = 3  
GO TO 445  
430 JSW = 4  
440 = NZL  
442 = NZL+1  
443 = NZL  
450 IF(JSW=2) 455,455,468  
455 WRITE((1w,13) )  
GO 458 I=1,PR1  
4 = XMPT(1)  
N = XM7(1)  

230
C

STARTING POINT CALCULATION ROUTINE

K = 3

L = 1

NAPI = 1

NAU = K-NBR

IF (NAU) 510,515,502

RLX(L) = 510(NAB)

LYL(L) = 510(NAB)

MP = 1

ISGND(L) = 1

GJ = GJ 1(550, 546, 547, 555), 15s

AX = 0

KX = 0

J = J

IF (J) 510, 515, 502

RLX(L) = 510(RLX(L) + ARLX(L), 15s, ARLX(L), 15s)

NEXT 503, 504

C

Approved for Public Release
AJ = XM
CONTINUE
GO TO 512

AM = XM * XM
CONTINUE
GO TO 512

IF(MODE) 51F, 51G, 51H
KLY(L) = Z(I(NB1))
MP = XM(I(NB1))
ISGNL(L) = 4
GO TO 520

KLY(L) = Z(I(NB1))
KLY(L) = Z(I(NB1))
MP = XM(I(NB1))
ISGNL(L) = 5

IERR(L) = BNDY(I(NB1)) 55L, 522, 522
IERR(L) = BNDY(I(NB1)) 55L, 522, 522
IERR(L) = BNDY(I(NB1)) 55L, 522, 522
IERR(L) = BNDY(I(NB1)) 55L, 522, 522

CALL RTELOC(I(NB1),RKL(L),KLY(L),E,0,E,E,E,M,15W,NT,D,D,D)
IF(ISMS) 54C, 54E, 54F
CALL RTELOC(I(NB1),RKL(L),KLY(L),E,0,E,E,E,M,15W,NT,D,D,D)
IF(ISMS) 54C, 54E, 54F

ANGL = (-SANG + TWOPXJ)/XM
GO TO 55C

ANGL = (-SANG + TWOPXJ)/XM
GO TO 55C

APX(I) = Z(I(NT))
APY(I) = Z(I(NT))
GO TO 54E

APX(I) = PRINT
APY(I) = PRINT

L = L + 1
K1 = X1 + 1
IF(LSHP) 55L, 554, 554
NC = 1
GO TO 575

IF(L-2) 55C, 55C, 55C
IF(L-NR) 55C, 55C, 55C
IF(L-L1) 55C, 55C, 55C
NC = L - 1
GO TO 575
NC = 3
L = 1
C BRANCH CALCULATION ROUTINE (3 BRANCHES)
NPINT = 1

232
GU TU 492
541 NW = NC
542 NPAGE = 1
543 NWOUT = 0
544 NFP = 1
545 NFMT = 1
546 NFMTL = 1
547 GO 600 I=1,NC
548 IF(MOUT) 562,562,564
550 IDOUT(I) = I
551 GO TO 59C
552 IDOUT(I) = MOUT
553 IF(MAGAIN) 566,566,566
554 IGAIN(I) = I
555 GO TO 59C
556 IGAIN(I) = AGAIN
557 IF(MPLOT) 552,552,594
559 IPLOT(I) = 1
560 GO TO 59C
561 IPLOT(I) = MPLOT
562 IF(MPLOTG) 551,551,598
564 IPLOTG(I) = 1
566 GO TO 66C
568 IPLOTV(I) = MPLOTC
570 IRK(I) = -1
573 IF(MPLOT) (24,14,604
574 IF(IPSAM(2-NSYM(II) - 2) 14,606,606
576 GO 610 I=1,NC
577 KM = KM + 1
579 IF(KM=10) 61G,61G,61G
580 KB = 1
581 NSYM(I) = KB
584 IF(MOUT) 6247,6247,615
C WRITE FRANCH: READING
590 MD = K1 + 1-NC
591 NLINE = 1
C CALL CLASS(INCLS)
593 WRITE(?0,oit) TITLE
595 WRITE(?0,11) NRUN
597 IF(NOUTU) 616,616,616
615 NPAGE = NPAGE + 1
616 WRITE(?0,99) NPAGE
618 IF(NOUTU) 624,624,624
617 GO TO 6222,622,622
618 WRITE(?0,12) (1+1-KM,K1)
619 GO TO 6247
620 WRITE(?0,14) (I=KM,K1)
621 GO TO 6247
622 WRITE(?0,15) K1
623 GO TO 6247
624 WRITE(?0,40) K1
625 IF(I(I)) 6242,6242,6241

233
6241 WRITE(70, 621)
6242 IF(ISNL(1)) 6244, 6246, 6249
6243 WRITE(70, 622)
6244 IF(ISNL(1)) 6246, 6248, 6245
6245 WRITE(70, 611)
6246 WRITE(70, 644)
6247 IF(NROUT) 625, 625, 716
625 L=1
626 IF(ISNL(1)) 631, 700, 632
631 (RLX(T1) = 1) NSG = NSNL(1)
632 (FIOSW) 631, 635, 6312
633 X0 = RLX(1)
634 Y0 = RLY(1)
635 SG = C.
636 ANGD = ANGI(1)
637 GU TO 6313
638 SANG = 0.
639 ANGLE = ANGI(1)
640 X1 = 1
641 NLC = 0
642 NSTEP = 1
643 GU TO 635
644 (FIIOFLAG(1) 633, 633, 760)
645 ROUS LAGUS CALCULATION ROUTINE.
646 NSNL = 1
647 NSTEP = 1
648 IF(IFLAG(1)) 1015, 1026, 1100
649 1000 IF(IFLAG(1)=10.4) 1026, 1102, 1100
650 1102 IF(IFIOSW) 1009, 1102, 1104
651 1003 X0 = XT
652 Y0 = YT
653 ANGD = AT
654 SG = D
655 GU TO 1008
656 ANGLE = ANGI(1)
657 SANG = E
658 GU TO 1008
659 NSNL = IFLAG(1) + 2
660 1008 IFLAG(1) = 0.
661 IF(NSNL = 6) 1010, 1555, 1555
662 1010 AX(I) = APX(I)
663 RLY(I) = APY(I)
664 IF(NSNL = 7) 1959, 1955, 1012
665 1012 X0 = RLX(1)
666 Y0 = RLY(1)
667 GU TO 1999
668 1015 STEP = -FLAG(1)
669 IFLAG(1) = C.
670 X = APX(I)
671 Y = APY(I)
672 GU TO 1025

234
1020   STEP = DELT
       X = RAX(I)
       Y = RLY(I)

1025   ANGLE = ANG(I)
       CALL RTLOC(NLC,G,X,Y,ANGLE,STEP,TOL,ACHG,15,SANG,1S\n       NS,NS,DX,DX)
       AND(SANGD)

       GO TO 1100,1105,1100,1120,1120

1100   X = X
       Y = Y
       ANG = ANGLE
       IF(INPREC) 1130,1030,1200

1130   GO TO 1120,1125,1100,1105

1050   APX(I) = PRX(NS)

1055   APY(I) = PRY(NS)

1100   NNSNL = 3
         GO TO 1150

1120   NNSNL = 6
       GO TO 1195

C       BOUNDARY CHECK
1100   IF(XAYDY) 1120,1120,1105

1105   IF(XAYDY) 1110,1120,1120

1110   IF(XAYDY) 1120,1120,1115

1115   IF(XAYDY) 1125,1120,1120

1120   NNSNL = 3
       GO TO 1150

1125   ANG(I) = ANGLE

1130   ALU(I) = Y

C       ADAPTIVE LOOP (STEP SIZE ADJUSTMENT IN CRITICAL AREAS)
1100   CALL RTLOC(R1,E,E,E,E,E,E,E,E,NS,NS,DX,DX,DX)

1105   NNSNL = 6
         CALL RTLOC(NNSNL,E,E,E,E,E,E,E,E,NS,NS,DX,DX,DX)

1110   CALL RTLOC(NNSNL,E,E,E,E,E,E,E,E,NS,NS,DX,DX,DX)

1115   CALL RTLOC(NNSNL,E,E,E,E,E,E,E,E,NS,NS,DX,DX,DX)

1120   CALL RTLOC(NNSNL,E,E,E,E,E,E,E,E,NS,NS,DX,DX,DX)

1125   IF(INP) 1120,1120,1120

1130   IF(INP) 1121,1210,1212

1135   IF(INP) 1121,1210,1212

1140   IF(INP) 1121,1210,1212

1145   IF(INP) 1121,1210,1212

1150   IF(INP) 1121,1210,1212

1155   IF(INP) 1121,1210,1212

1160   IF(INP) 1121,1210,1212

1165   IF(INP) 1121,1210,1212

1170   IF(INP) 1121,1210,1212

1175   IF(INP) 1121,1210,1212

1180   IF(INP) 1121,1210,1212

1185   IF(INP) 1121,1210,1212

1190   IF(INP) 1121,1210,1212

1195   IF(INP) 1121,1210,1212

1200   IF(INP) 1121,1210,1212

1205   IF(INP) 1121,1210,1212

1210   IF(INP) 1121,1210,1212

1215   IF(INP) 1121,1210,1212

1220   IF(INP) 1121,1210,1212

1225   IF(INP) 1121,1210,1212

1230   IF(INP) 1121,1210,1212

1235   IF(INP) 1121,1210,1212

235

Approved for Public Release
1230 X = XD
    Y = YD
    ANGLE = @ANGD
    GO TO 112C
1235 JR = JR-1
    STEP = STEP+4.
    KH = NL(JR)
    GO TO 1215
1240 IF JR=3J 1245, 1245, 1246
1245 NL(JK) = KR
    JR = JR+1
    STEP = STEP+4.
    GO TO 1205
1246 GO TO (110C, 1248, 1241, 125C, 10659, 1SW)
1247 RLX(I) = PR(NS)
    RLY(I) = PI(NS)
    GO TO 1249
1248 RLY(I) = ZR(NS)
    RLY(I) = ZI(NS)
1249 NSIGNAL = ISW+2
    XC = RLX(I)
    YO = RLY(I)
    GO TO 1155
C  BREAKAWAY POINT LOCATION ROUTINE
1253 BPX = XD
    BPY = YD
    IR = 0
1255 CALL RTLOC3(3, 1, BPX, BPY, E, TOLBKY, D10L, 20, E, ISW, NS, XT, YT, U, D)
1256 IF(ISW=1255, 1256, 1252)
1257 IF(HR = 1) 1253, 1253, 1256
1258 IF(PR = X0, =STEP*COS(ANG(I))
    IF(BPY) 1254, 1251, 1254
1259 BPX = YC, 5*STEP*SIN(ANG(I))
    GO TO 1251
1260 XD = XT
    YD = YT
1261 X = XD
    Y = YD
1262 IF(LABS(Y-TOLBKY) 1250, 125B, 1260
1264 Y = 0.
1265 BPX = X
    BPY = Y
    SANG = C.
    SANG = D.
    ANCT = ANGC
C  FIND TAKEOFF ANGLE FROM BREAKAWAY POINT
1275 IR = 0
    NSGH = 2
    STEP = DELTA
1280 FLAG(I) = STEP-DELT
    GO 129C MA=11, 1C

236

Approved for Public Release
TANG = ANG1 - DANG1
CALL RLOC(4,1,\#STEP*COS(TANG),\#STEP*SIN(TANG),TANG,STEP,E,\#1,
\#6,\#15,\#NS,\#XY,\#YT,\#AT,\#D)
GO TO 1292
ANG2 = TANG
1285 CALL RLOC(2,1,\#PX,\#PY,\#ANG2,\#XY,\#YT,\#AND,\#AT,\#D)
GO TO 1290
CONTINUE
1290 IN(1,4) = 1294,1298,1299,1300
1292 STEP = DELT/10
IK = 1
GO TO 1280
1298 FLAG(1) = 4
GO TO 1170
1300 APX(11) = RPX
APX(11) = RPY
ANG2 = TANG
1110 IF(FLAG(1)) = 130,1320,1130
1320 FLAG(1) = 4C
GO TO 1170
1999 IF(INSONGL=1) = 63,634,635
634 IMAX(1) = 6
ICMK = 1
640 SIGNAL(1) = SGNL
IN(1,1) = IPLOT(1) = 1
IPLOT(1) = IPLOT(1) + 1
GO TO 660
638 IPLOT(1) = 1
CALL RLPROT(1,\#SWX,\#LX(1),\#LX(1),\#NSYM(1),\#SC,\#NSTCR,\#STR,\#NPX,
\#NDF,\#NDFN,\#MARK,\#INT,1,\#SYMX,\#GGRID,\#SCALE,\#TITLE)
643 IF(IPLOT(1)=\#MPNTC) = 642,644,660
642 IPLOT(1) = IPLOT(1) + 1
GO TO 660
644 IPLOT(1) = 1
CALL RLPROT(1,\#SWX,\#LX(1),\#LX(1),\#NSYM(1),\#SC,\#NSTCR,\#STR,\#NPX,
\#NDF,\#NDFN,\#MARK,\#INT,2,\#SYMX,\#GGRID,\#SCALE,\#TITLE)
560 IF(IUT(7)=\#MULTI) = 562,564,76C
662 IUT(1) = IUT(1) + 1
GO TO 67C
664 IUT(1) = 1
IF(\#GAIN(1)=\#MAGAIN) = \#66,\#66,\#670
660 \#GAIN(1) = \#GAIN(1) + 1
GO TO 67C
66A \#GAIN(1) = 1
IFLAG(1) = 1
673 GO TO \#66C,\#672,\#674,\#676,\#677,\#67A,\#GSONL
672 IFLAG(1) = 1
GO TO 67B
674 IFLAG(1) = 0
675 IF(IGFLAG(1) = 1)
680 IF(IGFLAG(1) = 7)
685 CALL RLTDCMODNAIN, RLX(1),RLY(1), E,F,E,F,M,GAIN(1),TOSW,NEFI,
690 Ch,Di,Dj)
695 IF(NEFI(1) = 0)
696 IF(IGFLAG(1) = 5)
697 IF(IGFLAG(1) = 1)
698 IF(IGFLG(1) = F1,7C1,7C1)
700 GAIN(1) = -GAIN(1)
701 IF(GAINFCTR = 0.01) GAIN(1) = -GAIN(1)
702 IF(NEFI(1C) = 1)
703 J1 = 1 + 1
704 GO TO 630
710 IF(MUUT) TIE, 715, 715
715 IF(IGFLAG(1) = IGFLAG(2) = IGFLAG(3) = NW) 625, 716, 716
716 NLINE = NLINE + 1
717 IF(NLINE = 5) 7162, 7162, 7162
718 NDUT = 1
719 GO TO 615
715 IF(IUSW) 717, 717, 717
717 NGN = IGFLAG(1) + IGFLAG(2) + IGFLAG(3)
718 NGF = NGN + 1
719 IF(NGFW = NW) 71F, 725, 725
720 IF(NGF) 721, 725, 720
720 IF(IGFLAG(1) = 7) 711, 721, 724
721 NGFMT = 4 + IGFLAG(2)
722 GO TO 725
724 NGFMT = 3 + IGFLAG(2)
725 NG = IGFL(1) + IGFL(2) + IGFL(3)
726 NGFL = NG + 1
727 IF(NGFL(1) = 7) 726, 726, 726
727 IF(NGFL(2) = 3) 732, 732, 732
728 NGFL = 4 + IGFL(2)
729 GO TO 73C
730 NGFL = 4 + IGFL(2)
731 GO TO 73C
732 NGFL = 4 + IGFL(2)
733 GO TO 73C
734 NGFL = 4 + IGFL(2)
735 NGFL = 4 + IGFL(2)
736 NGFL = 4 + IGFL(2)
737 NGFL = 4 + IGFL(2)
738 NGFL = 4 + IGFL(2)
739 NGFL = 4 + IGFL(2)
740 G3 TO (731, 732, 724, 727, 735, 736, 738, 737), NGFMT
741 WRITE(10w, 20) (SIGNAL(I),RLX(I),RLY(I),GAIN(I),I,1,NW)
742 WRITE(10w, 11) (SIGNAL(I),RLX(I),RLY(I),GAIN(I),I=1,NWR,2)
743 WRITE(10w, 22) (SIGNAL(I),RLX(I),RLY(I),GAIN(I),I=2,NWR)
744 WRITE(10w, 23) SIGNAL(3),RLX(3),RLY(3),GAIN(3)
745 GO TO 752
746 WRITE(10w, 24) (SIGNAL(I),RLX(I),RLY(I),I=1,NW)
747 WRITE(10w, 25) (SIGNAL(I),RLX(I),RLY(I),I=1,NWR,2)
748 GO TO 73A
749 WRITE(10w, 26) (SIGNAL(I),RLX(I),RLY(I),I=1,NWR,2)

238
GO TO 736
736 WRITE(14,27) SIGNAL(1),RLX(1),RLY(1)
740 GO TO (745,746,747,748,749) ,NGFM1
740 WRITE(14,28) (GAIN(I),I=1,4,NGM)
GO TO 740
740 WRITE(14,28) (GAIN(I),I=1,2,NGM)
GO TO 740
741 WRITE(14,29) (GAIN(I),I=1,AGM,2)
GO TO 744
742 WRITE(14,30) (GAIN(I),I=2,NGM)
GO TO 740
743 WRITE(14,31) GAIN(3)
744 NGFM1 = 1
750 GO TO (752,753,754,755,756) ,NGFM1
746 WRITE(14,28) (GAIN(I),I=1,1,NGL)
GO TO 75C
754 WRITE(14,29) (GAIN(I),I=1,1,AGSL,2)
GO TO 75C
755 WRITE(14,29) (GAIN(I),I=1,2,AGSL,2)
GO TO 75C
757 WRITE(14,31) GAIN(3),NE(3)
759 NGFM1 = 1
760 GO TO 754
762 G = 754 T = 1.3
10 IFLAG(11) = C
14 IFLAG(11) = C
16 IFLAG(11) = C
763 IF(CHKK) = .625,E75,.77C
770 IF(P=6)E11 .772,772,7EC
772 NFP=34,IBLK(2)
GO TO 785
773 IF(IGFLAG(11)(IGFL12)) = .775,775,774
774 WRITE(14,29) SIGNAL(1),RLX(1),RLY(1),SAND,ANGLE,GAIN(1),NE(1)
1 NIA,NID,NSTEP
GO TO 752
775 WRITE(14,29) SIGNAL(1),RLX(1),RLY(1),SAND,ANGLE,NIA,NID,NSTP
GO TO 752
776 IF(IGFLAG(11)(IGFL12)) = .776,776,777
777 WRITE(14,31) SIGNAL(1),XD,YD,SAND,SAND,AIA,NE(1),NIA,NID,
1 NSTEP
GO TO 752
778 WRITE(14,31) SIGNAL(1),XD,YD,SAND,SAND,AIA,NID,NSTEP
GO TO 752
190 NFP = 2 - IBLK(2)
785 NW = IBLK(11) + IBLK(21) + IBLK(13)
NBR = NW + 1
10 IKK = 0
12 IF(NW) .675C,79C,725
790 IF(ISW=21) = 754,754,752
792 ISW = 3
796 IF(ISWAC=1) = .8CC,7CC,756
796 ISWAC = 2
800 IF(ENAM) = .8EC,7EC,760
205 K = K+1
[END]MP] = 81C,815,815
CONTROLS

ISYM = 31
GO TO 930
930 IF(I2-RL1) 925, 925, 935
925 RL1 = 2R1(I2)
RL2 = I2(II)
ISYM = 32
940 IF(IPL5W=1) 95, 975, 96C
955 KSW = 2
965 GJ TO 945, 965, 96C, 96F, IPLSW
975 IPLSW = 2
985 GO TO 97E
990 ISYM = 1
NP0INT = 4
GO TO 992
995 IF(MPLOTIC.LE.3) G0 TO 10C
10C IPLSW = 3
GO TO 995
1050 IF(KP5L5W) 970, 965, 97C
1060 NSYM = NSYM + 1
1070 IF(NSYM=10) 984, 985, 570
1070 IPLSW = 4
KPOLOC = 0
GO TO 98C
975 CALI RLPLOTI(KSW,ISWA,RL1,RL2,ISYM,1,SC,NSTCR,KSTOR,NPX,NOF,
JXUN=0,NXN=1,1,CMGRID,MSCALE,TITLE)
GO TO 98E
980 CALI RLPLOTI(KSW,ISWA,RL1,RL2,ISYM,1,SCC,NSTCR,KSTOR,NPXC,
NUN,NCRLA,C,NXN1,1,CMGRID,MSCALE,TITLE)
985 I = I+1
990 G0 TO 51C
996 NP0INT = 5
GO TO 992
997 IF(ABS(KP5L5W)=2) SEE, 585, 599
998 LSWC = 1
NSYM = 1
GO TO 995
999 LSWC = 2
1000 KPSL5W = C
GO TO 101
C  TIMER R0LRT1N
1020 IF(ITSW) 593, 594, 595
1030 CONTINUE
N TIME = 100.C
T = N TIME/C.
TIME = 1-T/C
T = T
TRUN = TRUN + TIME
IF(POINT=1) 577, 994, 994
1040 IF(ITSW) 933, 934, 935
1050 WRITE(10,32) NRUN, TRUN
WRITE(10,33) T

241

Approved for Public Release
TAUK = 0
GO TO 101
997 WRITE(101,17) NPOINT,TIME
998 GO TO 1581,452,5CC,5SC,9E7,311,380,167,167,97E1NT
<table>
<thead>
<tr>
<th>REPORT NUMBER</th>
<th>APPDL-73-144, Volume II</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTHOR(S)</td>
<td>Donald W. Vogt</td>
</tr>
<tr>
<td>CONTRACT OR GRANT NUMBER</td>
<td>F33615-73-C-3064</td>
</tr>
<tr>
<td>CONTROLLING OFFICE NAME AND ADDRESS</td>
<td>Air Force Flight Dynamics Laboratory APPDL/FDC/VTI656 Wright-Patterson Air Force Base, Ohio 45433</td>
</tr>
<tr>
<td>DISTRIBUTION STATEMENT</td>
<td>Approved for public release; distribution unlimited</td>
</tr>
</tbody>
</table>

**KEY WORDS**
- COV Handling Qualities Analysis
- Pilot Vehicle Analysis
- Control Configured Vehicles
- Dynamic Handling Qualities
- Pilot Ratings
- Pilot Models

A method for predicting aircraft pilot ratings which is applicable to control configured vehicles (CDOVs) is developed based on aircraft performance and pilot workload. A discussion of this method of handling quality analysis, and a description of four random tracking pilot rating correlations developed using the method are presented in Volume I of the report. A literature survey classifying the most recent reports applicable to flying qualities is tabulated. A user's manual for McPilco, a general purpose digital computer...
program developed to implement the method and correlate pilot rating data.
is presented in Volume II. The correlations that were developed are used
to evaluate the pilot ratings of two conventional contemporary fighter
aircraft configurations. The correlations are also applied to two CCV
fighter configurations for which limited amounts of handling quality data
had been obtained through manned aircraft simulations.