ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM. VOLUME I

E. J. West, B. R. Umhel, R. F. Yurczyk
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM

Christopher L. West
Brian R. Ummel
Roger F. Yurczyk

Boeing Military Airplane Company
P.O. Box 3707
Seattle, Washington 98124

SEPTEMBER 1980

Final Report for Period May 1979 to September 1980

Approved for public release; distribution unlimited

Flight Dynamics Laboratory
Air Force Wright Aeronautical Laboratory
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433
NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

LANNY A. JINES
Project Engineer

James M. Peters, Acting Chief
Aircrew Escape Group
Crew Escape & Subsystems Branch
Vehicle Equipment Division

FOR THE COMMANDER

AMBROSE B. NUTT
Director
Vehicle Equipment Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIER, Wright-Patterson AFB, OH 45433 to help us maintain a current mailing list."

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/17 February 1981 — 75
<table>
<thead>
<tr>
<th>REPORT DOCUMENTATION PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLUME 1</td>
</tr>
</tbody>
</table>

1. REPORT TITLE
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM

2. TYPE OF REPORT
Final Rep.

3. DATE COVERED
May 1978-Sept. 1980

4. PREPARING ORG. REPORT NUMBER
F33615-79-C-3477

5. CONTRACT OR GRANT NUMBER(S)
F33615-79-C-3477

6. Performing ORG. REPORT NUMBER
F33615-79-C-3477

7. PROJECT NUMBER(S)

8. PRIME CONTRACTOR/AGENCY
Boeing Military Airplane Company
P.O. Box 3999
Seattle, Washington 98124

9. SOURCE AND DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited.

10. SECURITY CLASS. (of this report)
Unclassified

11. SECURITY CLASS. (of this report)
Unclassified

12. DISTRIBUTION STATEMENT (of this report)
Approved for public release; distribution unlimited.

13. NUMBER OF PAGES
675

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Flight Dynamics Laboratory
Air Force Wright Aeronautical Laboratories
Air Force Systems Command, WPAFB, Ohio 45433

15. KEY WORDS
Computer Simulation
Stability & Control
Ejection Seat

16. ABSTRACT
High performance combat aircraft have extended the maneuvering/operating range into regimes that exceed the capabilities of current ejection seat systems. One of the problems encountered involves the unstable rotational characteristics of the typical ejection seat, resulting in a decreased probability of survival due to the reorientation of the ejecting crewmember into an attitude less tolerant to acceleration. Furthermore, an unstable ejection seat may neither clear the airframe nor provide adequate ground clearance. The capability to simulate the trajectory of an escape system, and to determine its stability...
The Simulation and Analysis of In-Flight Escape System Techniques (SAFEST) computer program, developed by the AFFDL for the analysis of occupied ejection seat stability characteristics, is a six degree of freedom simulation of an ejection system. SAFEST uses a fourth order Runge-Kutta integrator with a fixed time step to calculate the trajectories for the seat/man, man alone, airplane, drag parachute, and the recovery parachute. However, SAFEST does not have the capability to perform classical stability analyses.

The EASY program, originally developed by Boeing under Air Force Contract, is a general purpose program for the linear and nonlinear analysis of system dynamics using classical techniques. It has been used to model a variety of systems including environmental control systems, aircraft flight controls and dynamics, space vehicle dynamics, electrical power generation, rapid transit vehicles and air cushion landing systems.

The objective of this development effort was to develop an ejection seat classical stability analysis capability by incorporating SAFEST simulation subroutines into the EASY standard component library. The resultant computer program described in this User Manual/document is EASY And SAFEST Integration for the Evaluation of Stability and Trajectory (EASIEST).

Although EASY was originally developed under contract to the Air Force, additional Boeing funded research and development effort was undertaken to improve the program and increase its capability. The resultant improved version, EASY5, formed the basis for development of EASIEST. Because these added capabilities were developed using Boeing funds, they remain proprietary to the Boeing Company. Therefore, the program documentation/user manual is contained in two volumes. Volume I is a "stand-alone" user manual describing the EASIEST program characteristics and complete information on the use of the program and how to apply it to ejection seat dynamics and control analysis. It contains listings of the procedure files, models, analysis, standard components, and subroutines. Volume II is Boeing proprietary and contains only the source code listings of EASY5.
This report describes research work performed by the Boeing Military Airplane Company, Seattle, Washington, for the Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratory, Wright-Patterson Air Force Base, Ohio, under Contract No. F33615-79-C-3407, Project 2402, "Vehicle Equipment Technology," work unit 24020328, "Ejection Seat Stability and Control Analytical Computer Program." Project engineer for the contract was Lanny A. Jines, AFWAL/FIER. This research work is part of an effort to develop an escape system computer simulation for performance analysis of ejection seat dynamics during escape. This report is in two volumes and combines the technical report and user manual. Volume I is the EASIEST "stand alone" user manual. Volume II contains the Boeing proprietary EASY5 source code. Volume II shall not be disclosed outside of Government agencies for a three-year period following completion of this contract and may be extended for an additional three-year period or successive three-year periods, by agreement between The Boeing Company and the Government.

The work reported herein was performed during the period of May 1979 to September 1980.

Roger F. Yurczyk served as the program manager. The technical work was performed by Christopher L. West and Brian R. Ummel, with consultation from John D. Burroughs.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.   INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II.  MODEL GENERATION</td>
<td>5</td>
</tr>
<tr>
<td>1.   NAMING CONVENTION</td>
<td>6</td>
</tr>
<tr>
<td>a. Standard Component Naming Conventions</td>
<td>6</td>
</tr>
<tr>
<td>b. State, Variable, Parameter, and Table Naming Conventions</td>
<td>6</td>
</tr>
<tr>
<td>2.   MODEL DESCRIPTION</td>
<td>9</td>
</tr>
<tr>
<td>a. Phrases and delimiters</td>
<td>9</td>
</tr>
<tr>
<td>b. Command Phrases</td>
<td>12</td>
</tr>
<tr>
<td>3.   MODEL SCHEMATIC</td>
<td>29</td>
</tr>
<tr>
<td>a. Standard Schematic Form</td>
<td>29</td>
</tr>
<tr>
<td>b. Input Quantity Labeling</td>
<td>29</td>
</tr>
<tr>
<td>c. Component Connection Paths</td>
<td>30</td>
</tr>
<tr>
<td>d. Additional Pages</td>
<td>30</td>
</tr>
<tr>
<td>e. Guidelines for Schematic Layout</td>
<td>30</td>
</tr>
<tr>
<td>4.   WARNING MESSAGES</td>
<td>32</td>
</tr>
<tr>
<td>III. DYNAMIC ANALYSIS OF CONTINUOUS OR DISCRETE SYSTEMS</td>
<td>38</td>
</tr>
<tr>
<td>1.   MODEL INPUT DATA</td>
<td>39</td>
</tr>
<tr>
<td>a. Scaler Data</td>
<td>39</td>
</tr>
<tr>
<td>b. Tabular Data</td>
<td>39</td>
</tr>
<tr>
<td>c. Matrix Data</td>
<td>43</td>
</tr>
<tr>
<td>2.   INITIAL CONDITION, ERROR, AND INTEGRATION CONTROLS</td>
<td>44</td>
</tr>
<tr>
<td>3.   INITIAL CONDITION COMMANDS</td>
<td>46</td>
</tr>
<tr>
<td>4.   SIMULATION COMMANDS</td>
<td>47</td>
</tr>
<tr>
<td>5.   PLOT DESIGNATION COMMANDS</td>
<td>52</td>
</tr>
<tr>
<td>6.   STEADY STATE COMMANDS</td>
<td>55</td>
</tr>
<tr>
<td>7.   LINEAR ANALYSIS COMMANDS</td>
<td>56</td>
</tr>
<tr>
<td>8.   STABILITY MARGIN COMMANDS</td>
<td>58</td>
</tr>
<tr>
<td>9.   FREQUENCY RESPONSE COMMANDS</td>
<td>59</td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. ROOT LOCUS COMMANDS</td>
<td>60</td>
</tr>
<tr>
<td>11. EIGENVALUE SENSITIVITY COMMANDS</td>
<td>62</td>
</tr>
<tr>
<td>12. FUNCTION SCAN COMMANDS</td>
<td>63</td>
</tr>
<tr>
<td>13. OPTIMAL CONTROLLER DESIGN COMMANDS</td>
<td>64</td>
</tr>
<tr>
<td>14. WARNING MESSAGES</td>
<td>69</td>
</tr>
<tr>
<td>15. RENAMING MODEL INPUTS AND OUTPUTS</td>
<td>77</td>
</tr>
<tr>
<td>16. COMPUTING TYPE ZERO TRANSFER FUNCTIONS WITH EASY</td>
<td>78</td>
</tr>
<tr>
<td>IV. EASIEST STANDARD COMPONENTS AND SUBROUTINES</td>
<td>80</td>
</tr>
<tr>
<td>1. STANDARD COMPONENTS</td>
<td>80</td>
</tr>
<tr>
<td>2. SUBROUTINES</td>
<td>94</td>
</tr>
<tr>
<td>3. MODELING WITH THE EASIEST COMPONENTS</td>
<td>94</td>
</tr>
<tr>
<td>4. SEQUENCING AN EASIEST MODEL</td>
<td>99</td>
</tr>
<tr>
<td>V. PROCEDURES FOR INSTALLING THE EASY5 PROGRAM AND SUBMITTING AN EASIEST RUN</td>
<td>101</td>
</tr>
<tr>
<td>1. INSTALLING THE EASY5 PROGRAM</td>
<td>101</td>
</tr>
<tr>
<td>2. SUBMITTING AN EASIEST RUN</td>
<td>103</td>
</tr>
<tr>
<td>VI. EJECTION SEAT ANALYSIS EXAMPLE</td>
<td>105</td>
</tr>
<tr>
<td>VII. CREATING AND MODIFYING STANDARD COMPONENTS</td>
<td>145</td>
</tr>
<tr>
<td>1. MODIFYING THE FORTRAN SUBROUTINE OF AN EXISTING STANDARD COMPONENT</td>
<td>147</td>
</tr>
<tr>
<td>2. MODIFYING THE RANDOM ACCESS FILE EZSTDBF</td>
<td>148</td>
</tr>
<tr>
<td>3. CREATING A NEW EASIEST STANDARD COMPONENT</td>
<td>150</td>
</tr>
<tr>
<td>4. LIBRARY EZSTLIB SIZE REDUCTION</td>
<td>151</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIII. DESCRIPTION AND GUIDE TO USE OF NUMERICAL INTEGRATION</td>
<td>152</td>
</tr>
<tr>
<td>1. CHANGES IN INTEGRATORS</td>
<td>152</td>
</tr>
<tr>
<td>2. GENERAL SELECTION GUIDELINES</td>
<td>153</td>
</tr>
<tr>
<td>3. ACCURACY AND ERROR CONTROL</td>
<td>156</td>
</tr>
<tr>
<td>4. STABILITY</td>
<td>159</td>
</tr>
<tr>
<td>IX. DISCRETE SYSTEM ANALYSIS TECHNIQUES</td>
<td>166</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>166</td>
</tr>
<tr>
<td>2. SYSTEM EQUATIONS</td>
<td>166</td>
</tr>
<tr>
<td>a. Continuous System Stability Matrix</td>
<td>168</td>
</tr>
<tr>
<td>b. Discrete System Transition Matrix</td>
<td>169</td>
</tr>
<tr>
<td>c. Continuous System Transition Matrix</td>
<td>169</td>
</tr>
<tr>
<td>d. Calculation of Continuous System Transition Matrix</td>
<td>171</td>
</tr>
<tr>
<td>3. COMBINED SYSTEM TRANSITION MATRIX</td>
<td>172</td>
</tr>
<tr>
<td>a. Single Sample Rate</td>
<td>172</td>
</tr>
<tr>
<td>b. Integer Multiple Sampling Rate</td>
<td>175</td>
</tr>
<tr>
<td>c. Noninteger Multiple Sampling Rates</td>
<td>179</td>
</tr>
</tbody>
</table>

REFERENCES 182

APPENDIX A. MODEL GENERATION COMMANDS 183
APPENDIX B. ANALYSIS COMMANDS 185
APPENDIX C. CHECKLISTS 189
APPENDIX D. EASIEST INPUT/OUTPUT LISTS AND ASSOCIATED FIGURES 194
APPENDIX E. PROGRAM AEROMED 268
APPENDIX F. EASIEST PROCEDURE FILE 272
APPENDIX G. EASIEST STANDARD COMPONENTS 279
APPENDIX H. EASIEST SUBROUTINES 438
<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPENDIX I. FILOAD INPUT DATA</td>
<td>485</td>
</tr>
<tr>
<td>APPENDIX J. EASIEST F-4E MANEUVERING COEFFICIENTS</td>
<td>499</td>
</tr>
<tr>
<td>APPENDIX K. EASY5 INPUT/OUTPUT LISTS</td>
<td>510</td>
</tr>
<tr>
<td>APPENDIX L. EASY PROGRAM ANALYSIS DESCRIPTION</td>
<td>597</td>
</tr>
<tr>
<td>APPENDIX M. OPTIMAL CONTROLLER DESIGN WITH THE EASY PROGRAM</td>
<td>625</td>
</tr>
<tr>
<td>APPENDIX N. EASIEST EXAMPLE</td>
<td>657</td>
</tr>
<tr>
<td>FIGURE</td>
<td>PAGE</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Character Assignment in Input/Output or Table Name</td>
</tr>
<tr>
<td>2</td>
<td>Model Generation Program Schematic</td>
</tr>
<tr>
<td>3</td>
<td>Typical Component Input/Output Configurations</td>
</tr>
<tr>
<td>4</td>
<td>Component Connection Paths</td>
</tr>
<tr>
<td>5</td>
<td>Model Generation Program Input File</td>
</tr>
<tr>
<td>6</td>
<td>EASY Schematic</td>
</tr>
<tr>
<td>7</td>
<td>Analysis Program Input File</td>
</tr>
<tr>
<td>8</td>
<td>Steady State Output</td>
</tr>
<tr>
<td>9</td>
<td>Simulation Output</td>
</tr>
<tr>
<td>10</td>
<td>Example of Printer Plots</td>
</tr>
<tr>
<td>11</td>
<td>Stability Regions for Runge-Kutta Methods; Orders 1-5</td>
</tr>
<tr>
<td>12</td>
<td>Stability Regions for Adams-Moulton Formulas</td>
</tr>
<tr>
<td>13</td>
<td>Stability Regions for Predictor - Corrector Pairs</td>
</tr>
<tr>
<td>14</td>
<td>Stability Regions for STIFF GEAR Formulas of Orders 1-3</td>
</tr>
<tr>
<td>15</td>
<td>Stability Regions for STIFF GEAR Formulas of Orders 4-6</td>
</tr>
<tr>
<td>16</td>
<td>Example of Continuous, Delay, and Sample States</td>
</tr>
<tr>
<td>17</td>
<td>Single Sampling Rate Example</td>
</tr>
<tr>
<td>18</td>
<td>Pictorial Representation of Single Sampling Rate Transition Matrices</td>
</tr>
<tr>
<td>19</td>
<td>Multisampling Rate Example</td>
</tr>
<tr>
<td>20</td>
<td>Pictorial Representation of Multisampling Rate Transition Matrices-Integer Multiple Rates</td>
</tr>
<tr>
<td>21</td>
<td>Pictorial Representation of Multisampling Rate Transition Matrices-Noninteger Multiple Rates</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS (CONT'D)

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Description</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>Standard Component &quot;AP&quot; Input/Output Overview</td>
<td>208</td>
</tr>
<tr>
<td>23</td>
<td>Standard Component &quot;AS&quot; Input/Output Overview</td>
<td>212</td>
</tr>
<tr>
<td>24</td>
<td>Standard Component &quot;CT&quot; Input/Output Overview</td>
<td>224</td>
</tr>
<tr>
<td>25</td>
<td>Standard Component &quot;DR&quot; Input/Output Overview</td>
<td>227</td>
</tr>
<tr>
<td>26</td>
<td>Standard Component &quot;PC&quot; Input/Output Overview</td>
<td>245</td>
</tr>
<tr>
<td>27</td>
<td>Standard Component &quot;RL&quot; Input/Output Overview</td>
<td>250</td>
</tr>
<tr>
<td>28</td>
<td>Standard Component &quot;SP&quot; Input/Output Overview</td>
<td>260</td>
</tr>
<tr>
<td>29</td>
<td>Standard Component &quot;SP&quot; Gimbal Spring and Vernier Rocket</td>
<td>261</td>
</tr>
<tr>
<td>30</td>
<td>Standard Component &quot;SR&quot; Input/Output Overview</td>
<td>264</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>49</td>
</tr>
<tr>
<td>System Model Description</td>
<td>10</td>
</tr>
<tr>
<td>EASY5 Program Language Delimiters</td>
<td>12</td>
</tr>
<tr>
<td>Integration Method Selection</td>
<td>48</td>
</tr>
<tr>
<td>Print Control Values</td>
<td>49</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>AFFDL</td>
<td>Air Force Flight Dynamics Laboratory</td>
</tr>
<tr>
<td>ASD</td>
<td>Aeronautical Systems Division</td>
</tr>
<tr>
<td>DART</td>
<td>Directional Automatic Realignment of Trajectory</td>
</tr>
<tr>
<td>EASIEST</td>
<td>EASY And SAFEST Integration for the Evaluation of Stability and Trajectory</td>
</tr>
<tr>
<td>EASY</td>
<td>Environmental control Analysis SYstem</td>
</tr>
<tr>
<td>SAFEST</td>
<td>Simulation and Analysis of In-Flight Escape System Techniques</td>
</tr>
</tbody>
</table>
SUMMARY

High performance combat aircraft have extended the maneuvering/operating range into regimes that exceed the capabilities of current ejection seat systems. One of the problems encountered involves the unstable rotational characteristics of the typical ejection seat, resulting in a decreased probability of survival due to the reorientation of the ejecting crew-member into an attitude less tolerant to acceleration. Furthermore, an unstable ejection seat may neither clear the airframe, nor provide adequate ground clearance. The capability to simulate the trajectory of an escape system, and to determine its stability characteristics using classical stability and control methods, is required to enhance the development of both active and passive stability augmentation systems.

The Simulation and Analysis of In-Flight Escape System Techniques (SAFEST) computer program, developed by the AFFDL for the analysis of occupied ejection seat stability characteristics, is a six-degree-of-freedom simulation of an ejection system. SAFEST uses a fourth order Runge-Kutta integrator with a fixed time step to calculate the trajectories for the seat/man, man alone, airplane, drag parachute, and the recovery parachute. However, SAFEST does not have the capability to perform classical stability analyses.

The EASY program, originally developed by Boeing under Air Force Contract, is a general purpose program for the linear and nonlinear analysis of system dynamics using classical techniques. It has been used to model a variety of systems, including environmental control systems, aircraft flight controls and dynamics, space vehicle dynamics, electrical power generation, rapid transit vehicles and air cushion landing systems.

The objective of this development effort was to develop an ejection seat classical stability analysis capability by incorporating SAFEST simulation subroutines into the EASY standard component library. The resultant computer program described in this user manual/document is the EASY And SAFEST Integration for the Evaluation of Stability and Trajectory (EASIEST).
Although EASY was originally developed under contract to the Air Force, additional Boeing funded research and development effort was undertaken to improve the program and increase its capability. The resultant improved version, EASY5, formed the basis for development of EASIEST. Because these added capabilities were developed using Boeing funds, they remain proprietary to The Boeing Company. Therefore, the program documentation/user manual is contained in two volumes. Volume I is a "stand alone" user manual describing the EASIEST program and complete information on the use of the program and how to apply it to ejection seat dynamics and control analysis. Volume II is Boeing proprietary and contains only the EASY5 source code.
SECTION I
INTRODUCTION

The objective of the research work described in this document was to develop a stability analysis capability for ejection seat performance. This was accomplished by modifying ejection seat simulation subroutines from an Air Force Flight Dynamics Laboratory (AFFDL) computer program, Simulation and Analysis of In-Flight Escape System Techniques (SAFEST), into a component library compatible with the EASY computer program. The resultant computer program described in this document has been termed the EASY and SAFEST Integration for the Evaluation of Stability and Trajectory (EASIEST).

Technology improvements in advanced combat aircraft have expanded the operational maneuvering envelope beyond the capability of current ejection seats. The aerodynamic instability of ejection seats during entrance into the air stream has led to tumbling, spinning, parachute shroud fouling, and a variety of system failures. The resultant loads may exceed the tolerance limits of the human body. Experience from combat aircraft involving fatalities and severe injuries points to the need for the development of stable ejection seats whose performance is designed to be within human tolerance limits.

The AFFDL has an active technology program to enhance the stability of an ejection seat. One aspect of the current technology has been the development of SAFEST, an escape system computer program for performance analysis of ejection seat dynamics. However, an ejection seat stability study utilizing the SAFEST program demands numerous simulation runs. The results obtained then require followup analytical data reduction to identify the system stability characteristics.

The EASY program was originally developed under an Air Force contract to provide methods for modeling and analyzing aircraft environmental control systems. In 1976, a second Air Force contract extended the application of the program to include aircraft flight dynamics. Since October 1976, a
Boeing-funded research and development effort has been undertaken to modify the program for use on a wide variety of control system analyses. Additional effort during the last half of 1977 and 1978 resulted in the development of the EASY5 program. The program now includes component models for many types of vehicles and control components, matrix and vector notation at all program levels, capability to model and analyze continuous and discrete systems, larger modeling capacity, and the ability to store time history data on magnetic tape, to name a few.

EASY5, with its additional capability, was used as the basis for the development of EASIEST. Because the advanced features of EASY5 were developed by Boeing-funded research, they remain proprietary to the Boeing Company. Therefore, the program has been documented in two separate volumes. Volume I is a complete "stand alone" user manual. Volume II is Boeing proprietary and contains only the listings of the EASY5 source code.

In the context of this document, EASY refers to the basic dynamics analysis program (Model Generation Program and Analysis Program) as developed under Air Force Contract F33615-76-C-3100 and modified under contract F33615-76-C-3165. EASY5 refers to the latest version of the EASY program which is Boeing proprietary. EASIEST refers to the standard components and algorithms developed specifically for ejection seat system analysis.

The EASY5 program is a user oriented computer program designed to provide a simplified way to describe and analyze linear and nonlinear dynamic systems. This simplified system description is then used for a wide variety of system analyses including conventional linear analysis and nonlinear simulation. The EASY5 computer program consists of a Model Generation Program and an Analysis Program. Both continuous and sampled data systems may be described and analyzed. The modeling of most of the systems is accomplished by describing the system in terms of standard components which are subroutines that model specific hardware items, like rate gyros, or perform certain functions such as wind gust generation. The models of these standard components have been constructed in a general fashion so that by proper choice of input parameters and tables, a wide range of
specific, required system components can be modeled by each standard component. If a portion of a particular system to be studied cannot be described by using one of the standard components, FORTRAN statements can be directly included in the model description to implement those portions of the system. Using a simplified description of the system model, the EASY5 Model Generation Program generates the required FORTRAN subroutines which accurately represent the model in program form. This computer generated model can then be analyzed by any of the nonlinear, linear, dynamic, or steady state evaluation techniques available in the EASY5 Analysis Program. The capabilities include the following:

- Algebraic sensitivity
- Eigenvalue and Eigenvalue sensitivity* determination
- Frequency response (Bode, Nyquist, and Nichols plots)
- Linear model generation
- Nonlinear simulation (time histories)
- Optimal control synthesis*
- Root locus*
- Stability margins*
- Stability matrix calculation
- Steady state analysis

*These analyses are not available for discrete systems.

Volume I of this document provides information on the use of the EASIEST program and how to apply it to ejection seat dynamics and control analysis. Section II of Volume I presents the details of how to use the Model Generation Program to construct a model. Section III presents the details of how to conduct a system analysis with the Analysis Program. It discusses how to input the model data, set initial conditions, designate plots and to select the different analysis options. Section IV describes the EASIEST components which are used to form the ejection seat dynamic models. Section V contains the procedure for program execution. Section VI presents an ejection seat analysis example. Section VII describes the procedure for the modification of a standard component. Section VIII contains a
discussion of the numerical integration options available. Section IX presents a discussion of the methods used for discrete system analysis.

Lists of Model Generation and Analysis Program commands for easy reference are available in Appendices A and B.

Appendix C presents a program checklist to help ensure that the program is being properly utilized.

Appendix D contains input and output tables for all the EASIEST standard components. Descriptive figures are also presented for the more complex standard components.

Appendix E contains the listing of program AEROMED, a postprocessor which calculates the aeromedical variables.

Appendix F contains a listing of the EASIEST procedure file.

Appendix G presents listings of the EASIEST standard components, and Appendix H contains associated subroutine listings.

Appendix I has the FILOAD input data. FILOAD is a program which creates a random access file from input data that defines the variable names on the calling sequence for each standard component.

Appendix J contains the EASIEST F-4E maneuvering coefficients for the airplane component.

Appendix K contains input and output tables for the EASY5 standard components developed under previous contracts.

Appendices L and M present descriptions of analysis calculations and optimal controller design, reproduced from Sections 4.4 and 4.5 of reference 1.

Appendix N presents a supplementary ejection seat analysis example.
SECTION II
MODEL GENERATION

The EASY5 Model Generation Program uses a block diagram type of approach for constructing the different system models. It is based upon the assumption that the system analyst will construct a detailed schematic block diagram of the system to be analyzed. This detailed schematic will then be changed to a form containing standard components FORTRAN. The parts of a system which cannot be modeled using these standard components are included by appropriate FORTRAN statements in the system description.

All interconnections between the different standard components and the aforementioned FORTRAN statements are accomplished by the Model Generation Program. The analyst draws the block diagram by specifying the location of each standard component or FORTRAN block in the schematic diagram and all of the components that provide inputs to that component. The Model Generation Program then generates name labels and the proper interconnections between the specified components. This is accomplished by matching the input quantities required by each component to the output quantities of the components specified as providing inputs.

After processing the complete system model description, the Model Generation Program generates the schematic diagram of the model showing all of the interconnections between the components in a manner similar to the analyst's original diagram. It shows the quantities such as forces, moments, velocities, etc., that are used to form each interconnection. This schematic is produced on the lineprinter and provides a rapid graphic check on the program's interpretation of the model description.

In addition, the program produces a complete list of the input data that will be required by each component to complete the model description. The scalar and vector parameters and tabular data required for the analysis are included in this list. The program assumes that any quantity not supplied by another component will be supplied as a fixed parameter by the analyst.
Thus, requests for nonparameter items in the input data list reveal any connections that have been omitted from the system model description.

1. **NAMING CONVENTION**

   Every variable or state must have a unique name. FORTRAN limits these to seven characters. For standard components, the name is associated with the standard component name.

   a. **Standard Component Naming Conventions**

   All standard components are given names consisting of two characters, the first of which is alphabetical. Thus we have LA for lag, CT for catapult, SL for sled, etc. A specific component in a model is distinguished from other components of the same type by adding one or two more characters to the standard component name. These characters are usually numeric but can also be alphabetical or blanks. For example, a model using ten of the same type may have these components designated as:

   LA 1, LA 2, LA 3, ....LA10

   If matrix component notation is used, a single component may be defined as:

   LA 1, N=10

   This results in a single component LA 1 with a 10 vector assigned to those inputs and outputs with variable array length capability.

   b. **State, Variable, Parameter, and Table Naming Conventions**

   A consistent approach has been taken to the naming of inputs and outputs for standard components. This convention is denoted by Figure 1. As described in the figure, the standard component name is shown as the fourth and fifth character of the total element name. The last two characters are used to distinguish between several of the same component. The first three characters are used to designate the inputs and outputs of the components. The specific names of the input and output quantities for the
INPUT/OUTPUT OR TABLE NAMES

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY</th>
<th>STANDARD NAME</th>
<th>SPECIFIC NAME</th>
<th>COMP. NAME</th>
<th>COMP. IDENTIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORT NUMBER</td>
<td></td>
<td></td>
<td></td>
<td>SPECIFIC COMP. NAME</td>
</tr>
</tbody>
</table>

*Figure 1. Character Assignment in Input/Output or Table Name*
EASIEST components are listed on Appendix D. If a variable is a vector, subscripts must be added to the name when referring to a particular element in the array. An example of this would be S2 LA09 (2).

All of the input, output, and tabular quantities required by each component in a system model must have unique FORTRAN names. For standard components, these quantities are given names consisting of up to three characters that describe the physical quantity they represent. Since a single component may have several inputs or outputs of the same physical type, the program adds a "port" number as the second or third character of the physical quantity name to prevent such a duplication.

The physical quantities that are outputs of a standard component are specifically identified by adding the four character name of that component to the three character name of the physical quantity. In this way, unique seven character FORTRAN names are generated for all output quantities of the system model components. As an example, the output for standard component LA23 would be 52 LA23.

Input quantities to a component that are generated by another component carry the names of the component that generates them. Any inputs that are not satisfied by other model components are assumed to be parameters and are assigned the name of the component where they are an input.

If a component requires tabular data as an input, unique table names are generated just as scalar input quantity names by adding the component name to the table name. As an example, the input table for standard component SR11 would be TRFSR11.

All parameter, variable, and state quantities are set as real quantities even if their name starts with the FORTRAN integer letters I, J, K, L, M, N. Names added to the model via the ADD commands can consist of any valid FORTRAN name of up to seven characters. These names must not duplicate any name generated by the precompiler or other ADD statement.
2. MODEL DESCRIPTION

The Model Generation Program is a sophisticated precompiler which accepts model description instructions, and uses them to generate a FORTRAN model of the system. An EASY5 system model description contains numeric values, standard component names, and standard input and output quantity names. The instructions, referred to as "program commands," are made up of one or more functionally descriptive words.

The EASY5 commands may be best understood by using an example to describe a simple ejection seat model. The EASY5 system model description for it is given in Table 1.

As is seen in Table 1, the model description consists of a series of statements. Each statement specifies the location of each component in the schematic diagram and a list of all of the components that provide inputs to that component. The purpose of the location of the component in the schematic is to allow the Model Generation Program to use the line printer to draw a schematic of the model, such as shown in Figure 2. On the line printer drawn schematic, the input quantities to each component are shown. This can then be used to check functional flow for the diagram.

a. Phrases and Delimiters

The system model description is interpreted by the Model Generation Program from the command phrases following the program commands. The phrases must be separated by any one of the delimiter symbols shown in Table 2.

Comments can be inserted in the model description or analysis data by placing a "*" in column 1. These data cards will be ignored by the Model Generation or analysis programs.
TABLE 1  SYSTEM MODEL DESCRIPTION
EASY5 MODEL GENERATION PROGRAM  VERSION 3.1.2

INPUT COMMANDS

COMMAND CARD ---> MODEL DESCRIPTION=MODEL CONTAINING AG, RL, CT, SE, RS, AND CE COMPONENTS
COMMAND CARD ---> LOCATION=029  AG
COMMAND CARD ---> LOCATION=027  FORT
COMMAND CARD ---> LOCATION=027  ADD VARIABLES=CTFLAG
COMMAND CARD ---> IF (TIME .LT. 0.5) CTFLAG = 0
COMMAND CARD ---> IF (TIME .GE. 0.5) CTFLAG = 1.0

COMMAND CARD ---> LOCATION=022  SL  INPUTS=SL, SE(1:1), FORT(CTFLAG=SW)
COMMAND CARD ---> LOCATION=025  CT  INPUTS=SL, SE(1:1)
COMMAND CARD ---> LOCATION=025  RL  INPUTS=SL, SE(1:1)
COMMAND CARD ---> LOCATION=057  RSCS  INPUTS=SE(SOP=XPB, UST=UPB, EST=EPD, WST=WPB)
COMMAND CARD ---> LOCATION=053  SE  INPUTS=RSCS(FPB=F2, TPB=T2, 2)
COMMAND CARD ---> LOCATION=058  CE  INPUTS=RSCS

COMMAND CARD ---> END OF MODEL
COMMAND CARD ---> PRINT
TABLE 2

EASY5 Command Phrase Delimiters

= equal sign
, comma
( left parenthesis
) right parenthesis
three or more blanks

b. Command Phrases

The EASY5 command phrases are described in this section. They are presented in a sequence similar to that in which they would be used in system model descriptions. For easy reference, they are listed at the end of this section in alphabetical order and in Appendix A.

MODEL DESCRIPTION

The MODEL DESCRIPTION program command is used to indicate the start of a new system model. This command may be followed, on the same line, by a title of up to 60 characters. This title will be used throughout the printout to identify various program output schematics and program listings. In the example shown in Table 1, the title is "MODEL CONTAINING AG, SL, RL, CT, SE, RS, AND CE COMPONENTS".

LOCATION

The LOCATION program command indicates the start of a new component in the system model. This command must be followed by a numeric value phrase that specifies the location of the component on the model schematic. Thus, in the example of Table 1, the location number of the component AG is 029 and component SE is 055, etc. To be a valid component location, the last two digits of this number must be a number between 1 and 80. The unit column of this number refers to a column on the schematic, while the tens column refers to a row. The hundreds column is used to specify additional pages,
if needed, for the schematic. Thus the numbers which would be valid location numbers for components on the first page, PAGE 0, of a system schematic are:

001, 013, 051, 080

These same locations on the second page of the schematic, PAGE 1, would be:

101, 113, 151, 180

The location number phrase is followed by the name of the component at that location. A LOCATION command must be given only once for each component. This means that once a LOCATION statement is started for a component, the complete description of that component must be given.

Certain components have variable length vectors associated with them. The number of elements in these vectors can be specified by providing a component dimension statement, \( N= \) or \( M= \). Examples of this are:

\[
\text{LOCATION}=002 \quad \text{LG} \quad 1 \quad N=3 \quad \text{INPUTS}=....
\]

\[
\text{LOCATION}=524 \quad \text{SM} \quad N=12 \quad \text{INPUTS}=....
\]

\[
\text{LOCATION}=913 \quad \text{IM} \quad N=3, M=4 \quad \text{INPUTS}=....
\]

The \( N \) or \( M \) command must be the next command following the component name in the location statement. The phrase following the \( N \) or \( M \) command must be a number which specifies the dimensions of the arrays used by the component. The \( N \) or \( M \) commands can be applied to only those standard components which are designated to be capable of vector or matrix use as shown in the standard components lists contained in Appendix K. (None of the EASIEST components described in Appendix K require this command.)

Two characters immediately following the component name are used to designate multiple occurrences of the same type of component within the model description. Thus the following are all valid component identifiers:
LG 1  LG15  LGIN  LG2

This implies four occurrences of the component LG.

Component arrays can also be identified in the same fashion.

LG1,N=3  LG15,N=4  LG2,N=5  LG,N=3

The above example identifies different distinctive lag filters with dimensions of 3, 4, 5, and 3 respectively. In each of the above examples, the Model Generation Program will use the blank space as a character in identifying the components. Thus LG 1 and LG1 are different components.

If a portion of a system cannot be conveniently modeled using standard components, a block of FORTRAN statements may be used. The location of the FORTRAN block in the system schematic diagram is specified by using the component name FORT. The use of this technique is described in the FORTRAN STATEMENTS section.

INPUTS

The INPUTS command indicates that the comma separated phrases following this command contain the names of the components that provide the necessary inputs to the component at that location.

In order to better understand the ways to connect component inputs and outputs, a description of these characteristics is needed. Figure 3 shows the three typical types of components and their connections. The first example in this figure shows an input/output configuration that has one input and one output, both designated by the letter S. Part 1 specifies the input, while part 2 the output. This type of component usually performs a mathematical operation. A second type of input/output configuration is also used for components that model specific physical items. For these components, the labels represent quantities that have a definitive meaning. Component TD in Figure 3 is an example of this. The input
Figure 3. Typical Component Connections
quantity, T, is a vector which represents the torque applied to the vehicle. The output quantities are the vectors WD, angular acceleration, W, angular rate and EA, Euler angle. A third type of component has multiple inputs and/or outputs designated by S with a port associated with it. Component SW in Figure 3 is an example of this type. Extra care must be used defining the inputs and output connections to this device to assure proper signal hookup.

Between the components, three different levels of connection specification can be used in a model description:

1. **Default (only component names are specified)**
   Connections are made between all unconnected inputs and outputs to the first ports where a match of physical quantity names occurs. (Non-port inputs and outputs are also connected if a name match occurs.) For example:
   
   LOCATION = 045       SE       INPUTS = RL

2. **Ports Specified**
   Connections are made between matching physical quantities for all unconnected inputs and outputs of the specified ports. (Non-port inputs and outputs are also connected if a name match occurs.) For example:
   
   LOCATION = 045       SE       INPUTS = RL (1=1)

3. **Physical Quantities Specified**
   Connections are made between only those quantities specified. Previous connections cannot be over-ridden. For example:
   
   LOCATION = 045       RS       INPUTS = SE (SRP=XPB)

For many components, the input and output are single quantities and their connections can be made through the program default option without specifying the variable names. Thus, in the following example, component LG 1 at location 002 receives inputs from component MC 1:
In this example, the command phrase INPUTS is followed by a component name MC 1. The output name of MC and the input to the LG component have the same name, i.e., "S". Under this condition, no instruction other than specifying the input component is required.

For some components, there are multiple input and/or output "ports", which require the use of port numbers (S1, S2, S3, S4 etc.). The designation of these port numbers are defined in the standard components input/output lists. For multiple input ports, the port number must be specified as part of the INPUTS statement as shown in the following example:

LOCATION=110  MC 1  INPUTS=IT 1(S=S,1), TF 1(S=S,3)

It must be noted that the output quantity comes first, followed by the INPUT quantity name and port designation.

Port numbers refer to different physical connection points on a standard component. Once a connection is made between a port, such as port 2, of one component to another port, such as port 1, of a second component, inputs and outputs for ports other than 1 and 2 will not be connected even though they may have matching physical quantity names.

Some standard components can be used with variable dimensions. This feature allows the array length of a standard component with this capability to be specified. Thus, in the following example, the multiply and add component MC 1 and integrator components IT 1 and IT 2 are each defined to have three vectors as their inputs and outputs. The INPUTS function connects the three integrator outputs (IT 1) to the port 1 inputs and the integrator outputs (IT 2) to the port 3 inputs as shown in the following example:

LOCATION=052  MC 1  N=3  INPUTS=IT 1(S=S,1),IT 2(S=S,3)
If the input ports are not specified, the program default option will make the port selection in the order that they appear in the standard components list description. Thus, the following coding example would have accomplished the same objective.

LOCATION=032 IT 1 N=3 INPUTS=MC 1(S=S)

LOCATION=072 IT 2 N=3 INPUTS=MC 1(S=S)

For certain components, such as control elements, the inputs to the component can be any physical quantity in the model. For these components, the input component names must be supplemented by the name of the particular output quantity that is to provide the input. As an example, consider a component that represents a linear first order lag transfer function. If the transfer function component’s output, S, is to be the input torque, T, of the seat equations of motion, then the following statement would indicate to the program that, of the outputs of LG 1, S was to be used as the input, T, to the Seat Equations of Motion, SE 1:

LOCATION=005 SE 1 INPUTS=LG 1(S=T)

Input/output quantities may be either scalar, vector, or two dimensional arrays. Connections between array quantities are checked for compatible dimensions by the EASY5 Model Generation Program precompiler. An element of an output array can be used to drive a scalar input. Such a connection can be specified as:

LOCATION=043 LA INPUTS=MM(A(2,3)=S)
Here A is a two dimensional array output by a component MM. Element 2, 3 of this array will drive input S of component LA. Numeric values following an output quantity array name are assumed to be element designations if enclosed in parenthesis. If any other delimiter is used, they are assumed to be port designations.

Inputs to standard components from FORTRAN blocks are provided by using the name FORT for the component name in the input expression, i.e.:

\[
\text{LOCATION}=024 \text{ LA } \text{ INPUTS}=\text{FORT(COMP2(2)=S)}
\]

The FORTRAN component subscripted output quantity COMP2(2) will be connected to the input, S, of the standard component, LA. A discussion of using FORTRAN components is provided in the FORTRAN section. If a standard component is driven by both standard components and FORTRAN blocks, the standard component inputs must be specified before the FORT inputs.

Inputs to FORTRAN blocks may be either the outputs of standard components or the outputs of other FORTRAN blocks. Since the FORTRAN blocks do not have predefined input quantity names, the format used for specifying their inputs is different than that used for standard components. The complete name of the output quantities providing the inputs are required. The output names must contain enough information to uniquely define the source of the input. Thus, the complete output name of any standard component output must be given, i.e.:

\[
\text{LOCATION}=63 \text{ FORT } \text{ INPUTS}=S2 \text{ LA, PITCH, ROLL}
\]

Here the quantity S2 LA is the output of the standard component LA. PITCH and ROLL are the outputs of some other FORTRAN block. The above INPUTS statement refers to the output of the scalar LA component as S2 LA, not S,2 LA. The output quantity names must always be defined this way for use in FORTRAN component inputs since the EASY precompiler would interpret S, 2 LA as two separate input names.
FORTRAN STATEMENTS

The FORTRAN STATEMENTS program command allows the system analyst to supplement the standard EASY5 components with FORTRAN statements. Using this feature, the analyst can introduce his own program logic, DO loops, etc., as necessary to model any system not conveniently described with standard EASY5 components. Using this feature of the program, the analyst must perform many of the detailed connections and naming of variables that are normally accomplished by the EASY5 program. In return for these added tasks, the analyst gains a great deal of freedom and flexibility in forming details of his system model. To add a block of FORTRAN statements to the model, have it drawn on the schematic and included in implicit equation checking, the following convention must be used:

- A LOCATION statement with the component name FORT is placed before the FORTRAN STATEMENTS command. Input variables are specified by giving their names following the INPUTS command as described previously. These names may be either standard component output names or the outputs of other FORTRAN components, but must conform to the convention defined above.

- Outputs are specified by placing the ADD VARIABLES command following the INPUTS command. These quantities, either scalar or matrix or a combination, will be added to the model and assigned as outputs from the specific FORTRAN component. These output names may have up to seven characters.

- Parameter values, either scalar or matrix, are specified by the ADD PARAMETERS or ADD TABLES commands. These commands are added after the ADD VARIABLES command. These quantities will be added to the model and their values will be set in the Analysis Program. Parameter and table names may also have up to seven characters.

Thus the form for each FORTRAN component is:
LOCATION=063  FORT  INPUTS=S2 LA,ALPHA

ADD VARIABLES=BETA,GAMMA(3,3)

ADD PARAMETERS=COEFFS(3,2),GAIN

ADD TABLES=AEROTAB(250),3,AIRDATA(500),1

FORTRAN STATEMENTS

The lines before the FORTRAN STATEMENTS command (except ADD PARAMETERS and ADD TABLES) are required to specify the schematic location and the inputs and outputs to the block. If all of these are omitted, the FORTRAN statements will not appear in the schematic and will not be included in the implicit equation checking, which is described later under END OF MODEL. Only those quantities designated by ADD VARIABLES can be visibly connected to other standard components or FORTRAN blocks. The ADD commands are discussed next and details of the model schematic drawing appear in Section II.3. The ADD commands are used instead of dimension statements for the terms too be used in the FORTRAN statements. The FORTRAN statements can then include any FORTRAN IV required to describe the item being modeled. To simplify a number of these statements, a matrix arithmetic language has been developed which can be used within the FORTRAN statements to simplify the model description. A complete description of the matrix macro language is contained in Section IV.

ADD VARIABLES

ADD PARAMETERS

ADD TABLES

The ADD commands are used in conjunction with the FORTRAN STATEMENTS command to add variables, parameters, and tables that occur within the user supplied FORTRAN statements, to the EASY5 generated system model.
Quantities that are not specified by one of these commands cannot be accessed or manipulated by the EASY5 Analysis Program. See the examples in the FORTRAN section above for the proper order and use of the LOCATION, INPUTS, ADD, and FORTRAN STATEMENTS commands. Before discussing these commands, a few definitions of the terms are in order.

Variables: Variables are all dynamic time varying scalar or matrix quantities in the system model that are not states. In general, variables are related to states by fixed algebraic relationships.

Parameters: Parameters are constant scalar or matrix quantities in the system model. Parameters can be manipulated by the analyst to alter the system model. Default values are provided for certain parameters. The parameter values are set during the analysis option of the program.

Tables: Tables are constant nonscalar quantities in the system model. Tables are used to represent algebraic functional relationships with one, two or three independent variables. All table values are input as part of the analysis option of the program.

The format for the ADD commands is that the command is followed by one or more phrases that contain the names of the variables, parameters, or tables. These names must be unique. All parameter, and variable quantities are typed as Real quantities even if their name starts with the FORTRAN integer letters I, J, K, L, M, or N. Names added to the model via the ADD commands can consist of any valid FORTRAN name of up to seven characters. These names must not duplicate any name generated by the precompiler or another add statement. Variables or parameters may be scalar, vector, or two dimension arrays. The integrator components, IT or IN, should be used to define the state variables for the new component applications if additional states are required. The integrator components are straightforward in their use for adding new differential equations to be solved.
Matrix parameters are added to the model by placing dimension information, enclosed in parenthesis, after the parameter name, e.g.,

```
ADD PARAMETERS=ARRAY(3,6) COEF(6) . . 
```

Note: The ( and ) delimiters must be used to enclose dimension information. Dimensions must be between 1 and 99.

Matrix outputs are created by placing dimension information, enclosed in parenthesis, after the quantity names, e.g.,

```
ADD VARIABLES=VAR(3,2)
```

In addition to each table name, two numbers which specify the amount of storage to be allocated for the table and the number of independent variables must follow the table name. Thus to add three tables to a model, the instruction would be:

```
ADD TABLES=AEROTAB(120)2, TARGET(260)3, NOISE(500)1
```

This would add the two dimensional table AEROTAB with 120 words of storage; the three dimensional table TARGET with 260 words of storage; and the one dimensional table NOISE with 500 words of storage. The amount of storage is given by the formula:

\[ N = I + J + K + D \]

where 
- \( N \) = the total storage required by the table, in words. 
- \( I \) = the number of data points in the first independent variable table. 
- \( J \) = the number of data points in the second independent variable table. (\( J = 0 \) if there is only one independent variable.) 
- \( K \) = the number of data points in the third independent variable table (\( K = 0 \) if there are only one or two independent variables.)
D= the number of data points in the dependent variable table.
D=I if there is only one independent variable.
D=I*J if there are two independent variables.
D=I*J*K if there are three independent variables.

TABLE DIMENSIONS

The TABLE DIMENSIONS command can be used to specify Standard Component table dimensions. This is used when the default value for a Standard Component's table; as specified in the input/output lists, is too large or too small. This may be used as shown in the following example.

LOCATION=27  FV  INPUTS=LA1, LA2  
TABLE DIMENSIONS=FTAFV=500

The TABLE DIMENSIONS command in this example would increase the data storage for table FTA of the component FV from the default value of 171 to 500 words.

O.C. INPUTS

O.C. OUTPUTS

The O.C. INPUTS, O.C. OUTPUTS, and other commands starting with the letters "O.C." are used to include an optimal controller in the system model. A complete description of the calculation methods and theoretical basis for the optimal controller are presented in Appendix N. An optimal controller is a general purpose control component which can have an arbitrary number of inputs and outputs. It is, therefore, necessary for the system analyst to specify the identity of each optimal controller input and output. This is done using the O.C. INPUTS and O.C. OUTPUTS commands rather than the INPUTS command that is used for the other components. Optimal controller inputs are output quantities, either variables or states, from components which are used to sense the response of the system being controlled. Optimal controller outputs are input quantities, either variables or
parameters, to components that serve as the actuators to the system being controlled.

**O.C. CRITERIA**

The O.C. CRITERIA command is used to specify those output quantities from the components that are to be used as the criteria for designing the optimal controller. These quantities are specified in the same format as O.C. INPUTS. If no O.C. CRITERIA are specified, the O.C. INPUTS are used as the design criteria. A complete discussion of the use of O.C. CRITERIA is given in Appendix M.

**O.C. ORDER**

The O.C. ORDER command can be used to specify the order of the optimal controller. If the optimal controller order is not specified, it will be taken as the order of the system model. This will result in a total system order, (optimal controller plus system model), that is twice the order of the system model. In most cases, such a high order optimal controller is unnecessarily complex and impractical. The O.C. ORDER is limited to values between zero and the system model order.

**O.C. MODEL ORDER**

The O.C. MODEL ORDER command can be used to specify that a model order lower than that of the given system model, be used for the optimal controller design. This command is used when optimal controllers are to be designed for high order systems. By using a lower order model, the computer memory requirements and computation time can be greatly reduced. A complete discussion of the use of reduced model orders is given in Section 4.4 of reference 1. This section is reproduced in Appendix N.

**O.C. ANALYSIS**

The O.C. ANALYSIS command is used to specify that computer memory requirements provided in the system need only be large enough for the analysis of
an optimal controller. The memory required to analyze a system with an optimal controller is considerably less than that required to do an optimal controller design. Thus, if the purpose of a run is to analyze the performance of an optimal controller which was designed on a previous run, the O.C. ANALYSIS command can be used to reduce computing costs and flow time.

END OF MODEL

The END OF MODEL command phrase indicates that model description has been completed and that the Model Generation Program should proceed with the generation of the model subroutines. As part of the subroutine generation, the model components are checked for implicit relationships. An implicit relationship occurs when a variable is used as an input to a component before it has been calculated. This can occur if a variable is used as an input to a component that precedes the component that generates the variable. Implicit relations such as this can often be resolved by reordering the sequence of the components in the model. If such reordering occurs, a warning message is printed identifying the components affected. It is possible to create models in which the implicit relationships cannot be resolved by such a reordering. In this case, a warning message will be printed stating that analysis results will be invalid. The implicit relationship must then be resolved by changing this model. Changes such as placing an additional state in the implicit loop or solving the implicit relationship algebraically can be used.

PRINT

The PRINT command phrase causes the program to: (1) draw a schematic of the system model, as shown in Figure 2, (2) print a list of input requirements for the model; and (3) print a source listing of the FORTRAN subroutines that were generated for the model. The Model Generation Program then terminates.
LIST STANDARD COMPONENTS

The LIST STANDARD COMPONENTS command phrase causes the program to print a list of all standard components. For each standard component, lists of inputs, outputs, and tables for that component are provided. For each input, the physical quantity name and port number is given. For each output, the physical quantity name, port number, and the word STATE is given, if the quantity is a state. For each table, the table name, the number of independent variables and the default value for data storage is provided. This command is usually given as the first command of a model description and will result in a list of all standard component information as the first output from the Model Generation Program.

PRINT STATEMENTS

The simulation operation of the EASY5 Analysis Program has several print output options. Most of these, as described in Section III, consist of fixed formats such as: all states, all variables, or a user furnished list of variables. An additional option is to execute a set of user furnished print statements. These print statements are specified as part of the model description via the PRINT STATEMENTS command. The PRINT STATEMENTS command must be followed by valid FORTRAN statements. These statements will be executed only when the Analysis program PRINT CONTROL = 8 is specified along with the desired print output periods. In general, only FORTRAN PRINT, WRITE, and FORMAT statements would be included as PRINT STATEMENTS. However, other valid FORTRAN statements can be included if additional calculations or control logic is desired. Any state, rate, variable, or parameter in the model is available for use in the PRINT STATEMENTS. The PRINT STATEMENTS command can appear only once in a model, anywhere between the MODEL DESCRIPTION and END OF MODEL commands. An example of the PRINT STATEMENT command is given below:

PRINT STATEMENTS
WRITE (6,111) AMISS, XLOC, YLOC, TIME

The DEBUG command may be used to place print statements between each Standard Component in the model. These print statements will be executed only when the PRINT command is given to the Analysis Program. The printout that occurs will be that specified by the PRINT CONTROL command. This command is very helpful in locating the cause of arithmetic errors in a model. This command should be placed before the END OF MODEL command. It should be removed from the model description once the model is free of arithmetic errors.

ALPHABETICAL LIST OF COMMANDS

ADD PARAMETERS
ADD TABLES
ADD VARIABLES
DEBUG

END OF MODEL
FORTRAN STATEMENTS
INPUTS

LIST STANDARD COMPONENTS
LOCATION
MODEL DESCRIPTION

O.C. ANALYSIS
O.C. CRITERIA
O.C. INPUTS
O.C. MODEL ORDER
O.C. ORDER
O.C. OUTPUTS

PRINT
PRINT STATEMENTS
TABLE DIMENSION
3. MODEL SCHEMATIC

The Model Generation Program produces a schematic diagram of the system being modeled. This schematic is generated on the line printer with the computer printout. Its purpose is to provide a means of rapidly locating errors in the model description.

In order to construct a schematic diagram in an efficient manner with a reasonable size program, it was necessary to establish some simple rules for symbol generation, component connection paths, and labeling. If these rules are kept in mind when laying out a schematic for the system, the EASY5 produced schematic will match that developed by the analyst. If the rules are violated by the analyst's schematic, the EASY5 schematic will still be correct but may contain some unusual component connection paths, and some labeling information may be overwritten.

a. Standard Schematic Form

The EASY5 schematic diagrams are produced on a standard 11" by 14" lineprinter page with 80 component locations per page. A standard form containing only the location numbers can be obtained by executing the EASY5 Model Generation Program with the single program command, PRINT. This form can then be reproduced and the copies used as forms for drawing system model schematics.

b. Input Quantity Labeling

The names of the physical quantities that are input to one component from another component are listed adjacent to the downstream component symbol. The physical quantity name, i.e., first three characters of the quantity being driven, is also given. These labels are placed near the connecting line that joins the two components. Since these names are composed of the physical quantity name and the name of the component that generates the information, the source of the input is evident from the name itself. Parameter and tabular inputs to a component are not shown on the schematic.
c. Component Connection Paths

In order to simplify the EASY5 schematic drawing subroutine, it was necessary to limit the types of connecting paths between components to a few basic routes. These paths are shown in Figure 4. Connections between components on the same horizontal or vertical line are straightforward. However, connections between components that do not share a horizontal or vertical line require at least a two segment path. These paths have been arbitrarily chosen to follow a clockwise route. It is, therefore, advisable that components that are on diagonal locations be placed in a clockwise sequence. If counterclockwise flow between components is necessary, it can be accommodated by placing the components on the same horizontal or vertical lines. The EASY5 schematic drawing subroutine does not go around components that are on a connection path. Such components are "run-over" by the connecting line.

d. Additional Pages

The EASY5 schematic diagram may be broken down into as many pages as are necessary. No attempt is made to draw connecting paths between components located on different pages. It is, therefore, advisable to minimize the number of connecting paths between pages. This can usually be done by grouping components with many interconnections on the same page and placing page boundaries between such groups of components.

e. Guidelines For Schematic Layout

The following guidelines will help in creating schematic layouts that can be easily produced by the Model Generation Program.

- Try to place connected components on the same horizontal or vertical line.
- Avoid placing components on adjacent location points.
- Place diagonal components so that flow is clockwise.
- Group components to minimize flow paths between pages.
Figure 4. Component Connection Paths
4. WARNING MESSAGES

One or more of the following warning messages will occur if the program is unable to interpret a portion of the model description or encounters problems in assembling the system model. These messages will be preceded by: ***WARNING*** or ***NOTICE***. The symbols xxx and zzz are used to indicate phrases from the model description that are included as part of the warning message. The following messages are listed in alphabetical order:

1. **ADD COMMAND MUST FOLLOW A "LOCATION=N FORT" COMMAND**

The ADD VARIABLES command must follow a FORTRAN component location command.

2. **CAN'T IDENTIFY SOURCE OF xxx INPUT TO LOCATION U**

Cannot locate the source of xxx which is an input to component at location U.

3. **CAN'T IDENTIFY xxx AS A STANDARD COMPONENT.**

xxx will contain the first two characters of the phrase which cannot be identified as a command or standard component. This message will often follow other warning messages as the program makes successive attempts to interpret the given phrase.

4. **CAN'T IDENTIFY xxx AS A VALID INPUT TO zzz**

The input quantity xxx for component zzz cannot be identified.

5. **CAN'T IDENTIFY xxx AS A VALID OUTPUT FROM zzz**

The quantity xxx cannot be identified as an output from zzz.
6. CAN'T LOCATE FORTRAN COMPONENT xxx

Cannot locate FORTRAN component xxx statements.

7. CAN'T LOCATE O.C. INPUT, xxx, WILL RENAME AS: zzz

Check spelling of name xxx or that the quantity xxx has been renamed as a result of being driven by another component.

8. CAN'T LOCATE O.C. OUTPUT, xxx

Check spelling of name xxx.

9. COMPONENT xxx DEFINITION WASN'T COMPLETED BEFORE STARTING THE DEFINITION OF COMPONENT zzz

The command INPUTS was not given between the component names xxx and zzz. Check for proper spelling of INPUTS and a valid delimiter after the phrase xxx.

10. COMPONENT xxx HAS ALREADY BEEN DEFINED

11. CROSS PRODUCT IS ONLY DEFINED FOR 3 VECTORS

12. DIMENSIONS HAVE NOT BEEN GIVEN FOR xxx

Dimensions of input matrices must be defined before being used in a matrix expression.

13. DIMENSIONS OF xxx AND zzz ARE INCOMPATIBLE

Dimensions of input matrices in matrix expressions are incompatible.

14. DIMENSIONS OF xxx DO NOT MATCH THOSE OF zzz

Dimension mismatch occurred during interconnection of matrices.
15. LOCATION NO. xxx FOR COMPONENT zzz HAS LAST TWO DIGITS OUTSIDE THE ALLOWABLE RANGE 1 TO 80. NO SYMBOL WILL BE PLACED IN SCHEMATIC FOR THIS COMPONENT

This message will occur at the end of the model description for a component zzz which has an invalid location number. The system model may still be valid, but the schematic will not contain this component.

16. MATRIX xxx IS BEING DRIVEN BY A SCALAR QUANTITY zzz

This is likely to produce erroneous results.

17. MODES CANNOT BE SPECIFIED FOR COMPONENT xxx

The dimensions statements N=, M= can only be used on designated components.

18. NO OPTIMAL CONTROL INPUTS WERE SPECIFIED

Check that "O.C. INPUTS" command was used to specify optimal inputs.

19. NO OPTIMAL CONTROL OUTPUTS WERE SPECIFIED

Check that "O.C. OUTPUTS" command was used to specify optimal controller outputs.

20. NO xxx OUTPUTS MATCH UNSATISFIED zzz INPUTS

Check that it was intended to drive component zzz with component xxx or that the inputs to zzz have been previously satisfied by other component connections.

21. O.C. MODEL ORDER CANNOT BE SPECIFIED GREATER THAN MODEL ORDER

O.C. model order will be set to n.
22. O.C. ORDER CANNOT BE SPECIFIED GREATER THAN MODEL ORDER

O.C. order will be set to n.

23. ONLY 63 INPUTS + OUTPUTS ARE ALLOWED

Each component is limited to 63 inputs + outputs.

24. ONLY 100 VARIABLE DIMENSION COMPONENTS ARE ALLOWED

Only 100 variable dimension components are allowed in a given model.

25. SCALAR QUANTITY xxx IS BEING DRIVEN BY MATRIX zzz

The first element of matrix will be used to drive the scalar.

26. SYNTAX ERROR

Syntax error occurred in matrix expression.

27. TABLE NAME xxx MUST BE FOLLOWED BY A NUMERIC DIMENSION RATHER THAN zzz

When using the ADD TABLES command, it is necessary to provide the maximum amount of storage to be allocated for the table as well as the table name. This storage value must be a numeric quantity.

28. THE FOLLOWING COMPONENTS FORM AN IMPLICIT LOOP. MODEL RESULTS WILL BE INVALID. xxx, zzz, 

Models must be explicit. Implicit loops can often be corrected by inserting a component with a state variable as its output, e.g., a simple linear lag, LA.

29. THE NUMBER OF O.C. INPUTS, OUTPUTS, OR CRITERIA VARIABLES MUST BE 63 OR LESS XXX WILL NOT BE LOADED
30. THE SEQUENCE OF THE FOLLOWING COMPONENTS HAS BEEN ALTERED TO FORM AN
EXPLICIT MODEL. xxx, zzz, ....

The model component sequence as given contained an implicit relationship.
By altering the component sequence, it was possible to form an explicit
model.

31. xxx IS NOT A VALID DIMENSION

The phrase xxx should be numeric to be a dimension phrase.

32. xxx IS NOT A VALID INPUT QUANTITY OR PORT DESIGNATION FOR COMPONENT
zzz

The phrase xxx cannot be located as one of the input quantities or input
ports of the component zzz. No connections will occur. Check the list of
standard components for the proper spelling or port designations for this
component.

33. xxx IS NOT A VALID LOCATION NUMBER

The LOCATION command must be followed by a numeric location number.

34. xxx IS NOT A VALID PORT DESIGNATION FOR INPUT COMPONENT zzz. ERRO-
NEOUS CONNECTIONS MAY OCCUR.

The phrase xxx cannot be located as a valid input port for the component
zzz. Connections will be attempted using the upstream output port that was
identified.

35. xxx IS NOT A VALID SUBSCRIPT

Subscripts must be numeric. The use of parenthesis as delimiter after
array name implies a subscript is given.
36. xxx IS NOT A VALID SUBSCRIPT FOR FORTRAN OUTPUT zzz

The quantity xxx is not a valid subscript for FORTRAN output quantity zzz.

37. xxx IS NOT AVAILABLE AS INPUT

Cannot locate xxx as FORTRAN input to standard component.

38. xxx ISN'T NUMERIC O.C. ORDER MUST BE NUMERIC QUANTITY.

39. xxx MUST BE A SQUARE MATRIX

Simultaneous equation solution is valid only for square coefficient matrix.
SECTION III
DYNAMIC ANALYSIS OF CONTINUOUS OR DISCRETE SYSTEMS

The EASY5 Analysis Program allows several different dynamic, static, linear, or nonlinear analysis techniques to be used on the dynamic system model generated by the Model Generation Program. In addition to normal analysis techniques, optimal linear controllers based on Kalman optimal linear regulator and Kalman filter theory can be synthesized by the program. The performance of such optimal controllers when operating with the nonlinear system can be analyzed using any of the analysis techniques.

Both continuous systems, i.e., those described by ordinary nonlinear differential equations, and discrete systems, i.e., those described by differential and discrete difference equations, can be modeled and analyzed by the EASY5 program. The analysis techniques automatically switch to discrete methods* if one of the discrete components, DE, DF, DL, DT, DZ, or SH is included in the system model. All data input, output, and analysis commands are the same for both continuous and discrete systems. The only restriction for discrete systems is that the total number of sampling periods is restricted to 10.** This refers to the sampling period parameters, TAU, for each discrete component. The name of these parameters must always start with the letters TAU, and no other parameter may start with the letters TAU.

A description of the control of the program and of the analytical methods is given in Sections III.1 through III.16. An alphabetical listing of the analysis program commands is given in Appendix B of this document. Check lists for each analysis are given in Appendix C. For a description of continuous system techniques and numerical methods, see reference 1, Section 4. For discrete methods, see Section IX.

---

*The Root Locus, stability margin, eigenvalue sensitivity, and optimal controller design options are not available for discrete systems.

**Sample periods must be integer multiples of one another.
1. MODEL INPUT DATA

A dynamic system model requires that the values of numerous model parameters, tables and initial conditions, be provided to complete the model description. Sections III.1, III.2, and III.3 describe the methods used to specify parameter values, tables, and matrices.

a. Scaler Data

PARAMETER VALUES

This program command allows the numeric values of parameters to be loaded into the system model. The PARAMETER VALUES command is followed by one or more parameter names followed by a numeric value of ten characters or less. Each name and its value are separated by commas or another one of the standard delimiter symbols. This command is used to specify the values of all system model parameters at the beginning of an analysis. It may also be used at any point between analyses to modify the value of one or more model parameters. A default value of .99999 is provided for all parameters not specified.

PARAMETER VALUES = MASS = 10., AREA = 50., SW AG = 1,
CCGSE=.48,0,-.75, CW SE=210, STIPC=10.57,....

b. Tabular Data

TABLE

If tabular data is required by the system model, it should be loaded with the other parameter values before any of the analysis commands described in Sections III.4 to III.13 are issued. Tables may be modified between analyses by loading new values. The tables required by an EASY5 generated model are specified in the Model Generation Program Input Requirements List. These tables may have either one, two, or three independent variables. All data items are in a free field format with each item having
10 characters or less separated by commas or other standard delimiter. The data items required for each table are placed in the following format:

Line 1 TABLE Table name NX NY NZ
Line 2* Z table values
Line 3* Y table values
Line 4* X table values
Line 5* D table values

For this input, the following definitions apply:

Table Name - The seven character table name generated by the EASY Model Generation Program.
NX - The number of points in the first independent variable table.
NY** - The number of points in the second independent variable table.
NZ*** - The number of points in the third independent variable table.
Z table*** - Table of NZ third independent variable values.
Y table** - Table of NY second independent variable values.
X table - Table of NX first independent table values.
D table - Tables of dependent variable values.

*As many lines or cards as required may be used. Each table must start with a new line or card and NZ, NY, NX, and NX*NY*NZ points must be given per table.
**These items are omitted for tables with one independent variable.
***These items are omitted for tables with one or two independent variables.
A copy of all tabular input data is printed as it is interpreted from the data, unless the OMIT TABLE PRINTOUT command has been given. The following example shows the data for a one and a two independent variable table.

<table>
<thead>
<tr>
<th>Line</th>
<th>TABLE, TAB-ONE, 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line 2</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10</td>
</tr>
<tr>
<td>Line 3</td>
<td>11, 12, 13, 14, 15, 16, 17, 18, 19, 110</td>
</tr>
<tr>
<td>Line 4</td>
<td>TABLE, TAB-TWO, 5, 4</td>
</tr>
<tr>
<td>Line 5</td>
<td>10.3, 20.4, 30.5, 40.6</td>
</tr>
<tr>
<td>Line 6</td>
<td>1, 2, 3, 4, 5</td>
</tr>
<tr>
<td>Line 7</td>
<td>11, 12, 13, 14, 15</td>
</tr>
<tr>
<td>Line 8</td>
<td>21, 22, 23, 24, 25</td>
</tr>
<tr>
<td>Line 9</td>
<td>31, 32, 33, 34, 35</td>
</tr>
<tr>
<td>Line 10</td>
<td>41, 42, 43, 44, 45</td>
</tr>
</tbody>
</table>

The printout of these tables would be:

**TABLE TAB-ONE**

FIRST INDEPENDENT VARIABLE TABLE

<table>
<thead>
<tr>
<th>1.000</th>
<th>2.000</th>
<th>3.000</th>
<th>4.000</th>
<th>5.000</th>
<th>6.000</th>
<th>7.000</th>
<th>8.000</th>
<th>9.000</th>
<th>10.00</th>
</tr>
</thead>
</table>

DEPENDENT VARIABLE TABLE

<table>
<thead>
<tr>
<th>11.00</th>
<th>12.00</th>
<th>13.00</th>
<th>14.00</th>
<th>15.00</th>
<th>16.00</th>
<th>17.00</th>
<th>18.00</th>
<th>19.00</th>
<th>110.00</th>
</tr>
</thead>
</table>

**TABLE TAB-TWO**

SECOND INDEPENDENT VARIABLE TABLE

<table>
<thead>
<tr>
<th>10.30</th>
<th>20.40</th>
<th>30.50</th>
<th>40.60</th>
</tr>
</thead>
</table>

FIRST INDEPENDENT VARIABLE TABLE

<table>
<thead>
<tr>
<th>1.000</th>
<th>2.000</th>
<th>3.000</th>
<th>4.000</th>
<th>5.000</th>
</tr>
</thead>
</table>
DEPENDING VARIABLE TABLE

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11.00</td>
<td>12.00</td>
<td>13.00</td>
<td>14.00</td>
<td>15.00</td>
</tr>
<tr>
<td>21.00</td>
<td>22.00</td>
<td>23.00</td>
<td>24.00</td>
<td>25.00</td>
</tr>
<tr>
<td>31.00</td>
<td>32.00</td>
<td>33.00</td>
<td>34.00</td>
<td>35.00</td>
</tr>
<tr>
<td>41.00</td>
<td>42.00</td>
<td>43.00</td>
<td>44.00</td>
<td>45.00</td>
</tr>
</tbody>
</table>

THREE INDEPENDENT VARIABLE TABLE EXAMPLE

| Line 1 | TABLE=FTAFW | 3 | 2 | 2 |
| Line 2 | 1,2 |
| Line 3 | 3,4 |
| Line 4 | 5,6,7 |
| Line 5 | 111,112,113 |
| Line 6 | 121,122,123 |
| Line 7 | 211,212,213 |
| Line 8 | 221,222,223 |

The printout of this table would be:

==== TABLE FTAFW ====

THIRD INDEPENDENT VARIABLE TABLE

1,000 2,000

SECOND INDEPENDENT VARIABLE TABLE

3,000 4,000

FIRST INDEPENDENT VARIABLE TABLE

5,000 6,000 7,000

DEPENDING VARIABLE TABLE
THIRD INDEPENDENT VARIABLE = 1,000

111.0  112.0  113.0
121.0  122.0  123.0

THIRD INDEPENDENT VARIABLE = 2,000

211.0  212.0  213.0
221.0  222.0  223.0

OMIT TABLE PRINTOUT

The OMIT TABLE PRINTOUT command may be used to suppress the printback of table data. This command is often used on production runs or models with large amounts of constant tabular data. A second occurrence of this command causes table printback to be restored.

c. Matrix Data

Matrix Parameters can be one or two dimensional arrays. The matrix input format must contain the matrix name, the input method, and the appropriate matrix elements. If the input method is not specified, a default of input by columns is assumed. If the default mode is used, however, the user must be careful to:

o Input the exact number of elements defined by the dimensions in the Model Generation Program since the maximum dimensions are not checked by EASY5. With this method, the user must accept this responsibility.

o Not exceed ten characters per matrix element.
If the default option is not used, parameter arrays can be loaded by any of the following conventions after inserting the PARAMETER VALUES command:

**COLUMN INPUT**
- `ADATA, C (1, 1) 1, 2, 3, 4, 5` Starts at element 1, 1
- `ADATA, C (1, 2) 6, 7, 8, 9, 10` Starts at element 1, 2

**ROW INPUT**
- `BDATA, R (2, 3) 7, 8, 9, 10` Starts at element 2, 3
- `BDATA, R (1, 2) 3, 6, 9, 10` Starts at element 1, 2

**DIAGONAL INPUT**
- `COEF, D (2, 4) .3, .4, .5` Starts at element 2, 4

**ZERO Array - then load by row**
- `COEF, Z, R (2, 2) 1, 2, 3`

Set array to infinite, "Infinite" = $10^{36}$
- `CUEF, I`

Input by Column starting at element 1, 1 (default option)
- `VECTOR = 1, 2, 3, 4, 5`

**ELEMENT Input**
- `ADATA (1, 2) = 12, (3, 4) = 16, (2, 3) = 21`

Note: "(" must be used as delimiter immediately following array name.

2. INITIAL CONDITION, ERROR, AND INTEGRATION CONTROLS

**INITIAL CONDITIONS**
**ERROR CONTROLS**
**INT CONTROLS**
These program commands may be used to specify integrator initial condition values, error controls, or status, whether active (= 1) or frozen (= 0). The default values that are provided are 0.0 for initial conditions, 0.001 for error controls, and 1 for integration controls. These are furnished by the EASY5 Analysis Program. However, it is strongly recommended that values appropriate to the particular system model be furnished for the initial conditions and error controls.

Each of these commands is followed by phrases of the form of a state name followed by a numeric value. State quantities that are vectors or matrices may be input by the same conventions as for parameters. The following shows an example of how these commands are used:

```
INITIAL CONDITIONS = VELOCITY = 50., ANGLE = 2., U SD = 512, 362, 0.
ERROR CONTROLS = VELOCITY = .1, ANGLE = .01, U SD(3) = .0001
INT CONTROLS = VELOCITY = 0, ANGLE = 1, STROKE = 1

ALL STATES
NO STATES
```

These program commands may be used to activate or freeze all system integrators. These commands are normally used together with the INT CONTROLS command to specify the desired integrator configuration.

```
INITIAL TIME = t
```

This program command allows the initial value of time to be specified. The default value of initial time is zero. The INITIAL TIME command is used with models that contain time dependent features where it may be desirable to have time at the beginning of a simulation run or during a steady state analysis be some value other than zero.

```
PRINT
```

45
This command, PRINT, causes the states to be set to the initial conditions, time to equal INITIAL TIME, and the model executed and printed output requested via the PRINT CONTROL command.

3. INITIAL CONDITION COMMANDS

XIC-X
XIC-XIC1
XIC-XIC2
XIC-XIC3
XIC1-XIC
XIC2-XIC
XIC3-XIC

These program commands are used to transfer data from the current state vector, X, to the initial condition vector, XIC, and between the XIC vector and three auxiliary initial condition vectors XIC1, XIC2, XIC3. The following shows how these commands would be used:

XIC1-XIC, XIC-X, XIC2=XIC

The three program commands shown above would take the current operating point (initial condition vector) and store it in vector XIC1; then transfer the current state, X, into XIC; and then store that value of XIC in XIC2.

CALC XIC

This command allows initial conditions to be calculated from manually input parameters or initial conditions. This command, CALC XIC, causes the state to be set to the values input manually for XIC; an integer flag in common block /CICCAL/ to be set to 1, and the model to be executed. Initial condition calculations can be placed in the model that will be executed only if the flag equals 1. Upon exiting from the model, the initial condition array XIC is set equal to the state array X and the print routine is called. The initial condition flag is reset to 0.
4. SIMULATION COMMANDS

SIMULATE

This program command initiates simulation operation. Before the simulate command is used, the following program values must be set:

- **TINC** = time increment, seconds
- **TMAX** = duration of the simulation run, seconds
- **INT MODE** = integrator mode control
- **OUTRATE** = output rate
- **PRATE** = print rate
- **PRINT CONTROL** = print control variable

These program commands specify the integration time increment, duration of simulation run, the integration type, the simulation output rate, the printing rate, and the quantity of printing, at each point in time. These quantities must be specified before the first use of the SIMULATE command.

For discrete systems, the time increment, TINC, should be an integer submultiple of the sample periods. Thus, if sample periods were .01 and .04, TINC should be selected such that: n*TINC=.01, where n is an integer. The EASY5 Analysis Program will check TINC and adjust it if necessary to satisfy this requirement. The output control OUTRATE will also be adjusted to maintain approximately the same data output rate.

The integration mode control, INT MODE, allows one of six different integration methods to be selected according to the description given in Table 3. The default value of INT MODE is 6. A description of these integration methods and a guide to their use is given in Section VIII.
TABLE 3
Integration Method Selection

<table>
<thead>
<tr>
<th>INT MODE</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Variable Step, Variable Order Gear</td>
</tr>
<tr>
<td>2</td>
<td>Variable Step 4th Order Runga-Kutta</td>
</tr>
<tr>
<td>3</td>
<td>Fixed Step Huen Method, 2nd Order</td>
</tr>
<tr>
<td>4</td>
<td>Fixed Step Euler, 1st Order</td>
</tr>
<tr>
<td>5</td>
<td>Adams-Bashforth predictor/Adams-Moulton Corrector, Orders 2-12.</td>
</tr>
<tr>
<td>6</td>
<td>Stiff Gear</td>
</tr>
</tbody>
</table>

The time increment, TINC, provides the integrator time step size, in seconds, for the fixed step integrators. TINC also provides the report interval for which data will be available for printing or plotting. The default value for TINC is 0.1.

The duration of a simulation calculation is specified by the TMAX parameter in seconds. The default value of TMAX is 1.

The output rate parameter, OUTRATE, determines the rate at which simulation data is added to plots. Thus, if OUTRATE is set equal to 10, data will be plotted every 10th time increment, TINC. This feature is normally used only when a fixed step size integrator is specified. With such an integrator, the time increment is usually quite small, and excessive plotted output would be generated if it were not for this sampling feature provided by the OUTRATE parameter. The default value of OUTRATE is 1. OUTRATE should only be set to positive integer values.

The number of data samples plotted for a simulation analysis is given by:

\[
\text{No. of Plotted Samples} = \frac{\text{TMAX}}{\text{TINC} \times \text{OUTRATE}} + 1
\]
For most simulation operations, the plot output is the primary data. The line printer output options provided by the PRINT CONTROL parameter allow a wide range in the amount of detailed information about the simulated system to be printed. The value of the PRINT CONTROL parameter sets the quality of data printed at each print report interval according to Table 4. Options 1 through 4 give "snap-shots" of all states, rates, variables, and parameters of the system model at a particular point in time. Option 5 provides tabular lists of up to 40 specified quantities. Options 6 and 7 are used with the steady state analysis options. Options 6 and 7 are used with the steady state analysis options. Option 8 uses the user provided print statements from the model description. The default value for PRINT CONTROL is 0.

<table>
<thead>
<tr>
<th>PRINT CONTROL</th>
<th>Resultant Lineprinter Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>All states, rates, and time</td>
</tr>
<tr>
<td>2</td>
<td>All states, rates, variables, and time</td>
</tr>
<tr>
<td>3</td>
<td>All states, rates, variables, and parameters at time = 0</td>
</tr>
<tr>
<td>4</td>
<td>All states, rates, variables, and parameters</td>
</tr>
<tr>
<td>5</td>
<td>Time and the quantities specified via PRINT VARIABLES command</td>
</tr>
<tr>
<td>6</td>
<td>All states, rates, variables, and parameters at each STEADY STATE iteration</td>
</tr>
<tr>
<td>7</td>
<td>All states, rates, variables, parameters, and system Jacobian matrix at each STEADY STATE iteration</td>
</tr>
<tr>
<td>8</td>
<td>User furnished PRINT STATEMENTS (See Model Generation Section II.2.b)</td>
</tr>
</tbody>
</table>

The PRATE parameter determines the sampling rate at which the simulation data specified by the PRINT CONTROL parameter is presented on the lineprinter. Thus, if PRATE is set equal to five, data will be printed on the
line printer every fifth time it is added to the output plots. The default value of PRATE is 1. PRATE should only be set to positive integer values.

The number of data samples printed for a simulation analysis is thus given by:

\[
\text{No. of Plotted Samples} = \frac{T_{\text{MAX}}}{T_{\text{INC}} \times \text{OUTRATE} \times \text{PRATE}} + 1
\]

An example of the use of these commands is shown below:

PRINT CONTROL = 3, TINC = .01, TMAX = 10.,
INT MODE = 3, OUTRATE = 10, PRATE = 10, SIMULATE

In the example, the fixed step Huen integration method would be used with a step size of .01 second. The simulation would run for 10 seconds. Plotted output would occur every .1 seconds, (10*.01), and printed output would occur every 1.0 seconds (10*10*.01).

TINC2
OUTRATE2
PRATE2
PRINT2
PRINT2 FROM, \(t_1\), TO, \(t_2\)

For some applications, a single set of output controls is not satisfactory. For example, it might be desirable to have a high sampling rate during an initial transient followed by a slower sampling rate, or to have a high sampling rate around a critical event. To satisfy this requirement, a second set of control values can be assigned to the program values TINC, OUTRATE, PRATE, and PRINT CONTROL. These are specified as:

TINC2, OUTRATE2, PRATE2, PRINT2
These values can be requested during a time interval via the command:

\[
\text{PRINT2 FROM, } t_1, \ TO, \ t_2
\]

Here \( t_1 \) is the time to start the second output option and \( t_2 \) is the time to revert to the original output option as given by: TINC, OUTRATE, PRATE, and PRINT CONTROL. An example of the analysis commands for this type of operation is:

\[
\begin{align*}
\text{PRINT CONTROL} & = 4, \ TINC = .01, \ TMAX = 10 \\
\text{OUTRATE} & = 10, \ PRATE = 10, \ OUTRATE2 = 1, \\
\text{PRINT2} & = 8, \ \text{PRINT2 FROM, } 8., \ TO, \ 9., \ \text{SIMULATE}
\end{align*}
\]

In the example, the simulation would run for 10 seconds with a step size of .01 seconds. The initial plotted output would be every 0.1, \((10 \times 0.01)\), seconds and printed output would occur every 1., \((10 \times 10 \times 0.01)\) second. Between 8 and 9 seconds, the plotted and printed output rates would be increased to every .01, \((1 \times 0.01)\), and 0.1, \((10 \times 1 \times 0.01)\) seconds and would consist of model furnished PRINT STATEMENTS (print option 8).

The second output options can also be activated by events occurring within the model. This can be done by setting a print flag variable, PFLAG, within the model EQMU subroutine to a non-zero value. As long as PFLAG has a non-zero value, the second output options will be in effect. When PFLAG is set to zero, the original output options are restored. PFLAG can be set by an IF test contained in a FORTRAN STATEMENT in the model. An example of this type operation is:

\[
\begin{align*}
\text{FORTRAN STATEMENT} \\
PFLAG & = 0 \\
\text{IF (RANGE .LT. 100.) PFLAG = 1}
\end{align*}
\]

51
In this example, if the variable range becomes less than 100, the second print option will occur.

PRINT VARIABLES

This program command allows up to 40 variables to be specified for printing using option 5 of the PRINT CONTROL. This command is followed by from one to 40 state, rate, variable scalar, or subscripted names separated by delimiters. This command deletes all previously stored PRINT VARIABLES names. A column format will be used if the number of quantities being printed is less than or equal to 10. If more than 10 quantities are specified, the name and value of each scalar or subscripted vector quantity will be printed in a format similar to that of print options 1, 2, or 3. An example of this use is:

PRINT VARIABLES = S1 DE1, S1 DE2, W1 DE2, S2 LA(3)

5. PLOT DESIGNATION COMMANDS

DISPLAY1
DISPLAY2
DISPLAY3
DISPLAY4
DISPLAY5
DISPLAY6

These program commands are used to define the quantities to be displayed by off-line plots or written on external tapes for simulation or steady state calculations. These commands must be issued before the simulation or steady state analysis is requested. From one to five plots may be specified per display. Each plot is specified by stating the dependent variable and the independent variable separated by the letters VS. If desired, the dependent and independent axis scale ranges can also be specified. These scales will be used if the MANUAL SCALES commands are given. The independent scale range is specified by the word XRANGE followed by the minimum
and maximum values for this scale. The dependent scale similarly is specified by the word YRANGE. If scale ranges are not specified, values will be used that span the given data. For more than one plot on a page, a common independent variable must be used.

The following example shows two ways to specify plots:

DISPLAY1
ANGLE, VS, TIME, YRANGE = -2,4
STROKE, VS, TIME, YRANGE = -.5,.5
P1 DE1, VS, TIME, YRANGE = 0,60

DISPLAY2
P1, CE, VS, TIME, YRANGE = -20,20
P1, DE2, VS, TIME, YRANGE = -15,15
PRESSURE, VS, TIME, YRANGE = -100,100
THECE, VS, TIME, YRANGE = -5,5

DISPLAY3
STROKE, VS, PRESSURE, YRANGE = -1.5, XRANGE = 300,500

SI MANUAL SCALES
SS MANUAL SCALES
SI AUTO SCALES
SS AUTO SCALES

The SI MANUAL SCALES and SS MANUAL SCALES commands allow the plotted output requested by the DISPLAY commands to be plotted on manual scales specified by the YRANGE and XRANGE commands. If manual scales are requested, manual scales must be given and will be used for all plots. The SI prefix is for simulation data and the SS is for steady state analysis. The SI AUTO SCALES and SS AUTO SCALES commands can be used to return plotting to the automatic scaling mode. Auto Scales are selected so that they span each plotted quantity. The auto scale option is the default used until manual scales are requested.

PLOT ON
PLOT OFF
These program commands allow the plotted output to be turned on or off. The default condition is PLOT OFF. It is, therefore, necessary to include the PLOT ON command before requesting any analysis from which plots are desired. The PLOT OFF and PLOT ON commands can be issued between analysis requests if it is desired to omit the plotting of certain analysis results.

**OMIT PLOT POINTS**

Boxes are normally drawn around each plotted data point. This command suppresses these boxes. A second occurrence of this command restores the boxes around plotted data points.

**CALCOMP**

**PRINTER PLOTS**

**SC4020**

**MTS PLOTS**

Plots are routed to a particular physical device by specifying the above commands prior to the analysis which generates plotted data. Printer plots, MTS plots, and either CALCOMP or SC4020 plots may be generated in the same run.

**PLOT ID**

**TITLE**

The PLOT ID program command allows an identification label to be placed as the first page of plotted output. Up to 48 characters may follow the delimiter that follows the PLOT ID command. This identification must be used to place mailing information on the plotted output.

The TITLE command allows a common title to be placed on all plotted output. Up to 74 characters may follow the delimiter that follows the TITLE command. The TITLE command may be changed before each analysis. Once
defined, the title remains in effect until a new title is entered. Examples of these commands are shown below:

PLOT ID = EX USER **M/S 70-16**
TITLE = FLEX MODE CASE

6. STEADY STATE COMMANDS

STEADY STATE

This program initiates the calculation of the system steady state. Associated with this command are the program name and values:

1. SS PARAMETER = steady state parameter.
2. SS START = initial value of steady state parameter.
3. SS STOP = final value of steady state parameter.
4. SS POINTS = number of values the steady state parameter takes going from SS START to SS STOP.
5. SS ITERATIONS = maximum number of iterations allowed per steady state calculation.
6. PRINT CONTROL = print control variable.

SS PARAMETER specifies the parameter to scan from the value SS START to SS STOP in SS POINTS steps. SS ITERATIONS specifies an upper limit on the number of iterations to be used to calculate a steady state. The default value of SS ITERATIONS is 30. If the SS PARAMETER is blank, a single steady state calculation will occur. The steady state parameter can be any valid parameter name.

The PRINT CONTROL parameter provides all the print control functions described in Section III.4 for simulation operation plus two extra forms, 6 and 7, which may be used to track the steady state iteration process.

The following example will scan the parameter RPM over the range from 19000 to 16000 in five steps. At the end of the steady state calculation, the
system stability will be checked to assure that a stable steady state exists.

\[ \text{SS PARAMETER} = \text{RPM}, \quad \text{SS START} = 19000, \quad \text{SS STOP} = 16000 \]
\[ \text{SS POINTS} = 5, \text{STEADY STATE}. \]

If plots of the steady state scan are desired, these plots should be defined using the DISPLAY commands prior to initiating the steady state calculations. Only those plots which have an independent variable different from time will be plotted.

In the following example, the steady state parameter is set to a blank phrase. This is accomplished by placing the SS PARAMETER command phrase at the end of a command line. If it is desired to follow the SS PARAMETER program name with other instructions, then the form: \text{SS PARAMETER = NONE} may be used. In either case, this causes a single steady state calculation to occur at the current operating point. The results of this calculation are then loaded into the initial condition vector, XIC. The initial default value of SS PARAMETER is a blank phrase so that single steady state calculations will be performed, unless this parameter is set to a non blank name.

\[ \text{SS PARAMETER} = \]
\[ \text{STEADY STATE} \]
\[ \text{XIC-X} \]

7. LINEAR ANALYSIS COMMANDS

LINEAR ANALYSIS

This program command causes the calculation of a linearized version of the given nonlinear model at the operating point specified by XIC and then calculates the eigenvalues of this linear approximation. A printout of the following quantities are generated by this command:
1. The state operating point (INITIAL CONDITIONS)
2. The state perturbation size (ERROR CONTROL)
3. The integrator status (INT CONTROL)
4. The rates at the operating point

For continuous systems:

5. The system stability matrix
6. A measure of the linearity of each element of the stability matrix if a nonlinear condition is detected.
7. The system eigenvalues, real and imaginary parts, natural frequencies, and damping ratios.

For discrete systems:

8. Continuous states stability matrix (displays inputs to continuous states)
9. Transition matrix for each sample period (displays inputs to discrete states at each sample period)
10. Total system transition matrix
11. System eigenvalues, real and imaginary parts in both Z and S planes and natural frequencies and damping ratios in the S plane.

EIGENVECTOR

The EIGENVECTOR command is similar to the LINEAR ANALYSIS command. However, in response to this command, the modal matrix comprised of the system eigenvectors is also calculated and printed. This command can only be used with models that contain an optimal controller, due to core requirements.
8. STABILITY MARGIN COMMANDS

STABILITY MARGINS

This program command initiates the calculation of the stability margins for those parameters specified by the SM PARAMETERS command. The maximum and minimum values that each specified parameter can take for stable system operation and the oscillation frequencies that result if either boundary is violated are determined.

SM PARAMETERS

This program command allows up to ten parameters to be specified for stability margin calculations. The command is followed by from one to ten parameter names separated by delimiters. This command destroys all previously stored stability margin parameters.

An example use of these commands is given below:

SM PARAMETERS = GK1TC, GK2TC

STABILITY MARGINS

These commands cause the stability margins to be calculated for the two parameters, GK1TC and GK2TC.

A summary of stability margins and frequencies is printed along with the nominal system eigenvalues, and the system eigenvalues with each stability margin parameter set equal to zero. If no upper or lower stability margin is located for a particular stability margin parameter, the summary array will contain the number 11111 in those locations for which no margin limit was determined.

The stability margin search is limited to parameter values of the same algebraic sign as the nominal value. Thus, for example, zero is the lowest magnitude that will be considered for the lower stability boundary of a parameter with a positive nominal value.
9. FREQUENCY RESPONSE COMMANDS

TRANSFER FUNCTION
TF INPUT
TF OUTPUT

These program commands are used to initiate the calculation of a frequency response function, between any two specified points in the model. The following command phrases are used to set up the desired transfer function:

TF INPUT = transfer function input variable
TF OUTPUT = transfer function output variable

They are used to specify the input and output points in the system model. These quantities must be set to the desired names before requesting the frequency response calculation. They may be set to any valid state, rate, variable, or parameter name. The command TRANSFER FUNCTION causes the frequency response function to be executed at that point.

The transfer function poles and zeros are printed output. For discrete systems, these roots are given in both the Z plane and S plane.

BODE
NYQUIST
NICHOLS

These program commands specify the format to be used for the frequency response plots. The format must be specified before requesting the TRANSFER FUNCTION analysis. If not specified, the default will be a Bode plot format.

TF AUTO SCALES
TF MANUAL SCALES
FREQ MIN
FREQ MAX
These program commands are used to set the frequency range of the frequency response plots. It can be either automatically determined by the range of eigenvalues or be specified by the following command phrases:

1. FREQ MIN = minimum frequency, r.p.s.
2. FREQ MAX = maximum frequency, r.p.s.

The default condition is for automatic scales.

In the automatic mode, the minimum and maximum frequencies will be one decade below and one decade above the lowest non zero and highest natural frequency. For discrete systems, the upper frequency is bounded by the Nyquist frequency of the system. Frequency points are concentrated around lightly damped natural frequencies to better define these critical areas.

The following example will generate a transfer function from C4 MC to S2 LA with automatic frequency values for the plotted results in a Nichol's chart format.

TF INPUT = C4 MC, TF OUTPUT = S2 LA
NICHOLS, TRANSFER FUNCTION

10. ROOT LOCUS COMMANDS

ROOT LOCUS
RL PARAMETER
RL START
RL STOP
RL POINTS

These program commands initiate the calculation of a root locus. The following commands are used to select the parameter and the ranges for the locus.
1. RL PARAMETER = root locus parameter name
2. RL START = initial value of root locus parameter
3. RL STOP = final value of root locus parameter
4. RL POINTS = number of rootings to be made going from RL START to RL STOP

They specify the parameter to scan from the value RL START to RL STOP in RL POINTS steps. The default values of RL PARAMETER, RL START, RL STOP, and RL POINTS are: blank, 0., 1., and 6. respectively.

The root locus parameter, like the steady state parameter, can be either a valid parameter name or a state variable name followed by the phrase IC. This latter usage is meaningful only if the specified state variable has been frozen using the INT CONTROL command. In this way, a root locus can be performed as a function of the operating point value of a frozen state variable.

   RL PARAMETER = ZO TF, RL START = 0, RL STOP = 5, RL POINTS = 6,
   ROOT LOCUS

In this example, the root locus parameter ZO TF is scanned from 0 to 5 in six equally spaced steps.

   RL MANUAL SCALES, REAL MAX=5, IMAG MAX=5, INT CONTROL, SPEED=0
   RL PARAMETER = SPEED, IC, RL START = 35, RL STOP = 45
   ROUT LOCUS

In this example, manual scales are specified for the root locus plots. The SPEED state variable is then frozen and a root locus is performed on the SPEED operating point.
These program commands allow the scales of the root locus plots to be either automatically determined by the range of eigenvalues or to be specified by control commands. The following command definitions are used to set plot scales:

1. REAL MIN = minimum real axis range, r.p.s.
2. REAL MAX = maximum real axis range, r.p.s.
3. IMAG MIN = minimum imaginary axis range, r.p.s.
4. IMAG MAX = maximum imaginary axis range, r.p.s.

The default condition is for automatic scales.

11. EIGENVALUE SENSITIVITY COMMANDS

EIGEN PARAMETER
EIGEN SENSITIVITY

These program commands cause a linear approximation of the given nonlinear model to be generated and then evaluates the sensitivity of the system eigenvalues to a parameter specified by the command phrase EIGEN PARAMETER.

In the following example, the sensitivity of system eigenvalues to the parameter GPITF will be calculated.

EIGEN PARAMETER = GPITF, EIGEN SENSITIVITY
12. FUNCTION SCAN COMMANDS

SCAN1
SCAN2
DEPEN
INDEP1
INDEP2
START1
STOP1
START2
DELTA2
CURVES2

These program commands initiate and control the calculation of general algebraic functions of one or two independent variables. The following definitions are used to specify the control parameters and bounds for the calculation.

1. DEPEN = dependent variable
2. INDEP1 = 1st independent variable
3. INDEP2 = 2nd independent variable
4. START1 = starting point of 1st independent variable
5. STOP1 = stopping point of 1st independent variable
6. START2 = starting point of 2nd independent variable
7. DELTA2 = increment of 2nd independent variable
8. CURVES2 = number of values of 2nd independent variable

These commands specify the dependent and independent variables and scan ranges of these quantities. These quantities must be set to their desired values, before requesting the general algebraic function evaluation. If a single function is requested, i.e., SCAN1, only items 1,2,4, and 5 need be specified.

DEPEN = W2 TU, INDEP1 = EH SH, INDEP2 = S1 DE2, START1 = -30
STOP1 = 100, START2 = 10, DELTA2 = 20, CURVES2 = 6
SCAN2
In the above example, the quantity $W_2$ TU will be calculated as a function of quantities $EN$ SH and $S_1$ DE2. Six curves will be generated with $W_2$ TU ranging from -30 to 100 and $S_1$ DE2 being stepped from 10 to 20 in 6 steps of 2 each.

13. OPTIMAL CONTROLLER DESIGN COMMANDS

In order to design an optimal controller using the EASY program, it is necessary to specify the inputs and outputs of the optimal controller as part of the system model description. This is accomplished as described in Section II.2.b. Once a model has been generated that contains an optimal controller and the specified input-output connections to the other model components, many different controllers can be designed. These variations are made by varying the operating point or the optimal controller design criteria. The following paragraphs describe how the optimal controller operating point and criteria are specified.

Once an optimal controller has been designed, it may be desired to save that design for further analysis on subsequent analysis runs. Program commands are provided to save the data arrays which specify a particular optimal controller and to read such data on subsequent analysis runs.

**O.C. DATA**

The O.C. DATA command specifies that the following command phases contain data for one or more of the ten different data arrays related to optimal controllers. The name of each of these arrays and a brief description of its use is given below. For a more complete description of each array and its use, see Section 4.5 of reference 1.

Optimal Controller - Operating Point Specification

**YOP** - Optimal controller input operating point (set-point array. $YOP$ is an $n_s$ dimensional array, where $n_s$ is the number of inputs to the optimal controller. Default values of zero are provided for this array.
UOP - Optimal controller output operating point (set-point) array. UOP is an \( n_u \) dimensional array, where \( n_u \) is the number of outputs from the optimal controller. Default values of zero are provided for this array.

Optimal Controller Criteria Specification

Q - Optimal controller criteria weights array. Q is an \( n_c \) dimensional array, where \( n_c \) is the number of optimal controller criteria variables. Q contains the diagonal elements of the positive semi-definite weighting matrix which gives the importance of the various criteria variables relative to each other and the controller outputs. Off diagonal elements are assumed equal to zero. If the criteria variables are not specified, they are assumed to be the optimal controller inputs. Default values of 1 are provided for this array.

RU - Optimal controller control weights array. RU is an \( n_u \) dimensional array, where \( n_u \) is the number of optimal controller outputs. RU contains the diagonal elements* of the positive definite matrix which gives the importance of the various controller outputs relative to each other and the criteria variables. Off diagonal elements are assumed equal to zero. Default values of 1 are provided for this array.

CD - System model disturbance covariance array. DC is an \( n_x \) dimensional array, where \( n_x \) is the order of the system model. DC contains the diagonal elements* of the model disturbance covariance matrix which gives the uncertainty of various model states relative to each other and the sensed quantities. Off diagonal elements are assumed equal to zero. Larger values in CD imply greater uncertainty (less confidence) in the system model accuracy. Default values based on the ERROR vector and the model stability matrix are provided for this array.
CS - Optimal controller inputs disturbance covariance array. CS is an \( n_s \) dimensional array, where \( n_s \) is the number of inputs to the optimal controller. CS contains the diagonal elements* of the sensed quantity disturbance covariance matrix which gives the uncertainty of various sensed quantities relative to each other and the model states. Off diagonal examples are assumed equal to zero. Larger values in CS imply greater uncertainty (less confidence) in the sensed quantity accuracy. Default values based on the ERROR vector and the model sensor matrix are provided for this array.

Optimal Controller Specification

These inputs are required only for reloading a previously designed optimal controller. Default values of zero are provided for these arrays until nonzero values are calculated via the DESIGN O.C. command.

\( G \) - Optimal controller gain array. \( G \) is an \( n_u \) by \( n_{rc} \) dimensional array, where \( n_u \) is the number of outputs from the \( n_{rc} \) is the order of the optimal controller.

\( S \) - Optimal controller sensor array. \( S \) is an \( n_{rc} \) by \( n_s \) dimensional array, where \( n_{rc} \) is the order of the optimal controller and \( n_s \) is the number of inputs to the optimal controller.

\( AK \) - Optimal controller stability matrix array. \( AK \) is an \( n_{rc} \) by \( n_{rc} \) dimensional array where \( n_{rc} \) is the order of the optimal controller.

\( FK \) - Optimal controller d.c. gain matrix array. \( FK \) is an \( n_u \) by \( n_s \) dimensional array where \( n_u \) is the number of outputs from and \( n_s \) is the number of inputs to the optimal controller.
Optimal controller array data may be entered in a free field format with each data item separated by a comma or another one of the standard delimiters. Data may be entered along either a row, column or diagonal line of the array. The row and column location is given for only the first element specified. The following input values are loaded in the subsequent row, column, diagonal elements of the array. The letters C, R, and D signal the start of a new Column, Row, or Diagonal input. They must be followed by the row and column number at which data loading is to start. A column number of 1 must be given for the one dimensional arrays: YOP, UOP, Q, RU, CD, and CS. The letter Z causes all elements of the array to be set to zero. This command may be used to advantage when loading a sparse array.

If the number of data values exceeds either the row or column dimension of the array, the excess values are ignored by the program.

The following example demonstrate the loading of data into the optimal controller arrays.

PROGRAM COMMANDS

O.C. DATA
YOP = C (1,1) 553.2, 546, -2.56, 7

RESULTS - Assuming YOP is a 4x1 array.

553.2
546.
YOP = -2.56
7.00
DESIGN O.C.

The DESIGN O.C. command initiates the optimal controller design process. Before issuing this command, the following items should be accomplished:

1. Specify the optimal controller operating point by loading the arrays YOP and UOP.

2. Place the system model at the desired operating point.

3. Specify those optimal controller criteria arrays Q, RU, CD, and CS which you wish to differ from the default values.

The DESIGN O.C. command causes a linear model of the system to be generated and an optimal controller to be designed. The design results are printed and loaded into the optimal controller arrays G, S, AK, and FK. Manual modifications to the optimal controller can be made via the O.C. DATA command.

SAVE O.C.

The SAVE O.C. command causes the optimal controller arrays G, S, AK, and FK to be placed on local file TAPE3 in a format compatible with the O.C. DATA command. This file may be saved as a permanent file or punched as data cards by the appropriate control cards. By including these cards or records in the input data for subsequent analysis runs, it is possible to perform further analyses on a previously calculated optimal controller. Such optimal controller data could be used in conjunction with the O.C. ANALYSIS command to the Model Generation Program. As described in Section II.2.b, the O.C. ANALYSIS command allows analyses to be performed on a previously designed optimal controller with less computer central memory than is required to perform the optimal controller design.
14. WARNING MESSAGES

One or more of the following warning messages will occur if the program encounters difficulty in interpreting analysis instructions or performing an analysis. These messages will be preceded by:

*** WARNING ***.

The symbols xxx, zzz, or nnn are used to indicate phrases from the analysis description that are included as part of the warning message. The following messages are listed in alphabetical order:

1. A VALID PARAMETER NAME MUST PRECEDE THE NUMERIC VALUE nnn

This message indicates that a valid parameter name was not identified preceding the numeric value nnn. Check for missing delimiters or misspelled parameter name.

2. ALGEBRAIC LOOP WITH GAIN OF nnn EXISTS BETWEEN INPUT AND OUTPUT THIS TRANSFER FUNCTION CAN NOT BE DETERMINED.

See Appendix M for a description of this limitation to the transfer function analysis method.

3. ALL ROOTS CANCELED. THIS CASE WILL BE SKIPPED

This indicates TF output is not connected to TFD input. Check model, TF input, and TF output specifications.

4. nn IS NOT A VALID SUBSCRIPT

Subscripts must be numeric.
5. xxx IS NOT A VALID TABLE NAME
Check spelling of table name.

6. xxx IS NOT A VALID TABLE NAME FOR THIS MODEL. DATA WILL BE IGNORED
Check spelling of table name.

7. CAN'T FIND GREATEST COMMON DIVISOR FOR THE FOLLOWING SAMPLE RATES
Check sample period values.

8. CAN'T FIND LEAST COMMON MULTIPLE FOR THE FOLLOWING SAMPLE RATES
Check sample period values.

9. CAN'T IDENTIFY xxx AS A VALID EIGENVALUE SENSITIVITY PARAMETER
Check spelling of eigenvalue sensitivity parameter or for missing delimiters.

10. CAN'T IDENTIFY xxx AS A VALID PRINT VARIABLE
Check spelling of xxx or for missing delimiters.

11. CAN'T IDENTIFY xxx AS A VALID ROOT LOCUS
Check spelling of xxx or for missing delimiters.

12. CAN'T IDENTIFY xxx AS A VALID SCAN PARAMETER
Check spelling of xxx or for missing delimiters.
13. CAN'T IDENTIFY xxx AS A VALID STABILITY MARGIN PARAMETER

Check spelling of xxx or for missing delimiters.

14. CAN'T IDENTIFY xxx AS A VALID STEADY STATE PARAMETER

Check spelling of xxx or for missing delimiters.

15. CAN'T IDENTIFY xxx AS A VALID TRANSFER FUNCTION INPUT (OUTPUT) PARAMETER

Check spelling of xxx or for missing delimiters.

16. xxx CAN'T BE SET EQUAL TO zzz. VALUE MUST BE NUMERIC

Check for missing numeric value or delimiters.

17. CAN'T IDENTIFY xxx VALUE WILL BE IGNORED

This will result in not setting the quantity intended by xxx to its new value. Check for spelling of xxx or for missing delimiters.

18. CAN'T INTERPRET xxx

The phrase xxx cannot be recognized as a valid program command, program name, or program value. Check spelling of xxx or for missing delimiters.

19. CAN'T LOAD CRITERIA ARRAYS WHEN IN ANALYSIS ONLY MODE

The O.C. ANALYSIS command was issued to the Model Generation program when it created the system model. Therefore, an optimal control design, which used this criteria arrays, cannot be performed.
20. INVALID SUBSCRIPT DETECTED

Subscript outside valid range for this array.

21. SUBSCRIPT VALUES nn OR nn ARE TOO LARGE FOR xxx

Subscripts outside allowable range.

22. WORK SPACE WAS NOT PROVIDED IN MODEL FOR OPTIMAL CONTROLLER DESIGN OR EIGENVECTOR CALC.

An optimal controller must be specified in model description in order to have work storage for optimal control design of eigenvector calculation.

23. nnn EXCEEDS THE ALLOWABLE INDEX RANGE FOR xxx THIS QUANTITY WILL NOT BE DEFINED

The number nnn was outside the allowable range of states, rates, variables, or parameters. Therefore, the name xxx cannot be assigned as a name for the nnnth state, rate, variable or parameter.

24. nn IS OUTSIDE ALLOWABLE INDEX RANGE. zzz WILL NOT BE DEFINED

Index number nn must be between 1 and number of states, variables, or parameters, (whichever is applicable).

25. FAILED TO CONVERGE TO ZERO PHASE

The search procedure described in Appendix M failed to converge to zero phase. The stability margin for the indicated parameter cannot be determined by this method.
26. MORE THAN 10 UNIQUE SAMPLE RATES LOCATED

Only 10 different sample rates allowed.

27. NO SAMPLING PERIODS ARE GIVEN

Sampling period parameters TAU xxx could not be located. These names can not be redefined.

28. NOMINAL SYSTEM UNSTABLE

The nominal system is unstable. The stability margins of the specified parameters will be calculated, but these bounds will be "non-critical" bounds since the nominal system is unstable. See Section 4.4.4 of reference 1 for a discussion of critical and noncritical stability boundaries.

29. NON-ALPHA NAME ON THIS CARD --- xxx. WILL IGNORE THIS CARD

The table inputs routine expected an alphanumeric table name but encountered a numeric value on the data card printed. Check the sequence and number of tabular data cards to assure that they match those required by the model's tables and table input formats. See Section III.1.b for correct formats.

30. NON-NUMERIC DATA ON THIS CARD --- xx. WILL READ NEXT TABLE

The table input routine expected a numeric value but encountered an alphanumeric name on the data card printed. Check that the sequence and number of tabular data cards matches the model's tables and table input formats. See Section III.1.b for correct formats.
31. **nnn PRIMARY AND xxx SECONDARY INDEPENDENT VARIABLE POINTS EXCEEDS THE**
**zzz WORD STORAGE LIMIT FOR THE FOLLOWING TABLE. SOME DATA WILL BE**
**LOST**

See Section II.2 for a discussion on how to set the maximum number of data points allowed for each table.

32. **SIMULATION WILL NOT BE RUN DUE TO FAILURE TO REACH VALID STEADY STATE**

A failure of the steady state analysis followed by a request to transfer X into XIC causes an interlock to be set which will prevent a simulation run from beginning from an erroneous initial condition.

33. **WORK SPACE WAS NOT PROVIDED IN MODEL FOR OPTIMAL CONTROLLER DESIGN**

Either no optimal controller was specified to the Model Generation Program or the O.C. ANALYSIS mode was indicated. In either case, only analyses and not DESIGN O.C. can be performed with this model.

34. ***** WARNING *** MATRIX IS SINGULAR *** INITIAL SYSTEM IS NOT DIAGONALIZABLE**

This message is generated in the system reduction program and is the result of multiple eigenvalues with a single eigenvector. This means that the system is not able to be diagonalized and that a Jordan type reduction is required. Processing is stopped and reduction is not completed. This message can arise either in the reduction of the initial model equations or in the reduction of the controller.
35. *** WARNING *** QR FAILED TO CONVERGE IN XX STEPS

This message generated in the system reduction program is the result of the extremely rare event of the eigenvalue calculation failure.

36. ** DUE TO xxx UNSTABLE EIGENVALUES. SYSTEM REDUCTION TO xxx IS IMPOSSIBLE

This message generated in the system reduction program is the result of the number of unstable eigenvalues in the system to be reduced being greater than the requested order for the reduced system. This message can arise either in the reduction of the initial system or in the reduction of the controller.

37. ** CONTROL WEIGHTING NOT POSITIVE DEFINITE

This message generated in the calculation of the optimal feedback matrix in the result of loss of significance in the calculation of the control weighting matrix. Since the default check is made, this is a rare event.

38. **... QR ALGORITHM FAILED TO CONVERGE
   **... SYSTEM MAY BE UNSTABILIZABLE

This message generated in the calculation of the optimal feedback matrix is the result of the QR algorithm failure and is a rare event.

39. **... SPECTRAL FACTORIZATION OF EIGENVALUES NOT OBTAINED
   **... SYSTEM MAY BE UNSTABILIZABLE

This message generated in the calculation of the optimal feedback matrix is the result of an eigenvalue with a zero real part preventing spectral factorization. It is the result normally of an uncontrollable mode with an eigenvalue with a zero or very small real part.
40. **... MATRIX IS SINGULAR
   **... SYSTEM PLUS ADJOINT EQUATIONS NOT DIAGONALIZABLE OR
   SYSTEM IS UNSTABILIZABLE

This message generated in the calculation of the optimal feedback matrix is
the result of the set of pseudo eigenvectors calculated for the partitioned
eigenvalues being singular in the top block. This condition normally means
that an unstable, uncontrollable mode existed in the original system. Another, but rare possibility is that due to multiple eigenvalues, the
system plus adjoint equations was not diagonalizable.

41. **... OR FAILED TO CONVERGE
   **... SYSTEM MAY BE UNOBSERVABLE

This message is generated during the calculation of the Kalman filter and
is the result of the QR algorithm failure and is a rare event.

42. **... SPECTRAL FACTORIZATION OF EIGENVALUES NOT OBTAINED
   **... SYSTEM MAY BE UNOBSERVABLE

This message is generated during the calculation of the Kalman filter and
is the result of an eigenvalue with zero real part preventing spectral
factorization. It is normally the result of an unobservable mode with an
eigenvalue with zero or very small real part.

43. **... MATRIX IS SINGULAR
   **... SYSTEM MAY BE UNOBSERVABLE

This message is generated during the calculation of the Kalman filter and
is normally the result of an unstable unobservable mode. Like the case in
the gain matrix calculation (Section 4.6.30 of reference 1), it can rarely
be the result of the system and adjoint equations being undiagonalizable.
44. **... QR ALGORITHM FAILED TO CONVERGE

This message occurs when during a simple eigenvalue calculation, convergence was not obtained. This is a rare event.

45. **... SYSTEM HAS SINGULAR ALGEBRAIC LOOP

This message generated during the adjustment of the controller is the result of cancellation in algebraic feedforward and feedback loops. It can normally be corrected by the use of an alternative adjustment method.

15. RENAMING MODEL INPUTS AND OUTPUTS

For some applications, it may be desirable to rename the parameters, states, rates, and variables created by EASY5 standard components. This can be done by the following analysis program commands:

```
DEFINE PARAMETERS
DEFINE STATES
DEFINE RATES
DEFINE VARIABLES
```

Each command is followed by pairs of names. The first name is the EASY5 standard component name. The second name is the desired new name. For example, the outputs of the lag component LA may be changed to AILERON, and the lag gain and time constant may be changed to KSERVO and TSERVO.

```
DEFINE STATES = $2 LA = AILERON
DEFINE PARAMETERS = GAILA = KSERVO, TC LA = TSERVO
```

Once a quantity has been redefined, all references to that quantity in analysis program commands must utilize the new name. The subroutine EQM0, which is prepared by the EASY5 Model Generation Program, will still refer to all quantities by their original EASY5 generated names.
16. COMPUTING TYPE ZERO TRANSFER FUNCTIONS WITH EASY

A continuous dynamical system (with prescribed input and output quantities) has a Type Zero transfer function if either:

1. A change quantity in the input has an immediate change in the output quantity
or equivalently:
2. The order of the numerator of the transfer function is the same as the order of the denominator

The method currently used by the EASY Dynamic Analysis Program is unable to compute transfer functions of systems of Type Zero. This will be remedied in the future, but the following provides an interim method:

A. In the model description file:
   1. Add a new LA standard component. We will name this component LATF but you may use any unused component identifier.
   2. Connect the output of the new LA component to the original system input quantity.

B. In the system analysis file:
   1. Set the parameters for the new LA component as
      \[ \text{ZOLATF} = 1 \quad \text{ZILATF} = 0 \quad \text{POLATF} = -1.0 \times 10^{28} \]
   2. Change the TF INPUT quantity from the original quantity to S1 LATF

C. Submit job using new model description and analysis files.

D. The results of the TRANSFER FUNCTION analysis will provide:
   1. The zeros and poles of the original system plus a pole at \(10^{28}\) radians per second. This extra pole should be ignored.
2. The frequency response will be the correct frequency response for the original system up to frequencies above $10^{20}$ radians per second.

These high frequency values can be suppressed from the lineprinter output and the graphs by using the TF MANUAL SCALES option.
SECTION IV

STANDARD COMPONENTS AND EASIEST SUBROUTINES

This section describes the EASIEST standard components available for system modeling that were designed from the SAFEST computer program. Other components that may be used by the analyst in conjunction with the EASIEST routines are described in Appendix K.

1. Standard Components

The following is a list of the EASIEST standard components:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>Attached body (survival kit)</td>
</tr>
<tr>
<td>AE</td>
<td>Airplane</td>
</tr>
<tr>
<td>AG</td>
<td>Atmospheric properties</td>
</tr>
<tr>
<td>AM</td>
<td>Aeromedical</td>
</tr>
<tr>
<td>AP</td>
<td>Aerodynamic plate</td>
</tr>
<tr>
<td>AS</td>
<td>Seat aerodynamics</td>
</tr>
<tr>
<td>CE</td>
<td>Crewperson</td>
</tr>
<tr>
<td>CS</td>
<td>Airplane control surfaces</td>
</tr>
<tr>
<td>CT</td>
<td>Catapult</td>
</tr>
<tr>
<td>DR</td>
<td>DART</td>
</tr>
<tr>
<td>GP</td>
<td>Simple parachute mortar and restraints</td>
</tr>
<tr>
<td>LI</td>
<td>Parachute lines</td>
</tr>
<tr>
<td>MP</td>
<td>Parachute mortar and restraints</td>
</tr>
<tr>
<td>PC</td>
<td>parachute</td>
</tr>
<tr>
<td>RL</td>
<td>Rails</td>
</tr>
<tr>
<td>RS</td>
<td>Restraints</td>
</tr>
<tr>
<td>SE</td>
<td>Seat equations of motion</td>
</tr>
<tr>
<td>SL</td>
<td>Sled</td>
</tr>
<tr>
<td>SP</td>
<td>STAPAC</td>
</tr>
<tr>
<td>SR</td>
<td>Sustainer rocket</td>
</tr>
<tr>
<td>WB</td>
<td>Weight and balance</td>
</tr>
</tbody>
</table>
This section gives an explanation of each of the aforementioned ejection seat components. These descriptions are intended to assist the user in utilizing them to model escape systems. Input/output tables and descriptive figures for each of these components are presented in alphabetical order in Appendix D, and should be thoroughly examined before modeling an ejection system.

A source listing of the EASIEST components and associated subroutines are presented in Appendices G and H. These listings have been thoroughly commented to provide additional information on how the algorithms were coded and to assist in solving special case errors.

STANDARD COMPONENT AB

This component is simply the equations of motion for a point mass. It was designed to model a survival kit attached to the crew member, but can be used to simulate any object that might be attached to the escape system. Component restraints (RS) is used to restrain AB to its parent object. The input/output list for this component is given in Appendix D. Inputs include the forces and torques that act on the point mass, as well as its inertial properties.

STANDARD COMPONENT AE

This component models the EASIEST airplane. The airplane is internally trimmed by the STEADY STATE command to the airspeed and altitude specified by the user. Control surface and thrust commands that maneuver the airplane after trim are interpreted as being an addition to the settings required for trim. Additional inputs include the forces and torques from the DART, rails, and catapult components. An example of a model that uses component AE is given in Appendix N. Additional airplane information is presented in Section IV.3.

Component AE was written to use existing SAFEST aerodynamic coefficient tables and table look-up routines with the exception that coefficient
input data has been reorganized so they contain the coefficients in the following order:

<table>
<thead>
<tr>
<th>NR</th>
<th>COEFFICIENT LOCATION</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CZO</td>
<td>Z axis bias coefficient</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CZAD</td>
<td>Variation of CZO with alpha dot</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CZQ</td>
<td>Variation of CZO with pitch rate</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CZDE</td>
<td>Variation of CZO with elevator position</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CZDA</td>
<td>Variation of CZO with aileron position</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CXO</td>
<td>X axis bias coefficient</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CXDA</td>
<td>Variation of CXO with aileron position</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CMO</td>
<td>Pitching moment bias coefficient</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CMAD</td>
<td>Variation of CMO with alpha dot</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CZQ</td>
<td>Variation of CMO with pitch rate</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CMDE</td>
<td>Variation of CMO with elevator position</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>CMDA</td>
<td>Variation of CMO with aileron position</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>CYB</td>
<td>Variation of CY with beta</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>CYP</td>
<td>Variation of CY with roll rate</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>CYR</td>
<td>Variation of CY with yaw rate</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>CYDR</td>
<td>Variation of CY with rudder position</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>CYDA</td>
<td>Variation of CY with aileron position</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>CLB</td>
<td>Variation of CI with beta</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>CLP</td>
<td>Variation of CI with roll rate</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>CLR</td>
<td>Variation of CI with yaw rate</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>CLDR</td>
<td>Variation of CI with rudder position</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>CLDA</td>
<td>Variation of CI with aileron position</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>CNB</td>
<td>Variation of Cn with beta</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>CNP</td>
<td>Variation of Cn with roll rate</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CNR</td>
<td>Variation of Cn with yaw rate</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>CNDR</td>
<td>Variation of Cn with rudder position</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>CNDA</td>
<td>Variation of Cn with aileron position</td>
<td></td>
</tr>
</tbody>
</table>
A listing of the F4E airplane maneuvering coefficients modified to be used with EASIEST is shown in Appendix J.

Component CS (airplane control surfaces) can be used to maneuver the airplane. The method employed to do this is described in this section under the heading STANDARD COMPONENT CS. This component is also included in the example presented in Appendix N.

STANDARD COMPONENT AG

Component AG calculates the atmospheric density and the speed of sound, while supplying the wind velocity to the model. It should be the first component specified in the Model generation Program input file, and must be included in all EASIEST models.

Note that variables H, BP, and TE must be initialized if a non-standard atmosphere is to be used with the model. Setting variable BP to zero, which is its default, establishes a standard atmosphere. The wind velocity input vector provides the capability to model an ejection system where adverse winds (i.e., storm cells, turbulence, down drafts, etc.) could be a factor in an ejection seat design. This feature may be valuable when using the EASIEST program to investigate an aircraft accident.

During initialization (CALC XIC command), component AG establishes the atmospheric properties from the input parameters. Subsequent passes through the model updates the wind vector. If a standard component needs atmospheric data, it is acquired by a call to subroutine ATMOS, which refers to the ENTRY ATMOS statement in component AG.

STANDARD COMPONENT AM

This component acts essentially as the interface between program Aeromed, the aeromedical post processor, and either component SE (seat equations of motion) or CE (crewperson). The routine writes onto TAPE 7 the aeromedical parameters and variables required by Aeromed. This process is initiated by
a flag that is an input into the component. No more than 4000 variable sets can be written to this tape at a time interval no less than 0.001 seconds, or the integrator report interval, TINC, whichever is the largest. (See Section III.4 for an explanation of TINC.)

Components CE and SE both calculate the aeromedical variables, and either one can be used to drive the aeromedical inputs in this component. Note that most of the required parameter inputs have specific defaults, which can be adjusted by the user if necessary.

STANDARD COMPONENT AP

This EASY module calculates the seat body axis force and torque components acting on the ejection seat from an attached object, such as an airfoil device or inflatable afterbody designed to augment the stability of the ejection seat. Appendix D presents its input/output lists. Inputs include the tables that define the x-axis and the z-axis force coefficients, the plate centroid in the seat coordinate system, and the airplane z-axis position at the point where the plate centroid enters the windstream. The plate centroid acts as the origin of the plate coordinate system, and the plate can be rotated about this point with respect to the seat. Figure 22 provides an input/output overview for this component.

STANDARD COMPONENT AS

Component AS determines the aerodynamic forces and torques that are exerted on the seat. It employs the same coefficient input data and table look-up routines as the SAFEST program. The input/output information is contained in Appendix D. Inputs include emergence coefficients, the yaw, pitch, and roll damping derivatives, and a table that defines the exposed area of the seat as a function of the exposed length during emergence. Figure 23 presents a diagram that helps to explain the function of component AS.
Both the rocket on and rocket off aerodynamic coefficient tables are available at any given time to accommodate the situation where two ejection seats are being modeled, one of which has its rocket on, the other off. Each of these coefficient tables are hard coded into this component, and contain the six basic aerodynamic coefficients: the three body axis force coefficients \((C_X, C_Y, C_Z)\), and the three body axis torque coefficients \((C_l, C_m, C_n)\).

STANDARD COMPONENT CE

This EASIEST standard component computes the aerodynamic forces and torques acting on the percentile crewperson that is specified in the CE input data. These forces and torques are then summed with the other forces and torques acting on him (parachute lines, seat restraints, etc.) to determine the linear and angular rates to be used by the integrator. The input/output listings are presented in Appendix D.

Note that the moments and products of inertia for the crewmember are required inputs. The values for these parameters should reflect the inertial properties of a seated crewmember whose percentile is approximately the same as that specified in the input data. At seat/crewmember separation, new moment and product of inertia vectors are calculated via a table look-up on data hard coded into the component, with the independent variable being the crew member percentile. The aerodynamic reference area and length are also determined by this table look-up, as is the crewmember weight. The weight of the crewmember's clothing and equipment is a separate parameter input.

STANDARD COMPONENT CS

This component can be employed to move the rudder, elevator, and ailerons of the airplane component (AE). All three control surfaces may be moved simultaneously or individually according to the input parameters specified by the user. These parameters include the simulation time after which the control surface rates are calculated, the commanded position, and a time
constant that is employed by a first order lag function to determine the rates. The input/output data is given in Appendix D.

COMPONENT CT

Component CT determines the forces and moments acting on the seat and airplane from a closed tube catapult. The states in this module include the internal friction energy, heat loss, catapult work, and the propellant web consumed. These states are used to calculate the internal temperature of the catapult, from which the pressure is calculated by using the equation of state with the chamber volume and the mass of the burned propellant. The force can then be calculated from the geometry of the catapult pressure chamber.

The input/output parameters are shown in Addendix D, and include the flag for catapult ignition, the unloaded catapult length, and the catapult propellant consumption table. Figure 24 presents an overview of some of the required inputs, and should be helpful in visualizing the geometry and operation of the catapult. Note that input TDE is available as the time interval over which the catapult force decays to zero after stripoff. This decay period should prevent the variable step integrators from the difficulties associated with sudden rate changes.

STANDARD COMPONENT DR

This standard component simulates the "DART" stabilizing device that can be used by an ejection seat to correct for adverse pitch and roll induced by aerodynamic torques and the offset caused by improper alignment of the seat center of gravity with the sustainer rocket thrust vector. It is not effective in providing corrective torques about the yaw axis.

The DART is a simple device which consists of a line that is connected at one end to the airplane, and at the other end to a bridle attached to the bottom of the seat. This line passes through a braking device, whose force is calculated from a table that is an input into the component. This
table, as well as the other input/outputs, are explained in Appendix D. In addition, Figure 25 provides a descriptive diagram for this component.

STANDARD COMPONENT GP

This standard component is a simplified version of component MP (parachute mortar), in that a table look-up is used to find the mortar force as a function of time, instead of the equation of state method employed by MP. The input/output list is given in Appendix D. Due to a configuration where the mortar force vector may not pass through the parachute center of gravity, inputs for both the position of the parachute attachment point and the seat deployment impulse arm are required. In this situation, the force imparted from the gun to the pack is assumed to act parallel to that of the gun impulse vector.

Component GP also has the task of restraining the parachute to the seat prior to mortar initiation. When the mortar is fired, and the chute is propelled away from the seat, the restraint logic prevents the parachute from moving perpendicular to the mortar impulse vector until the mortar reaches stripoff. Mortar stripoff is defined as the time the mortar force reaches zero, which is set in the mortar force input table. When the mortar reaches stripoff, the forces and torques acting on the seat and parachute calculated by the restraint logic are set to zero. However, these forces and torques may be gradually reduced to zero over a time period, defined by input DCE, if desired by the user. This capability was included in the component to prevent the variable step integrators from having difficulty with a sudden rate change.

STANDARD COMPONENT LI

This component calculates the forces and torques that are imparted from a loaded parachute line onto an object that is being decelerated by a parachute. The input/output list is shown in Appendix D. The inputs include the states from both the decelerated object and parachute. Additional
inputs define the bridle configuration and the parachute line characteristics.

The subroutines that are used by component LI include LILOAD, which calculates the line load; LIBRIDL, a routine that determines the force application point; and LILINE, an algorithm that calculates various line parameters. LILOAD is the line model described in reference 2. Subroutine LIBRIDL can accommodate bridles that have one through four attachment points. If there is only one attachment point, the force application point is set equal to the position of attachment point one, and the input defining the bridle apex, namely APX, should be set to zero. Variables calculated in LILINE include the parachute line length, defined as the distance from the stretched canopy center of gravity to the force application point.

STANDARD COMPONENT MP

This module is the EASIEST parachute mortar model, and closely resembles components CT (catapult) and RS (restraints), in that logic similar to that in CT is employed to calculate the force generated by a closed tube telescoping catapult, while the RS logic is used to maintain the parachute's position on the seat until the mortar is initiated. From mortar initiation until stripoff, the restraint logic maintains the parachute on a path that is defined by its seat attachment point and the mortar force vector.

Appendix D gives the inputs and outputs for this component. Inputs include parameters that define the characteristics of the mortar's performance and the spring and damping constants for the restraints. Input TDE is the time interval over which the mortar and restraint forces decay to zero. This input was included to prevent the variable step integrators from having difficulties with sudden rate changes.

STANDARD COMPONENT PC

This module is the EASIEST parachute model. It is capable of modeling either a drag chute or a recovery chute by setting the input data to
correspond to the type of parachute desired. The inputs include variables from components LI and the parachute mortar (GP or MP), as indicated in the input/output descriptions in Appendix D. Additional information concerning the input data is presented in Figure 26.

This component calculates rates for both the parachute pack, defined as the parachute container and the canopy/lines contained within it, and the canopy. Prior to linestretch, the mass of the canopy is set at one pound and driven to the calculated stretched canopy center of gravity by a spring, whose characteristics are defined by input parameters CSP and DPG. After linestretch, the parachute container separates from the canopy, with only the force of gravity acting on it. However, since the container has a coordinate system attached to it, its rotation must be stopped to prevent the Euler angle singularity, an occurrence which reduces execution efficiency when using the variable step integrators. This is accomplished with input DPG, a user defined vector which induces a braking torque about all three axes of the pack's coordinate system. Another input, TEM, is the time duration over which the aerodynamic forces are factored during parachute emergence into the windstream. It also performs a similar function when the lines are severed, ensuring variable step integrator efficiency.

This algorithm is separated into three distinct phases. Phase one is concerned with the parachute dynamics prior to parachute launch. Forces acting on the parachute include the mortar and the restraints. Forces acting on the canopy, which is treated as a separated object, are the spring forces that maintain its position in the pack. Phase two models the parachute from launch to linestretch. Forces that act on the pack include the parachute stripout force and the aerodynamic forces. Forces that are exerted on the canopy are the spring forces that drive the canopy to its center of gravity position along the parachute lines. The center of gravity position is passed to this component from component LI (parachute lines). Phase three takes into account the forces that act on the canopy after linestretch, which include the aerodynamic and line forces, as well as the mass acquisition force as the parachute inflates.
STANDARD COMPONENT RL

This standard component determines the forces and moments that act on the vehicle and the seat while the slider blocks are in contact with the rails. The resulting forces and moments acting on the seat and the vehicle are due to rail elasticity and rail to slider block friction forces. The input/output table is given in Appendix D. Note that states from components SE and the vehicle (AE or SL) are required inputs, and must be accounted for by the component hookups in the Model Generation Program input data. Other inputs include the slider block friction coefficient, and the ejection direction flag. Figure 27 provides an additional explanation for some of the inputs, and helps to explain the rail/slider block geometry.

STANDARD COMPONENT RS

This EASIEST component is the module which restrains one object to another, such as the crewmember to the seat. The input and output data is given in Appendix D. The nomenclature for this component defines the parent body as that object in whose coordinate system the attachment point is defined. The second object is referred to as the attached body. The inputs to this component include the attachment point where the attached body is constrained. The two bodies are held in the relative position defined by the input data by a set of springs which exert both torques and forces on the constrained bodies. The bodies are held together until a switch is set by the sequencer, which is described in Section IV.3.

STANDARD COMPONENT SE

This component sums the forces and torques that act on the seat, and then determines the seat body axis angular and linear rates. The composite seat inertial properties are fed to this component from component WB (weight and balance) if an object is pinned to the seat, as in the case of the sustainer rocket (SR). Otherwise, the inertial properties are inputted directly into the component. Note that the equations of motion were
written so that the linear states apply to the seat reference point rather than the seat center of gravity.

The input/output variables and parameters are given in Appendix D. All pyrotechnic devices, such as the catapult, should have their forces and torques feed into SE via the ports labeled F1 and T1. The forces and torques sent to this component from non-pyrotechnic sources, such as the aerodynamics, should use ports F2 and T2. This constraint is to help the user to organize the inputs into component SE.

STANDARD COMPONENT SL

Component SL is the EASIEST sled model. The linear velocity and position vectors should be initialized in the Analysis Program input data by the INITIAL CONDITIONS command. The angular velocity vector must be initialized to zero, as explained in Section IV.3. Note that the names of the SL states have the same names as those of the airplane, simplifying the process of interchanging the two vehicles in a model file. Appendix D gives a list of the input/output information. Note that the velocity vectors are defined with respect to the sled body axis.

STANDARD COMPONENT SP

This component simulates the STAPAC ejection seat stability device. It consists of a vernier rocket motor connected to a single-degree-of-freedom gyroscope. It can be mounted on the ejection seat to provide a correcting torque for either an adverse yaw, pitch, or roll.

Appendix D supplies the input/output names assigned to this component, while Figure 28 explains the coordinate systems attached to the rocket and the gyroscope. The Euler angles that define the orientation of the rocket and the gyroscope coordinate systems in the seat reference frame are states. Consequently, they must be initialized in the analysis file. Proper initialization can model either a yaw, pitch, or roll STAPAC. Once the gyroscope wheel is spun up and the gimbal uncaged, the seat body axis
angular velocities are projected onto the gimbal axis. If an angular velocity component exists on the input axis of the gyroscope, as shown in Figure 29, the gyro processes, rotating the vernier rock to provide a correcting torque. The forces and torques generated by this rocket are then passed to component SE (seat equations of motion).

Figure 29 provides additional information on the inputs to this component. It explains the biasing effect of the gimbal spring, and what is meant by the thrustline offset. In addition, input TSU specifies a time duration over which the gyroscope wheel accelerates to its uncaged angular velocity. This prevents the variable step integrators from encountering an extreme rate change.

STANDARD COMPONENT SR

The purpose of this module is to calculate the forces and torques that act on the ejection seat from the sustainer rocket. In addition, the inertial properties of the rocket propellant grain are calculated as the rocket burns, and made available to component WB (weight and balance) for the composite seat weight and balance calculation.

Appendix D contains a list of SR input/output descriptions. Figure 30 presents a pictorial explanation of some of these inputs and variables. As shown in the figure, the rocket has a coordinate system attached to the propellant grain center of gravity. In addition, the rocket nozzle has its own coordinate system, with the thrust vector acting along the negative direction of its z-axis. The location of the origin of the propellant grain is with respect to the seat coordinate system, as are its Euler angles. The location of the rocket nozzle's origin and Euler angles are defined with respect to the propellant grain coordinate system. Because the propellant weight is a state, it must be initialized in the analysis file.

During initialization, the specific impulse of the rocket and the initial propellant moments of inertia are calculated. Once the rocket is switched on by the sequencer, the force generated by the rocket is determined by a
table look-up, the propellant consumption rate is calculated, and the moments and products of inertia of the propellant are updated and rotated into the seat coordinate system.

An additional capability of this module includes utilizing it to model an ejection seat with a "thrust vector control" sustainer rocket. This is demonstrated in the model that is presented in Appendix N.

STANDARD COMPONENT WB

This EASIEST component determines the composite center of gravity and inertial properties of the ejection seat. The sustainer rocket propellant is included in this calculation, but ejection seat components which utilize springs to couple themselves to the seat are excluded. This component can accommodate up to three attached bodies.

The input/output information for WB is given in Appendix D. The inputs include the number of attached bodies, the seat body axis position vector of the basic seat center of gravity, the basic seat moments and products of inertia about the seat center of gravity, and the basic seat weight. In addition, the seat system location of each attached body center of gravity is a required input, along with its weight, and the moments and products of inertia rotated into the seat system. The outputs include the following composite seat properties:

a. Weight
b. Center of gravity in the seat body axis system
c. Moments of inertia about the seat center of gravity
d. Products of inertia about the seat center of gravity

These outputs are passed to component SE to be utilized by the seat equations of motion.
2. SUBROUTINES

The EASIEST subroutines (not standard components) listed in Appendix H that are utilized by the EASIEST standard components are available to the analyst for system modeling, and can be used with the FORTRAN STATEMENTS command. Additional subroutines, whose listings are available in Volume II, Section III, of this document, can also be used in system modeling.

3. MODELING WITH THE EASIEST COMPONENTS

This section covers modeling requirements and methods which must be satisfied when an analyst models an escape system with the EASIEST components. It also will help to explain how to resolve certain problems that may be encountered.

Any of the EASIEST components may be employed as often as required in system modeling. However, component AG (atmospheric properties) must be included in all EASIEST models, since it controls a common statement variable used by some EASIEST components, and supplies atmospheric information to PC (parachute), AS (seat aerodynamics), CE (crewperson), and AE (airplane).

A specific sequence of analysis commands should be followed to properly define input parameters and to initialize the model. The analysis file for the examples in Section VI and Appendix N demonstrate this procedure, and it is listed as follows:

(1) TABLE - allows for the input of a required table.

(2) PARAMETER VALUES - precedes the defining of parameter values.

(3) INITIAL CONDITIONS - permits the initialization of state variables (seat velocity, for example).

(4) CALC XIC - allows for the calculation of variables derived from input parameters (the sustainer rocket's specific impulse, for
example). Parameters not defined after the PARAMETER VALUES command are set equal to their default values.

5) INT CONTROLS - freeze the required states prior to the issuance of the STEADY STATE command.

6) STEADY STATE - drives all objects attached to the seat by the restraint components to their attachment position, and determines their velocities.

7) XIC-X - transfers the states calculated by the steady state solver into the initial conditions vector.

8) ALL STATES or INT CONTROLS - specifies which states will be used by the subsequent analysis commands.

9) Desired analysis commands (SIMULATE, LINEAR ANALYSIS, etc.).

A trim scheme has been devised to initialize the states of the physical objects attached to the ejection seat by the restraint components, such as the crewperson and the parachutes. If the sled or airplane is used in the component, the only states that need to be initialized are the vehicle's linear velocity, angular position, and the linear position vectors. (Note: The angular velocity vector of the vehicle must be set to zero, since the steady state scheme cannot accommodate non-zero angular velocities.) After the CALC XIC command is given, all of the vehicle's states are then frozen by the INT CONTROLS command. Model states that are not directly associated with the dynamics of physical objects must also be frozen. These include the states associated with the catapult (CT), mortar (MP), parachute lines (LI), sustainer rocket (SR), and STAPAC (SP). If any of these states are not frozen, the EASY steady state solver will not be able to solve for a steady state, and the command will terminate.
The user must be aware that the STEADY STATE command can calculate an undesired steady state, with the seat driven to an attitude where the plane formed by the slider blocks is perpendicular to the rails. An inverted steady state is also possible. This situation can easily be avoided by initializing the states of the seat as near to their steady state operating point as possible.

Another method to assist the steady state solver is to set the value of parameter SW in component AG to 0.0 before issuing the STEADY STATE command. This prevents the parent objects in the model from "seeing" the forces and torques applied to them by the restraint components. For example, the seat component (SE) will receive rail and catapult forces and torques, but will not receive the forces and torques from the components which restrain the crewmember and the parachutes to it. Likewise, the crewmember will receive forces and torques from the restraints which hold him in the seat, but will not see any forces or torques from anything attached to him. Once a steady state has been calculated with SW AG set to zero, SW AG must be redefined to a value of 1.0, an XIC-X command given, and then the STEADY STATE command repeated. This scheme has been included in the model only as an additional capability, and as a rule it does not have to be implemented.

If an analyst desires to determine the steady state of a seat in a model where there is no vehicle (i.e., the seat is unsupported), then the user must perform the following tasks within the previously described command sequence:

1. Freeze all of the states in component SE.
2. Define SW AG to be equal to zero. (Not required if there are no objects attached to the seat.)
3. Set TM SE equal to the desired earth frame linear trim velocity.
4. Issue the STEADY STATE command.
(5) Redefine SW AG to be equal to one. (Not required if there are no objects attached to the seat.)

The parameter SW AG, when set to 0.0, has the additional capability of setting the acceleration of gravity to zero throughout the model. If the acceleration of gravity is not set to zero before issuing the STEADY STATE command when the seat is unsupported, the restraint springs would have to load up to resist the acceleration of gravity. After unfreezing the seat states, the model would no longer be at a steady state operating point.

The implementation of the airplane component requires a slightly different procedure. The basic sequence of simulation commands outlined earlier in this section should be adhered to; however, prior to the STEADY STATE command, the only airplane states that need to be frozen are EAPAE(1), XAPAE(1), and XAPAE(2). In addition, the states associated with the control surface component (CS) must be frozen if it is employed in the model. The earth system trim velocity and altitude are required inputs into component AE, and should be set to the desired values. Appendix N contains an example of an EASY model that employs the AE component. The aforementioned method of assisting the EASY steady state solver with SW AG is also demonstrated in this example.

When component AE is included in a model, the airplane aerodynamic coefficients must be made available to it. The procedure used to submit an EASIEST run that includes component AE is explained in Section V. An example of a set of coefficients formatted for component AE is given in Appendix J.

When employing any of the restraint components (namely, RL, RS, GP, or MP) in a model, the spring and damping constants associated with them must be defined in such a manner as to set the system's natural frequencies below approximately 1000, and the damping ratios between 0.6 and 0.9. The recommended approach to do this is to first set the angular and linear spring terms according to the magnitude of the attached object's inertial properties when compared to those of its parent object. In other words, a
crewperson attached to the seat must have larger spring constants than, let's say, a parachute that is also mounted on the seat, since the crewperson has the larger inertial properties of the two objects. The user should ensure that the spring terms are large enough so that very little deflection is required to impart the force required to drive the attached object along with the escape system.

The example given in Section VI can be used as a basis for establishing the appropriate spring and damping constants. The next step is to execute the analysis program through a STEADY STATE task, and then investigate the natural frequencies and damping ratios to ensure they are within tolerances. If they are not, the integrator could have difficulties with the system during a simulation. Therefore, the damping constants and spring terms should be manipulated until reasonable results are obtained. Due to the nature of a complex model, such as the one shown in Section VI, there could be some low damping ratios that are very difficult to eliminate. Note that both components CE and SE contain the human spine model, whose 0.2240 damping ratio cannot be manipulated by the user.

After a simulation is made with a variable step integrator, a time step limitation count is printed for each model state. A time step limitation occurs when the integrator encounters an extreme rate change. If this should happen, the integrator reduces its timestep and performs a recovery process to ensure simulation accuracy. However, this can significantly increase the central processor time required for the simulation. Consequently, many of the EASIEST components have schemes to prevent large changes in rates. For example, the catapult force can be decayed over a time period specified by the user, instead of abruptly being set to zero at stripoff. Specific information on components which have this capability is presented in Section IV.1.

The approach the analyst takes to construct a complex system model can influence the amount of time it requires. Perhaps the most efficient method is to assemble the model a few components at a time, modifying the
model and analysis input files during each design iteration to accommodate the components being added. As an example, the user could construct a model using only the sled, rails, and seat components. Once this small model is verified by the various analysis capabilities available in the EASY program, a crewperson component could then be added, and the checkout process repeated. This approach lends itself to correcting problems as they occur, as well as building better designed models.

4. SEQUENCING AN EASIEST MODEL

During the operation of an ejection seat, a variety of discrete events occur that mark transition points in the ejection sequence. Examples include the ignition of a rocket, the burnout of a rocket, and the separation of one object from another. Each such event occurs when either some timing device within the ejection seat triggers it, an event that occurred in some other part of the seat caused a physical switch to be thrown which triggers it, or the event is defined by the physical status of all or part of the ejection seat and that status has been attained. For example, the seat leaving the guide rails can trigger the sustainer rocket ignition, or the deployment of a parachute can trigger the sustainer rocket ignition, or the deployment of a parachute may be triggered by time in one type of seat design, or by seat speed and/or altitude in another.

In order to allow the EASIEST program to be used in modeling many types of ejection seats, a flexible system has been developed for simulating this event triggering. The fundamental elements of this system are:

a. If an event occurs in one component and knowledge of this occurrence is required by other components, the component in which the event occurs is provided with an output which is:

(1) Set equal to zero if the event has not yet occurred (or in some cases, has occurred but is no longer occurring)
(2) Set equal to one if the event has occurred (or in some cases is now occurring)

This type of flag is called an "event triggered flag."

b. If an event inside a component is triggered by something outside the component (including time), then that component has been provided with an input which must be:

(1) Set equal to zero if the event is not to begin (or, if occurring, should stop)
SECTION V

PROCEDURES FOR INSTALLING AND SUBMITTING
AN EASIEST RUN

The purpose of this section is to explain the EASY5 installation procedure on the ASD computer, and how to submit an EASIEST run.

1. INSTALLING THE EASY5 PROGRAM

The source code for the EASY5 computer program and the EASIEST standard components was delivered to AFWAL/FIER on tape LØ2377. This tape contains 17 files in the following order (the volume and section where the corresponding file listing resides is given in parenthesis where appropriate):

1. EZSTPRC - EASIEST procedure file (Volume I, Appendix F)

2. BACOMPS - Source for the EASY5 standard components (not EASIEST), associated subroutines and functions (Volume II, Section 11.5)

3. COMPASS - Assembly Language Utility Program (Volume II, Section 11.5)

4. EZSTFTN - Source for the EASIEST standard components, associated subroutines and functions (Volume I, Appendices G and H)

5. FILOADS - Source for FILOAD. (Volume II, Section 11.4)

6. FILDAT - Input data for FILOAD (Volume I, Appendix I)
7. **EASYS** - Source for the Model Generation program (Volume 2, Section II.1)

8. **EASY5** - Relocatables for the Model Generation Program

9. **NONSIMS** - Source for the Analysis Program (Volume 2, Section II.2)

10. **NONSIM5** - Relocatables for the Analysis Program

11. **NSMPPTS** - Source for the Printer Plot Program (Volume 2, Section III.3)

12. **AEROMED** - Source for the EASIEST Aeromedical post processor (Volume 1, Appendix E)

13. **F4EMAN** - EASIEST F-4E aerodynamic maneuvering coefficients (Volume 1, Appendix J)

14. **MCORR** - Model description for the example in Volume 1, Section VI

15. **ACORR** - Analysis file for the example in Volume 1, Section VI

16. **MODAPP** - Model description for the example in Volume 1, Appendix N

17. **ANALAPP** - Analysis file for the example in Volume 1, Appendix N.

To execute the procedure to install the entire EASY5/EASIEST package from the delivery tape, route the following deck to the ASD computer input queue after instructing the tape library to mount tape number L02377:
This procedure will unload the delivery tape, compile the source programs, and catalog all necessary files. In addition, a sample EASIEST run will be submitted, using the same model description and analysis files as the example in Volume I, Section VI.

2. PROCEDURE FOR SUBMITTING AN EASIEST RUN

The following method provides a simple procedure for submitting an EASIEST run into the batch input queue of the ASD computer:

1. Prepare the EASIEST model description file and the EASIEST analysis file as described in the previous section. These files should be stored on your account as permanent files.

2. From an ASD Intercom terminal, which is in the command mode, attach the EASIEST Procedure file using the following command:

```
ATTACH,EZSTPRC.
```

To perform correctly, this file must be attached with local file name EZSTPRC.

3. To initiate the procedure, type:

```
BEGIN, SUBRUN,EZSTPRC,mfname,afname,TIME=t,INOUT=i,CORE=c, COEF=j,NOLIST,AEROMED..
```

where:

a. "mfile" is the name of the permanent file containing your model description (this entry is required),
b. "afname" is the name of the permanent file containing your analysis data (this entry is required),
c. "t" is the cpu time in seconds to be allocated for this run (this entry is required only if you wish the allocation to differ from the default of t=100),
d. "i" is the input-output time in seconds to be allocated for this run (this entry can occur anywhere in the BEGIN statement after afname and is required only if you wish the allocation to differ from the default of i=100),
e. "c" is the cpu core space in octal to be allocated for this run (this entry can occur anywhere after afname and is only required is you wish the allocation to differ from the default of c=115000),
f. "j" is the name of the permanent file which contains the aerodynamic coefficients for the EASIEST airplane. If the airplane is not included in the model, "COEF=j" should not be entered.
g. If entered, "NOLIST" deletes the FTN listing from the SUBRUN procedure. Do not include this entry if you wish the FTN listing to be written to output.
h. If specified, "AEROMED" executes the aeromedical post-processor. To suppress execution, do not include this entry.
SECTION VI
EJECTION SEAT ANALYSIS EXAMPLE

This section presents an example of an ejection seat simulation for a model that was assembled using the EASIEST components. All of the EASIEST components were employed in this model, with the exception of AE (Airplane), CS (Airplane Control Surfaces), DR (DART), and AP (Aerodynamic Plate). The implementation of these four components into a model is demonstrated in Appendix N, which also includes a thrust vector control system that was added by using the FORTRAN STATEMENTS command.

Figure 5 presents the model description file used to define the escape system model. The instructions on how this file was assembled are given in Section II. Figure 6 shows the flow chart that was constructed by the Model Generation Program from the instructions contained in the model file. Figure 7 contains the analysis file that was used to define the input tables and parameters. It also initializes the states, and contains the commands that dictate how the model is to be analyzed. An explanation of the commands used by this file is presented in Section III. Figures 8 and 9 show the respective outputs of the steady state analysis and the simulation analysis. Printer plots are shown in Figure 10.
Figure 5. Model Generation Program Input File
Figure 6. (Continued)
Figure 7. Analysis Program Input File
Figure 7. (Continued)
Figure 7. (Continued)
PT MP=10  C BMP=8.51E-4
CM MP=0.0001  PWM=35.9  GAMMP=1.199  TF MP=2800
CM MP=130  C MP=380  B MP=0.035  B1FMP=0.46
T1 MP=294  TDMP=0.03

- ----- PARAMETERS FOR COMPONENT "PCDC" (DRAG CHUTE) -----

STIFPCDC=10.57  BSCPCDC=19.63  RMPPCDC=0  RDPFPCDC=0
BSPFPCDC=0  BPCDC=0  CTPCDC=0.548 -.24 4.64 4.69
CMPCDC=-31.2 -211 -41.97
CFFPCDC=0.64 0.2  PMPFPCDC=8.1
PMFPCDC=0.034 0.05 0.014  PPFFPCDC=Z
SMPFPCDC=0.03  CSPFPCDC=2000  CDPFPCDC=14
DPPFPCDC=2.32

- ----- PARAMETERS FOR COMPONENT "LDIC" (DRAG CHUTE LINES) -----

DCTIME=0.15
BLDLDIC=2  APFLVIC=-3.84 0.1 0.04
APFLDC=-32.51 1.1 0.04  APFLDCC=-32.51 1.1 0.04
FSLLDIC=10  ULLLDIC=2000  UL LDIC=0.4
GOLLDIC=10  FIRLLDIC=0.5  T PFLDIC=1

- ----- PARAMETERS FOR COMPONENT "OP" (DRAG CHUTE MORTAR) -----

OFTIME=100
UVP=0.0 0.342 -307 1  SIMQP=0.45 1.206 1.345
ZTQP=0.33 0.33 0.7  EAP=Z
ER GP=3500  ED GP=3500  ED GP=3500  ED GP=3500
IDQ=0

- ----- PARAMETERS FOR COMPONENT "AG" (ATMOSPHERE AND GRAVITY) -----

AGA=1.25  WMAAG=Z
DF AG=1.2576  TE AG=72.8  AG=0

- ----- PARAMETERS FOR COMPONENT "AM" (AEROMEDICAL) -----

FL AM=1.

- ----- INITIALIZATION -----

INITIAL CONDITIONS=WAPSZ=21.0 0.0 ZAPSZ=0.0 0.0
W1 SR=5.9  WAPSZ=1418.0 520 USTSE=8615.50 177.2
ESTSE=0.12 5.6 WSTSE=2
EGSP=12.38 0.0
EARS=14.0 0.0

CALC ZIC
INT CONTROLS XAPSZ=Z  WAPSZ=Z  EAPSZ=Z  WAPSZ=Z
EF CTGZ  CTGZ  CTGZ  CTGZ
EF MPZ  MPZ  MPZ  MPZ
W1 SR=0.0  ECI LCDC=0.7  LCDC=0
EC LCDC=0.7  LCDC=0
EGSP=Z  EARS=Z  W0 SP=0

SS iterations=80
STEADY STATE
ZIC=0
PARAMETER VALUES
DPQRL=0.0
ALL STATES

- ----- SIMULATION -----

DEFINE RATES R43=UDS1

Figure 7. (Continued)
Figure 7. (Continued)
### Steady State Analysis

A maximum of 60 iterations can be used.

**Time = 0.**

<table>
<thead>
<tr>
<th>States</th>
<th>Case No. 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 EF CT = 0</td>
<td>4 WB CT = 0</td>
</tr>
<tr>
<td>2 EL CT = 0</td>
<td>5 WK CT = 0</td>
</tr>
<tr>
<td>UAPSL (1) = 0</td>
<td>UAPSL (1) = 0</td>
</tr>
<tr>
<td>EAPSL (1) = 0</td>
<td>EAPSL (1) = 0</td>
</tr>
<tr>
<td>EAPSL (1) = 0</td>
<td>EAPSL (1) = 0</td>
</tr>
<tr>
<td>ESOSP (1) = 0</td>
<td>ESOSP (1) = 0</td>
</tr>
<tr>
<td>ESOSP (1) = 0</td>
<td>ESOSP (1) = 0</td>
</tr>
<tr>
<td>19 UF MP = 0</td>
<td>20 EL MP = 0</td>
</tr>
<tr>
<td>21 UF LIRC = 0</td>
<td>20 UF LIRC = 0</td>
</tr>
<tr>
<td>UPPCRC (1) = 0</td>
<td>UPPCRC (1) = 0</td>
</tr>
<tr>
<td>UPPCRC (1) = 0</td>
<td>UPPCRC (1) = 0</td>
</tr>
<tr>
<td>EPPCRC (1) = 0.0</td>
<td>EPPCRC (1) = 0.0</td>
</tr>
<tr>
<td>UPPCRC (1) = 0.0</td>
<td>UPPCRC (1) = 0.0</td>
</tr>
<tr>
<td>EPPCRC (1) = 0.0</td>
<td>EPPCRC (1) = 0.0</td>
</tr>
<tr>
<td>10 EC UOC = 0</td>
<td>50 SC UOC = 0</td>
</tr>
<tr>
<td>80 SC LIRC = 0</td>
<td>10 SC LIRC = 0</td>
</tr>
<tr>
<td>UOABAT (1) = 0</td>
<td>UOABAT (1) = 0</td>
</tr>
<tr>
<td>XABAT (1) = 0</td>
<td>XABAT (1) = 0</td>
</tr>
<tr>
<td>XABAT (1) = 0</td>
<td>XABAT (1) = 0</td>
</tr>
<tr>
<td>EABAT (1) = 0</td>
<td>EABAT (1) = 0</td>
</tr>
<tr>
<td>EABAT (1) = 0</td>
<td>EABAT (1) = 0</td>
</tr>
<tr>
<td>USTE (1) = 0</td>
<td>USTE (1) = 0</td>
</tr>
<tr>
<td>XSTE (1) = 0</td>
<td>XSTE (1) = 0</td>
</tr>
<tr>
<td>XSTE (1) = 0</td>
<td>XSTE (1) = 0</td>
</tr>
<tr>
<td>ESTE (1) = 0</td>
<td>ESTE (1) = 0</td>
</tr>
<tr>
<td>ESTE (1) = 0</td>
<td>ESTE (1) = 0</td>
</tr>
<tr>
<td>107 SCUDE</td>
<td>107 SCUDE</td>
</tr>
</tbody>
</table>

**Rates**

<table>
<thead>
<tr>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 8. Steady State Output**
Figure 8. (Continued)
<table>
<thead>
<tr>
<th>95 SFLAG = 0.</th>
<th>96 PHASP = 0.</th>
<th>F1ISPF (2) = 0.</th>
<th>104 FISP (3) = 0.</th>
<th>105 MFL = 0.</th>
<th>126 CPRC (2) = 0.</th>
<th>127 TRMP = 0.</th>
</tr>
</thead>
<tbody>
<tr>
<td>103 IFISP = 0.</td>
<td>104 CEISP = 0.</td>
<td>F1ISPF (2) = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (3) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (4) = 0.</td>
</tr>
<tr>
<td>F1IMP = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (5) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (6) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>F1IMP = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (7) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (8) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1104 FIMP = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (9) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (10) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1105 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (11) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (12) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1106 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (13) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (14) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1107 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (15) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (16) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1108 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (17) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (18) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1109 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (19) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (20) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1110 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (21) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (22) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1111 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (23) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (24) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1112 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (25) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (26) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1113 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (27) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (28) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1114 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (29) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (30) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
<tr>
<td>1115 MFL = 0.</td>
<td>104 ECISP = 0.</td>
<td>105 MFL = 0.</td>
<td>126 CPRC (31) = 0.</td>
<td>127 TRMP = 0.</td>
<td>128 CPRC (32) = 0.</td>
<td>129 TRMP = 0.</td>
</tr>
</tbody>
</table>

**Figure 8. (Continued)**
000

W

o: -0:

00CW2

o

00

*

0.

ooo

go:

co

to

:

."o

.. W

am

WQU

. .

0

.

0

E X..

.

0

QQ=

9.
0

0

0
o

u

00

=a

o0

o

am

0

0

0

O

0

0

00

0

m

-uc

z~w.

-

o0oo

.

.flS

oaO
oococaanf

I

-

NN

c-

2

oo

ooo0

o

-

. . . .

0

am

o

. . . . . ..

-...............
.
•..
.

0

.
. . ...
. . . . . . . .f. .
..........
- - - - - - - - - -- - - - - - - - - 2.0
-

4i

o..

0

0

.

.

0..
122

LLL

0

0

00.000


<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>-57,3168</td>
<td>+80,0001</td>
<td>98,2963</td>
<td>50159</td>
</tr>
<tr>
<td>13</td>
<td>-58,3277</td>
<td>+80,2717</td>
<td>98,4678</td>
<td>50159</td>
</tr>
<tr>
<td>14</td>
<td>-58,4818</td>
<td>+71,6005</td>
<td>98,8462</td>
<td>497253</td>
</tr>
<tr>
<td>15</td>
<td>-68,5246</td>
<td>+71,0360</td>
<td>102,3056</td>
<td>655316</td>
</tr>
<tr>
<td>16</td>
<td>-68,9418</td>
<td>+76,2598</td>
<td>102,2829</td>
<td>655316</td>
</tr>
<tr>
<td>17</td>
<td>-83,2050</td>
<td>+80,6098</td>
<td>97,3350</td>
<td>655423</td>
</tr>
<tr>
<td>18</td>
<td>-91,2906</td>
<td>+256,0081</td>
<td>294,1557</td>
<td>655423</td>
</tr>
<tr>
<td>19</td>
<td>-99,1337</td>
<td>+44,2746</td>
<td>107,9000</td>
<td>655423</td>
</tr>
<tr>
<td>20</td>
<td>-101,3393</td>
<td>+48,2746</td>
<td>107,9000</td>
<td>655423</td>
</tr>
<tr>
<td>21</td>
<td>-102,1334</td>
<td>+242,8153</td>
<td>557,9910</td>
<td>264060</td>
</tr>
<tr>
<td>22</td>
<td>-105,1206</td>
<td>+257,9500</td>
<td>225,9239</td>
<td>455369</td>
</tr>
<tr>
<td>23</td>
<td>-120,2727</td>
<td>+377,3325</td>
<td>394,1778</td>
<td>204731</td>
</tr>
<tr>
<td>24</td>
<td>-126,8266</td>
<td>+295,8755</td>
<td>620102</td>
<td>455407</td>
</tr>
<tr>
<td>25</td>
<td>-135,2948</td>
<td>+108,8901</td>
<td>241,6600</td>
<td>455407</td>
</tr>
<tr>
<td>26</td>
<td>-156,2156</td>
<td>+110,7385</td>
<td>252,6668</td>
<td>887442</td>
</tr>
<tr>
<td>27</td>
<td>-156,2156</td>
<td>+110,7385</td>
<td>252,6668</td>
<td>887442</td>
</tr>
<tr>
<td>28</td>
<td>-194,9803</td>
<td>+91,5644</td>
<td>451,8465</td>
<td>887442</td>
</tr>
<tr>
<td>29</td>
<td>-229,2807</td>
<td>+76,9873</td>
<td>377,1564</td>
<td>887442</td>
</tr>
<tr>
<td>30</td>
<td>-459,3034</td>
<td>+633,6132</td>
<td>611,4444</td>
<td>499029</td>
</tr>
<tr>
<td>31</td>
<td>-566,8003</td>
<td>+718,2809</td>
<td>920,9274</td>
<td>761212</td>
</tr>
<tr>
<td>32</td>
<td>-936,2907</td>
<td>+1092,97</td>
<td>1424,82</td>
<td>642030</td>
</tr>
</tbody>
</table>

**Figure 8.** (Continued)
## Simulation Analysis

<table>
<thead>
<tr>
<th>CASE NO.</th>
<th>80/12/17</th>
<th>03/24/03</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>.0000E+01</th>
<th>.0000E+01</th>
<th>.2000E-01</th>
<th>.3000E-01</th>
<th>.4000E-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDST (1)</td>
<td>3.4681</td>
<td>.48635</td>
<td>.45406</td>
<td>.43506</td>
<td>.45106</td>
</tr>
<tr>
<td>UDST (2)</td>
<td>.45200E-02</td>
<td>.178.25</td>
<td>.52497</td>
<td>.180.00</td>
<td>.24989E-02</td>
</tr>
<tr>
<td>UDST (3)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (4)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (5)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (6)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (7)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (8)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (9)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (10)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (11)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (12)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (13)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (14)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (15)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (16)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (17)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (18)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (19)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
<tr>
<td>UDST (20)</td>
<td>.32448E-02</td>
<td>.190.06</td>
<td>.180.00</td>
<td>.190.06</td>
<td>.190.06</td>
</tr>
</tbody>
</table>

**Figure 9. Simulation Output**
<table>
<thead>
<tr>
<th>CF CT</th>
<th>2293.1</th>
<th>UCPC (1)</th>
<th>803.82</th>
<th>UCPC (2)</th>
<th>47331E-01</th>
<th>UCPC (3)</th>
<th>156.48</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICPC (1)</td>
<td>11.481</td>
<td>UCPC (2)</td>
<td>20644E-01</td>
<td>ICPC (2)</td>
<td>80.845</td>
<td>ICPC (3)</td>
<td>16433E-01</td>
</tr>
<tr>
<td>ICPC (3)</td>
<td>2.2057</td>
<td>XPCPCDC (1)</td>
<td>81.758</td>
<td>XPCPCDC (2)</td>
<td>-32596</td>
<td>XPCPCDC (3)</td>
<td>31253</td>
</tr>
</tbody>
</table>

**Time:** 0.1000

<table>
<thead>
<tr>
<th>UDST (1)</th>
<th>-5.2800</th>
<th>UDST (2)</th>
<th>16854E-02</th>
<th>UDST (3)</th>
<th>-230.58</th>
<th>UDST (4)</th>
<th>803.47</th>
</tr>
</thead>
<tbody>
<tr>
<td>USTSE (2)</td>
<td>.64898E-01</td>
<td>USTSE (3)</td>
<td>153.14</td>
<td>SNRPSE (1)</td>
<td>90.794</td>
<td>SNRPSE (2)</td>
<td>84013E-04</td>
</tr>
<tr>
<td>SNRPSE (3)</td>
<td>-5.928E-01</td>
<td>SNRPSE (4)</td>
<td>154.54</td>
<td>WNSTSE (1)</td>
<td>21.896</td>
<td>WNSTSE (2)</td>
<td>87343E-03</td>
</tr>
<tr>
<td>WNSTSE (2)</td>
<td>-1.4580</td>
<td>WNSTSE (3)</td>
<td>-5.875E-01</td>
<td>WNSTSE (4)</td>
<td>8.9942</td>
<td>EADAM</td>
<td>-34093E-01</td>
</tr>
<tr>
<td>CF CT</td>
<td>3203.1</td>
<td>UCPC (1)</td>
<td>803.02</td>
<td>UCPC (2)</td>
<td>34093E-01</td>
<td>UCPC (3)</td>
<td>153.39</td>
</tr>
<tr>
<td>ICPC (1)</td>
<td>80.302</td>
<td>ICPC (2)</td>
<td>16856E-01</td>
<td>ICPC (3)</td>
<td>-91947</td>
<td>ICPC (4)</td>
<td>91911E-03</td>
</tr>
<tr>
<td>ICPC (3)</td>
<td>11.481</td>
<td>ICPC (4)</td>
<td>20644E-01</td>
<td>ICPC (5)</td>
<td>-32596</td>
<td>XPCPCDC (1)</td>
<td>31253</td>
</tr>
</tbody>
</table>

**Time:** 0.1500

<table>
<thead>
<tr>
<th>UDST (1)</th>
<th>-11.1005</th>
<th>UDST (2)</th>
<th>1.3700</th>
<th>UDST (3)</th>
<th>-520.18</th>
<th>UDST (4)</th>
<th>803.28</th>
</tr>
</thead>
<tbody>
<tr>
<td>USTSE (2)</td>
<td>.64898E-01</td>
<td>USTSE (3)</td>
<td>153.14</td>
<td>SNRPSE (1)</td>
<td>90.794</td>
<td>SNRPSE (2)</td>
<td>84013E-04</td>
</tr>
<tr>
<td>SNRPSE (3)</td>
<td>-5.928E-01</td>
<td>SNRPSE (4)</td>
<td>154.54</td>
<td>WNSTSE (1)</td>
<td>21.896</td>
<td>WNSTSE (2)</td>
<td>87343E-03</td>
</tr>
<tr>
<td>WNSTSE (2)</td>
<td>-1.4580</td>
<td>WNSTSE (3)</td>
<td>-5.875E-01</td>
<td>WNSTSE (4)</td>
<td>8.9942</td>
<td>EADAM</td>
<td>-34093E-01</td>
</tr>
<tr>
<td>CF CT</td>
<td>3203.1</td>
<td>UCPC (1)</td>
<td>803.02</td>
<td>UCPC (2)</td>
<td>34093E-01</td>
<td>UCPC (3)</td>
<td>153.39</td>
</tr>
<tr>
<td>ICPC (1)</td>
<td>80.302</td>
<td>ICPC (2)</td>
<td>16856E-01</td>
<td>ICPC (3)</td>
<td>-91947</td>
<td>ICPC (4)</td>
<td>91911E-03</td>
</tr>
<tr>
<td>ICPC (3)</td>
<td>11.481</td>
<td>ICPC (4)</td>
<td>20644E-01</td>
<td>ICPC (5)</td>
<td>-32596</td>
<td>XPCPCDC (1)</td>
<td>31253</td>
</tr>
</tbody>
</table>

**Time:** 0.2000

<table>
<thead>
<tr>
<th>UDST (1)</th>
<th>-13.1800</th>
<th>UDST (2)</th>
<th>-7.718E-01</th>
<th>UDST (3)</th>
<th>-261.58</th>
<th>UDST (4)</th>
<th>803.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>USTSE (2)</td>
<td>.71890E-01</td>
<td>USTSE (3)</td>
<td>166.54</td>
<td>SNRPSE (1)</td>
<td>106.12</td>
<td>SNRPSE (2)</td>
<td>82755E-04</td>
</tr>
<tr>
<td>SNRPSE (3)</td>
<td>-1.0000E-01</td>
<td>SNRPSE (4)</td>
<td>185.68</td>
<td>WNSTSE (1)</td>
<td>115.46</td>
<td>WNSTSE (2)</td>
<td>87343E-03</td>
</tr>
<tr>
<td>WNSTSE (2)</td>
<td>-1.1000E-01</td>
<td>WNSTSE (3)</td>
<td>-24447E-01</td>
<td>WNSTSE (4)</td>
<td>30485E-02</td>
<td>EADAM</td>
<td>-34093E-01</td>
</tr>
<tr>
<td>CF CT</td>
<td>3202.3</td>
<td>UCPC (1)</td>
<td>803.25</td>
<td>UCPC (2)</td>
<td>34093E-01</td>
<td>UCPC (3)</td>
<td>153.39</td>
</tr>
<tr>
<td>ICPC (1)</td>
<td>80.302</td>
<td>ICPC (2)</td>
<td>16856E-01</td>
<td>ICPC (3)</td>
<td>-91947</td>
<td>ICPC (4)</td>
<td>91911E-03</td>
</tr>
<tr>
<td>ICPC (3)</td>
<td>11.481</td>
<td>ICPC (4)</td>
<td>20644E-01</td>
<td>ICPC (5)</td>
<td>-32596</td>
<td>XPCPCDC (1)</td>
<td>31253</td>
</tr>
</tbody>
</table>

**Time:** 0.2500

<table>
<thead>
<tr>
<th>UDST (1)</th>
<th>-7.1138</th>
<th>UDST (2)</th>
<th>94732</th>
<th>UDST (3)</th>
<th>-592.04</th>
<th>UDST (4)</th>
<th>803.13</th>
</tr>
</thead>
<tbody>
<tr>
<td>USTSE (2)</td>
<td>.71227E-01</td>
<td>USTSE (3)</td>
<td>164.86</td>
<td>SNRPSE (1)</td>
<td>114.91</td>
<td>SNRPSE (2)</td>
<td>86623E-04</td>
</tr>
<tr>
<td>SNRPSE (3)</td>
<td>-1.0151E-01</td>
<td>SNRPSE (4)</td>
<td>185.18</td>
<td>WNSTSE (1)</td>
<td>225.07</td>
<td>WNSTSE (2)</td>
<td>87343E-03</td>
</tr>
<tr>
<td>WNSTSE (2)</td>
<td>-1.1000E-01</td>
<td>WNSTSE (3)</td>
<td>-24447E-01</td>
<td>WNSTSE (4)</td>
<td>30485E-02</td>
<td>EADAM</td>
<td>-34093E-01</td>
</tr>
<tr>
<td>CF CT</td>
<td>4102.3</td>
<td>UCPC (1)</td>
<td>803.25</td>
<td>UCPC (2)</td>
<td>34093E-01</td>
<td>UCPC (3)</td>
<td>154.45</td>
</tr>
<tr>
<td>ICPC (1)</td>
<td>11.593</td>
<td>ICPC (2)</td>
<td>22489E-01</td>
<td>ICPC (3)</td>
<td>-31390</td>
<td>ICPC (4)</td>
<td>18590E-01</td>
</tr>
<tr>
<td>ICPC (3)</td>
<td>11.481</td>
<td>ICPC (4)</td>
<td>20644E-01</td>
<td>ICPC (5)</td>
<td>-32596</td>
<td>XPCPCDC (1)</td>
<td>31253</td>
</tr>
</tbody>
</table>

**Time:** 0.3000

<table>
<thead>
<tr>
<th>UDST (1)</th>
<th>-21.242</th>
<th>UDST (2)</th>
<th>-99128</th>
<th>UDST (3)</th>
<th>-162.2</th>
<th>UDST (4)</th>
<th>803.04</th>
</tr>
</thead>
<tbody>
<tr>
<td>USTSE (2)</td>
<td>-94938E-01</td>
<td>USTSE (3)</td>
<td>136.95</td>
<td>SNRPSE (1)</td>
<td>123.66</td>
<td>SNRPSE (2)</td>
<td>89292E-04</td>
</tr>
<tr>
<td>SNRPSE (3)</td>
<td>-4990E-01</td>
<td>SNRPSE (4)</td>
<td>153.8</td>
<td>WNSTSE (1)</td>
<td>225.07</td>
<td>WNSTSE (2)</td>
<td>87343E-03</td>
</tr>
<tr>
<td>WNSTSE (2)</td>
<td>-4990E-01</td>
<td>WNSTSE (3)</td>
<td>-13002</td>
<td>WNSTSE (4)</td>
<td>30485E-02</td>
<td>EADAM</td>
<td>-34093E-01</td>
</tr>
<tr>
<td>CF CT</td>
<td>2815.7</td>
<td>UCPC (1)</td>
<td>802.59</td>
<td>UCPC (2)</td>
<td>34093E-01</td>
<td>UCPC (3)</td>
<td>150.39</td>
</tr>
<tr>
<td>ICPC (1)</td>
<td>122.97</td>
<td>ICPC (2)</td>
<td>20644E-01</td>
<td>ICPC (3)</td>
<td>-14928</td>
<td>ICPC (4)</td>
<td>32782E-02</td>
</tr>
<tr>
<td>ICPC (3)</td>
<td>11.481</td>
<td>ICPC (4)</td>
<td>20644E-01</td>
<td>ICPC (5)</td>
<td>-32596</td>
<td>XPCPCDC (1)</td>
<td>31253</td>
</tr>
<tr>
<td>ICPC (5)</td>
<td>2.2405</td>
<td>XPCPCDC (1)</td>
<td>81.758</td>
<td>XPCPCDC (2)</td>
<td>-32596</td>
<td>XPCPCDC (3)</td>
<td>31253</td>
</tr>
</tbody>
</table>

**Figure 9. (Continued)**
Figure 9. (Continued)
Figure 9. (Continued)
<table>
<thead>
<tr>
<th>Time</th>
<th>UST</th>
<th>UST (2)</th>
<th>UST (3)</th>
<th>UST (4)</th>
<th>VSPE (1)</th>
<th>VSPE (2)</th>
<th>VSPE (3)</th>
<th>VSPE (4)</th>
<th>ESTE (1)</th>
<th>ESTE (2)</th>
<th>ESTE (3)</th>
<th>ESTE (4)</th>
<th>UCSPE (1)</th>
<th>UCSPE (2)</th>
<th>UCSPE (3)</th>
<th>UCSPE (4)</th>
<th>XPCPCRC (1)</th>
<th>XPCPCRC (2)</th>
<th>XPCPCRC (3)</th>
<th>XPCPCRC (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5000</td>
<td>-2393.6</td>
<td>-7813.6</td>
<td>-1060.6</td>
<td>-1832.4</td>
<td>309.40</td>
<td>232.36</td>
<td>320.54</td>
<td>317.64</td>
<td>-2.0808</td>
<td>-2.1466</td>
<td>2.3957</td>
<td>-1552.3</td>
<td>538.56</td>
<td>1552.3</td>
<td>-1514.4</td>
<td>-1570.3</td>
<td>54463-01</td>
<td>1552.3</td>
<td>-1552.3</td>
<td>-1514.4</td>
</tr>
<tr>
<td>.4350</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.5175</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.5870</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Pitch StopARC Burnout at Time:** .5175 sec

**Sustainer Rocket Off at Time:** .5870 sec

---

**Figure 9. (Continued)**
Figure 9. (Continued)
<table>
<thead>
<tr>
<th>TIME</th>
<th>USST (1)</th>
<th>USST (2)</th>
<th>WST (1)</th>
<th>WST (2)</th>
<th>WPSE (1)</th>
<th>WPSE (2)</th>
<th>ESTSE (1)</th>
<th>ESTSE (2)</th>
<th>CF CT</th>
<th>UC PCCE (1)</th>
<th>UC PCCE (2)</th>
<th>ECPC (1)</th>
<th>ECPC (2)</th>
<th>AP PCPCRC (1)</th>
<th>AP PCPCRC (2)</th>
<th>AP PCPCRC (3)</th>
<th>AP PCPCRC (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.100</td>
<td>-32.907</td>
<td>24.057</td>
<td>-1817.6</td>
<td>212.95</td>
<td>-1646.3</td>
<td>336.36</td>
<td>539.30</td>
<td>864.98</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>212.95</td>
<td>-34.178</td>
<td>1080.1</td>
<td>3305.3</td>
<td>212.95</td>
<td>-1646.3</td>
<td>336.36</td>
<td>539.30</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2.200</td>
<td>336.36</td>
<td>539.30</td>
<td>539.30</td>
<td>539.30</td>
<td>0</td>
<td>539.30</td>
<td>539.30</td>
<td>539.30</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>1198.9</td>
<td>0</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3.200</td>
<td>-1646.3</td>
<td>336.36</td>
<td>336.36</td>
<td>336.36</td>
<td>-279.30</td>
<td>336.36</td>
<td>336.36</td>
<td>336.36</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>212.95</td>
<td>-1646.3</td>
<td>336.36</td>
<td>336.36</td>
<td>-279.30</td>
<td>336.36</td>
<td>336.36</td>
<td>336.36</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4.200</td>
<td>-34.178</td>
<td>212.95</td>
<td>-1817.6</td>
<td>212.95</td>
<td>-1646.3</td>
<td>336.36</td>
<td>539.30</td>
<td>864.98</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>212.95</td>
<td>-34.178</td>
<td>1080.1</td>
<td>3305.3</td>
<td>212.95</td>
<td>-1646.3</td>
<td>336.36</td>
<td>539.30</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5.200</td>
<td>336.36</td>
<td>539.30</td>
<td>539.30</td>
<td>539.30</td>
<td>0</td>
<td>539.30</td>
<td>539.30</td>
<td>539.30</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>1198.9</td>
<td>0</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6.200</td>
<td>-1646.3</td>
<td>336.36</td>
<td>336.36</td>
<td>336.36</td>
<td>-279.30</td>
<td>336.36</td>
<td>336.36</td>
<td>336.36</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>212.95</td>
<td>-1646.3</td>
<td>336.36</td>
<td>336.36</td>
<td>-279.30</td>
<td>336.36</td>
<td>336.36</td>
<td>336.36</td>
<td>-</td>
<td>206.13</td>
<td>234.76</td>
<td>1198.9</td>
<td>336.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Figure 9. (Continued)**
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WSCPE (1)</td>
<td>1.4408</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.1009</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
<tr>
<td>WSCPE (1)</td>
<td>1.3403</td>
<td>WSCPE (2)</td>
<td>1.1964</td>
<td>WSCPE (3)</td>
<td>1.0914</td>
<td>WSCPE (4)</td>
<td>1.8615</td>
</tr>
</tbody>
</table>

**Figure 9. (Continued)**
HUMAN TOLERANCE ANALYSIS THROUGH 3.042 SECONDS OF THE SIMULATION

AERONAUTICAL SIGN CONVENTION......

Gx = +ACCEL, Gy = +ACCEL, Gz = -ACCEL......

Gimax = 7.71 TIME = 3.009
Gymax = 0.25 TIME = 1.246
Gimin = 20.85 TIME = 2.097
Gmin = -31.41 TIME = 0.589
Gymin = -32.03 TIME = 0.610
Gymin = -15.91 TIME = 0.442
Drimax = 24.02 TIME = 0.924
Radmax = 2.55 TIME = 0.910

FIGURES OF MERIT.................

EXPERIENCE FACTOR - TOTAL LOAD = 0.770
EXPERIENCE FACTOR - SAFE LOAD = 0.703
EXPERIENCE FACTOR - UNSAFE LOAD = 0.067

Figure 9. (Continued)
SECTION VII
CREATING AND MODIFYING STANDARD COMPONENTS

EASIEST is a collection of components designed to be used with the EASY Dynamic Analysis System (EASY), and any additions or modifications to the EASIEST components should take this into account. Before we describe the steps required to modify EASIEST, we give a brief introduction to the structure of EASY emphasizing the constraints that the structure puts on components used with it.

EASY consists of the EASY Model Generation Program and the EASY Analysis Program, plus various routines, files, and procedures to maintain and execute these two programs. EASY uses the EASY Model Generation Program to convert the user's model description file into a FORTRAN subroutine called EQMO. Each time a standard component is specified in the user's model a call is generated in EQMO to the subroutine with the same two character name as the standard component name. The EASY Model Generation Program generates these calls using data contained in a random access file called EZSTDBF. This file contains the names and specifications of all inputs, outputs, and tables for each component. EQMO is used by the EASY Analysis Program under direction of the user's analysis file in the following way: given the value of time and the values of all the state variables in the model, compute the rates at which the state variables would be changing at that time. Note that the values of the state variables are not computed by EQMO or its subroutine (including standard component subroutines). The states are computed by the EASY Analysis Program using the rate data provided by EQMO. The rate data is used in different ways during different analyses (simulation, steady state, linear analysis, etc.). Therefore we have:

Constraint #1. The user should make no changes to a standard component subroutine or any subroutine called by a standard component subroutine which results in a value to be assigned to a state variable.
Exception: during the CALC XIC analysis the EASY Analysis Program expects EQMO to compute state variables. Examples of how to set this up can be seen by examining the FORTRAN code of some of the existing standard components.

Each standard component consists of a FORTRAN subroutine in the EASIEST library with the same name as the two character standard component name and three records on a random access permanent file EZSTDBF. If QZ was an EASIEST standard component, then EZSTDBF would contain records called QZINPT, QZOUTP, and QZTABS. QBTABS contains one word for each table used by the component plus one word containing the number of tables. QZOUTP contains one word for each of the component's output quantities plus one word containing the number of output quantities. QZINPT contains one word for each input quantity (excluding tables) and one word containing the number of input quantities. These records are used by the EASY Model Generation Program to construct the calls in EQMO to the standard component subroutines. The calling sequences are constructed in the following order: one entry for each table, followed by one entry for each output quantity, followed by one entry for each input quantity. The exception is that an output quantity of a component which is declared to be a state variable will have only one entry in EZSTDBF, but will have three entries in the calling sequence of the component's FORTRAN SUBROUTINE, one for the value of the state variable itself (typed real), one for the value of the rate of the state variable (real), and one for a integration control flag (integer), in that order. The order of the calling sequence generated must correspond to the order of quantities in the SUBROUTINE card in the standard component subroutine. Therefore we have:

Constraint #2: Every change effecting the SUBROUTINE card of the standard component subroutine must be accompanied by a corresponding change in the component's records in EZSTDBF, and visa versa.

The steps required to modify EASIEST depend upon the type of modification being made. Each type is discussed below.
1. MODIFYING THE FORTRAN SUBROUTINE OF AN EXISTING STANDARD COMPONENT

The FORTRAN source code for the EASIEST standard components is stored on permanent file EZSTFTN (SN=AFFDL, no passwords). This file also contains source code of routines used by the standard component routines. The contents of this file can be cataloged by editing the file with INTERCOM EDITOR and typing:

L,A,/SUBROUT/

The listing produced on the terminal will be called the "catalog listing". Note that function subroutines do not appear in the catalog. They are located at the end of EZSTFIN, and should not affect the modifying procedures. Each subroutine in the listing resides on a separate record of EZSTFTN and you should note the record number of the subroutine you wish to change. Also, the line numbers on the catalog listing can be used in conjunction with the line numbers on the current FTN output listing to locate the line(s) of EZSTFTN to be changed.

Once the changes have been made, the edit file should be saved and cataloged as a new cycle of EZSTFTN, and the previous cycle should be purged from the disk. The EASIEST library EZSTLIB must now be updated to reflect the changes made to the source code. To do this attach file EZSTPRC (SN=AFFDL, PW=PSWD) and type:

BEGIN,COMPILE,EXSTPRC,n,CODE=cc

where:
1. n is the record number of the record on EZSTFTN you changed (this number can be obtained by counting down on the catalog listing described above),
2. cc is a two character code used in the output listing filename,
3. tid is identifier of the terminal into whose print queue you wish the FTN output listing placed (this entry is required only if you wish the output listing directed to a terminal other than the default terminal AB).

A successful execution of this procedure means that EZSTFTN has now been updated to reflect your change. If the FTN compiler does not accept the changes you made to EZSTFTN, the COMPILE procedure will leave the EASIEST library unchanged and make the FTN output listing containing the error...
description available as local file FTNLIST. This file can be examined from
the terminal using the INTERCOM EDITOR or PAGE utilities. When the trouble is
located, correct EZSTFTN and rerun the compile procedure as described above.

2. MODIFYING THE RANDOM ACCESS FILE EZSTDBF.

If changes are made to a standard component subroutine involving either
the number or characteristics of the components inputs, outputs, or tables,
then in addition to the steps given in section VII.1 for altering the
component's FORTRAN subroutine, the component's EZSTBDF records must be
altered so that the EASY Model Generation Program will alter the generated
calling sequences for the component. EZSTDBF is altered using a program called
FILOAD which in turn is executed from an INTERCOM terminal using a procedure
called DBFMOD contained on the procedure file EZSTPRC. DBFMOD requires the
user to supply a permanent file containing all the data to build the record or
records being modified. This file can have any otherwise unused name; for
illustrative purposes we will assume it is called DBFDATA. For each record of
EZSTDBF being modified the file DBFDATA must contain the following data:

i. A line describing the number on entries in the record in the form:
    "xxINPUTS=n", or "xxOUTPS=n", or "xxTABS=n"
where xx is the component name and n is the number of inputs, outputs, or
tables.
ii. One or more lines containing the names and specifications of the
    inputs, outputs, or tables for the component. Each of these lines (except
    possibly the last) must contain entries for eight quantities. Each entry
    consists of exactly ten characters including spaces and must begin in
    columns 1,11,21,31,41,51,61,71, or 81 of the line. These entries must be
    placed eight to a line until the specified number of quantities has been
given. Each of the entries has the following format:

    Character   Contents
    1-3         the quantity name (inputs, outputs, or tables)
    5-6         the quantity row dimension, if any (inputs, outputs)
    7-8         the quantity column dimension if any (inputs, outputs)
    9           the quantity port number if any (inputs, outputs)
   10         =S if a state (outputs only)
   11-14       total storage allocation (tables only)
   15-18       number of independent variables (tables only)
The dimensions can be one or two digit numbers or can be the symbols N or M which allows the dimensions of the quantity to be set in the model description file. If any input or output quantity of a component is to have variable dimensions, the DBFDATA file should also have a separate line of the form:

\[ \text{MODES} = \text{xx} \]

where xx is the component name.

If more than one record of EZSTDBF is to be modified, the input data for each record can be placed on successive lines of DBFDATA. An easy way to generate the file DBFDATA is to have the procedure DBFMOD generate a local file TMPDATA which contains all the input data to build EZSTDBF as it is now. To do this:

1. Create a permanent file DUMPFIL (no password) containing "DUMP FILE" on a single line of text.
2. Attach the procedure file EZSTPRC;
3. While in INTERCOM command mode type:
   \[ \text{BEGIN, DBFMOD, EZSTPRC, DUMPFIL, EZSTDBF, TMPDATA} \]

Upon successful completion of DBFMOD you will have a local file TMPDATA containing all the input data required to generate the current version of EZSTDBF. The file DUMPFIL can now be purged. Using the INTERCOM EDITOR utility, delete all lines of TMPDATA pertaining to records of EZSTDBF not being modified, make the desired changes to the remaining lines, and save the edit file as DBFDATA.

Once you have the revised DBFMOD input data prepared on file DBFDATA and have cataloged DBFDATA on your account (with no password), attach the file EZSTPRC as before if you have returned it, and type:

\[ \text{BEGIN, DBFMOD, EZSTPRC, DBFDATA, EZSTDBF} \]

Upon successful completion of this procedure, EZSTDBF will have been updated. You may now purge the file DBFDATA. It is recommended that the model description file of the next EASIEST run you submit contain the first line

\[ \text{LIST STANDARD COMPONENTS} \]

This will cause the lineprinter output from that run to contain a listing of all the input, output, and table data for all the standard components. From this listing you can verify that the desired changes have been made to EZSTDBF.
3. CREATING A NEW EASIEST STANDARD COMPONENT

Creating a new component for EASIEST consists of constructing the source FORTRAN code, merging that code into the EASIEST library and constructing the input, output, and table descriptions for the random access file EZSTDBF. The FORTRAN source code for the new component subroutine and any new subroutines needed by your component subroutine should be prepared on a separate file following the constraints above. This code can then be merged into the EASIEST source as follows:

1. Attach the file EZSTFTN and, using INTERCOM EDITOR utility, obtain a "catalog listing" of EZSTFTN as described in section VII.1. Determine the proper position for your new subroutine so that the "Catalog listing" will remain alphabetical.
2. Request a permanent file PF by typing:
   REQUEST,PF,*PF
3. Copy the subroutines that are to precede the new subroutine on EZSTFTN onto the file PF by typing:
   COPYCR,EZSTFTN,PF,n
   where n is the number of subroutine to precede the new one. n can be obtained by counting down on the "catalog listing" of EZSTFTN.
4. Copy the source code of the new subroutine onto PF using
   COPY,f,PF
   where f is the name of the file containing the new source code. Note that file f must be attached before you do the copy.
5. Copy all the remaining subroutines from EZSTFTN onto PF using
   COPYCR,EZSTFTN,PF,999
   The terminal will respond with the number of records copied. This number should be checked against the "catalog listing" to make sure that all the subroutines have been copied. As added insurance, use the INTERCOM EDITOR utility to make a "catalog listing" of file PF and check that PF has the expected structure.
6. Catalog PF as new cycle of EZSTFTN using
   CATALOG,PF,EZSTFTN,RP=999
7. Purge the previous high cycle of EZSTFTN
   PURGE,EZSTFTN
   RETURN,EZSTFTN,PF

8. The new routine can now be compiled and merged into EZSTLIB using the
    procedure COMPILE as described in section VII.1. If more than one subroutine is
    to be added, repeat the above steps.

To include the input, output, and table data for the new component into EZSTDBF,
create a permanent file DBFDATA as described in section VII.2. Usually you will
have to supply data for three EZSTDBF records, xxINPT, xxOUTP, and xxTABS, where
xx is the new component name. However, if the new component has no quantities of a
certain type (inputs, outputs, or tables), then no input data of that type need
be given. When the file DBFDATA is prepared and cataloged (no password), you can
execute the procedure DBFMOD by typing:
   ATTACH,EZSTPRC.
   BEGIN,DBFMOD,EZSTPRC,DBFDATA,EZSTDBF
The terminal will type (among other things):
   xx WILL BE ADDED AS A NEW STANDARD COMPONENT

   You should include LIST STANDARD COMPONENTS command in the model descrip-
   tion file of your next EASIEST run to verify that the inputs, outputs, and tables
   have been specified correctly.

4. LIBRARY EZSTLIB SIZE REDUCTION

   Every time the procedure COMPILE is execute, the EASIEST library file
   EZSTLIB will grow in size. When this size becomes unreasonable EZSTLIB should be
   rebuilt anew from the source file EZSTFTN by typing the following sequence from
   an INTERCOM terminal in command mode:
   ATTACH,EZSTPRC.
   BEGIN,COMPALL, EZSTPRC,EZSTFIN,EZSTLIB
The successful completion of this procedure will mean that a new (smaller) cycle
of EZSTLIB has been cataloged. The previous high cycle can then be deleted. The
FORTRAN output listing from the FTN compilation phase is left available for
routing to a lineprinter as local file ALLLIST.
SECTION VIII
DESCRIPTION AND GUIDE TO USE OF NUMERICAL INTEGRATION

The purpose of this section is: (1) to document changes (as they relate to the user) in integration methods used in the EASY program; (2) to describe local error control procedures in the three automatic integrators - NRKVS, STIFF GEAR, and ADAMS; and (3) to discuss the appropriate use of each method.

1. CHANGES IN INTEGRATORS

Several inadequacies in the integrators used in early versions of EASY were identified and subsequently remedied in the EASY5 program. In particular, the error control technique in the NRKVS integrator was reworked and the Hindmarsh version of C. W. Gear's integrator was implemented. The Hindmarsh version, called GEAR, also includes minor changes, such as dynamic dimensioning and the capability to input EASY5 error controls.

The resulting set of improved integrators are accessed by the EASY5 user through the integration method parameter, INT MODE. INT MODE can be set to any integer from 1 to 6 with the default being 6. The six integrators which are available are listed below.

a. DIFSUB: The original version of Gear's method.
b. NRKVS: The improved Runge-Kutta variable step integrator.
c. HEUNS: Second order fixed step explicit method.
d. Euler: First order fixed step explicit method.
e. ADAMS: Automatic step-size/order selection methods using Adams-Bashforth predictor/Adams-Moulton corrector pairs of (2nd through 12th) order. (Non-stiff option of GEAR.)
f. STIFF GEAR: The stiffly stable GEAR formulas.
The choice of the best integration method depends on a number of considerations. User requirements, problem characteristics, and the stability and accuracy of the method all must be considered. A more complete discussion of these considerations can be found in standard texts. It is the purpose of this section to present summary information to help the user with his integrator selection. The second and third sections discuss accuracy, error control, and stability in more detail and can be consulted if integration problems develop or simply to gain a better understanding of the processes involved.

2. GENERAL SELECTION GUIDELINES

Many times the best and only way to choose a method is by trial and error. Below are some general observations:

a. If no special knowledge is available about the system, try Method 5: ADAMS.

b. If a large amount of output is desired at small time increments, Methods 5 or 6 will use interpolation rather than generate smaller time steps if output points are smaller than current step sizes. However frequent restarting will cause the cost of an entire transient simulation to increase.

c. If function evaluation can only be calculated at fixed time steps due to sampling data or tabular information, use Methods 3 or 4. Method 3 is more efficient if the time step is obviously small enough to generate necessary accuracy. Given a fixed time step, Method 4 will be more accurate than method 3 provided \( h_0 \) is within the stability region (Figure 11) for the methods.

d. If the system has frequent derivative discontinuities (shocks, phase changes, hard step-like forces, etc) Method 2: NRKVS is recommended. Unlike Methods 5 and 6, Method 2 is negatively impacted by a large number of output points at small time increments (i.e., if the output
Figure 11. Stability Regions For Runge-Kutta Methods: Orders 1-5
time increments are smaller than the natural step size, then the method will be found to use more integration steps and thus be more costly).

e. Method 1 is not recommended. If the system is initially unstable or discontinuous and eventually is stiff, we recommend using Method 2, then switching to Method 6 rather than using Method 1.

f. If the problem is stiff (i.e., large spread in eigenvalues), Method 6: STIFF GEAR is recommended. This is also the default option if no method is specified.

It should be noted, however, that problems with large eigenvalues (with negative real parts) do not automatically indicate that one should use STIFF GEAR. For example, consider the system:

\[
\begin{align*}
(1) \quad x_1 &= -x_1 \quad \text{for time } 0 \leq t \leq b \\
\quad x_2 &= -1000x_2
\end{align*}
\]

This is an uncoupled system (and might seem artificial), but coupled systems often display the behavior of rapidly damping components such as \(x_2\). If one was integrating (1) as a system and the important variable was \(x_1\) and \(b\) was large, then a large step size could be used provided the numerical integration of \(x_2\) was damping to zero (i.e., stable). In such a case, a STIFF method would be appropriate. On the other hand, if \(b\) was small (e.g., \(b = 0.0001\)) and \(x_2\) was the component of interest (where relative accuracy is important), then an efficient integrator of Adams type or perhaps a Runge-Kutta method would be appropriate. Thus, the decision to use STIFF GEAR or not depends on both the user requirements for accuracy and the eigenvalues of the system.
3. ACCURACY AND ERROR CONTROL

It is useful to establish notation and review some basic concepts. Consider the ordinary differential equation (ODE)

\[ X(t) = f(t, X(t)) \text{ with } a \leq t \leq b \]

with the initial condition \( X(a) = X_0 \). Equation (2) is an initial value problem. The EASY5 program sets up and solves first order systems of such equations (i.e., equations of the form of equation (2) with \( X \) a vector and \( f \) a vector valued function). Initial value methods for integrating ODE's produce a sequence \( X_j \) of approximations to the solution \( X \) such that \( X_j \approx X(t_j) \) where \( t_0 = a \) and \( t_j = t_{j-1} + h_j \) for \( j = 1, N \). The sequence \( h_j \) are called steps or step sizes. For Methods 3 and 4 (Heun's and Euler's methods), the step sizes are fixed throughout the integration and are set by the user through the parameter TINC. For the other methods, the step sizes are selected by the integration algorithm as the integration proceeds. These "adaptive" methods estimate the local truncation error at each step of the integration, accept or reject the approximation, and predict the next step size to be tried. Local truncation error can be loosely thought of as the error incurred during one-step of the integration process given that all previous approximates are exact. The order of a method is a crude measure of accuracy. A method is said to be of order \( p \) if it is exact for \( p \)-th order polynomials. The adaptive EASY5 integrators (Methods 1, 2, 5, and 6) strive to keep the step size small enough to insure reasonable local error which in turn should produce a small global error. Whether or not the global error is indeed small will depend on both the problem and the stability of the method. (Stability is discussed in the next section.)

The adaptive integrators measure the local truncation error by comparing two estimates of the solution that theoretically differ in only high order terms from the Taylor's expansion of the solution over the current step. The details of how this is done in each method is not important here except as to how it relates to the EASY5 integration controls. The user is asked
to input an array of controls associated with each state of the system via the ERROR CONTROL command. The array, which we shall call ERROR(I), is a measure of significance of the corresponding Ith state of the system. To be precise ERROR(I) is a value below which the Ith state is in some sense considered negligible by integrators 1, 2, 5, and 6. There are two methods of error control employed by the four methods. Method 2, NRKVS, is described first, Error control in Methods 1, 5, and 6 are basically the same and will be discussed second.

In NRKVS, the initial step size $H_0$ is chosen as a function of TINC. To be precise $H_0 = .01 * \text{TINC}$.

Subsequent step sizes are selected on the basis of local error control estimates. There are a number of refinements in NRKVS that will not be discussed; however, the basic error control is governed by the following quantity, $Q$,

$$Q = \max_I \left[ \frac{\text{LTE}(I)}{\text{ERROR}(I) + |X(I)| * \text{ERROR}(I)} \right]$$

where LTE(I) is the local truncation error estimate for the Ith state of the solution as calculated by comparing a 4th order solution to a 5th order solution, $X(I)$ is a recent history size measure of the Ith state (initially set to the initial value), and ERROR(I) is the user input error control. The integrator strives to make $Q = 1$. If $Q < 1$, the step size on the next integration step is increased. If $Q > 10$, the current step is rejected and a new smaller step size is calculated for another attempt. In order to interpret the effect of the input controls, ERROR(I), one need only set $Q = 1$ (the desired value for Q) and examine the relation (2) for the maximal choice of I. That is, for some I, if $Q = 1$, then

$$Q = \frac{\text{LTE}(I)}{\text{ERROR}(I) + |X(I)| * \text{ERROR}(I)}$$
Thus by rewriting (4) we have that

\[ \text{LTE}(I) = \text{ERROR}(I) + \bar{\Omega}(I) \cdot \text{ERROR}(I). \]

i.e., the LTE is close to the \( \text{ERROR} + \bar{\Omega} \cdot \text{ERROR} \). If \( \bar{\Omega}(I) \) has been small, \( \text{ERROR}(I) \) dominates the right hand side of (4), and the error control is essentially absolute error. On the other hand, if \( \bar{\Omega}(I) \) is very large, \( \bar{\Omega}(I) \cdot \text{ERROR}(I) \) will dominate; and thus relative error is controlled. As a rule of thumb, the user should input the level at which he considers the solution negligible (i.e., tolerably small enough to ignore). If the solution gets large, then \( \log_{10}(\text{ERROR}) \) will roughly give the number of significant digits of accuracy (locally).

The use of input controls \( \text{ERROR}(I) \) differs for Methods 1, 5 and 6. A local truncation error LTE is computed by the integrator. The Euclidean error is controlled, i.e.,

\[
\sum_{I=1}^{\text{NEQ}} \left( \frac{\text{LTE}(I)}{\text{XMAX}(I)} \right)^2
\]

is required to be less than \( (\text{EPS})^2 \) where \( \text{NEQ} \) is the number of equations, \( \text{XMAX}(I) \) is the maximum of the \( I \)th component of \( X \) over the course of the integration. The user impacts this control by effecting the initialization of \( \text{XMAX}(I) \) and the choice of \( \text{EPS} \). \( \text{EPS} \) is chosen as follows:

\[
\text{EPS} = \min \left( \text{ERROR}(I) \right) \\
(1)
\]

with the constraint that \( \text{EPS} \leq .01 \). If \( \text{ERROR}(I) < 1.0 \times 10^{-12} \) for all \( I \), then \( \text{EPS} \) is set to \( 1.0 \times 10^{-4} \). The initialization of \( \text{XMAX}(I) \) is given by

\[
\text{XMAX}(I) = \frac{\text{ERROR}(I)}{\text{EPS}} \\
\text{IF} (\text{XMAX}(I) \neq 0) \text{XMAX}(I) = 1.
\]
The net effect of these initializations for EPS and the XMAX array result in the ERROR array being used in a similar manner to its use in NRKVS. For example, if EPS = .001 and ERROR = .001, then XMAX = 1.0 and error control is essentially absolute error until the solution X(I) exceeds 1. If X(I) grows the error processing will gradually become relative since XMAX is set equal to X whenever X exceeds it. If the solution grows to a maximum value, and then decays, the error control will be relative to that maximum.

The user must remember that EPS is set by the smallest ERROR(I). Thus, in a two component system, if ERROR(1) = .001, and ERROR(2) = 1.0, the resulting controls will be as follows:

\[ \text{EPS} = .001; \text{XMAX}(1) = 1; \text{XMAX}(2) = 100. \]

Thus, if X(1) = X(2) = 0 initially the integrator considers values less than 0.001 negligible for X(1) and values less than 1.0 negligible for X(2). This is quite similar to what NRKVS would do with these same inputs for ERROR(1) and ERROR(2).

4. STABILITY

The theoretical basis for error control and convergence of numerical integration methods is rooted in the underlying assumption that the step size is small (in fact, approaching zero). In practice, of course, the step size is not necessarily small and certainly not zero. In fact, the larger the step size, the fewer the steps required, and hence, the greater the economy of integration. The behavior of integration methods when the step size gets large will generally depend on both the problem and the "stability" of the method. All the EASY5 integrators are at least "conditionally stable". That is, there exists a threshold size, \( h_0 \), such that for steps of \( h < h_0 \) the integration procedure will produce damping approximations to damping components. To be precise, consider the equation

\[ X(t) = \lambda X \]
where $\lambda$ is any complex number. If $\lambda$ has negative real part, the equation is said to be mathematically stable, and its solution may be oscillatory but definitely will damp with time. Given a method, one can calculate a stability region in the complex plane which depicts the region in the $h$ plane for which the integration scheme will produce a damping solution to equation (5). That is, given a $\lambda$, the product $h\lambda$ must be within the absolute stability region for the method to produce a damping solution. Generally, if one uses a step size $h$ outside this region for more than a few successive steps, numerical instability will occur producing a divergent "solution" even for a stable system. This, in fact, often happens with fixed step methods. Adaptive integrators will automatically reject these numbers and cut the step size, thereby increasing work (not because of accuracy) but because of stability. For systems of nonlinear differential equations, in equation (5) corresponds to the eigenvalues of the system. In Figure 11, the stability regions for Runge-Kutta methods of orders 1-5 are shown. The method will be stable provided $h\lambda$ is within these closed regions.

The region marked $p=1$ is valid for Euler's method (No. 4 in EASY5). Thus if, for example, $\lambda = -1000$, $h\lambda$ is required to be $> -2$ in order to produce meaningful results. This, in fact, implies that $h < .002$. The region $p=2$ corresponds to Heun's method which is Method 3 in EASY5. For $\lambda = -1000$, $h$ must also be less than .002 for stability. In this case, since Method 4 uses only one function evaluation per step and Method 3 uses two, the Euler method would be more efficient on $X(t) = -1000X$ if minimal accuracy were needed. On the other hand, if extremely accurate results were required, and the user intended to use small step sizes well within the stable regions, then the higher order accuracy of Heun's method would more than justify its extra function evaluation.

For a certain class of equations (stiff equations) the eigenvalues may vary considerably between components. For example, consider

\[
\begin{align*}
\dot{x}_1 &= -1000 \ x_1 \\
\dot{x}_2 &= -x_2
\end{align*}
\]
The user often demands greater accuracy in $X_2$ than in $X_1$. Assume for the moment these equations are coupled. Then $X_1$ drives the step size used for the system. It is in this situation that large stability regions are desirable, for then a large step size can be used.

The $p = 4$ and $p = 5$ regions in Figure 11 are then stability regions that apply to the Runge-Kutta method NRKVS. The underlying method is 4th order in NRKVS; however, the error control mechanism performs an extrapolation to achieve a fifth order approximation to the computed 4th order estimate. The difference between the two is then used to estimate the error. The reported solution is the fifth order estimate; hence $p = 5$ is the true region of interest.

The Adams-Moulton formulas, Method 5, have stability regions given in Figure 12 for orders 3-6. (The Adams-Moulton methods are the corrector of the predictor-corrector pairs used in Method 5.) A corrector formula is implicit and if iterated to convergence, will have the stability shown in Figure 12. However, the implementation of the Adams formulas in this code (and most Adams' codes) uses prediction with only one correction. The resulting stability regions are reduced. A sample of these regions for orders 1, 2, 3, 4, 5, 6, 9, and 10 are given in Figure 13a through h. The solid lines are for the Adams-Bashforth predictors; the dotted lines are for the Adams-Bashforth predict with the Adams-Moulton corrector of the same order (one correction); and the dashed lines are order $k$ predict/order $k+1$ correct. The dotted lines represent the actual implementation in EASY5.

Thus far, the stability regions discussed have all been finite (bounded) regions of the plane. Consequently, to remain in the stability region of the plane for any $\lambda$ with very large absolute value, one must use a very small step size. The advantage of the STIFF GEAR formulas (Method 6 in EASY5) is that their stability regions have infinite extent. This is shown graphically in Figures 14 and 15. For orders $k=1,2$, these methods are A-stable which means that for any $\lambda$ with negative real part $\lambda h$ will fall within the stable region for any $h > 0$. The higher order formulas (Figure 15) impose restrictions on the size of the imaginary part of that
Figure 12. Stability Regions For Adams-Moulton Formulas
Figure 13a thru h. Stability Regions for Predictor-Corrector Pairs
Figure 14. Stability Regions for STIFF GEAR Formulas of Orders 1-3

Figure 15. Stability Regions for STIFF GEAR Formulas of Orders 4-6
allow for large values of h. For example, the sixth order formula requires small h if \( \lambda = -1000 + 1000i \), but for \( \lambda = -1000 \) the sixth order formula is stable for all values of h > 0. In the EASY5 program, implementation of STIFF GEAR the order may vary from one to five. Since Methods 1, 5, 6 are variable order codes, they will range over various orders during the integration. If the eigenvalues are close to the imaginary axis, Method 6 will probably use only orders 1, 2, and possibly 3 if it is constrained by stability. These highly stable methods generally require more function evaluations than the other methods mentioned (due to internal approximations to the Jacobian of the system required to solve implicit equations).
1. INTRODUCTION

The discrete system analyses of the EASY program are based on the state space approach described by Kalman and Bertram in Reference 1. The EASY analyses utilize the single and multirate sampling capabilities of the original analysis. Other capabilities such as the analysis of nonsynchronous, noninstantaneous, multiple order, and random sampling are not currently implemented in the EASY program. The EASY program analyses parallel those of the M-DELTA program. However, whereas the M-DELTA program requires the user to input the A and B matrices that described the system, the EASY program calculates these matrices from a nonlinear system model described in terms of standard modeling components.

Only the linear analyses of the EASY program utilize the techniques of Kalman and Bertram. Since only the eigenvalues of the system are used in these analyses, the system equations will be simplified in the following derivations by treating the system as autonomous.

2. SYSTEM EQUATIONS

A discrete system may be described by the following three types of states:

a. Continuous States
b. Delay States
c. Sample and Hold States

The continuous states may vary continuously as a function of time and are each defined by a first order ordinary differential equation. Delay states are defined at only discrete points in time by first order difference equations. Sample and hold states maintain constant values except at discrete points in time where they may jump to new values. Figure 16 shows an example of each state type.
Figure 16. Example of Continuous, Delay, and Sample States
Let the continuous delay and sample state be grouped together as three state vectors:

- $X_c$: VECTOR OF CONTINUOUS STATES
- $X_d$: VECTOR OF DELAY STATES
- $X_s$: VECTOR OF SAMPLE STATES

The total system state vector of dimension $\gamma + \delta + \sigma$ is formed into the single partitioned vector:

$$
X = \begin{bmatrix}
X_c \\
\cdots \\
X_d \\
\cdots \\
X_s
\end{bmatrix}
$$

a. Continuous System Stability Matrix

Between sample instants, the autonomous system behavior is described by:

$$
\dot{X} = AX
$$

The system stability matrix $A$ between sampling instants may be expressed as the partitioned matrix:

$$
A = \begin{bmatrix}
A_{cc} & 0 & A_{cs} \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
$$

The form of the system stability matrix demonstrates that only the continuous states have non-zero rates, i.e., can change between sampling
instants, and that the continuous state rates are functions of only the continuous states and sample states.

b. Discrete System Transition Matrix

At sampling instants, the system behavior is described by:

\[ \mathbf{x}(t+1) = \mathbf{Bx}(t) \]  

(4)

For a single rate sampling system, the discrete transition matrix \( \mathbf{B} \) will be of the form:

\[
\mathbf{B} = \begin{bmatrix}
1 & 0 & 0 \\
\mathbf{B}_{dc} & \mathbf{B}_{dd} & 0 \\
\mathbf{B}_{sc} & \mathbf{B}_{sd} & 0
\end{bmatrix}
\]

(5)

The form of the transition matrix at sampling instants demonstrates that the continuous states remain unchanged, i.e., the upper \( Y \) rows contain only an identity matrix. The discrete states are functions of only the continuous and delay states at the previous sample instant.

c. Continuous System Transition Matrix

Equation (4) describes the instantaneous changes that occur in the system at sample instants while equation (2) describes the system between sampling instants. In order to combine these two types of behavior, we will convert the continuous description of (2) into a transition matrix that describes the transition between two sample instants.
Expanding (2):
\[
\begin{align*}
\dot{x}_c &= \Delta_{cc} x_c + \Delta_{cs} x_s \\
\dot{x}_d &= 0 \\
\dot{x}_s &= 0
\end{align*}
\] (6)

Take Laplace transform:
\[
\begin{align*}
sX_c(s) - x_c(0) &= \Delta_{cc} X_c(s) + \Delta_{cs} X_s(s) \\
sX_d(s) - x_d(0) &= 0 \\
sX_s(s) - x_s(0) &= 0
\end{align*}
\] (7)

Rearrange terms to solve for \(X_c(s), X_d(s),\) and \(X_s(s)\):
\[
\begin{align*}
X_c(s) &= \left[sl - \Delta_{cc}\right]^{-1} x_c(0) + \left[sl - \Delta_{cc}\right]^{-1} \frac{sl - \Delta_{cc}}{s} \Delta_{cs} X_s(0) \\
X_d(s) &= \frac{x_d(0)}{s} \\
X_s(s) &= \frac{x_s(0)}{s}
\end{align*}
\] (8)

Take inverse Laplace transform:
\[
\begin{align*}
x_c(\tau) &= e^{\Delta_{cc} \tau} x_c(0) + \Delta_{cc}^{-1} \left[e^{\Delta_{cc} \tau} - 1\right] \Delta_{cs} x_s(0) \\
x_d(\tau) &= x_d(0) \\
x_s(\tau) &= x_s(0)
\end{align*}
\] (9)
Equation (9) is in the form of a transition equation from an initial time to a final time $\tau$. It is also of the same form as equation (4) and may be written as:

$$x(\tau) = \Phi(\tau) x(0)$$  \hspace{1cm} (10)

Where:

$$\Phi(\tau) = \begin{bmatrix} e^{A_{cc}\tau} & 0 & A_{cc}^{-1}[e^{A_{cc}\tau} - 1]A_{cs} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ \hspace{1cm} (11)

When written in this form, we see that the transition matrix of the system between sampling instants is composed of the exponential decay term $e^{A_{cc}\tau}$ due to the continuous states plus the effect of the step input from the sample states. The discrete states are constant between sampling instants as evidenced by the identity terms.

d. Calculation of Continuous System Transition Matrix

If the continuous system matrix $A_{cc}$ has $Y$ independent eigenvectors, the exponential function $e^{A_{cc}\tau}$ may be expressed as:

$$e^{A_{cc}\tau} = W e^{\Lambda \tau} W^{-1}$$ \hspace{1cm} (12)

where: $W$ modal matrix of $A_{cc}$ eigenvectors $\Lambda$ diagonal matrix of $A_{cc}$ eigenvalues

The second term in the $\Phi(\tau)$ matrix may be expressed as:

$$A_{cc}^{-1}[e^{A_{cc}\tau} - 1]A_{cs} = W \Lambda^{-1}[e^{\Lambda \tau} - 1]W^{-1}A_{cs}$$ \hspace{1cm} (13)
This spectral factorization approach is used by the EASY program to calculate the transition matrix $\Psi$.

If the continuous system stability matrix $A_{cc}$ does not have $Y$ independent eigenvectors, this is detected by the program and a Padé approximation method is used to calculate $e^{A_{cc}t}$. This occurs in a continuous system in which components with exactly the same eigenvalues appear in a series connection. The sixth order Padé approximation is:

$$e^{A_{cc}t} = \left[1 - \frac{t}{2} A_{cc} + \frac{t^2}{10} A_{cc}^2 - \frac{t^3}{120} A_{cc}^3\right]^{-1} \left[1 + \frac{t}{2} A_{cc} + \frac{t^2}{10} A_{cc}^2 + \frac{t^3}{120} A_{cc}^3\right]$$

(14)

3. Combined System Transition Matrix

It is proved by Kalman and Bertram in reference 1 that the stability of a periodic system is determined by the eigenvalues of the combined system transition matrix, that is, the transition matrix that describes one complete system of the system operation.

a. Single Sample Rate

For a single sample rate system, the transition matrix would be obtained by the product of one $B$ matrix, as given in (4) with one $\Phi$ matrix as given in (10). Such a system is shown in Figure 17. The continuous system stability matrix for this system would be:

$$A = \begin{bmatrix}
-10 & 0 & 10 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}$$

(15)
Figure 17. Single Sampling Rate Example

Figure 18. Pictorial Representation of Single Sampling Rate Transition Matrices
The discrete system transition matrix would be:

\[ B = \begin{bmatrix} 1.0 & 0 & 0 \\ -0.009901 & 0.9900 & 0 \\ -0.004975 & 1.0 & 0 \end{bmatrix} \]  

\[ (16) \]

The continuous system transition matrix for this system is:

\[ \Phi(0.01) = e^{0.01A} = \begin{bmatrix} 0.904837 & 0 & 0.0951625 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

\[ (17) \]

A complete cycle of this system occurs after one sample period as shown in Figure 18. The system transition is given by:

\[ x(0.01+) = \Phi(0.01)Bx(0) = \Phi(0.01)x(0) \]

\[ (18) \]

The total system transition matrix:

\[ \Psi(0.01) = \begin{bmatrix} 0.90436 & 0.09516 \\ -0.009901 & 0.9900 \\ -0.004975 & 1.0 \end{bmatrix} \]

\[ (19) \]

\[ \Psi(0.01) = \begin{bmatrix} 0.90436 & 0.09516 \\ -0.009901 & 0.9901 \end{bmatrix} \]

\[ (20) \]
Note that the final system transition matrix product is shown as a 2 x 2 rather than a 3 x 3 matrix. The sample state, \( X_3 \), has a zero column in the final transition matrix, and therefore, contributes nothing to the state of the system at the next sample period. The row and column corresponding to this state may, therefore, be dropped from the total system stability matrix at this point in the analysis. This will occur in general for all sample states in a model. However, in order to express the total system transition matrix as a simple product of matrices, it is necessary to carry the sample states along in the matrix calculation until the final transition matrix is formed.

b. Integer Multiple Sampling Rate

For a multiple sampling rate system, we will first consider the special case where the larger sample periods are all integer multiples of all smaller sample periods. An example of such a system is shown in Figure 19. Here the sampling periods are: \( \tau_1 = .01 \) and \( \tau_2 = .04 \). Our objective is to build the total system transition matrix, \( \Phi \), that spans one complete cycle of the multirate system as shown in Figure 20, one complete cycle occurs for this system after four samples of the fastest sampling rate. The total system transition matrix, \( \Phi \), can be expressed as:

\[
\Phi = \left[ \Phi_{.01} \right] \left[ B_{.01} \right] _{.04}^4 \tag{22}
\]

by means of the transition property of transition matrices. For the multirate case, there is a \( B \) matrix for each sampling rate. The multirate \( B \) matrices shown in \( \Phi \) differ only slightly from the single rate form of \( 5 \). They are of the form:

\[
B = \begin{bmatrix}
1 & 0 & 0 \\
B_{dc} & B_{dd} & 0 \\
B_{sc} & B_{sd} & B_{ss}
\end{bmatrix}
\tag{23}
\]

The rows of \( B \) corresponding to discrete states which do not change at period \( \tau \) are equal to the corresponding row from an identity matrix. The
Figure 19. Multisampling Rate Example
Figure 20. Pictorial Representation of Multisampling Rate
Transition Matrices-Integer Multiple Rates
rows of $B_{\tau}$ corresponding to sampler states of the period $\tau$ have zero elements in $B_{ss}$. Thus the only difference between $B_{\tau}$ and the $B$ matrix shown in (5) is the possible addition of ones on the diagonal of $B_{ss}$ for those sample states corresponding to periods other than $\tau$.

This may be seen by examining the matrices for the example system of Figure 19.

Continuous system stability matrix:

$$A = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

(24)

Discrete transition matrix for sample period .01:

$$B_{.01} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & -2.994 & .8176 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 18.42 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

(25)

Discrete transition matrix for sample period .04:

$$B_{.04} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
2.560 & 47.15 & 2.560 & .6629 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
-8.595 & -158.3 & -8.595 & 1 & 0 & 0
\end{bmatrix}$$

(26)

One point should be made regarding the model of Figure 19. The sample state $X_5$ is redundant since it is in a path that only leads to other discrete states. Sample states are normally used only in paths that lead from delay states to continuous states. In order to simplify the assembly of discrete system models, the EASY program models of all digital filters
contain a sample state on their output. However, during the calculation of the B matrices by the EASY Analysis program, these samplers are treated as being closed, for all sample periods which are modulo their sample rate. This causes the sampler \( X_5 \) to pass information from continuous state \( X_2 \) and delay state \( X_3 \) on to discrete states \( X_4 \) and \( X_6 \). Thus, the \( B_{0.04} \) matrix has the correct non-zero elements \((4,2), (4,3), (6,2), \) and \((6,3)\) that would occur if the sample state \( X_5 \) had been omitted from the model.

The functional form of equation (22) can be extended to any number of sampling rates as long as each larger sample period is an integer multiple of the next lower sample period. Thus if:

\[
N_2 = \frac{T_2}{T_1} \\
N_3 = \frac{T_3}{T_2} \\
\vdots \\
N_n = \frac{T_n}{T_{n-1}}
\]

then the total system transition matrix is

\[
\Psi = \left[ B_{T_1}^{N_2} B_{T_2}^{N_3} \ldots B_{T_n}^{N_n} \right]
\]

The EASY program is currently dimensioned for \( n = 10 \), i.e., up to ten different sampling rates may occur in one model.

c. Noninteger Multiple Sampling Rates

For noninteger multiple sampling rates, the simple expression of (28) cannot be used. However, the same technique of building up the total system transition matrix from a continuous system transition matrix and a series of discrete system transition matrices still applies. For example, consider the system shown in Figure 19 with sample periods of 0.02 and 0.03 in place of 0.01 and 0.04. Figure 21 shows a pictorial representation of the transitions that take place to complete a cycle.

The total system transition matrix can be expressed in terms of the basic transition matrices as follows:
Figure 21. Pictorial Representation of Multisampling Rate
Transition Matrices-Noninteger Multiple Rates
In this case it is necessary to introduce a continuous system transition matrix that spans a period, \( \tau_1 = .01 \), which is less than the smallest given sampling period, \( \tau_1 = .02 \). The total system period, \( .06 \), is also greater than the largest given sampling period, \( \tau_2 = .03 \).

In general, the continuous system transition matrix is required for a period, \( \tau_0 \), which is the greatest common divisor of the sample periods:

\[
\tau_0 = \text{g.c.d.}(\tau_1, \tau_2, \ldots , \tau_n)
\]

(30)

The total system period, \( T_{\text{max}} \), will be the least common multiple of the sample periods.

\[
T_{\text{max}} = \text{l.c.m.}(\tau_1, \tau_2, \ldots , \tau_n)
\]

(31)

In order to form the total system transition matrix, the quantities \( \tau_0 \) and \( T_{\text{max}} \) are calculated. The total period \( T_{\text{max}} \) is then scanned in increments of \( \tau_0 \) and the appropriate power of \( \Phi(\tau_0) \), and \( B_\tau \) matrices are multiplied together to form the total system transition matrix. This capability is not currently available in the EASY program.
REFERENCES


**APPENDIX A**

**EASY5 - MODEL GENERATION - COMMANDS**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD PARAMETERS = $q_1, q_2(n_1, n_2)$</td>
<td>Add parameters to model (also dimensions)</td>
</tr>
<tr>
<td>ADD TABLES = $t_1, n_1, n_2, t_2, n_2, t_3, n_3$</td>
<td>Add tables to model</td>
</tr>
<tr>
<td>ADD VARIABLES = $q_1, q_2(n_1, n_2)$</td>
<td>Add variables to model (also dimensions)</td>
</tr>
<tr>
<td>*Comment #</td>
<td>Add comment to model description</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Add debug print statements to model</td>
</tr>
<tr>
<td>DIAGNOSTIC CONTROL = n</td>
<td>Control diagnostic printout from model genera-n program</td>
</tr>
<tr>
<td>END OF MODEL</td>
<td>Specify end of model description</td>
</tr>
<tr>
<td>FORT</td>
<td>Specify user Fortran Component</td>
</tr>
<tr>
<td>FORTRAN STATEMENTS</td>
<td>Specify start of FORTRAN statements</td>
</tr>
<tr>
<td>$L_1$, $L_2$, ...</td>
<td></td>
</tr>
<tr>
<td>INPUTS = $C_1(q_{out} = q_{in})$, FORT($q_{out} = q_{in}$)</td>
<td>Specify source of inputs to components</td>
</tr>
<tr>
<td>LIST STANDARD COMPONENTS</td>
<td>Request listing of standard components</td>
</tr>
<tr>
<td>LOCATION = $n_1$, $n_2$, $n_3$</td>
<td>Specify component location on schematic</td>
</tr>
<tr>
<td>Matrix arithmetic #</td>
<td>Compact Matrix Algebra</td>
</tr>
<tr>
<td>MODEL DESCRIPTION = test</td>
<td>Specify start of model description</td>
</tr>
<tr>
<td>O.C. ANALYSIS</td>
<td>Specify only analyses-no O.C. DESIGN</td>
</tr>
<tr>
<td>O.C. CRITERIA = $q_1, q_2, \ldots$</td>
<td>Specify O.C. criteria variables</td>
</tr>
<tr>
<td>O.C. INPUTS = $q_1, q_2$</td>
<td>Specify O.C. input variables</td>
</tr>
<tr>
<td>O.C. MODEL ORDER = n</td>
<td>Specify model order to be used for O.C. DESIGN</td>
</tr>
<tr>
<td>O.C. ORDER = n</td>
<td>Specify optimal controller order</td>
</tr>
<tr>
<td>O.C. OUTPUTS = $q_1, q_2, \ldots$</td>
<td>Specify O.C. output variables</td>
</tr>
</tbody>
</table>
Format Description

PRINT Request printed model output
Standard Components # Standard Components----see list
 C,N = n_1, M = n_j Dimension Standard Component
TABLE DIMENSION = t_1=n_1, Specify table standard component
     t_2=n_2,.... Table dimensions
/*EOR # End of record for mini-time-share file

#Not a command

Modifier Notations Phrase Delimiters
C_1 - Standard component name = equal sign
L_1 - Line of FORTRAN source code , comma
n_1 - Integer number ( left parenthesis
q_1 - Input or output quantity ) right parenthesis
     name three or more blanks
t_1 - Table name

184
APPENDIX B

EASY5 - ANALYSIS - COMMANDS

ALL STATES
CALCOMP
CALC XIC
DEFINE PARAMETERS = p1 = p2, ...
DEFINE RATES = r1 = r2, ...
DEFINE STATES = s1 = s2, ...
DEFINE VARIABLES = v1 = v2, ...
DESIGN O.C.
DISPLAY i = 1, 2, 3, 4, 5, 6
    q1, vs, TIME
    Max
    q2, vs, q3
    ...
    ...
EIGEN SENSITIVITY
    EIGEN PARAMETER = p1
    ERROR CONTROL = s1 = n1, ...
    INITIAL CONDITIONS = s1 = n1, ...
INITIAL TIME = n
INT CONTROL = s1 = n1, ...
LINEAR ANALYSIS
Matrix Parameters*
MTS PLOTS
NO STATES
O.C. DATA
    YOP; UOP; Q; RU; CD;
    CS; G; S; A; FK

Activate all model states (DEFAULT)
Requests plots on CalComp plotter
Allows manual I.C. calculations
Define parameter names
Define rate names
Define state names
Define variable names
Initiate optimal controller design
Specify quantities to be plotted
(5 plots/display 6 displays = max 30 plots
3000 points/display set)
Initiate eigenvalue sensitivity calculation
Specify integrator error controls
Specify initial conditions/operating point
Specify initial value of time
Activate or freeze model states
Initiate linear analysis
Input matrix parameter values
Requests plots on MTS plotter
Freeze all model states
Input optimal controller data
OMIT PLOT POINTS
OMIT TABLE PRINTOUT
PARAMETER VALUES = \( p_1 \) = \( n_1 \), ...
PLOT ALL TABLES
PLOT ID = text
PLOT OFF
PLOT ON
PLOT TABLES = \( t_1, t_2, \ldots \)
PRINT
PRINT2
PRINT VARIABLES = \( q_1, \ldots, q_{10} \)

PRINTER PLOTS
ROOT LOCUS
   RL PARAMETER = \( p \)
   RL START = \( n \)
   RL STOP = \( n \)
   RL POINTS = \( n \)
   RL MANUAL SCALES
      REAL MIN = \( n \)
      REAL MAX = \( n \)
      IMAG MIN = \( n \)
      IMAG MAX = \( n \)
   RL AUTO SCALES

SAVE O.C.

SCAN1
   DEPEN = \( q \)
   START2 = \( n \)
   DELTA2 = \( n \)
   CURVES2 = \( n \)
(Also requires DEPEN, INDEP1, START1, STOP1)

Omit boxes around plot points
Omit print back of table inputs
Input parameter values
Request plots of all tables
Specify plot identification
Deactivate plotting (DEFAULT)
Activate plotting
Requests plots of specified tables
Initiate single pring pass through model
Specify second print option
Specify columnar option print variables
(\( \text{PRINT CONTROL}=5 \))
Requests plots on line printer
Initiate root locus analysis
Specify root locus parameter
Specify initial value of RL PARAMETER
Specify final value of RL PARAMETER
Specify number of root locus points
Request manual root locus plot scales
Real axis minimum scale value
Real axis maximum scale value
Imaginary axis min. scale value
Imaginary axis max. scale value
Request auto plot scales (DEFAULT)
Write optimal controller arrays to TAPE3
Initiate one dimensional function scan
Specify 2nd dependent variable
Specify initial value of INDEP2
Specify increment size for INDEP2
Specify number of values for INDEP2
SC4020

Request plots on SC4020 microfilm

SIMULATE
  PRINT CONTROL = n
  PRINT2 = n
  PRATE = n
  PRATE2 = n
  OUTRATE = n
  OUTRATE2 = n
  INT MODE = n
  TINC = n
  TINC2 = n
  TMAX = n
  SI MANUAL SCALES
  SI AUTO SCALES

  STABILITY MARGINS
    SM PARAMETERS = p_1, ..., p_10

  STEADY STATE
    SS PARAMETER = p
    SS START = n
    SS STOP = n
    SS POINTS = n
    SS ITERATIONS = n
    SS MANUAL SCALES
    SS AUTO SCALES

  TABLE = t, n, n, n
      (table data)

  TITLE = text

Initiate simulation (Time History)
  Specify print option
  Request printout every n plot intervals
  Request plot points every n*TINC
  Specify integration method
  Specify integrator report interval
  Specify time history duration
  Request manual simulation plot scales
  Request auto plot scales (DEFAULT)

Initiate stability margin calculation
  Specify stability margin parameters

Initiate steady state calculation
  Specify SS parameter (optional)
  Specify initial value of SS PARAMETER
  Specify final value of SS PARAMETER
  Specify number of SS calculations
  Specify number of iterations to be used
  Request manual plot scales
  Request auto plot scales (DEFAULT)

Input tabular data
  Specify plot title

187
TRANSFER FUNCTION
  TF INPUT = q
  TF OUTPUT = q

BODE
  NICHOLS
  NYQUIST
  TF MANUAL SCALES
    FREQ MIN = n
    FREQ MAX = n
  TF AUTO SCALES

XIC-X
  XIC_t-XIC  i=1,2,3
  XIC-XIC_i  i=1,2,3

/*EOF End of file for mini-time-share file

#Not a Command

Modifier Notations
  n_i - numeric value
  p_i parameter name
  q_i - parameter, variable, state, or rate name
  r_i - rate name
  s_i - state name
  t_i - table name
  v_i - variable name

Phrase Delimiters
  = equal sign
  , comma
  ( left parenthesis
  ) right parenthesis
  three or more blanks

Initiate transfer function calculation
  Specify transfer function input quantity
  Specify transfer function output quantity
  Request Bode format for plots
  Request Nichols format for plots
  Request Nyquist format for plots
  Request manual plot scales
  Specify minimum frequency r.p.s.
  Specify maximum frequency r.p.s.
  Request auto plot scales (DEFAULT)

Transfer state to initial condition vector
  Transfer XIC to one of 3 storage vectors
  Retrieve XIC from one of 3 storage vectors
  End of file for mini-time-share file
APPENDIX C

ANALYSIS CHECKLISTS

Before requesting any of the EASY5 analyses, certain program commands should be issued to assure that the analysis will be successful. These program commands will place the system model in the proper configuration and complete the analysis specification. The following pages provide check lists of program commands that should be considered before requesting each analysis. The analyses are listed in alphabetical order.

LINEAR ANALYSIS

Model Data
  TITLE
  PARAMETER VALUES
  TABLES
  INITIAL CONDITIONS

Integrator Configuration
  INT CONTROL
  ERROR CONTROL

O.C. DESIGN

Model Data
  TITLE
  PARAMETER VALUES
  TABLES
  INITIAL CONDITIONS
  O.C. DATA: YOP,UOP,Q,RU,CD,CS

Integrator Configuration
  ALL STATES
  ERROR CONTROL
ROOT LOCUS

Model Data
TITLE
PARAMETER VALUES
TABLES
INITIAL CONDITIONS

Integration Configurations
INT CONTROL
ERROR CONTROL

Root Locus Specifications
RL PARAMETER
RL START
RL STOP
RL POINTS

Output Controls
RL MANUAL SCALES
RL AUTO SCALES
REAL MIN
REAL MAX
IMAG MIN
IMAG MAX

SCAN1, SCAN2

Model Data
PARAMETER VALUES
TITLE
PARAMETER VALUES
TABLES
INITIAL CONDITIONS
Scan Specifications
  DEPEN
  INDEP1
  INDEP2
  START1
  STOP1
  START2
  DELTA2
  CURVES2

  SIMULATE

Integration Control
  TINC
  TINC2
  TMAX
  INT MODE
  ERROR CONTROL
  INT CONTROL

Output Controls
  OUTRATE
  OUTRATE2
  PRATE
  PRATE2
  PRINT CONTROL
  PRINT2
  DISPLAY1, 2, 3, 4, 5
  PLOT ON
  PLOT TITLE
  PLOT ID
  SI MANUAL SCALES
  SI AUTO SCALES
  PRINTER PLOTS
  PRINT2 FROM, __, TO, __
STABILITY MARGINS

Model Data
TITLE
PARAMETER VALUES
TABLES
INITIAL CONDITIONS

Integration Configuration
INT CONTROL
ERROR CONTROL

Stability Margin Specification
SM PARAMETERS

STeady State

Model Data
TITLE
PARAMETER VALUES
TABLES
INITIAL CONDITIONS

Integration Configuration
INT CONTROL
ERROR CONTROL

Note: Steady state cannot be found for system with eigenvalue at origin.

Output Controls
PRINT CONTROL
DISPLAY 1, 2, 3, 4, 5, 6
PLOT ON
PRINTER PLOT
PLOT TITLE
PLOT ID
SS MANUAL SCALES
SS AUTO SCALES
Steady State Specifications
  SS PARAMETER
  SS START
  SS STOP
  SS POINTS
  SS ITERATIONS

TRANSFER FUNCTION

Model Data
  TITLE
  PARAMETER VALUES
  TABLES
  INITIAL CONDITIONS

Integrator Configuration
  INT CONTROL
  ERROR Control

Transfer Function Specification
  TF INPUT
  TF OUTPUT
  BODE, NICHOLS, NYQUIST

Output Controls
  TF MANUAL SCALES
  TF AUTO SCALES
  FREQ MIN
  FREQ MAX
This appendix contains input and output tables for all the EASIEST standard components. Descriptive figures are also presented for the more complex components.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>WT</td>
<td></td>
<td></td>
<td>WEIGHT OF THE ATTACHED BODY</td>
<td>LB</td>
</tr>
<tr>
<td>BMI(3)</td>
<td></td>
<td></td>
<td>ATTACHED BODY MOMENTS OF INERTIA (IXX, IYY, IZZ)</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>BPI(3)</td>
<td></td>
<td></td>
<td>ATTACHED BODY PRODUCTS OF INERTIA (IXY, IXZ, IYZ)</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>FAB(3)*</td>
<td>RS</td>
<td></td>
<td>X, Y, Z BODY AXIS FORCE COMPONENTS</td>
<td>LB</td>
</tr>
<tr>
<td>TAB(3)*</td>
<td>RS</td>
<td></td>
<td>X, Y, Z BODY AXIS TORQUE COMPONENTS</td>
<td>FT-LB</td>
</tr>
<tr>
<td>FAU(3)*</td>
<td></td>
<td></td>
<td>AUXILIARY X, Y, Z BODY AXIS FORCE COMPONENTS</td>
<td>LB</td>
</tr>
<tr>
<td>TAU(3)*</td>
<td></td>
<td></td>
<td>AUXILIARY X, Y, Z BODY AXIS TORQUE COMPONENTS</td>
<td>FT-LB</td>
</tr>
<tr>
<td>TRM(3)*</td>
<td>RS</td>
<td></td>
<td>X, Y, Z PARENT BODY EARTH VELOCITY COMPONENTS FOR CALCULATING THE LINEAR POSITION RATES DURING TRIM</td>
<td>FT/SEC</td>
</tr>
</tbody>
</table>

*Default value = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAB(3)*</td>
<td></td>
<td>X,Y,Z BODY AXIS LINEAR VELOCITY VECTOR OF THE ATTACHED BODY</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>XAB(3)*</td>
<td></td>
<td>X,Y,Z EARTH LINEAR POSITION VECTOR OF THE ATTACHED BODY</td>
<td>FT</td>
</tr>
<tr>
<td>WAB(3)*</td>
<td></td>
<td>X,Y,Z BODY AXIS ANGULAR VELOCITY VECTOR OF THE ATTACHED BODY</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>EAB(3)*</td>
<td></td>
<td>EARTH TO ATTACHED BODY EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
</tbody>
</table>

*These output quantities are states*
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AW</td>
<td></td>
<td></td>
<td>AIRPLANE WEIGHT</td>
<td>LB</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>WINGSPAN OF AIRPLANE</td>
<td>FT</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>MEAN AERODYNAMIC CHORD</td>
<td>FT</td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
<td>REFERENCE AREA</td>
<td>FT²</td>
</tr>
<tr>
<td>XCP</td>
<td></td>
<td></td>
<td>AIRPLANE X-AXIS POSITION OF THE CENTER OF PRESSURE</td>
<td>FT</td>
</tr>
<tr>
<td>AMI(3)</td>
<td></td>
<td></td>
<td>MOMENT OF INERTIA VECTOR ABOUT THE AIRPLANE C.G.</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(IXX, IYY, IZZ)</td>
<td></td>
</tr>
<tr>
<td>API(3)</td>
<td></td>
<td></td>
<td>PRODUCT OF INERTIA VECTOR ABOUT THE AIRPLANE C.G.</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(IXY, IXX, IYZ)</td>
<td></td>
</tr>
<tr>
<td>THR*</td>
<td></td>
<td></td>
<td>EXTERNAL THRUST SETTING</td>
<td>LB</td>
</tr>
<tr>
<td>AIL*</td>
<td></td>
<td></td>
<td>EXTERNAL AILERON SETTING</td>
<td>DEG</td>
</tr>
<tr>
<td>ELE*</td>
<td></td>
<td></td>
<td>EXTERNAL ELEVATOR SETTING</td>
<td>DEG</td>
</tr>
<tr>
<td>RUD*</td>
<td></td>
<td></td>
<td>EXTERNAL RUDDER SETTING</td>
<td>DEG</td>
</tr>
<tr>
<td>XEN(3)</td>
<td></td>
<td></td>
<td>X, Y, Z AIRPLANE BODY AXIS POSITION VECTOR OF THE</td>
<td>FT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ENGINE</td>
<td></td>
</tr>
<tr>
<td>END(3)</td>
<td></td>
<td></td>
<td>AIRPLANE BODY AXIS DIRECTION COSINES OF THE ENGINE</td>
<td>—</td>
</tr>
<tr>
<td>TAL</td>
<td></td>
<td></td>
<td>DESIRED TRIM AIRPLANE ALTITUDE</td>
<td>FT</td>
</tr>
<tr>
<td>TVE</td>
<td></td>
<td></td>
<td>DESIRED TRIM AIRPLANE VELOCITY</td>
<td>FT/SEC</td>
</tr>
</tbody>
</table>

*Default value = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRA(3)*</td>
<td>1</td>
<td>RL</td>
<td>X, Y, Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS</td>
<td>LB</td>
</tr>
<tr>
<td>TRA(3)*</td>
<td>1</td>
<td>RL</td>
<td>X, Y, Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS</td>
<td>FT-LB</td>
</tr>
<tr>
<td>FCA(3)*</td>
<td>1</td>
<td>CT</td>
<td>X, Y, Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT</td>
<td>LB</td>
</tr>
<tr>
<td>TCA(3)*</td>
<td>1</td>
<td>CT</td>
<td>X, Y, Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT</td>
<td>FT-LB</td>
</tr>
<tr>
<td>FDA(3)*</td>
<td>1</td>
<td>DR</td>
<td>X, Y, Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART</td>
<td>LB</td>
</tr>
<tr>
<td>TDA(3)*</td>
<td>1</td>
<td>DR</td>
<td>X, Y, Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART</td>
<td>FT-LB</td>
</tr>
<tr>
<td>FRA(3)*</td>
<td>2</td>
<td>RL</td>
<td>X, Y, Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS</td>
<td>LB</td>
</tr>
<tr>
<td>TRA(3)*</td>
<td>2</td>
<td>RL</td>
<td>X, Y, Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS</td>
<td>FT-LB</td>
</tr>
<tr>
<td>FCA(3)*</td>
<td>2</td>
<td>CT</td>
<td>X, Y, Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT</td>
<td>LB</td>
</tr>
</tbody>
</table>

*Default value = 0.*

198
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCA(3)*</td>
<td>2</td>
<td>CT</td>
<td>X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT</td>
<td>FT-LB</td>
</tr>
<tr>
<td>FDA(3)*</td>
<td>2</td>
<td>DR</td>
<td>X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART</td>
<td>LB</td>
</tr>
<tr>
<td>TDA(3)*</td>
<td>2</td>
<td>DR</td>
<td>X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART</td>
<td>FT-LB</td>
</tr>
<tr>
<td>CPF</td>
<td></td>
<td></td>
<td>PRINT FLAG FOR THE AERODYNAMIC COEFFICIENTS</td>
<td></td>
</tr>
</tbody>
</table>

*Default value = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAP(3)*</td>
<td></td>
<td>X,Y,Z AIRPLANE BODY AXIS LINEAR VELOCITY VECTOR OF THE AIRPLANE</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>XAP(3)*</td>
<td></td>
<td>X,Y,Z EARTH LINEAR POSITION VECTOR OF THE AIRPLANE</td>
<td>FT</td>
</tr>
<tr>
<td>WAP(3)*</td>
<td></td>
<td>X,Y,Z AIRPLANE BODY AXIS ANGULAR VELOCITY VECTOR OF THE AIRPLANE</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>EAP(3)*</td>
<td></td>
<td>EARTH TO AIRPLANE EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>TRM(4)*</td>
<td></td>
<td>TRIM CONTROL SETTINGS</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>1)</td>
<td>THROTTLE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2)</td>
<td>AILERON</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3)</td>
<td>ELEVATOR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4)</td>
<td>RUDDER</td>
<td></td>
</tr>
<tr>
<td>ALP</td>
<td></td>
<td>AIRPLANE ANGLE OF ATTACK</td>
<td>DEG</td>
</tr>
<tr>
<td>BET</td>
<td></td>
<td>AIRPLANE SIDESLIP ANGLE</td>
<td>DEG</td>
</tr>
<tr>
<td>VM</td>
<td></td>
<td>AIRPLANE MACH NUMBER</td>
<td>—</td>
</tr>
<tr>
<td>ALT</td>
<td></td>
<td>AIRPLANE ALTITUDE ABOVE SEA LEVEL</td>
<td>FT</td>
</tr>
</tbody>
</table>

*These output quantities are states*
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td></td>
<td>REFERENCE ALTITUDE WITH RESPECT TO SEA LEVEL</td>
<td>FT</td>
</tr>
<tr>
<td>WIN(3)</td>
<td></td>
<td>X,Y,Z INERTIAL SYSTEM WIND COMPONENTS</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>BP*</td>
<td></td>
<td>BAROMETRIC PRESSURE AT REFERENCE ALTITUDE</td>
<td>IN HG</td>
</tr>
<tr>
<td>TE</td>
<td></td>
<td>TEMPERATURE AT REFERENCE ALTITUDE</td>
<td>DEG F</td>
</tr>
<tr>
<td>SW**</td>
<td></td>
<td>GRAVITY SWITCH FOR UNSUPPORTED SEAT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = GRAVITY OFF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = GRAVITY ON</td>
<td></td>
</tr>
<tr>
<td>VS</td>
<td></td>
<td>VELOCITY OF SOUND</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>RHO</td>
<td></td>
<td>AIR DENSITY</td>
<td>SLUG/FT³</td>
</tr>
</tbody>
</table>

*Default value = 0
**Default value = 1

NOTE: H, BP, AND TE MUST BE INITIALIZED FOR A NON-STANDARD ATMOSPHERE. A STANDARD ATMOSPHERE IS ESTABLISHED WHEN BP EQUALS ZERO (DEFAULT)
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL</td>
<td></td>
<td></td>
<td>FLAG TO INITIATE AEROMED CALCULATIONS (1 = START)</td>
<td>—</td>
</tr>
<tr>
<td>PRT</td>
<td></td>
<td></td>
<td>FLAG SENT TO PROGRAM AEROMED TO PRINT THE AEROMEDICAL VARIABLES (1 = PRINT) <strong>DEFAULT = 0</strong></td>
<td>—</td>
</tr>
<tr>
<td>EXP</td>
<td></td>
<td></td>
<td>MEDICAL INJURY EXPONENT <strong>DEFAULT = 2</strong></td>
<td>—</td>
</tr>
<tr>
<td>GXP</td>
<td></td>
<td></td>
<td>THE LIMIT VALUE FOR THE X-AXIS POSITIVE AEROMED LOAD FACTOR <strong>DEFAULT = 35</strong></td>
<td>G's</td>
</tr>
<tr>
<td>GXN</td>
<td></td>
<td></td>
<td>THE LIMIT VALUE FOR THE X-AXIS NEGATIVE AEROMED LOAD FACTOR <strong>DEFAULT = 30</strong></td>
<td>G's</td>
</tr>
<tr>
<td>GYL</td>
<td></td>
<td></td>
<td>THE LIMIT VALUE FOR THE Y-AXIS AEROMED LOAD FACTOR <strong>DEFAULT = 15</strong></td>
<td>G's</td>
</tr>
<tr>
<td>GZL</td>
<td></td>
<td></td>
<td>THE LIMIT VALUE FOR THE Z-AXIS NEGATIVE AEROMED LOAD FACTOR <strong>DEFAULT = 12</strong></td>
<td>G's</td>
</tr>
<tr>
<td>DRP</td>
<td></td>
<td></td>
<td>LIMIT VALUE OF THE DYNAMIC RESPONSE WHEN THE ACCELERATION VECTOR IS FORWARD OF THE PLANE OF THE SEAT BACK <strong>DEFAULT = 18</strong></td>
<td>—</td>
</tr>
<tr>
<td>ORN</td>
<td></td>
<td></td>
<td>LIMIT VALUE OF THE DYNAMIC RESPONSE WHEN THE ACCELERATION VECTOR IS AFT OF THE PLANE OF THE SEAT BACK <strong>DEFAULT = 16</strong></td>
<td>—</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>NORMALLY DRIVEN BY</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>---------------------</td>
<td>------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>RDL</td>
<td></td>
<td></td>
<td>ACCELERATION RADICAL LIMIT</td>
<td>-</td>
</tr>
<tr>
<td>DR</td>
<td>SE or CE</td>
<td></td>
<td>DYNAMIC RESPONSE</td>
<td>-</td>
</tr>
<tr>
<td>GX</td>
<td>SE or CE</td>
<td></td>
<td>X-AXIS LOAD FACTOR</td>
<td>G's</td>
</tr>
<tr>
<td>GY</td>
<td>SE or CE</td>
<td></td>
<td>Y-AXIS LOAD FACTOR</td>
<td>G's</td>
</tr>
<tr>
<td>GZ</td>
<td>SE or CE</td>
<td></td>
<td>Z-AXIS LOAD FACTOR</td>
<td>G's</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>-----------------------------------------------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>DRE</td>
<td></td>
<td>DYNAMIC RESPONSE</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>RAD</td>
<td></td>
<td>ACCELERATION RADICAL</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>PTS</td>
<td></td>
<td>CURRENT NUMBER OF DATA SETS WRITTEN TO TAPE 7</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>PTI</td>
<td></td>
<td>VALUE OF TIME WHEN THE LAST DATA SET WAS WRITTEN TO TAPE 7</td>
<td>SEC</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO</td>
<td>NORMALLY DRIVEN BY</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>---------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>TCX</td>
<td></td>
<td></td>
<td>PLATE SYSTEM X-AXIS FORCE COEFFICIENT TABLE: PLATE ANGLE OF ATTACK (INDEPENDENT) PLATE X-AXIS COEFFICIENT (DEPENDENT)</td>
<td>DEG</td>
</tr>
<tr>
<td>TCZ</td>
<td></td>
<td></td>
<td>PLATE SYSTEM Z-AXIS FORCE COEFFICIENT TABLE: PLATE ANGLE OF ATTACK (INDEPENDENT) PLATE Z-AXIS COEFFICIENT (DEPENDENT)</td>
<td>DEG</td>
</tr>
<tr>
<td>UP**</td>
<td></td>
<td></td>
<td>EJECTION DIRECTION FLAG WITH RESPECT TO THE AIRPLANE 1 = UPWARD -1 = DOWNWARD</td>
<td></td>
</tr>
<tr>
<td>XPC(3)</td>
<td></td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE PLATE CENTROID</td>
<td>FT</td>
</tr>
<tr>
<td>PA</td>
<td></td>
<td></td>
<td>REFERENCE AREA OF THE ATTACHED PLATE</td>
<td>FT²</td>
</tr>
<tr>
<td>EPL(3)</td>
<td></td>
<td></td>
<td>SEAT TO PLATE EULER ANGLES</td>
<td>DEG</td>
</tr>
<tr>
<td>ZEM*</td>
<td></td>
<td></td>
<td>AIRPLANE BODY Z-AXIS POSITION OF THE PLATE CENTROID WHEN IT ENTERS THE WINDSTREAM</td>
<td>FT</td>
</tr>
<tr>
<td>SRP(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT</td>
</tr>
<tr>
<td>UST(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT/SEC</td>
</tr>
</tbody>
</table>

*Default value = 0.
**Default value = 1.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EST(3)</td>
<td>SE</td>
<td></td>
<td>EARTH TO SEAT EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WST(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>XAP(3)*</td>
<td>AE or SL</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE AIRPLANE CENTER OF GRAVITY</td>
<td>FT</td>
</tr>
<tr>
<td>EAP(3)*</td>
<td>AE or SL</td>
<td></td>
<td>EARTH TO AIRPLANE EULER ANGLES</td>
<td>DEG</td>
</tr>
</tbody>
</table>

*Default value = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE ATTACHED PLATE</td>
<td>LB</td>
</tr>
<tr>
<td>T2(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE ATTACHED PLATE</td>
<td>FT-LB</td>
</tr>
<tr>
<td>SW</td>
<td></td>
<td>FLAG SET WHEN THE PLATE CENTROID PENETRATES THE WINDSTREAM (1 = PENETRATION)</td>
<td>—</td>
</tr>
<tr>
<td>ALP</td>
<td></td>
<td>PLATE ANGLE OF ATTACK</td>
<td>DEG</td>
</tr>
<tr>
<td>CX</td>
<td></td>
<td>X-AXIS FORCE COEFFICIENT</td>
<td>—</td>
</tr>
<tr>
<td>CZ</td>
<td></td>
<td>Z-AXIS FORCE COEFFICIENT</td>
<td>—</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>NORMALLY DRIVEN BY</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>TAE</td>
<td></td>
<td></td>
<td>EXPOSED AREA TABLE:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EXPOSED LENGTH (INDEPENDENT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EXPOSED AREA (DEPENDENT)</td>
</tr>
<tr>
<td>OFF**</td>
<td>RL</td>
<td></td>
<td>FLAG/TO INDICATE SEAT/RAIL SEPARATION (1 = SEPARATION)</td>
</tr>
<tr>
<td>UP**</td>
<td></td>
<td></td>
<td>EJECTION DIRECTION FLAG WITH RESPECT TO THE AIRPLANE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+1 = UPWARD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-1 = DOWNWARD</td>
</tr>
<tr>
<td>ZWS*</td>
<td></td>
<td></td>
<td>AIRPLANE BODY Z-AXIS POSITION OF THE WINDSTREAM BOUNDARY LAYER AT THE POINT OF SEAT PENETRATION</td>
</tr>
<tr>
<td>XEM(3)*</td>
<td></td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE POINT ON THE SEAT TO INITIALLY PENETRATE THE WINDSTREAM</td>
</tr>
<tr>
<td>CDX*</td>
<td></td>
<td></td>
<td>APPROXIMATE SEAT BODY X-AXIS POSITION OF THE CENTER OF PRESSURE DURING EMERGENCE</td>
</tr>
<tr>
<td>ECX**</td>
<td></td>
<td></td>
<td>SEAT BODY X-AXIS EMERGENCE COEFFICIENT</td>
</tr>
<tr>
<td>ECM**</td>
<td></td>
<td></td>
<td>SEAT Y-AXIS EMERGENCE COEFFICIENT</td>
</tr>
<tr>
<td>ECZ**</td>
<td></td>
<td></td>
<td>SEAT Z-AXIS EMERGENCE COEFFICIENT</td>
</tr>
<tr>
<td>CLP*</td>
<td></td>
<td></td>
<td>ROLL DAMPING DERIVATIVE</td>
</tr>
<tr>
<td>CMQ*</td>
<td></td>
<td></td>
<td>PITCH DAMPING DERIVATIVE</td>
</tr>
<tr>
<td>CNR*</td>
<td></td>
<td></td>
<td>YAW DAMPING DERIVATIVE</td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
<td>SEAT REFERENCE AREA</td>
</tr>
</tbody>
</table>

*Default value = 0
**Default value = 1
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRP(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z EARTH LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT</td>
</tr>
<tr>
<td>UST(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EST(3)</td>
<td>SE</td>
<td></td>
<td>EARTH TO SEAT EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WST(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>DSA(3,3)*</td>
<td>RL</td>
<td></td>
<td>SEAT TO AIRPLANE DIRECTION COSINE MATRIX</td>
<td>–</td>
</tr>
<tr>
<td>SRA(3)*</td>
<td>RL</td>
<td></td>
<td>X,Y,Z AIRPLANE BODY AXIS LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT</td>
</tr>
<tr>
<td>RON*</td>
<td>SR</td>
<td></td>
<td>SUSTAINER ROCKET FLAG (1=ON 0=OFF)</td>
<td>–</td>
</tr>
</tbody>
</table>

*Default = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS AERODYNAMIC FORCE COMPONENTS</td>
<td>LB</td>
</tr>
<tr>
<td>T2(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS AERODYNAMIC TORQUE COMPONENTS</td>
<td>FT-LB</td>
</tr>
<tr>
<td>ALP</td>
<td></td>
<td>SEAT ANGLE OF ATTACK</td>
<td>DEG</td>
</tr>
<tr>
<td>BET</td>
<td></td>
<td>SEAT SIDESLIP ANGLE</td>
<td>DEG</td>
</tr>
<tr>
<td>VM</td>
<td></td>
<td>SEAT MACH NUMBER</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td></td>
<td>DYNAMIC PRESSURE</td>
<td>LB</td>
</tr>
<tr>
<td>CX</td>
<td></td>
<td>SEAT BODY X-AXIS FORCE COEFFICIENT</td>
<td></td>
</tr>
<tr>
<td>CY</td>
<td></td>
<td>SEAT BODY Y-AXIS FORCE COEFFICIENT</td>
<td></td>
</tr>
<tr>
<td>CZ</td>
<td></td>
<td>SEAT BODY Z-AXIS FORCE COEFFICIENT</td>
<td></td>
</tr>
<tr>
<td>CL</td>
<td></td>
<td>SEAT BODY AXIS ROLLING MOMENT COEFFICIENT</td>
<td></td>
</tr>
<tr>
<td>CM</td>
<td></td>
<td>SEAT BODY AXIS PITCHING MOMENT COEFFICIENT</td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td></td>
<td>SEAT BODY AXIS YAWING MOMENT COEFFICIENT</td>
<td></td>
</tr>
<tr>
<td>EXL</td>
<td></td>
<td>SEAT EXPOSED LENGTH DURING EMERGENCE</td>
<td>FT</td>
</tr>
<tr>
<td>EXA</td>
<td></td>
<td>SEAT EXPOSED AREA DURING EMERGENCE</td>
<td>FT²</td>
</tr>
<tr>
<td>CEN(3)</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE EMERGED AREA CENTROID</td>
<td>FT</td>
</tr>
<tr>
<td>TCZ(20)</td>
<td></td>
<td>SEAT Z-AXIS EXPOSED AREA CENTROID LOCATION ARRAY</td>
<td>FT</td>
</tr>
<tr>
<td>HD</td>
<td></td>
<td>HYDRAULIC DIAMETER</td>
<td>FT</td>
</tr>
</tbody>
</table>

211
STANDARD COMPONENT "AS" CALCULATES THE AERODYNAMIC FORCES AND TORQUES THAT ACT ON THE SEAT

Figure 23. Standard Component "AS" Input/Output Overview
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW*</td>
<td></td>
<td></td>
<td>FLAG FOR SEAT/CREWPERSON SEPARATION (1 = SEPARATION)</td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td></td>
<td>CREWPERSON PERCENTILE</td>
<td></td>
</tr>
<tr>
<td>CEW</td>
<td></td>
<td></td>
<td>WEIGHT OF THE CREWPERSON</td>
<td>LB</td>
</tr>
<tr>
<td>CMI(3)</td>
<td></td>
<td></td>
<td>CREWPERSON MOMENT OF INERTIA VECTOR ABOUT HIS C.G. (IXX, IYY, IZZ)</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>CPI(3)</td>
<td></td>
<td></td>
<td>CREWPERSON PRODUCT OF INERTIA VECTOR ABOUT HIS C.G. (IXY, IXZ, IYZ)</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>CLP</td>
<td></td>
<td></td>
<td>AERODYNAMIC ROLL DAMPING COEFFICIENT</td>
<td>1/DEG</td>
</tr>
<tr>
<td>CMQ</td>
<td></td>
<td></td>
<td>AERODYNAMIC PITCH DAMPING COEFFICIENT</td>
<td>1/DEG</td>
</tr>
<tr>
<td>CNR</td>
<td></td>
<td></td>
<td>AERODYNAMIC YAW DAMPING COEFFICIENT</td>
<td>1/DEG</td>
</tr>
<tr>
<td>XSP(3)*</td>
<td></td>
<td></td>
<td>X,Y,Z CREWPERSON SYSTEM POSITION VECTOR OF THE BASE OF THE SPINE</td>
<td>FT</td>
</tr>
<tr>
<td>FAB(3)*</td>
<td>RS</td>
<td></td>
<td>X,Y,Z BODY AXIS FORCE COMPONENTS</td>
<td>LB</td>
</tr>
<tr>
<td>TAB(3)*</td>
<td>RS</td>
<td></td>
<td>X,Y,Z BODY AXIS TORQUE COMPONENTS</td>
<td>FT-LB</td>
</tr>
<tr>
<td>FDO(3)*</td>
<td>LI</td>
<td></td>
<td>X,Y,Z BODY AXIS FORCE COMPONENTS</td>
<td>LB</td>
</tr>
<tr>
<td>TDO(3)*</td>
<td>LI</td>
<td></td>
<td>X,Y,Z BODY AXIS TORQUE COMPONENTS</td>
<td>FT-LB</td>
</tr>
</tbody>
</table>

*Default = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAU(3)*</td>
<td></td>
<td></td>
<td>X,Y,Z BODY AXIS FORCE COMPONENTS (AUXILIARY INPUT)</td>
<td>LB</td>
</tr>
<tr>
<td>TAU(3)*</td>
<td></td>
<td></td>
<td>X,Y,Z BODY AXIS TORQUE COMPONENT (AUXILIARY INPUT)</td>
<td>FT-LB</td>
</tr>
<tr>
<td>TRM(3)*</td>
<td></td>
<td></td>
<td>X,Y,Z PARENT BODY INERTIAL VELOCITY COMPONENTS TO DETERMINE POSITION RATES DURING TRIM</td>
<td>FT/SEC</td>
</tr>
</tbody>
</table>

*Default = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT  NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCP(3)*</td>
<td></td>
<td>X,Y,Z CREWPERSON BODY AXIS LINEAR VELOCITY VECTOR</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>XCP(3)*</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE CREWPERSON C.G.</td>
<td>FT</td>
</tr>
<tr>
<td>WCP(3)*</td>
<td></td>
<td>X,Y,Z CREWPERSON BODY AXIS ANGULAR VELOCITY VECTOR</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>ECP(3)*</td>
<td></td>
<td>EARTH TO CREWPERSON EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>SCD*</td>
<td></td>
<td>SPINAL COMPRESSION VELOCITY</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>SC*</td>
<td></td>
<td>SPINAL COMPRESSION</td>
<td>FT</td>
</tr>
<tr>
<td>GX</td>
<td></td>
<td>CREWPERSON X-AXIS LOAD FACTOR</td>
<td>G's</td>
</tr>
<tr>
<td>GY</td>
<td></td>
<td>CREWPERSON Y-AXIS LOAD FACTOR</td>
<td>G's</td>
</tr>
<tr>
<td>GZ</td>
<td></td>
<td>CREWPERSON Z-AXIS LOAD FACTOR</td>
<td>G's</td>
</tr>
<tr>
<td>DR</td>
<td></td>
<td>DYNAMIC RESPONSE</td>
<td>—</td>
</tr>
<tr>
<td>FAD(3)</td>
<td></td>
<td>X,Y,Z CREWPERSON BODY AXIS AERODYNAMIC FORCE COMPONENTS</td>
<td>LB</td>
</tr>
<tr>
<td>TAD(3)</td>
<td></td>
<td>X,Y,Z CREWPERSON BODY AXIS AERODYNAMIC TORQUE COMPONENTS</td>
<td>FT-LB</td>
</tr>
<tr>
<td>WT</td>
<td></td>
<td>WEIGHT OF THE CREWPERSON PLUS CLOTHING AND EQUIPMENT</td>
<td>LB</td>
</tr>
</tbody>
</table>

*These output quantities are states.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td>AERODYNAMIC REFERENCE AREA</td>
<td>FT²</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>AERODYNAMIC LATERAL REFERENCE LENGTH</td>
<td>FT</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>AERODYNAMIC LONGITUDINAL REFERENCE LENGTH</td>
<td>FT</td>
</tr>
<tr>
<td>CIN(4)</td>
<td></td>
<td>CREWPERSON INERTIA PROPERTIES AFTER SEAT CREWPERSON SEPARATION (IXX, IYY, IZZ, IXZ)</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>CX</td>
<td></td>
<td>X-AXIS AERODYNAMIC FORCE COEFFICIENT</td>
<td>-</td>
</tr>
<tr>
<td>CY</td>
<td></td>
<td>Y-AXIS AERODYNAMIC FORCE COEFFICIENT</td>
<td>-</td>
</tr>
<tr>
<td>CZ</td>
<td></td>
<td>Z-AXIS AERODYNAMIC FORCE COEFFICIENT</td>
<td>-</td>
</tr>
<tr>
<td>CL</td>
<td></td>
<td>AERODYNAMIC ROLLING MOMENT COEFFICIENT</td>
<td>-</td>
</tr>
<tr>
<td>CM</td>
<td></td>
<td>AERODYNAMIC PITCHING MOMENT COEFFICIENT</td>
<td>-</td>
</tr>
<tr>
<td>CN</td>
<td></td>
<td>AERODYNAMIC YAWING MOMENT COEFFICIENT</td>
<td>-</td>
</tr>
<tr>
<td>ALP</td>
<td></td>
<td>CREWPERSON ANGLE OF ATTACK</td>
<td>DEG</td>
</tr>
<tr>
<td>BET</td>
<td></td>
<td>CREWPERSON SIDESLIP ANGLE</td>
<td>DEG</td>
</tr>
<tr>
<td>VM</td>
<td></td>
<td>CREWPERSON MACH NUMBER</td>
<td>-</td>
</tr>
<tr>
<td>Q</td>
<td></td>
<td>DYNAMIC PRESSURE</td>
<td>LB/FT²</td>
</tr>
<tr>
<td>ALT</td>
<td></td>
<td>CREWPERSON ALTITUDE</td>
<td>FT</td>
</tr>
<tr>
<td>FL</td>
<td></td>
<td>SEAT/CREWPERSON SEPARATION FLAG FOR OUTPUT (1 = SEPARATION)</td>
<td>-</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>-----</td>
<td>----------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>COA*</td>
<td></td>
<td>AILERON COMMANDED POSITION</td>
<td>DEG</td>
</tr>
<tr>
<td>TCA*</td>
<td></td>
<td>AILERON TIME CONSTANT</td>
<td>SEC</td>
</tr>
<tr>
<td>TDA*</td>
<td></td>
<td>AILERON RESPONSE TIME DELAY</td>
<td>SEC</td>
</tr>
<tr>
<td>COE*</td>
<td></td>
<td>ELEVATOR COMMANDED POSITION</td>
<td>DEG</td>
</tr>
<tr>
<td>TCE*</td>
<td></td>
<td>ELEVATOR TIME CONSTANT</td>
<td>SEC</td>
</tr>
<tr>
<td>TDE*</td>
<td></td>
<td>ELEVATOR RESPONSE TIME DELAY</td>
<td>SEC</td>
</tr>
<tr>
<td>COR*</td>
<td></td>
<td>RUDDER COMMANDED POSITION</td>
<td>DEG</td>
</tr>
<tr>
<td>TCR*</td>
<td></td>
<td>RUDDER TIME DELAY</td>
<td>SEC</td>
</tr>
<tr>
<td>TDR*</td>
<td></td>
<td>RUDDER RESPONSE TIME DELAY</td>
<td>SEC</td>
</tr>
<tr>
<td>TRM(4)</td>
<td>AE</td>
<td>AIRPLANE CONTROL SURFACE POSITIONS AT TRIM</td>
<td>DEG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1) --NOT USED--</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) AILERON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3) ELEVATOR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4) RUDDER</td>
<td></td>
</tr>
</tbody>
</table>

*Default values = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIL*</td>
<td></td>
<td>AILERON DEFLECTION FROM ITS TRIM POSITION</td>
<td>DEG</td>
</tr>
<tr>
<td>ELE*</td>
<td></td>
<td>ELEVATOR DEFLECTION FROM ITS TRIM POSITION</td>
<td>DEG</td>
</tr>
<tr>
<td>RUD*</td>
<td></td>
<td>Rudder DEFLECTION FROM ITS TRIM POSITION</td>
<td>DEG</td>
</tr>
</tbody>
</table>

*These output quantities are states
**TCP**

CATAPULT PROPELLANT

CONSUMPTION TABLE:

- PROPELLANT WEB CONSUMED (INDEPENDENT)
- PROPELLANT CONSUMED (DEPENDENT)

**SW**

FLAG FOR CATAPULT IGNITION

(1 = CATAPULT ON)

**UP**

EJECTION DIRECTION FLAG

WITH RESPECT TO THE AIRPLANE

- +1 = UPWARD
- -1 = DOWNWARD

**SAP(3)**

X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE CATAPULT ATTACHMENT POINT ON THE SEAT

**AAP(3)**

X,Y,Z AIRPLANE BODY AXIS LINEAR POSITION VECTOR OF THE CATAPULT ATTACHMENT POINT ON THE AIRPLANE

**UCL**

UNLOADED CATAPULT LENGTH

**CSK**

CATAPULT STROKE

**VI**

INITIAL FREE VOLUME

**PA**

PISTON AREA

**PT**

TANG RELEASE PRESSURE

**CBP**

CATAPULT BURST PRESSURE

**C**

MASS OF TOTAL PROPELLANT

**CI**

IGNITER PROPELLANT MASS

**PMW**

PROPELLANT MOLECULAR WEIGHT

**SK**

CATAPULT SPRING CONSTANT

---

*Default value = 1.

---

**CT**

---

**219**
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK</td>
<td></td>
<td></td>
<td>CATAPULT DAMPING CONSTANT</td>
<td>LB/FT/SEC</td>
</tr>
<tr>
<td>CAM</td>
<td></td>
<td></td>
<td>RATIO OF SPECIFIC HEATS</td>
<td>—</td>
</tr>
<tr>
<td>TF</td>
<td></td>
<td></td>
<td>CONSTANT VOLUME FLAME TEMPERATURE</td>
<td>DEG K</td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td>FRICITION PROPORTIONALITY CONSTANT</td>
<td>LB/LB/IN$^2$</td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td>HEAT LOSS CONSTANT</td>
<td>—</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>NORMALLY DRIVEN BY</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>BURN RATE PROPORTIONALITY CONSTANT</td>
<td>IN/SEC/(LB/IN(^2))</td>
</tr>
<tr>
<td>BXP</td>
<td></td>
<td></td>
<td>BURN RATE EXPONENT</td>
<td>—</td>
</tr>
<tr>
<td>TI</td>
<td></td>
<td></td>
<td>CATAPULT TEMPERATURE PRIOR TO IGNITION</td>
<td>DEG K</td>
</tr>
<tr>
<td>TDE*</td>
<td></td>
<td></td>
<td>CATAPULT FORCE DECAY TIME</td>
<td>SEC</td>
</tr>
<tr>
<td>SRP(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT</td>
</tr>
<tr>
<td>UST(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EST(3)</td>
<td>SE</td>
<td></td>
<td>EARTH TO SEAT EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WST(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>XAP(3)</td>
<td>AE or SL</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE AIRPLANE</td>
<td>FT</td>
</tr>
<tr>
<td>UAP(3)</td>
<td>AE or SL</td>
<td></td>
<td>X,Y,Z AIRPLANE BODY AXIS LINEAR VELOCITY VECTOR OF THE AIRPLANE CENTER OF GRAVITY</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EAP(3)</td>
<td>AE or SL</td>
<td></td>
<td>EARTH TO AIRPLANE EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WAP(3)</td>
<td>AE or SL</td>
<td></td>
<td>X,Y,Z AIRPLANE BODY AXIS ANGULAR VELOCITY VECTOR OF THE AIRPLANE</td>
<td>DEG/SEC</td>
</tr>
</tbody>
</table>

*Default value = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EF*</td>
<td></td>
<td>INTERNAL FRICTION ENERGY</td>
<td>FT-LB</td>
</tr>
<tr>
<td>EL*</td>
<td></td>
<td>HEAT LOSS</td>
<td>FT-LB</td>
</tr>
<tr>
<td>WK*</td>
<td></td>
<td>CATAPULT WORK</td>
<td>FT-LB</td>
</tr>
<tr>
<td>WB*</td>
<td></td>
<td>PROPELLANT WEB CONSUMED</td>
<td>IN</td>
</tr>
<tr>
<td>FL</td>
<td></td>
<td>CATAPULT MODE FLAG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = PRIOR TO IGNITION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = CATAPULT IGNITION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = CATAPULT STRIPOFF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = CATAPULT OFF</td>
<td></td>
</tr>
<tr>
<td>FON</td>
<td></td>
<td>STRIPOFF FLAG FOR SUSTAINER ROCKET COMPONENT</td>
<td></td>
</tr>
<tr>
<td>FCA(3)</td>
<td>1</td>
<td>X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS OF THE CATAPULT ON THE AIRPLANE</td>
<td>LB</td>
</tr>
<tr>
<td>TCA(3)</td>
<td>1</td>
<td>X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS OF THE CATAPULT ON THE AIRPLANE</td>
<td>FT-LB</td>
</tr>
<tr>
<td>F1(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE CATAPULT ON THE SEAT</td>
<td>LB</td>
</tr>
<tr>
<td>T1(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE CATAPULT ON THE SEAT</td>
<td>FT-LB</td>
</tr>
<tr>
<td>CF</td>
<td></td>
<td>CATAPULT FORCE MAGNITUDE</td>
<td>LB</td>
</tr>
<tr>
<td>CEX</td>
<td></td>
<td>CATAPULT EXTENSION</td>
<td>FT</td>
</tr>
<tr>
<td>CV</td>
<td></td>
<td>CATAPULT EXTENSION VELOCITY</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>TLØ</td>
<td></td>
<td>INITIAL LENGTH OF THE CATAPULT PRESSURE CHAMBER</td>
<td>IN</td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td>CIRCUMFERENCE OF THE CATAPULT PRESSURE CHAMBER</td>
<td>IN</td>
</tr>
</tbody>
</table>

*These output quantities are states.*
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td></td>
<td>GAS CONSTANT</td>
<td>FT-LBF/SLUG-K</td>
</tr>
<tr>
<td>CVH</td>
<td></td>
<td>CONSTANT VOLUME SPECIFIC HEAT</td>
<td>FT-LBF/SLUG-K</td>
</tr>
<tr>
<td>TSO</td>
<td></td>
<td>CATAPULT STRIPOFF TIME</td>
<td>SEC</td>
</tr>
<tr>
<td>FSO</td>
<td></td>
<td>CATAPULT FORCE AT STRIPOFF</td>
<td>LB</td>
</tr>
</tbody>
</table>

223
Figure 24. Standard Component "CT" Input/Output Overview

- STANDARD COMPONENT "CT" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE AIRPLANE AND SEAT
- SUBROUTINE "CAD" COMPUTES THE CATAPULT FORCE
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBF</td>
<td></td>
<td></td>
<td>DART BRAKING FORCE TABLE:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LINE LENGTH (INDEPENDENT)</td>
<td>FT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BRAKING FORCE (DEPENDENT)</td>
<td>LB</td>
</tr>
<tr>
<td>DAP(3)</td>
<td></td>
<td></td>
<td>$X,Y,Z$ AIRPLANE BODY AXIS  LINEAR POSITION VECTOR OF THE DART ATTACHMENT POINT</td>
<td>FT</td>
</tr>
<tr>
<td>DBA(3)</td>
<td></td>
<td></td>
<td>$X,Y,Z$ SEAT BODY AXIS  LINEAR POSITION VECTOR OF THE DEPLOYED DART BRIDLE APEX</td>
<td>FT</td>
</tr>
<tr>
<td>XAP(3)</td>
<td>AE or SL</td>
<td></td>
<td>$X,Y,Z$ EARTH LINEAR POSITION VECTOR OF THE AIRPLANE CENTER OF GRAVITY</td>
<td>FT</td>
</tr>
<tr>
<td>EAP(3)</td>
<td>AE or SL</td>
<td></td>
<td>EARTH TO AIRPLANE EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>SRP(3)</td>
<td>SE</td>
<td></td>
<td>$X,Y,Z$ EARTH LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT</td>
</tr>
<tr>
<td>EST(3)</td>
<td>SE</td>
<td></td>
<td>EARTH TO SEAT EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>F2(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE DART ON THE SEAT</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>T2(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE DART ON THE SEAT</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>FDA(3)</td>
<td>1</td>
<td>X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS OF THE DART ON THE AIRPLANE</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>TDA(3)</td>
<td>1</td>
<td>X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS OF THE DART ON THE AIRPLANE</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>DLL</td>
<td></td>
<td>DISTANCE BETWEEN THE BRIDLE APEX AND THE AIRPLANE ATTACHMENT POINT</td>
<td>FT</td>
<td></td>
</tr>
<tr>
<td>DBF</td>
<td></td>
<td>DART BRAKING FORCE</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td></td>
<td>DART MODE FLAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = PRIOR TO DART</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = DART ON</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = DART OFF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- STANDARD COMPONENT "DR" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE AIRPLANE AND SEAT

- THE FIRST AND LAST ELEMENTS IN THE DART BRAKING FORCE TABLE (TBF) SHOULD BE THE FORCES AND THE APPROPRIATE LINE LENGTHS AT THE POINTS OF BRAKE INITIATION AND TERMINATION

- THE DART LINE LENGTH IS DEFINED AS THE DISTANCE FROM DBA TO DAP

Figure 25. Standard Component "DR" Input/Output Overview
<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMF</td>
<td>PARACHUTE MORTAR FORCE TABLE: TIME (INDEPENDENT VARIABLE)</td>
<td>SEC</td>
</tr>
<tr>
<td></td>
<td>MORTAR FORCE (DEPENDENT VARIABLE)</td>
<td>LB</td>
</tr>
<tr>
<td>SW</td>
<td>FLAG TO INITIATE MORTAR (1 = ON)</td>
<td>—</td>
</tr>
<tr>
<td>UV(3)</td>
<td>X,Y,Z SEAT BODY AXIS MORTAR FORCE DIRECTION UNIT VECTOR ACTING ON THE PARACHUTE PACK</td>
<td>—</td>
</tr>
<tr>
<td>XMO(3)</td>
<td>X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE DEPLOYMENT IMPULSE MOMENT ARM</td>
<td>FT</td>
</tr>
<tr>
<td>XYZ(3)</td>
<td>X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE PACK ATTACHMENT POINT</td>
<td>FT</td>
</tr>
<tr>
<td>EA(3)</td>
<td>SEAT TO PARACHUTE PACK ATTACHMENT ATTITUDE EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>XR</td>
<td>PARACHUTE SHELF LINEAR SPRING CONSTANT</td>
<td>LB/FT</td>
</tr>
<tr>
<td>XD</td>
<td>PARACHUTE SHELF LINEAR DAMPING CONSTANT</td>
<td>LB/FT/SEC</td>
</tr>
<tr>
<td>ER(3)</td>
<td>X,Y,Z PARACHUTE SHELF ANGULAR SPRING CONSTANTS</td>
<td>FT-LB/DEG</td>
</tr>
<tr>
<td>ED(3)</td>
<td>X,Y,Z PARACHUTE SHELF ANGULAR DAMPING CONSTANTS</td>
<td>FT-FT/DEG/SEC</td>
</tr>
<tr>
<td>TDE*</td>
<td>TIME DURATION FOR THE MORTAR FORCES AND TORQUES TO DECAY AFTER STRIPOFF</td>
<td>SEC</td>
</tr>
<tr>
<td>SRP(3)</td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT</td>
</tr>
</tbody>
</table>

*Default value = 0.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UST(3)</td>
<td>SE</td>
<td></td>
<td>$X,Y,Z$ SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EST(3)</td>
<td>SE</td>
<td></td>
<td>EARTH TO SEAT EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WST(3)</td>
<td>SE</td>
<td></td>
<td>$X,Y,Z$ SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>XPP(3)</td>
<td>PC</td>
<td></td>
<td>$X,Y,Z$ EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE PACK</td>
<td>FT</td>
</tr>
<tr>
<td>UPP(3)</td>
<td>PC</td>
<td></td>
<td>$X,Y,Z$ PARACHUTE PACK BODY AXIS LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EPP(3)</td>
<td>PC</td>
<td></td>
<td>EARTH TO PARACHUTE PACK EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WPP(3)</td>
<td>PC</td>
<td></td>
<td>$X,Y,Z$ PARACHUTE PACK BODY AXIS ANGULAR VELOCITY VECTOR OF THE PARACHUTE PACK</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>FL</td>
<td></td>
<td>PARACHUTE MODE FLAG:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = PRIOR TO INITIATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = PARACHUTE INITIATION UP TO LAUNCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = PARACHUTE LAUNCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = FORCES AND TORQUES OFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FMT</td>
<td></td>
<td>PARACHUTE MORTAR FORCE MAGNITUDE</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>F1</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS FORCE VECTOR ACTING ON THE SEAT</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>T1</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS TORQUE VECTOR ACTING ON THE SEAT</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>FPP(3)</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM FORCE VECTOR ACTING ON THE PARACHUTE PACK</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>TPP(3)</td>
<td></td>
<td>X,Y,Z PARACHUTE PACK BODY AXIS TORQUE VECTOR ACTING ON THE PARACHUTE PACK</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>TIN</td>
<td></td>
<td>PARACHUTE MORTAR INITIATION TIME</td>
<td>SEC</td>
<td></td>
</tr>
<tr>
<td>FSO(3)</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>TSO(3)</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>FPO(3)</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM FORCE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>TPO(3)</td>
<td></td>
<td>X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>TRM(3)</td>
<td></td>
<td>X,Y,Z SEAT INERTIAL VELOCITY COMPONENTS TO PASS TO THE PARACHUTE COMPONENT DURING TRIM</td>
<td>FT/SEC</td>
<td></td>
</tr>
</tbody>
</table>

230
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCW</td>
<td></td>
<td></td>
<td>STRETCHED CANOPY WEIGHT TABLE: STRETCHED LENGTH (INDEPENDENT) STRETCHED WEIGHT (DEPENDENT)</td>
<td>FT, LB</td>
</tr>
<tr>
<td>OFF*</td>
<td></td>
<td></td>
<td>FLAG TO SEVER LINES 0 = LINES ATTACHED 1 = LINES SEVERED</td>
<td></td>
</tr>
<tr>
<td>BLI</td>
<td></td>
<td></td>
<td>NUMBER OF BRIDLE LINES</td>
<td></td>
</tr>
<tr>
<td>APX(3)*</td>
<td></td>
<td></td>
<td>X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE BRIDLE APEX</td>
<td>FT</td>
</tr>
<tr>
<td>API(3)</td>
<td></td>
<td></td>
<td>X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE FIRST BRIDLE LINE ATTACHMENT POINT</td>
<td>FT</td>
</tr>
<tr>
<td>AP2(3)*</td>
<td></td>
<td></td>
<td>X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE SECOND BRIDLE LINE ATTACHMENT POINT</td>
<td>FT</td>
</tr>
<tr>
<td>AP3(3)*</td>
<td></td>
<td></td>
<td>X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE THIRD BRIDLE LINE ATTACHMENT POINT</td>
<td>FT</td>
</tr>
<tr>
<td>AP4(3)*</td>
<td></td>
<td></td>
<td>X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE FOURTH BRIDLE LINE ATTACHMENT POINT</td>
<td>FT</td>
</tr>
<tr>
<td>FTR</td>
<td></td>
<td></td>
<td>PARACHUTE LINE MULTIPLICATION FACTOR</td>
<td></td>
</tr>
<tr>
<td>FSO</td>
<td></td>
<td></td>
<td>CANOPY STRIPOUT FORCE</td>
<td>LB</td>
</tr>
<tr>
<td>ULL</td>
<td></td>
<td></td>
<td>PARACHUTE SUSPENSION LINE ULTIMATE LOAD</td>
<td>IN/IN</td>
</tr>
<tr>
<td>ULS</td>
<td></td>
<td></td>
<td>PARACHUTE SUSPENSION LINE ULTIMATE STRAIN</td>
<td>IN/IN</td>
</tr>
</tbody>
</table>

*Default value = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOR</td>
<td></td>
<td></td>
<td>NUMBER OF PARACHUTE GORES</td>
<td>—</td>
</tr>
<tr>
<td>TYP</td>
<td></td>
<td></td>
<td>TYPE OF PARACHUTE (1 = DRAG 2 = RECOVERY)</td>
<td>—</td>
</tr>
<tr>
<td>FL</td>
<td>MP or GP</td>
<td></td>
<td>MORTAR MODE FLAG 0 = PRIOR TO INITIATION 1 = INITIATION 2 = LAUNCH</td>
<td>—</td>
</tr>
<tr>
<td>XDO(3)</td>
<td>SE or CE</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE DECELERATED OBJECT</td>
<td>FT</td>
</tr>
<tr>
<td>UDO(3)</td>
<td>SE or CE</td>
<td></td>
<td>X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR VELOCITY VECTOR</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EDO(3)</td>
<td>SE or CE</td>
<td></td>
<td>EARTH TO DECELERATED OBJECT EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WDO(3)</td>
<td>SE or CE</td>
<td></td>
<td>X,Y,Z DECELERATED OBJECT BODY AXIS ANGULAR VELOCITY COMPONENTS</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>XPP(3)</td>
<td>PC</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE PACK</td>
<td>FT</td>
</tr>
<tr>
<td>UPP(3)</td>
<td>PC</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EPP(3)</td>
<td>PC</td>
<td></td>
<td>EARTH TO PARACHUTE PACK EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>XPC(3)</td>
<td>PC</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE CANOPY</td>
<td>FT</td>
</tr>
<tr>
<td>UPC(3)</td>
<td>PC</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE CANOPY</td>
<td>FT/SEC</td>
</tr>
</tbody>
</table>

232
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC*</td>
<td></td>
<td>CREEP STRAIN IN PARACHUTE LINES</td>
<td>IN/IN</td>
</tr>
<tr>
<td>TF*</td>
<td></td>
<td>TIME DURATION OF A NON-ZERO LOAD ON THE LINES</td>
<td>SEC</td>
</tr>
</tbody>
</table>
| FLA  |          | PARACHUTE PHASE  
0 = PRIOR TO INITIATION  
1 = INITIATION  
2 = LAUNCH  
3 = LINES STRETCH  
4 = LINES SEVERED |       |
| SWI  |          | FLAG SET WHEN PARACHUTE IS BEHIND THE BRIDLE APEX
(1 = BEHIND) |       |
| FDO(3)|          | X,Y,Z DECELERATED OBJECT BODY AXIS FORCE COMPONENTS | LB |
| TDO(3)|          | X,Y,Z DECELERATED OBJECT BODY AXIS TORQUE COMPONENTS | FT-LB |
| FLP(3)|          | X,Y,Z EARTH SYSTEM FORCE COMPONENTS ACTING ON THE PARACHUTE CANOPY | LB |
| FAP(3)|          | X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE FORCE APPLICATION POINT | FT |
| VAP(3)|          | X,Y,Z EARTH SYSTEM VELOCITY COMPONENTS OF THE FORCE APPLICATION POINT | FT/SEC |
| FLL  |          | LINE LOAD | LB |
| ELM  |          | MAXIMUM STRAIN EXPERIENCED BY THE PARACHUTE LINE DURING ITS LOADING HISTORY | IN/IN |

*These output quantities are states.*
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELC</td>
<td></td>
<td>MAXIMUM STRAIN EXPERIENCED BY THE PARACHUTE LINE DURING THE CURRENT LOADING CYCLE ONLY</td>
<td>IN/IN</td>
</tr>
<tr>
<td>DEM</td>
<td></td>
<td>MAXIMUM NEGATIVE STRAIN RATE EXPERIENCED BY THE PARACHUTE LINE DURING ITS LOADING HISTORY</td>
<td>1/SEC</td>
</tr>
<tr>
<td>RMN</td>
<td></td>
<td>MAXIMUM NEGATIVE STRAIN RATE EXPERIENCED BY THE PARACHUTE LINE DURING THE CURRENT UNLOADING CYCLE ONLY</td>
<td>1/SEC</td>
</tr>
<tr>
<td>DIS</td>
<td></td>
<td>THE DISTANCE FROM THE ORIGIN OF THE DECELERATED OBJECT TO THE BRIDLE APEX</td>
<td>FT</td>
</tr>
<tr>
<td>CON(4)</td>
<td></td>
<td>COEFFICIENTS IN THE EQUATION FOR THE PLANE FORMED BY THE BRIDLE ATTACHMENT POINTS</td>
<td>–</td>
</tr>
<tr>
<td>TCG(20)</td>
<td></td>
<td>STRETCHED CANOPY CENTER OF GRAVITY LOCATION ARRAY</td>
<td>FT</td>
</tr>
<tr>
<td>UVL(3)</td>
<td></td>
<td>PARACHUTE LINE UNIT VECTOR</td>
<td>–</td>
</tr>
<tr>
<td>RL</td>
<td></td>
<td>PARACHUTE LINE LENGTH</td>
<td>FT</td>
</tr>
<tr>
<td>RLO</td>
<td></td>
<td>UNLOADED PARACHUTE LINE LENGTH</td>
<td>FT</td>
</tr>
<tr>
<td>VL</td>
<td></td>
<td>RATE OF CHANGE OF LINE LENGTH</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>VCG</td>
<td></td>
<td>VELOCITY OF THE STRETCHED CANOPY CENTER OF GRAVITY ALONG THE LINES</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>PCG</td>
<td></td>
<td>STRETCHED CANOPY CENTER OF GRAVITY MEASURED ALONG THE PARACHUTE LINE FROM THE PARACHUTE PACK</td>
<td>FT</td>
</tr>
<tr>
<td>CWT</td>
<td></td>
<td>WEIGHT OF THE CANOPY PULLED FROM THE PARACHUTE PACK</td>
<td>LB</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>--------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>TPE</td>
<td></td>
<td>TYPE OF PARACHUTE</td>
<td>(1 = DRAG 2 = RECOVERY)</td>
</tr>
<tr>
<td>PVL</td>
<td></td>
<td>PREVIOUS TIMESTEP LINE VELOCITY</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>TLS</td>
<td></td>
<td>TIME AT LINESTRETCH</td>
<td>SEC</td>
</tr>
<tr>
<td>VLS</td>
<td></td>
<td>RATE OF CHANGE OF LINE LENGTH AT LINESTRETCH</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>NORMALLY DRIVEN BY</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>---------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td>FRICTION PROPORTIONALITY CONSTANT</td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td>HEAT LOSS CONSTANT</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>BURN RATE PROPORTIONALITY CONSTANT</td>
</tr>
<tr>
<td>BXP</td>
<td></td>
<td></td>
<td>BURN RATE EXPONENT</td>
</tr>
<tr>
<td>TI</td>
<td></td>
<td></td>
<td>MORTAR TEMPERATURE PRIOR TO IGNITION</td>
</tr>
<tr>
<td>TDE*</td>
<td></td>
<td></td>
<td>MORTAR FORCE DECAY TIME</td>
</tr>
<tr>
<td>SRP(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT</td>
</tr>
<tr>
<td>UST(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT</td>
</tr>
<tr>
<td>EST(3)</td>
<td>SE</td>
<td></td>
<td>EARTH TO SEAT EULER ANGLES</td>
</tr>
<tr>
<td>WST(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT</td>
</tr>
<tr>
<td>XPP(3)</td>
<td>PC</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE PACK</td>
</tr>
<tr>
<td>UPP(3)</td>
<td>PC</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK</td>
</tr>
<tr>
<td>EPP(3)</td>
<td></td>
<td></td>
<td>EARTH TO PARACHUTE PACK EULER ANGLES</td>
</tr>
<tr>
<td>WPP(3)</td>
<td></td>
<td></td>
<td>X,Y,Z PARACHUTE PACK BODY AXIS ANGULAR VELOCITY VECTOR OF THE PARACHUTE PACK</td>
</tr>
</tbody>
</table>

*Default value = 0
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW</td>
<td></td>
<td></td>
<td>FLAG TO INITIATE THE MORTAR (1 = ON)</td>
<td></td>
</tr>
<tr>
<td>XYZ(3)</td>
<td></td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE PACK ATTACHMENT POINT ON THE SEAT</td>
<td>FT</td>
</tr>
<tr>
<td>EA(3)</td>
<td></td>
<td></td>
<td>SEAT TO PARACHUTE PACK ATTACHMENT EULER ANGLES</td>
<td>DEG</td>
</tr>
<tr>
<td>XR</td>
<td></td>
<td></td>
<td>PARACHUTE SHELF LINEAR SPRING CONSTANT</td>
<td>LB/FT</td>
</tr>
<tr>
<td>XD</td>
<td></td>
<td></td>
<td>PARACHUTE SHELF LINEAR DAMPING CONSTANT</td>
<td>LB/FT/SEC</td>
</tr>
<tr>
<td>ER(3)</td>
<td></td>
<td></td>
<td>X,Y,Z PARACHUTE SHELF ANGULAR SPRING CONSTANT</td>
<td>FT-LB/DEG</td>
</tr>
<tr>
<td>ED(3)</td>
<td></td>
<td></td>
<td>X,Y,Z PARACHUTE SHELF ANGULAR DAMPING CONSTANT</td>
<td>FT-LB/DEG/SEC</td>
</tr>
<tr>
<td>UV(3)</td>
<td></td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS MORTAR FORCE UNIT VECTOR</td>
<td></td>
</tr>
<tr>
<td>CSK</td>
<td></td>
<td></td>
<td>MORTAR STROKE</td>
<td>FT</td>
</tr>
<tr>
<td>VI</td>
<td></td>
<td></td>
<td>INITIAL FREE VOLUME</td>
<td>IN³</td>
</tr>
<tr>
<td>PA</td>
<td></td>
<td></td>
<td>PISTON AREA</td>
<td>IN²</td>
</tr>
<tr>
<td>PT</td>
<td></td>
<td></td>
<td>TANG RELEASE PRESSURE</td>
<td>LB/IN²</td>
</tr>
<tr>
<td>CBP</td>
<td></td>
<td></td>
<td>MORTAR BURST PRESSURE</td>
<td>LB/IN²</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>MASS OF TOTAL PROPELLANT</td>
<td>SLUGS</td>
</tr>
<tr>
<td>CI</td>
<td></td>
<td></td>
<td>IGNITER PROPELLANT MASS</td>
<td>SLUGS</td>
</tr>
<tr>
<td>PMW</td>
<td></td>
<td></td>
<td>PROPELLANT MOLECULAR WEIGHT</td>
<td>LB/LB-MOLE</td>
</tr>
<tr>
<td>GAM</td>
<td></td>
<td></td>
<td>RATIO OF SPECIFIC HEATS</td>
<td></td>
</tr>
<tr>
<td>TF</td>
<td></td>
<td></td>
<td>CONSTANT VOLUME FLAME TEMPERATURE</td>
<td>DEG K</td>
</tr>
</tbody>
</table>

237
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EF*</td>
<td></td>
<td>INTERNAL FRICTION ENERGY</td>
<td>FT-LB</td>
</tr>
<tr>
<td>EL*</td>
<td></td>
<td>HEAT LOSS ENERGY</td>
<td>FT-LB</td>
</tr>
<tr>
<td>WK*</td>
<td></td>
<td>MORTAR WORK</td>
<td>FT-LB</td>
</tr>
<tr>
<td>WB*</td>
<td></td>
<td>PROPELLANT WEB BURNED</td>
<td>IN</td>
</tr>
<tr>
<td>FL</td>
<td></td>
<td>MORTAR MODE FLAG</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = PRIOR TO INITIATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = INITIATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = LAUNCH</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = MORTAR OFF</td>
<td></td>
</tr>
<tr>
<td>F1(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE MORTAR AND RESTRAINTS ON THE SEAT</td>
<td>LB</td>
</tr>
<tr>
<td>TI(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE MORTAR AND RESTRAINTS ON THE SEAT</td>
<td>FT-LB</td>
</tr>
<tr>
<td>FPP(3)</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM FORCE COMPONENTS OF THE MORTAR AND RESTRAINTS ON THE PARACHUTE PACK</td>
<td>LB</td>
</tr>
<tr>
<td>TPP(3)</td>
<td></td>
<td>X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS OF THE MORTAR AND RESTRAINTS ON THE PARACHUTE PACK</td>
<td>FT-LB</td>
</tr>
<tr>
<td>FM</td>
<td></td>
<td>MORTAR FORCE MAGNITUDE</td>
<td>LB</td>
</tr>
<tr>
<td>EXM</td>
<td></td>
<td>MORTAR EXTENSION</td>
<td>FT</td>
</tr>
<tr>
<td>VM</td>
<td></td>
<td>MORTAR EXTENSION VELOCITY</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>TSO</td>
<td></td>
<td>MORTAR STRIPOFF TIME</td>
<td>SEC</td>
</tr>
<tr>
<td>FSO</td>
<td></td>
<td>FORCE AT MORTAR STRIPOFF</td>
<td>LB</td>
</tr>
<tr>
<td>TRM(3)</td>
<td></td>
<td>X,Y,Z SEAT EARTH SYSTEM VELOCITY COMPONENTS TO PASS TO THE PARACHUTE COMPONENT DURING TRIM</td>
<td>FT/SEC</td>
</tr>
</tbody>
</table>

*These output quantities are states
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>STI</td>
<td></td>
<td></td>
<td>INFLATED PARACHUTE DRAG AREA</td>
<td>FT²</td>
</tr>
<tr>
<td>RCS</td>
<td></td>
<td></td>
<td>CIRCUMFERENCE OF THE FILLED CANOPY PLUS ONE QUARTER OF THAT DISTANCE</td>
<td>FT</td>
</tr>
<tr>
<td>RFM</td>
<td></td>
<td></td>
<td>REEF MODE FLAG</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = CHUTE NOT REEFED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = TIME OF DISREEF SET AT PARACHUTE INITIATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = TIME OF DISREEF SET AT LINESTRETCH</td>
<td></td>
</tr>
<tr>
<td>RFD</td>
<td></td>
<td></td>
<td>REEF DELAY TIME</td>
<td>SEC</td>
</tr>
<tr>
<td>RFS</td>
<td></td>
<td></td>
<td>PRODUCT OF REFERENCE AREA AND TANGENT FORCE COEFFICIENT WHEN REEFED</td>
<td>FT²</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>CONSTANT USED IN THE EQUATION THAT CALCULATES SCD OF THE REEFED PARACHUTE</td>
<td>-</td>
</tr>
<tr>
<td>CI</td>
<td></td>
<td></td>
<td>CONSTANT USED IN THE EQUATION TO COMPUTE THE CANOPY INFLATION TIME</td>
<td>-</td>
</tr>
<tr>
<td>CT(3)</td>
<td></td>
<td></td>
<td>CONSTANTS USED IN THE EQUATION THAT CALCULATES THE TANGENTIAL DRAG AREA</td>
<td>-</td>
</tr>
<tr>
<td>CN(3)</td>
<td></td>
<td></td>
<td>CONSTANTS USED IN THE EQUATION THAT CALCULATES THE NORMAL DRAG AREA</td>
<td>-</td>
</tr>
<tr>
<td>CM(2)</td>
<td></td>
<td></td>
<td>CONSTANTS USED IN THE MACH EFFECTS EQUATION</td>
<td>-</td>
</tr>
<tr>
<td>FD</td>
<td></td>
<td></td>
<td>WAKE TO FREE STREAM RATIO</td>
<td></td>
</tr>
<tr>
<td>PWT</td>
<td></td>
<td></td>
<td>TOTAL WEIGHT OF THE PARACHUTE PACK</td>
<td>LB</td>
</tr>
<tr>
<td>PMI(3)</td>
<td></td>
<td></td>
<td>PARACHUTE PACK MOMENTS OF INERTIA (IXX, IYY, IZZ)</td>
<td>SLUG-FT²</td>
</tr>
</tbody>
</table>

239
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPI(3)</td>
<td></td>
<td></td>
<td>PARACHUTE PACK PRODUCTS OF INERTIA (Ixy,Ixz,Iyz)</td>
<td>Slug-FT²</td>
</tr>
<tr>
<td>TEM*</td>
<td></td>
<td></td>
<td>TIME DURATION FOR PARACHUTE EMERGENCE</td>
<td>SEC</td>
</tr>
<tr>
<td>CSP**</td>
<td></td>
<td></td>
<td>PARACHUTE CANOPY SPRING CONSTANT</td>
<td>LB/FT</td>
</tr>
<tr>
<td>CDP***</td>
<td></td>
<td></td>
<td>PARACHUTE CANOPY DAMPING CONSTANT</td>
<td>LB/FT/SEC</td>
</tr>
<tr>
<td>FLA</td>
<td>LI</td>
<td></td>
<td>PARACHUTE MODE FLAG&lt;br&gt;0 = PRIOR TO INITIATION&lt;br&gt;1 = INITIATION&lt;br&gt;2 = LAUNCH&lt;br&gt;3 = LINESTRETCH&lt;br&gt;4 = LINES SEVERED</td>
<td>-</td>
</tr>
<tr>
<td>FLP(3)</td>
<td>LI</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM FORCE COMPONENTS ACTING ON THE PARACHUTE FROM THE LINES</td>
<td>LB</td>
</tr>
<tr>
<td>FPP(3)</td>
<td>GP or MP</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM FORCE COMPONENTS ACTING ON THE PACK FROM THE RESTRAINTS</td>
<td>LB</td>
</tr>
<tr>
<td>TPP(3)</td>
<td>GP or MP</td>
<td></td>
<td>X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS ACTING ON THE PACK FROM THE RESTRAINTS</td>
<td>FT-LB</td>
</tr>
<tr>
<td>VAP</td>
<td>LI</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM VELOCITY COMPONENTS OF THE FORCE APPLICATION POINT</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>UVL(3)</td>
<td>LI</td>
<td></td>
<td>EARTH SYSTEM PARACHUTE LINE UNIT VECTOR</td>
<td>-</td>
</tr>
</tbody>
</table>

*Default value = 0  
**Default value = 2000.  
***Default value = 14.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL</td>
<td>LI</td>
<td></td>
<td>PARACHUTE LINE LENGTH</td>
<td>FT</td>
</tr>
<tr>
<td>VCG(3)</td>
<td>LI</td>
<td></td>
<td>VELOCITY OF THE CANOPY CENTER OF GRAVITY ALONG THE PARACHUTE LINES</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>PCG</td>
<td>LI</td>
<td></td>
<td>STRETCHED CANOPY CENTER OF GRAVITY MEASURED ALONG THE PARACHUTE LINE FROM THE PARACHUTE PACK</td>
<td>FT</td>
</tr>
<tr>
<td>CWT</td>
<td>LI</td>
<td></td>
<td>WEIGHT OF THE CANOPY DRAWN FROM THE PACK</td>
<td>LB</td>
</tr>
<tr>
<td>TPE</td>
<td>LI</td>
<td></td>
<td>TYPE OF PARACHUTE 1 = DRAG 2 = RECOVERY</td>
<td>-</td>
</tr>
<tr>
<td>TRM(3)</td>
<td>GP or MP</td>
<td></td>
<td>X,Y,Z PARENT BODY EARTH SYSTEM VELOCITY COMPONENTS TO DETERMINE THE POSITION RATES DURING TRIM</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>---------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>UPP(3)*</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK CENTER OF GRAVITY</td>
<td>FT/SEC</td>
<td></td>
</tr>
<tr>
<td>XPP(3)*</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE PACK CENTER OF GRAVITY</td>
<td>FT</td>
<td></td>
</tr>
<tr>
<td>WPP(3)*</td>
<td></td>
<td>X,Y,Z PARACHUTE PACK BODY AXIS ANGULAR VELOCITY VECTOR</td>
<td>DEG/SEC</td>
<td></td>
</tr>
<tr>
<td>EPP(3)*</td>
<td></td>
<td>EARTH TO PARACHUTE PACK EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
<td></td>
</tr>
<tr>
<td>UPC(3)*</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE CANOPY</td>
<td>FT/SEC</td>
<td></td>
</tr>
<tr>
<td>XPC(3)*</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE CANOPY</td>
<td>FT</td>
<td></td>
</tr>
<tr>
<td>PHA</td>
<td></td>
<td>PARACHUTE PHASE</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = PRIOR TO PARACHUTE LAUNCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = FROM LAUNCH UP TO LINESTRETCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = AFTER LINESTRETCH</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*These output quantities are states*
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW</td>
<td></td>
<td>FLAG TO INDICATE PARACHUTE AERODYNAMIC CALCULATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MODE:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = PRIOR TO LAUNCH</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = FROM PARACHUTE LAUNCH TO LINESTRETCH</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = DURING INFLATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = DURING REEFING</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 = AFTER REEFING</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 = PARACHUTE INFLATED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>FLI(3)</strong> X,Y,Z EARTH SYSTEM LB AERODYNAMIC LIFT COMPONENTS</td>
<td>LB</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>FDR(3)</strong> X,Y,Z EARTH SYSTEM LB AERODYNAMIC DRAG COMPONENTS</td>
<td>LB</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>FMA(3)</strong> X,Y,Z EARTH SYSTEM LB FORCE COMPONENTS ACTING ON THE CANOPY DUE TO</td>
<td>LB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AIR MASS ACQUISITION FORCE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>RM</strong> RADIUS OF THE SPHERE REPRESENTING THE INFLATED CANOPY</td>
<td>FT</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VOL</strong> VOLUME OF THE FILLED CANOPY</td>
<td>FT³</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TLA</strong> PARACHUTE LAUNCH TIME OR LINE SEVERING TIME</td>
<td>SEC</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TLS</strong> LINESTRETCH TIME</td>
<td>SEC</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TDS</strong> TIME AT WHICH DISREEF OCCURS</td>
<td>SEC</td>
</tr>
</tbody>
</table>

*Acting on the pack before linestretch  
Acting on the canopy after linestretch
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTI</td>
<td></td>
<td>PARACHUTE CANOPY INFLATION TIME</td>
<td>SEC</td>
</tr>
<tr>
<td>TDU</td>
<td></td>
<td>TIME DURATION OF REEFED PARACHUTE</td>
<td>SEC</td>
</tr>
<tr>
<td>TRF</td>
<td></td>
<td>TIME AT WHICH THE CHUTE IS REEFED</td>
<td>SEC</td>
</tr>
</tbody>
</table>
STANDARD COMPONENT "PC" CALCULATES THE ANGULAR AND LINEAR RATES FOR THE PARACHUTE PACK AND CANOPY

STRETCHED CANOPY CENTER OF GRAVITY (EARTH REFERENCE FRAME)

FORCE APPLICATION POINT (SEAT REFERENCE FRAME)

EARTH COORDINATE SYSTEM

Figure 26. Standard Component "PC" Input/Output Overview
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL1(3)</td>
<td></td>
<td></td>
<td>$X,Y,Z$ SEAT AXIS POSITION VECTOR OF THE RIGHT LOWER BLOCK</td>
<td>FT</td>
</tr>
<tr>
<td>BL2(3)</td>
<td></td>
<td></td>
<td>$X,Y,Z$ SEAT AXIS POSITION VECTOR OF THE RIGHT MIDDLE BLOCK</td>
<td>FT</td>
</tr>
<tr>
<td>BL3(3)</td>
<td></td>
<td></td>
<td>$X,Y,Z$ SEAT AXIS POSITION VECTOR OF THE RIGHT UPPER BLOCK</td>
<td>FT</td>
</tr>
<tr>
<td>BL4(3)</td>
<td></td>
<td></td>
<td>$X,Y,Z$ SEAT AXIS POSITION VECTOR OF THE LEFT LOWER BLOCK</td>
<td>FT</td>
</tr>
<tr>
<td>BL5(3)</td>
<td></td>
<td></td>
<td>$X,Y,Z$ SEAT AXIS POSITION VECTOR OF THE LEFT MIDDLE BLOCK</td>
<td>FT</td>
</tr>
<tr>
<td>BL6(3)</td>
<td></td>
<td></td>
<td>$X,Y,Z$ SEAT AXIS POSITION VECTOR OF THE LEFT UPPER BLOCK</td>
<td>FT</td>
</tr>
</tbody>
</table>
| UP    |          |                     | EJECTION DIRECTION FLAG  
+1 = UPWARD WRT THE AIRPLANE  
-1 = DOWNWARD WRT THE AIRPLANE |       |
| RLR   |          |                     | RIGHT RAIL Z COORDINATE OF THE END OF THE RIGHT RAIL                        | FT    |
| XRR(3)|          |                     | $X,Y,Z$ AIRPLANE POSITION VECTOR OF THE ORIGIN OF THE RIGHT RAIL COORDINATE SYSTEM | FT    |
| RLL   |          |                     | LEFT RAIL Z COORDINATE OF THE END OF THE LEFT RAIL                           | FT    |

246
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>XRL(3)</td>
<td></td>
<td></td>
<td>X,Y,Z AIRPLANE POSITION VECTOR OF THE ORIGIN OF THE LEFT RAIL COORDINATE SYSTEM</td>
<td>FT</td>
</tr>
<tr>
<td>ERL(3)</td>
<td></td>
<td></td>
<td>AIRPLANE TO RAILS EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>SPR(2)</td>
<td></td>
<td></td>
<td>X,Y RAIL SPRING CONSTANTS</td>
<td>LB/FT</td>
</tr>
<tr>
<td>DPG (2)</td>
<td></td>
<td></td>
<td>X,Y RAIL DAMPING CONSTANTS</td>
<td>LB/FT/SEC</td>
</tr>
<tr>
<td>SBF</td>
<td></td>
<td></td>
<td>SLIDER BLOCK FRICTION COEFFICIENT</td>
<td>-</td>
</tr>
<tr>
<td>ZTS</td>
<td></td>
<td></td>
<td>RIGHT RAIL AXIS Z COORDINATE OF THE KEY BLOCK AT TRIP SWITCH CONTACT</td>
<td>FT</td>
</tr>
<tr>
<td>BTS</td>
<td></td>
<td></td>
<td>TRIP SWITCH KEY BLOCK NUMBER</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = BOTTOM RIGHT BLOCK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = MIDDLE RIGHT BLOCK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 = TOP RIGHT BLOCK</td>
<td></td>
</tr>
<tr>
<td>CPT(3)</td>
<td></td>
<td></td>
<td>X,Y,Z AIRPLANE POSITION VECTOR OF THE CRITICAL CLEARANCE POINT</td>
<td>FT</td>
</tr>
<tr>
<td>SRP(3)</td>
<td></td>
<td>SE</td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT</td>
</tr>
<tr>
<td>UST(3)</td>
<td></td>
<td>SE</td>
<td>X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EST(3)</td>
<td></td>
<td>SE</td>
<td>EARTH TO SEAT EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WST(3)</td>
<td></td>
<td>SE</td>
<td>X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>XAP(3)</td>
<td></td>
<td>AE or SL</td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE AIRPLANE</td>
<td>FT</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>NORMALLY DRIVEN BY</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>UAP(3)</td>
<td>AE or SL</td>
<td></td>
<td>X,Y,Z AIRPLANE BODY AXIS LINEAR VELOCITY VECTOR OF THE AIRPLANE</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EAP(3)</td>
<td>AE or SL</td>
<td></td>
<td>EARTH TO AIRPLANE EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WAP(3)</td>
<td>AE or SL</td>
<td></td>
<td>X,Y,Z AIRPLANE BODY AXIS ANGULAR VELOCITY VECTOR</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>F2(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS ON THE SEAT FROM THE RAILS</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>T2(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS ON THE SEAT FROM THE RAILS</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>FR1(3)</td>
<td>1</td>
<td>X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS ON THE AIRPLANE FROM THE RAILS</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>TR1(3)</td>
<td>1</td>
<td>X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ON THE AIRPLANE FROM THE RAILS</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>FL</td>
<td></td>
<td>STROKE FLAG - 0 = GUIDED 1 = UNGUIDED</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>FTS</td>
<td></td>
<td>TRIP SWITCH CONTACT FLAG (1 = ON)</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>TTS</td>
<td></td>
<td>TRIP SWITCH CONTACT TIME</td>
<td>SEC</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td></td>
<td>SEAT/RAIL SEPARATION FLAG (1 = SEPARATION)</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>DSA (3,3)</td>
<td></td>
<td>SEAT TO AIRPLANE DIRECTION COSINE MATRIX</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SRA(3)</td>
<td></td>
<td>X,Y,Z AIRPLANE COORDINATE SYSTEM LINEAR POSITION VECTOR OF THE SRP</td>
<td>FT</td>
<td></td>
</tr>
<tr>
<td>DIS</td>
<td></td>
<td>DISTANCE FROM THE CRITICAL POINT TO THE SEAT REFERENCE POINT</td>
<td>FT</td>
<td></td>
</tr>
<tr>
<td>TM(3)</td>
<td></td>
<td>X,Y,Z EARTH VELOCITY COMPONENTS OF THE VEHICLE TO PASS TO THE SEAT DURING TRIM</td>
<td>FT/SEC</td>
<td></td>
</tr>
</tbody>
</table>
- STANDARD COMPONENT "RL" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE AIRPLANE AND SEAT FROM THE RAILS
- THE FORCES AND TORQUES THAT ARE CALCULATED ARE A FUNCTION OF THE LINEAR DISPLACEMENT OF THE BLOCKS FROM THE RAILS
- EACH RAIL HAS A COORDINATE SYSTEM ATTACHED TO IT

**Figure 27. Standard Component "RL" Input/Output Overview**
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL</td>
<td></td>
<td></td>
<td>FLAG TO RELEASE ATTACHED BODY (1 = RELEASE)</td>
<td>-</td>
</tr>
<tr>
<td>XYZ(3)</td>
<td></td>
<td></td>
<td>X,Y,Z PARENT BODY AXIS LINEAR POSITION VECTOR OF THE ATTACHMENT POINT</td>
<td>FT</td>
</tr>
<tr>
<td>EA(3)</td>
<td></td>
<td></td>
<td>PARENT BODY TO ATTACHED BODY ATTACHMENT POSITION EULER ANGLES (YAW,PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>XPB(3)</td>
<td></td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARENT BODY</td>
<td>FT</td>
</tr>
<tr>
<td>UPB(3)</td>
<td></td>
<td></td>
<td>X,Y,Z PARENT BODY AXIS LINEAR VELOCITY VECTOR OF THE PARENT BODY</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EPB(3)</td>
<td></td>
<td></td>
<td>EARTH TO PARENT BODY EULER ANGLES (YAW,PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WPB(3)</td>
<td></td>
<td></td>
<td>X,Y,Z BODY AXIS ANGULAR VELOCITY VECTOR OF THE PARENT BODY</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>XAB(3)</td>
<td></td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE ATTACHED BODY</td>
<td>FT</td>
</tr>
<tr>
<td>UAB(3)</td>
<td></td>
<td></td>
<td>X,Y,Z ATTACHED BODY AXIS LINEAR VELOCITY VECTOR OF THE ATTACHED BODY</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EAB(3)</td>
<td></td>
<td></td>
<td>EARTH TO ATTACHED BODY EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>WAB(3)</td>
<td></td>
<td></td>
<td>X,Y,Z ATTACHED BODY AXIS ANGULAR VELOCITY VECTOR OF THE ATTACHED BODY</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>NORMALLY DRIVEN BY</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>---------------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>XR</td>
<td></td>
<td></td>
<td>LINEAR SPRING CONSTANT</td>
<td>LB/FT</td>
</tr>
<tr>
<td>XD</td>
<td></td>
<td></td>
<td>LINEAR DAMPING CONSTANT</td>
<td>LB/FT/SEC</td>
</tr>
<tr>
<td>ER(3)</td>
<td></td>
<td></td>
<td>X,Y,Z ANGULAR SPRING CONSTANT</td>
<td>FT-LB/DEG</td>
</tr>
<tr>
<td>ED(3)</td>
<td></td>
<td></td>
<td>X,Y,Z ANGULAR DAMPING CONSTANT</td>
<td>FT-LB/DEG/SEC</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>------------------------------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>FPB(3)</td>
<td></td>
<td>X,Y,Z PARENT BODY AXIS FORCE VECTOR</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>TPB(3)</td>
<td></td>
<td>X,Y,Z PARENT BODY AXIS TORQUE VECTOR</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>FAB(3)</td>
<td></td>
<td>X,Y,Z ATTACHED BODY AXIS FORCE VECTOR</td>
<td>LB'</td>
<td></td>
</tr>
<tr>
<td>TAB(3)</td>
<td></td>
<td>X,Y,Z ATTACHED BODY AXIS TORQUE VECTOR</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>TRM(3)</td>
<td></td>
<td>X,Y,Z PARENT BODY EARTH SYSTEM VELOCITY COMPONENTS TO PASS TO THE ATTACHED BODY</td>
<td>FT/SEC</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>NORMALLY DRIVEN BY</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>F1(3)*</td>
<td>1 TO 9</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS GENERATED BY A PYROTECHNIC DEVICE</td>
<td>LBS</td>
</tr>
<tr>
<td>T1(3)*</td>
<td>1 TO 9</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS ABOUT THE SRP GENERATED BY A PYROTECHNIC DEVICE</td>
<td>FT-LBS</td>
</tr>
<tr>
<td>F2(3)*</td>
<td>1 TO 9</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS GENERATED BY A NON-PYROTECHNIC DEVICE</td>
<td>LBS</td>
</tr>
<tr>
<td>T2(3)*</td>
<td>1 TO 9</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS ABOUT THE SRP GENERATED BY A NON-PYROTECHNIC DEVICE</td>
<td>FT-LBS</td>
</tr>
<tr>
<td>CW</td>
<td>WB</td>
<td></td>
<td>COMPOSITE WEIGHT OF THE SEAT</td>
<td>LBS</td>
</tr>
<tr>
<td>CCG(3)</td>
<td>WB</td>
<td></td>
<td>X,Y,Z SEAT AXIS SYSTEM COMPOSITE CENTER OF GRAVITY</td>
<td>FT</td>
</tr>
<tr>
<td>CMI(3)</td>
<td>WB</td>
<td></td>
<td>MOMENT OF INERTIA VECTOR ABOUT THE SEAT REFERENCE POINT FOR THE COMPOSITE SEAT (IXY, IYY, IZZ)</td>
<td>SLUG-FT2</td>
</tr>
<tr>
<td>CPI(3)</td>
<td>WB</td>
<td></td>
<td>PRODUCT OF INERTIA VECTOR ABOUT THE SEAT REFERENCE POINT FOR THE COMPOSITE SEAT (IXY, IXZ, IYZ)</td>
<td>SLUG-FT2</td>
</tr>
<tr>
<td>TM(3)</td>
<td>RL</td>
<td></td>
<td>X,Y,Z VEHICLE EARTH VELOCITY COMPONENTS TO DETERMINE THE POSITION RATE DURING TRIM</td>
<td>FT/SEC</td>
</tr>
</tbody>
</table>

* Default = 0.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UST(3)*</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>SRP(3)*</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT</td>
<td>FT</td>
</tr>
<tr>
<td>WST(3)*</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>EST(3)*</td>
<td></td>
<td>EARTH TO SEAT EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>ALT</td>
<td></td>
<td>SEAT ALTITUDE</td>
<td>FT</td>
</tr>
</tbody>
</table>

* These output quantities are states.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UD(3)*</td>
<td></td>
<td>X,Y,Z SLED SYSTEM LINEAR VELOCITY RATE VECTOR</td>
<td>FT/SEC/SEC</td>
</tr>
<tr>
<td>WD(3)*</td>
<td></td>
<td>X,Y,Z SLED SYSTEM ANGULAR VELOCITY RATE VECTOR</td>
<td>DEG/SEC/SEC</td>
</tr>
</tbody>
</table>

*Default value = 0.

<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAP(3)*</td>
<td></td>
<td>X,Y,Z SLED BODY AXIS LINEAR VELOCITY COMPONENTS</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>XAP(3)*</td>
<td></td>
<td>X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SLED</td>
<td>FT</td>
</tr>
<tr>
<td>WAP(3)*</td>
<td></td>
<td>X,Y,Z SLED BODY AXIS ANGULAR VELOCITY COMPONENTS</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>EAP(3)*</td>
<td></td>
<td>EARTH TO SLED EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
</tbody>
</table>

*These output quantities are states.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRF</td>
<td></td>
<td></td>
<td>ROCKET TABLE:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TIME (INDEPENDENT)</td>
<td>SEC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FORCE (DEPENDENT)</td>
<td>LBS</td>
</tr>
<tr>
<td>TMA</td>
<td></td>
<td></td>
<td>MECHANICAL ADVANTAGE TABLE:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GIMBAL ANGLE (INDEPENDENT)</td>
<td>DEG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MECHANICAL ADVANTAGE (DEPENDENT)</td>
<td>-</td>
</tr>
<tr>
<td>TST</td>
<td></td>
<td></td>
<td>SPRING MOMENT TABLE:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GIMBAL ANGLE (INDEPENDENT)</td>
<td>DEG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPRING TORQUE (DEPENDENT)</td>
<td>FT-LBS</td>
</tr>
<tr>
<td>FL</td>
<td></td>
<td></td>
<td>STAPAC IGNITION FLAG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1 = STAPAC ON)</td>
<td></td>
</tr>
<tr>
<td>YPR</td>
<td></td>
<td></td>
<td>STAPAC APPLICATION FLAG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = YAW STAPAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = PITCH STAPAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 = ROLL STAPAC</td>
<td></td>
</tr>
<tr>
<td>AVW</td>
<td></td>
<td></td>
<td>ANGULAR VELOCITY OF GYROSCOPE WHEEL</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>WMI</td>
<td></td>
<td></td>
<td>MOMENT OF INERTIA OF THE WHEEL ABOUT ITS SPIN AXIS</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>SMI</td>
<td></td>
<td></td>
<td>MOMENT OF INERTIA OF THE SYSTEM LESS THE ROCKET ABOUT THE GIMBAL AXIS</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>RII</td>
<td></td>
<td></td>
<td>MOMENT OF INERTIA OF THE ROCKET PRIOR TO IGNITION</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>RIF</td>
<td></td>
<td></td>
<td>MOMENT OF INERTIA OF THE ROCKET AFTER BURNOUT</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>XR(3)</td>
<td></td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE ROCKET NOZZLE</td>
<td>FT</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DRIVEN BY</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>UV(3)**</td>
<td></td>
<td></td>
<td>X,Y,Z ROCKET FORCE UNIT VECTOR IN THE ROCKET COORDINATE SYSTEM</td>
<td>-</td>
</tr>
<tr>
<td>GSA</td>
<td></td>
<td></td>
<td>GIMBAL MOTION STOP IN THE NEGATIVE ROLL DIRECTION (MEASURED FROM THE CAGED POSITION)</td>
<td>DEG</td>
</tr>
<tr>
<td>GSF</td>
<td></td>
<td></td>
<td>GIMBAL MOTION STOP IN THE POSITIVE ROLL DIRECTION (MEASURED FROM THE CAGED POSITION)</td>
<td>DEG</td>
</tr>
<tr>
<td>SPR</td>
<td></td>
<td></td>
<td>GIMBAL STOP ANGULAR RIGIDITY</td>
<td>FT-LB/DEG</td>
</tr>
<tr>
<td>DPG</td>
<td></td>
<td></td>
<td>GIMBAL STOP ANGULAR DAMPING</td>
<td>FT-LB/DEG/SEC</td>
</tr>
<tr>
<td>FMT</td>
<td></td>
<td></td>
<td>LOAD AT MAXIMUM FRICTION</td>
<td>LBS</td>
</tr>
<tr>
<td>TMX</td>
<td></td>
<td></td>
<td>MAXIMUM FRICTION</td>
<td>FT-LB</td>
</tr>
<tr>
<td>TNF</td>
<td></td>
<td></td>
<td>FRICTION AT NO THRUST</td>
<td>FT-LB</td>
</tr>
<tr>
<td>TOS</td>
<td></td>
<td></td>
<td>THRUSTLINE OFFSET</td>
<td>FT</td>
</tr>
<tr>
<td>TSU*</td>
<td></td>
<td></td>
<td>GYROSCOPE WHEEL SPINUP TIME (SEC)</td>
<td>SEC</td>
</tr>
<tr>
<td>GMA*</td>
<td></td>
<td></td>
<td>GIMBAL ANGULAR VELOCITY AT MAXIMUM FRICTION</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>WST(3)</td>
<td>SE</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT</td>
<td>DEG/SEC</td>
</tr>
</tbody>
</table>

*Defaults = 0.
**Defaults: UV(1) = 0.
   UV(2) = 0.
   UV(3) = -1.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>WG*</td>
<td></td>
<td>GIMBAL SYSTEM X-AXIS ANGULAR VELOCITY</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>ESG(3)*</td>
<td></td>
<td>SEAT TO GIMBAL EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>ESR(3)*</td>
<td></td>
<td>SEAT TO ROCKET EULER ANGLES (YAW,PITCH,ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>PHA</td>
<td></td>
<td>STAPAC OPERATIONAL PHASE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = BEFORE IGNITION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = STAPAC IGNITION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = STAPAC BURNOUT</td>
<td></td>
</tr>
<tr>
<td>F1(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF STAPAC ON THE SEAT</td>
<td>LB</td>
</tr>
<tr>
<td>T1(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF STAPAC ON THE SEAT</td>
<td>FT-LB</td>
</tr>
<tr>
<td>TIN</td>
<td></td>
<td>TIME AT STAPAC INITIATION</td>
<td>SEC</td>
</tr>
<tr>
<td>ECA</td>
<td></td>
<td>SEAT TO GIMBAL ROLL EULER ANGLE AT THE CAGED POSITION</td>
<td>DEG</td>
</tr>
</tbody>
</table>

*These output quantities are states.*
- STANDARD COMPONENT "SP" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE SEAT FROM STAPAC.

- THE EULER ANGLES FOR THE GYROSCOPE AND THE ROCKET ARE STATES WITH RESPECT TO THE SEAT COORDINATE SYSTEM.

NOTE: The rocket thrust unit vector, $\mu V(3)$, is with respect to the rocket coordinate system. (Default shown in figure)

**Figure 28. Standard Component "SP" Input/Output Overview**
Figure 29. Standard Component "SP" Gimbal Spring and Vernier Rocket
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRF</td>
<td></td>
<td></td>
<td>ROCKET TABLE: TIME (INDEPENDENT) FORCE (DEPENDENT)</td>
<td>SEC LBS</td>
</tr>
<tr>
<td>FON</td>
<td>CT</td>
<td></td>
<td>SUSTAINER IGNITION FLAG (1 = ROCKET ON)</td>
<td>-</td>
</tr>
<tr>
<td>PCG(3)</td>
<td></td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE PROPELLANT CENTER OF GRAVITY</td>
<td>FT</td>
</tr>
<tr>
<td>EA(3)</td>
<td></td>
<td></td>
<td>SEAT TO ROCKET PROPELLANT EULER ANGLES (YAW, PITCH, ROLL)</td>
<td>DEG</td>
</tr>
<tr>
<td>XRN(3)</td>
<td></td>
<td></td>
<td>X,Y,Z PROPELLANT SYSTEM POSITION VECTOR OF THE ROCKET NOZZLE</td>
<td>FT</td>
</tr>
<tr>
<td>YAW</td>
<td></td>
<td></td>
<td>YAW EULER ANGLE OF THE THRUST VECTOR IN THE PROPELLANT COORDINATE SYSTEM</td>
<td>DEG</td>
</tr>
<tr>
<td>PIT</td>
<td></td>
<td></td>
<td>PITCH EULER ANGLE OF THE THRUST VECTOR IN THE PROPELLANT COORDINATE SYSTEM</td>
<td>DEG</td>
</tr>
<tr>
<td>PL</td>
<td></td>
<td></td>
<td>PROPELLANT GRAIN LENGTH</td>
<td>FT</td>
</tr>
<tr>
<td>POD</td>
<td></td>
<td></td>
<td>PROPELLANT GRAIN OUTSIDE DIAMETER</td>
<td>FT</td>
</tr>
<tr>
<td>PID</td>
<td></td>
<td></td>
<td>PROPELLANT GRAIN INSIDE DIAMETER</td>
<td>FT</td>
</tr>
<tr>
<td>NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>W*</td>
<td>1</td>
<td>WEIGHT OF UNBURNED PROPELLANT</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>PHA</td>
<td></td>
<td>ROCKET PHASE</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = BEFORE IGNITION</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = ROCKET BURN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = ROCKET OFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RON</td>
<td></td>
<td>ROCKET ON FLAG</td>
<td>(1 = ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = OFF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS FORCE COMPONENTS</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>T1(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS</td>
<td>FT-LB</td>
<td></td>
</tr>
<tr>
<td>X(3)</td>
<td>1</td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE PROPELLANT CENTER OF GRAVITY</td>
<td>FT</td>
<td></td>
</tr>
<tr>
<td>BM(3)</td>
<td>1</td>
<td>X,Y,Z UNBURNED ROCKET PROPELLANT MOMENTS OF INERTIA (IXX,IXY,IZZ)</td>
<td>SLUG-FT²</td>
<td></td>
</tr>
<tr>
<td>BP(3)</td>
<td>1</td>
<td>UNBURNED ROCKET PROPELLANT PRODUCTS OF INERTIA (IXY,IXZ,IFY)</td>
<td>SLUG-FT²</td>
<td></td>
</tr>
<tr>
<td>FR</td>
<td></td>
<td>SUSTAINER ROCKET FORCE MAGNITUDE</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>PWI</td>
<td></td>
<td>INITIAL WEIGHT OF THE PROPELLANT</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>SPI</td>
<td></td>
<td>ROCKET PROPELLANT SPECIFIC IMPULSE</td>
<td>LB-SEC/LB</td>
<td></td>
</tr>
<tr>
<td>RHO</td>
<td></td>
<td>ROCKET PROPELLANT DENSITY</td>
<td>LB/FT³</td>
<td></td>
</tr>
<tr>
<td>VWI</td>
<td></td>
<td>INITIAL VIRTUAL WEIGHT</td>
<td>LB</td>
<td></td>
</tr>
<tr>
<td>TMI(3)</td>
<td></td>
<td>PROPELLANT MOMENTS OF INERTIA AS IF IT WERE A SOLID GRAIN</td>
<td>SLUG-FT²</td>
<td></td>
</tr>
<tr>
<td>TIG</td>
<td></td>
<td>ROCKET IGNITION TIME</td>
<td>SEC</td>
<td></td>
</tr>
</tbody>
</table>

*This output quantity is a state.*
NOTE: The yaw and pitch euler angles of the rocket nozzle are with respect to the propellant coordinate system. The thrust vector acts in the negative Z direction with respect to the nozzle coordinate system.

- STANDARD COMPONENT "SR" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE SEAT FROM THE SUSTAINER ROCKET
- UPDATED INERTIAL PROPERTIES ARE FED TO STANDARD COMPONENT "WB" (WEIGHT AND BALANCE)

![Diagram](image_url)

*Figure 30. Standard Component "SR" Input/Output Overview*
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td></td>
<td></td>
<td>NUMBER OF ATTACHED BODIES</td>
<td>-</td>
</tr>
<tr>
<td>SW</td>
<td></td>
<td></td>
<td>BASIC SEAT WEIGHT</td>
<td>LB</td>
</tr>
<tr>
<td>SX(3)</td>
<td></td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE BASIC SEAT CENTER OF GRAVITY</td>
<td>FT</td>
</tr>
<tr>
<td>SM(3)</td>
<td></td>
<td></td>
<td>MOMENT OF INERTIA VECTOR ABOUT THE C.G. FOR THE BASIC SEAT (IXX, IYY, IZZ)</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>SP(3)</td>
<td></td>
<td></td>
<td>PRODUCT OF INERTIA VECTOR ABOUT THE C.G. FOR THE BASIC SEAT (IXY, IXZ, IYZ)</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>W*</td>
<td>1</td>
<td>SR</td>
<td>WEIGHT OF BODY ONE</td>
<td>LB</td>
</tr>
<tr>
<td>X(3)*</td>
<td>1</td>
<td>SR</td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE CENTER OF GRAVITY FOR BODY ONE</td>
<td>FT</td>
</tr>
<tr>
<td>BM(3)*</td>
<td>1</td>
<td>SR</td>
<td>MOMENT OF INERTIA VECTOR FOR BODY ONE TRANSFORMED INTO THE SEAT SYSTEM (IXX, IYY, IZZ)</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>BP(3)*</td>
<td>1</td>
<td>SR</td>
<td>PRODUCT OF INERTIA VECTOR FOR BODY ONE TRANSFORMED INTO THE SEAT SYSTEM (IXY, IXZ, IYZ)</td>
<td>SLUG-FT²</td>
</tr>
<tr>
<td>W*</td>
<td>2</td>
<td></td>
<td>WEIGHT OF BODY TWO</td>
<td>LB</td>
</tr>
<tr>
<td>X(3)*</td>
<td>2</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE CENTER OF GRAVITY FOR BODY TWO</td>
<td>FT</td>
</tr>
<tr>
<td>BM(3)*</td>
<td>2</td>
<td></td>
<td>MOMENT OF INERTIA VECTOR FOR BODY TWO TRANSFORMED INTO THE SEAT SYSTEM (IXX, IYY, IZZ)</td>
<td>SLUG-FT²</td>
</tr>
</tbody>
</table>

*Default value = 0.*
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>NORMALLY DRIVEN BY</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP(3)*</td>
<td>2</td>
<td></td>
<td>PRODUCT OF INERTIA VECTOR FOR BODY TWO TRANSFORMED INTO THE SEAT SYSTEM (IXY,IXZ,IYZ)</td>
<td>Slug-FT²</td>
</tr>
<tr>
<td>W*</td>
<td>3</td>
<td></td>
<td>WEIGHT OF BODY THREE</td>
<td>LB</td>
</tr>
<tr>
<td>X(3)*</td>
<td>3</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE CENTER OF GRAVITY FOR BODY THREE</td>
<td>FT</td>
</tr>
<tr>
<td>BMI(3)*</td>
<td>3</td>
<td></td>
<td>MOMENT OF INERTIA VECTOR FOR BODY THREE TRANSFORMED INTO THE SEAT SYSTEM (IXX,IYY,IZZ)</td>
<td>Slug-FT²</td>
</tr>
<tr>
<td>BP(3)*</td>
<td>3</td>
<td></td>
<td>PRODUCT OF INERTIA VECTOR FOR BODY THREE TRANSFORMED INTO THE SEAT SYSTEM (IXY,IXZ,IYZ)</td>
<td>Slug-FT²</td>
</tr>
</tbody>
</table>

*Default Value = 0.

Note - All moments and products of inertial must be rotated into the seat coordinate system.
<table>
<thead>
<tr>
<th>NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CW</td>
<td></td>
<td>COMPOSITE WEIGHT OF THE SEAT</td>
<td>LB</td>
</tr>
<tr>
<td>CCG(3)</td>
<td></td>
<td>X,Y,Z SEAT BODY AXIS</td>
<td>FT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COMPOSITE CENTER OF GRAVITY</td>
<td></td>
</tr>
<tr>
<td>CMI(3)</td>
<td></td>
<td>MOMENT OF INERTIA VECTOR</td>
<td>Slug-FT²</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABOUT THE C.G. FOR THE COMPOSITE SEAT (IXX, IYY, IZZ)</td>
<td></td>
</tr>
<tr>
<td>CPI(3)</td>
<td></td>
<td>PRODUCT OF INERTIA VECTOR</td>
<td>Slug-FT²</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABOUT THE C.G. FOR THE COMPOSITE SEAT (IXY, IXZ, IYZ)</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX E

PROGRAM AEROMED

This appendix contains program AEROMED, the aeromedical post processor.
PROGRAM AEROMED (OUTPUT, TAPE7, TAPE6=OUTPUT)
C
DIMENSION TIME(4000), GX(4000), GY(4000), GZ(4000), DR(4000),
- RAD(4000), RADXY(4000), RADZ(4000)
C
NOTICE...........
C
A PORTION OF THIS PROGRAM EVALUATES ACCELERATION DATA
C ESSENTIALLY IN ACCORDANCE WITH ACCEPTED AEROMEDICAL
C PROCEDURES. THESE KINDS OF RESULTS ARE THEN NORMALLY USED
C TOGETHER WITH OTHER FACTORS TO DETERMINE ACCEPTABILITY OF
C ACCELERATION LOADS APPLIED TO THE EJECTEE.
C
THE EVALUATION METHOD HAS BEEN ADOPTED HERE TO SERVE AS A
C FOUNDATION FOR CREATING FIGURES OF MERIT RELATED TO THE
C PERFORMANCE OF A PARTICULAR ESCAPE SYSTEM CONFIGURATION.
C THERE IS THEN AN OPPORTUNITY TO COMPARE, BY STANDARD MEANS,
C THE PERFORMANCE OF ONE CONFIGURATION WITH THAT OF OTHERS
C AND COMPARED IN TERMS OF THE MOST FUNDAMENTAL AND CRITICAL
C PARAMETERS OF PERFORMANCE OF ANY ESCAPE SYSTEM, NAMELY,
C ACCELERATION LOADS ON THE HUMAN EJECTEE.
C
THIS PROGRAM THEREFORE SERVES AS AN ENGINEERING TOOL ONLY
C AND SHOULD NOT BE CONSIDERED AN ACCEPTABLE AEROMEDICAL
C EVALUATION TOOL TO MEASURE ACCEPTABILITY OF AN ESCAPE SYSTEM
C FOR SAFE OPERATIONAL USE.
C
********** INITIALIZATION **********
C
GXMAX = GYMAX = GZMAX = DRMAX = ROMAX = -10.
GAMIN = GYMIN = GZMIN = DRMIN = ROMIN = 10.
TINJURY = EXPERNC = 0
C
***** READ THE AEROMED PARAMETERS FROM TAPE7 *****
C
REWIND 7
READ(?,10) PRT, EXP, GXN, GYL, GZR, URP, DRN, RDL
10 FORMAT(9F12.4)
C
***** READ THE AEROMED VARIABLES FROM TAPE7 *****
C
NPTS = 4000
I = 0
20 I = I + 1
IF(I.GT.4000) GOTO 35
READ(?,4000) TIME(I), DR(I), GX(I), GY(I), GZ(I)
C
IF(EOF(7)) 30, 25
C
25 GX(I) = -GX(I)
GY(I) = -GY(I)
GOTO 20
C
30 NPTS = I - 1
C
***** CALCULATE RADXY AND RADZ *****
C
35 DO 40 I=1, NPTS

269
GXL = GXP 
IF(GX(I).LT.0.0) GXL = GXN 
DRL = DRP 
IF(GX(I).LT.0.0) DRL = DRN 
RADXY(I) = (GX(I)/GXL)**2 + (GY(I)/GYL)**2 
40 RADZ(I) = (DR(I)/DRL)**2 
C 
L ***** CALCULATE THE Z-AXIS TOLERANCE RATIO FOR EACH WINDOW ***** 
C 
N = 0 
50 N = N + 1 
I = 0 
60 J = N + 1 
C 
IF(TIME(J).GT.TIME(N)+0.063) GO TO 70 
IF(GZ(J).GE.0.) RADZMAX = AMAX1(RADZ(J),RADZMAX) 
IF(GZ(J).LT.0.) RADZMAX = AMAX1((GZ(J)/GZL)**2,RADZMAX) 
IF(J.EQ.00PTS) GO TO 90 
I = 1 + 1 
GO TO 60 
C 
C DETERMINE THE ACCELERATION RADICAL ...... 
C 
70 RAD(N) = SQRT(RADZMAX + RADXY(N)) 
C 
C UPDATE THE UNSAFE LOAD EXPERIENCE FACTOR ...... 
C 
IF(RDL.LE.RAD(N)) GO TO 80 
TINJURY = TINJURY + (RAD(N)-RDL)**EXP * (TIME(N+1)-TIME(N)) 
C 
C UPDATE THE TOTAL LOAD EXPERIENCE FACTOR ...... 
C 
80 EXPERNC = EXPERNC + RAD(N)**EXP * (TIME(N+1)-TIME(N)) 
C 
GO TO 50 
90 N = N - 1 
TIME = TIME(N) 
C 
C ***** CALCULATE THE SAFE LOAD EXPERIENCE FACTOR ***** 
C 
TINJURY = TINJURY/TIME(N) 
EXPERNC = EXPERNC/TIME(N) 
THREAT = EXPERNC - TINJURY 
C 
C ***** CALCULATE THE MAXIMUM AND MINIMUM AEROMEDICAL VARIABLES ***** 
C 
DU 100 I=1,N 
C 
GXMAX = AMAX1(GXMAX,GX(I)) 
IF(GXMAX.EQ.GX(I)) GXMAXT = TIME(I) 
GYMAX = AMAX1(GYMAX,GY(I)) 
IF(GYMAX.EQ.GY(I)) GYMAXT = TIME(I) 
GZMAX = AMAX1(GZMAX,GZ(I)) 
IF(GZMAX.EQ.GZ(I)) GZMAXT = TIME(I) 
DRMAX = AMAX1(DRMAX,DR(I)) 
IF(DRMAX.EQ.DR(I)) DRMAXT = TIME(I) 
ROMAX = AMAX1(ROMAX,RD(I)) 
270
IF(KDMAX.EQ.RAD(I)) RDMAXT = TIME(I)

C

GXMIN = AMIN1(GXMIN,GX(I))
IF(GXMIN.EQ.GX(I)) GXMIINT = TIME(I)
GYMIN = AMIN1(GYMIN,GY(I))
IF(GYMIN.EQ.GY(I)) GYMINT = TIME(I)
GZMIN = AMIN1(GZMIN,GZ(I))
100 IF(GZMIN.EQ.GL(I)) GZMIINT = TIME(I)

C

***** WRITE TO THE OUTPUT FILE *****

C

WRITE (6,110) TMAX
110 FORMAT (1H1, *HUMAN TOLERANCE ANALYSIS THROUGH *,F10.3,
* SECONDS OF THE SIMULATION*/*, *
* AEROMEDICAL CONVENTION *//, *
* GX = +ACCEL, GY = +ACCEL, GZ = -ACCEL *///)

C

IF(PRT.EQ.1.) WRITE(6,120)
120 FORMAT (1X,*TIME*,9X,*GX*,10X,*GY*,10X,*GZ*,10X,*DR1*,8X,*RAD*,//,
1X,F7.3,4F12.2,F11.2)

C

IF(PRT.EQ.1.) WRITE(6,130) (TIME(I),GX(I),GY(I),GZ(I),JR(I),
RAD(I),I=1,N)

30 FORMAT(1X,F7.3,4F12.2,F11.2)

C

WRITE(6,140) GMAX,GXMAX,GYMAX,GZMAX,GMAX,GZMAX,
GMIN,GXMIN,GYMIN,GZMIN,GZMIN,
DRMAX,RDMAX,DRM,
140 FORMAT (2(1H0/),*,GAMAX = *,F14.2,*, TIME = *,F14.3,/,
* GYMAX = *,F14.2,*, TIME = *,F14.3,/,
* GZMAX = *,F14.2,*, TIME = *,F14.3,/,
* GDMIN = *,F14.2,*, TIME = *,F14.3,/,
* GYMIN = *,F14.2,*, TIME = *,F14.3,/,
* GZMIN = *,F14.2,*, TIME = *,F14.3,/,
* DRMINT = *,F13.2,*, TIME = *,F14.3,/,
* RDMAX = *,F13.2,*, TIME = *,F14.3)

C

WRITE (6,150) EXPERIENCE,THREAT,INJURY
150 FORMAT (2(1H0/),*, FIGURES OF MERIT.....................*,///,
* EXPERIENCE FACTOR - TOTAL LOAD = *,
*F14.3,/,
* EXPERIENCE FACTOR - SAFE LOAD = *,
*F14.3,/,
* EXPERIENCE FACTOR - UNSAFE LOAD = *,
*F14.3)

C

END
This appendix contains listings of the EASIEST procedure files. The procedure for attaching these files and submitting an EASIEST run is given in Section V.
EASIEST PROCEDURE FILE - LATEST REVISION DEC 12, 1980

* THIS FILE CONTAINS THE CCL PROCEDURES REQUIRED TO EXECUTE AND MAINTAIN THE EASIEST CREW ESCAPE SIMULATION PROGRAM

* PROCEDURE DIRECTORY

* SUBRUN - PROCEDURE TO SUBMIT A BATCH EASIEST RUN
DBFMOD - PROCEDURE TO MODIFY THE EASIEST DATA BASE FILE
COMPILE - PROCEDURE TO COMPIL e A SINGLE EASIEST COMPONENT
COMPALL - PROCEDURE TO COMPIL e AN ENTIRE SOURCE LIBRARY
EZSTGEN - PROCEDURE TO GENERATE EASIEST FROM DELIVERY TAPE

* SEE THE EASIEST MANUAL FOR COMPLETE USAGE INFORMATION

* EOR

.PROC, SUBRUN, MODFILE, ANLFILE, TIME=100, INOUT=100,
CORE=1150000, IDENT=EZ5, COEF=0, NOLIST=OUTPUT/0, AEROMED=0/YES.
RETURN, JOB, PF, MODFILE, ANFILE.
REQUEST, JOB, *Q.
COPYCR, JOBFILE, JOB.
ATTACH, MODFILE.
COPYCF, MODFILE, JOB.
ATTACH, ANLFILE.
COPYCF, ANLFILE, JOB.
ROUTE(JOB, DC=IN, TID=Z1, ST=CSA)
RETURN, MODFILE, ANLFILE, JOB, JOBFILE.

.DATA, JOBF ILE
IDENT, TIME, IO, INOUT, CM, CORE. D790183, CREW ESCAPE EASIEST JOB
ATTACH(COMPLIB, MR=1)
ATTACH(EZSTLIB, MR=1)
LIBRARY(EZSTLIB, COMPLIB)
COPYCF, INPUT, MODEL.
REWIND, MODEL.
ATTACH(EASY5, MR=1)
ATTACH(TAPE78=EZSTDBF, MR=1)
MAP(OFF)
LDSET(PRESET=ZERO)
EASY5(MODEL)
RETURN(MODEL, EASY5, EASY, TAPE78, TAPE7, TAPE8, TAPE10, TAPE11, TAPE12)
REWIND(TAPE9)
RFL, CORE.
FTN(I=TAPE9, B=EZFORT, R=2, EL=F, L=NOLIST, ROUND)
COPYCF, INPUT, ANFIL.
REWIND, ANFIL.
RETURN(TAPE3)
IF() NOT.NUM(COE F), NOAIRP.
ATTACH(TAPE3=COEF, MR=1)
ENDIF_NOAIRP.
REWIND(EZFORT)
ATTACH(NONSIM5, MR=1)
COPYLM(NONSIM5, EZFORT, NONSIMT)
REWIND(NONSIMT)
RETURN(EZFORT,NONSIM5,MAPFILE)
LDSET(PRESET=ZERO,MAP=SB/MAPFILE)
NONSIMT(ANFIL)
SKIP,NOMAP.
EXIT,U.
REWIND,MAPFILE.
COPYCF,MAPFILE,OUTPUT.
EXIT.
ENDIF,NOMAP.
IFE,.NOT.NUM(AEROMED),NOAERO
REWIND,TAPE7.
ATTACH(AROMEDB,MR=1)
LDSET(PRESET=ZERO)
AROMEDB.
RETURN,AROMEDB.
ENDIF,NOAERO.
EXIT,U.
REWIND(TAPE30)
RETURN(TAPE25,INIT,INTERP,NONSIM,SIBTCH,TFBTCH,RLBTCH)
RETURN(SMBTCH,ANFIL,NONSIMT)
ATTACH(NSMPPT,MR=1)
LDSET(PRESET=ZERO)
NSMPPT(PL=99999)
EXIT.

*EOR******************************************************************************************

*EOR
.PROC,DBFMOD,INFILE,DFILE=EZSTDBF,LSTFILE.
RETURN,TAPE3,TAPE78,COMPLIB,FLOAD5.
RETURN,LSTFILE,INFILE,PF,DFILE,EZSTDBF.
ATTACH,TAPE3=INFILE.
EXIT,U.
ATTACH,TAPE78=DFILE.
EXIT,U.
SET,R1=0.
IFE,FILE(TAPE78,AS),PURGE.
SET,R1=1.
ENDIF,PURGE.
REQUEST,TAPE79,*PF.
ATTACH,FLOAD5.
ATTACH,COMPLIB.
LIBRARY,COMPLIB.
MAP,OFF.
LDSET,PRESET=ZERO.
FLOAD5.
LIBRARY.
CATALOG,TAPE79,DFILE,RP=999.
IFE,R1=1,NOPURGE.
PURGE,TAPE78.
ENDIF,NOPURGE.
RETURN,DFILE.
CONNECT,OUTPUT.
COPYCR,MESSFILE,OUTPUT.
RETURN, TAPE78, TAPE79, FILOAD5, COMPLIB, TAPE3, MESFILE.
IFE, FILE(TAPE9, AS), Nolist.
REWIND, TAPE9.
COPYCF, TAPE9, LSTFILE.
RETURN, TAPE9.
COPYCR, MESFILE, OUTPUT.
ENDIF, Nolist.
REVERT.
EXIT.
LIBRARY.
CONNECT, OUTPUT.
SKIPF, MESFILE, 2.
COPYCR, MESFILE, OUTPUT.
RETURN, TAPE78, TAPE79, FILOAD5, COMPLIB, TAPE3, MESFILE.
REVERT.
.DATA, MESFILE.
DBFMOD PROCEDURE HAS SUCCESSFULLY EXECUTED.
A NEW CYCLE OF DBFILE HAS BEEN CREATED.
THE PREVIOUS HIGHEST NUMBERED CYCLE OF DBFILE
(IF ONE EXISTED) HAS BEEN PURGED.....
.EOR
COMPONENT INPUT DATA IS AVAILABLE ON LOCAL
FILE LSTFILE.....
.EOR
DBFMOD PROCEDURE HAS ABORTED.....
NO NEW CYCLE OF DBFILE HAS BEEN CREATED...
PREVIOUS CYCLE (IF ANY) STILL EXISTS.
*EOR
************************************************************************************
*EOR
.PROC, COMPILE, N, CODE=Q.
RETURN, PF, EZSTFTN, ONEREL, ONEFTN, FTNLIST, LIBLIST.
REQUEST, FTNLIST, Q.
ATTACH, EZSTFTN.
SKIPF, EZSTFTN, N.
BKSP, EZSTFTN.
COPYCR, EZSTFTN, ONEFTN.
REWIND, ONEFTN.
RETURN, EZSTFTN.
FTN, I=ONEFTN, B=ONEREL, R=2, L=FTNLIST.
ROUTE, FTNLIST, DC=PR, TID=21, ST=CSA, FID=F N CODE.
SKIP, A1.
EXIT, S.
REWIND, MESFILE.
CONNECT, OUTPUT.
COPYBR, MESFILE, OUTPUT.
RETURN, ONEFTN, ONEREL, MESFILE.
REVERT, ABORT.
ENDIF, A1.
RETURN, EZSTLIB.
ATTACH, EZSTLIB.
EDITLIB, I=DIRECT, L=LIBLIST.
EXTEND, EZSTLIB.
REWIND, MESFILE.
SKIPF,MESFILE.
CONNECT,OUTPUT.
COPYBR,MESFILE,OUTPUT.
RETURN,ONEFTN,EZSTLIB,ONEREL,LIBLIST,DIRECT,MESFILE.
REVERT.
EXIT,S.
REWIND,MESFILE.
SKIPF,MESFILE.
CONNECT,OUTPUT.
COPYBR,MESFILE,OUTPUT.
RETURN,DIRECT,ONEREL,MESFILE.
REVERT,ABORT.
.DATA,DIRECT.
LIBRARY(EZSTLIB,OLD)
REWIND(ONEREL)
REPLACE(*,ONEREL)
FINISH.
ENDRUn.
.EOF
.DATA,MESFILE.
FORTRAN ERRORS DURING COMPILATION.PROCEDURE ABORTED
FORTRAN LISTING WITH ERROR DESCRIPTION AVAILABLE ON FILE FTNLIST
.EROR
COMPILE PROCEDURE SUCCESSFULLY EXECUTED
.EROR
LIBRARY MODIFY ERROR....COMPILE PROCEDURE TERMINATED
.EOF
*EOR

********************************************************************************************************************************************************************************************************
*EOR
.PROC,COMPALL,SOURCE,LIBRARY,NOLIST.
RETURN,SOURCE,RELOC,PACK,LIBRARY.
ATTACH,SOURCE.
COMBINE,SOURCE,PACK,999.
RETURN,SOURCE.
REWIND,PACK.
FTN,I=PACK,L=LIST,B=RELOC,R=2,OPT=2,ROUND.
REQUEST,LIBRARY,*PF.
EDITLIB,I=DIRECT,L=Q.
CATALOG,LIBRARY,RP=999.
EXIT,U.
RETURN,RELOC,PACK,LIBRARY.
REVERT.
.DATA,DIRECT.
LIBRARY(LIBRARY,NEW)
REWIND(RELOC)
ADD(*,RELOC)
FINISH.
ENDRUn.
.EOF
*EOR

********************************************************************************************************************************************************************************************************
PROCEDURE EZSTGEN

THIS PROCEDURE WILL GENERATE THE EASIEST PROGRAM FROM THE EASIEST DELIVERY TAPE

INSTRUCTIONS

TO EXECUTE THIS PROCEDURE SUBMIT THE FOLLOWING DECK TO THE ASD COMPUTER INPUT QUEUE AFTER INSTRUCTING THE TAPE LIBRARY TO MOUNT TAPE NUMBER L02377:

EZ5,T300,101000,CM100000,NT1. D79018383,EASIEST TAPE RUN
REQUEST,TAPE,NT,PE,VSN=L02377.
COPYBF,TAPE,TEMP.
BEGIN,EZSTGEN,TEMP,TPW=

SUBMITTING THE ABOVE DECK WILL BOOTSTRAP LOAD AND EXECUTE THE FOLLOWING PROCEDURE

*EOR
.PROC,EZSTGEN,TPW.
REWIND,TAPE.
COPYF,EZSTPRC,2
COPYF,BACOMPS,1
COPYF,COMPASS,1.
REQUEST,TSOUR,*PF.
COPYBR,BACOMPS,TSOUR,999.
COPYBR,COMPASS,TSOUR,999.
CATALOG,TSOUR.
RETURN,TSOUR.
RETURN,BACOMPS,COMPASS.
BEGIN,COMPALL,TEMP,TSOUR,COMPLIB LIST=Q.
ATTACH,TSOUR.
PURGE,TSOUR.
RETURN,TSOUR.
COPYF,EZSTFTN,1.
BEGIN,COMPALL,TEMP,EZSTFTN,EZSTLIB,LIST=Q.
COPYF,FILOAD5,1.
COMPL,FILOADS,FILOAD5,Q.
COPYF,FILOAD.
BEGIN,DBFMOD,TEMP,FILOAD.
COPYF,EASYS,Q.
COPYF,EASY,1
COMPL,EASYS,EASY5,1.
COPYF,NONSIM5,Q.
COPYF,NONSIM5,1.
COMPL,NONSIM5,NONSIM5,1.
COPYF,NSMPPPTS,1.
COMPL,NSMPPTS,NSMPPTS,Q.
COPYF,AEROMED,1.
COMPL,AEROMED,AEROMDB,Q.
COPYF,F4EMAN,2.
COPYF,MCORR,2.
COPYF,ACORR,2.
COPYTF, MODAPP, 2.
COPYTF, ANALAPP, 2.
BEGIN, SUBRUN, TEMP, MCORR, ACORR, TIME=1500, INOUT=1500, CORE=230000, AEROMED.
.DA, COPYTF.
.PROC, COPYTF, FILE, CODE.
SET, RL=CODE.
RETURN, FILE.
REQUEST, FILE, *PF.
COPYBF, TAPE, FILE.
REWIND, FILE.
IFE, RL, NE, 2, NOPASS.
IFE, RL, NE, 1, NOPASS.
CATALOG, FILE, RP=999, TK=TPW.
REVERT.
ENDIF, NOPASS.
CATALOG, FILE, RP=999.
REWIND, FILE.
IFE, RL, NE, 1, NODROP.
RETURN, FILE.
ENDIF, NODROP.
REVERT.
.EOF
.DATA, COMPL.
.PROC, COMPL, SOURCE, FILE, CODE.
SET, RL=CODE.
REWIND, SOURCE, PACK.
COMBINE, SOURCE, PACK, 999.
RETURN, SOURCE, RELOC, PF.
REWIND, PACK.
REQUEST, PF, *PF.
IFE, RL, NE, 0, NOCOPY.
FTN, I=PACK, L=Q, ROUND, OPT=2, B=RELOC.
COPYLM, FILE, RELOC, PF.
RETURN, RELOC, FILE.
ELSE, NOCOPY.
FTN, I=PACK, L=Q, OPT=2, ROUND, B=PF.
IFE, RL, NE, 2, NOPASS.
ENDIF, NOCOPY.
CATALOG, PF, FILE, RP=999.
RETURN, PF, PACK.
REVERT.
ENDIF, NOPASS.
CATALOG, PF, FILE, RP=999.
RETURN, PF, PACK.
REVERT.
This appendix contains listings of the EASIEST standard components which include the following:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>Attached body (Survival Kit)</td>
</tr>
<tr>
<td>AE</td>
<td>Airplane</td>
</tr>
<tr>
<td>AG</td>
<td>Atmospheric properties</td>
</tr>
<tr>
<td>AM</td>
<td>Aeromedical</td>
</tr>
<tr>
<td>AP</td>
<td>Aerodynamic plate</td>
</tr>
<tr>
<td>AS</td>
<td>Seat aerodynamics</td>
</tr>
<tr>
<td>CE</td>
<td>Crewperson</td>
</tr>
<tr>
<td>CS</td>
<td>Airplane control surfaces</td>
</tr>
<tr>
<td>CT</td>
<td>Catapult</td>
</tr>
<tr>
<td>DR</td>
<td>DART</td>
</tr>
<tr>
<td>GP</td>
<td>Simple parachute mortar and restraints</td>
</tr>
<tr>
<td>LI</td>
<td>Parachute lines</td>
</tr>
<tr>
<td>MP</td>
<td>Parachute mortar</td>
</tr>
<tr>
<td>PC</td>
<td>Parachute</td>
</tr>
<tr>
<td>RL</td>
<td>Rails</td>
</tr>
<tr>
<td>RS</td>
<td>Restraints</td>
</tr>
<tr>
<td>SE</td>
<td>Ejection seat</td>
</tr>
<tr>
<td>SL</td>
<td>Sled</td>
</tr>
<tr>
<td>SP</td>
<td>STAPAC</td>
</tr>
<tr>
<td>SR</td>
<td>Sustainer rocket</td>
</tr>
<tr>
<td>WB</td>
<td>Weight and balance</td>
</tr>
</tbody>
</table>
SUBROUTINE AB (UAB,UAAB,IAAB,XAB,XABD,IXAB,WAB,WABD,IAWAB,
   EAAB,EABD,IEAB,  
   WT,UMI,DP1,FAU,TAB,FAU,TAU,TRM)

EASIEST ATTACHED BODY COMPONENT

DESIGNED BY C.L. WEST
LAST MODIFIED - DECEMBER 9, 1980

*************** AB OUTPUTS ***************

LINEAR VELOCITIES - BODY AXIS

UAB(3)  -  X,Y,Z LINEAR VELOCITY VECTOR OF THE ATTACHED BODY (FT/SEC)
UABD(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR OF THE ATTACHED BODY (FT/SEC/SEC)
IAAB(3) - INTEGRATION CONTROL

LINEAR POSITIONS - EARTH SYSTEM

XAB(3)  -  X,Y,Z LINEAR POSITION VECTOR OF THE ATTACHED BODY (FT)
XABD(3) - X,Y,Z LINEAR POSITION RATE VECTOR OF THE ATTACHED BODY (FT/SEC)
IXAB(3) - INTEGRATION CONTROL

ANGULAR VELOCITIES - BODY AXIS

WAB(3)  -  X,Y,Z ANGULAR VELOCITY COMPONENTS - P,Q,R (DEG/SEC)
WABD(3) - X,Y,Z ANGULAR VELOCITY RATE COMPONENTS (DEG/SEC/SEC)
IAWB(3) - INTEGRATION CONTROL

EULER ANGLES -- EARTH TO ATTACHED BODY -- YAW,PITCH,ROLL

LAMB(3) - EARTH TO ATTACHED BODY EULER ANGLES (DEG)
LAMB(3) - EULER ANGLE RATES (DEG/SEC)
LIEAB(3) - INTEGRATION CONTROL

*************** AB INPUTS ***************

WT  -  WEIGHT OF THE ATTACHED BODY (LB)
UMI(3)  -  ATTACHED BODY MOMENTS OF INERTIA - IXX,IYY,IZZ  
            (SLUG-FT**2)
DP1(3)  -  ATTACHED BODY PRODUCTS OF INERTIA - IXY,IXZ,IYZ  
            (SLUG-FT**2)
FAAB(3) - X,Y,Z BODY AXIS FORCE COMPONENTS FROM THE RESTRAINTS (LB)
TAAB(3) - X,Y,Z BODY AXIS TORQUE COMPONENTS FROM THE RESTRAINTS (LB)
FAU(3)  -  AUXILIARY X,Y,Z BODY AXIS FORCE COMPONENTS (LB)
TAU(3)  -  AUXILIARY X,Y,Z BODY AXIS TORQUE COMPONENTS (FT-LB)
TRM(3)  -  X,Y,Z PARENT BODY EARTH VELOCITY COMPONENTS FOR  
            CALCULATING THE LINEAR POSITION RATES DURING TRIM (FT/SEC)

DIMENSIONS OF CALLING ARGUMENTS .......

DIMENSION UAB(3),UAAB(3),IAAB(3),XAB(3),XABD(3),IXAB(3),  
   WAB(3),WABD(3),IWB(3),EAB(3),EABD(3),IEAB(3),  
   UMI(3),DP1(3),FAU(3),TAB(3),FAU(3),TAU(3),TRM(3)
INTERNAL DIMENSIONS ......

DIMENSION TINEP(3,3), TEMP(3), TEMP2(3), TEMP3(3), WABIR(3),
  EABIR(3), DE(3,3), DAE(3,3), F(3), T(3)

COMMON / CICAL / ICAL
COMMON / COVRKLY/ INST
COMMON / CSSFLG/ SSFLG
COMMON / CI0 / IREAD, IWRITE, IUIAG
DATA RPD, DPE / .01745329, 57.29578 /
DATA GRAV / 32.174 /

*******************************
**** INITIALIZATION ****
*******************************

IF (ICAL .NE. 1) GO TO 20

20 DO 10 I = 1, 3
  IF (FA(1) .EQ. 0.99999) FAB(I) = 0
  IF (FAU(I) .EQ. 0.99999) FAU(I) = 0
  IF (TAB(I) .EQ. 0.99999) TAB(I) = 0
  IF (TAU(I) .EQ. 0.99999) TAU(I) = 0
  TRM(I) = TRM(2) = TRM(3) = 0

==================================

SET UP THE ATTACHED BODY INERTIA TENSOR ....

20 TINER(1,1) = BMI(1)
  TINER(1,2) = -BPI(1)
  TINER(1,3) = -BPI(2)
  TINER(2,1) = -BPI(1)
  TINER(2,2) = BMI(2)
  TINER(2,3) = -BPI(3)
  TINER(3,1) = -BPI(1)
  TINER(3,2) = -BPI(2)
  TINER(3,3) = BMI(3)

CHANGE FROM DEGREES TO RADIANS ....

30 DO 30 I = 1, 3
  WABIR(I) = WAB(I) * RPD
  30 EABIR(I) = EAB(I) * KPD

CALL JINCUS (DE, EABIR)
CALL TRANS (DEA, DEA, 3, 3)

CALL THE TOTAL FORCE AND TORQUE DUE TO THE EXTERNAL
FORCES AND GRAVITY ....

40 DO 40 I = 1, 3
  F(I) = FAB(I) + FAU(I) + WT * DEA(I, 3) * SSFLG
  40 T(I) = TAB(I) + TAU(I)
CALL MATPY (TEMP1, TINER, WABIR, 3, 3, 1)

CALL WABIR X (TINER * WABIR) ......

CALL CRSRDU (TEMP2, WABIR, TEMP1)

SUM TERMS TO OBTAIN TOTAL TORQUE ......

DO 50 I = 1, 3
   TEMP3(I) = T(I) - TEMP2(I)

DO 50 I = 1, 3
   TCMP3(I) = TEMP3(I)

CALL WABDIR ......

CALCULATE LUEQS (TINER, TEMP1, TEMP3, TEMP2, 3, 1, 3, 3, 3, 1.E-14, IERROR)

IF (IERROR.NE.1) GO TO 70

MK13(I0, 0)

FORMAT(10, INERTIA MATRIX OF THE ATTACHED BODY IS SINGULAR ........
   \* RUN STOPPED\*

STOP

70 DO I = 1, 3
   IF (IMAB(I).NE.0) WABU(I) = TEMP1(I) * DPR

DO 70 I = 1, 3
   IF (IAB(I).NE.0) EABU(I) = TEMP1(I) * DPR

DO 60 I = 1, 3
   WABU(I) = TEMP2(I) - TEMP1(I)

DO 100 I = 1, 3
   UABD(I) = TEMP2(I) - TEMP1(I)

DO 110 I = 1, 3

CRSOLVE (TEMP1, WABIR, EABIR)

CALL ESRATE (TEMP1, WABIR, EABIR)

DO 90 I = 1, 3
   IF (IAB(I).NE.0) EABU(I) = TEMP1(I) * DPR

FUNCTION ABMRT = WT/GRAV

DO 100 I = 1, 3
   TEMP2(I) = F(I)/ABMRT

CALL UABD ......

DO 110 I = 1, 3
   IF (UABD(I).NE.0) UABD(I) = TEMP2(I) - TEMP1(I)

CALL UABD ......

DO 110 I = 1, 3
   IF (UABD(I).NE.0) UABD(I) = TEMP2(I) - TEMP1(I)

CALL UABD ......

DO 110 I = 1, 3
   IF (UABD(I).NE.0) UABD(I) = TEMP2(I) - TEMP1(I)
CALL MATMPY (TEMP1,DATA,UAD,3,3,1)
DO 120 I=1,3
  120 IF(I*XAD(I).NE.0) XAD(I) = TEMP1(I)
C
C SUBTRACT TRIM VELOCITY FROM POSITION RATES DURING TRIM .......
C
  IF(INST.NE.31) GO TO 140
  GO 130 I=1,3
  130 IF(I*XAD(I).NE.0) XAD(I) = XAD(I) - TRM(I)
C
140 RETURN
END
SUBROUTINE AE (UAP,UAPU,IUAP,XAP,XAPU,IXAP,WAP,WAPU,IWAP,
  .
  EAP,EAPU,ICAP,TRM,TRMU,ITKM,ALPHA,BETA,VMACH,ALT,
  .
  AN,AS,SYAPU,ZCP,AMY,API,
  .
  ATTM,AXIL,AELE,ARUO,XEN,END,TALT,TVEL,
  .
  FRA1,TRA1,FCA1,TCAL,FUA1,TUA1,
  .
  FRA2,TRA2,FCA2,TCA2,FUA2,TUA2,CPF)

***** THE EASIEST AIRPLANE COMPONENT *****

THIS ROUTINE IS A SIX DEGREE OF FREEDOM MODEL OF AN AIRPLANE

THE AERODYNAMIC COEFFICIENTS ARE READ FROM TAPE3

THE AIRPLANE TRIM IS PROVIDED INTERNALLY USING EASY STEADY STATE

ANALYSIS

CONTROL SURFACE AND THRUST COMMANDS INPUT BY THE USER AFTER

TRIM WILL BE INTERPRETED AS BEING AN ADDITION TO THE SETTINGS

REQUIRED FOR TRIM

DESIGNED BY B. UMMEL AND C.L. WEST

LAST MODIFIED - DECEMBER 6, 1980

************* AE OUTPUTS *************

LINEAR VELOCITIES - BODY AXIS

UAP(3) - X,Y,Z LINEAR VELOCITY VECTOR OF THE AIRPLANE CENTER

OF GRAVITY (FT/SEC)

UAPU(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR OF THE AIRPLANE

CENTER OF GRAVITY (FT/SEC/SEC)

IUAP(3) - INTEGRATION CONTROL

LINEAR POSITIONS - EARTH SYSTEM

XAP(3) - X,Y,Z LINEAR POSITION VECTOR OF THE AIRPLANE CENTER

OF GRAVITY (FT)

XAPU(3) - X,Y,Z LINEAR POSITION RATE VECTOR OF THE AIRPLANE

CENTER OF GRAVITY (FT/SEC)

IXAP(3) - INTEGRATION CONTROL

ANGULAR VELOCITIES - BODY AXIS

WAP(3) - X,Y,Z ANGULAR VELOCITY COMPONENTS - P,J,R (DEG/SEC)

WAPU(3) - X,Y,Z ANGULAR VELOCITY RATE COMPONENTS (DEG/SEC/SEC)

IWAP(3) - INTEGRATION CONTROL

EULER ANGLES -- EARTH TO BODY AXIS -- YAW,PITCH,ROLL

EAP(3) - EARTH TO AIRPLANE EULER ANGLES (DEG)

EAPU(3) - EULER ANGLE RATES (DEG/SEC)

IEAP(3) - INTEGRATION CONTROL

TRIM CONTROL STATES -- TRM(4),TRMD(4),ITRM(4)

TRM(1) = TRIM THROTTLE SETTING

TRM(2) = TRIM AILERON SETTING

284
TRM(3) = TRIM ELEVATOR SETTING
TRM(4) = TRIM RUDGER SETTING

ALPHA = ANGLE OF ATTACK (DEG)
OCTA = SIJESLIP ANGLE (DEG)
VMACH = MACH NUMBER
ALT = ALTITUDE ABOVE SEA LEVEL (FT)

*********** AE INPUTS ***********

AW - AIRPLANE WEIGHT (LB)
B - WINGS Span (Feet)
C - MEAN AERODYNAMIC CHORD (FEET)
S - REFERENCE AREA (FT**2)
XLP - AIRPLANE BODY X-AXIS POSITION OF THE CENTER OF PRESSURE (FT)
ZCP - AIRPLANE BODY Z-AXIS POSITION OF THE CENTER OF PRESSURE (FT)
AM(3) - MOMENTS OF INERTIA -- Ixx, Iyy, Izz (SLUu-FT**2)
API(3) - PRODUCTS OF INERTIA -- Ixy, Ixz, Iyz (SLUu-FT**2)
XIth - EXTERNAL THRUST SETTING (LB)
XAIL - EXTERNAL AILERON SETTING (DEG)
XAELE - EXTERNAL ELEVATOR SETTING (DEG)
XKJD - EXTERNAL RUDDER SETTING (DEG)
AEN(3) - x, y, z AIRPLANE BODY AXIS POSITION VECTOR OF THE ENGINE (FT)
ENU(3) - AIRPLANE BODY AXIS DIRECTION COSINES OF THE ENGINE THRUST VECTOR
TALT - DESIRED TRIM AIRPLANE ALTITUDE (FT)
IVEL - DESIRED TRIM AIRPLANE SPEED (FT/SEC)
FKAI(3) - PORT ONE x, y, z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS (LB)
TKAI(3) - PORT ONE x, y, z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS (FT-LB)
FCAI(3) - PORT ONE x, y, z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT (LB)
TLAI(3) - PORT ONE x, y, z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT (FT-LB)
FUA1(3) - PORT ONE x, y, z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART (LB)
TUAI(3) - PORT ONE x, y, z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART (FT-LB)
FKAZ(3) - PORT TWO x, y, z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS (LB)
TRAZ(3) - PORT TWO x, y, z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS (FT-LB)
FCAZ(3) - PORT TWO x, y, z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT (LB)
TCAZ(3) - PORT TWO x, y, z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT (FT-LB)
FUAZ(3) - PORT TWO x, y, z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART (LB)
TUAZ(3) - PORT TWO x, y, z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART (FT-LB)
CPF - PRINT FLAG FOR AERODYNAMIC COEFFICIENTS
*************** DATA DECLARATIONS ***************

CALLING SEQUENCE DIMENSIONS

DIMENSION XAP(3), XAPU(3), IAAP(3), UAP(3), UAPD(3), IUAP(3),
  MAP(3), MAPU(3), IAEP(3), EAP(3), EAPD(3), IEAP(3),
  IRM(4), IRMU(4), 1IRM(4),
  AMl(3), API(3), XEN(3), END(3),
  FRA(3), TRA(3), FCA(3), TCA(3), FDA(3), TDA(3),
  FRA3), TRa3), FCa3), TCA3), FDA3), TDA3),

INTERNAL DIMENSIONS

DIMENSION UWI(3), UCI(3), UWE(3), DEA(3,3), DAE(3,3), TINER(3,3),
  TEMPI(3,3), F(3), FGRAV(3), FENG(3), FCRK(3), FCRD(3),
  FAE50(3), I(3), ICENG(3), FCRD(3), TAE50(3), WAPIR(3),
  TEMPI(3), TEMP2(3), TEMP3(3), EAPIR(3), WAPIR(3),
  R(2000), ITPLT(60), FORMAT(8), TITLE(6)

COMMON/REGIONS/NX(60)
COMMON/CLCCAL/LLCAL
COMMON/COVRLY/INST
COMMON/CIU/IRAUU, IWRITE, IDIA
DATA GRAV /32.1747/

DATA RPO, DP, IFLAG / 0.1745329, 57.29578, 0 /

*************** INITIALIZATION ***************

*************** INITIALIZATION ***************

IF(ILLCAL .NE. 1) GO TO 110
XAP(3) = -TA1
UAPU(1) = TVEL
IF(XXP .EQ. 0.99999) XCP = 0
IF(XCP .EQ. 0.99999) ZCP = 0
IF(XXHR .EQ. 0.99999) XTHR = 0
IF(XAIL .EQ. 0.99999) XAIL = 0
IF(XELE .EQ. 0.99999) ACLE = 0
IF(AKU .EQ. 0.99999) AKU = 0
IF(CPF .EQ. 0.99999) CPF = 0

110 IF(I = 1,3)
  IF(FRA(1) .EQ. 0.99999) FRA(1) = 0
  IF(TRA(1) .EQ. 0.99999) TRA(1) = 0
  IF(FCA(1) .EQ. 0.99999) FCA(1) = 0
  IF(TCA(1) .EQ. 0.99999) TCA(1) = 0
  IF(FDA(1) .EQ. 0.99999) FDA(1) = 0
  IF(TDA(1) .EQ. 0.99999) TDA(1) = 0
  IF(FRA2(1) .EQ. 0.99999) FRA2(1) = 0
  IF(TRA2(1) .EQ. 0.99999) TRA2(1) = 0
  IF(FCA2(1) .EQ. 0.99999) FCA2(1) = 0
  IF(TCA2(1) .EQ. 0.99999) TCA2(1) = 0
  IF(FDA2(1) .EQ. 0.99999) FDA2(1) = 0
  IF(TDA2(1) .EQ. 0.99999) TDA2(1) = 0
CONTINUE

286
C \---------- SET UP AIRPLANE INERTIA TENSOR \----------

TINER(1,1) =AMI(1)
TINER(1,2) =-API(1)
TINER(1,3) =-API(2)
TINER(2,1) =-API(1)
TINER(2,2) =AMI(2)
TINER(2,3) =-API(3)
TINER(3,1) =-API(2)
TINER(3,2) =-API(3)
TINER(3,3) =AMI(3)

C \---------- READ AERODYNAMIC TABLES FROM TAPE3 \----------

IF(IFLAG.EQ.1) GO TO 11U
IFLAG=1
RERING 3

C IF(LCPF.EQ.1,O) WRITE(0,10)
10 FORMAT(25X,*AERODYNAMIC COEFFICIENTS FOR THE AIRPLANE*)
READ(3,20) NTEMP
20 FORMAT(18I4)
READ(5,X) (ITPLOT(I),I=1,NTEMP)
NTEMP=1 NTEMP-1
NK(1)=5
GO TO 30 I=1,NTEMP+1
30 NK(I+1)=NK(I)+ITPLOT(I)+1
IF(LCPF.EQ.1.O) WRITE(5,40) (I,NR(1),I=1,NTEMP)
40 FORMAT(9(13,15))
IF(LCPF.EQ.1.O) WRITE(6,5O)
50 FORMAT(1HO)
60 READ(5,70) (ITPLOT(I),I=1,4),(TITLE(I),I=1,6)
70 FORMAT(414,9A10)
NTEMP =ITPLOT(1)
IF(NTEMP.LE.0) GO TO 110
NTEMP=NK(NTEMP)+ITPLOT(2)
NTEMP1=ITPLOT(3)
NTEMP1=NK(NTEMP1)+ITPLOT(4)
READ(3,80) FORMAT
80 FORMAT(B10)
READ(10,FORMAT) (R(I),I=1,NTEMP,NTEMP1)
IF(LCPF.EQ.1.O) WRITE(6,90) (ITPLOT(I),I=1,4),(TITLE(I),I=1,6)
90 FORMAT(410,9A10)
IF(LCPF.EQ.1.O) WRITE(6,100) (I,R(I),I=1,NTEMP,NTEMP1)
100 FORMAT(4(16,F14.6))
GO TO 60

C \----------------------------------------

C \---------- CONVERT ANGULAR RATES AND EULER ANGLES TO RADIANS \----------

C GO 120 I=1,3
EAPIR(I) = EAPI(I)*KPU
WAPIR(I) = WAPI(I)*KPU

C \---------- COMPUTE EARTH TO AIRPLANE AND AIRPLANE TO EARTH \----------

287
CALL DIRCOS (UCA, EAPIK)
CALL TRANS (UAE, UCA, 3, 3)

--- CONTROL SURFACE SETTINGS ---
DA = 2.0 * (RM(2) + XAIL)
AUA = ABS(DA)
DE = (IWM(3) + XELE)
DR = (TRM(4) + XRUD)

--- OBTAIN SPEED OF SOUND, AIR DENSITY, AND WIND VELOCITY ---
ALI = -XAP(3)
CALL ATMOS (A2, RH0, ALT, UW, 0, 0, 0)

--- PUT WIND INTO BODY COORDINATES ---
CALL MATMPY (UWB, UEA, UW, 3, 3, 1)

--- ADD WIND VELOCITY TO AIRPLANE VELOCITY ---
UG(1) = UAP(1) - UWB(1)
UG(2) = UAP(2) - UWB(2)
UG(3) = UAP(3) - UWB(3)

--- AERO VARIABLES ---
IF(UO(1), EQ, 0.0, AND, UO(3), EQ, 0.0) UO(1) = 0.01
ALPHA = ARTAN2(UO(3), UO(1)) * DPR
CUSA = COS(ALPHA * KPU)
SINA = SIN(ALPHA * KPU)
CALL OTIPRO (VBAR2, UO, UO, 3)
VBAR = SQRT(VBAR2)
DETA = ASIN(UO(2) / VBAR) * UPR
VMAL = VBAR / AL
QAS = 0.5 * KHO * VBAR2 * S

--- COMPUTE STABILITY AXIS ANGULAR RATES ---
WAPIK(1) = WAPIK(1) * CUSA + WAPI(3) * SINA
WAPIK(2) = WAPIK(2)
WAPIK(3) = -WAPIK(1) * SINA + WAPI(3) * CUSA

***********************************************************************
***********************************************************************
***********************************************************************

***********************************************************************
***********************************************************************
***********************************************************************

--- TRANSFER AERO VARIABLES TO THE R ARRAY ---
K(1) = VMACH
K(2) = ALPHA
K(3) = BETA

--- 2 AXIS FORCE COEFFICIENTS
BIAS COEFFICIENT FOR TRIM .......

288
CALL LOOK (NR(1), R, CZ0)
C VARIATION OF CZ0 WITH ALPHA DOT .......
CALL LOOK (NR(2), R, CZAD)
C VARIATION OF CZ0 WITH PITCH RATE .......
CALL LOOK (NR(3), R, C2W)
C VARIATION OF CZ0 WITH ELEVATOR POSITION .......
CALL LOOK (NR(4), R, CZDE)
C VARIATION OF CZ0 WITH AILERON POSITION .......
CALL LOOK (NR(5), R, CZDA)

C _________ X-AXIS FORCE COEFFICIENTS
C
C BIAS COEFFICIENT FOR TRIM .......
R[4]=CZ0
CALL LOOK (NR(6), R, CX0)
C VARIATION OF CX0 WITH AILERON POSITION .......
CALL LOOK (NR(7), R, CXDA)
C
C _________ PITCHING MOMENT COEFFICIENTS
C
C BIAS COEFFICIENT FOR TRIM .......
CALL LOOK (NR(8), R, CMG)
C VARIATION OF CMG WITH ALPHA DOT .......
CALL LOOK (NR(9), R, CMGD)
C VARIATION OF CMG WITH PITCH RATE .......
CALL LOOK (NR(10), R, CMG)
C VARIATION OF CMG WITH ELEVATOR POSITION .......
CALL LOOK (NR(11), R, CMGE)
C VARIATION OF CMG WITH AILERON POSITION .......
CALL LOOK (NR(12), R, CMDA)
C
C _________ SIDE FORCE COEFFICIENTS
C
C VARIATION OF CY WITH BETA .......
CALL LOOK (NR(13), R, CYB)
C VARIATION OF CY WITH ROLL RATE .......
CALL LOOK (NR(14), R, CYP)
C VARIATION OF CY WITH YAW RATE .......
CALL LOOK (NR(15), R, CYR)
C VARIATION OF CY WITH RUDDER POSITION .......
CALL LOOK (NR(16), R, CYDR)
C VARIATION OF CY WITH AILERON DEFORMATION .......
CALL LOOK (NR(17), R, CYDA)
C
C _________ ROLLING MOMENT COEFFICIENTS
C
C VARIATION OF CL WITH BETA .......
CALL LOOK (NR(18), R, CLB)
C VARIATION OF CL WITH ROLL RATE .......
CALL LOOK (NR(19), R, CLP)
C VARIATION OF CL WITH YAW RATE .......
CALL LOOK (NR(20), R, CLK)
C VARIATION OF CL WITH RUDDER DEFORMATION .......
CALL LOOK (NR(21), R, CLUK)
C VARIATION OF CL WITH AILERON DEFORMATION .......
CALL LOOK (NR(22), R, CLDA)
C
C _________ YAWING MOMENT COEFFICIENTS

289
C VARIATION OF CN WITH BETA ...
   CALL LOOK (NR(23),K,CNA)
L VARIATION OF CN WITH ROLL RATE ...
   CALL LOOK (NR(4),K,CNP)
C VARIATION OF CN WITH YAW RATE ...
   CALL LOOK (NR(25),K,CNR)
C VARIATION OF CN WITH ROLLER DEFORMATION ...
   CALL LOOK (NR(20),K,CNRK)
L VARIATION OF CN WITH ALLERON DEFORMATION ...
   CALL LOOK (NR(27),K,CNUA)
C PRINT AERO COEFFICIENTS DURING PRINT TASK ONLY
C
IF(INST.>60)WRITE(6,145)VMACH,ALPHA,BETA,CZO,CZAD,
   * CZG,CZUE,CZUA,CXO,CXUA,CMO,CMAD,CMQ,CMDE,CMDA,CYB,CYP,
   * CYR,CYDR,CYDA,CLP,CLR,CLUR,CLUA,CNB,CNP,CNR,CNDR,CNDA
125 FORMAT(/* AIRPLANE AERO COEFFICIENTS FOR MACH=*,G12.5, /
   * ALPHA=*,G12.5,* BETA=*,G12.5,* CZO =*,G12.5,* CZAD=*,G12.5, /
   * CZG =*,G12.5,* CZUE=*,G12.5,* CZUA=*,G12.5,* CXO =*,G12.5/
   * CXOA=*,G12.5,* CMQ =*,G12.5,* CMDE=*,G12.5,* CMAD=*,G12.5, /
   * CMDA=*,G12.5,* CMQ =*,G12.5,* CMDE=*,G12.5,* CMAD=*,G12.5, /
   * CMDA=*,G12.5,* CMQ =*,G12.5,* CMDE=*,G12.5,* CMAD=*,G12.5, /
   * CMDA=*,G12.5)
C
C COMPUTE THE FORCE DUE TO GRAVITY -------
C
AMASS = AW/GRAV
FGRAV(1) = AM * DEAI(1,3)
FGRAV(2) = AM * DEAI(2,3)
FGRAV(3) = AM * DEAI(3,3)
C
C CALCULATE THE FORCE DUE TO THE CORIOLIS ACCELERATION -------
C
   CALL CRSPRD (FCOR,WAPIR,UAP)
   DU 130 I=1,3
130 FCOR(I) = -FCOR(I) * AMASS
C
C CALCULATE THE ENGINE FORCES -------
C
   ETHRUST = TKR(1)+XTHR
   DU 140 I=1,3
140 FENG(I) = ETHRUST * END(I)
C
C COMPUTE THE FORCES FROM THE RAILS, CATAPULTS, AND THE DARTS
C
   DU 150 I=1,3
150 FRCO(I) = FRA1(I) + FRA2(I) + FCA1(I) + FCA2(I) + 
   * FUA1(I) + FOA2(I)
C
C CALCULATE THE BODY AXIS AERODYNAMIC FORCES -------
C
290
(EXCEPT THOSE USING ALPHA DOT)

C
BWZV = B/(VBAR+V0AR)
C0ZV = C/(VBAR+V0AR)
C
FX = WAS*(CXO+CXDA*AUA)
FY = WAS*(CYB+ETA+CYP+APIRS(1)+CYR+APIRS(3))*B0ZV*CYD*DR
   + CYDA*UA
FZ = W,AS*(CZ0+CZDA*DE+CZDA*ADA+C0ZV*CZQ*APIRS(2))
C
CHANGE FROM STABILITY AXIS TO BODY AXIS ...... 
C
FAERO(1) = FZ*SINA - FX*COSA
FAERO(2) = FY
FAERO(3) = -FZ*COSA - FX*SINA
C
--- TOTAL FORCES ACTING ON AIRPLANE EXCEPT FOR ALPHA DOT EFFECTS
C
JO 160 I=1..3
160 F(I) = FORAV(I) + FCOK(I) + FENG(I) + FRCD(I) + FAERO(I)
C
--- SOLVE FOR LINEAR ACCELERATIONS USING FORCES INCLUDING ALPHA DOT EFFECT
C
VAR = (CO2V * CZAD * WAS ) / (UAP(1)**2 + UAP(3)**2)
DEN = AMASS - VAR * (UAP(1)*COSA - UAP(3)*SINA)
C
TEMP1(I) = (F(I) - F(I)*COSA + F(3)*SINA)*VAR*UAP(1)/AMASS/DEN
TEMP2(I) = F(2) / AMASS
TEMP3(I) = (F(3) - F(I)*COSA + F(3)*SINA)*VAR*UAP(3)/AMASS/DEN
C
JO 170 I=1..3
170 IF(UAP(I)*NE.0) UAPU(I) = TEMP(I)
C
******************************************************************************
C ***** LINEAR POSITION EQUATIONS *****
******************************************************************************
C
CALL MATPY (TEMP1,DAE,UAP,3,3,1)
DO 180 I=1..3
180 IF(IXAP(I)*NE.0) XAPU(I) = TEMP1(I)
C
******************************************************************************
C ***** ANGULAR VELOCITY EQUATIONS *****
******************************************************************************
C
----- CALCULATE THE ENGINE TORQUE ----- 
C
CALL CRSPRO (TENG,XEN,FENG)
C
----- CALCULATE THE TORQUE DUE TO THE ----- 
C RAILS, CATAPULTS, AND DARTS
C
DO 190 I=1..3
190 TRCD(I) = TRA1(I) + TRA2(I) + TCA1(I) + TCA2(I) +
   TDA1(I) + TDA2(I)
C
----- CALCULATE THE AERODYNAMIC TORQUE ----- 
C
291
ALDI1 = (UAP(1) - UAP(3) * UAPD(1) / (UAP(3) ** 2 * UAP(1) ** 2)

TX = UAPD(1) * (CLD * ETA + UAPR1S(1) + CLR * UAPR3(3)) * 0.02V + CLDR * OR

TY = UAPD(1) * (CMG + CO2V * CMAX + ALU * LMG * WAPRS(2) + CMDE + CDN * CDN * ADA)

TZ = UAPD(1) * (CLN * ETA + LNP * WAPR1S(1) + LNR * WAPRS(3) * 0.02V + CDN * OR

CHANGE FROM STABILITY AXES TO BODY AXES.....

TAERO(1) = TX * COSA - TZ * SINA - ZCP * FAERO(2)

TAERO(2) = TY + XCP * FAER0(3) + LCP * FAERO(1)

TAERO(3) = TX * SINA + LZ * COSA + XCP * FAERO(2)

----- CALCULATE THE TOTAL TORQUE ACTING ON THE AIRPLANE -----  

DU 200 I=1,3

----- PRINT AIRPLANE FORCES AND TUKES DURING PRINT TASK ONLY ----

IF(NSET, EQ, 0) WRITE(0, 210) (FGRAV(1)(1), I=1,3), (FCOR(1), I=1,3)

. IF(NSET, I, I=1,3) , (IFUR(1), I=1,3), (TAERO(1), I=1,3), FX, FY, FZ, TX, TY, TZ, ALU	

210 FORMAT(* AIRPLANE FORCES AND TUKES */ FRC.GRAV. **3G12.5,

* FRC.CURULS=.3G12.5/* FRC.ENINE =**3G12.5,* TRQ.ENINE =**3G12.5/*

. 3G12.5/* FRC.EJSET =**3G12.5/* TRQ.EJSET =**3G12.5/

. * FRL.AERO =**3G12.5/* TRQ.AEHU =**3G12.5/

. * FAER0.ST.AX =**3G12.5,* TAERO.ST.AX =**3G12.5/

. * AIRPLANE ALPHA UOT =**3G12.5/*)

CALCULATE TINER * WAPIK .......

CALL MATMPY (TEMP1, TINER, WAPIK, I=3,3,1)

CALCULATE WAPIK X (TINER * WAPIK)

CALL CRSRED (TEMP2, WAPIK, TEMPI)

SUM TERMS TO OBTAIN TOTAL TORQUE .......

DU 230 I=1,3

230 TEMP3(I) = TEMP(I) - TEMP2(I)

SET UP TEMPORARY INERTIA TENSOR .......

DJ 336 I=1,3

DJ 256 J=1,3

330 TEMP(1, J) = TINER(I, J)

CALCULATE TAPM .......

CALL LUCUS (TEMP1, TEMPI, TEMP2, TEMP3, TEMP4, 1, 3, 5, 3, 1, 3, 5, 3, 1.0, 14, IERROR)

IF (ERROK, NE, 1) DU TO 250

. KIIEF(0, 246)

240 FORMAT(* INERTIA MAIRIX OF AIRPLANE IS SINGULAR...RUN STOPPED*)

STOP

250 CONTINUE
C     DU 260 I=1,3
260 IF(IWAP(1).NE.0) WAPU(1) = TEMPI(1)*GPR
C
C ***********************************************************************
C ***** EULER ANGLE EQUATIONS *****
C ***********************************************************************
C
CALL EARATE (TEMPI,WAPIR,EAPIR)
C     DU 270 I=1,3
C
270 IF(IAP(1).NE.0) EAPD(1) = TEMPI(1)*GPR
C
C ***********************************************************************
C ********** TRIM LOGIC **********
C ***********************************************************************
C
TKMD(1)=TRMD(2)=TRMD(3)=TRMD(4)=0
IF(INST.NE.31) GO TO 280
IF(ITRM(1).NE.0) TRMD(1) = TVEL - VBAR
IF(ITRM(2).NE.0) TRMD(2) = +.01*WAPIR(1)+ EAPIR(3)
IF(ITRM(3).NE.0) TRMD(3) = +.01*WAPIR(2)-.001*XAPD(3)
      -.0001*(TALT+XAP(3))
      IF(ITRM(4).NE.0) TRMD(4) = +.01*WAPIR(3)
C
280 RETURN
END
SUBROUTINE AG (VS,RHOS)

     * M,WIN,DP,TE,SW)
     * DIMENSION WIN(3),WIND(3)
     * COMMON /CCAL/ ICCAL
     * COMMON /CSSFLG/ SSFLG
     * COMMON /COVRLY/ INST
     * COMMON /CIO/ IRCAL,WRITE,IODAu
     * DATA FL1,FL2 /0,0/

DESIGNED BY C.R. WEST
LAST MODIFIED - DECEMBER 6, 1980

THE STANDARD COMPONENT WHICH DETERMINES THE AIR DENSITY, SPEED OF
SOUND, AND THE WIND VELOCITY AT A PRESCRIBED ALTITUDE IN A STANDARD
OR NON-STANDARD ATMOSPHERE. IN ADDITION, IT SETS A FLAG WHICH FORCES
THE ACCELERATION OF GRAVITY TO BE ZERO FOR THE STEADY STATE CALCULATION
OF AN UNSUPPORTED SEAT. THIS FLAG CAN ALSO BE USED TO ASSIST THE STEADY
STATE SOLUTION WITH A SUPPORTED SEAT, AS EXPLAINED IN THE DOCUMENT.
THIS COMPONENT MUST BE INCLUDED IN THE MODEL GENERATION PROGRAM INPUT
FILE FOR ALL EASIER MODELS.

************** AQ OUTPUTS **************

VS - VELOCITY OF SOUND (FT/SEC)
RHOS - AIR DENSITY (SLUG/FT**3)

************** AQ INPUTS **************

H - HEIGHT ABOVE SEA LEVEL
WIN - X,Y,Z EARTH SYSTEM WIND COMPONENTS (FT/SEC)
DP - BAROMETRIC PRESSURE AT THE REFERENCE ALTITUDE (IN., HG)
     (AN UNINITIALIZED OR NON-POSITIVE VALUE OF BP
     CAUSES A STANDARD ATMOSPHERE TO BE USED)
TCL - TEMPERATURE AT THE REFERENCE ALTITUDE (DEF F)
SW - GRAVITY SWITCH FOR UNSUPPORTED SEAT STEADY STATE CALCULATION
     0 = GRAVITY OFF (UNSUPPORTED SEAT)
     1 = GRAVITY ON

******************************************************************************

DO 5 I=1,3
5    WIND(I) = WIN(I)

SSFLG = 1.
IF(SW.NE.0) FL1 = SW
IF(FL1.EQ.0 .AND. INST.EQ.31) SSFLG = 0

*** CHECK TO SEE IF THE CALC XIC COMMAND HAS BEEN GIVEN ***

IF((FL2.EQ.1) .OR. FL2) GO TO 70
IF(INCAL.EQ.1) GO TO 20
WRITE(6,10)
10 FORMAT(1x,1x,\"WARNING - THE CALC XIC COMMAND HAS NOT BEEN*,
         \* GIVEN \rangle\rangle EXECUTION TERMINATED. */\))
STOP

******************************************************************************

***** INITIALIZATION *****

294
C ***************
C 20 VS = RHO = 0
   IF(SW.EQ.0.99999) SW = 1.0
   FL = SW
   FL2 = 1.
   IF(BP .LE. 0.0 .OR. BP .EQ. 0.999999) BP=0.0
   BPE = BP
   IF(BPE .EQ. 0.0) H = TE = 0.0
   IF(TE.EQ.0.0) GO TO 70
   ADP = (BP + 144.1)/0.038
   ATE = TE + 460.
   TG = ATE + 0.003566 * H
   TRATIO = ATE/TG
   PO = ADP * (TRATIO)**5.256
   GO TO 70
C
C ENTRY ATMOS
C
30   WIN(1) = WINO(1)
C
C IF(3BP .GE. 0.0) GO TO 60
C
C ***** STANDARD ATMOSPHERE *****
C
   IF(H.GT.35332.) GO TO 40
C
C ALTITUDE BELOW THE TROPOPAUSE.......
C
   TRATIO = 1.0 - 0.0000066709 * H
   PRATIO = TRATIO**5.256
   VS = 1110.75 * SQR(TKATIO)
   GO TO 50
C
C ALTITUDE ABOVE THE TROPOPAUSE.......  
C
40    PRATIO = 10.**((4705.-H)/4621.)
   VS = 970.9979
C
50    RHO = 2962.*PRATIO/(VS**2)
   GO TO 70
C
C ***** NON-STANDARD ATMOSPHERE *****
C
60    T = 10. - 0.003566 * H
   P = PO * (T/TO)**5.456
   VS = (49.02) * SQR(T)
   RHO = P/(1715.*T)
C
70    RETURN
CEND
SUORoutine AM (OKE, RAD, PTS, PT1, * FL, PRT, EXP, XR, GAP, GYL, GZL, ORN, RDL, * DR, GA, GY, GZ)

This routine writes onto tape7 aeroMedical Parameters and variables
to be used by program AEROMED. no more than 4000 variable sets are
written at a time interval of no less than 0.001 seconds.

Design by C.L. West
Last modified - December 6, 1980

***** Outputs *****

OKE - Dynamic response
RAD - acceleration radical
PT5 - Current number of data sets written to tape 3
PTI - Value of time when the last data set was written
      onto tapes

***** Inputs *****

FL - Flag to initiate aeroMedical calculation (1 = start)
PRT - Program aeroMedical flag to print the load factors, dynamic
      response, and the acceleration radical (1 = print)
EXP - Medical injury exponent
GAP - The limit value for the x-axis positive aeroMedical
      load factor (G)
GAN - The limit value for the x-axis negative aeroMedical
      load factor (G)
GYP - The limit value for the y-axis aeroMedical load factor (G)
GZL - The limit value for the z-axis negative aeroMedical load
      factor (G)
DOK - Limit value of the dynamic response when the acceleration
      vector is forward of the plane of the seat back
URN - Limit value of the dynamic response when the acceleration
      vector is aft of the plane of the seat back
RDL - Acceleration radical limit
DR - Dynamic response
GA - X axis load factor (G)
GY - Y axis load factor (G)
GZ - Z axis load factor (G)

COMMON /CICAL/ ICICAL
COMMON /COUNLY/ INSI
COMMON /CTIME/ TIME
COMMON /CPFLAG/ ULM, ITINC
COMMON /CIO/ LREAD, LWRITE, IUDAG

************** Initialization **************

******** Initialization **************

IF(ICAL.NE.1) GO TO 20
IF(PRT.EQ.0.99999) PRT = 0.
IF(EXP.EQ.0.99999) EXP = 2.
IF(GAP.EQ.0.99999) GAP = 35.
IF(GAN.EQ.0.99999) GAN = 30.
IF(GYL.EQ.0.99999) GYL = 15.
IF(GZL.EQ.0.99999) GZL = 12.
IF(URP.EQ.0.99999) URN = 12.
IF(DRN.EQ.0.99999) DRN = 12.
IF(RDL.EQ.0.99999) RDL = 1.0
PTS = 0
PII = 0
C
WRITE AEROMEDICAL PARAMETERS ONTO TAPE7 .......

WRITE(7,10) PRT, EXP, GXP, GXN, GYL, GZL, DRP, DRN, RDL
10 FORMAT(9F12.4)
C
///////////////////////////////////////////////////////////////////////////////
C
20 DKR = DR
C
CALCULATE THE ACCELERATION RADICAL ......

GXL = GXP
IF(-GXP.LT.0) GXL = GXN
DRL = DRP
IF(-DRP.LT.0) DRL = DRN
IF(GZ.GE.0) RADZ = (DRP/URL)**2
IF(GZ.LT.0) RADZ = (GZ/GZL)**2
RAD = SQRT((GX/GXL)**2 + (GY/GYL)**2 + RADZ)
C
WRITE AEROMEDICAL VARIABLES ONTO TAPE7 .......

IF(FL.NE.1.) GO TO 30
IF(PTS.GE.4000.) GO TO 30
IF(TIME.LT.PTI+GGL) GO TO 30
IF(TITINC.NE.1) GO TO 30
C
IF(INST.EQ.26) WRITE (7,10) TIME, DR, GX, GY, GZ
PTI = TIME
PTS = PTS + 1.
C
30 RETURN
END
SUBROUTINE AP (TCX,TCL,F,SW,ALPHA,CX,CZ, 
* UP,XPC,PA,EPL,ZEM,SRP,UST,EST,WST,XAP,EAP)

***** FORCES AND MOMENTS ON A SEAT FROM AN ATTACHED PLATE *****

DESIGNED BY C.L. WEST
LAST MODIFIED - DECEMBER 6, 1980

************* AP TABLES *************

TCX - PLATE SYSTEM X-AXIS FORCE COEFFICIENT TABLE
THE INDEPENDENT VARIABLE IS THE PLATE ANGLE OF ATTACK (DEG).
THE DEPENDENT VARIABLE IS THE PLATE X-AXIS FORCE COEFFICIENT.

TCZ - PLATE SYSTEM Z-AXIS FORCE COEFFICIENT TABLE
THE INDEPENDENT VARIABLE IS THE PLATE ANGLE OF ATTACK (DEG).
THE DEPENDENT VARIABLE IS THE PLATE Z-AXIS FORCE COEFFICIENT.

************* AP OUTPUTS *************

F(3) - X,Y,Z SEAT BODY AXIS FORCE COMPONENTS (LB)
T(3) - X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS (FT-LB)
SW - FLAG SET WHEN THE PLATE CENTROID ENTERS THE WINDSTREAM (1 = PENETRATION)
ALPHA - PLATE ANGLE OF ATTACK (DEG)
CX - X AXIS FORCE COEFFICIENT
CZ - Z AXIS FORCE COEFFICIENT

************* AP INPUTS *************

UP - EJECTION DIRECTION FLAG WRT THE AIRPLANE
   (+ = UPWARD  - = DOWNWARD)
XPC(3) - X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE PLATE CENTROID (FT)
PA - REFERENCE AREA OF THE ATTACHED PLATE (FT**2)
EPL(3) - SEAT TO PLATE EULER ANGLES (DEG)
ZEM - AIRPLANE BODY Z-AXIS POSITION OF THE PLATE CENTROID WHEN IT ENTERS WINDSTREAM (FT)
SRP(3) - X,Y,Z EARTH POSITION VECTOR OF THE SEAT REFERENCE POINT (FT)
UST(3) - X,Y,Z SEAT BODY AXIS SYSTEM VELOCITY COMPONENTS OF THE SEAT (FT/SEC)
EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
WST(3) - X,Y,Z SEAT BODY AXIS SYSTEM ANGULAR VELOCITY COMPONENTS OF THE SEAT (DEG/SEC)
XAP(3) - X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE AIRPLANE CENTER OF GRAVITY (FT)
EAP(3) - EARTH TO AIRPLANE EULER ANGLES (DEG)

///CALCULATION DIMENSIONS /////////////

DIMENSION TCX(5),TCL(5),F(3),T(3),XPC(3),EPL(3),
SRP(3), UST(3), EST(3), WST(3), XAP(3), EAP(3)

* INTERNAL DIMENSIONS ....

DIMENSION EPLLR(3), EAPIR(3), ESTIR(3), WSTIR(3), DES(3, 3),
    D3ST(3, 3), D3EA(3, 3), XPCA(3), XPCE(3), UPLE(3),
    D3SP(3, 3), D3EP(3, 3), UPL(3), UW(3), VU(3),
    D3PS(3, 3), FP(3)

COMMUN /CTIME/ TIME
COMMUN /CICAL/ ICICAL
COMMUN /COVRAL/ INST
COMMUN /C1O/ IREAD, IWRITE, I1OAG

DATA FP(2) / 0. /
DATA RPD, DPR / 0.1745329, 57.29578 /

********************************************************************************
**** INITIALIZATION *****
********************************************************************************

IF (ICICAL .NE. 1) GO TO 20
IF (UP .EQ. 0.99999) UP = 1.
IF (ZEM .EQ. 0.99999) ZEM = 0
SW = 0
IF (ZEM .EQ. 0) SW = 1.
DO 14 I = 1, 3
IF (XAP(I) .EQ. 0.99999) XAP(I) = 0
IF (EAP(I) .EQ. 0.99999) EAP(I) = 0
F(I) = 0
14 T(I) = 0

 будем для воды

BYPASS ROUTINE IF DURING STEADY STATE WHEN THE PLATE IS NOT INITIALLY
IN THE WINDSTREAM ....

20 IF (INST .EQ. 31) AND. SW .EQ. 0) GO TO 70

CONVERT FROM DEGREES TO RADIANS ....

DO 30 I = 1, 3
30 ESTIR(I) = EST(I) * RPD

CALCULATE THE DIRECTION COSINE MACRIGIES ....

CALL DIRCOS (DES, ESTIR)
CALL TRANS (DEST, DES, 3, 3)

CONTROL FLAGS ....

IF (SW .EQ. 1.0) GO TO 50

CALCULATE THE CENTROID POSITION IN THE AIRPLANE SYSTEM ....

DO 40 I = 1, 3
40 EAPIR(I) = EAP(I) * RPD
CALL DIRCOS (DEA, EAPIR)
CALL VECXYZ (XPCE, XPC, SRP, DEST, 2)
CALL VECXYZ (XPC, XPE, XAP, DEA, 1)

RETURN IF THE PLATE HAS NOT PENETRATED THE WINDSTREAM

IF(ZEM*UP, LT, XPC(A(3))*UP) GO TO 70

WRITE EMERGENCE MESSAGE

IF(INST.EQ.26 .AND. SW.EQ.0) WRITE(6,45) TIME

45 FORMAT(5X,'AERODYNAMIC PLATE PENETRATION AT TIME = ',

   F10.4, F10.4, F8.2)

IF(1CCAL.NE.1) SW = 1

***** PLATE PENETRATION *****

CONVERT FROM DEGREES TO RADIANS

DO 55 I=1,3
   EPLK(I) = EPL(I) * RPD
   WSTK(I) = WST(I) * RPD

CALCULATE THE DIRECTION COSINE MATRICES

CALL DIRCOS (DSP, EPLIR)
CALL TRANS (DSP, D53, 3)
CALL MATMPY (DEP, DSP, DE5, 3, 3, 3)

DETERMINE THE VELOCITY OF THE PLATE CENTROID IN THE EARTH SYSTEM

CALL VELXYZ (UPLE, UST, XPC, WSTIR, DEST)

OBTAIN THE AIR DENSITY AND WIND VELOCITY

CALL ATMOS (VS, KHU, -SKP(3), WM, U, O, O)

SUBTRACT THE WIND VELOCITY FROM THE PLATE VELOCITY

DO 60 I=1,3
   UT(I) = UPLE(I) - WM(I)

TRANSFORM THE EARTH VELOCITY INTO THE PLATE SYSTEM

CALL MATMPY (UPL, DEP, UT, 3, 3, 3)

CALCULATE THE AIRSPEED OF THE PLATE

CALL DOTPRD (VoAR2, UPL, UPL, 3)

DETERMINE THE PLATE ANGLE OF ATTACK

ALPHA = ARTAN2(UPL(3), UPL(1)) * DPR

PERFORM THE TABLE SEARCH FOR CX AND CALCULATE ITS FORCE

NCIA = TCX(2)
CX = TBLUX (ALPHA, TCX(4), TCX(NTCX*4), 1, -NTCX)
FP(1) = CX * .5 * KHO * VBAR2 * PA

C PERFORM THE TABLE SEARCH FOR CZ AND CALCULATE ITS FORCE .......

C NTCZ = TCZ(2)
C CZ = TBLU1 (ALPHA, TCZ(4), TCZ(NTCZ+4), 1, -NTCZ)
FP(3) = CZ * .5 * KHO * VBAR2 * PA

C TRANSFORM THE FORCES TO THE SEAT SYSTEM .......

C CALL TRANS (OPS, DSP, 3, 3)
CALL MATMPY (F, DPS, FP, 3, 3, 1)

C CALCULATE THE MOMENTS ON THE SEAT FROM THE PLATE .......

C CALL CRSPRD (T, XPAR, F)

70 RETURN
END
**SuoROIINt**

*F,T,ALPHA,BETA,VMACH,Q,CX,CY,CZ,CL,CM,CN,EXL,EXA,*

*OFF,UP,ZWS,XEM,CDX,ECX,ECY,ECZ,CLP,CMQ,CRN,S,SRP,*

*UST,EST,WST,DSA,SRA,RUN*

************** AS TABLES **************

**TAE - EXPOSED AREA TABLE**

- **THE INDEPENDENT VARIABLE IS THE EXPOSED LENGTH (FT).**
- **THE DEPENDENT VARIABLE IS THE EXPOSED AREA (FT**²**).**

************** AS OUTPUTS **************

- **F,I(3) - X,y,z SEAT BODY AXIS AERO-DYNAMIC FORCE COMPONENTS (LB)**
- **T,I(3) - X,y,z SEAT BODY AXIS AERO-DYNAMIC TORQUE COMPONENTS (FT-LB)**
- **ALPHA - SEAT ANGLE OF ATTACK (DEG)**
- **BETA - SEAT SLIP ANGLE (DEG)**
- **VMACH - SEAT MACH NUMBER**
- **Q - DYNAMIC PRESSURE (LB/FT**²**)**
- **CX - SEAT BODY X-AXIS FORCE COEFFICIENT**
- **CY - SEAT BODY Y-AXIS FORCE COEFFICIENT**
- **CZ - SEAT BODY Z-AXIS FORCE COEFFICIENT**
- **CL - SEAT BODY AXIS ROLLING MOMENT COEFFICIENT**
- **CM - SEAT BODY AXIS PITCHING MOMENT COEFFICIENT**
- **CN - SEAT BODY AXIS YAWING MOMENT COEFFICIENT**
- **EXL - SEAT/CREW EXPRESSED LENGTH DURING EMERGENCE (FT)**
- **EXA - EXPRESSED AREA DURING EMERGENCE (FT**²**)**
- **CENT(3) - X,y,z SEAT BODY AXIS POSITION VECTOR OF THE CENTROID OF THE EMERGED AREA (FT)**
- **TCZ(20) - SEAT CENTROID LOCATION ARRAY (FT)**
- **MU - HYDRAULIC DIAMETER (FT)**

************** AS INPUTS **************

- **OFF - FLAG TO INDICATE SEAT/RAIL SEPARATION (1 = SEPARATION)**
- **UP - EJECTION DIRECTION FLAG WRt THE AIRPLANE (+1 = UPWARD, -1 = DOWNWARD)**
- **ZWS - AIRPLANE BODY Z-AXIS POSITION OF THE WINDSTREAM**
- **BOUNDARY LAYER AT THE POINT OF SEAT PENETRATION (FT)**
- **XEM - X,y,z SEAT BODY AXIS POSITION VECTOR OF THE INITIAL POINT TO PENETRATE THE WINDSTREAM (FT)**
- **CDX - SEAT BODY X-AXIS POSITION OF THE CENTER OF PRESSURE DURING SEAT EMERGENCE (FT)**
- **ECX - SEAT BODY X-AXIS EMERGENCE COEFFICIENT**
- **ELY - SEAT BODY Y-AXIS EMERGENCE COEFFICIENT**
- **ELZ - SEAT BODY Z-AXIS EMERGENCE COEFFICIENT**
- **CLP - ROLL DAMPING DERIVATIVE (DEG-1)**
- **CMQ - PITCH DAMPING DERIVATIVE (DEG-1)**
- **CRN - YAW DAMPING DERIVATIVE (DEG-1)**
- **S - SEAT REFERENCE AREA (FT**²**)**
- **SRP(3) - X,y,z EARTH POSITION VECTOR OF THE SEAT REFERENCE POINT (FT)**
- **UST(3) - X,y,z SEAT BODY AXIS SYSTEM VELOCITY COMPONENTS OF THE SEAT (FT/SEC)**
- **EST(3) - EARTH TO SEAT EULER ANGLES (DEG)**
- **WST(3) - X,y,z SEAT BODY AXIS SYSTEM ANGULAR VELOCITIES OF THE SEAT (DEG/SEC)**

302
C  DSA(3,3) - SEAT TO AIRPLANE DIRECTION COSINES
C  SRA  - X,Y,Z AIRPLANE BODY AXIS POSITION VECTOR OF THE
C  RON  - SUSTAINER ROCKET FLAG (1=ON,0=OFF)
C
C******************************************************************************
C
C DIMENSION TAE(5),F(3),T(3),XEM(3),SRP(3),UST(3),
C    EST(3),WST(3),DSA(3,3),SRA(3)
C DIMENSION ALF(72),BLT(6),AMACH(4),
C    COEF(6),CENT(3),DES(3,3),UW(3),UWB(3),UO(3),
C    XI(3),CONS(4),SC(3),XEMA(3),ESTIR(3),TCZ(20)
C
C COMMON /CICCAL/ ICCAL
C COMMON /COVRLY/ INST
C COMMON /CIO/ IRAD,WRITE,DIAG
C
C COMMON /RKTON/
C    ICXON(18,6,4),
C    ICYON(18,6,4),
C    ICZON(18,6,4),
C    ICLON(18,6,4),
C    ILMON(18,6,4),
C    ICNON(18,6,4)
C
C COMMON /RKTOFF/
C    ICAOFF(18,6,4),
C    ICYOFF(18,6,4),
C    ICZOFF(18,6,4),
C    ICLOFF(18,6,4),
C    ICNON(18,6,4),
C    ICMOFF(18,8,4)
C
C DATA RPD,DPR / .01745329, 57.29578 /
C
C DATA ALF /
C    0.0 , 5.0 , 10.0 , 15.0 , 20.0 , 25.0 ,
C    30.0 , 35.0 , 40.0 , 45.0 , 50.0 , 55.0 ,
C    60.0 , 65.0 , 70.0 , 75.0 , 80.0 , 85.0 ,
C    90.0 , 95.0 ,100.0 ,105.0 ,110.0 ,115.0 ,
C   120.0 ,125.0 ,130.0 ,135.0 ,140.0 ,145.0 ,
C   150.0 ,155.0 ,160.0 ,165.0 ,170.0 ,175.0 ,
C   180.0 ,185.0 ,190.0 ,195.0 ,200.0 ,205.0 ,
C   210.0 ,215.0 ,220.0 ,225.0 ,230.0 ,235.0 ,
C   240.0 ,245.0 ,250.0 ,255.0 ,260.0 ,265.0 ,
C   270.0 ,275.0 ,280.0 ,285.0 ,290.0 ,295.0 ,
C   300.0 ,305.0 ,310.0 ,315.0 ,320.0 ,325.0 ,
C   330.0 ,335.0 ,340.0 ,345.0 ,350.0 ,355.0 /
C
C DATA BET /
C    0.0 , 5.0 , 10.0 , 15.0 , 30.0 , 45.0 /
C
C NOTE - BY CLASSIC DEFINITION OF TERMS, BETA HERE IS ACTUALLY PSI,
C WHICH IS ALSO (-BETA).
C
C THIS PECULIARITY WAS ADOPTED TO ACCOMMODATE CONVENTIONAL TABLE
C LOOK UP ROUTINES WHICH DEMAND THAT THE INDEPENDENT VARIABLE BE
C LISTED IN ASCENDING ORDER.
C
C 303
The aerodynamic data here are packed in octal integer form at the rate of four coefficients per computer word according to following rationale and procedure.

1. All coefficients lie within the range of -1.5 to 1.5.
2. The maximum positive octal integer available in 1/4 of a computer word is 37777. (10383 decimal)
3. An integer version of each coefficient is calculated as follows:

A. Let the coefficient total range of 3.0 correspond to the available integer range of 16383.
B. Then the integer representation is obtained

ICX = (((CX - (-1.5))/3.0)*16383.1

The resulting integer is automatically stored as an octal number and is an accurate representation of the coefficient to approximately four decimal places.

C. The values of ICML, ICMN and ICY have been set equal to zero where-beta is equal to zero.

DATA (((ICXUN (I,J,K),I=1,16),J=1,1),K=1,1) /
 0.012006040670207.5x8,4022105451151115628,121413166313540137328,14121432415031163638,202342104321600203238,20382055021127216368,234362417324644250162,24733245452512726375,2726273402721263768,264132576325551250770,2506257625762576255468,2531122767724343253508,23465227122217214342,03011731415724150058,141601345512706121378,1121012530757551706386304057506455518,05557057250042036148
*
DATA (((ICXUN (I,J,K),I=1,18),J=2,2),K=1,1) /
 0.061700300505072558,07031954311113117068,124561277613526142478,14564140715203171288,202452073320762062058,2071521013212224217368,23511242224613251168,25241251425344265168,273772741273712270608,26493269162552267078,617420362614220078,255052513324407236128,2357127172214212343,02401272160514125177258,140613244215257711378,11060161110741067238,05221057405613054568,05561056106624065718
*
DATA (((ICXUN (I,J,K),I=1,1,2),J=3,3),K=1,1) /
 0.0601303597070307305b,010031037510777116378,12474131631353412128,17771554716652176138,2436226026722160521268,2112021147121752258,23450241432712253128,27514205262770275298,3000227427512272358,267026743463304518,2668547352704264043,2575525322407340708,2372521352223412123b,200511704416007147786,1444413271254016298,1671011056074071849,0404310050105601053378,09531056206197063308
*
DATA (((ICXUN (I,J,K),I=1,1),J=4,4),K=1,1) /
 0.0611203406620673418,0764510401111111222078,12545132441370444148,

304
DATA ((ICYON (1,JvK)tl1118))vJ=1v1)vK=2vZ)/
*1777717777177777777B,17771777777777777778,17777177771777777778,
*17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
17777177771777777778,17777177771777777778,17777177771777777778,
| DATA | ((ILIL)ON (1,J,K),1=3,3),K=3,3) | 1.03b16102371000307454a,072c30734007864016618,10263104711537106568,076a2507507763641017b,1003j1226213704150585,150651833007173058,175c406571751717517b,170522034520570211675,215232074c5004227618,23252236852407241566,24462250162553126322,26670270152710127224b,2733027342736627238b,267682643626144250078,25250254562404732350b |
| DATA | ((ILIL)ON (1,J,K),1=1,1d),J=3,3),K=3,3) | 1.0c552347272231421335b0,202c13711950744153b8,13001232711417107585,1055410e71071307424b,0727007u7074u1100564b,1030701545054210635b,07557575007613100748,165b42064013361644278,15151362471623311172,175b620475716217758b,200u42032520e61211568,2411220012234722664b,231l6235752413424465b,2506c34406265576242z8,26746271252725427255b,272602724726727234,2627502505210325553b,2520525374202233048 |
| DATA | ((ILIL)ON (1,J,K),1=1,1b),J=4,4),K=3,3) | 2.7311231302<15212438,201217001575146058,15323124421163511164,10770106510146075478,07273074230762310038,1020110303104601063b,075b50761077631022,10701171212761137228,15061156361576216638,17442200302400113b,2030020732070212238b,21455217162242432368,255552474244424762b,2951266542636266118b,276732710u2772747355b,274052732716672249b,2671026052610525578,25151445023706230278 |
| DATA | ((ILIL)ON (1,J,K),1=1,1a),J=5,5),K=3,3) | 2.344322846220021216a,20343174361653503b53036b,14255135561253112037b,1143311012019511072b,101521013710240103505b,103530606110111258b,16243163341516110478d,113b2117341234713171085,14c61451015176716564b,17364200233732365b8,210b621212132621404b5,21542216722120322468b,2737423622432406b8,253213776225026600b8,26720270427716227238b,271442711655070b,28b0652632934725059b,24595242152390423008b |
| DATA | ((ILIL)ON (1,J,K),1=1,1),J=6,6),K=3,3) | 2.1634225422220421494b,20751206313724164778,15041150161416413515b,1315112561123101287b,1172711760117201171b,116b4117301205412252b,11341114791164612148b,124661309153941417b,150014503161741668b,172741772303520736b8,2171121422b00127058,21776221052230225748,2211422057234523763b0,2435324676702510625338b,255342565425666b256278b,25b0425402533325138b,246b642432422420108b,235b2324022743224018b |
| DATA | ((ILIL)ON (1,J,K),1=1,1),J=7,7),K=3,3) | 2.3675315222727121260b,200631666015020145728,134741251411751111504b,111b651571015460b5b0,07535673x00730730c57b,07630755207661102028,0714047610750310066b,10727120441330341321b,157111572163531663b8,1720151200321566b6,204b120624211221474b9,20275223722616230438b,252122660240262158b1,24474273121526221728b,2067027077727027475b9,275b12763b2760327259b,27391267776443265068b,256432474744112386b8 |
| DATA | ((ILIL)ON (1,J,K),1=1,1),J=8,8),K=3,3) | 2.374123062215611b6b,2010016771352123145126b,135612510120511650b,115c1112126731448b,07434716307107140b,07271074007652102128,0744223460935510105b,10601166410527137618,14750155221607116474b, |
DATA (((ICM0N(I,J,K),I=1,18),J=5,5),K=3,3) / 
.21232132021450214434,14502145221634,1217652204622076220528, 
.220142174621654216006,214767212132521448,212302116721204207028, 
.203332011747517175,166521635516065150740,159571539415127147428, 
.14546145671421314117,141221410421314078,143671545314751151468, 
.15352135611604162578,16521170517322176368,201472042220737211178, 
.21415215702170421748,216512153721477213656,213132123241330213378
/ 
DATA (((ICM0N(I,J,K),I=1,18),J=5,5),K=3,3) / 
.2063420726207221118,2116121732125213558,21412214762157216018, 
.216212154521774211418,213502130321224211238,210112065420476207218, 
.2026520003175317474,170371660216734161628,175505551415355151478, 
.147721403341541516,14520145334047147058,150451523515405156118, 
.1564310711633516758,17461721417413175758,177432010220254204268, 
.20606270020741210005,203221050210522100118,210742107221100211188
/ 
DATA (((ICM0N(I,J,K),I=1,18),J=5,5),K=3,3) / 
.21511110052172722060,2065220342175721708,216432162421605215188, 
.2150121021621210109,217442163210125538,210271521512165714178, 
.21011265072045020026,1747107346311155594,152001463741450147278, 
.1432014116137213682,1352101311137313140215,141761443414742153068, 
.15562151416444170228,1725417661766020778,203212001210200124158, 
.21752210222562223448,2334723423235221618,2204221612246721328
/ 
DATA (((ICM0N(I,J,K),I=1,18),J=5,5),K=3,3) / 
.21511110052172722060,2065220342175721708,216432162421605215188, 
.2150121021621210109,217442163210125538,210271521512165714178, 
.21011265072045020026,1747107346311155594,152001463741450147278, 
.1432014116137213682,1352101311137313140215,141761443414742153068, 
.15562151416444170228,1725417661766020778,203212001210200124158, 
.21752210222562223448,2334723423235221618,2204221612246721328
/
<table>
<thead>
<tr>
<th>DATA</th>
<th>(ICJNON(I,J,K), I=1,18), J=2,2, K=3,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200222001222004220131b, 2013222120122720121b, 201152010420113201268</td>
</tr>
<tr>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td></td>
<td>204472004620451204238, 20401203622034203558, 20367203612037020438,</td>
</tr>
<tr>
<td></td>
<td>20424204322042368, 2037420352021202068, 201222012042120201028, 201152010420113201268</td>
</tr>
</tbody>
</table>
DATA (((ICNON(I,J,K), I=1,13), J=1,4), K=1,4) / 
20722277122664260418, 2011206120566205584, 2054520555120551205068, 
2054520511205352018, 20541205620411203348, 203472034203020024, 
20267327463952065200618, 202642061720537205028, 
20456240253203228, 202712020120120021, 20112011201201436, 
202272630123631203678, 202605241320413203738, 20344203142022376, 
2025420330241204098, 02020545205052050564, 202642057120562205568, 

DATA (((ICNON(J,K), I=1,13), J=1,4), K=1,4) / 
14552145121432521422, 2137521412131213058, 212062123121121208, 
212252126211421500, 212212017521052012036, 2077120722072020704, 
2005606662071020748, 210102013220321031068, 21033210042073320676, 
2060420612055205008, 0454020402901205407, 2040020422044320476, 
20655207032073226748, 207402074620752207678, 2077621005210120158, 
21044211002123211548, 212232124621310213318, 21353213521365213668, 

DATA (((ICXUFF(I,J,K), I=1,13), J=1,4), K=1,4) / 
22103206742203720104, 22033220221767217568, 21716216671630216358, 
215762155521522524, 215352415224142268, 213752133421310212748, 
211272123621242428, 221232120361212038, 2120021722144211058, 
2074210072075220122, 20665206412062406056, 2057720612060206528, 
21036210632121611648, 211442115521167212108, 21241212751346214148, 
21453215362156316139, 21635215121602161758, 21076217172177217758, 

DATA (((ICXOFF(I,J,K), I=1,18), J=1,4), K=1,4) / 
0636064030710717778, 1055114731242134278, 140621473715621162378, 
16533167341722460168, 211002175216215458, 22221227723342241768, 
25901606226325270658, 2755302243042305746, 30500303330335630148, 
27451267262621603129, 22560571225512051430, 24522744227350231768, 
2252230632242153040, 0300167401027151518, 1420013217244111738, 
112701062216034100358, 0346067104067300666568, 00622067160669667038, 

DATA (((ICXOFF(I,J,K), I=1,18), J=2,2), K=1,4) / 
066220636007045075538, 103471125712153126468, 1351111401515004154168, 
161361637316741172604, 17502209310213244328, 20475213012323721037, 
25100256226360272560303330211303308, 304303363017277338, 
27232050421725768, 25637256042541234178, 2457124012377231578, 
23664252342104211738, 203276523154214444, 13564130101267115748, 
1076010275101260758, 071356671305642063678, 06376063090640364758, 

DATA (((ICOFF(I,J,K), I=1,13), J=3,3), K=1,4) / 
0765908773225108, 10101203115517778, 126271313313071431618, 
1521115161335150717777320310, 210142170322723232648, 
2563260562657127448, 27550301630275304078, 3050130423205527728, 
26052627732655130378, 215236513526022524268, 25016244552401323218, 
234172256204027778, 17642165611540145248, 135601272121120114228, 
10037100357650375118, 071366640406063018, 06323005506167063459, 

DATA (((ICOFF(I,J,K), I=1,13), J=4,4), K=1,4) / 
04524647685043405528, 003604714707474104048, 11015115021224113108, 
1371513621590161208, 2456714737157174618, 201020213622050267258, 
251042160420653227430, 2764330152034500478, 3091250313302107268, 
274542724027114536096, 26052824276537266719, 2531024556542402348, 
23402427722175523024, 1791616455155144656, 1354426241174312418, 
195701931707751075128, 075200710206246664120, 06070061106251564758, 

324
DATA ((ICLOFF(I,J,K),I=1,1b),J=1,1),K=3,3 /
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
 177711777117771177711777,177711777117711777,17771177711777117777,
<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2023</td>
<td>12024</td>
<td>02626</td>
<td>02157</td>
<td>20115</td>
<td>02006</td>
</tr>
<tr>
<td></td>
<td>02200</td>
<td>02717</td>
<td>02776</td>
<td>17764</td>
<td>02005</td>
</tr>
<tr>
<td></td>
<td>52021</td>
<td>12203</td>
<td>12204</td>
<td>21152</td>
<td>12105</td>
</tr>
<tr>
<td></td>
<td>36421</td>
<td>20220</td>
<td>02201</td>
<td>12332</td>
<td>02113</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02120</td>
<td>12208</td>
<td>20132</td>
<td>02113</td>
</tr>
<tr>
<td></td>
<td>02112</td>
<td>12203</td>
<td>12204</td>
<td>21152</td>
<td>12105</td>
</tr>
<tr>
<td></td>
<td>02006</td>
<td>02717</td>
<td>02776</td>
<td>17764</td>
<td>02005</td>
</tr>
<tr>
<td></td>
<td>52021</td>
<td>12203</td>
<td>12204</td>
<td>21152</td>
<td>12105</td>
</tr>
<tr>
<td></td>
<td>36421</td>
<td>20220</td>
<td>02201</td>
<td>12332</td>
<td>02113</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02112</td>
<td>12202</td>
<td>12203</td>
<td>12204</td>
</tr>
<tr>
<td></td>
<td>12152</td>
<td>02113</td>
<td>02112</td>
<td>12203</td>
<td>12204</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02112</td>
<td>12202</td>
<td>12203</td>
<td>12204</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02112</td>
<td>12202</td>
<td>12203</td>
<td>12204</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02112</td>
<td>12202</td>
<td>12203</td>
<td>12204</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02112</td>
<td>12202</td>
<td>12203</td>
<td>12204</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02112</td>
<td>12202</td>
<td>12203</td>
<td>12204</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02112</td>
<td>12202</td>
<td>12203</td>
<td>12204</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02112</td>
<td>12202</td>
<td>12203</td>
<td>12204</td>
</tr>
<tr>
<td></td>
<td>20137</td>
<td>02112</td>
<td>12202</td>
<td>12203</td>
<td>12204</td>
</tr>
</tbody>
</table>

343
* Initialization *

** IF (ICCAL.NE.1) GO TO 40 **

** SET CERTAIN UNINITIALIZED PARAMETERS TO ZERO **

DO 5 J=1,3
   IF (XEM(J).EQ.0.9999) XEM(J) = 0
   IF (SRA(I).EQ.0.9999) SRA(I) = 0
   IF (DSA(J).EQ.0.9999) DSA(J,1) = 0
   IF (OFF.T.EQ.0.9999) OFF = 1.
   IF (UP.T.EQ.0.9999) UP = 1.
   IF (ZMS.T.EQ.0.9999) ZMS = 0
   IF (CDX.T.EQ.0.9999) CDX = 0
   IF (LCX.T.EQ.0.9999) LCX = 1.
   IF (LGY.T.EQ.0.9999) LGY = 1.
   IF (LCZ.T.EQ.0.9999) LCZ = 1.
   IF (CLP.T.EQ.0.9999) CLP = 0.
   IF (CMQ.T.EQ.0.9999) CMQ = 0.
   IF (CNR.T.EQ.0.9999) CNR = 0.
   IF (KON.T.EQ.0.9999) KON = 0.

** SET UP CONSTANTS FOR THE BOUNDARY LAYER PLANE EQUATION **

CONS(1) = CONS(2) = 0
CONS(3) = 1.

** SET UP THE CENTROID VECTOR **

CENT(1) = CDX
CENT(2) = CENT(3) = 0

** DETERMINE THE HYDRAULIC DIAMETER **

344
HO = SQRT(4.*S/3.14159)

C CALCULATE THE CENTROID TABLE ......
C
DO 10 I=1,20
10 TCZ(I) = 0
WRITE(0,15)
15 FORMAT('!5X,*--- CENTROID TABLE CALCULATED FOR COMPONENT*,
! * AS ---*,!16X,LENGTH*,8X,CENTROID*)
NTAE = TAE(2)
DO 20 I=2,NTAE
K=NTAE+2+1
20 TCZ(I)=(TCZ(I-1)*TAE(K)+.5*(TAE(K+1)-TAE(K))
! (TAE(3+1)+TAE(2+1)))/TAE(K+1)
WRITE(6,30) (TAE(I+3),TCZ(I),I=1,NTAE)
30 FORMAT('16X,F5.2,10X,F7.4)
C
ALPHA = BETA = VMACH = Q = EXL = EXA = 0
CX = CY = CZ = CL = CM = CN = 0
C
C ZERO OUT THE AERO FORCES AND TORQUES ......
C
DO 50 I=1,3
50 F(I) = T(I) = 0
C
C BYPASS ROUTINE DURING STEADY STATE WITH THE RAIL COMPONENT IN THE
C MODEL ......
C
IF (INSI.EQ.31 .AND. OFF.EQ.0) GO TO 110
C
IF (OFF.EQ.1.) GO TO 60
C
C CALCULATE XEM IN THE AIRPLANE SYSTEM ......
C
CALL VECXYZ (XEMA,XEM,SRHA,USA,2)
C
C CHECK TO SEE IF SEAT HAS PENETRATED THE BOUNDARY LAYER ......
C
IF(UMS*UP.LE.XEMA(3)*UP) GO TO 110
C
C CONVERT FROM DEGREES TO RADIANS ......
C
DO 70 I=1,3
70 ESTIK(I) = EST(I) * RPD
C
C DETERMINE ATMOSPHERIC PROPERTIES ......
C
CALL ATMS (VS,RHO,-SRP(3),WM,0,0,0)
C
C PUT THE WIND INTO THE BODY COORDINATES ......
C
CALL WIRCOS (DES,ESTIK)
CALL MATMPY (UM8,DES,UM3,3,1)
C
C ADD THE WIND VELOCITY TO THE SEAT VELOCITY ......
Determine the aero variables:

\[ U(1) = UST(1) - UWB(1) \]
\[ U(2) = UST(2) - UWB(2) \]
\[ U(3) = UST(3) - UWB(3) \]

\[ \text{IF}(U(1) \cdot E Q \cdot 0.0 \text{ AND } U(3) \cdot E Q \cdot 0.0) \]
\[ \text{ALPHA} = \text{ARCTAN2}(U(1), U(3)) \cdot UPR \]
\[ \text{CALL UOTPRD(VBAR2, U, U(3))} \]
\[ \text{VBAR} = \text{SQRT(VBAR2)} \]
\[ \text{BETA} = \text{ASIN}(U(1)) / \text{VBAR2} \]
\[ \text{VMACH} = \text{VBAR} / \text{VS} \]
\[ Q = .5 \cdot \text{RHO} \cdot \text{VBAR2} \]

Perform table lookup for aerodynamic coefficients:

\[ \text{TBLALPH} = \text{ALPHA} \]
\[ \text{IF}(\text{ALPHA} < 0.0) \]
\[ \text{TBLALPH} = \text{ALPHA} + 360.0 \]
\[ \text{TBLBETA} = \text{ABS}(\text{BETA}) \]
\[ \text{IF}(\text{BETAX} = 0.0) \text{ CALL TLU} \]
\[ \text{ICXOFF} = 72, 6, 4, \text{ALF}, \text{BET}, \text{AMACH}, \text{TBLALPH}, \]
\[ \text{TBLBETA}, \text{VMACH}, \text{COEF}, 6 \]
\[ \text{IF}(\text{BETAX} \neq 0.0) \text{ CALL TLU} \]
\[ \text{ICXON} = 72, 6, 4, \text{ALF}, \text{BET}, \text{AMACH}, \text{TBLALPH}, \]
\[ \text{TBLBETA}, \text{VMACH}, \text{COEF}, 6 \]

\[ \text{C} = \text{COEF}(1) \]
\[ \text{CY} = -\text{COEF}(2) \cdot \text{SIGN}(1., \text{BETA}) \]
\[ \text{CZ} = \text{COEF}(3) \]
\[ \text{CL} = -\text{COEF}(4) \cdot \text{SIGN}(1., \text{BETA}) \]
\[ \text{CM} = \text{COEF}(5) \]
\[ \text{CN} = -\text{COEF}(6) \cdot \text{SIGN}(1., \text{BETA}) \]

Bypass emerge calculations if seat is off rails:

\[ \text{IF}(\text{UFF} = 0.0) \text{ GO TO 90} \]

Calculate the aerodynamic forces and torques acting upon the seat/man as it is emerging from the airplane:

Calculate the seat z-axis unit vector direction cosines with respect to the airplane system:

\[ \text{DO } 60 \text{ I} = 1, 3 \]
\[ \text{DC}(1) = \text{DSA}(1, 3) \]

Calculate the point of intersection between the boundary layer plane and the line that both passes through XEMA and is parallel with the seat system z axis:

\[ \text{CONS}(4) = -2 \cdot \text{WS} \]
\[ \text{CALL LINEPL(XI, CU, NS, XEMA, UC)} \]

Determine the seat/man exposed length:

\[ \text{EXL} = \sqrt{((XI(1) - XEMA(1))^2 + (XI(2) - XEMA(2))^2 + (XI(3) - XEMA(3))^2)} \]

Calculate the exposed area from the table.
EXA = TBLU1(EXL,TAE(4),TAE(NTAE+4),1,-NTAE)

C CALCULATE THE AERO FORCES FROM THE AERO COEFFICIENTS, THE
C EXPOSED AREA, AND THE EMERGENCE COEFFICIENTS ........
C
QAREA = Q * EXA
F(1) = CX * QAREA * ECX
F(2) = CY * QAREA * ECY
F(3) = CZ * QAREA * ECZ

C CALCULATE THE Z-AXIS POSITION OF THE CENTROID ........
C
CENT(3) = XEM(3) - SIGN(1.,XEM(3)) * TBLU1(EXL,TAE(4),TCL(1),1,-NTAE)
C CALCULATE THE RAIL/SEAT TORQUES ........
C
CENT(1) = CDX
CALL CKSPRD (T,CENT,F)
C
GO TO 110
C
ADD DAMPING TERMS FOR AN AIRSPEED GREATER THAN .1 FT/SEC
C
90 IF(VBAR.LE.0.1) GO TO 100
C
HDU2V = HD/(VBAR+VBAR)
C
ADD ROLL DAMPING ........
C
CL = CL + CLP * WST(1) * HDU2V
C
ADD PITCH DAMPING ........
C
CM = CM + CMQ * WST(2) * HDU2V
C
ADD YAW DAMPING ........
C
CN = CN + CNK * WST(3) * HDU2V
C
C COMPUTE THE AERO FORCES AND MOMENTS ABOUT THE SRP ........
C
100 QS = Q * S
F(1) = CX * QS
F(2) = CY * QS
F(3) = CZ * QS
T(1) = CL * QS * HD
T(2) = CM * QS * HD
T(3) = CN * QS * HD
C
110 RETURN
END
SUBROUTINE CS (AIL, AILDOT, IAIL, ELE, ELEDOT, IELE, RUD, RUDDOT, IRUD, COA, TCA, TDA, COE, TCE, TDE, COR, TCR, TUR, TRM)

**** EASIEST AIRPLANE CONTROL SURFACE COMPONENT ****

DESIGNED BY C.L. WEST
LAST MODIFIED - DECEMBER 6, 1980

************** CS OUTPUTS **************
C AIL - AILERON DEFLECTION FROM TRIM POSITION (DEG)
C AILDOT - AILERON RATE (DEG/SEC)
C IAIL - INTEGRATION CONTROL
C ELE - ELEVATOR DEFLECTION FROM TRIM POSITION (DEG)
C ELEDOT - ELEVATOR RATE (DEG/SEC)
C IELE - INTEGRATION CONTROL
C RUD - RUDDER DEFLECTION FROM TRIM POSITION (DEG)
C RUDDOT - RUDDER RATE (DEG/SEC)
C IRUD - INTEGRATION CONTROL

************** CS INPUTS **************
C COA - AILERON COMMANDED POSITION (DEG)
C TCA - AILERON TIME CONSTANT (SEC)
C TDA - TIME DELAY AFTER WHICH THE AILERON RATE IS CALCULATED (SEC)
C COE - ELEVATOR COMMANDED POSITION (DEG)
C TCE - ELEVATOR TIME CONSTANT (SEC)
C TDE - TIME DELAY AFTER WHICH THE ELEVATOR RATE IS CALCULATED (SEC)
C COR - RUDDER COMMANDED POSITION (DEG)
C TCR - RUDDER TIME CONSTANT (SEC)
C TDR - TIME DELAY AFTER WHICH THE RUDDER RATE IS CALCULATED (SEC)
C TRM(4) - AIRPLANE THRUST AND CONTROL SURFACE POSITIONS AT TRIM
C TRM(1) - ENGINE THRUST (LB) -- NOT USED --
C TRM(2) - AILERON POSITION (DEG)
C TRM(3) - ELEVATOR POSITION (DEG)
C TRM(4) - RUDDER POSITION (DEG)

DIMENSION TRM(4)
COMMON /CTIME/ TIME
COMMON /CICCAL/ ICCAL
COMMON /CIU/ IREAD, IWRITE, IDIAG

**************
****** INITIALIZATION ******
**************

IF(ICCAL.NE.1) GO TO 10

IF(COA.EQ.0.99955) COA = 0
IF(COE.EQ.0.99999) COE = 0
IF(COR.EQ.0.99999) COR = 0

348
C
IF(TCA.EQ.0.99999) TCA = 0
IF(TCE.EQ.0.99999) TCE = 0
IF(TCR.EQ.0.99999) TCR = 0
C
IF(TDA.EQ.0.99999) TDA = 0
IF(TDE.EQ.0.99999) TDE = 0
IF(TDR.EQ.0.99999) TDR = 0
C
C ///////////////////////////////////////////////////
C
C ***** AILERON *****
C
10 IF(TCA.LE.0) AILD = 0
   IF(TCA.GT.0) CALL LAG (AILD,COA,AIL,TRM(2),TCA,TIME,TDA)
   IF(AILTNE.0) AILDOT = AILD
C
C ***** ELEVATOR *****
C
   IF(TCE.LE.0) ELED = 0
   IF(TCE.GT.0) CALL LAG (ELED,COE,ELE,TRM(3),TCE,TIME,TDE)
   IF(IELE.NE.0) ELEDOT = ELED
C
C ***** RUDDER *****
C
   IF(TCR.LE.0) RUDD = 0
   IF(TCR.GT.0) CALL LAG (RUDD,COR,RUD,TRM(4),TCR,TIME,TDR)
   IF(IRUDDNE.0) RUDDOT = RUDD
C
RETURN
END
SUBROUTINE CT (TCP,
               EF, EFDOT, IEF, EL, ELDOT, IEL, WK, WKDOT, IWK,
               WB, WBDOT, IWb,
               FL, FDN, FLA, TCA, FCS, TCS, CF, CEX, CV, TLO, PC, R, CVH, TSO,
               FSO, SW, UP, SAP, AAP, UCL, CSK, VI, PA, PT, CBP, CI, PMW, SK,
               CK, GAM, TF, CL, C2, b, BXP, TI, TDE, SRP, UST, EST, WST, XAP,
               UAP, EAP, WAP)

C
C *************** EASIEST CATAPULT COMPONENT **************
C
C DESIGNED BY C.L. WEST
C LAST MODIFIED - DECEMBER 6, 1980
C
C FORCES AND MOMENTS ACTING ON THE VEHICLE AND THE SEAT FROM
C A CLOSED TELESCOPING TUBE CATAPULT
C
C *************** CATAPULT TABLES ***************
C TCP - CATAPULT PROPELLANT CONSUMPTION TABLE
C THE INDEPENDENT VARIABLE IS THE PROPELLANT
C WEB CONSUMED (IN) AND THE DEPENDENT VARIABLE
C IS THE PROPELLANT CONSUMED (SLUGS)
C
C *************** CATAPULT OUTPUTS ***************
C
C INTERNAL FRICTION ENERGY ....
C
C EF - INTERNAL FRICTION ENERGY (FT-LB)
C EF DOT - INTERNAL FRICTION ENERGY RATE (FT-LB/SEC)
C IEF - INTEGRATION CONTROL
C
C HEAT LOSS ....
C
C EL - HEAT LOSS (FT-LB)
C EL DOT - HEAT LOSS RATE (FT-LB/SEC)
C IEL - INTEGRATION CONTROL
C
C CATAPULT WORK ....
C
C WK - CATAPULT WORK (FT-LB)
C WK DOT - CATAPULT WORK RATE (FT-LB/SEC)
C IWK - INTEGRATION CONTROL
C
C PROPELLANT WEB BURNED ....
C
C WB - PROPELLANT WEB BURNED (IN)
C WBDOT - PROPELLANT WEB BURN RATE (IN/SEC)
C IWb - INTEGRATION CONTROL
C
C FL - CATAPULT MODE FLAG
C 0 = PHLOW TO INITIATION
C 1 = CATAPULT IGNITION UP TO STRIPOFF
C 2 = CATAPULT STRIPOFF
C 3 = CATAPULT OFF
C FON - STRIPOFF FLAG FOR SUSTAINER ROCKET COMPONENT
C (1 = ROCKET ON)
FCA(3) - $x_yz$ AIRPLANE BODY AXIS FORCE COMPONENTS OF THE CATAPULT ON THE AIRPLANE (LB)

TCA(3) - $x_yz$ AIRPLANE BODY AXIS TORQUE COMPONENTS OF THE CATAPULT ON THE AIRPLANE (FT-LB)

FLS(3) - $x_yz$ SEAT BODY AXIS FORCE COMPONENTS OF THE CATAPULT ON THE SEAT (LB)

TCS(3) - $x_yz$ SEAT BODY AXIS TORQUE COMPONENTS OF THE CATAPULT ON THE SEAT (FT-LB)

CF - CATAPULT FORCE MAGNITUDE (LB)

CEx - CATAPULT EXTENSION (FT)

CV - CATAPULT EXTENSION VELOCITY (FT/SEC)

TLO - INITIAL LENGTH OF CATAPULT PRESSURE CHAMBER (IN)

PC - CIRCUMFERENCE OF CATAPULT PRESSURE CHAMBER (IN)

R - GAS CONSTANT (FT-LBF/Slug-K)

CVh - CONSTANT VOLUME SPECIFIC HEAT (FT-LBF/Slug-K)

TSO - CATAPULT STRIPOFF TIME (SEC)

FSD - CATAPULT FORCE AT STRIPOFF (LB)

*************** CATAPULT INPUTS ***************

SW - FLAG FOR CATAPULT IGNITION ( 1 = CATAPULT ON )

UP - EJECTION DIRECTION FLAG WRT THE AIRPLANE

-1 = UPWARD EJECTION

1 = DOWNWARD EJECTION

SAP(3) - SEAT ATTACHMENT POINT FOR THE CATAPULT (FT)

AAP(3) - AIRPLANE ATTACHMENT POINT FOR THE CATAPULT (FT)

UCL - UNLOADED CATAPULT LENGTH (FT)

CSK - CATAPULT STROKE (FT)

VI - INITIAL FREE VOLUME (IN**3)

PA - PISTON AREA (IN**2)

PT - TANG RELEASE PRESSURE (LB/IN**2)

CBP - CATAPULT BURST PRESSURE (LB/IN**2)

C - MASS OF TOTAL PROPELLANT (SLUGS)

CI - IGNITER PROPELLANT MASS (SLUGS)

PMW - PROPELLANT MOLECULAR WEIGHT (LB/(LB-MOLE))

SK - CATAPULT SPRING CONSTANT (LB/FT)

CK - CATAPULT DAMPING CONSTANT (LB/FT/SEC)

GAM - RATIO OF SPECIFIC HEATS

Tr - CONSTANT VOLUME FLAME TEMPERATURE (DEG K)

C1 - FRICTION PROPORTIONALITY CONSTANT

C2 - HEAT LOSS CONSTANT

B - BURN RATE PROPORTIONALITY CONSTANT (IN/SEC/(LB/IN**2))

BXP - BURN RATE EXPONENT

T1 - CATAPULT TEMPERATURE PRIOR TO IGNITION (DEG K)

TDE - CATAPULT FORCE DECAY TIME (SEC)

SKP(3) - $x_yz$ EARTH SYSTEM POSTION VECTOR OF THE SEAT REFERENCE POINT (FT)

UST(3) - $x_yz$ SEAT BODY AXIS VELOCITY VECTOR OF THE SEAT (FT/SEC)

EST(3) - EARTH TO SEAT EULER ANGLES (DEG)

WST(3) - $x_yz$ SEAT BODY AXIS ANGULAR VELOCITY OF THE SEAT (DEG/SEC)

XAP(3) - $x_yz$ EARTH SYSTEM POSITION VECTOR OF THE AIRPLANE (FT)

UAP(3) - $x_yz$ AIRPLANE BODY AXIS VELOCITY VECTOR OF THE AIRPLANE (FT/SEC)

EAP(3) - EARTH TO AIRPLANE EULER ANGLES (DEG)

WAP(3) - $x_yz$ AIRPLANE BODY AXIS ANGULAR VELOCITY
OF THE AIRPLANE (DEG/SEC)

DIMENSIONS OF CALLING ARGUMENTS .......

DIMENSION FCA(3), TCA(3), FCS(3), TCS(3), SAP(3), AAP(3),
- SRP(3), UST(3), EST(3), WST(3), XAP(3), UAP(3),
- EAP(3), WAP(3)

INTERNAL DIMENSIONS

DIMENSION DES(3,3), DSE(3,3), DEA(3,3), DAE(3,3),
- SAPE(3), AAPE(3), DKL(3), EXT(3), USAPE(3),
- UAPE(3), CDV(3), FCP(3), FSS(3), FSO(3),
- FC(3), CAUV(3), ESTIR(3), WSTIR(3), EAPIR(3), WAPIR(3)

COMMON / CTIME / TIME
COMMON / CICCAL / ICCAL
COMMON / COVRLY / INST
COMMON / CSSFLG / SSFLG
COMMON / CIQ / IREAD, IWRITE, IDIAG

DATA RPD / .01745329 /

**********************************************************************
****** INITIALIZATION ******
**********************************************************************

IF(ICCAL .NE. 1) GO TO 10

COMPUTE THE INITIAL LENGTH (TL0) AND CIRCUMFERENCE (PC) OF THE
CATAPULT PRESSURE CHAMBER .......

TL0 = V1/PA
PC = 2*SQRT(3.14159*PA)

CALCULATE THE GAS CONSTANT (R) AND THE CONSTANT VOLUME
SPECIFIC HEAT (CVH) .......

R = 69475.694/PMW
CVH = R/(GAM-1.0)

TYPE = BHCATAPULT
CF = FL = TSO = FSO = FON = 0
IF(UP .EQ. 0.99999) UP = 1.0
IF(TL .EQ. 0.99999) TDE = 0

DO 5 1=1,3
5 FCA(1) = TCA(1) = FCS(1) = TCS(1) = 0

**********************************************************************

BYPASS THE REMAINING CODE IF THE CATAPULT IS PAST THE
STRIP-OFF POINT ......

10 IF(FL .LE. 3.1) GO TO 170
FCP(1) = FCP(2) = FCP(3) = 0

CHANGE ANGULAR STATES FROM DEGREES TO RADIANS .......

352
DO 20 I=1,3
ESTIR(I) = EST(I) * RPD
WSTIR(I) = WST(I) * RPD
EAPIR(I) = EAP(I) * RPD
20 WAPIR(I) = WAP(I) * RPD

*------------------------------------------------------------------*
* DETERMINE THE VARIABLES CALCULATED FROM THE                     *
* POINT AND THE SEAT ATTACHMENT POINT                             *
*------------------------------------------------------------------*

C

DEFINE THE SEAT CATAPULT ATTACHMENT POINT IN THE EARTH SYSTEM (SAPE) ....

CALL DIRCOS (DES, ESTIR)
CALL TRANS (DSE, DES, 3, 3)
CALL VECXYZ (SAPE, SAP, SRP, DSE, 2)

C

DEFINE THE AIRPLANE CATAPULT ATTACHMENT POINT IN THE EARTH SYSTEM (AAPE) ....

CALL DIRCOS (DEA, EAPIR)
CALL TRANS (DAE, DEA, 3, 3)
CALL VECXYZ (AAPE, AAP, XAP, DAE, 2)

C

CALCULATE THE CATAPULT LENGTH COMPONENTS ......

DO 30 I=1,3
30 DXL(I) = SAPE(I) - AAPE(I)

DETERMINE THE DEFLECTED CATAPULT LENGTH ......

CATL = SQRT(DXL(1)**2 + DXL(2)**2 + DXL(3)**2)

DETERMINE UNIT VECTOR ALONG THE CATAPULT EXTENSION ......

DO 40 I=1,3
40 IF (CATL.NE.0) CXUV(I) = DXL(I) / CATL

CALCULATE THE CATAPULT EXTENSION ......
(CORRECTING FOR CATAPULT DIRECTION DURING TRIM)

FUDGE = 1
IF(INST.EQ.31. AND. DXL(3)*UP*DAE(3,3).GT.0.0) FUDGE = -1.
CEX = CATL - FUDGE * UCL

CALCULATE THE CATAPULT EXTENSION COMPONENTS ......

DO 50 I=1,3
50 EXT(I) = CEX * CXUV(I)

*------------------------------------------------------------------*
*
DETERMINE THE VARIABLES CALCULATED FROM THE
* EARTH VELOCITIES OF THE AIRPLANE ATTACHMENT
* POINT AND THE SEAT ATTACHMENT POINT
*
******************************************************************************

DETERMINE THE SEAT CATAPULT ATTACHMENT POINT VELOCITY COMPONENTS
IN THE EARTH SYSTEM (USAPE) .......

CALL VELXYZ (USAPE,UST,SAP,WSTIR,DSE)

DETERMINE THE AIRPLANE CATAPULT ATTACHMENT POINT VELOCITY COMPONENTS
IN THE EARTH SYSTEM (UAAPE) .......

CALL VELXYZ (UAAPE,UAP,AAP,WAPIR,DAE)

CALCULATE THE RELATIVE VELOCITY BETWEEN CATAPULT ENDS

DO 60 I=1,3
60 CDV(I) = USAPE(I) - UAAPE(I)

CALCULATE THE CATAPULT EXTENTION RATE (CV)

CALL OUTPRD (CV,CDV,CXUV,3)

CALCULATE EXTENTION VELOCITY VECTOR

DO 70 I=1,3
70 CDV(I) = CV * CXUV(I)

******************************************************************************

* CATAPULT LOGIC *
******************************************************************************

BYPASS IF PRIOR TO CATAPULT IGNITION .......

IF(SW.NE.1.) GO TO 90

COMPUTE THE EXPOSED THERMAL AREA OF THE CATAPULT CHAMBER .......

THA = PC * (TLO + CEX*12.) + PA * 2.

COMPUTE THE FORCE DUE TO THE CATAPULT PRESSURE .......

CALL CAO (CF,EF,EFDOT,IEF,IEL,ELODIT,IEL,WK,WKDOT,IWK,MB,MBDOT,IB,FL,TCF,TIME,CEX,CSK,CI,CI,VI,PA,TF,TVH,CBP,CI,CV,C2,TI,THAB,BX,PT,TYPE,TSO,FSO,TDE)

IF(IFL.EQ.2.) FGN = 1.

FIND THE EARTH SYSTEM COMPONENTS OF THE CATAPULT PRESSURE .......

DO 80 I=1,3
80 FCP(I) = -CF * CXUV(I)

******************************************************************************
C * CATAPULT STRUCTURAL SUPPORT *
C * *****************************************************
C CHECK TO SEE IF THE CATAPULT MUST SUPPORT THE SEAT .......
C IF(CAIL.GT.UCL) GO TO 120
C FORCES DUE TO CATAPULT STRUCTURAL SPRING CONSTANT .......
C 90 DO 160 I=1,3
100 FSS(I) = SK * EXI(I)
C FORCE DUE TO CATAPULT STRUCTURAL DAMPING .......
C DO 110 I=1,3
110 FSD(I) = CK * COV(I)
GO TO 140
C ZERO OUT THE CATAPULT STRUCTURAL FORCES AND MOMENTS WHEN
C THE CATAPULT CAN SUPPORT THE SEAT .......
C 120 DO 130 I=1,3
130 FSO(I) = 0.
135 FSS(I) = 0.
C *****************************************************
C ******** TOTAL CATAPULT FORCES ********
C *****************************************************
C 140 DO 150 I=1,3
150 FC(I) = FCP(I) + FSS(I) + FSO(I)
C ****** FORCES AND MOMENTS ON THE AIRPLANE ******
C *****************************************************
C TRANSFORM THE EARTH SYSTEM FORCE COMPONENTS INTO THE
C AIRPLANE BODY AXIS .......
C CALL MATMPY (FCA,DEA,FC,3,3,1)
C CATAPULT MOMENTS ON THE AIRPLANE .......
C CALL CRSPRD (TCA,AAP,FCA)
C ZERO THE FORCES AND TORQUES ACTING ON THE AIRPLANE IF SSFLG
C IS EQUAL TO ZERO .......
C IF(SSFLG.NE.0) GO TO 160
DO 155 I=1,3
155 FCA(I) = TCA(I) = 0
C ****** FORCES AND MOMENTS ON THE SEAT ******
C *****************************************************
C CATAPULT FORCES ON THE SEAT .......

355
C 160 DO 165 I=1,3
165 FC(I) = -FC(I)
C
C TRANSFORM EARTH SYSTEM FORCE COMPONENTS INTO THE SEAT
C BODY AXIS .......
C
C    CALL MAIMPY (FCS, DES, FC, 3, 3, I)
C
C CATAPULT MOMENTS ON THE SEAT
C
C    CALL CKSPRD (TGS, SAP, FCS)
C
170 CONTINUE
C
C RETURN
END
160 DO 165 I=1,3
165 FC(I) = -FC(I)

C TRANSFORM EARTH SYSTEM FORCE COMPONENTS INTO THE SEAT
C BODY AXIS .......
C
CALL MATMPY (FCS,DES,FC,3,3,1)

C CATAPULT MOMENTS ON THE SEAT
C
CALL CKSPRD (TCS,SAP,FCS)

170 CONTINUE
C
RETURN
END
SUBROUTINE DR (TBF,
   
   FDS,TDS,FDA,TDA,DL1,DFB,SW,
   
   DAP,DBA,XAP,EAP,SRP,EST)

COMMON /CICCAL/ IC.CAL
COMMON /COVRLY/ INST
COMMON /CTIME/ TIME
COMMON /CIO/ IREAD, IWRITE, IDIAG

*************** DART TABLES ***************

ToF - DART BRAKING FORCE TABLE

THE INDEPENDENT VARIABLE IS THE LINE LENGTH (FT).
THE DEPENDENT VARIABLE IS THE BRAKING FORCE (LB).

*************** DART OUTPUTS ***************

FDS(3) - X,Y,Z BODY AXIS FORCE COMPONENTS ON THE SEAT (LB)
TDS(3) - X,Y,Z BODY AXIS MOMENT COMPONENTS ON THE SEAT (FT-LB)
FDA(3) - X,Y,Z BODY AXIS FORCE COMPONENTS ON THE AIRPLANE (FT)
TDA(3) - X,Y,Z BODY AXIS MOMENT COMPONENTS ON THE AIRPLANE (FT-LB)
DLL - DISTANCE BETWEEN THE BRIDLE APEX AND THE AIRPLANE ATTACHMENT POINT (FT)

Ubf - DART BRAKING FORCE (LB)
SW - DART MODE FLAG
0=PRIOR TO DART FORCE
1=DART ON
2=DART OFF

*************** DART INPUTS ***************

DAP(3) - X,Y,Z AIRPLANE BODY AXIS POSITION VECTOR OF THE DART ATTACHMENT POINT (FT)
DBA(3) - X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE DEPLOYED DART BRIDLE APEX (FT)
XAP(3) - X,Y,Z EARTH POSITION VECTOR OF THE AIRPLANE (FT)
EAP(3) - EARTH TO AIRPLANE EULER ANGLES (DEG)
SRP(3) - X,Y,Z EARTH POSITION VECTOR OF THE SEAT REFERENCE POINT (FT)

EST(3) - EARTH TO SEAT EULER ANGLES (DEG)

DIMENSION TBF(5),FDS(3),TDS(3),FDA(3),TDA(3),DAP(3),DBA(3),
   
   XAP(3),EAP(3),SRP(3),EST(3)

DIMENSION DSE(3,3),DSE(3,3),DEA(3,3),DEA(3,3),
   
   DAP(3),DBA(3),EAP(3),DSE(3,3),DF(3),
   
   ESTIR(3),EAPIR(3)

DATA RPD /0.01745329/

*************** INITIALIZATION ***************

IF (ICLAL.NE.1) GO TO 20
SW = 0
DLL = DBF = 0

357
C /************************************************************/
C ZERO OUT THE DART FORCES ......
C
20   DO 30 I=1,3
30   FOS(I) = TOS(I) = FDA(I) = TDA(I) = 0
C BYPASS COMPONENT DURING STEADY STATE OR IF THE DART IS OFF ......
C
1F(INST.EQ.31 OR. SW.EQ.2.) GO TO 100
C
C CONVERT EULER ANGLES FROM DEGREES TO RADIANS
C
DO 40 I=1,3
   ESTIK(I) = EST(I) * RPD
40   EAPIK(I) = EAP(I) * RPD
C
C COMPUTE THE DIRECTION COSINE MATRICES ......
C
CALL DIRKOS (DES,ESTIK)
CALL TRANS (DSE,DES,3,3)
CALL WIRCOS (DEA,EAPIK)
CALL TRANS (DAE,DEA,3,3)
C
EARTH AXIS POSITION OF THE AIRPLANE DART LINE ATTACHMENT
C POINT ......
C
CALL VECXYZ (DAPE,DAP,XAP,DAE,2)
C
EARTH AXIS POSITION OF THE DEPLOYED DART BRIDLE APEX ......
C
CALL VECXYZ (DBAE,DBAE,SRP,DSE,2)
C
C CALCULATE THE DART LINE LENGTH ......
C
DO 50 I=1,3
50   DELTA(I) = DAPE(I) - DBAE(I)
   DLL = SQRT (DELTA(1)**2 + DELTA(2)**2 + DELTA(3)**2)
C
C DETERMINE THE DART BRAKING FORCE ......
C
NIBF = IBF(2)
IF(DLL .LT. TBF(I)) GO TO 100
IF(DLL .LT. TBF(3+NIBF)) GO TO 60
IF(ICCAL.NE.1) SW = 2.
IF(INST.EQ.26) WRITE(6,55) TIME
   25 FORMAT(*X,*DART OFF AT TIME = *,F10.4,* SEC*)
   GO TO 20
60   IF(INST.EQ.26 .AND. SW.EQ.0.) WRITE(6,65) TIME
65   FORMAT(*X,*DART ON AT TIME = *,F10.4,* SEC*)
   IF(ICCAL.NE.1) SW = 1.
   DBF = TOLU1(DLL,1DF(4),1DF(NIBF+4),1,-NIBF)
C
C CALCULATE THE DIRECTION COSINES OF THE DART LINE ......
C
DO 70 I=1,3
70   DC(I) = DELTA(I)/DLL

358
C EARTH COMPONENTS OF THE UART LINE LOAD ON THE SEAT ......
C
DO 80 1=1,3
80  DF(1) = DBF * DC(1)
C
*************** SEAT FORCES AND MOMENTS ****************
C BODY AXIS FORCE COMPONENTS ON THE SEAT ......
C CALL MATMPY (FDS, DES, DF, 3, 3, 1)
C
CALL CRSPRD (TDS, DBA, FDS)
C
*************** AIRPLANE FORCES AND MOMENTS **************
C BODY AXIS FORCE COMPONENTS OF THE AIRPLANE ......
C
DO 90 I=1,3
90  DF(1) = -DF(1)
C CALL MATMPY (FDA, DEA, DF, 3, 3, 1)
C
CALL CRSPRD (TDA, DAP, FDA)
C
100 RETURN
END
SUBROUTINE GP (TMF,
   .FL,FTM,FST,FPP,TPP,TIN,TLS,FSO,TSO,FPO,
   .TPU,TRM,
   .SW,UV,XMO,XYZ,EA,XR,ER,ED,SRP,UST,EST,WST,
   .XPP,UPP,EPPIWPP)

************** GP TABLES **************

TMF - PARACHUTE MORTAR FORCE TABLE

THE INDEPENDENT VARIABLE IS TIME (SEC)
THE DEPENDENT VARIABLE IS THE MORTAR FORCE (LB)

************** GP OUTPUTS **************

FL - MORTAR MODE FLAG
   0 = PRIOR TO INITIATION
   1 = INITIATION UP TO LAUNCH
   2 = PARACHUTE LAUNCH
   3 = FORCES AND TORQUES OFF

FMT - PARACHUTE MORTAR FORCE MAGNITUDE (LB)

FST(3) - X,Y,Z SEAT BODY AXIS FORCE VECTOR ACTING ON THE SEAT (LB)

TST(3) - X,Y,Z SEAT BODY AXIS TORQUE VECTOR ACTING ON THE SEAT (FT/LB)

FPP(3) - X,Y,Z EARTH SYSTEM FORCE VECTOR ACTING ON THE PARACHUTE PACK (LB)

TPP(3) - X,Y,Z PARACHUTE PACK BODY AXIS TORQUE VECTOR ACTING ON THE PARACHUTE PACK (LB)

TIN - PARACHUTE MORTAR INITIATION TIME (SEC)

TLA - PARACHUTE LAUNCH TIME (SEC)

FSO(3) - X,Y,Z SEAT BODY AXIS FORCE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF (LB)

TSO(3) - X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF (FT-LB)

FPO(3) - X,Y,Z PARACHUTE PACK BODY AXIS FORCE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF (FT-LB)

TPO(3) - X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS EXERTED ON THE PACK AT STRIPOFF (FT-LB)

TRM(3) - X,Y,Z SEAT EARTH VELOCITY COMPONENTS TO PASS TO THE PARACHUTE COMPONENT DURING TRIM (FT/SEC)

************** GP INPUTS **************

SW - FLAG TO INITIATE THE MORTAR (1 = ON)

UV(3) - X,Y,Z SEAT BODY AXIS MORTAR FORCE UNIT VECTOR

XMO(3) - X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE DEPLOYMENT IMPULSE MOMENT ARM (FT)

XYZ(3) - X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE PACK (FT)

EA(3) - SEAT TO PARACHUTE PACK EULER ANGLES (DEG)

XR - PARACHUTE SHELF LINEAR SPRING CONSTANT (LB/FT)

XO - PARACHUTE SHELF LINEAR DAMPING CONSTANT (LB/FT/SEC)

ER(3) - X,Y,Z PARACHUTE SHELF ANGULAR SPRING CONSTANTS (FT-LB/DEG)

ED(3) - X,Y,Z PARACHUTE SHELF ANGULAR DAMPING CONSTANTS

360
DIMENSIONS OF CALLING ARGUMENTS ...... 

COMMON /CTIME/ TIME
COMMON /CICAL/ ICCAL
COMMON /COVRLY/ INST
COMMON /CSSFLG/ SSFLG
COMMON /CIO/ READ, WRITE, IUG

DATA KPU, UPR / .01745329, 57.29578 / 

*************
**** INITIALIZATION ****
*************

IF (ICCAL .NE. 1) GO TO 10

DO 2 I = 1, 3
  EAIR(I) = EA(I) * KPO
  CALL UKCOS (UCA, EAIR)
  CALL TRANS (DCET, UCA, 3, 3)
  IF (TUE.EQ.0.99999) TUE = 0
  FL = FMT = TIN = TLA = TIMOR = 0
  DO 5 J = 1, 3
    TRM(J) = FSU(J) = ISO(J) = FP0(J) = TPO(J) = 0
    TYPE = 3HGN
    LEN
    C /---------------------------------------------------------------/
C BYPASS CALCULATIONS IF THE PARACHUTE PACK HAS BEEN
C RELEASED AND THE FORCES AND TURQUES HAVE DECAYED ...... 
C 10 IF(FL.EQ.3.) GO TO 250 
C FACTOR FORCES AND TURQUES TO ZERO AFTER STRIPOFF ...... 
C IF(FL.NE.2.) GO TO 25 
C TOFF = TLA + TUE 
C DELTA = TOFF - TIME 
C FACTOR = DELTA/TUE 
C IF(DELTA.LE.0) FL = 3. 
C IF(FL.EQ.3.) FACTUR = 0 
C DU 20 I=1,3 
C FST(I) = FSQ(I) * FACTUR 
C TST(I) = TSQ(I) * FACTUR 
C FPP(I) = FPQ(I) * FACTUR 
C TPP(I) = TPQ(I) * FACTUR 
C 20 GO TO 250 
C SET THE TMURT AND FMORT VECTORS TO ZERO ...... 
C DU 25 I=1,3 
C TMURT(I) = 0 
C FMORT(I) = 0 
C NM1 = IMF(2) 
C ***** CHANGE FROM DEGREES TO RADIANS ***** 
C DU 35 I=1,3 
C ESTIR(I) = EST(I) * RPD 
C WSTIR(I) = WST(I) * RPD 
C EPPIR(I) = EPP(I) * RPD 
C WPPIR(I) = WPP(I) * RPD 
C ***** CALCULATE THE DIRECTION COSINE MATRICES ***** 
C CALCULATE THE EARTH TO SEAT MATRIX ...... 
C CALL UIRCOS (OES,ESTIR) 
C CALCULATE THE SEAT TO EARTH MATRIX ...... 
C CALL TRANS (OEST,UES,3,3) 
C CALCULATE THE EARTH TO PARACHUTE PACK MATRIX ...... 
C CALL UIRCOS (UDEP,EPPIR) 
C CALCULATE THE PARACHUTE PACK TO EARTH MATRIX ...... 
C CALL TRANS (DEPT,UDEP,3,3) 
C CALCULATE THE SEAT TO PARACHUTE PACK MATRIX ...... 
C CALL MATHPY (DSP,UDEP,DEST,3,3,3) 
C
forces due to linear displacement

------ linear spring forces ------

calculate the parachute pack linear position vector in the seat coordinate system ....

    call veclxyz (xs,xpp,sp,sps,es,1)

determine the linear displacement from the attachment point, and calculate the spring forces in the seat system acting on the seat ....

do 40 i=1,3
    deltax(i) = xs(i) - ay2(i)
    spring(i) = deltax(i) * xr

------ linear damping forces ------

determine the earth velocity of the position the parachute pack occupies in the seat coordinate system ....

    call veclxyz (uxse,ust,xs,wstir,dest)

determine the relative velocity wrt the earth frame ....

do 45 i=1,3
    deltav(i) = upp(i) - uxse(i)

transform this difference into the seat system ....

    call matmpy (rvcl,des,deltav,3,3,1)

compute the damping force acting on the seat ....

    do 50 i=1,3
    damp(i) = rvcl(i) * xd

------ sum the spring and damping forces acting on the seat ------

    do 60 i=1,3
    sum(i) = spring(i) + damp(i)

mortar logic

if(sum.ne.1.) go to 130

if(fl.ne.0.) go to 60
    if(inst.eq.20) write(6,70) type,time
70 format(5x,a8,*) ignition at time = *;fi0.4,* sec*/
    tin = time
    fl = 1.

calculate the mortar force ....
C 40 TIMOR = TIME - TIN
   FMT = TBLU1 (TIMOR, IMF(4), TMF(NMT+4), 1, -NMT)
C C CALCULATE THE SEAT BODY AXIS MORTAR FORCE COMPONENTS
C ACTING ON THE SEAT ......
C DO 90 I=1,3
   90 FMORT(I) = -1. * FMT * UV(I)
C C CALCULATE THE TORQUE ON THE SEAT FROM THE MORTAR ......
C CALL CASPRD (TMORT, XM0, FMORT)
C C DETERMINE THE LINEAR SPRING FORCES ONTO THE MORTAR UNIT VECTOR ......
C CALL DOTPRD (DOT, SPRING, UV, 3)
C IF THE SIGN OF THE DOT PRODUCT IS NEGATIVE, RETAIN THE SHELF FORCE ......
C IF(DOT.LE.0) GO TO 130
C C DETERMINE THE TOTAL LINEAR RESTRAINT FORCE ONTO THE UNITVECTOR ......
C CALL DOTPRD (DOT, FST, UV, 3)
C C DETERMINE THE VECTOR COMPONENTS OF THE PROJECTION OF THE
C RESTRAINT FORCE ONTO THE UNIT VECTOR ......
C DO 100 I=1,3
   100 PROJ(I) = DOT * UV(I)
C C DETERMINE THE FORCE VECTOR NORMAL TO THE UNIT VECTOR ......
C DO 110 I=1,3
   110 FST(I) = FST(I) - PROJ(I)
C ***********************************************
C DETERMINE THE TORQUE ON THE SEAT FROM THE RESTRAINTS ......
C 130 CALL CASPRD (TOKQUE, XS, FST)
C C CALCULATE THE TOTAL FORCE ACTING ON THE SEAT ......
C DO 140 I=1,3
   140 FST(I) = FST(I) + FMORT(I)
C C CALCULATE THE FORCES ACTING ON THE PARACHUTE PACK IN THE
C EARTH SYSTEM ......
C CALL MATMPLY (FPP, DEST, FST, 3, 3, 1)
DO 150 I=1,3
   150 FPP(I) = -FPP(I)
C ***********************************************
C **** TORQUE DUE TO ANGULAR DISPLACEMENT ****
C ***********************************************
--- ANGULAR SPRING FORCES ---

CALL COSDIR (ANG, DSP)

DETERMINE THE ANGULAR DISPLACEMENT FROM THE ATTACHMENT ANGLE,
AND CALCULATE THE SPRING COMPONENTS ACTING ON THE SEAT IN THE
ATTACHMENT AXIS SYSTEM

DO 160 I=1,3
DELTAX(I) = ANG(I)*DPR - EA(I)
160 SPRING(I) = DELTAX(I) * ER(I)

--- ANGULAR DAMPING FORCES ---

DETERMINE THE ANGULAR VELOCITY OF THE PARACHUTE PACK IN THE
ATTACHMENT AXIS SYSTEM

CALL MATMPLY (WSTE, DEST, MST, 3, 3, 1)
CALL MATMPLY (WPPE, DEPT, WPP, 3, 3, 1)
DO 170 I=1,3
170 DELTAV(I) = WPPE(I) - WSTE(I)
CALL MATMPLY (TME, DES, DELTAV, 3, 3, 1)
CALL MATMPLY (RVEL, DLL, TEM, 3, 3, 1)

CALCULATE THE ANGULAR DAMPING TORQUE, AND SUM WITH THE ANGULAR
SPRING TORQUE

DO 180 I=1,3
DAMP(I) = RVEL(I) * ED(I)
180 TEMP(I) = SPRING(I) + DAMP(I)

MOVE THE RESTRAINT TORQUES INTO THE SEAT SYSTEM

CALL MATMPLY (TST, DCEAT, TEMP, 3, 3, 1)

CALCULATE THE BODY AXIS TORQUE CONSTANTS ACTING ON THE
PARACHUTE PACK

CALL MATMPLY (TPP, DSP, TST, 3, 3, 1)
DO 190 I=1,3
190 TPP(I) = -TPP(I)

CALCULATE THE TOTAL MOMENT ON THE SEAT

DO 200 I=1,3
200 TST(I) = TST(I) + TMRT(I) + TORQUE(I)

IF THE MORTAR IS AT STRIPOFF

IF (TIMOR.LT.TMF(NMT+3)) GO TO 225
TLA = TIME
FL = 2.
IF (TLT.EQ.0) FL = 3.
IF (L.EQ.3.) GO TO 215
DO 210 I=1,3

365
FS0(1) = FST(1)
TS0(1) = TST(1)
FPO(1) = FPP(1)
=10  TPO(1) = TPP(1)
C
215 IF(INST.EQ.26) WRITE(6,220) TYPE,TIME
220 FORMAT(’/5X,A8,* STRIPUFF AT TIME = ’,F10.4,* SEC*/)
C
4 ZERO THE FORCES AND TORQUES ACTING ON THE SEAT IF SSFLG
C IS EQUAL TO ZERO .......
C
225 IF(SSFLG.NE.0) GO TO 240
230 FST(1) = TST(1) = 0
C
SEND DATA TO PARACHUTE PACK BODY TO ALLOW IT TO COMPUTE THE
SEAT EARTH VELOCITY DURING TRIM .......
C
240 IF (INST.NE.31) GO TO 250
CALL MATMPY (TRM,DEST,UST,3,3,1)
C
250 RETURN
END
SUBROUTINE LI (ICW, 
- EC, ECD, IEC, TF, TFD, ITF, 
- FLA, SWI, FDO, TDO, FLP, FAP, VAP, FLL, ELM, ELC, DEM, 
- RN, DLS, CON, TCW, UVL, RLO, VL, VCG, PCG, CWT, TPE, PV), 
- TLS, VLS, 
- OFF, BLA, AP1, AP2, AP3, AP4, FTR, FSC, ULL, ULS, GTR, 
- TYP, FL, XDO, UDO, E00, WDO, XPP, UPP, EPP, XPC, UPC)

DESIGNED BY C.L. WEST
LAST MODIFIED - DECEMBER 6, 1980

THE EASIEST PARCHUTE LINE MODEL

*************** LI TABLES ***************

TCW - STRETCHED PARACHUTE CANOPY WEIGHT TABLE

THE INDEPENDENT VARIABLE IS THE STRETCHED LENGTH (FT)
THE DEPENDENT VARIABLE IS THE STRETCHED WEIGHT (LB)

*************** LI OUTPUTS ***************

CREEP STRAIN IN PARACHUTE LINES

EC - CREEP STRAIN IN PARACHUTE LINES (IN/IN)
ELU - CREEP STRAIN RATE (IN/IN/SEC)
IEL - INTEGRATION CONTROL

TIME DURATION OF PARACHUTE LINE LOAD (CHARACTERISTIC FUNCTION)

TF - TIME PARACHUTE LINES EXPERIENCE A NON-ZERO LOAD (SEC)
TFD - RATE EQUALS ONE WHEN LINES ARE UNDER LOAD, OTHERWISE ZERO
ITF - INTEGRATION CONTROL

FLA - PARACHUTE PHASE
0 = PRIOR TO INITIATION
1 = INITIATION
2 = LAUNCH
3 = MCREAR OFF
4 = LINES STRETCH
5 = LINES SEVERED

SWI - FLAG SET WHEN THE PARACHUTE IS BEHIND THE BRIDLE APEX
FDO(3) - X,Y,Z DECELERATED OBJECT BODY AXIS FORCE COMPONENTS ACTING
ON THE DECELERATED OBJECT (LB)
TDO(3) - X,Y,Z DECELERATED OBJECT BODY AXIS TORQUE COMPONENTS ACTING
ON THE DECELERATED OBJECT (FT-LB)
FLP(3) - X,Y,Z FORCE COMPONENTS ACTING ON THE PARACHUTE (LB)
(AXIS FOR PACK - EARTH SYSTEM FOR CANOPY)
FAP(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE
FORCE APPLICATION POINT (FT)
VAP(3) - X,Y,Z EARTH SYSTEM VELOCITY COMPONENTS OF THE FORCE
APPLICATION POINT (FT/SEC)
FLL - LINE LOAD (LB)
ELM - MAXIMUM STRAIN EXPERIENCED BY THE PARACHUTE LINE
DURING IT'S LOADING HISTORY (IN/IN)
ELC - MAXIMUM STRAIN EXPERIENCED BY THE PARACHUTE LINE
DURING THE CURRENT LOADING CYCLE ONLY (IN/IN)

367
DEM - MAXIMUM POSITIVE STRAIN RATE EXPERIENCED BY THE PARACHUTE LINE DURING ITS LOADING HISTORY (I/SEC)

RMN - MAXIMUM NEGATIVE STRAIN RATE EXPERIENCED BY THE PARACHUTE LINE DURING THE CURRENT UNLOADING CYCLE ONLY (I/SEC)

DIS - THE DISTANCE FROM THE ORIGIN OF THE DECELERATED OBJECT TO THE BRIDLE APEX (FT)

CUN(4) - COEFFICIENTS IN THE EQUATION FOR THE PLANE FORMED BY THE BRIDLE ATTACHMENT POINTS

TCG(20) - PARACHUTE CENTER OF GRAVITY LOCATION ARRAY (FT)

UVL(3) - PARACHUTE LINE UNIT VECTOR

RL - PARACHUTE LINE LENGTH (FT)

RLU - UNLOADED PARACHUTE LINE LENGTH (FT)

VL - RATE OF CHANGE OF LINE LENGTH (FT/SEC)

VCG - VELOCITY OF THE CANOPY CENTER OF GRAVITY ALONG THE PARACHUTE LINES (FT/SEC)

PCG - STRETCHED CANOPY CENTER OF GRAVITY MEASURED ALONG THE PARACHUTE LINE FROM THE PARACHUTE PACK (FT)

CWT - WEIGHT OF CANOPY PULLED FROM THE PARACHUTE PACK (LB)

TPE - TYPE OF PARACHUTE (1=DRAG 2=RECOVERY)

PVL - PREVIOUS TIMESTEP LINE VELOCITY (FT/SEC)

TLS - TIME AT LINESSTRETCH (SEC)

VLS - RATE OF CHANGE OF LINE LENGTH AT LINESSTRETCH (FT/SEC)

*************** LI INPUTS ******************

OFF - FLAG TO SEVER LINES

U = LINES ATTACHED

1 = LINES SEVERED

BLI - NUMBER OF BRIDLE LINES

APX(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE BRIDLE APEX (FT)

AP1(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE FIRST BRIDLE LINE ATTACHMENT POINT (FT)

AP2(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE SECOND BRIDLE LINE ATTACHMENT POINT (FT)

AP3(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE THIRD BRIDLE LINE ATTACHMENT POINT (FT)

AP4(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE FOURTH BRIDLE LINE ATTACHMENT POINT (FT)

FIR - PARACHUTE LINE MULTIPLICATION FACTOR

FSU - CANOPY STRIPOUT FORCE (LB)

ULL - PARACHUTE SUSPENSION LINE ULTIMATE LOAD (LB)

ULS - PARACHUTE SUSPENSION LINE ULTIMATE STRAIN (IN/IN)

GUR - NUMBER OF PARACHUTE GORES

TYP - TYPE OF PARACHUTE (1=DRAG 2=RECOVERY)

FL - MORTAR MODE FLAG

0 = PRIOR TO INITIATION

1 = INITIATION UP TO LAUNCH

2 = PARACHUTE LAUNCH

3 = MORTAR OFF

dODY (FT)

U0U(3) - X,Y,Z DECELERATED OBJECT BODY AXIS VELOCITY VECTOR OF THE DECELERATED BODY (FT/SEC)

W0U(3) - X,Y,Z DECELERATED OBJECT BODY AXIS ANGULAR VELOCITY COMPONENTS OF THE DECELERATED OBJECT (DEG/SEC)

XPP(3) - X,Y,Z EARTH FRAME POSITION VECTOR OF THE PARACHUTE PACK (FT)

368
<table>
<thead>
<tr>
<th>Analysis of Ejection Seat Stability Using EASY Program. Volume...etc. (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEP 80 C L West, B R Wohlen, R F Yurczyk.</td>
</tr>
<tr>
<td>F36415-79-C-3007.</td>
</tr>
<tr>
<td>UNCLASSIFIED.</td>
</tr>
</tbody>
</table>
UPP(3) - X, Y, Z PARACHUTE PACK EARTH SYSTEM VELOCITY VECTOR (FT/SEC)
EPP(3) - EARTH TO PARACHUTE PACK EULER ANGLES (DEG)
XPC(3) - X, Y, Z EARTH SYSTEM POSITION VECTOR OF THE PARACHUTE CANOPY (FT)
UPC(3) - X, Y, Z EARTH SYSTEM VECTOR VECTOR OF THE PARACHUTE CANOPY (FT)

DIMENSION OF CALLING ARGUMENTS ......

DIMENSION TCL(5), TDW(3), TUQ(3), FLP(3), FAP(3), VAP(3), CON(4),
    TCG(2G), UWL(3), APX(3), AP2(3), AP3(3), AP4(3),
    XDO(3), UGO(3), EDO(3), WDO(3), XPP(3), UPP(3), EPP(3),
    XPC(3), UPC(3)

INTERNAL DIMENSIONS ......

DIMENSION WDQ1R(3), EDQ1R(3), DEQ(3,3), DOE(3,3), XPPDO(3),
    UVP(3), FPSV(3), FAVQ(3), UPPPOS(3),
    UPPREL(3), UPPDO(3), EPP1R(3), FDOT(3),
    XPCS(3), XPCDO(3)

COMMON /CTIME/TIME
COMMON /CCICAL/CCICAL
COMMON /CVOVLY/INST
COMMON /CPFLAG/UMITINC
COMMON /C10/IRREAD, IWRITE, IDIAG

DATA RPD / .01745329 /
DATA GRAV /32.174/

**** INITIALIZATION *****

***************

IF(ICALCAL.NE.1) GO TO 70

MISC INITIALIZATION ......

TPE = TYP
FLA = SW1 = FLL = ELM = ELC = DEM = RMN = RL = RLO = 0
VCG = PCG = CWT = PVL = TLS = VLS = 0
IF(OFF .EQ. 0.99999) OFF = 0
DO 10 I=1,3
   IF(APX(I) .EQ. 0.99999) APX(I) = 0
   IF(AP2(I) .EQ. 0.99999) AP2(I) = 0
   IF(AP3(I) .EQ. 0.99999) AP3(I) = 0
   IF(AP4(I) .EQ. 0.99999) AP4(I) = 0
10 CONTINUE

CALCULATE THE DISTANCE FROM THE ORIGIN OF THE DECELERATED OBJECT
TO THE BRIDLE APiX ......

DIS = SQRT ( APX(1)**2 + APX(2)**2 + APX(3)**2 )

CALCULATE THE CONSTANTS FOR THE EQUATION DEFINING THE
BRIDLE ATTACHMENT PLANE ......

369
CON(1) = DET3(API(1),AP1(2),API(3),1.,AP2(2),AP2(3),1.,AP3(2),AP3(3))
CON(2) = DET3(API(1),1.,AP1(3),AP2(1),AP2(2),AP3(1),1.,AP3(3))
CON(3) = DET3(API(1),AP1(2),1.,AP2(1),AP2(2),1.,AP3(1),AP3(2))
CON(4) = DET3(API(1),API(1),API(3),AP2(2),AP2(1),AP2(3),AP3(2),AP3(1),AP3(3))

C COMPUTE THE PARACHUTE CANOPY CG TABLE

DO 15 I=1,20
  TCG(I) = 0
  WRITE(6,20)
  FORMAT(*5x,*--- STRETCHED CANOPY CG TABLE FOR COMPONENT*,
        * LI -----,//13X,*LINE*,12X,*CG*/)
  NA = TCW(2)
  TOTALM = 0
  TOTALW = TCW(2*NA+3)
  DO 30 I=2,NA
     TOTALM = TOTALM + ((TCW(3+1) - TCW(2+1))/2.*TCW(2+1))
     TOTALW = TOTALW + (TCW(3+1) - TCW(2+1))/2.*TCW(2+1)
  30 CONTINUE
  TCG(I) = (TOTALM + (TOTALW - TCW(NA+3+1))*TCW(3+1))/TOTALW
  WRITE(6,40) (TCW(I+3), TCG(I), I=1,NA)
  FORMAT(/5X,F5.2,2,1UX,F7.4)

DO 60 I=1,3
  FDO(I) = 0
  TUO(I) = 0
  FLP(I) = 0
  VAP(I) = 0
  UVL(I) = 0

60 CONTINUE

C --- BYPASS THE COMPONENT IF FL DOES NOT EQUAL 2 OR FLA EQUALS 4 ---

IF(FL.EQ.1.) FLA = 1.
IF(FL.LE.1. .OR. FLA.EQ.5.) GO TO 330

IF(TYP.EQ.1.) TYPE = 4HDRAZ
IF(TYP.EQ.2.) TYPE = 8KRERRECOVERY

C --- IF THE LINES HAVE BEEN SEVERED ---

IF(TOFF.NE.1.) GO TO 100
  FLA = 5.
  FLL = 0.
  DO 80 I=1,3
     FDO(I) = TUO(I) = FLP(I) = UVL(I) = VAP(I) = FAP(I) = 0
     IF(INST.EQ.6) WRITE(6,90) TYPE,TIME
  80 CONTINUE

FORMAT(/5X,A8,* CHUTE LINES SEVERED AT TIME = *,F10.4,* SEC*/
GO TO 330

C --- CHANGE FROM DEGREES TO RADIANS ---

90 FORMAT(/5X,A8,* CHUTE LINES SEVERED AT TIME = *,F10.4,* SEC*/
GO TO 330

C --- CHANGE FROM DEGREES TO RADIANS ---

100 DO 110 I=1,3
     WDOR(I) = WDO(I) * RPD
     EDIR(I) = EDO(I) * RPD
  110 CONTINUE

370
----- CALCULATE DEO ----- 
CALL DIRCOS (DEO,EDUIK) 
IF (FLA.EQ.4.) GO TO 280 
FLA = 2. 

*****
** PRIOR TO LINESTRETCH **
*****

IF (FLA.EQ.3.) FLA = 3. 

----- IF THE CHUTE IS INSIDE THE BRIDLE ----- 
IF (BLI.EQ.1.) GO TO 175 
IF (SW1.EQ.1.) GO TO 150 
CALL VECXYZ (XPPDO,XPP,XDD,DEJ,1) 
IF (3RT(XPPDO(I)**2+XPPDO(2)**2+XPPDO(3)**2) .GE. DIS+1.1) GO TO 140 

CALL CALCULATE THE EARTH SYSTEM VELOCITY OF THE PARACHUTE PACK 
POSITION IN THE DECELERATED OBJECT COORDINATE SYSTEM 

CALL TRANS (DOE,DEO,3,3) 
CALL VECXYZ (UPPPOS,UDD,XPPDO,WDOIR,DOE) 

DO 120 I=1,3 
120 UPPREL(I) = UPPPOS(I) - UPP(I) 

DETERMINE THE RELATIVE VELOCITY OF THE PARACHUTE PACK IN THE 
DECELERATED OBJECT SYSTEM 

CALL MATMPY (UPPDO,UEG,UPPREL,3,3,1) 

CALL CALCULATE THE UNIT VECTOR OF UPPDO 
RESULT = SQRT(UPPDO(1)**2+UPPDO(2)**2+UPPDO(3)**2) 
DO 130 I=1,3 
130 UVV(I) = UPPDO(I)/RESULT 

APPROXIMATE THE FORCE APPLICATION POINT FROM THE VELOCITY 
VECTOR 

CALL LIBRUL (FAP, 
* APX,AP1,AP2,AP3,AP4,CON,BLI,UVV,XPPDO) 
GO TO 180 

SW1 = 1. 

----- CALCULATE THE FORCE APPLICATION POINT ----- 

DETERMINE THE UNIT VECTOR FROM THE PARACHUTE PACK TO THE BRIDLE 
APEX IN THE DECELERATED OBJECT COORDINATE SYSTEM 

371
CALL VECKYZ (XPPDG, XPP, XDU, XED, 1)

UU 160 I = 1, 3
160 UVL(I) = APX(I) - XPPDG(I)

RESULT = SQRT(UVL(1)**2 + UVL(2)**2 + UVL(3)**2)

DU 170 I = 1, 3
170 UVL(I) = UVL(I)/RESULT

175 CALL LIBKIDL (FAP, 
               APX, AP1, AP2, AP3, AP4, CON, 6LI, UVL, XPPDG)

---- CALCULATE THE LINE VARIABLES ----

180 CALL LLINE (RL, UVL, VAP, 
               FAP, XDU, XED, DOIR, DOIR, XPP, UPP, XED)

---- DETERMINE THE CANOPY CG POSITION AND WEIGHT ----

NA = TCW(2)
PCG = TBLUL (RL, TCW(4), TCW(1), 1, -NA)
CGT = TBLUL (RL, TCW(4), TCW(NA+4), 1, -NA)

---- CHECK FOR LINESTRETCH ----

IF(RL .GE. TCW(NA+3)) GO TO 205

---- CALCULATE THE CANOPY STRIPWUT FORCE ----

UU 190 I = 1, 3
190 FSIP(I) = FSO * (-UVL(I))

---- CALCULATE THE FORCE ACTING ON THE DECELERATED OBJECT RESULTING ----

FROM PULLING THE PARACHUTE FROM THE PACK

DU 200 I = 2, NA
200 IF(RL .LT. TCW(I+3)) GO TO 210

210 WMUL = (TCW(NA+I+3) - TCW(NA+I+4))/TCW(I+3) - TCW(I+2)

DMUL = WMUL/GRAv

MDOT = DMUL * VL

DU 220 I = 1, 3
220 FADO(I) = MDOT * (-UVL(I))

---- SUM THE FORCES ACTING ON THE DECELERATED OBJECT ----

DU 230 I = 1, 3
230 FDOT(I) = FSIP(I) + FADO(I)

CALL MAIMPY (FDG, FDU, FDOT, 3, 3, 1)

---- CALCULATE THE TURBO ACTING ON THE DECELERATED OBJECT ----

CALL CRSPRD (7OOFAPFDU)

SUM THE FORCES ACTING ON THE PARACHUTE PACK ----
C
DO 240 I = 1, 3
240 FLP(I) = -FSTP(I)
C
C ----- CALCULATE THE CANOPY CG VELOCITY ALONG THE PARACHUTE ----- 
C  LINES WITH RESPECT TO THE FORCE APPLICATION POINT 
C
DO 250 I = 2, NA
250 IF(RL.LT.TCW(i+1)) GO TO 260
C
260 DLGD1 = (TCG(I)-TCG(I-1))/(TCW(I+1)-TCW(I+2))
   VCG = VL - VL * DLGD1
   GO TO 530
C
C ****** AT LINESSTRETCH ******
C
265 FLA = 4.
   TLS = TIME
   VCG = 0
   DO 270 I = 1, 3
   TOD(I) = FOD(I) = FLP(I) = 0
270 CONTINUE
C
C CALCULATE THE UNLOADED LINE LENGTH ...... 
C
   CALL VECXYZ(APC0,APC,XDU,DEQ,1)
   RLO = SQRT((FAP(I)-XPCS(I))**2 + (FAP(I)-XPCS(I))**2 + 
            (FAP(I)-XPCS(I))**2)
   RL = RLO
C
C WRITE THE LINESSTRETCH MESSAGE .... 
C
IF(INST.EQ.26) WRITE (6,275) TYPE,TIME
275 FORMAT(*5X, A*, CHUTE LINESSTRETCH AT TIME = *,F10.4, SEC*)
C
C ************************************
C **
C ** AFTER LINESSTRETCH **
C **
C ************************************
C
C -- CALCULATE THE FORCE APPLICATION POINT --
C C
C "DTERMINE THE UNIT VECTOR FROM THE PARACHUTE CANOPY TO THE BRIDLE 
C APEX IN THE DECELERATED OBJECT COORODINATE SYSTEM ...... 
C
280 IF(BLI.EQ.0.) GO TO 365
   CALL VECXYZ(APC0,APC,XDU,DEU,1)
C
   DO 290 I = 1, 3
   290 UVL(I) = APX(I) - XPCDO(I)
C
RESULT = SQRT(UVL(1)**2+UVL(2)**2+UVL(3)**2)
C
   DO 300 I = 1, 3
   300 UVL(I) = UVL(I)/RESULT
C
373
CALL LIBRIUL (FAP,
- AP1,AP2,AP3,AP4,CON,BLI,UVL,XPCDO)

-- CALCULATE THE LINE VARIABLES --

CALL LILINE (RL,UVL,VL,VAP,
- FAP,ALU,UVL,EDU,MDL,XPC,UPC,DEO)

IF(VLS.EQ.0) VLS = VL

-- CALCULATE THE PARACHUTE LINE LOAD --

LOGIC TO DETERMINE THE LINE ACCELERATION.....

AL = 0
IF(INST.EQ.20) AL = VL - PVL
IF(INST.EQ.1 .AND. INST.EQ.20) PVL = VL

TALS = TIME - T1.
IF(TALS.LT.0.) TALS = 0.
CALL LILOAD (FLL,FTK,EC,ECO,IEC,TF,TDF,ITF,
- TALS,AL,VL,RL,RLU,FOR,UULS,US,TYPE,
- ELM,LC,D,DEM,EMN)

-- CALCULATE THE FORCES AND TORQUES ACTING ON THE OBJECT --

DO 310 I=1,3
310 FDOT(I) = FLL * (-UVL(I))
CALL MAIMPY (FDO,DIO,FOU,3,3,1)
CALL LRSPKD (TDO,FAP,FOU)

-- CALCULATE THE FORCES ACTING ON THE PARACHUTE CANOPY --

DO 320 I=1,3
320 FLP(I) = -FDOT(I)

RETURN

END
SUBROUTINE LIBRIDL (FAP, APX, AP1, AP2, AP3, AP4, CON, BLI, UV, XPDD)

C COMMON /CIO/ READ, WRITE, IDIAG

**** LIBRIDL OUTPUTS ****

C FAP(I) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE FORCE APPLICATION POINT (FT)

**** LIBRIDL INPUTS ****

C APX(I) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE BRIDLE APEX (FT)
C AP1(I) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE FIRST BRIDLE ATTACHMENT POINT (FT)
C AP2(I) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE SECOND BRIDLE ATTACHMENT POINT (FT)
C AP3(I) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE THIRD BRIDLE ATTACHMENT POINT (FT)
C AP4(I) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE FOURTH BRIDLE ATTACHMENT POINT (FT)
C CON(4) - CONSTANT IN THE EQUATION FOR A PLANE
C BLI - NUMBER OF BRIDLE LINES
C UV(I) - UNIT VECTOR FROM THE PARACHUTE PACK TO THE BRIDLE APEX IN THE DECELERATED OBJECT COORDINATE SYSTEM
C APUD(I) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE (FT)


C DO 10 (10, 30, 40, 50), BLI

10 DO 20 I = 1, 3
20 FAP(I) = AP1(I)
   GO TO 60
30 CALL BRIDL2 (FAP, APX, XPDD, AP1, AP2)
   GO TO 60
40 CALL LINEPL (XI, CON, APX, UV)
   CALL BRIDL3 (FAP, APX, UV, XPDD, AP1, AP2, AP3, XI)
   GO TO 60
50 CALL LINEPL (XI, CON, APX, UV)
   CALL BRIDL4 (FAP, APX, UV, XPDD, AP1, AP2, AP3, AP4, XI)
60 RETURN
END
SUBROUTINE LILINE (RL,UVL,VL,VAP, FAP,DDO,DOO,DOE,XPC,UPC,DEO)

***** LILINE OUTPUTS *****

RL - DISTANCE FROM THE FORCE ATTACHMENT POINT TO THE PARACHUTE CENTER OF GRAVITY (FT)
UVL(3) - PARACHUTE LINE UNIT VECTOR
VL - RATE OF CHANGE OF THE PARACHUTE LINE LENGTH (FT/SEC)
VAP(3) - X,Y,Z EARTH SYSTEM VELOCITY VECTOR OF THE FORCE APPLICATION POINT (FT/SEC)

***** LILINE INPUTS *****

FAP(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE FORCE APPLICATION POINT (FT)
XDO(3) - X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE DECELERATED OBJECT CENTER OF GRAVITY (FT)
UDO(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR VELOCITY VECTOR OF THE DECELERATED OBJECT (FT/SEC)
EDO(3) - EARTH TO DECELERATED OBJECT EULER ANGLES (RAD)
WDO(3) - X,Y,Z DECELERATED OBJECT BODY AXIS ANGULAR VELOCITY VECTOR OF THE DECELERATED OBJECT (KAD/SEC)
XPC(3) - X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE (FT)
UPC(3) - X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE (FT/SEC)
DEO(3,3) - EARTH TO DECELERATED OBJECT DIRECTION COSINE MATRIX

** DIMENSION UVL(3),VAP(3),FAP(3),XDO(3),UDO(3),EDO(3),WDO(3),
XPC(3),UPC(3),DEO(3,3),DOE(3,3),FAPE(3),DETA(3) **

************** CALCULATE THE LINE LENGTH VARIABLES **************

LOCATE THE FORCE APPLICATION POINT IN THE EARTH SYSTEM ..... CALL TRANS (DDO,DEO,3,3)
CALL VELXY (FAPE,FAP,DDO,DEO,2)

COMPUTE THE RESULTANTS AND DIRECTION COSINES

DO 10 I=1,3
10 DELTA(I) = FAP(I) - XPC(I)

RL = SQRT(DELTA(1)**2+DELTA(2)**2+DELTA(3)**2)

CALCULATE THE LINE UNIT VECTOR ......

DO 20 I=1,3
20 UVL(I) = DELTA(I)/RL

************** CALCULATE THE LINE VELOCITY VARIABLES **************

DETERMINE THE EARTH SYSTEM VELOCITY OF THE FAP ......

CALL VELXY (VAP,UDO,FAP,DDO,DOE)

CALCULATE THE EARTH VELOCITY DIFFERENCE ......
DO 30 I=1,3
   DELTA(I) = VAP(I) - UPC(I)
C   PROJECT THE DIFFERENCE ON THE PARACHUTE LINE ......
C   CALL DUTPRD (VL,DELTA,UVL,3)
C   RETURN
END
SUBROUTINE LLOAD (FTT, FCTR, EC, ECUOT, IEC, TF, TFD, ITF, IALS, AX, VX, X, LO, GORES, ULTLD, ULTS1, TYPE, ELM, ELM1, DELOMAX, RMAXD2)

**** LLOAD OUTPUTS *****

FTT - TENSILE LOAD (Lb)
FCTR - ULTIMATE STRENGTH MULTIPLICATION FACTOR
EC - CREEP STRAIN IN TENSILE MEMER (IN/IN)
ECUOT - CREEP STRAIN RATE (1/SEC)
IEC - INTEGRATION CONTROL FLAG
TF - TIME DURATION OF PARACHUTE LINE LOAD (SEC)
TFD - TIME DURATION RATE (EQUALS ONE UNDER LOAD)
ITF - INTEGRATION CONTROL FLAG

**** LLOAD INPUTS *****

IALS - TIME AFTER LINESSTRETCH (SEC)
AX - RATE OF CHANGE OF VX (FT/SEC/SEC)
VX - RATE OF CHANGE OF THE LINE LENGTH (FT/SEC)
LO - ORIGINAL UNSTRESSED LENGTH OF THE PARACHUTE LINES (FT)
GORES - NUMBER OF PARACHUTE GORES
ULTLD - ULTIMATE STRENGTH OF A PARACHUTE SUSPENSION LINE (LB)
ULTSI - ULTIMATE STRAIN OF A PARACHUTE SUSPENSION LINE (IN/IN)
TYPE - ALPHANUMERIC FOR PARACHUTE TYPE

***** ARGUMENTS INCLUDED TO SAVE VALUES *****

ELM - MAXIMUM STRAIN EXPERIENCED BY THE TENSILE MEMBER (IN/IN)
ELM1 - MAXIMUM STRAIN EXPERIENCED DURING THE CURRENT LOADING CYCLE (IN/IN)
DELOMAX - THE MAXIMUM POSITIVE STRAIN RATE EXPERIENCED DURING THE CURRENT LOADING CYCLE (1/SEC)
RMAXD2 - THE MAXIMUM NEGATIVE STRAIN RATE EXPERIENCED DURING THE CURRENT UNLOADING CYCLE (1/SEC)

JANUARY 1978 EDITION OF TENSILE LOAD-ELONGATION ANALOG FOR MIL-C-5040E NYLON CORE-SLEEVE CORD . .

REFERENCE - AFFDL-TR-78-169 SIMULATION OF THE DYNAMIC TENSILE CHARACTERISTICS OF NYLON PARACHUTE MATERIALS

AUTHOR - ROBERT E. MCCARTY 513-255-52516
AFFDL/FER W-PAFB, OHIO 45433

DIMENSION EXA(6), EXU(6), EXC(5,3), TC(6), FC(3), CSR(6,3),
   DPA(6), DPO(6), UPC(5,3), ERLC(3), ELRRC(3), RATC(3)

COMMON /GOVRLY/ INSI
COMMON /UTIME/ TIME
COMMON /CIU/ IREAD, IWRITE, IREAD

REAL KRL, KCR, KDP, L, LO

------ CREEP STRAIN RATE DATA FOR TABLE LOOK UP ------
C IC IS AN INDEPENDENT VARIABLE ARRAY FOR TIME (SEC) .......
C DATA TC / 0., .2, .5, .9, 1.8, 20. /
C FC IS AN INDEPENDENT VARIABLE ARRAY FOR TENSILE LOAD (LB) ......
C DATA FC / 0., 80., 1600. /
C CSR IS A DEPENDENT ARRAY FOR CREEP STRAIN RATE (1/SEC) .......
C DATA CSR / 0., 0., 0., 0., 0., 0.,
   .06779, .03510, .01876, .01010, .001919, .001919, .001919, .001919, .001919, .001919 /
C ------ MATERIAL UNLOADING CURVE FIT PARAMETERS ------
C DPA IS AN ARRAY OF ABSCISSAE FOR THE SIX FIXED KNOTS
C IN A CUBIC SPLINE CURVE FIT USED TO REPRESENT MATERIAL
C UNLOADING CHARACTERISTICS ......
C DATA DPA / 0.0, 0.072, 0.272, 0.65339, 0.73397, 1.0 /
C DC (LB) IS AN ARRAY OF ORDINATES FOR THE SIX FIXED KNOTS .......
C DATA DC / 0.67719, 15.146, 36.968, 17.945, 11.383, 0.0/
C UPC IS AN ARRAY OF CUBIC SPLINE COEFFICIENTS .......
C DATA UPC / 91.891, 218.90, 2.3375, -80.664, -71.035,
   -2776.2, -1013.6, -69.191, -148.32, 260.65,
   -17556.0, 1574.1, -69.126, 1696.0, -605.77 /
C ------ LOADING CURVE FIT PARAMETERS ------
C EXA (IN/IN) IS AN ARRAY OF ABSCISSAE FOR THE SIX FIXED KNOTS
C IN THE CUBIC SPLINE CURVE FIT USED TO REPRESENT MATERIAL
C LOADING CHARACTERISTICS ......
C DATA EXA / 0.0, .037220, .058852, .178888, .210448, .237515 /
C EXD (LB) IS AN ARRAY OF ORDINATES FOR THE SIX FIXED KNOTS .......
C DATA EXD / 0.0, 10.7213, 33.0481, 156.476, 205.580, 251.117 /
C EAC IS AN ARRAY OF CUBIC SPLINE COEFFICIENTS USED TO REPRESENT
C THE MATERIAL LOADING CHARACTERISTICS .......
C DATA EAC / 122.991, 756.471, 113.19, 1216.87, 2107.30,
   -3718.97, 20735.9, -3315.61, 4012.75, 24201.0,
   -18974.0, -370731.0, 20350.3, 213220.0, -1484210. /
C ------ PLASTIC STRAIN CHARACTERISTIC ------
C DATA ELRLC / -0.558, -2178, 3.5989 /
C DATA ELRLC / -0.558, -2178, 3.5989 /
C ------ DAMPING STRAIN DEPENDENCE DATA ------
DATA KATG / -2.7208, 122.01, -272.36/

----- MISC. DATA ----- 

DATA KRL,KCR,KUP,VSDFM / 3*1.0, 0.034 /

*****************************************************
****** ELONGATION ******
*****************************************************

EL (IN/IN) IS STRAIN BASED ON ORIGINAL UNSTRESSED LENGTH ......

EL = FSW (X/LO-1., 0., 0., X/LO-1.)
EL = EL * .23751/ULTST

ELD (IN/IN) IS THE STRAIN EXCLUDING CREEP STRAIN ......

ELD = FSW (EL-EC, 0., 0., EL-EC)

ELM (IN/IN) IS THE MAXIMUM STRAIN EXPERIENCED DURING THE
LOADING HISTORY ......

ELM = AMAX1(ELD,ELM)

ELM1 (IN/IN) IS THE MAXIMUM STRAIN EXPERIENCED DURING THE
CURRENT LOADING CYCLE ......

ELM1 = FSW (VX, AMAX1(ELD,ELM1), ELO, ELO)

ELRL (IN/IN) IS THE UPPER BOUND FOR RESIDUAL STRESS ......

ELRL = ((ELRLC3)*ELM+ELRLC2)*ELM+ELRLC1)*ELM+.0016

ELRR (IN/IN) IS THE LOWER BOUND FOR RESIDUAL STRAIN ......

cELRR = ((ELRRC3)*ELM+ELRRC2)*ELM+ELRRC1)*ELM+.0018

TS (SEC) IS THE CUMULATIVE TIME FOR WHICH THE MEMBER EXPERIENCED
ZERO LOAD ......

TS = TALS - TF

TSS IS THE RATIO OF TS TO THE VALUE OF RELAXATION TIME FOR
THE MATERIAL ......

TSS = FSW ((TS/.3)-1., TS/.3, 1., 1.)

ELR (IN/IN) IS THE RESIDUAL STRAIN ......

ELR = ELKL - TSS * ABS(ELRL-ELRR)
ELR = RLIM (KRL*ELR, 0., KRL*ELR)

ELUT (IN/IN) IS THE LINEAR TRANSFORM OF STRAIN ......

ELUT = (ELU-ELR)*ELM/(ELM-ELR+.0001)

ELS IS THE NORMALIZED STRAIN ......
ELS = (ELMI-EL0)/(ELMI-ELR+0.0001)
ELS = FSW (ELS-1, ELS, ELS, 1.0)
ELS = FSW (ELS, 0., 0., ELS)

L (ft) IS THE CURRENT UNSTRESSED LENGTH .......
L = LU * (1. + ELR)
EL01 (IN/IN) IS THE LINEAR TRANSFORMATION OF STRAIN ......
EL01 = RLIM(EL0T, 0., ELO)
DELO (1/SEC) IS THE STRAIN RATE BASED ON ORIGINAL UNSTRESSED
LENGTH ......
DELO = VX/L0
DELOMAX (1/SEC) IS THE MAXIMUM POSITIVE STRAIN RATE EXPERIENCED
DURING THE LOADING HISTORY ......
DELOMAX = AMAX1 (DELO, DELOMAX)
RMAXD1 (1/SEC) HAS THE VALUE OF DELO WHEN THE STRAIN RATE IS
NEGATIVE ......
RMAXD1 = FSW (DELO, DELO, 0.00001, 0.00001)
RMAXD2 (1/SEC) IS THE MAXIMUM NEGATIVE STRAIN RATE EXPERIENCED
DURING THE CURRENT UNLOADING CYCLE ......
RMAXD2 = FSW (DELO, AMIN1 (RMAXD1, RMAXD2), 0.00001, 0.00001)
CHECK TO SEE IF PARACHUTE LINES HAVE FAILED ......
DO 10 I = 1, 5
   IF (EXA(I) .LE. ELO. AND. ELO .LT. EXA(I+1)) GO TO 30
10   CONTINUE
   IF (INST .EQ. 26) WRITE(0,20) TYPE,TIME
20   FORMAT (/5X,5A,*, CHUTE LINES FAILED AT TIME = *,F10.4,* SEC*,
               * ===== RUN STOPPED ===== */)
STOP

************************************
SPRING FORCE ******
************************************
F50 (LB) IS THE TENSILE LOAD CALCULATED FROM THE CUBIC SPLINE
FIT ......
30   D = ELO-EXA(1)
   F50 = ((EXC(1,3)*D+EXC(1,2))*D+EXC(1,1))*D+EXO(I)-EXO(1)
   DU 40 I = 1, 5
   IF (EXA(I) .LE. EL0T. AND. EL0T .LT. EXA(I+1)) GO TO 50
40   CONTINUE
F5K (LD) IS THE TENSILE LOAD CALCULATED FROM THE CUBIC SPLINE
C FIT FOR THE MATERIAL REPEATED LOADING CHARACTERISTICS .......

50  D = ELOT-EXA(I)
    FSR = ((EXC(I,3)*D+EXC(I,2))*D+EXC(I,1))*D+EXC(I)-EXO(I)

C FSOL (LB) IS FSL LIMITED TO POSITIVE VALUES .......
    FSOL = RLIM (FSO, 0., FSO)

C FSRL (LB) IS FSR LIMITED TO POSITIVE VALUES .......
    FSRL = RLIM (FSR, 0., FSR)

C FS2 (LB) HAS THE VALUE OF FSOL FOR INITIAL LOADING AND
THE VALUE OF FSRL FOR REPEATED LOADING .......
    FS2 = FSW (ELO-ELM, FSRL, FSOL)

C FS1 IS THE SAME AS FS2, BUT IS ZERO WHEN THE LENGTH IS
LESS THAN THE CURRENT UNSTRESSED LENGTH .......
    FS1 = FSW(ELO-ELR,0.0,0.0,FS2)

C FS (LB) IS THE CURRENT LOAD .......
    FS = FS1

*****************************
***** DAMPING EFFECT *****
*****************************

C RATIO IS A SCALAR QUANTITY USED TO ADJUST THE MAGNITUDE OF
LOAD .......
    RATIO = ((RATC(3)*ELM1+RATC(2))*ELM1*RATC(1))*ELM1

DO 60 I=1,5
  IF(DPA(I).LE.ELS.AND.ELS.LT.DPA(I+1)) GO TO 70
60  CONTINUE

C FS4 (LB) IS THE LOAD CALCULATED FROM THE CUBIC SPLINE
FIT FOR THE MATERIAL UNLOADING CHARACTERISTIC .......
70  D = ELS-DPA(I)
    FS4 = ((DPC(I,3)*D+DPC(I,2))*D+DPC(I,1))*D+DPO(I)

C VSF0 (SEC) IS THE LINEAR FUNCTION OF THE MAXIMUM STRAIN RATE .......
    VSF0 = 0.90 + VSFOM * DELOMAX

C FU3 (LB) IS THE VALUE OF FU* SCALED FOR CURRENT CYCLE MAXIMUM
STRAIN AND MODIFIED BY A LINEAR VISCOUS DAMPING TERM .......
    FU3 = FD4*RATIO*KDP*VSFD*ELM1-ELR/(ELM-ELR+1.E-6)

C FU3 = FSW (FD3, 0., 0., FD3)
    FU3 = FSW (FD3-FS, FD3, FU3, FS)

C FU1 (LB) IS THE SAME AS FD3 BUT LIMITED TO ZERO WHenever
THE STRAIN RATE IS ZERO OR POSITIVE  

FD1 = FSW(VX,FD3,0.,0.)  

FD2 (LB) IS THE SAME AS FD1 EXCEPT THAT IT HAS THE VALUE ZERO  
WHENEVER THE LENGTH IS LESS THAN THE CURRENT UNSTRESSED LENGTH  

FD2 = FSW(L-X,FD1,0.,0.,0.)  

FD (LB) IS THE CURRENT UNLOADING DECREMENT DERIVED FROM FD2  

FACTOR = SQRT(DELD/NMAXD2)  
IF(A/L.E.10.) FACTOR = AX*(FACTOR-1.)/10. + 1.  
FD = FSW(AX,FD2,FD2,FACTOR*FD2)  

CALCULATE THE TENSILE LOAD (FT)  

FT = FS - FD  
FIT = FT * GORES * FCTR * UTDL/D351.117  

DETERMINE THE CURRENT CREEP STRAIN RATE  

TF(SEC) IS THE CUMULATIVE TIME FOR WHICH TENSILE MEMBER EXPERIENCED NONZERO LOAD  

TFU = U.  
IF(FT.GT.0.0 .AND. ITF.NE.0) TFU = 1.  
IF(FT.GT.TC(6)) GO TO 80  

DEC (1/SEC) IS THE CURRENT CREEP STRAIN RATE  

DEC = TOLU2 (TF,FT,TC,FC,CSR,1,1.0,-3.0,3)  
 0 IF(FT.LT.0.0) DEC = U.0  
 0 IF(FT.GT.TC(6)) DEC = U.0  
 0 IF(IEC.NE.0) ECDOT = DEC*KCR*1.8  

RETURN  

END
SUBROUTINE LINDST (XYZ,R3I,OC, PT1,PT2,PT3)

DIMENSION XYZ(3),OC(3),PT1(3),PT2(3),PT3(3),DEL13(3)

C THIS ROUTINE CALCULATES THE COORDINATES OF THE INTERSECTION
C OF A NORMAL DRAWN FROM POINT THREE TO THE VECTOR PT1,PT2.
C THE DIRECTION COSINES AND MAGNITUDE OF THE NORMAL ARE ALSO
C CALCULATED.
C
***** LINDST OUTPUTS *****
C
XYZ(3) - X,Y,Z POSITION VECTOR OF THE INTERSECTION (FT)
R3I - MAGNITUDE OF THE NORMAL VECTOR (FT)
OC(3) - DIRECTION COSINES OF THE NORMAL VECTOR

C DETERMINE THE MAGNITUDE OF VECTOR PT1,PT2. DETERMINE ITS DIRECTION
C COSINES .......
C
R12=SQR((PT1(1)-PT2(1))**2+(PT1(2)-PT2(2))**2+(PT1(3)-PT2(3))**2)
DO 10 I=1,3
10 OC12(I)=(PT2(I)-PT1(I))/R12
C CALCULATE THE INTERSECTION POSITION VECTOR .......
C
DO 15 I=1,3
15 DELI3(I)=PT3(I)-PT1(I)
CALL DOTPRO (R11,DELI3,OC12,3)
C
DO 20 I=1,3
20 XYZ(I)=PT1(I)+R1I*OC12(I)
C CALCULATE THE DIRECTION COSINES OF THE NORMAL .......
C
R3I = SQRT ((XYZ(1)-PT3(1))**2+(XYZ(2)-PT3(2))**2+(XYZ(3)-PT3(3))**2)
RMIN = .02*R12
IF (R3I - RMIN) .GT .03,30,40
C
30 OC(1)=OC(2)=OC(3)=0
GO TO 60
C
40 DO 50 I=1,3
50 OC(I)=(XYZ(I)-PT3(I))/R3I
C
RETURN
END
SUBROUTINE LINEPL (X, C, XL, DC)
DIMENSION X(3), C(4), XL(3), DC(3)

C THIS ROUTINE DETERMINES THE COORDINATES OF THE INTERSECTION OF
C A LINE AND A PLANE.
C X(3) ARE THE COORDINATES OF THE INTERSECTION OF THE
C LINE WITH THE PLANE.
C THE PLANE IS DEFINED AS C(1)*X + C(2)*Y + C(3)*Z + C(4) = 0.
C THE LINE IS DEFINED AS HAVING DIRECTION COSINES DC(3), PASSING
C THROUGH A POINT WITH COORDINATES XL(3).

DP=C(1)*DC(1)+C(2)*DC(2)+C(3)*DC(3)
IF(DP.EQ.0.0)T=0
IF(DP.NE.0.0)T=(-C(4)-C(1)*XL(1)-C(2)*XL(2)-C(3)*XL(3))/DP
DO 10 I=1,3
10 X(I) = XL(I) + T*DC(I)
RETURN
END

385
SUBROUTINE LOOK(NN,R,VOUT)

C ================================= CALLING ARGUMENTS =================================
C NN - LOCATION IN R ARRAY OF DEPENDENT VARIABLE TABLE
C R - ARRAY CONTAINING AIRPLANE AERODYNAMIc TABLES
C VOUT - VALUE OF THE DEPENDENT VARIABLE DESIRED (OUTPUT)
C
C ================================================================================
C
C DIMENSION R(J),NIV(3),NSI(3),IND(3),NR(60)

COMMON /REGIONS/
  1 NR1, NR2, NR3, NR4, NR5, NR6, NR7, NR8, NR9, NR10, NR11,
  2 NR12, NR13, NR14, NR15, NR16, NR17, NR18, NR19, NR20, NR21, NR22,
  3 NR23, NR24, NR25, NR26, NR27, NR28, NR29, NR30, NR31, NR32, NR33,
  4 NR34, NR35, NR36, NR37, NR38, NR39, NR40, NR41, NR42, NR43, NR44,
  5 NR45, NR46, NR47, NR48, NR49, NR50, NR51, NR52, NR53, NR54, NR55,
  6 NR56, NR57, NR58, NR59, NR60
C
C EQUIVALENCE (NIV(1),NIV1),(NIV(2),NIV2),(NIV(3),NIV3),
C 1 (NSI(1),NSI1),(NSI(2),NSI2),(NSI(3),NSI3),
C 2 (IND(1),IND1),(IND(2),IND2),(IND(3),IND3)
C 3 (NR(I),NR1)
C
C NUMBER OF INDEPENDENT VARIABLES .......
C NI = R(NN)
C
C SET VOUT EQUAL TO ZERO IF THE NUMBER OF INDEPENDENT VARIABLES IS
C ZERO ......
C
C IF(NI .NE. 0) GO TO 10
C VOUT = 0.
C GO TO 50
C
C 10 K = NN + NI
C DO 20 I=1,NI
C
C LOCATION OF INDEPENDENT VARIABLE TABLES ......
C
C NIT = R(NN+1)
C NR11 = NR(NIT)
C
C NUMBER OF VALUES IN INDEPENDENT VARIABLE TABLE ......
C
C NIV(I) = K(NR11)
C
C LOCATION OF FIRST VALUE IN TABLE ......
C
C NSI1(I) = NRIT + 1
C
C LOCATION OF INDEPENDENT VARIABLE ......
C
C L = K(K+1) + 1
C IND(I) = L
C 20 CONTINUE
C
C LOCATION OF FIRST VALUE IN DEPENDENT VARIABLE TABLE ......

386
L
C
NO = NN + 2*N1 + 1
C
IF(N1.EQ.2) GO TO 30
IF(N1.EQ.3) GO TO 40
C
VOUT = TBLU1(R(IND1),R(NS11),R(ND),1,-NIV1)
GO TO 50
30 VOUT=TBLU2(R(IND1),R(IND2),R(NS11),R(NS12),R(ND),1,1,
   -NIV1,-NIV2,NIV1,NIV2)
   GO TO 50
   40 CALL TBLU3 (R(IND1),R(IND2),R(IND3),R(NS11),R(NS12),R(NS13),
                  R(ND),2,2,2,-NIV1,-NIV2,-NIV3,NIV1,NIV2,NIV3)
C
50 RETURN
END
SUBROUTINE MP (TMP,  
  EF, EFDOT, IEF, EL, ELDO T, IEL, WK, WKDOT, IWK,  
  WB, WBUDOT, IWB,  
  FL, FST, TST, FPP, TPP, FM, EXM, VM, TLO, PC, R, CVH,  
  TSO, FSD, TRM,  
  SW, XYZ, EA, XR, XD, ER, ED, UV, CSK, VI, PA, PT, CBP, C,  
  CI, PMW, GAM, TF, CI, C2Z, BXP, TI, TDE, SRP, UST, EST, WST,  
  XPP, UPP, LPP, WPP)

C

C DESIGNED BY C.L. WEST
C
C THE EASIEST PARACHUTE MORTAR COMPONENT
C
C ************** MP TABLES **************
C
C TMP - MORTAR PROPELLANT CONSUMPTION TABLE
C
C THE INDEPENDENT VARIABLE IS THE PROPELLANT
C WEB CONSUMED (IN) AND THE DEPENDENT VARIABLE
C IS THE PROPELLANT CONSUMED (SLUGS)
C
C ************** MP OUTPUTS **************
C
C INTERNAL FRICTION ENERGY .......
C
C EF - INTERNAL FRICTION ENERGY (FT-LB)
C EFDO T - INTERNAL FRICTION ENERGY RATE (FT-LB/SEC)
C IEF - INTEGRATION CONTROL
C
C HEAT LOSS ........
C
C EL - HEAT LOSS (FT-LB)
C ELDOT - HEAT LOSS RATE (FT-LB/SEC)
C IEL - INTEGRATION CONTROL
C
C MORTAR WORK .......
C
C WK - MORTAR WORK (FT-LB)
C WKDOT - MORTAR WORK RATE (FT-LB/SEC)
C IWK - INTEGRATION CONTROL
C
C PROPELLANT WEB BURNED ........
C
C WB - PROPELLANT WEB BURNED (IN)
C WBDOT - PROPELLANT WEB BURN RATE (IN/SEC)
C IWB - INTEGRATION CONTROL
C
C FL - MORTAR MODE FLAG
C 0 = PRIOR TO INITIATION
C 1 = INITIATION UP TO LAUNCH
C 2 = PARACHUTE LAUNCH
C 3 = MORTAR OFF
C
C FST(3) - XYZ SEAT BODY AXIS FORCE COMPONENTS OF THE
C MORTAR AND RESTRAINTS ON THE SEAT (LB)
C
C TST(3) - XYZ SEAT BODY AXIS TORQUE COMPONENTS OF THE
C MORTAR AND RESTRAINTS ON THE SEAT (FT-LB)
C
C FPP(3) - XYZ EARTH SYSTEM FORCE COMPONENTS OF THE
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPP(3)</td>
<td>X,Y,Z parachute pack body axis torque components of the mortar and restraints on the parachute pack (LB)</td>
</tr>
<tr>
<td>FM</td>
<td>Mortar force magnitude (LB)</td>
</tr>
<tr>
<td>EXM</td>
<td>Mortar extension (FT)</td>
</tr>
<tr>
<td>VM</td>
<td>Mortar extension velocity (FT/SEC)</td>
</tr>
<tr>
<td>TLO</td>
<td>Initial length of the mortar pressure chamber (IN)</td>
</tr>
<tr>
<td>PC</td>
<td>Circumference of catapult pressure chamber (IN)</td>
</tr>
<tr>
<td>K</td>
<td>Gas constant (FT-LBF/Slug-K)</td>
</tr>
<tr>
<td>CVH</td>
<td>Constant volume specific heat (FT-LBF/Slug-K)</td>
</tr>
<tr>
<td>TSO</td>
<td>Mortar stripoff time (SEC)</td>
</tr>
<tr>
<td>FSO</td>
<td>Force at mortar stripoff (LB)</td>
</tr>
<tr>
<td>TRM(3)</td>
<td>X,Y,Z seat Earth velocity components to pass to the parachute component during trim (FT/SEC)</td>
</tr>
<tr>
<td>SW</td>
<td>Flag if initiate mortar (1 = ON)</td>
</tr>
<tr>
<td>XYZ(3)</td>
<td>X,Y,Z seat body axis linear position vector of the parachute pack attachment point on the seat (FT)</td>
</tr>
<tr>
<td>EA(3)</td>
<td>Seat to parachute pack Euler angles (DEG)</td>
</tr>
<tr>
<td>XK</td>
<td>Parachute shelf linear spring constant (LB/FT)</td>
</tr>
<tr>
<td>XD</td>
<td>Parachute shelf linear damping constant (LB/FT/SEC)</td>
</tr>
<tr>
<td>ER(3)</td>
<td>X,Y,Z parachute shelf angular spring constant (FT-Lbf/Slug)</td>
</tr>
<tr>
<td>ED(3)</td>
<td>X,Y,Z parachute shelf angular damping constant (FT-Lbf/Slug/SEC)</td>
</tr>
<tr>
<td>UV(3)</td>
<td>X,Y,Z seat body axis mortar force unit vector</td>
</tr>
<tr>
<td>CSK</td>
<td>Mortar stroke (FT)</td>
</tr>
<tr>
<td>V1</td>
<td>Initial free volume (IN**3)</td>
</tr>
<tr>
<td>PA</td>
<td>Piston area (IN**2)</td>
</tr>
<tr>
<td>PT</td>
<td>Tang release pressure (LB/IN**2)</td>
</tr>
<tr>
<td>CBP</td>
<td>Mortar burst pressure (LB/IN**2)</td>
</tr>
<tr>
<td>CI</td>
<td>Mass of total propellant (SLUGS)</td>
</tr>
<tr>
<td>PMW</td>
<td>Propellant molecular weight (LB/(LB-MOLE))</td>
</tr>
<tr>
<td>GAM</td>
<td>Ratio of specific heats</td>
</tr>
<tr>
<td>TF</td>
<td>Constant volume flame temperature (DEG K)</td>
</tr>
<tr>
<td>C1</td>
<td>Friction proportionality constant</td>
</tr>
<tr>
<td>C2</td>
<td>Heat loss constant</td>
</tr>
<tr>
<td>BAP</td>
<td>Burn rate proportionality constant (IN/SEC/(LB/IN**2))</td>
</tr>
<tr>
<td>RAP</td>
<td>Burn rate exponent</td>
</tr>
<tr>
<td>TL</td>
<td>Mortar temperature prior to ignition (DEG K)</td>
</tr>
<tr>
<td>TUE</td>
<td>Mortar force decay time (SEC)</td>
</tr>
<tr>
<td>SKP(3)</td>
<td>X,Y,Z Earth system position vector of the seat reference point (FT)</td>
</tr>
<tr>
<td>UST(3)</td>
<td>X,Y,Z seat body axis velocity vector of the seat (FT/SEC)</td>
</tr>
<tr>
<td>EST(3)</td>
<td>Earth to seat Euler angles (DEG)</td>
</tr>
<tr>
<td>WST(3)</td>
<td>X,Y,Z seat body axis angular velocity of the seat (DEG/SEC)</td>
</tr>
<tr>
<td>XPP(3)</td>
<td>X,Y,Z Earth system position vector of the parachute pack (FT)</td>
</tr>
<tr>
<td>UPP(3)</td>
<td>X,Y,Z Earth system velocity vector of the parachute pack (FT/SEC)</td>
</tr>
<tr>
<td>EPP(3)</td>
<td>Earth to parachute pack Euler angles (DEG)</td>
</tr>
<tr>
<td>WPP(3)</td>
<td>X,Y,Z parachute pack body axis angular velocity of the parachute pack (DEG/SEC)</td>
</tr>
</tbody>
</table>

389
DIMENSIONS OF CALLING ARGUMENTS

DIMENSION TMP(5),FST(3),IST(3),FPP(3),TPP(3),XYZ(3),EA(3),ER(3)
  ,ED(3),UV(3),SRP(3),UST(3),EST(3),WST(3),XPP(3),UPP(3),
  EPP(3),WPP(3),TRM(3)

INTERNAL DIMENSIONS

DIMENSION DES(3,3),DEST(3,3),DEP(3,3),DEPT(3,3),DSP(3,3),
  DELTAV(3),DELTAX(3),ESTIR(3);WSTIR(3),EPPIR(3),
  WPPIR(3),XS(3),UXSE(3),RVEL(3),
  DAMP(3),ANG(3),WSTE(3),WPPE(3),PROJ(3),
  FCAD(3),TURGUE(3),EAIR(3),OCEA(3,3),OCEAT(3,3),TEMP(3)

COMMON / CTIME / TIME
COMMON / CICCAL / CICCAL
COMMON / COVRLY / INST
COMMON / CSSFLG / CSSFLG
COMMON / CIW/IREAD,WRITE,IDIAG

DATA RPL,OPR / .01745329, 57.29576 /

******************************
INITIALIZATION ******
******************************

IF(IICCAL.NE.1) GO TO 40

COMPUTE THE INITIAL LENGTH (TLU) AND CIRCUMFERENCE (PC) OF THE MORTAR PRESSURE CHAMBER

TLU = VI/PA
PC = 2*SQRT(R(3.14159*PA))

COMPUTE THE CONSTANT VOLUME SPECIFIC HEAT (CVH) FOR THE MORTAR PROPELLANT GIVEN THE GAS CONSTANT (GC) AND THE PROPELLANT MOLECULAR WEIGHT (PMW)

R = 879475.894/PMW
CVH = R/(GAM-1.0)

TYPE = CMHMTAR
VM = EXM = FM = FL = TSO = FSO = 0
TRM(1) = TRM(2) = TRM(3) = 0
IF(TOE .EQ. 0.99999) TOE = 0
DO 30 I = 1,3
  30 EAIR(I) = EAIR(I) * RPD
CALL DIRCOS (OCEA,EAIR)
CALL TRANS (OCEA,OCEA,3,3)

BYPASS THE REMAINING CODE IF THE MORTAR IS PAST THE STRIPOFF

IF(FL.EQ.3.0) GO TO 260
C CHANGE ANGULAR STATES FROM DEGREES TO RADIANS

DU 50 1=1,3
ESTIK(I) = EST(I) * RPD
WSTIR(I) = WST(I) * RPD
EPPIR(I) = EPP(I) * RPD
50 WPPIR(I) = WPP(I) * RPD

C CALCULATE THE EARTH TO SEAT MATRIX

CALL DIRLOS (DES,ESTIR)

C CALCULATE THE SEAT TO EARTH MATRIX

CALL TRANS (DEST,DES,3,3)

C CALCULATE THE EARTH TO PARACHUTE PACK MATRIX

CALL DIRLOS (DEP,EPPIR)

C CALCULATE THE PARACHUTE PACK TO EARTH MATRIX

CALL TRANS (DEPT,DEP,3,3)

C CALCULATE THE SEAT TO PARACHUTE PACK MATRIX

CALL MATMPY (DSP,DEP,DEST,3,3,3)

******************************************************************************
**** FORCES AND TURQUES DUE TO LINEAR DISPLACEMENT ******
******************************************************************************

------ LINEAR SPRING FORCES -------

CALL VELOCITY (XS,XPP,SKP,DES,1)

Determine the linear displacement from the attachment point
and calculate the spring forces in the seat system acting on
the seat

DU 60 1=1,3
DELTAX(I) = XS(I) - XYL(I)
60 SPRLNG(I) = DELTAX(I) * AR

------- LINEAR DAMPING FORCES -------

Determine the earth velocity of the position the parachute pack
occupies in the seat coordinate system

CALL VELOCITY (UX,UST,XS,WSTIR,DEST)

Determine the relative velocity wrt the earth system

DU 70 1=1,3
70 DELTAV(I) = UPP(I) - UXSE(I)
TRANSFORM THIS DIFFERENCE INTO THE SEAT SYSTEM ... 
CALL MATMPY (RVEL, DES, DELTAV, 3, 1)

COMPRESS THE DAMPING FORCE ACTING ON THE SEAT ... 
DO 60 I = 1, 3
  60  DAMP(I) = RVEL(I) * XD

SUM THE SPRING AND DAMPING FORCES ACTING ON THE SEAT --- 
DO 90 I = 1, 3
  90  FST(I) = SPRING(I) + DAMP(I)

**********
*** MORTAR LOGIC ***
**********

IF (SW.NE.1.) GO TO 170

CALCULATE THE MORTAR EXTENSION ... 
CALL DOTPRO (EXM, DELTAX, UV, 3)

CALCULATE THE MORTAR EXTENSION VELOCITY ... 
CALL DOTPRO (VM, DELTAV, UV, 3)

COMPUTE THE EXPOSED THERMAL AREA OF THE MORTAR CHAMBER ... 
THA = PC * (TLO + EXM*12.) + PA * 2.

COMPUTE THE FORCE DUE TO THE MORTAR PRESSURE ... 
CALL CAD (FM, EF, EFOOT, IF, IEL, ELDO, IEL, WK, WKDOT, IWK, WB, WBOOT, IW8,
  FL, TMP, TIME, EXM, CSA, CI, C, VI, PA, T, CV, CBP, CI, VM, C2, TI,
  THA, D, EXP, PI, R, TYPE, TSO, FSO, TOE)

IF THE MORTAR IS AT STIPPOFF ... 
IF (FL.NE.3.) GO TO 120
DO 110 I = 1, 3
  110  FST(I) = TST(I) = FPP(I) = TPP(I) = 0
GO TO 260

CALCULATE THE SEAT BUDY AXIS MORTAR FORCE COMPONENTS 
ACTING ON THE SEAT ... 
DO 120 I = 1, 3
  120  FCAD(I) = -1. * FM * UV(I)

DOT THE LINEAR SPRING FORCE ONTO THE MORTAR UNIT VECTOR ... 
CALL DOTPRO (DOT, SPRING, UV, 3)

IF THE SIGN OF THE DOT PRODUCT IS NEGATIVE, RETAIN THE SHELF FORCE ... 

IF(UU*T.LE.0) GO TO 155
C
C OUT THE TOTAL LINEAR RESTRAINT FORCE ON THE UNIT VECTOR .......
C
CALL DOTPRD (DOT,FST,UV,3)
C
C CALCULATE THE COMPONENTS OF THIS PROJECTION .......
C
DO 140 I=1,3
140 PKOJ(I) = DOT * UV(I)
C
Determine the force vector normal to the unit vector .......
C
DO 150 I=1,3
150 FST(I) = FST(I) - PKOJ(I)
C
CALCULATE THE TOTAL FORCES AND MOMENTS ACTING ON THE SEAT .......
C
155 DO 160 I=1,3
160 FST(I) = FCAD(I) + FST(I)
C
**********************************************************************
C
170 CALL CRSPRD (TORQUE, XS, FST)
C
CALCULATE THE FORCES ACTING ON THE PARACHUTE PACK .......
C
CALL MATMPY (FPP, DEST, FST, 3, 3, 1)
DO 160 I=1,3
160 FPP(I) = -FPP(I)
C
**********************************************************************
C
***** Torsion due to angular displacement *****
**********************************************************************
C
------ Angular Spring Forces ------
C
CALCULATE THE SEAT TO PARACHUTE PACK EULER ANGLES .......
C
CALL COSDIR (ANG, D3P)
C
DETERMINE THE ANGULAR DISPLACEMENT FROM THE ATTACHMENT ANGLE,
C AND CALCULATE THE SPRING COMPONENTS ACTING ON THE SEAT IN THE
C ATTACHMENT AXIS SYSTEM .......
C
DO 190 I=1,3
190 DELTAX(I) = ANG(4-I)*UPK - EA(4-I)
190 SPRING(I) = DELTAX(I) * ER(I)
C
------ Angular damping forces ------
C
CALCULATE THE BODY AXIS ANGULAR DAMPING CONSTANTS ACTING
C ON THE SEAT IN THE ATTACHMENT AXIS SYSTEM .......
C
CALL MATMPY (WS,EDEST, WST, 3, 3, 1)
CALL MATMPY (WPPE, DEP, WPP, 3, 3, 1)
DO 200 I=1,3
DO 200 I=1,3
200 DELTAV(I) = WPPE(I) - WSTE(I)
  CALL MATMPY (TEMP, DELS, DELTAV, 3, 3, 1)
  CALL MATMPY (VEL, DGEA, TEMP, 3, 3, 1)
  DO 210 I=1,3
  DAMP(I) = RVEL(I) * ED(I)
210  TEMP(I) = SPRING(I) + DAMP(I)
C
C MOVE THE RESTRAINT TORQUES INTO THE SEAT SYSTEM .......
C
  CALL MATMPY (IST, DGEAT, TEMP, 3, 3, 1)
C
C CALCULATE THE BODY AXIS TORQUE CONSTANTS ACTING ON THE
C PARACHUTE PACK .......
C
  CALL MATMPY (TPP, DSP, IST, 3, 3, 1)
  DO 220 I=1,3
220  TPP(I) = -TPP(I)
C
C CALCULATE THE TOTAL MOMENT ON THE SEAT .......
C
  DO 230 I=1,3
230  TST(I) = TST(I) + TORQUE(I)
C
C ZERO THE FORCES AND TORQUES ACTING ON THE SEAT IF SSFLG
C IS EQUAL TO ZERO .......
C
  IF(SSFLG.NE.0) GO TO 250
  DO 240 I=1,3
240  FST(I) = TST(I) = 0
C
C SEND DATA TO PARACHUTE PACK BODY TO ALLOW IT TO COMPUTE
C SEAT EARTH VELOCITY DURING TRIM .......
C
  IF (INST.NE.31) GO TO 260
  CALL MATMPY (TRM, DEST, UST, 3, 3, 1)
C
  260 RETURN
END
SUBROUTINE PAXIS (BMI, BPI, BM, BP, BMASS, DISP)

PARALLEL AXIS THEOREM FOR TRANSFERRING THE MOMENTS AND
PRODUCTS OF INERTIA TO THE SEAT BODY AXIS

**** CALLING PARAMETERS ****

OUTPUT .......

BMI    - MASS MOMENT OF INERTIA WITH RESPECT TO THE SEAT
        BODY AXIS (SLUG-FT**2)
BPI    - MASS PRODUCT OF INERTIA WITH RESPECT TO THE SEAT
        BODY AXIS (SLUG-FT**2)

INPUT .......

BM    - MASS MOMENT OF INERTIA ABOUT THE BODY MASS CENTER
        (SLUG-FT**2)
BP    - MASS PRODUCT OF INERTIA ABOUT THE BODY MASS CENTER
        (SLUG-FT**2)
BMASS - BODY MASS (SLUGS)
DISP  - X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE BODY
        MASS CENTER (FT)

DIMENSION BMI(3), BPI(3), BM(3), BP(3), DISP(3)

INTERNALLY DEFINED FUNCTIONS .......

TRANSM(A,B,C,D) = A+B*(C**2+D**2)
TRANSP(A,B,C,D) = A+B*C*D

COMPUTE NEW INDIVIDUAL INERTIA PROPERTIES .......

BM1(1) = TRANSM (BM1(1), BMASS, DISP(2), DISP(3))
BM1(2) = TRANSM (BM1(2), BMASS, DISP(1), DISP(3))
BM1(3) = TRANSM (BM1(3), BMASS, DISP(1), DISP(2))
BPI(1) = TRANSP (BPI(1), BMASS, DISP(1), DISP(2))
BPI(2) = TRANSP (BPI(2), BMASS, DISP(1), DISP(3))
dP1(3) = TRANSP (dP1(3), BMASS, DISP(2), DISP(3))

RETURN
END
SUBROUTINE PC (UPP, UPPD, IUUPP, XPP, XPPD, IXPP, WPP, WPPD, IWP5,
  EPS, EPDP, IEPP, UPPC, UPDC, IUPC, XPC, XPDC, IXPC,
  PHA, SW, FLF1T, FDRAG, FMDOT, RM, VOL, TLA, TLS, IDS, DTI,
  TOU, TRF, STI, RSC, RFM, RFD, RFS, BFI, CT, CN, CM, FO, PWT,
  PMI, PPI, TEM, CSF, COP, DPG, FLA, FLP, FP, TP, VAP, UVL, RL,
  VCG, PCG, CWT, TPE, TRM)

C DESIGNED BY C.L. WEST
C LAST MODIFIED - December 8, 1980
C
THE EASIEST PARACHUTE MODEL

*************** PC OUTPUTS ***************

PARACHUTE PACK LINEAR VELOCITIES - EARTH SYSTEM

UPP(3)  - X,Y,Z LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK
         CENTER OF GRAVITY (FT/SEC)

UPPD(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR OF THE PARACHUTE
         PACK CENTER OF GRAVITY (FT/SEC)

IUUPP(3) - INTEGRATION CONTROL

PARACHUTE PACK LINEAR POSITIONS - EARTH SYSTEM

XPP(3)  - X,Y,Z LINEAR POSITION VECTOR OF THE PARACHUTE PACK
         CENTER OF GRAVITY (FT)

XPPD(3) - X,Y,Z LINEAR POSITION RATE VECTOR OF THE PARACHUTE
         PACK CENTER OF GRAVITY (FT/SEC)

IXPP(3) - INTEGRATION CONTROL

PARACHUTE PACK ANGULAR VELOCITIES - BODY AXIS

WPP(3)  - X,Y,Z ANGULAR VELOCITY COMPONENTS - P,Q,R (DEG/SEC)

WPPD(3) - X,Y,Z ANGULAR VELOCITY RATE COMPONENTS (DEG/SEC/SEC)

IWPP(3) - INTEGRATION CONTROL

EULER ANGLES --- EARTH TO PARACHUTE PACK --- YAW,PITCH,ROLL

EP5(3)  - EARTH TO PARACHUTE PACK EULER ANGLES (DEG)

EPDP(3) - EULER ANGLE RATES (DEG/SEC)

IEPP(3) - INTEGRATION CONTROL

PARACHUTE CANOPY LINEAR VELOCITIES - EARTH SYSTEM

UPC(3)  - X,Y,Z LINEAR VELOCITY VECTOR OF THE PARACHUTE
         CANOPY CENTER OF GRAVITY (FT/SEC)

UPCD(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR OF THE PARACHUTE
         CANOPY CENTER OF GRAVITY (FT/SEC/SEC)

IUUPC(3) - INTEGRATION CONTROL

PARACHUTE CANOPY LINEAR POSITION - EARTH SYSTEM

XPC(3)  - X,Y,Z POSITION VECTOR OF THE PARACHUTE CANOPY
         CENTER OF GRAVITY (FT)

XPCD(3) - X,Y,Z POSITION RATE VECTOR OF THE PARACHUTE CANOPY
         CENTER OF GRAVITY (FT/SEC)
C IXPC(3) - INTEGRATION CONTROL
C
C PHA - PARACHUTE PHASE
1 = PRIOR TO PARACHUTE LAUNCH
2 = FROM LAUNCH UP TO LINESRETCH
3 = AFTER LINESRETCH
C SW - FLAG TO INOLIATE AERODYNAMIC CALCULATION MODE
0 = PRIOR TO LAUNCH
1 = FROM PARACHUTE LAUNCH TO LINESRETCH
2 = DURING INFLATION
3 = DURING REEFING
4 = AFTER REEFING
5 = PARACHUTE INFLATED
C FLIFT(3) - x,y,z EARTH SYSTEM AEROYDYNAMIC LIFT COMPONENTS (LB)
ACTING ON THE PACK BEFORE LINESRETCH
ACTING ON THE CANOPY AFTER LINESRETCH
C FURAG(3) - x,y,z EARTH SYSTEM AEROYDYNAMIC DRAG COMPONENTS (LB)
ACTING ON THE PACK BEFORE LINESRETCH
ACTING ON THE CANOPY AFTER LINESRETCH
C FMDOT(3) - x,y,z EARTH SYSTEM FORCE COMPONENTS ACTING ON THE
CANOPY DUE TO AIR MASS ACQUISITION FORCE (LB)
C RM - RADIUS OF THE SPHERE REPRESENTING THE INFLATED CANOPY (FT)
C VOL - VOLUME OF THE FILLED CANOPY (FT**3)
C TLA - PARACHUTE LAUNCH TIME / LINE SEVERING TIME (SEC)
C TLS - LINESRETCH TIME (SEC)
C TUS - TIME AT WHICH DISREEF OCCURS (SEC)
C UTI - PARACHUTE CANOPY INFLATION TIME (SEC)
C TDU - TIME DURATION OF REEFED PARACHUTE (SEC)
C TRF - THE TIME AT WHICH THE CHUTE IS REEFED (SEC)
C
************** PC INPUTS **************
C STI - INFLATED PARACHUTE DRAG AREA (FT**2)
C RSC - CIRCUMFERENCE OF THE FILLED CANOPY PLUS ONE QUARTER
OF THAT DISTANCE (FT)
C RFM - REEF MODE FLAG
0 = CHUTE IS NOT REEFED
1 = TIME OF DISREEF SET AT PARACHUTE INITIATION
2 = TIME OF DISREEF SET AT LINESRETCH
C RFU - REEF DELAY TIME (SEC)
C RFS - PRODUCT OF REFERENCE AREA AND TANGENT FORCE
COEFFICIENT WHEN REEFED (FT**2)
C B - CONSTANT USED IN THE EQUATION FOR CALCULATING
SCD OF THE REEFED PARACHUTE
C CI - CONSTANT USED IN THE EQUATION TO COMPUTE THE CANOPY
INFLATION TIME
C CT(3) - CONSTANTS USED IN THE EQUATION THAT CALCULATES THE
TANGENTIAL DRAG AREA
C CN(3) - CONSTANTS USED IN THE EQUATION THAT CALCULATES THE
NORMAL DRAG AREA
C CM(2) - CONSTANTS USED IN THE MACH EFFECTS EQUATION
C PD - WAKE TO FREE STREAM RATIO
C PWT - TOTAL WEIGHT OF THE PARACHUTE PACK (LB)
C PMH(3) - PARACHUTE PACK MOMENTS OF INERTIA - IXX, IYY, IZZ
(SLUGS*FT**2)
C PPI(3) - PARACHUTE PACK PRODUCTS OF INERTIA - IXY, IXZ, IYZ
(SLUGS*FT**2)
C TEM - TIME DURATION FOR PARACHUTE EMERGENCE (SEC)

397
CSP - PARACHUTE CANOPY SPRING CONSTANT (LB/FT)
CDP - PARACHUTE CANOPY DAMPING CONSTANT (LB/FT/SEC)
DPG(3) - PARACHUTE PACK DAMPING AFTER MORTAR IS OFF (1/SEC)

FLA - PARACHUTE MODE FLAG
  0 = PRIOR TO INITIATION
  1 = INITIATION
  2 = LAUNCH
  3 = MORTAR OFF
  4 = LINESTRETCH
  5 = LINES SEVERED

FLP(3) - X,Y,Z FORCE COMPONENTS ACTING ON THE PARACHUTE FROM
  THE LINES (LB)
  (BODY AXIS FOR THE PACK - EARTH SYSTEM FOR THE CANOPY)

FP(3) - X,Y,Z PARACHUTE PACK BODY AXIS FORCE COMPONENTS ACTING
  ON THE PACK FROM THE MORTAR OR GUN (LB)

TP(3) - X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS ACTING
  ON THE PACK FROM THE MORTAR OR GUN (FT-LB)

VAP(3) - X,Y,Z EARTH SYSTEM VELOCITY COMPONENTS OF THE
  FORCE APPLICATION POINT (FT/SEC)

UVL(3) - EARTH SYSTEM PARACHUTE LINE UNIT VECTOR

RL - PARACHUTE LINE LENGTH (FT)

VCG - VELOCITY OF THE CANOPY CENTER OF GRAVITY ALONG THE
  PARACHUTE LINES (FT/SEC)

PCG - STRETCHED CANOPY CENTER OF GRAVITY MEASURED ALONG THE
  PARACHUTE LINE FROM THE PARACHUTE PACK (FT)

CWI - WEIGHT OF THE CANOPY DRAWN FROM THE PACK (LB)

TPt - TYPE OF PARACHUTE (1=DKAG  2=RECOVERY)

TRM(3) - X,Y,Z PARENT BODY EARTH VELOCITY COMPONENTS
  TO DETERMINE THE POSITION RATES DURING TRIM (FT/SEC)

DIMENSION OF CALLING ARGUMENTS .......

DIMENSION UPP(3),UPPD(3),XPP(3),XPPD(3),IXPP(3),
  WPP(3),WPPU(3),WPPD(3),EPP(3),EPPD(3),IEPP(3),
  UPC(3),UPCU(3),UPCD(3),XPC(3),XPCD(3),IXPC(3),
  FLP(3),TP(3),CT(3),CN(3),CM(2),
  PMI(3),PPI(3),DPG(3),UVL(3),VAP(3),TRM(3)

INTERNAL DIMENSIONS ......

DIMENSION XPPR(3),WPPR(3),FFSPR(3),FDAMP(3),FP(3),
  TP(3),FPC(3),FLIFT(3),FDRAG(3),FMDOT(3),
  TEMPI(3),TEMP2(3),TEMP3(3),TIER(3),
  DEP(3,3),DEPT(3,3),XCG(3),UCG(3)

COMMON /CICCAL/CCCAL
COMMON /CTIME/ CTIM
COMMON /COVRLY/ INST
COMMON /CSSFLG/ CSSFL
COMMON /CIEL/ IREAD,WRITE,IOUT

DATA RPU,DPK /0.01745329, 57.29578 /
DATA GRAV /32.174/

************************************
**** INITIALIZATION *****

398
C **-----------------------------**
C IF(ICCAL.NE.1) GO TO 10
C CALCULATE THE FILLED RADIUS AND VOLUME ....
C RM = .0360 * RSC
VGL = 4.168 * RM**3
C MIS Initialization ....
C DO 5 I=1,3
  5 FLIFT(I) = FDKAG(I) = FMDGI(I) = 0
  PHA = 1.
  TLA = TIS = TUS = LII = TDU = TRF = SW = 0
  TKM(I) = TRM(2) = TKM(3) = 0
  IF(TEM.EQ.0.99999) TEM = 0
  IF(CSP.EQ.0.99999) CSP = 0.000.
  IF(CDP.EQ.0.99999) CDP = 14.
  IF(RFM.EQ.0.99999) KFM = 0
  IF(RFU.EQ.0.99999) RFD = 0
  IF(RFS.EQ.0.99999) RFS = 0
  IF(RFE.EQ.0.99999) RFE = 0
C /\\\\
C ------- COMPUTE THE INERTIA TENSOR -------
C 10 TINER(1,1) = PMI(1)
    TINER(1,2) = -PPI(1)
    TINER(1,3) = -PPI(2)
    TINER(2,1) = -PPI(1)
    TINER(2,2) = PMI(2)
    TINER(2,3) = -PPI(3)
    TINER(3,1) = -PPI(2)
    TINER(3,2) = -PPI(3)
    TINER(3,3) = PMI(3)
C CONVERT FROM DEGREES TO RADIANS ....
C DU <0 I=1,3
  EPPIR(I) = EPP(I) * RPD
  WPPIR(I) = WPP(I) * RPD
C CALCULATE THE DIRECTION COSINE MATRICES ....
C CALL DIRCOS (DEP,EPPIR)
  CALL TRANS (DEPT,DEP,3,3)
C DEFINE CHUTE ....
C  IF(TPE.EQ.1.) CHUTE = 4HDRAG 
  IF(TPE.EQ.2.) CHUTE = 6HRRECOVERY
C  IF(PHA.EQ.2.) GO TO 90
  IF(PHA.EQ.3.) GO TO 266
C **-----------------------------**
** PHASE 1 **

** PRIOR TO PARACHUTE LAUNCH **

************************************************************

--- DEFINE VARIABLES AT PARACHUTE INITIATION AND LAUNCH ----

IF (FLA.EQ.0) GO TO 40

AT PARACHUTE INITIATION ....

IF(RFM.EQ.1 .AND. TUS.EQ.0) TDS = TIME + RFD

AT PARACHUTE LAUNCH ....

IF (FLA.EQ.1.0) GO TO 40

PMA = 2.0

SW = 1.

IF(INST.EQ.26) WRITE (6,30) CHUTE, TIME

30 FORMAT(5A,6B,* CHUTE LAUNCH AT TIME = *,F10.4,* SEC*1)

GO TO 90

--- DRIVE THE PARACHUTE CANOPY TO ITS CG POSITION ----

CALCULATE THE SPRING FORCE ON THE CANOPY ....

DU 50 I=1,3

50 FSPR(I) = CSP * (APP(I) - APC(I))

CALCULATE THE DAMPING FORCE ON THE CANOPY ....

DU 60 I=1,3

60 FDAMP(I) = CDP * (UPP(I) - UPC(I))

--- SUM FORCES AND TURQUES ACTING ON THE PARACHUTE PACK ----

DO 70 I=1,3

70 FPP(I) = FP(I)

IP TP(I) = TP(I)

FPP(1) = FPP(1) + PWT * SSFLG

PMASS = PWT/IGRAV

--- SUM THE FORCES ACTING ON THE PARACHUTE CANOPY ----

DU 80 I=1,3

80 FPC(I) = FSPR(I) + FDAMP(I)

FPC(3) = FPC(3) + 1. * SSFLG

CMASS = 1.0/IGRAV

GO TO 370

--- PHASE 2 ---

*******************************************************
** FROM PARACHUTE LAUNCH TO LINESSTRETCH **

************

90 IF (FLA.EQ.4.) GO TO 240

CALL PCAAERO (FLIFT,FDRAG,FMDOT,SCT,
   
SM,XPP,UPP,TLS,DTI,TDO,VOL,UVL,CT,
   
-CN,CM,FO,B,STI,RFS,FLA,TLA,TEM)

FACTOR THE AERODYNAMIC FORCES DURING EMERGENCE......

DELTAT = TIME - TLA
IF(DELTAT.GE.TLM) GO TO 120
FACTOR = 0
IF(TLM.NE.0) FACTOR = DELTAT/TLM
DO 110 I=1,3
   FLIFT(I) = FLIFT(I) * FACTOR
   FDRAG(I) = FDRAG(I) * FACTOR
110

DRIVE THE PARACHUTE CANOPY TO ITS CG POSITION......

DO 120 J=1,3
120 XCG(I) = XPP(I) + PCG * UVL(I)

DETERMINE THE SPRING FORCE ACTING ON THE CANOPY......

DO 140 I=1,3
140 FSPR(I) = CSP * (XCG(I) - XPC(I))

CALCULATE THE VELOCITY OF THE PARACHUTE PACK RELATIVE TO THE
FORCE APPLICATION POINT......

DO 150 I=1,3
150 TEMP1(I) = UPP(I) - VAP(I)

DETERMINE THE VECTOR COMPONENT OF THIS RELATIVE VELOCITY NORMAL
TO THE LINES......

CALL DOTPKD (DIST,TEMP1,UVL,3)
DO 160 I=1,3
160 TEMP2(I) = DIST * UVL(I)
DO 170 I=1,3
170 TEMP3(I) = TEMP1(I) - TEMP2(I)

RATIO THIS VECTOR ACCORDING TO THE POSITION OF THE CANOPY CG ALONG
THE LINES......

RATIO = (RL-PCG)/RL
DO 180 I=1,3
180 TEMP3(I) = TEMP3(I) * RATIO

COMPUTE THE EARTH VELOCITY OF THE CANOPY CG POSITION ON THE
LINES......

401
DO 196 I=1,3
196 UCG(I) = VAP(I) + TEMP3(I) - VCG * UVL(I)
C
DETERMINE THE EARTH VELOCITY DIFFERENCE BETWEEN THE CANOPY
C AND THE CANOPY CG POSITION ......
C
DO 200 I=1,3
200 TEMPI(I) = UCG(I) - UPC(I)
C CALCULATE THE DAMPING FORCE ON THE CANOPY ......
C
DO 210 I=1,3
210 FDAMP(I) = COP * TEMPI(I)
C ---- SUM THE FORCES AND TORQUES ACTING ON THE PARACHUTE PACK ----
C
WDIFF = PW - CW
DO 220 I=1,3
220 FPP(I) = FLIFT(I) + FDRAG(I) + FLP(I) + FP(I)
FPP(3) = FPP(3) + WDIFF * SSFLG
PMASS = WDIFF / GRAV
C ---- SUM THE FORCES ACTING ON THE PARACHUTE CANOPY ----
C
DO 230 I=1,3
230 FPL(I) = FSPR(I) + FDAMP(I)
FPC(3) = FPC(3) + 1.0 * SSFLG
CMASS = 1.0 / GRAV
C
GO TO 370
C ---- AT LINESTRETCH ----
C
PHA = 3.
SW = 2.
TLS = TIME
TLA = U
C
SET WITHF TIME ......
C
IF(KFM.EQ.2) TUS = TIME + RFW
C
CALCULATE THE CHUTE INFLATION TIME ......
C
VBAR = SQRT(VAP(1)**2 + VAP(2)**2 + VAP(3)**2)
DT1 = C1 * 2.0 * RSC/VBAR
C
**************************************************************************
C **
C ** PHASE 3 **
C **
C ** AFTER LINESTRETCH **
C
**************************************************************************

402


C ----- CALCULATE THE AERODYNAMIC FORCES ----- 

200 GO TO (270,270,290,310,340), SW

**** SW = 2 (DURING INFLATION) ****

270 IF(TIME.GE.TLS+DT1) GO TO 320
   GO TO 340

275 IF(SCT.LT.RFS) GO TO 345
   SW = 3.
   TRF = TIME
   TDU = TDS - TIME
   IF(INST.EQ.26) WRITE(0,280) CHUTE,TIME
   280 FORMAT(5X,A8,* CHUTE REEDED AT TIME = *,F10.4,* SEC*)
   GO TO 345

**** SW = 3 (DURING REEETING) ****

290 IF(TIME.LT.TDS) GO TO 340

   SW = 4.
   IF(INST.EQ.26) WRITE(0,300) CHUTE,TIME
   300 FORMAT(5X,A8,* CHUTE DISREEDED AT TIME = *,F10.4,* SEC*)
   GO TO 340

**** SW = 4 (AFTER REEING) ****

310 IF(TIME.GE.TLS+DT1+TDU) GO TO 320
   GO TO 340

C AT THE TIME THE CANOPY IS FILLED

320 SW = 5.
   IF(INST.EQ.26) WRITE(0,330) CHUTE,TIME
   330 FORMAT(5X,A8,* CANOPY FILLED AT TIME = *,F10.4,* SEC*)

C DETERMINE THE LIFT AND DRAG FORCES

340 CALL PCAERJ (FLIFT,FDRAG,FMDOT,SCT,
               SW,XPC,UPC,TLS,DTI,TDU,VL,VU,CT,
               CN,CM,FD,BS,STI,RFS,FLA,TLA,TEM)

   IF(RFM.NE.0 .AND. SW.EQ.2.) GO TO 275

C ----- SUM THE FORCES ACTING ON THE PARACHUTE CANOPY ----- 

345 DO 350 I=1,3
   350 FPC(I) = FLIFT(I) + FDRAG(I) + FMDOT(I) + FLP(I)
   FPC(3) = FPC(3) + CM*SSFLG
   CMASS = CM/GRAV

C ----- SUM FORCES AND TURWUES ACTING ON THE PARACHUTE PACK ----- 

360 WDIFF = PW - CM
   DO 360 I=1,3
   360 FPP(I) = TPP(I) = 0
   FPP(3) = WDIFF * SSFLG

        403
PMASS = WDIFF/GRAV

***** PARACHUTE PACK EQUATIONS OF MOTION *****

CALCULATE TINEK * WPPiR

370 CALL MATMPY (TEMPI, TINEK, WPPiR, 3, 3, 1)

CALL WPPiR X (TINEK * WPPiR)

CALL CRSPRD (TEMPI, WPPiR, TEMPI)

SUM TERMS TO OBTAIN TOTAL TORQUE......

DO 360 I=1,3
360 TEMP3(I) = TPIR(I) - TEMP2(I)

CALCULATE WPPR......

CALL LUEGS (TINEK, TEMPI, TEMP3, TEMP2, 3, 1, 3, 1, E-14, 1ERROR)

IF (1ERROR .NE. 1) GO TO 460

WRITE(6, 390) CHUTE

STOP

FORMAT(* INERTIA MATRIX OF *AB* CHUTE IS SINGULAR...RUN STOPPED*)

DO 400 I=1,3
400 IF (1WPP(I).NE.0) WPPD(I) = TEMPI(I) * DPR

DO 420 I=1,3
420 IF (1EPP(I).NE.0) EPPD(I) = TEMPI(I) * DPR

CALL LARATE (TEMPI, WPPiR, EPPiR)

DO 440 I=1,3
440 IF (1UPP(I).NE.0) UPPD(I) = FPP(I)/PMASS

DO 450 I=1,3
450 IF (1XPP(I).NE.0) XPPD(I) = APPD(I)

DO 460 I=1,3
460 IF (1XPP(I).NE.0) XPPD(I) = APPD(I) - TRM(I)

CONTINUE

DO 470 I=1,3
470 CONTINUE

CALL LUEGS (TINEK, TEMPI, TEMP3, TEMP2, 3, 1, 3, 1, E-14, 1ERROR)

IF (1ERROR .NE. 1) GO TO 460

WRITE(6, 390) CHUTE

STOP

FORMAT(* INERTIA MATRIX OF *AB* CHUTE IS SINGULAR...RUN STOPPED*)

DO 400 I=1,3
400 IF (1WPP(I).NE.0) WPPD(I) = TEMPI(I) * DPR

DO 420 I=1,3
420 IF (1EPP(I).NE.0) EPPD(I) = TEMPI(I) * DPR

CALL LARATE (TEMPI, WPPiR, EPPiR)

DO 440 I=1,3
440 IF (1UPP(I).NE.0) UPPD(I) = FPP(I)/PMASS

DO 450 I=1,3
450 IF (1XPP(I).NE.0) XPPD(I) = APPD(I)
C *******************************************************************************
C --- LINEAR VELOCITY EQUATIONS ---
C DO 480 I=1,3
480 IF(1UPC(I).NE.0) UPCD(I) = FPC(I)/CMASS
C --- LINEAR POSITION EQUATIONS ---
C DO 490 I=1,3
490 IF(IXPC(I).NE.0) XPCD(I) = UPC(I)
C --- DURING TRIM SUBTRACT TRIM VELOCITY FROM POSITION RATES ---
C IF(INST.NE.31) GO TO 510
DO 506 I=1,3
506 IF(IXPL(I).NE.0) XPCD(I) = XPCD(I) - TRM(I)
510 CONTINUE
C RETURN
END
SUBROUTINE RL (FRS, TRS, FRA, TRA, RL, RLR, L, FRS, TRS, FRA, TRA, RL, RLR, L, FRS, TRS, FRA, TRA, RL, RLR, L
* COMMON /CTIME/ TIME
* COMMON /CICCAL/ ICAL
* COMMON /COVRLY/ INST
* COMMON /CSSFLG/ SFLG
* COMMON /CIO / IREAD, IWRITE, IDIAG

C DESIGNED BY C.L. WEST
C LAST MODIFIED - DECEMBER 6, 1980
C
C FORCES AND TORQUES ON THE VEHICLE AND SEAT FROM RAIL ELASTICITY AND
C RAIL TO BLOCK FRICTION forces
C BLOCKS STARTING AT THE BOTTOM OF THE RIGHT RAIL AND GOING UP ARE
C NUMBERED 1, 2, 3; AT THE BOTTOM OF THE LEFT RAIL AND GOING UP ARE
C NUMBERED 4, 5, 6

************** RAIL OUTPUTS ***************
C FRS(3) - X, Y, Z SEAT AXIS FORCE COMPONENTS ON THE SEAT FROM
C THE RAILS (LB)
C TRS(3) - X, Y, Z SEAT AXIS TORQUE COMPONENTS ON THE SEAT FROM
C THE RAILS (FT-LB)
C FRA(3) - X, Y, Z AIRPLANE AXIS FORCE COMPONENTS ON THE AIRPLANE
C FROM THE RAILS (LB)
C TRA(3) - X, Y, Z AIRPLANE AXIS TORQUE COMPONENTS ON THE AIRPLANE
C FROM THE RAILS (FT-LB)
C FL - STROKE FLAG (0 = GUIDED 1 = UNGUIDED)
C FTS - TRIP SWITCH CONTACT FLAG (1 = ON)
C TTS - TRIP SWITCH CONTACT TIME (SEC)
C OFF - SEAT/RAIL SEPARATION FLAG (1 = SEPARATION)
C RSA(3,3) - SEAT TO AIRPLANE DIRECTION COSINE MATRIX
C SRA(3) - X, Y, Z AIRPLANE COORDINATE SYSTEM LINEAR POSITION
C VECTOR OF THE SRP (FT)
C DIS - DISTANCE FROM THE CRITICAL POINT TO THE SEAT
C REFERENCE POINT (FT)
C TM(3) - X, Y, Z VEHICLE EARTH VELOCITY COMPONENTS TO PASS
C TO THE SEAT COMPONENT DURING TRIM (FT/SEC)

************** RAIL INPUTS ***************
C BL1(3) - X, Y, Z SEAT AXIS POSITION VECTOR OF RIGHT LOWER BLOCK (FT)
C BL2(3) - X, Y, Z SEAT AXIS POSITION VECTOR OF RIGHT MIDDLE BLOCK (FT)
C BL3(3) - X, Y, Z SEAT AXIS POSITION VECTOR OF RIGHT UPPER BLOCK (FT)
C BL4(3) - X, Y, Z SEAT AXIS POSITION VECTOR OF LEFT LOWER BLOCK (FT)
C BL5(3) - X, Y, Z SEAT AXIS POSITION VECTOR OF LEFT MIDDLE BLOCK (FT)
C BL6(3) - X, Y, Z SEAT AXIS POSITION VECTOR OF LEFT UPPER BLOCK (FT)
C UP - EJECTION DIRECTION FLAG
+1 = UPWARD WRT THE VEHICLE
-1 = DOWNWARD WRT THE VEHICLE
C RLR - RIGHT RAIL Z COORDINATE OF THE END OF THE RIGHT RAIL (FT)
C XLR(3) - X, Y, Z AIRPLANE POSITION VECTOR OF THE ORIGIN OF
C THE RIGHT RAIL COORDINATE SYSTEM (FT)
C RLL - LEFT RAIL Z COORDINATE OF THE END OF THE LEFT RAIL (FT)
XRL(3) - X, Y, Z AIRPLANE POSITION VECTOR OF THE ORIGIN OF
THE LEFT RAIL COORDINATE SYSTEM (FT)
ERL(3) - AIRPLANE TO RAILS EULER ANGLES (DEG)
SPK(2) - RAIL SPRING CONSTANT (LB/FT)
DPG(2) - RAIL DAMPING CONSTANT (LB/FT/SEC)
SBF - SLIDER BLOCK FRICTION COEFFICIENT
ZTS - RIGHT RAIL AXIS Z COORDINATE OF THE KEY BLOCK AT
TRIP SWITCH CONTACT (FT)
BTS - TRIP SWITCH KEY BLOCK NUMBER
1 = BOTTOM RIGHT BLOCK
2 = MIDDLE RIGHT BLOCK
3 = TOP RIGHT BLOCK
CPT(3) - X, Y, Z AIRPLANE POSITION VECTOR OF THE CRITICAL CLEARANCE
POINT FOR THE SEAT (FT)
SRP(3) - X, Y, Z EARTH POSITION VECTOR OF THE SEAT REFERENCE POINT (FT)
UST(3) - X, Y, Z SEAT VELOCITY VECTOR OF THE SRP (FT/SEC)
EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
WSI(3) - X, Y, Z SEAT ANGULAR VELOCITY VECTOR OF THE SRP (DEG/SEC)
XAP(3) - X, Y, Z EARTH POSITION VECTOR OF THE AIRPLANE (FT)
UAP(3) - X, Y, Z AIRPLANE VELOCITY VECTOR OF THE AIRPLANE (FT/SEC)
EAP(3) - EARTH TO AIRPLANE EULER ANGLES (DEG)
WAP(3) - X, Y, Z AIRPLANE ANGULAR VELOCITY VECTOR OF THE AIRPLANE
(DEG/SEC)

DIMENSIONS OF CALLING ARGUMENTS ........
DIMENSION FKS(3), TKS(3), FRA(3), TPA(3), DSA(3,3), SSA(3), TM(3),
* BL1(3), BL2(3), BL3(3), BL4(3), BL5(3), BL6(3), XRR(3),
* XRL(3), ERL(3), SPR(2), DPG(2), CPT(3), SRP(3), UST(3), EST(3),
* WST(3), XAP(3), UAP(3), EAP(3), WAP(3)

INTERNAL DIMENSIONS ........
DIMENSION DAR(3,3), URA(3,3), DTA(3,3), DAE(3,3), DES(3,3),
* DSR(3,3), ESR(3,3), GRS(3,3), DSR(3,3), OER(3,3),
*imir(3), WAPR(3), WAPR(3), EAPR(3),
* SRPR(3), SKPR(3), SW(0)

DIMENSION FSB1(3), FAB1(3), TSB1(4), TAB1(3),
* FS2B(3), FAB2(3), TSB2(3), TAB2(3),
* FSB3(3), FAB3(3), TSB3(3), TAB3(3),
* FSB4(3), FAB4(3), TSB4(3), TAB4(3),
* FSB5(3), FAB5(3), TSB5(3), TAB5(3),
* FSB6(3), FAB6(3), TSB6(3), TAB6(3)

DATA KPD / .01745329 /

******************************
***** INITIALIZATION *****
******************************

IF(ILLCAL.NE.1) GO TO 5

INITIALIZE VARIABLES .......
OFF = FL = FTS = TTS = 0
TM(1) = TM(2) = TM(3) = 0

407
IF(CPT(1).EQ.0.99999) CPT(1) = 0
IF(CPT(2).EQ.0.99999) CPT(2) = 0
IF(CPT(3).EQ.0.99999) CPT(3) = 0
IF(UP.EQ.0.99999) UP = 1.
IF(BTS.EQ.0.99999) BTS = 1.

CALL VECXYZ (SRA,SRP,XAP,DEA,1)
CALL VECXYZ (SRPR,SRP,SRP,DEA,1)

CALL TRANS (DRA,DER,3,3)
CALL TRANS (UAE,DEA,3,3)
CALL DIHCOS (SDE,ESTI)
CALL TRANS (SDE,DEA,3,3)
CALL MATMPY (DEA,DER,DEA,3,3)
CALL TRANS (DRE,DER,3,3)
CALL MATMPY (DRA,DER,DEA,3,3)
CALL MATMPY (DSR,DER,3,3)

CALL BLOCK (FSB,FA,TSB,TA,2B,SRPR,SRA,RLR,XRR,DL,SRP,
**Determine Seat Reference Point in the Left Rail System**

50 CALL VECXYZ (SRPRL, SRA, XRL, DAR, 1)

**Bottom Block (5)**

CALL BLOCK (FSB4, FAB4, TSB4, TAB4, DUM, SRPRL, SRA, RLL, XRL, BL4, SPR, DPG, SBF, UST, WSTIR, UAP, WAPIR, DAE, DER, DRS, DRA, DSA, DSE, DSR, UP, SW(4))

**Middle Block (6)**

CALL BLOCK (FSB5, FAB5, TSB5, TAB5, DUM, SRPRL, SRA, RLL, XRL, BL5, SPR, DPG, SBF, UST, WSTIR, UAP, WAPIR, DAE, DER, DRS, DRA, DSA, DSE, DSR, UP, SW(5))

**Upper Block (7)**

CALL BLOCK (FSB6, FAB6, TSB6, TAB6, DUM, SRPRL, SRA, RLL, XRL, BL6, SPR, DPG, SBF, UST, WSTIR, UAP, WAPIR, DAE, DER, DRS, DRA, DSA, DSE, DSR, UP, SW(6))

**Check If Blocks Are Off Rails**

10 IF(FL.EQ.1.) GO TO 70
IF(SW(2).NE.1. AND. SW(5).NE.1.) GO TO 70

**Write End of Guided Stroke on Output File**
IF(INST.EQ.26) WRITE(6,100) TIME
   100 FORMAT('SEAT/RAIL SEPARATION AT TIME = *,F7.4,* SEC*/)
   FL = 1.

************************************************************************************************************************
**** TOTAL FORCES AND MOMENTS ON THE SEAT ****
************************************************************************************************************************

DO 80 1=1,3
   FRS(I) = FSBI(I) + FSB2(I) + FSB3(I) + FSB4(I) + FSB5(I) + FSB6(I)
   TRS(I) = TSBI(I) + TSB2(I) + TSB3(I) + TSB4(I) + TSB5(I) + TSB6(I)
80 CONTINUE

TOTAL FORCES AND MOMENTS ON AIRPLANE

DO 90 1=1,3
   FRA(I) = (FAB1(I) + FAB2(I) + FAB3(I) + FAB4(I) + FAB5(I) + FAB6(I))
   * SSFLG
   TRA(I) = (TAB1(I) + TAB2(I) + TAB3(I) + TAB4(I) + TAB5(I) + TAB6(I))
   * SSFLG
90 CONTINUE

IF FOUR OUTER BLOCKS ARE OFF RAILS, SET FLAG TO BYPASS THIS COMPONENT ....

IF(SW(1) + SW(3) + SW(4) + SW(6) .EQ. 4) OFF=1.0
   IF(OFF.EQ.0) GO TO 130

WRITE SEAT/RAIL SEPARATION MESSAGE .......

IF(INST.EQ.26) WRITE(6,100) TIME
   100 FORMAT('SEAT/RAIL SEPARATION AT TIME = *,F7.4,* SEC*/)
   DO 110 1=1,3
      FRS(I)=0.
      TRS(I)=0.
      FRA(I)=0.
      TRA(I)=0.
      SRA(I) = 0.
110 CONTINUE
   DO 120 1=1,3
      DO 120 J=1,3
      OSA(I,J) = 0.
120 CONTINUE

SEND DATA TO DETERMINE TRIM EARTH VELOCITY TO SEAT ........

130 IF(INST .NE. 31) GO TO 140
   CALL MATMPI (TM,DAE,UAP,3,3,1)

140 RETURN

END
SUBROUTINE RS (FPB, TPB, FAB, TAB, TRM,  
  *  FL, XYZ, EA, XPB, UPB, EPB, WPB, XAB, UAB, EAB, WAB,  
  *  XR, XD, ER, EU)  
COMMON / CELLAL / ICLAL  
COMMON / COVRVL / INST  
COMMON / CSSFLG / SSFLG  
COMMON / CIU / IREAD, IWRITE, IDIA  

C STANDAD COMPONENT RS GENERATES THE FORCES AND TORQUES THAT  
C RESTRAINS ONE BODY TO ANOTHER (THE MAN IN THE SEAT, ETC.)  

*************** RS OUTPUTS ***************  
C FPB(3) - X, Y, Z PARENT BODY AXIS FORCE VECTOR (LB)  
C TPB(3) - X, Y, Z PARENT BODY AXIS TORQUE VECTOR (FT-LB)  
C FAB(3) - X, Y, Z ATTACHED BODY AXIS FORCE VECTOR (LB)  
C TAB(3) - X, Y, Z ATTACHED BODY AXIS TORQUE VECTOR (FT-LB)  
C TRM(3) - X, Y, Z PARENT BODY EARTH VELOCITY COMPONENTS  
  TO PASS TO THE ATTACHED BODY DURING TRIM (FT/SEC)  

*************** RS INPUTS ***************  
C FL - FLAG TO RELEASE ATTACHED BODY (1 = RELEASE)  
C XYZ(3) - X, Y, Z BODY AXIS POSITION VECTOR OF THE ATTACHED  
  BODY IN THE PARENT SYSTEM (FT)  
C EA(3) - PARENT BODY TO ATTACHED BODY EULER ANGLES (DEG)  
C XPB(3) - X, Y, Z EARTH SYSTEM POSITION VECTOR OF THE PARENT  
  BODY (FT)  
C UPB(3) - X, Y, Z PARENT BODY AXIS VELOCITY VECTOR OF THE  
  ATTACHED BODY (FT)  
C EPB(3) - EARTH TO PARENT BODY EULER ANGLES (DEG)  
C WPB(3) - X, Y, Z BODY AXIS ANGULAR VELOCITY VECTOR OF  
  THE PARENT BODY (DEG/SEC)  
C XAB(3) - X, Y, Z EARTH SYSTEM POSITION VECTOR OF THE ATTACHED  
  BODY (FT)  
C UAB(3) - X, Y, Z BODY AXIS VELOCITY VECTOR OF THE ATTACHED  
  BODY (FT/SEC)  
C EAB(3) - EARTH TO ATTACHED BODY EULER ANGLES (DEG)  
C WAB(3) - X, Y, Z BODY AXIS ANGULAR VELOCITY VECTOR OF  
  THE ATTACHED BODY (DEG/SEC)  
C XR - LINEAR SPRING CONSTANT (LB/FT)  
C AD - LINEAR DAMPING CONSTANT (LB/FT/SEC)  
C ER(3) - X, Y, Z ANGULAR SPRING CONSTANTS (FT-LB/DEG)  
C ED(3) - X, Y, Z ANGULAR DAMPING CONSTANTS (FT-LB/DEG/SEC)  

C DIMENSIONS OF CALLING ARGUMENTS .......  
C  
C DIMENSION FPB(3), TPB(3), FAB(3), TAB(3), TRM(3),  
  *  XYZ(3), EA(3), XPB(3), UPB(3), EPB(3), WPB(3),  
  *  XAB(3), UAB(3), EAB(3), WAB(3), ER(3), ED(3)  
C  
C INTERNAL DIMENSIONS .......  
C  
C DIMENSION DPB(3,3), UAB(3,3), UPBT(3,3), DPA(3,3),  
  *  X6(3), DELTA(3), SPRING(3), UXE(3), VEL(3),  
  *  DAMP(3), ANG(3), TORQUE(3), WPBE(3), WABE(3),  
  *  EPRIR(3), WPBI(3), EABIR(3), WABIR(3), DABE(3,3), UABE(3),  
  *  EAIR(3), UCEAT(3,3), UCEAT(3,3), TEMP(3)  

411
DATA RPD, DPR / .01745329, 57.29578 /

****** INITIALIZATION ******

IF(ILLCAL.NE.1) GO TO 5
TRM(1) = TRM(2) = TRM(3) = 0
ISW = 0
DO 2 I=1,3
2 EAIR(I) = EA(I) * RPD
CALL DIRCOS (DCEA, EAIR)
CALL TRANS (DCEAT, DCEA, 3, 3)

BYPASS CALCULATIONS IF THE FLAG IS SET TO RELEASE ........

5 IF(ISW.E..) GO TO 140
IF(FL.NE.1) GO TO 20
DO 10 I=1,3
FPB(I) = 0.
FAB(I) = 0.
TPB(I) = 0.
10 TAB(I) = 0.
ISW = 1
GO TO 140

****** CHANGE FROM DEGREES TO RADIANS ******

DO 30 I=1,3
EPBIR(I) = EPB(I) * RPD
WPBIR(I) = WPB(I) * RPD
EABIR(I) = EAB(I) * RPD
30 WABIR(I) = WAB(I) * RPD

CALCULATE THE DIRECTION COSINE MATRICES ........

CALL DIRCOS (DPB, EPBIR)
CALL DIRCOS (DAB, EABIR)
CALL TRANS (DAB, DAB, 3, 3)
CALL TRANS (DPBT, DPB, 3, 3)
CALL MATMPY (DPB, DAB, DPBT, 3, 3, 3)

FORCES AND TURNOVES DUE TO LINEAR DISPLACEMENT ******

CALCULATE THE ATTACHED BODY LINEAR POSITION VECTOR IN THE
PARENT BODY COORDINATE SYSTEM (XB) ........

CALL VECXYZ (XB, XAB, RPB, DPB, 1)

DETERMINE THE LINEAR DISPLACEMENT FROM THE ATTACHMENT POINT,
AND CALCULATE THE BODY AXIS SPRING COMPONENTS ACTING ON THE
PARENT BODY ........

DO 40 I=1,3
DELTA(I) = XB(I) - XYZ(I)
412
40 SPRING(I) = DELTA(I) * XR
C
C CALCULATE THE Buoy AXIS DAMPING COMPONENTS ACTING ON THE PARENT
C Buoy, AND SUM THE RESULTS WITH THE SPRING COMPONENTS....... 
C
CALL VELXZ (UXE,UPO,XB,WPBIR,DPBT)
CALL MATMPY (UABE,UABE,UAB,3,3,1)
DO 50 I=1,3
50 DELTA(I) = UABE(I) - UXE(I)
CALL MATMPY (VEL,UPO,DELTA,3,3,1)
DO 60 I=1,3
DAMP(I) = VEL(I) * XD
60 FPB(I) = SPRING(I) + DAMP(I)
C
CALL TORSION ON PARENT BODY DUE TO DISPLACEMENT OF ATTACHMENT
C POINT FROM PARENT BODY CENTER OF GRAVITY
C
CALL CRSROD (TORQUE,XYZ,FPB)
C
C CALCULATE THE BODY AXIS FORCE COMPONENTS ACTING ON THE
C ATTACHED BODY........
C
CALL MATMPY (FAB,UPA,FPB,3,3,1)
DO 70 I=1,3
70 FAB(I) = -FAB(I)
C
***********************************************************************
C ***** TORQUE DUE I) ANGULAR DISPLACEMENT *****
C ***********************************************************************
C
C CALCULATE THE PARENT TO ATTACHED BODY EULER ANGLES...........
C
CALL COSDIR (ANG,OPA)
C
C DETERMINE THE ANGULAR DISPLACEMENT FROM THE ATTACHMENT ANGLE,
C AND CALCULATE THE SPRING COMPONENTS ACTING ON THE SEAT IN THE
C ATTACHMENT AXIS SYSTEM........
C
DO 90 I=1,3
DELTA(I) = ANG(I-1)*DPK - EA(I-1)
90 SPRING(I) = DELTA(I) * ER(I)
C
C CALCULATE THE BODY AXIS ANGULAR DAMPING COMPONENTS ACTING ON
C THE PARENT BODY IN THE ATTACHMENT AXIS SYSTEM........
C
CALL MATMPY (WPBE,OPBT,WPB,3,3,1)
CALL MATMPY (WABE,UABE,WAB,3,3,1)
DO 90 I=1,3
90 DELTA(I) = WABE(I) - WPBE(I)
CALL MATMPY (TEMP,UPA,DELTA,3,3,1)
CALL MATMPY (VEL,DECA,TEMP,3,3,1)
DO 100 I=1,3
DAMP(I) = VEL(I) * ED(I)
100 TEMP(I) = SPRING(I) + DAMP(I)
C
C MOVE THE RESTRAINT TORQUES INTO THE SEAT SYSTEM........
C
CALL MATMPY (TPB,DCGEAT,TEMP,3,3,1)

413
C CALCULATE THE BODY AXIS TORQUE COMPONENTS ACTING ON THE
C ATTACHED BODY ......
C
CALL MATMPY (TAB, DPA, TPB, 3, 3, 1)
DO 110 I = 1, 3
110 TAB(I) = -TAB(I)
C
C CALCULATE THE TOTAL MOMENT ON THE PARENT BODY ......
C
DO 120 I = 1, 3
120 TPb(I) = TPb(I) + TORQUE(I)
C
C ZERO THE FORCES AND TORQUES ACTING ON THE PARENT BODY IF SSFLG IS
C EQUAL TO ZERO ......
C
IF (SSFLG .NE. 0) GO TO 135
DO 130 I = 1, 3
130 FPb(I) = TPb(I) = 0
C
C SEND DATA TO ATTACHED BODY TO ALLOW IT TO COMPUTE THE PARENT BODY
C EARTH VELOCITY DURING TRIM ......
C
135 IF (INST .NE. 31) GO TO 140
CALL MATMPY (TRM, DPBT, UB, 3, 3, 1)
C
140 RETURN
END
SUBROUTINE SE (UST, UOS, IUS, SRP, XOS, IXS, WST, WOS, IWS,
* EST, EDS, IES, SC0, SCDDOT, ISC0, SC, SCU0T, ISC,
* Gx, Gy, Gz, UR, ALT,
* F11, F12, F13, F14, F15, F16, F17, F18,
* T11, T12, T13, T14, T15, T16, T17, T18,
* F21, F22, F23, F24, F25, F26, F27, F28,
* T21, T22, T23, T24, T25, T26, T27, T28,
* LW, CGG, CHI, CPG, TM)

EASIEST SEAT EQUATIONS OF MOTION COMPONENT

DESIGNED BY C.L. WEST
LAST MODIFIED - DECEMBER 6, 1980

*************** SE OUTPUTS ***************

LINEAR VELOCITIES - BOGY AXIS
UST(3) - X, Y, Z LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT (FT/SEC)
WST(3) - X, Y, Z LINEAR ACCELERATION VECTOR OF THE SEAT REFERENCE POINT (FT/SEC/SEC)
IWS(3) - INTEGRATION CONTROL

LINEAR POSITIONS - EARTH SYSTEM
SRP(3) - X, Y, Z LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT (FT)
XOS(3) - X, Y, Z LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT (FT/SEC)
IXS(3) - INTEGRATION CONTROL

ANGULAR VELOCITIES - BOGY AXIS
WSI(3) - X, Y, Z ANGULAR VELOCITY COMPONENTS - P, Q, R (DEG/SEC)
WOS(3) - X, Y, Z ANGULAR ACCELERATION COMPONENTS (DEG/SEC/SEC)
IWS(3) - INTEGRATION CONTROL

EULER ANGLES -- EARTH TO BOGY -- YAW, PITCH, ROLL
EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
EDS(3) - EULER ANGLE RATES (DEG/SEC)
IES(3) - INTEGRATION CONTROL

SPINAL COMPRESSION VELOCITY
SCU - SPINAL COMPRESSION VELOCITY (FT/SEC)
SCU0T - SPINAL COMPRESSION VELOCITY RATE (FT/SEC/SEC)
ISCU - INTEGRATION CONTROL

SPINAL COMPRESSION
SC - SPINAL COMPRESSION (FT)
SCD0T - SPINAL COMPRESSION RATE (FT/SEC)
ISC - INTEGRATION CONTROL

415
C GX - SEAT X-AXIS LOAD FACTOR (G)
C GY - SEAT Y-AXIS LOAD FACTOR (G)
C GZ - SEAT Z-AXIS LOAD FACTOR (G)
C OR - DYNAMIC RESPONSE
C ALT - SEAT ALTITUDE (FT)

****************** SE INPUTS ******************
F11(3) THROUGH F16(3) - SEAT AXIS FORCE VECTORS ACTING ON THE
EJECTION SEAT WHICH ARE GENERATED BY
AN EXPLOSIVE CHARGE (LB)
T11(3) THROUGH T18(3) - SEAT AXIS TORQUE VECTORS ACTING ON THE
EJECTION SEAT WHICH ARE GENERATED BY
AN EXPLOSIVE CHARGE (FT-LB)
F21(3) THROUGH F28(3) - SEAT AXIS FORCE VECTORS ACTING ON THE
EJECTION SEAT WHICH ARE GENERATED BY
NON-EXPLOSIVE MEANS (LB)
T21(3) THROUGH T28(3) - SEAT AXIS TORQUE VECTORS ACTING ON THE
EJECTION SEAT WHICH ARE GENERATED BY
NON-EXPLOSIVE MEANS (FT-LB)
CW - COMPOSITE SEAT WEIGHT (LB)
CCLG(3) - X, Y, Z SEAT AXIS SYSTEM COMPOSITE CENTER OF GRAVITY (FT)
CM1L(3) - COMPOSITE SEAT MOMENT OF INERTIA VECTOR ABOUT ITS
CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)
CP1L(3) - COMPOSITE SEAT PRODUCT OF INERTIA VECTOR ABOUT ITS
CENTER OF GRAVITY - IXY, IXZ, IYZ (SLUG-FT**2)
TM(3) - X, Y, Z VEHICLE EARTH VELOCITY COMPONENTS TO
DETERMINE THE POSITION RATES DURING TRIM (FT/SEC)

DIMENSIONS OF CALLING ARGUMENTS

DIMENSION UST(3), UDS(3), IUS(3), SRP(3), AUS(3), IXS(3),
* WST(3), WDS(3), IWS(3), EST(3), EU(3), IES(3),
* CCG(3), CM1L(3), CP1L(3), TM(3)

* T21(3), T22(3), T23(3), T24(3), T25(3), T26(3), T27(3), T28(3)

INTERNAL DIMENSIONS

DIMENSION TINER(3,3), TEMP1(3), TEMP2(3), TEMP3(3), TEMP4(3), DSE(3,3),
* F(3), T(3), WSTIR(3), WDSIR(3), ESTIR(3), DES(3,3), FG(3), TG(3)

COMMON /ICCAL/ ICCAL
COMMON /COVRLY/ INSI
COMMON /CSSFLG/ SSFLG
COMMON /C10/ IREAD, IWRITE, IDIAG

DATA RPDP, DPR / .01745329, .57.29578 /
DATA GRAV /3.01747/
DO 10 I=1,3
IF(F1(I) .EQ. 0.99999) F1(I) = 0
IF(F1(I) .EQ. 0.99999) F2(I) = 0
IF(F1(I) .EQ. 0.99999) F3(I) = 0
IF(F1(I) .EQ. 0.99999) F4(I) = 0
IF(F1(I) .EQ. 0.99999) F5(I) = 0
IF(F1(I) .EQ. 0.99999) F6(I) = 0
IF(F1(I) .EQ. 0.99999) F7(I) = 0
IF(F1(I) .EQ. 0.99999) F8(I) = 0
IF(F1(I) .EQ. 0.99999) F9(I) = 0
IF(F1(I) .EQ. 0.99999) FA(I) = 0
IF(F1(I) .EQ. 0.99999) FB(I) = 0
IF(F1(I) .EQ. 0.99999) FC(I) = 0
IF(F1(I) .EQ. 0.99999) FD(I) = 0
IF(F1(I) .EQ. 0.99999) FE(I) = 0
IF(F1(I) .EQ. 0.99999) FF(I) = 0
IF(F1(I) .EQ. 0.99999) FG(I) = 0
IF(F1(I) .EQ. 0.99999) FH(I) = 0
IF(F1(I) .EQ. 0.99999) FI(I) = 0
IF(F1(I) .EQ. 0.99999) FJ(I) = 0
IF(F1(I) .EQ. 0.99999) FK(I) = 0
IF(F1(I) .EQ. 0.99999) FL(I) = 0
IF(F1(I) .EQ. 0.99999) FM(I) = 0
IF(F1(I) .EQ. 0.99999) FN(I) = 0
IF(F1(I) .EQ. 0.99999) FO(I) = 0
IF(F1(I) .EQ. 0.99999) FP(I) = 0
IF(F1(I) .EQ. 0.99999) FQ(I) = 0
IF(F1(I) .EQ. 0.99999) FR(I) = 0
IF(F1(I) .EQ. 0.99999) FS(I) = 0
IF(F1(I) .EQ. 0.99999) FT(I) = 0
IF(F1(I) .EQ. 0.99999) FU(I) = 0
IF(F1(I) .EQ. 0.99999) FV(I) = 0
IF(F1(I) .EQ. 0.99999) FW(I) = 0
IF(F1(I) .EQ. 0.99999) FX(I) = 0
IF(F1(I) .EQ. 0.99999) FY(I) = 0
IF(F1(I) .EQ. 0.99999) FZ(I) = 0
 IF(F1(I) .EQ. 0.99999) CONTINUE
TM(I) = TM(I) = TM(I) = 0

C ***** CHANGE FROM DEGREES TO RADIANS *****
C DO 30 I=1,3
WST(I) = WST(I) * RPD
EST(I) = EST(I) * RPD
C
C ***** SET UP SEAT INERTIA TENSOR *****
C
TINER(I,1) = CM(I)
TINER(I,2) = -CPI(I)
TINER(I,3) = -CPI(I)
TINER(2,1) = -CPI(I)
TINER(2,2) = CM(I)
TINER(2,3) = -CPI(I)
TINER(3,1) = -CPI(I)
TINER(3,2) = -CPI(I)
TINER(3,3) = CM(I)
C
C CALCULATE THE DIRECTION COSINE MATRICES ......
C
CALL DIRCOS (DES, ESTK)
C
417
CALL TRANS (DESE,DE5,3,3)

****** COMPUTE GRAVITY FORCES AND TORQUES ******

DO 40 I=1,3
  40 FG(I) = CM * DE5(I,3) * SSFLG
CALL CRSPRD (TG,CC6,F6)

****** SUM FORCES AND MOMENTS ******

DO 50 I=1,3
  F(I) = F11(I) + F12(I) + F13(I) + F14(I) + F15(I) + F16(I) +
   * F17(I) + F18(I) + F21(I) + F22(I) + F23(I) + F24(I) +
   * F25(I) + F26(I) + F27(I) + F28(I) + FG(I)
  T(I) = T11(I) + T12(I) + T13(I) + T14(I) + T15(I) + T16(I) +
   * T17(I) + T18(I) + T21(I) + T22(I) + T23(I) + T24(I) +
   * T25(I) + T26(I) + T27(I) + T28(I) + TG(I)
50 CONTINUE

C CALCULATE THE SEAT ALTITUDE .......
C ALT = -SRP(3)

C CALCULATE THE DYNAMIC RESPONSE .......
C DR = SC * .86.977

C ******************************************************
C ***** ANGULAR VELOCITY EQUATIONS *****
C ******************************************************

C CALCULATE TINER * WSTIR
C CALL MATMPY (TEMP1,TINER,WSTIR,3,3,1)
C CALCULATE WSTIR X (TINER * WSTIR)
C CALL CRSPRD (TEMP2,WSTIR,TEMP1)
C COMPUTE CCG X F...
C CALL CRSPRD (TEMP3,CCG,F)
C SUM TERMS TO OBTAIN TOTAL TORQUE .......

DO 60 I=1,3
  60 TEMP3(I) = T(I) - TEMP2(I) - TEMP3(I)
C CALCULATE WDS .......
C CALL LUEWS (TINER,TEMP1,TEMP3,TEMP2,3,1,3,3,3,1.E-14,IERRO)
IF (IERRO.NE.1) GO TO 80
WRITE(6,70)
70 FORMAT(* INERTIA MATRIX OF SEAT IS SINGULAR...RUN STOPPED*)
STOP
CONTINUE

DO 90 I=1,3
90 CONTINUE

418
IF(IS(1).NE.0) WSTIR(1) = TEMP(1)
  
WSTIR = WSTIR * DPR

C *******************************************************
C ***** EULER ANGLE EQUATIONS *****
C *******************************************************

CALL EARATE (TEMP1, WSTIR, ESTIR)
DO 100 I=1,3
100 IF(IES(1).NE.0) EDS(I) = TEMP1(I) * DPR

C *******************************************************
C ***** LINEAR VELOCITY EQUATIONS *****
C *******************************************************

CALL LRATE (WSTIR, xCG)
CALL CrSPRD (TEMP1, WSTIR, xCG)
CALCULATE WSTIR X xCG

CALL CrSPRD (TEMP2, WSTIR, xCG)
CALL CrSPRD (TEMP3, WSTIR, xCG)
CALCULATE WSTIR X (WSTIR X xCG)

CALL CrSPRD (TEMP1, WSTIR, TEMP2)
CALL CrSPRD (TEMP2, WSTIR, TEMP3)
CALL CrSPRD (TEMP1, WSTIR, UST)
CALL CrSPRD (TEMP2, WSTIR, UST)
CALL CrSPRD (TEMP3, WSTIR, UST)

CALL F/M

LMASS = CW/Grav
DO 120 I=1,3
120 TEMP4(I) = F(I)/LMASS

SUM THE ACCELERATION COMPONENTS

DO 130 I=1,3
130 IF(IS(1).NE.0) UDS(I) = TEMP4(I) - TEMP1(I) - TEMP2(I) - TEMP3(I)

C ========= DETERMINE THE LOAD FACTORS =========

GX = (TEMP1(1) + TEMP3(1) - TEMP4(1))/Grav
GY = (TEMP1(2) + TEMP3(2) - TEMP4(2))/Grav
GZ = (TEMP1(3) + TEMP3(3) - TEMP4(3))/Grav

C *******************************************************
C ***** LINEAR POSITION EQUATIONS *****
C *******************************************************

CALL MATMPY (TEMP1, GSE, UST, 3, 3, 1)
DO 140 I=1,3
140 IF(IS(1).NE.0) XDS(1) = TEMP1(I)

C *******************************************************
C ***** SPINAL COMPRESSION EQUATIONS *****

419
C  ********************************************************************
C  SPINAL COMPRESSION VELOCITY EQUATION ......  
C     IF(ISC.NE.0) SCVDOT = -23.6992 * SCQ - 2796.41 * SC
C             + GRAV * GZ
C  SPINAL COMPRESSION EQUATION ......  
C     IF(ISC.NE.0) SCQDOT = SCQ
C  DURING TRIM, SUBTRACT TRIM VELOCITY FROM POSITION RATES  
C     IF (INST.NE.31) GO TO 160
C     DU 160  I=1,3
C    160  IF(IXS(I).NE.6) XDS(I)=XDS(I)-TM(I)
C
C  RETURN
C    END
SUBROUTINE SL (USL,USLD,IUSL,XSL,XSLD,IXSL,WSL,WSLD,IWSL,
   ESL,ESLD,IESL,
   UD,WU)

IMPLICIT USL(3),USLD(3),IUSL(3),XSL(3),XSLD(3),IXSL(3),
   WSL(3),WSLD(3),IWSL(3),ESL(3),ESLD(3),IESL(3),
   UD(3),WU(3)

IMPLICIT TEMP(3),WSLIR(3),IWSLIR(3),DES(3,3),USE(3,3)

COMMON /CICCAL/ LICAL
COMMON /CJ4/ IREAD,WRITE,DIAG

DATA RPU,DPR / .01745329, 57.29578 /

* DESIGNED BY C.L. WEST
* LAST MODIFIED - DECEMBER 6, 1969

EASIEST SIX DEGREE OF FREEDOM SLED MODEL

============ SLED OUTPUTS ============

LINEAR VELOCITIES - BODY AXIS

USL(3) - X,Y,Z LINEAR VELOCITY VECTOR (FT/SEC)
USLD(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR (FT/SEC/SEC)
IUSL(3) - INTEGRATION CONTROL

LINEAR POSITIONS - EARTH SYSTEM

XSL(3) - X,Y,Z LINEAR POSITION VECTOR (FT)
XSLD(3) - X,Y,Z LINEAR POSITION RATE VECTOR (FT/SEC)
IXSL(3) - INTEGRATION CONTROL

ANGULAR VELOCITIES - BODY AXIS

WSL(3) - X,Y,Z ANGULAR VELOCITY VECTOR P,Q,R (DEG/SEC)
WSLD(3) - X,Y,Z ANGULAR VELOCITY RATE VECTOR (DEG/SEC/SEC)
IWSL(3) - INTEGRATION CONTROL

EULER ANGLES -- EARTH TO BODY -- YAW,PITCH,ROLL

ESL(3) - EARTH TO SLED EULER ANGLES (DEG)
ESLD(3) - EULER ANGLE RATES (DEG/SEC)
I ESL(3) - INTEGRATION CONTROL

============ SLED INPUTS ============

UD(3) - X,Y,Z SLED SYSTEM LINEAR VELOCITY RATE VECTOR (FT/SEC/SEC)
WU(3) - X,Y,Z SLED SYSTEM ANGULAR VELOCITY RATE VECTOR (DEG/SEC/SEC)


#########################################################################

**** INITIALIZATION ****

#########################################################################

IF(ICAL.NE.1) GO TO 20

421
C
DO 5 I=1,3
  IF(WD(I) .EQ. 0.99999) WD(I) = 0
  IF(WI(I) .EQ. 0.99999) WI(I) = 0

C
IF(WSL(I)+WSL(2)+WSL(3).EQ.0) GO TO 20

C
WRITE(6,10)
10  FORMAT(5X,"SLED ANGULAR VELOCITY IS NOT INITIALIZED AT ZERO *
      * ------ RUN STOPPED ------*/

C

CHANGE FROM DEGREES TO RADIANS ......

20  DO 30 I=1,3
    WSL(I) = WSL(I) * RPD
  30  ESLR(I) = ESL(I) * RPD

C
**************
C ***** ANGULAR EQUATIONS *****
C ***********************
C ANGULAR VELOCITY EQUATIONS ......
C
DO 40 I=1,3
40  IF(IWSL(I).NE.0) WSLD(I) = WD(I)

C
EULER ANGLE RATES ......

CALL EARATE (TEMP,WSLR,ESLR)
DO 50 I=1,3
50  IF(IESL(I).NE.0) ESLD(I) = TEMP(I) * DPR

C
**************
C ***** LINEAR EQUATIONS *****
C ***********************
C LINEAR VELOCITY EQUATIONS ......
C
CALL CRSRD (TEMP,WSLR,USL)
DO 60 I=1,3
60  IF(IUSL(I).NE.0) USLD(I) = UD(I) - TEMP(I)

C
LINEAR POSITION EQUATIONS ......
C
CALL GIRCOS (DES,ESLR)
CALL TRANS (DES,DES,3,3)
CALL MAMPY (TEMP,DES,USL,3,3,1)
DO 70 I=1,3
70  IF(IUSL(I).NE.0) XSLD(I) = TEMP(I)

C
RETURN
END
SUBROUTINE SP (TRF, TMA, TST,
   *   WG, WGD, IMG, ESG, ESGD, IESG, ESR, ESRD, IESR, PHA,
   *   FT, TIN, ECA,
   *   FL, YPR, AVW, MRI, RII, RIF, XR, UV,
   *   GSA, GSP, DPG, FMT, TMX, TIN, TOS, TSU, GMA, WST)

STANDARD COMPONENT SP CALCULATES FORCES AND TORQUES APPLIED
TO THE SEAT BY THE STAPAC STABILIZATION SYSTEM

*************** SP TABLES ***************

TRF - STAPAC ROCKET THRUST TABLE

   THE INDEPENDENT VARIABLE IS TIME (SEC)
   THE DEPENDENT VARIABLE IS ROCKET FORCE (LB)

TMA - MECHANICAL ADVANTAGE TABLE

   THE INDEPENDENT VARIABLE IS THE GIMBAL ANGLE (DEG)
   WITH RESPECT TO THE CAGED POSITION
   THE DEPENDENT VARIABLE IS THE MECHANICAL ADVANTAGE

TST - SPRING TORQUE TABLE

   THE INDEPENDENT VARIABLE IS THE GIMBAL ANGLE (DEG)
   WITH RESPECT TO THE CAGED POSITION
   THE DEPENDENT VARIABLE IS THE SPRING TORQUE (FT-Lb)

*************** SP OUTPUTS ***************

ANGULAR VELOCITY -- GIMBAL X-AXIS
(LESS THE SEAT ANGULAR VELOCITY PROJECTED UNTO THE GIMBAL X-AXIS)

   WG - ANGULAR VELOCITY (DEG/SEC)
   WGD - ANGULAR ACCELERATION (DEG/SEC/SEC)
   IMG - INTEGRATION CONTROL

EULER ANGLES -- SEAT TO GIMBAL -- YAW, PITCH, ROLL

   ESG(3) - SEAT TO GIMBAL Euler ANGLES (DEG)
   ESGD(3) - Euler ANGLE RATES (DEG/SEC)
   IESG(3) - INTEGRATION CONTROL

EULER ANGLES -- SEAT TO ROCKET -- YAW, PITCH, ROLL

   ESR(3) - ANGULAR POSITION (DEG)
   ESRD(3) - ANGULAR VELOCITY (DEG/SEC)
   IESR(3) - INTEGRATION CONTROL

PHA - STAPAC OPERATIONAL PHASE

   0 = BEFORE IGNITION
   1 = STAPAC IGNITION
   2 = STAPAC BURNOUT

F(3) - X, Y, Z SEAT BOY AXIS FORCE COMPONENTS (LB)

T(3) - X, Y, Z SEAT BOY AXIS TORQUE COMPONENTS (FT-Lb)

TIN - TIME AT STAPAC INITIATION (SEC)

ECA - SEAT TO GIMBAL ROLL EULER ANGLE AT THE CAGED POSITION (DEG)
*************** SP INPUTS ***************

FL - STAPAC IGNITION FLAG (1 = STAPAC ON)
YPK - STAPAC APPLICATION FLAG
   1 = YAW STAPAC
   2 = PITCH STAPAC
   3 = ROLL STAPAC
AVM - ANGULAR VELOCITY OF THE GYROSCOPE WHEEL (DEG/SEC)
WM1 - MOMENT OF INERTIA OF THE WHEEL ABOUT ITS
   SPIN AXIS (SLUG-FT**2)
SM1 - MOMENT OF INERTIA OF THE SYSTEM LESS ROCKET ABOUT
   THE GIMBAL AXIS (SLUG-FT**2)
R11 - MOMENT OF INERTIA OF THE ROCKET PRIOR TO
   IGNITION (SLUG-FT**2)
RIF - MOMENT OF INERTIA OF THE ROCKET AFTER
   BURNOUT (SLUG-FT**2)
XR(3) - X, Y, Z SEAT BODY AXIS POSITION VECTOR OF THE
   ROCKET NOZZLE (FT)
UV(3) - X, Y, Z ROCKET FORCE UNIT VECTOR IN THE ROCKET COORDINATE
   SYSTEM
GSA - GIMBAL MOTION STOP IN THE NEGATIVE ROLL DIRECTION WITH RESPECT
   TO THE CAGE POSITION (DEG)
GSF - GIMBAL MOTION STOP IN THE POSITIVE ROLL DIRECTION WITH RESPECT
   TO THE CAGE POSITION (DEG)
SPR - GIMBAL STOP ANGULAR RIGIDITY (FT-LB/DEG)
DPG - GIMBAL STOP ANGULAR DAMPING (FT-LB/DEG/SEC)
FMI - LOAD AT MAXIMUM FRICTION (LB)
TMX - MAX FRICTION (FT-LB)
TNF - FRICTION AT NO THRUST (FT-LB)
TOS - THRUSTLINE OFFSET (LB)
TSU - GYROSCOPE WHEEL SPINUP TIME (SEC)
GMA - GIMBAL ANGULAR VELOCITY AT MAXIMUM FRICTION (DEG/SEC)
WS1(3) - X, Y, Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF
   THE SEAT (DEG/SEC)

*******************************************************************************

--- DIMENSIONS OF CALLING ARGUMENTS ----

DIMENSION TRF(5), TMA(5), TST(5), ESG(3), ESGD(3), IESG(3),
   
   ESR(3), ESAD(3), IESR(3), IESG(3), F(3), T(3), XR(3), UV(3),
   
   WST(3)

--- INTERNAL DIMENSIONS ----

DIMENSION WGB(3), WRB(3), FRAT(3), WSTG(3), TEMP(3), DSR(3), DRS(3),
   
   DSG(3), ESRIR(3), ESGIR(3)

COMMON /CICAL/ IICAL
COMMON /CTIME/ TIME
COMMON /CIO/ IREAD, IWRITE, LDIAG

DATA RPD, DPR / .01745329, 57.29578 /
DATA PI / 3.14159 /
DATA WGB(2), WGB(3) / 0, 0, WRB(1), WRB(3) / 0, 0 /

*******************************************************************************

**** INITIALIZATION ****

424
C ********************
C IF(ICAL.NE.1) GO TO 20
C
C      PHA = TIN = 0
C      ELA = ESG(3)
C      DO 10 I=1,3
C
C      F(1) = 0.
C 10   T(I) = 0.
C      IF(TSU.EQ.0.99999) TSU = 0.005
C      IF(GMA.EQ.0.99999) GMA = 10.
C      IF(UV(1).EQ.0.99999) UV(1) = 0
C      IF(UV(2).EQ.0.99999) UV(2) = 0
C      IF(UV(3).EQ.0.99999) UV(3) = -1.
C
C 1000
C
C 30 FORMAT(5X,YAW STAPAC IGNITION AT TIME=*,F10.4,ZX9*SEC*)
C 40 FORMAT(5X,PITCH STAPAC IGNITION AT TIME=*,F10.4,ZX9*SEC*)
C 50 FORMAT(5X,ROLL STAPAC IGNITION AT TIME=*,F10.4,ZX9*SEC*)
C 60 TIN = TIME
C      PHA = 1.
C 90 CONTINUE
C
C ------- CHANGE FROM DEGREES TO RADIANS -------
C
C      DO 100 I=1,3
C 100  ESGIR(I) = ESG(I) * RPD
C      ESR1R(I) = ESR(I) * RPD
C
C ------- COMPUTE THE SEAT TO GIMBAL DIRECTION COSINE MATRIX -------
C
C CALL DIRLOS (DSG,ESGIR)
C
C ------- CALCULATE THE SEAT ANGULAR VELOCITY IN THE GIMBAL SYSTEM -------
C
C CALL MATMPY (WSTG,DSG,WST,3,3,1)
C
C ------- DETERMINE THE TIME INTO STAPAC -------
C
C TIS = TIME - TIN
C
C ------- DETERMINE THE ROCKET THRUST -------
NRT = TRF(2)
IF(TIS.GT.TRF(NRT+3)) GO TO 190
FR = TRF1(TIS,TRF(4),TRF(NRT+4),1,-NRT)

-- DETERMINE THE MECHANICAL ADVANTAGE --
DELTA = ESG(3) - ECA
NMA = TMA(2)
SMA = TRF1(Delta,TMA(4),TMA(NMA+4),1,-NMA)

-- CALCULATE THE SYSTEM INERTIA --
SYSMI = SMI + SMA**2*(RI1-(TIS-TRF(4))/(TRF(NRT+3)-TRF(4))
        *RI1-RI1))

-------------------------
***** DETERMINE THE GIMBAL X-AXIS TORQUE *****
-------------------------

-- CALCULATE THE THRUSTLINE OFFSET TORQUE --
TOFF = FR * TOS * SMA

-- CALCULATE THE FRICTIONAL TORQUE --
ANGV = (WG - WSTG(1))/GMA
TRICT = -SIGN(AMIN11,,ABS(ANGV)),ANGV) * ABS(SMA) *
        (TNF+FR/FMT*(TMX-TNF))

-- CALCULATE THE PRECESSIONAL TORQUE --
AVWIR = AVW * RPD
IF(TIS.LE.TSU) AVWIR = (1.+SIN(3.*PI/2.+TIS/TSU*PI))/2.*AVW*RPD
IF(TIS.LE.0) AVWIR = 0
TPREC = -WHI * AVWIR * WSTG(2) * RPD

-- DETERMINE THE SPRING TORQUE --
NST = TSI(2)
TSPR = TRF1(Delta,TST(4),TST(NST+4),1,-NST)

-- CALCULATE THE GYMBAL STOP TORQUE --
IF(DELTA.LT.GSA) GO TO 110
IF(DELTA.GT.GSA) GO TO 120
GO TO 140

CALCULATE SPRING TORQUE .......
110 TSTOP = SPR * (GSA - DELTA)
GO TO 130
120 TSTOP = SPRK * (GSK - DELTA)

CALCULATE DAMPING TORQUE .......
130 TSTOP = TSTOP - ANGV * DPG
GO TO 150
C
C SET SPRING AND DAMPING TORQUES EQUAL TO ZERO ....
C
140 TSTOP = 0.
C
C ------- SUM THE TORQUES -------
C
150 TSUM = TUFF + TFRICT + TREC + TSPR + TSTOP
C
***********************************************************************
C ***** CALCULATE THE RATES *****
C ***********************************************************************
C
C ------- CALCULATE THE GIMBAL X-AXIS ANGULAR VELOCITY RATE -------
C
IF(WG.NE.0) WGD = (TSUM/SYSMI) * DPR
C
C ------- DETERMINE THE GIMBAL EULER ANGLE RATES -------
C
WGB(I) = WG - WSTG(I)
CALL CARATE (TEM,WGB,ESGR)
DO 160 I=1,3
160 IF(I.EQG(I).NE.0) ESGD(I) = TEMP(I)
C
C ------- COMPUTE THE ROCKET EULER ANGULAR RATES -------
C
WRB(2) = WGB(I) * SMA
CALL CARATE (TEM,WRB,ESRI)
DO 170 I=1,3
170 IF(I.EQS(I).NE.0) ESRO(I) = TEMP(I)
C
***********************************************************************
C ***** CALCULATE THE ROCKET FORCES AND TORQUES *****
C ***********************************************************************
C
C ------- TRANSFORM THE ROCKET THRUST TO THE SEAT -------
C
CALL DIRCOS (DSR,ESRIK)
CALL TRANS (DKS,DSK,3,3)
DO 180 I=1,3
180 FRAT(I) = UV(I) * FR
CALL MATMPY (F,FRAT,FKKT,3,3,1)
C
C ------- COMPUTE THE SEAT BODY AXIS TORQUE COMPONENTS -------
C
CALL CRSRLO (T,XR,F)
GO TO 260
C
***********************************************************************
C ***** WHEN THE ROCKET SHOOTS DOWN .... *****
C ***********************************************************************
C
C ------- ZERO OUT RATES, FORCES, AND TORQUES -------
C
190 DO 200 I=1,3
ESGD(I) = 0.
ESRO(I) = 0.

427
F(1) = 0.
T(I) = 0.
WGO = 0.
PHA = 2.

C ---- WRITE BURNOUT MESSAGE ----
C
IF(YPR.EQ.2.) GO TO 220
IF(YPR.EQ.3.) GO TO 240
WRITE (6,210) TIME
210 FORMAT(/5X,*YAW STATPAC BURNOUT AT TIME=*,F10.4,2X,*SEC*)
GO TO 260
220 WRITE(b,230) TIME
230 FORMAT(/5X,*PITCH STATPAC BURNOUT AT TIME=*,F10.4,2X,*SEC*)
GO TO 260
240 WRITE(b,250) TIME
250 FORMAT(/5X,*KOLL STATPAC BURNOUT AT TIME=*,F10.4,2X,*SEC*)
C
260 RETURN
END
SUBROUTINE SR (TRF,

PW,PW0,IPW,

PHA,PHN,FST,IST,XCG,PMI,PPI,FR,PWI,SPI,RHO,

WIN,TM1,TIG,

FON,FPS,EXN,YAW,PIT,PL,POD,PLD)

FORCES AND MOMENTS ACTING ON THE SEAT FROM THE SUSTAINER ROCKET

DESIGNED BY C.L. WEST

LAST MODIFIED - DECEMBER 6, 1980

*************** ROCKET TABLES ***************

TRF - ROCKET THRUST TABLE

THE INDEPENDENT VARIABLE IS TIME (SEC)

THE DEPENDENT VARIABLE IS THE ROCKET FORCE (LB)

*************** ROCKET OUTPUTS ***************

PW - WEIGHT OF UNBURNED PROPELLANT (LB)
PMD - PROPELLANT BURN RATE (LB/SEC)

IPW - INTEGRATION CONTROL

PHA - ROCKET PHASE

0 = BEFORE IGNITION
1 = ROCKET BURN
2 = ROCKET OFF

RUN - ROCKET ON FLAG (1=ON 0=OFF)

FST(3) - X,Y,Z SEAT SYSTEM ROCKET FORCE COMPONENTS (LB)

IST(3) - X,Y,Z SEAT SYSTEM ROCKET TORQUE COMPONENTS (FT-LB)

XCG(3) - X,Y,Z SEAT SYSTEM POSITION VECTOR OF THE

PROPELLANT CENTER OF GRAVITY (FT)

PMI(3) - PROPELLANT MOMENTS OF INERTIA - IXX,IYY,IZZ (SLUG-FT**2)

PPI(3) - PROPELLANT PRODUCTS OF INERTIA - IXY,IXZ,IYZ (SLUG-FT**2)

FR - SUSTAINER ROCKET FORCE MAGNITUDE (LB)
PWI - INITIAL WEIGHT OF THE PROPELLANT (LB)

SPI - ROCKET PROPELLANT SPECIFIC impulse (LB-SEC/LB)

RHO - ROCKET PROPELLANT DENSITY (LB/FT**3)

WIN - INITIAL VIRTUAL WEIGHT (LB)

TM1(3) - SOLID GRAIN MOMENTS OF INERTIA - IXX,IYY,IZZ (SLUG-FT**2)

TIG - ROCKET IGNITION TIME (SEC)

*************** ROCKET INPUTS ***************

FUN - ROCKET ON FLAG (1=ON)

PCG(3) - INITIAL X,Y,Z SEAT SYSTEM POSITION VECTOR OF THE

PROPELLANT CENTER OF GRAVITY (FT)

EA(3) - SEAT TO ROCKET PROPELLANT EULER ANGLES (DEG)

XRN(3) - X,Y,Z PROPELLANT SYSTEM POSITION VECTOR OF THE ROCKET

NOZZLE (FT)

YAW - YAW EULER ANGLE OF THE THRUST VECTOR IN THE PROPELLANT

COORDINATE SYSTEM (DEG)

PIT - PITCH EULER ANGLE OF THE THRUST VECTOR IN THE PROPELLANT

COORDINATE SYSTEM (DEG)

PL - PROPELLANT GRAIN LENGTH (FT)

POD - PROPELLANT GRAIN OUTSIDE DIAMETER (FT)

PID - PROPELLANT GRAIN INSIDE DIAMETER (FT)
DIMENSIONS OF CALLING ARGUMENTS......

DIMENSION TRF(5),FS1(3),TST(3),XCG(3),PMI(3),PP1(3),
        TM1(3),PCG(3),TA1(3),XRN(3)

INTERNAL DIMENSIONS......

DIMENSION VMNI(3),DSP(3,3),OPS(3,3),DC(3),EAIR(3),TEMPI(3),
        XRNST(3)

COMMON /CTIME/ TIME
COMMON /CLCAL/ ILCAL
COMMON /CVRLY/ INST
COMMON /CID/ IRAD,1WRITE,1U1AG

DATA RPD /01745329/
DATA GRAV /32.174/

**********
***** INITIALIZATION *****
**********

IF(ILCAL.NE.1) GO TO 80

DEFINE THE PROPELLANT CENTER OF GRAVITY IN THE SEAT SYSTEM
FOR OUTPUT......

DJ LG I=1,3
10 XCG(1) = PCG(1)

MISC INITIALIZATION......

IF(PW.NE.0) GO TO 30
WRITE(6,20)
20 FORMAT(5X,*) === PROPELLANT WEIGHT NOT INITIALIZED - RUN*,
                  * STOP

30 PW1 = PW
PMA = KON = FR = TIG = 0

50 PP1(1) = FS1(1) = TST(1) = 0

C CALCULATE THE SUSTAINER ROCKET'S TOTAL IMPULSE......

TOTIMP = 0
NA = TRF(2)
DO 50 I = 2,NA
   UELIMP = (TRF(I+NA+3)-TRF(I+NA+2))/(TRF(I+3)-TRF(I+2))*0.5
50 TOTIMP = TOTIMP + UELIMP

C CALCULATE THE SPECIFIC IMPULSe......

SPI = TOTIMP/PW

C CALCULATE THE INITIAL GRAIN VOLUME......

430
PV = 0.7854 * PL * (POD**2 - PID**2)

CALCULATE THE DENSITY

RHO = PW/PV

INITIAL VIRTUAL WEIGHT (THE EMPTY PORTION OF THE GRAIN)

WVI = 0.7854 * PL * PID**2 * RHO

VIRTUAL MASS MOMENTS OF INERTIA

VMIMASS = WVI/GRAV
VMIN(1) = (VMIMASS/12.) * (3.*((POD/2.)**2 + PL**2)
VMIN(2) = VMIN(1)
VMIN(3) = (VMIMASS/2.) * (POD/2.)**2

TOTAL MASS AS IF IT WERE A COMPLETELY SOLID GRAIN

TMASS = 0.7854 * POD**2 * PL * RHO/GRAV

TOTAL MOMENT OF INERTIAS

TM1(1) = (TMASS/12.) * (3.*((POD/2.)**2 + PL**2)
TM1(2) = TM1(1)
TM1(3) = (TMASS/2.) * (POD/2.)**2

INITIAL PROPELLANT MOMENT OF INERTIAS

DO 60 I=1,3
60 PMI(I) = TM1(I) - VMIN(I)

ROTATE THE PROPELLENT INERTIAS INTO THE SEAT SYSTEM

DO 70 J=1,3
70 EAIR(I) = EA(1) * RPU
CALL WIRCOS (DSP,EAIR)
CALL TRANS (DPS,DSP,3,3)
CALL ROTATEI (PMI,PPI,DPS)

RETURN IF SUSTAINER ROCKET IS OFF

IF(FOM.EQ.0 .OR. PHA.EQ.2.) GO TO 160

ROCKET ON

IF(PHA.EQ.1.) GO TO 100

PHA = RON = 1.
TIG = TIME

IF(INST.EQ.26) WRITE(6,90) TIME
90 FORMAT(5X,'SUSTAINER ROCKET ON AT TIME = ',F10.4,* SEC/)

COMPUTE THE DIRECTION COSINE MATRICES
110 EAIR(1) = EA(1) * RPU
CALL DIRCOS (DSP, EAIR)
CALL TRANS (DPS, DSP, 3, 3)

C COMPUTE THRUST VECTOR DIRECTION COSINES ......
C
THE = PIT * RPU
PSI = YAW * RPU
UL(1) = COS(PSI) * SIN(THt)
UL(2) = SIN(PSI) * SIN(THt)
UL(3) = COS(THt)

C CALCULATE THE BODY AXIS FORCE AND TORQUE COMPONENTS ......
C
NA = TRF(2)
TINRK = TIME - T1G
IF(TINRK .GE. TRF(NA+3)) GO TO 130
FR = -TBLU1(TINRK, TRF(4), TRF(NA+4), 1, -NA)
DO 120 I=1,3
120 TEMP(I) = UL(I) * FR
CALL MAIMPY (FST, UPS, TEMP, 3, 3, 1)
CALL VELXYZ (XRNST, XRN, PCG, DPS, 2)
CALL CRSRD (TST, XRNST, FST)

C PROPELLANT CONSUMPTION RATE (LB/SEC)......
C
IF(IPW .NE. 0) PWU = -FR/SP1
C
C PROPELLANT MASS BURNED (SLUGS)......
C
BM = (PW1 - PW)/GRAV

C BURNED VOLUME (FT**3)......
C
BVOL = BM/RHO/GRAV
C
C BURNED RADIUS OF GRAIN (FT)......
C
RK = SQRT((BVOL/PL/0.7854) + PID**2)/2.0
C
C NEW VIRTUAL MASS (SLUGS)......
C
VMP = VWI/GRAV + BM
C
C NEW VIRTUAL INERTIAS......
C
VMIN(1) = (VMP/12.) * (3.*BR**2 + PL**2)
VMIN(2) = VMIN(1)
VMIN(3) = (VMP/2.) * BR**2

C INERTIAS OF REMAINING PROPELLANT......
C
PM(1) = TMI(1) - VMIN(1)
PM(2) = PMI(1)
PM(3) = TMI(3) - VMIN(3)
C
C ROTATE THE ROCKET PROPELLANT INERTIA PROPERTIES INTO THE
C SEAT AXIS SYSTEM ......
C
CALL ROTATE1 (PM1, PPI, DPS)
GO TO 160
C
C ============== AT ROCKET BURNOUT ==============
C
130 IF(NST.EQ.26) WRITE(6,140) TIME
140 FORMAT(5X,*SUSTAINER ROCKET OFF AT TIME=*F10.4,2X,*SEC*)
   PWD = RUN = 0
   PHA = 2.
   DO 150 I=1,3
150   FST(I) = TST(I) = PMI(I) = 0
C
160 RETURN
END
SUBROUTINE WB (CM, CCG, CMI, CPI)
   * A5, W5, X5, SM1, SPI, W1, X1, 6M1, BP1,
   * W2, X2, 6M2, BP2, W3, X3, BM3, BP3)
C
DIMENSION CCG(3), CMI(3), CPI(3),
   * XS(3), SM1(3), SPI(3), X1(3), 6M1(3), BP1(3),
   * X2(3), BM2(3), BP2(3), X3(3), BM3(3), BP3(3)
C
DIMENSION TSMI(3), T1M1(3), T2M1(3), T3M1(3),
   * TSM1(3), T1PI(3), T2PI(3), T3PI(3), DIFF(3)
C
COMMON /CICAL/ IC, CAL
COMMON /CIAL/ IREAD, IWRITE, IDIA
C
DATA GkAy /32.174/
L.
L.
DESIGNED BY C.L. WEST
L.
L.
LAST MODIFIED - DECEMBER 6, 1980
L.
L.
NOTE - ALL MOMENT AND PRODUCT OF INERTIA VECTORS INPUT
L.
L.
INTO THIS ROUTINE HAVE BEEN ROTATED INTO THE
L.
L.
SEAT COORDINATE SYSTEM.
L.
L.
*************** WB OUTPUTS ***************
L.
CM - COMPOSITE SEAT WEIGHT (Lb)
CCG(3) - X,Y,Z SEAT AXIS SYSTEM COMPOSITE CENTER OF GRAVITY (FT)
CMI(3) - COMPOSITE SEAT MOMENT OF INERTIA VECTOR ABOUT ITS
   CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)
CPI(3) - COMPOSITE SEAT PRODUCT OF INERTIA VECTOR ABOUT ITS
   CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)

*************** WB INPUTS ***************
L.
AB - NUMBER OF BODIES ATTACHED TO THE BASIC SEAT
W5 - BASIC SEAT WEIGHT (Lb)
XS(3) - X,Y,Z SEAT AXIS SYSTEM POSITION VECTOR OF THE
   BASIC SEAT CENTER OF GRAVITY (FT)
SM1(3) - MOMENT OF INERTIA VECTOR FOR THE BASIC SEAT ABOUT
   ITS CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)
SPI(3) - PRODUCT OF INERTIA VECTOR FOR THE BASIC SEAT ABOUT
   ITS CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)
W1 - WEIGHT OF BODY ONE (Lb)
X1(3) - X,Y,Z SEAT AXIS SYSTEM POSITION VECTOR OF THE
   CENTER OF GRAVITY FOR BODY ONE (FT)
6M1(3) - MOMENT OF INERTIA VECTOR FOR BODY ONE ABOUT ITS
   CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)
BP1(3) - PRODUCT OF INERTIA VECTOR FOR BODY ONE ABOUT ITS
   CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)
W2 - WEIGHT OF BODY TWO (Lb)
X2(3) - X,Y,Z SEAT AXIS SYSTEM POSITION VECTOR OF THE
   CENTER OF GRAVITY FOR BODY TWO (FT)
BM2(3) - MOMENT OF INERTIA VECTOR FOR BODY TWO ABOUT ITS
   CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)
BP2(3) - PRODUCT OF INERTIA VECTOR FOR BODY TWO ABOUT ITS
   CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)
W3 - WEIGHT OF BODY THREE (Lb)
X3(3) - X,Y,Z SEAT AXIS SYSTEM POSITION VECTOR OF THE
CENTER OF GRAVITY FOR BODY THREE (FT)
BM3(3) - MOMENT OF INERTIA VECTOR FOR BODY THREE ABOUT ITS
CENTER OF GRAVITY - IXX, IYY, IZZ (SLUG-FT**2)
BP1(3) - PRODUCT OF INERTIA VECTOR FOR BODY THREE ABOUT ITS
CENTER OF GRAVITY - IXY, IXZ, IYZ (SLUG-FT**2)

******************************************************************************
****** INITIALIZATION ******
******************************************************************************

IF(ICCAL .NE. 1) GO TO 80
C
IF(AB .EQ. 0.99999) AB = 0.
C
ZERO WEIGHTS AND INERTIAS OF NON-EXISTANT BODIES
C
IF(W1 .EQ. 0.99999) W1 = 0.
IF(W2 . EQ. 0.99999) W2 = 0.
IF(W3 . EQ. 0.99999) W3 = 0.
IF(AB .GE. 1.) GO TO 20
DO 10 I=1,3
10 IF(X1(I) .EQ. 99999) X1(I) = 0
IF(BM1(I) .EQ. 99999) BM1(I) = 0
IF(BP1(I) .EQ. 99999) BP1(I) = 0
20 IF(AB .GE. 2.) GO TO 40
DO 30 I=1,3
30 IF(X2(I) .EQ. 99999) X2(I) = 0
IF(BM2(I) .EQ. 99999) BM2(I) = 0
IF(BP2(I) .EQ. 99999) BP2(I) = 0
40 IF(AB .GE. 3.) GO TO 60
DO 50 I=1,3
50 IF(X3(I) .EQ. 99999) X3(I) = 0
IF(BM3(I) .EQ. 99999) BM3(I) = 0
IF(BP3(I) .EQ. 99999) BP3(I) = 0
C
ZERO OUT THE MOMENT AND PRODUCT VECTORS .......
C
DO 70 I=1,3
70 TS1MI(I) = T1MI(I) = T2MI(I) = T3MI(I) = 0
C
COMPUTE THE LOCATION OF THE COMPOSITE C.G. FROM THE SRP .......
C
80 CG = WS + W1 + W2 + W3
DO 90 I=1,3
90 CG6(I) = (WS*X5S(I) + W1*X1(I) + W2*X2(I) + W3*X3(I))/CG
C
******************************************************************************
****** SEAT INERTIA PROPERTIES ******
******************************************************************************

C
CALCULATE THE SEAT MASS .......
C
BMSSS = WS/GRV
C
COMPUTE THE INERTIA PROPERTIES .......
C
C DO 100 I=1,3
100  DIFF(I) = CCG(I) - XS(I)
    CALL PAXIS (TSM1,TSPI,SM1,SP1,BMASS,DIFF)
C
C **************************************************************************
C ***** BODY 1 INERTIA PROPERTIES *****
C **************************************************************************
C IF (AB.LT.1.0) GO TO 140
C CALCULATE THE MASS OF BODY 1 .......
C BM Sass = W1/GRAV
C COMPUTE THE INERTIA PROPERTIES .......
C DO 110 I=1,3
110  DIFF(I) = CCG(I) - X1(I)
    CALL PAXIS (T1M1,T1P1,SM1,SP1,BM ASS,DIFF)
C
C **************************************************************************
C ***** BODY 2 INERTIA PROPERTIES *****
C **************************************************************************
C IF (AB.LT.2.0) GO TO 140
C CALCULATE THE MASS OF BODY 2 .......
C BM ASS = W2/GRAV
C COMPUTE THE INERTIA PROPERTIES .......
C DO 120 I=1,3
120  DIFF(I) = CCG(I) - X2(I)
    CALL PAXIS (T2M1,T2P1,SM2,SP2,BM ASS,DIFF)
C
C **************************************************************************
C ***** BODY 3 INERTIA PROPERTIES *****
C **************************************************************************
C IF (AB.LT.3.0) GO TO 140
C CALCULATE THE MASS OF BODY 3 .......
C BM ASS = W3/GRAV
C COMPUTE THE INERTIA PROPERTIES .......
C DO 130 I=1,3
130  DIFF(I) = CCG(I) - X3(I)
    CALL PAXIS (T3M1,T3P1,SM3,SP3,BM ASS,DIFF)
C
C **************************************************************************
C ********** COMPUTE THE COMPOSITE BODY INERTIA PROPERTIES **********
C **************************************************************************
C 140 DO 150 I=1,3

436
CMI(I) = TSMI(I) + T1MI(I) + T2MI(I) + T3MI(I)
I50 CPI(I) = T5PI(I) + T4PI(I) + T2PI(I) + T3PI(I)
C
RETURN
END
APPENDIX H

EASIEST SUBROUTINES AND FUNCTIONS

This appendix contains listings of the EASIEST subroutines and functions, which include the following:

- ARTAN2
- BLOCK
- BRIDL2
- BRIDL3
- BRIDL4
- CAD
- CEAERO
- COSDIR
- DET3
- DIRCOS
- DISECT
- EARATE
- FSW
- LAG
- LIBRIDL
- LILINE
- LILOAD
- LINOST
- UNPACK
- DET
- LINEPL
- LOOK
- PAXIS
- PCAERO
- RATIO
- RLIM
- ROTATEI
- TBLU3
- TLU
- UNPACK
- VECXYZ
- VELXYZ

438
FUNCTION ARTAN2(A1, AR)

FOUR QUADRANT ARCTANGENT FUNCTION, A1 BEING THE NUMERATOR AND AR
BEING THE DENOMINATOR.

IF(ABS(AR) < .000001 * ABS(A1)) 10, 10, 30
10 IF(A1) 20, 50, 20
20 ARTAN2 = 1.55079 * SIGN(A1, A1)
GO TO 60

30 ARTAN2 = ATAN(A1/AR)
40 IF(AR) 40, 20, 60
40 ARTAN2 = 3.14159 + ARTAN2
N = ARTAN2/3.14159
EN = N
ARTAN2 = ARTAN2 - 0.26318*EN
GO TO 60

50 ARTAN2 = 0.
60 RETURN
END
SUBROUTINE BLOCK (FSEAT,FAIRP,TSEAT,TAIRP,ZBL,
        SRPR,SRPA,RAIL,XRAIL,XBS,SRP,DPR,FRLT,
        UST,MST,WAP,WAP,DAE,DER,DRS,DSA,OSR,UP,FLAG)

DESIGNED BY C.L. WEST
LAST MODIFIED - DECEMBER 0, 1980

CALCULATES THE FORCES AND MOMENTS ON THE SEAT AND AIRPLANE FROM
THE BLOCKS

*************** BLOCK OUTPUTS ***************

FSEAT(3) - X,Y,Z SEAT BODY AXIS FORCE COMPONENTS (LB)
FAIRP(3) - X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS (LB)
TSEAT(3) - X,Y,Z SEAT BODY AXIS MOMENT COMPONENTS (FT-LB)
TAIRP(3) - X,Y,Z AIRPLANE BODY AXIS MOMENT COMPONENTS (FT-LB)
ZBL - RAIL AXIS Z COORDINATE OF BLOCK

*************** BLOCK INPUTS ***************

SRPR(3) - X,Y,Z RAIL POSITION VECTOR OF THE SRP (FT)
SRPA(3) - X,Y,Z AIRPLANE POSITION VECTOR OF THE SRP (FT)
RAILL - RAIL LENGTH (FT)
XRAIL - X,Y,Z AIRPLANE POSITION VECTOR OF THE ORIGIN
         OF THE RAIL COORDINATE SYSTEM (FT)
         POINT ON THE AIRPLANE (FT)
ABS(3) - X,Y,Z SEAT POSITION VECTOR OF THE BLOCK (FT)
SPR - RAIL SPRING CONSTANT (LB/FT)
DPR - RAIL DAMPING CONSTANT (LB/FT/SEC)
FRLT - SLIDER BLOCK FRICTION COEFFICIENT
UST(3) - SEAT AXIS VELOCITY OF SEAT REFERENCE POINT (FT/SEC)
MST(3) - SEAT AXIS ANGULAR RATES OF SEAT (RAD/SEC)
UAP(3) - AIRPLANE AXIS VELOCITY OF AIRPLANE (FT/SEC)
WAP(3) - AIRPLANE AXIS ANGULAR RATES OF AIRPLANE (RAD/SEC)
DAE(3,3) - AIRPLANE TO EARTH DIRECTION COSINE MATRIX
DER(3,3) - EARTH TO RAILS DIRECTION COSINE MATRIX
DSE(3,3) - RAILS TO SEAT DIRECTION COSINE MATRIX
DRS(3,3) - RAILS TO AIRPLANE DIRECTION COSINE MATRIX
DSA(3,3) - SEAT TO AIRPLANE DIRECTION COSINE MATRIX
DSE(3,3) - SEAT TO EARTH DIRECTION COSINE MATRIX
DSK(3,3) - SEAT TO RAILS DIRECTION COSINE MATRIX
UP - EJECTION DIRECTION FLAG
+1 = UPWARD WRT THE AIRPLANE
-1 = DOWNWARD WRT THE AIRPLANE
FLAG - BLOCK POSITION SWITCH ( 0 = ON RAILS  1 = OFF RAILS )

DIMENSIONS OF CALLING ARGUMENTS ........

DIMENSION FSEAT(3),FAIRP(3),TSEAT(3),TAIRP(3),SKPR(3),SKPA(3),
        XRAIL(3),ABS(3),SPR(2),DPR(2),UST(3),MST(3),UAP(3),
        WAP(3),DAE(3,3),DER(3,3),DRS(3,3),DSA(3,3),
        DSE(3,3),DSK(3,3)

INTERNAL DIMENSIONS ........

DIMENSION XBR(3),UXBE(3),FDEFL(3),ARM(3),
        X8A(3),UABE(3),RVDE(3),RVR3,TEMP(3)

440
COMMON/COVRLY/INST
COMMON/CLT/IREAD,WRITE,1DIAG
DATA TEMP/0.0,0.0/

CALCULATION OF SLIDER BLOCK LOCATION IN THE RAIL AXIS SYSTEM .......

CALL VELXYZ (XBR,XBS,SRPR,DSR,2)
ZBL = XBR(3)
TEMP(3) = XBR(3)

SET FORCES = 0 IF BLOCK OFF RAILS (EXCEPT DURING INITIALIZATION) ......

FLAG = 0
IF(INST.EQ.31 .OR. INST.EQ.61) GO TO 20
IF(XBR(3)*UP.GT.RAILL*UP) GO TO 20
DO 10 I=1,3
F3EAT(I)=0.
F3AIRP(I)=0.
T3EAT(I)=0.
T3AIRP(I)=0.
10 CONTINUE
FLAG = 1.
GO TO 50

COMPUTE VELOCITY OF BLOCK IN EARTH AXES SYSTEM ......

CALL VELXYZ (USBE,UST,XBS,WST,DSE)

COORDINATES OF BLOCK IN AIRPLANE AXES SYSTEM ......

CALL VELXYZ (XBA,XBS,SRPA,DSA,2)

VELOCITY OF BLOCK POSITION WRT THE AIRPLANE IN EARTH AXES SYSTEM ......

CALL VELXYZ (UABE,UAP,XBA,WAP,DAE)

RELATIVE VELOCITY OF BLOCK WRT THE RAILS IN EARTH AXES SYSTEM ......

DO 30 I=1,3
30 RVBE(1) = USBE(1) - UABE(1)

RELATIVE VELOCITY OF BLOCK WRT RAILS IN RAIL AXES SYSTEM ......

CALL MATMPY (RVBR,Der,RVBE,3,3,1)

FORCES ON SEAT IN RAIL AXES DUE TO RAIL RIGIDITY AND DAMPING ......

FDEFL(1) = -SPR(1) * XBR(1) - DPG(1) * RVBR(1)
FDEFL(2) = -SPR(2) * XBR(2) - DPG(2) * RVBR(2)
FRVEL = SIGN(AABS(AABS(KVRBR(3))),1.0)*RVBR(3)
FDEFL(3) = -FRIC*T SQRT(FDEFL(1)**2+FDEFL(2)**2)*FRVEL

FORCES ON SEAT IN SEAT AXIS SYSTEM ......

CALL MATMPY (FSEAT,DRS,FDEFL,3,3,1)

FORCES ON AIRPLANE IN AIRPLANE AXIS SYSTEM ......

441
C DO 40 I=1,3
40 FDEFL(I)=-FDEFL(I)
    CALL MATMPY (FAIRP,DRA,FDEFL,3,3,1)
C AIRPLANE MOMENT ARM .......
C    CALL VECXYZ (ARM,TEMP,XRAIL,DRA,2)
C MOMENTS ON SEAT .......
C    CALL CRSRDP (TSEAT,XBS,FSEAT)
C MOMENTS ON AIRPLANE .......
C    CALL CRSRDP (TAIRP,ARM,FA,RP)
C 50 RETURN
END
SUBROUTINE BRIOL2 (FAP, APX, XPCOO, PT1, PT2)

C THIS ROUTINE CALCULATES THE FORCE APPLICATION POINT OF A FORCE
C APPLIED TO A TWO STRAND BRIDLE.

***** BRIOL2 OUTPUTS *****
C FAP(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
  OF THE FORCE APPLICATION POINT (FT)

***** BRIOL2 INPUTS *****
C APX(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
  OF THE BRIDLE APX (FT)
C XPCOO(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
  OF THE PARACHUTE (FT)
C PT1(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
  OF BRIDLE ATTACHMENT POINT ONE (FT)
C PT2(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
  OF BRIDLE ATTACHMENT POINT TWO (FT)

DIMENSION FAP(3), APX(3), PT1(3), PT2(3), XPCOO(3)

DIMENSION DELA1(3), DELA2(3), DCN(3), XI(3), DC31(3), XIN(3),
  APXT(3), DIFF(3), UVI(3)

C CALCULATE THE DIRECTION COSINES OF THE NORMAL TO VECTORS APX, PT1
C AND APX, PT2 .......

DO 10 I = 1, 3
   DELA1(I) = PT1(I) - APX(I)
10    DELA2(I) = PT2(I) - APX(I)

CALL CRSPRD (OCN, DELA1, DELA2)

RN = SQRT (DCN(1)**2 + DCN(2)**2 + DCN(3)**2 )
DO 20 I = 1, 3
20    DCN(I) = DCN(I)/RN

C CALCULATE THE NORMAL FROM APX TO VECTOR PT1, PT2 .......
CALL LINOST (XI, RA1, DC31, PT1, PT2, APX)

C CALCULATE THE UNIT VECTOR FROM XPCOO TO PT1 .......
DO 30 I = 1, 3
30    DIFF(I) = PT1(I) - XPCOO(I)

RESULT = SQRT (DIFF(1)**2 + DIFF(2)**2 + DIFF(3)**2)
DO 40 I = 1, 3
40    UVI(I) = DIFF(I)/RESULT

C CALCULATE THE LOCATION OF THE BRIDLE CONFLUENCE POINT .......

PHI = ARCTAN2 ( DCN(1)*UVI(1) + DCN(2)*UVI(2) + DCN(3)*UVI(3),
  DCN(3)*UVI(1) - DCN(1)*UVI(3) )
* -((DC31(I)*UV1(I)+DC31(2)*UV1(2)+DC31(3)*UV1(3))
SINPHI = SIN(PHI)
COSPHI = COS(PHI)
50 APXT(I) = XI(I) + RA1 * (-DC31(I)*COSPHI + DCN(I)*SINPHI)
C
CALCULATE THE UNIT VECTOR AND MAGNITUDE OF THE NORMAL FROM
THE DRUIDLE CONFLUENCE POINT TO VECTOR PT1, PT2 ......
C
CALL LINDST (XI, RA1, DC31, PT1, PT2, APXT)
C
CALCULATE THE UNIT VECTOR FROM XPCDO TO APXT ......
C
DO 60 I=1, 3
60 DIFF(I) = APXT(I) - XPCDO(I)
C
RESULT = SQRT(DIFF(1)**2+DIFF(2)**2+DIFF(3)**2)
C
DO 70 I=1, 3
70 UV1(I) = DIFF(I)/RESULT
C
DOT THE PARACHUTE LINE UNIT VECTOR ONTO DC31 ......
C
CALL DOTPROD (COSINE, UV31, UV1, 3)
C
DETERMINE THE MAGNITUDE OF THE VECTOR FROM THE CONFLUENCE POINT
TO VECTOR PT1, PT2 ALONG THE LINE FORCE UNIT VECTOR ......
C
IF(COSINE.NE.0.) R1 = RA1/COSINE
C
CALCULATE THE INTERSECTION OF THE PARACHUTE LINE FORCE VECTOR
WITH VECTOR PT1, PT2 ......
C
DO 60 I=1, 3
60 XIN(I) = APXT(I) + R1 * UV1(I)
C
DETERMINE THE FORCE APPLICATION POINT ......
C
TEST = (XIN(1)-PT1(1))*(PT2(1)-PT1(1))+(XIN(2)-PT1(2))*(PT2(2)-PT1(2))
* (XIN(3)-PT1(3))*(PT2(3)-PT1(3))
IF (TEST.LE.0) GO TO 120
C
R11N = SQRT((XIN(1)-PT1(1))**2+(XIN(2)-PT1(2))**2+
* (XIN(3)-PT1(3))**2)
R12 = SQRT ((PT2(1)-PT1(1))**2 + (PT2(2)-PT1(2))**2 +
* (PT2(3)-PT1(3))**2 )
IF (R12-K11N.GE.0.) GO TO 100
C
DO 90 I=1, 3
90 FAP(I) = PT2(I)
GO TO 140
C
100 DO 110 I=1, 3
110 FAP(I) = XIN(I)
GO TO 140
C
*20 DO 130 I=1, 3
130 FAP(I) = PT1(I)
SUBROUTINE BRIDL3 (FAP, APX, UV, XPCDO, PT1, PT2, PT3, XI)

C ROUTINE FOR COMPUTING THE FORCE APPLICATION POINT FOR A BRIDLE WITH THREE FLEXIBLE LINES.

***** BRIDL3 OUTPUTS *****
FAP(3) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE FORCE APPLICATION POINT (FT)

***** BRIDL3 INPUTS *****
APX(3) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE BRIDLE APEX (FT)
UV(3) - PARACHUTE LINE UNIT VECTOR
XPCDO(3) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE (FT)
PT1(3) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF BRIDLE ATTACHMENT POINT ONE (FT)
PT2(3) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF BRIDLE ATTACHMENT POINT TWO (FT)
PT3(3) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF BRIDLE ATTACHMENT POINT THREE (FT)
XI(3) - X, Y, Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE INTERSECTION OF THE BRIDLE ATTACHMENT POINTS PLANE WITH THE PARACHUTE LINE FORCE VECTOR (FT)

DIMENSION FAP(3), APX(3), UV(3), PT1(3), PT2(3), PT3(3), XI(3), XPCDO(3)
DIMENSION XNI23(3), DC123(3), XNI312(3), DC123(3), XN213(3)
XN123(3), DC213(3), XN123(3), DC123(3), XN312(3), DC312(3)

C COMPUTE THE INTERSECTION OF THE NORMAL FROM POINT 1 TO VECTOR 2,3 ......
CALL LINOST (XNI23, 0, DC123, PT2, PT3, PT1)
C COMPUTE THE INTERSECTION OF THE NORMAL FROM THE FORCE-PLANE INTERSECTION TO VECTOR 2,3 ......
CALL LINOST (XNI23, 0, DC123, PT2, PT3, XI)
C TEST FOR COMPRESSION IN LINE 1 .......
TEST = DC123(1)*DC123(1) + DC123(2)*DC123(2) + DC123(3)*DC123(3)
IF (TEST) 10, 10, 20
C LINE 1 UNDER COMPRESSION - COMPUTE THE FORCE APPLICATION POINT LYING ON VECTOR 2,3 ......
10 CALL BRIDL2 (FAP, APX, XPCDO, PT2, PT3)
GO TO 60
C COMPUTE THE NORMAL FROM POINT 2 TO VECTOR 1,3 ......

446
20 CALL LINDST (XN213,0,DC213,PT1,PT3,PT2)
C
C COMPUTE THE NORMAL FROM THE FORCE-PLANE INTERSECTION TO
C VECTOR 1,3 .......
C
CALL LINDST (XN113,0,DC113,PT1,PT3,X1)
C
------ TEST FOR COMPRESSION IN LINE 2 -------
C
TEST = DC213(1)*DC113(1) + DC213(2)*DC113(2) +
      * DC213(3)*DC113(3)
IF (TEST) 30,30,40
C
C LINE 2 UNDER COMPRESSION - COMPUTE THE FORCE APPLICATION
C POINT LYING ON VECTOR 1,3 .......
C
30 CALL BRIDL2 (FAP,APX,APDO,PT1,PT3)
GO TO 80
C
C COMPUTE THE NORMAL FROM POINT 3 TO VECTOR 1,2 .......
C
40 CALL LINDST (XN312,0,DC312,PT1,PT2,PT3)
C
C COMPUTE THE NORMAL FROM THE FORCE-PLANE INTERSECTION TO
C VECTOR 1,2 .......
C
CALL LINDST (XN112,0,DC112,PT1,PT2,X1)
C
------ TEST FOR COMPRESSION IN LINE 3 -------
C
TEST = DC312(1)*DC112(1) + DC312(2)*DC112(2) +
      * DC312(3)*DC112(3)
IF (TEST) 50,50,60
C
C LINE 3 UNDER COMPRESSION - COMPUTE THE FORCE APPLICATION
C POINT LYING ON VECTOR 1,2 .......
C
50 CALL BRIDL2 (FAP,APX,APDO,PT1,PT2)
GO TO 80
C
C ------- ALL THREE LINES IN TENSION -------
C
60 DO 70 I=1,3
70 FAP(I) = X1(I)
C
60 RETURN
END
SUBROUTINE BRIUL4 (FAP,
   APX,UV,XPCDU,AP1,AP2,AP3,AP4,XI)
C THIS ROUTINE DETERMINES THE THREE BRIDLE ATTACHMENT
   POINTS TO BE USED IN BRIUL3
C ***** BRIUL4 OUTPUTS *****
C FAP(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
   OF THE FORCE APPLICATION POINT (FT)
C ***** BRIUL4 INPUTS *****
C APX(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
   OF THE BRIDLE APEX (FT)
C UV(3) - PARACHUTE LINE FORCE UNIT VECTOR
C XPCDU(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
   OF PARACHUTE (FT)
C AP1(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
   OF BRIDLE ATTACHMENT POINT ONE (FT)
C AP2(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
   OF BRIDLE ATTACHMENT POINT TWO (FT)
C AP3(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
   OF BRIDLE ATTACHMENT POINT THREE (FT)
C AP4(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
   OF BRIDLE ATTACHMENT POINT FOUR (FT)
C XI(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
   OF THE INTERSECTION OF THE BRIDLE ATTACHMENT POINTS PLANE
   WITH THE PARACHUTE LINE FORCE VECTOR (FT)
C DIMENSION FAP(3),APX(3),UV(3),AP1(3),AP2(3),AP3(3),AP4(4),
   XI(3),XPCDU(3)
C DIMENSION XNI24(3),DC124(3),XNI24(3),DC124(3)
C THE FOUR ATTACHMENT POINTS OF THE BRIDLE ARE DESIGNATED AS 1,2,3,4
   (NUMBERED CONSECUTIVELY IN A COUNTER CLOCKWISE DIRECTION)
C LET POINTS 2 AND 4 DEFINE A LINE AND CHECK TO SEE WHICH SIDE OF
C THE LINE THE FORCE-PLANE INTERSECTION LIES ON
C CALL LINST (XNI24,DC124,AP2,AP4,AP1)
CALL LINST (XNI24,DC124,AP2,AP4,XI)
C TEST = DC124(1)*DC124(1)+DC124(2)*DC124(2)+DC124(3)*DC124(3)
IF (TEST .LE. 0.) GO TO 10
C CALL BRIUL3 (FAP,APX,UV,XPCDU,AP1,AP2,AP3,AP4,XI)
GO TO 20
C 10 CALL BRIUL3 (FAP,APX,UV,XPCDU,AP1,AP2,AP4,XI)
C 20 RETURN
END
SUBROUTINE CAD (CF,EF,EFDOT,IEF,EL,ELDOT,IEL,WK,WKDOT,IWK, W8,W8DOT,IM8, FL, TCP,TIME,CEX,CSK,C1,C,VI,PA,TF,CVH,CBP,C1, CV2,C2,THA,B,EXP,PT,R,TYPE, TSO,FSO,TDE)

C COMPUTES THE PERFORMANCE OF A CLOSED TELESCOPING TUBE
C ACTING AGAINST A LOAD IN ANY ENVIRONMENT AND USING A BURNING
C PROPELLANT AS A SOURCE OF ENERGY .......

DIMENSION TCP(5)
COMMON / COVRLY / INST
COMMON / CIU / IREAD, IWRITE, IDIAG
DATA PI02 / 1.57080 /

C PRINT CATAPULT IGNITION STATEMENT .......
IF(INST.NE.0) GO TO 20
IF(INST.EQ.26) WRITE(6,10) TYPE,TIME
10 FORMAT(/5X,AB,* IGNITION AT TIME = *,FL,0.4,* SEC*)
FL = 1.0

C CALCULATE THE CATAPULT FORCE DECAY AFTER STRIPOFF .......
IF(FL.LT.1.0) GO TO 40
TASO = TIME - TSO
IF(TASO.LT.TDE) GO TO 30
FL = 3.4
CF = 0
GO TO 150
30 IF(TDE.NE.0) CF = FSO * COS(TASO/TDE * PI02)
GO TO 150

C COMPUTE PROPELLANT CONSUMED .......
NA=TCP(2)
W = (1 + TBLU1 (WB,TCP(4),TCP(NA+4),1,-NA)

C HAS ALL THE PROPELLANT BURNED .......
IF(C.GE.W) GO TO 50

C ALL BURNED .......
W = C

C COMPUTE INTERNAL VOLUME .......
VOL = VI + PA * CEX * 12.

C DON'T LET THE VOLUME DECREASE BELOW INITIAL VALUE .......
IF(VOL.GE.VI) GO TO 60
VOL = VI
60 IF(W.NE.0.0) GO TO 70
TEMP = TF

449
GO TO 60
C 70 TEMP = TF - (WK + EF + EL)/(W * CVH)
C COMPUTE CHAMBER PRESSURE USING EQUATION OF STATE .......
C 60 PRESS = 12.0 * R * TEMP * W / VOL
C PRINT CATAPULT BURT STATEMENT IF REQUIRED) .......
C
   IF(CBP.GE.PRESS) GO TO 100
   IF(INST.EQ.26) WRITE(6,90) TYPE, PRESS, TIME
   90 FORMAT('/5X,*GRARRKDDBUDDMMMA/F',//5X,AB,'*
   * BURT AT *,F10.4,* LOS PRESSURE AT TIME = *,F10.4,* SEC//')
   STOP
C HAS THE PRESSURE UNLOCKED THE PISTON YET .......
C 100 IF(PRESS.GT.PT) GO TO 110
C STILL LOCKED - SET CATAPULT FORCE TO ZERO
C CF = 0.0
   GO TO 120
C UNLOCKED - HIT 'EM AND MOVE 'EM OUT .......
C 110 CF = PA*PRESS*(1.-C1)
C
---------------------------------------------------------------------------------
C * COMPUTE INTERNAL FRICTIONAL ENERGY RATE, HEAT LOSS RATE, *
C * CATAPULT WORK RATE, AND THE PROPELLANT BURN RATE *
C *---------------------------------------------------------------------------------
C COMPUTE THE INTERNAL FRICTIONAL ENERGY RATE (POWER) ......
C 120 IF(IEF.NE.0) EFDOOT = ABS(C1*PRESS*CV)
C COMPUTE THE HEAT LOSS RATE ...... 
C IF(IH.LT.0) HDOT = ABSC2*(TEMP - TT)*THA)
C COMPUTE CATAPULT WORK RATE ...... 
C IF(IWK.LT.0) WKDOT = ABSC*CV)
C COMPUTE PROPELLANT BURN RATE ...... 
C PbR = 0.0
   IF(W.le.C) GO TO 130
   PbR = B*ABS(PRESS)**EXP
C 130 IF(IWB.LT.0) WBDOOT = PbR
C //////////////////////////////////////////////////////////////////~~~~~~~~~~~~~~~~~
C WHEN STRIPOFF OCCURS ......
C
IF(CEX.LT.CSK) GO TO 150
FL = 2.*
EFDOT = 0.*
ELDOT = 0.*
WKKDOT = 0.*
WABB = 0.*
TSO = TIME
F50 = CF

C PRINT CATAPULT STRIPOFF STATEMENT
C
IF(INST.EQ.26) WRITE(6,140) TYPE,TIME
140 FORMAT(5X,AB,* STRIPOFF AT TIME = *,F10.4,* SEC*/)
C
150 RETURN
END
SUBROUTINE CE (UCP, UUCP, IUCP, XCP, XDCP, IXCP, WCP, WDCP, IWCP,
- ECP, EDCP, IECP, SC, SCDDOT, ISCD, SC, SCDDOT, ISC,
- GA, GY, GZ, DUCP, TAD, TAD, W1, S, B, CIN, CIN, CIN, CIN,
- CL, CM, CN, ALFA, BETA, VMACH, Q, ALT, SEP,
- SW, PW, CE, CI, CLP, CMQ, CNP, AX, AX, AX, AX)

************** CE OUTPUTS **************

LINEAR VELOCITIES - BODY AXIS

UCP(3) - X, Y, Z LINEAR VELOCITY VECTOR OF THE CREWPERSON (FT/SEC)
UWCP(3) - X, Y, Z LINEAR VELOCITY RATE VECTOR OF THE CREWPERSON
IUCP(3) - INTEGRATION CONTROL

LINEAR POSITIONS - EARTH SYSTEM

XCP(3) - X, Y, Z LINEAR POSITION VECTOR OF THE CREWPERSON (FT)
XDCP(3) - X, Y, Z LINEAR POSITION RATE VECTOR OF THE CREWPERSON (FT/SEC)
IXCP(3) - INTEGRATION CONTROL

ANGULAR VELOCITIES - BODY AXIS

WCP(3) - X, Y, Z ANGULAR VELOCITY COMPONENTS - P, Q, R (DEG/SEC)
WDCP(3) - X, Y, Z ANGULAR VELOCITY RATE COMPONENTS (DEG/SEC/SEC)
IWCP(3) - INTEGRATION CONTROL

EULER ANGLES -- EARTH TO BODY -- YAW, PITCH, ROLL

ECP(3) - EARTH TO CREWPERSON EULER ANGLES (DEG)
EDCP(3) - EULER ANGULAR RATES (DEG/SEC)
IECP(3) - INTEGRATION CONTROL

SPINAL COMPRESSION VELOCITY ........

SCD - SPINAL COMPRESSION VELOCITY (FT/SEC)
SCDQ01 - SPINAL COMPRESSION VELOCITY RATE (FT/SEC/SEC)
ISCD - INTEGRATION CONTROL

SPINAL COMPRESSION ........

SC - SPINAL COMPRESSION (FT)
SCDQT - SPINAL COMPRESSION RATE (FT/SEC)
ISC - INTEGRATION CONTROL

GA - CREWPERSON SYSTEM X-AXIS LOAD FACTOR (G)
GY - CREWPERSON SYSTEM Y-AXIS LOAD FACTOR (G)
GZ - CREWPERSON SYSTEM Z-AXIS LOAD FACTOR (G)
DR - DYNAMIC RESPONSE

FAD(3) - X, Y, Z BODY AXIS FORCE COMPONENTS OF THE AERODYNAMIC
- FORCE ACTING ON THE CREWPERSON (LB)
TAD(3) - X, Y, Z BODY AXIS TORQUE COMPONENTS OF THE AERODYNAMIC
- TORQUE ACTING ON THE CREWPERSON (FT-LB)
W1 - WEIGHT OF THE CREWPERSON CORRESPONDING TO HIS
- PERCENTILE PLUS CLOTHING AND EQUIPMENT (LB)
S - AERODYNAMIC REFERENCE AREA (FT**2)
C - AERODYNAMIC LATERAL REFERENCE LENGTH (FT)
C - AERODYNAMIC LONGITUDINAL REFERENCE LENGTH (FT)
CIN(4) - CREWPERSON INERTIA PROPERTIES TO BE USED AFTER
SEAT/CREWPERSON SEPARATION
CIN(1) = IXX
CIN(2) = IYY
CIN(3) = IZZ
CIN(4) = IXZ
CA - X AXIS AERODYNAMIC FORCE COEFFICIENT
CY - Y AXIS AERODYNAMIC FORCE COEFFICIENT
CZ - Z AXIS AERODYNAMIC FORCE COEFFICIENT
CM - AERODYNAMIC ROLLING MOMENT COEFFICIENT
CN - AERODYNAMIC YAWING MOMENT COEFFICIENT
ALPHA - CREWPERSON ANGLE OF ATTACK (DEG)
BETA - CREWPERSON SLIP ANGLE (DEG)
VMACH - CREWPERSON MACH NUMBER
Q - CREWPERSON DYNAMIC PRESSURE (LB/FT**2)
ALT - CREWPERSON ALTITUDE (FT)
SEP - SEAT/CREWPERSON SEPARATION FLAG FOR OUTPUT
(1 = SEPARATION)
************* CE INPUTS *************
Sw - FLAG FOR SEAT/CREWPERSON SEPARATION
(1 = SEPARATION)
PC - CREWPERSON PERCENTILE
CEW - WEIGHT OF CREWPERSON CLOTHING AND EQUIPMENT (LB)
CMI(3) - CREWPERSON MOMENT OF INERTIA VECTOR - IXX, IYY, IZZ
(SLUG-FT**2)
CPI(3) - CREWPERSON PRODUCT OF INERTIA VECTOR - IXY, IXZ, IYZ
(SLUG-FT**2)
CLP - AERODYNAMIC ROLL DAMPING COEFFICIENT (1/DEG)
CmP - AERODYNAMIC PITCH DAMPING COEFFICIENT (1/DEG)
CNK - AERODYNAMIC YAW DAMPING COEFFICIENT (1/DEG)
XSP(3) - X, Y, Z CREWPERSON SYSTEM POSITION VECTOR OF THE BASE
OF THE SPINE (FT)
FAB(3) - X, Y, Z BODY AXIS FORCE COMPONENTS ACTING ON THE CREWPERSON
FROM THE RESTRAINT COMPONENT (LB)
TAB(3) - X, Y, Z BODY AXIS TORQUE COMPONENTS ACTING ON THE CREWPERSON
FROM THE RESTRAINT COMPONENT (FT-LB)
FDO(3) - X, Y, Z BODY AXIS FORCE COMPONENTS ACTING ON THE CREWPERSON
FROM THE PARACHUTE LINE COMPONENT (LB)
TDO(3) - X, Y, Z BODY AXIS TORQUE COMPONENTS ACTING ON THE CREWPERSON
FROM THE PARACHUTE LINE COMPONENT (LB)
FAU(3) - X, Y, Z BODY AXIS FORCE COMPONENTS ACTING ON THE CREWPERSON
--- AN AUXILIARY INPUT --- (LB)
TAU(3) - X, Y, Z BODY AXIS TORQUE COMPONENTS ACTING ON THE CREWPERSON
--- AN AUXILIARY INPUT --- (FT-LB)
TRM(3) - X, Y, Z PARENT BODY EARTH VELOCITY COMPONENTS
TO DETERMINE POSITION RATES DURING TRIM (FT/SEC)
DIMENSIONS OF CALLING ARGUMENTS ........
DIMENSION UCP(3), UUCP(3), IUCP(3), XCP(3), XDCP(3), IXCP(3),
, WCP(3), WUCP(3), IWCP(3), ECP(3), EDCP(3), IECP(3),
, FAD(3), TAD(3), CIN(4),
, CMI(3), CPI(3), XSP(3), FAB(3), TAB(3), FDO(3), TDO(3),
C FAU(3), TAU(3), TRM(3)
C
C INTERNAL DIMENSIONS ......
C
DIMENSION TINER(3,:), TEMP1(3), TEMP2(3), TEMP3(3),
  UWE(3), UWE(3), UWE(3), ECPIR(3), WCPIR(3), EDCPIR(3),
  WCPIR(3), UCE(3), UCE(3), TBLCP(10), TBLCPWT(10),
  TBLIXX(10), TBLIYY(10), TBLIZZ(10), TBLIXZ(10), TBLS(10),
  TBLB(10), TBLC(10), T(3), T(3)
C
COMMON / CCICAL / ITCALL
COMMON / CCICAL / ITCALL
COMMON / CTIME / TIME
COMMON / CSSFLG / SSFLG
COMMON / CIO / IREAD, IWRITE, IDIAG
C
DATA KPO, DPK / .01745329, 57.29578 /
DATA GRAV / 32.174/
C
DATA TBLCP /* T,..., 15., 25., 35., 45., 65., 75., 85., 95.*/
DATA TBLCPWT /* T,..., 132.3, 142.7, 149.1, 154.3, 159.3, 164.6, 170.5, 177.4, 186.5, 200.9*/
DATA TBLIXX /* T,..., 10.53, 11.51, 12.10, 12.56, 13.00, 13.47, 14.00, 14.64, 15.51, 16.97*/
DATA TBLIYY /* T,..., 10.38, 11.57, 12.16, 12.61, 13.04, 13.50, 14.01, 14.63, 15.48, 16.92*/
DATA TBLIZZ /* T,..., 1.68, 1.78, 1.85, 1.90, 1.95, 2.01, 2.07, 2.14, 2.24, 2.41*/
DATA TBLIXZ /* T,..., -5.2, -5.1, -5.0, -5.0, -4.9, -4.8, -4.7, -4.6*/
DATA TBLB /* T,..., 1.38, 1.41, 1.44, 1.46, 1.48, 1.50, 1.52, 1.54, 1.57, 1.62*/
DATA TBLGC /* T,..., 5.43, 5.55, 5.63, 5.69, 5.74, 5.79, 5.84, 5.89, 5.97, 6.10*/
C
C ***********************
C *** INITIALIZATION ****
C
C
C IF (ICCAL, NE.1) GO TO 20
C
C CX = CY = CL = CM = CN = ALPHA = BETA = VMACH = G
Q = SEP = 0
DU 1U 1 = 1, 3
TRM(1) = FAU(1) = TAD(1) = 0
IF(XSP(1)) EQ. 0.99999) XSP(1) = 0
IF(FAB(1)) EQ. 0.99999) FAB(1) = 0
IF(TAB(1)) EQ. 0.99999) TAB(1) = 0
IF(FDO(1)) EQ. 0.99999) FDO(1) = 0
IF(TDO(1)) EQ. 0.99999) TDO(1) = 0
IF(FAU(1)) EQ. 0.99999) FAU(1) = 0
IF(TAU(1)) EQ. 0.99999) TAU(1) = 0
C 10 CONTINUE
C IF (SW, EQ. 0.99999) SW = 0
WT = TBLUL1(PC, TBLCP, TBLCPWT, 1, -10) + CEW
S = TBLUL1(PC, TBLCP, TBLCPWT, 1, -10)
V = TBLUL1(PC, TBLCP, TBLCPWT, 1, -10)
C = TBLUL1(PC, TBLCP, TBLCPWT, 1, -10)
C
C ****** CALCULATE THE CHEMPERSON INERTIAS FOR USE AFTER ******
SEAT/CREWPERSON SEPARATION

C
C CIN(1) = TBLU1(PC,TBLCP,TBLIX,1,1-10)
C CIN(2) = TBLU1(PC,TBLCP,TBLYY,1,1-10)
C CIN(3) = TBLU1(PC,TBLCP,TBLZZ,1,1-10)
C CIN(4) = TBLU1(PC,TBLCP,TBLIXZ,1,1-10)
C
C//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C
C CHANGE FROM DEGREES TO RADIANS ...... 
C
20 DO 30 I=1,3
30 ECPIR(I) = ECP(I) * RPD
30 WCPIR(I) = WCP(I) * RPD
C
C COMPUTE THE DIRECTION COSINE MATRICES ...... 
C
CALL DIRCOS (DEC,ECPIR)
CALL TRANS (DEC,DEC,3,3)
C
C ESTABLISH POSITIVE ALTITUDE ...... 
C
ALT = - XCP(3)
C
C BYPASS THE AERODYNAMIC CALCULATIONS UP TO SEAT/CREWPERSON 
C SEPARATION ...... 
C
IF(SW.EQ.1.) GO TO 40
C
C SET UP THE SEATED CREWPERSON INERTIA TENSOR ...... 
C
TINER(I,1) = CMI(1)
TINER(I,2) = -CPI(1)
TINER(I,3) = -CPI(2)
TINER(2,1) = -CPI(1)
TINER(2,2) = CMI(2)
TINER(2,3) = -CPI(3)
TINER(3,1) = -CPI(2)
TINER(3,2) = -CPI(3)
TINER(3,3) = CMI(3)
GO TO 110
C
C SET UP THE EXTENDED CREWPERSON INERTIA TENSOR ...... 
C
40 TINER(I,1) = CIN(1)
TINER(I,2) = 0
TINER(I,3) = -CIN(4)
TINER(2,1) = 0
TINER(2,2) = CIN(2)
TINER(2,3) = 0
TINER(3,1) = -CIN(4)
TINER(3,2) = 0
TINER(3,3) = CIN(3)
C
C WRITE THE SEAT/CREWPERSON SEPARATION MESSAGE ...... 
C
IF(SEP.EQ.1.) GO TO 60
SEP = 1.

455
IF(INST.EQ.26) WRITE(*,50) TIME
50 FORMAT(/5X,'SEAT/CREW PERSON SEPARATION AT TIME = *,F10.4,* SEC'/)

C-obtain speed of sound, air density, and wind velocity ......
C
C-call ATMS (VS,RHU,ALT,UM,0,0)
C
C-put the wind into body axis ......
C
C-call MAINPY (UMB,DEC,UM,3,3,1)
C
C-add the wind velocity to the crewperson velocity ......
C
DO 90 I=1,3
90 UM(I) = UCPI(I) - UWB(I)
C
C-calculate the aero variables ......
C
IF(UU(I).EQ.0. .AND. UU(3).EQ.0.) U0(I) = .01
ALPHA = ARTAN2(UU(3),UU(I)) * DPR
CALL UQTPRD (VBAR2,UM,0,3)
VBAR = SQRT(VBAR2)
BETA = ASIN(UU(2)/VBAR) * DPR
VMACH = VBAR/VS
C
C-compute dynamic pressure x reference area ......
C
Q = .5 * RHO * VBAR2
QAC = Q * S
C
C-calculate the aerodynamic coefficients ......
C
TBLALPH = ALPHA
IF(ALPHA. LT. 0) TBLALPH = ALPHA + 360.0
TBLBETA = ABS(BETA)
CALL CEERO (CX,CY,CZ,CL,CM,CN,TBLALPH,TBLBETA,PC)
C
CY = CY * SIGN(1.,BETA)
CL = -CL * SIGN(1.,BETA)
CN = -CN * SIGN(1.,BETA)
C
C-all damping terms for an airspeed greater than .1 ft/sec ......
C
IF(VBAR .LE. 0.1) GO TO 100
C
CO2V = C/(VBAR+VBAR)
BO2V = B/(VBAR+VBAR)
C
C-add roll damping ......
C
CL = CL + CLP * WCP(1) * BO2V
C
C-add pitch damping ......
C
CM = CM + CMQ * WCP(2) * CO2V
C
C-add yaw damping ......
C
**AERODYNAMIC FORCES**

\[ F_{AD}(I) = Q_{AC} \cdot CX \]
\[ F_{AD}(2) = Q_{AC} \cdot CY \]
\[ F_{AD}(3) = Q_{AC} \cdot CZ \]

**AERODYNAMIC TORQUES**

\[ T_{AD}(1) = Q_{AC} \cdot B \cdot CL \]
\[ T_{AD}(2) = Q_{AC} \cdot C \cdot CM \]
\[ T_{AD}(3) = Q_{AC} \cdot B \cdot CN \]

**SUM FORCES (INCLUDING GRAVITY) AND MOMENTS**

\[ F(i) = F_{AD}(I) + F_{DO}(I) + F_{AU}(I) + F_{AD}(I) + WT \cdot DEC(I,3) \]
\[ T(i) = T_{AB}(I) + T_{LU}(I) + TAU(I) + TAD(I) \]

**CALCULATE THE DYNAMIC RESPONSE**

\[ UK = SC \cdot 86.977 \]

**ANGULAR VELOCITY EQUATIONS**

**CALCULATE TINER X WCPIR**

**CALCULATE WCPIR X (TINER = WCPIR)**

**SUM TERMS TO OBTAIN TOTAL TORQUE**

\[ DO 130 I=1,3 \]
\[ TEMP3(I) = T(I) - TEMP2(I) \]

**CALCULATE WDCPIR**

**CALL LUEWS (TINER, TEMP1, TEMP3, TEMP2) (WDCPIR(I) = TEMP1(I))**

**CALL INERTIA MATRIX OF CREWPERSON IS SINGULAR...RUN STOPPED**

**STOP**

\[ GU 160 I=1,3 \]
\[ IF(1 \cdot WCPIR(I) .NE.0) WDCPIR(I) = TEMP1(I) \]
CALL EARATE (TEMP1, WCPIR, ECPIR)
DO 170 I=1,3
170 IF(IECP(I).NE.0) ECDCP(I) = TEMP1(I)
C
*****************************************************************************************
C ****** LINEAR VELOCITY EQUATIONS ******
C *****************************************************************************************
C CALCULATE THE CORIOLIS ACCELERATION (WCPIR X UCP) .......
C CALL CRSPRD (TEMP1, WCPIR, UCP)
C CALCULATE F/M .......
C
CPMASS = WT/GRAV
DO 180 I=1,3
180 TEMP2(I) = F(I)/CPMASS
C
SUM THE ACCELERATION COMPONENTS .......
DO 190 I=1,3
190 IF(IUCP(I).NE.0) UDCP(I) = TEMP2(I) - TEMP1(I)
C
===== CALCULATE THE LOAD FACTORS =====
C DETERMINE WCPIR X XSP .......
C CALL CRSPRD (TEMP1, WCPIR, XSP)
C DETERMINE WCPIR X (WCPIR X XSP) .......
C CALL CRSPRD (TEMP2, WCPIR, XSP)
CALL CRSPRD (TEMP3, WCPIR, TEMP2)
C
DETERMINE THE LOAD FACTORS .......
GA = -(F(1)/CPMASS + TEMP1(1) + TEMP3(1))/GRAV
GY = -(F(2)/CPMASS + TEMP1(2) + TEMP3(2))/GRAV
GZ = -(F(3)/CPMASS + TEMP1(3) + TEMP3(3))/GRAV
C
*****************************************************************************************
C ****** LINEAR POSITION EQUATIONS ******
C *****************************************************************************************
C CALL MATMPY (TEMP1, DCE, UCP, 3, 3, 1)
DO 200 I=1,3
200 IF(IULCP(I).NE.0) XLCP(I) = TEMP1(I)
C
*****************************************************************************************
C ****** SPINAL COMPRESSION EQUATIONS ******
C *****************************************************************************************
C SPINAL COMPRESSION VELOCITY EQUATION .......
C IF(ISCO.NE.0) SCDOOT = -23.6992 * SCO + 2796.41 * SC
     + GRAV * GZ
C
C SPINAL COMPRESSION EQUATION .......

458
IF(ISC.NE.0) SCDOT = SC

**DURING TRIM, SUBTRACT TRIM VELOCITY FROM POSITION RATES **

IF(INST.NE.31) GO TO 220
DO 210 I=1,3
210 IF(IYCP(I).NE.0) XDCP(I) = XDCP(I) - TRM(I)

** ***** CHANGE FROM RADIANS TO DEGREES *****

DO 230 I=1,3
230 EUCP(I) = EDCP1R(I) * DPR
230 WDCP(I) = WDCP1R(I) * DPR

RETURN
END
SUFRONTINE CEACROZ (CZ,CY,CZ,CZ,CZ,CM,CN,ALPHA,BETA,PC)

DIMENSION T(-8,13),T(-8,13),T(-8,13)

DATA (TCX(1,J), K=1,1)

* -0.7063, -0.7063, -0.6642, -0.6099, -0.5004, -0.3017, -0.0729
* -0.3017, -0.3174, -0.2144, -0.2471, -0.1553, -0.1235, -0.0898
* 0.0191, 0.0191, 0.0017, 0.0879, 0.0898, 0.0898, 0.0898
* 0.3017, 0.3017, 0.3017, 0.3017, 0.3017, 0.3017, 0.3017
* 0.0191, 0.0191, 0.0017, 0.0017, 0.0017, 0.0017, 0.0017
* 0.3017, 0.3017, 0.3017, 0.3017, 0.3017, 0.3017, 0.3017

DATA (TCY(1,J), K=1,1)

* -0.7327, -0.7327, -0.6597, -0.6126, -0.4624, -0.3180, -0.1855
* -0.0531, -0.0531, -0.0531, -0.0531, -0.0531, -0.0531, -0.0531
* 0.0156, 0.0156, 0.0156, 0.0156, 0.0156, 0.0156, 0.0156
* 0.2640, 0.2640, 0.2910, 0.2944, 0.1963, 0.0810, -0.0115
* 0.5038, 0.5038, 0.5038, 0.5038, 0.5038, 0.5038, 0.5038
* 0.7060, 0.7060, 0.7060, 0.7060, 0.7060, 0.7060, 0.7060
* 0.5014, 0.5014, 0.5014, 0.5014, 0.5014, 0.5014, 0.5014

DATA (TCZ(1,J), K=1,1)

* -0.7327, -0.7327, -0.6597, -0.6126, -0.4624, -0.3180, -0.1855
* -0.0531, -0.0531, -0.0531, -0.0531, -0.0531, -0.0531, -0.0531
* 0.0156, 0.0156, 0.0156, 0.0156, 0.0156, 0.0156, 0.0156
* 0.2640, 0.2640, 0.2910, 0.2944, 0.1963, 0.0810, -0.0115
* 0.5038, 0.5038, 0.5038, 0.5038, 0.5038, 0.5038, 0.5038
* 0.7060, 0.7060, 0.7060, 0.7060, 0.7060, 0.7060, 0.7060
* 0.5014, 0.5014, 0.5014, 0.5014, 0.5014, 0.5014, 0.5014

DATA (TPC(1,J), K=1,1)

* 0.0559, 0.1501, 0.4278, 0.5549, 0.7659, 0.7094, 0.6854
* 0.0273, 0.0408, 0.0401, 0.0172, 0.0803, 0.0828, 0.0854
* 0.0095, 0.1350, 0.3564, 0.5500, 0.6365, 0.6548, 0.6854
* 0.0230, 0.1069, 0.3064, 0.4497, 0.5521, 0.6126, 0.6854
* 0.0061, 0.1446, 0.3690, 0.5458, 0.6565, 0.6287, 0.6854
* 0.0652, 0.2422, 0.4668, 0.6133, 0.6874, 0.6840, 0.6854
* 0.2292, 0.2139, 0.6061, 0.6741, 0.7252, 0.7153, 0.6854
* 0.0401, 0.2845, 0.4925, 0.6047, 0.7066, 0.7355, 0.6854
* 0.0886, 0.2043, 0.4672, 0.6197, 0.7370, 0.7753, 0.6854
* 0.0842, 0.2098, 0.4341, 0.6554, 0.6032, 0.7623, 0.6854
* 0.0822, 0.2664, 0.4925, 0.7022, 0.7839, 0.7906, 0.6854
* 0.0840, 0.1922, 0.4524, 0.6568, 0.7583, 0.7583, 0.6854
* 0.0559, 0.1501, 0.4278, 0.5549, 0.7659, 0.7094, 0.6854

DATA (TALPHA(1,J), K=1,1)

* -0.0446, -0.1559, -0.3641, -0.5237, -0.0165, -0.5995, -0.5636
* -0.0829, -0.1411, -0.3290, -0.5048, -0.5711, -0.5610, -0.5830
* -0.1550, -0.1308, -0.2974, -0.4289, -0.5629, -0.5607, -0.5830
* -0.0293, -0.2288, -0.2666, -0.4045, -0.4979, -0.5530, -0.5830
* -0.2117, -0.1999, -0.3630, -0.4971, -0.5510, -0.5734, -0.5830
* 0.0676, 0.2762, 0.4458, 0.5490, 0.6332, 0.5640, 0.5830
* -0.0236, -0.2042, -0.4314, -0.5635, -0.6022, -0.6180, -0.5830

460
<table>
<thead>
<tr>
<th>C</th>
<th>DATA (((TCM(1),J,K), 1=1,8), J=1,13, K=1,1) /</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 0</td>
<td>0.0132, 0.0030, 0.0134, 0.0267, 0.0390, -0.0171, 0.0080</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0077, -0.0169, -0.0951, -0.0604, 0.0001, -0.0007, 0.0079</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0006, -0.0604, -0.0133, -0.0079, -0.0045, 0.0127, 0.0079</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0045, -0.0309, -0.0351, -0.0267, -0.0265, -0.0086, 0.0079</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0021, -0.0123, -0.0268, -0.0267, -0.0215, -0.0097, 0.0080</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0034, -0.0603, -0.0099, 0.0511, 0.0109, 0.0332, 0.0075</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0137, 0.0150, 0.0133, 0.0059, -0.0074, -0.0069, 0.0080</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0164, -0.0292, 0.0310, -0.0275, 0.0158, 0.0006, 0.0090</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>DATA (((TCM(1),J,K), 1=1,8), J=1,13, K=2,2) /</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 0</td>
<td>0.0234, -0.0234, -0.0191, -0.0135, -0.0097, -0.0029, -0.0005, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0461, -0.0401, -0.0460, -0.0238, -0.0189, 0.0079, 0.0045, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0232, -0.0232, -0.0270, -0.0120, -0.0092, -0.0017, -0.0011, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0035, -0.0085, -0.0079, -0.0039, -0.0060, -0.0043, 0.0002, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0425, 0.0425, 0.0444, 0.0421, 0.0351, 0.0192, 0.0028, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0543, 0.0543, 0.0439, 0.0408, 0.0310, 0.0512, 0.0082, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0249, 0.0249, 0.0199, 0.0165, 0.0107, 0.0049, 0.0050, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0144, -0.0104, 0.0120, -0.0104, -0.0119, -0.0175, -0.0100, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0441, -0.0441, -0.0415, -0.0269, -0.0193, -0.0151, -0.0054, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0125, -0.0125, -0.0101, -0.0058, -0.0081, -0.0044, 0.0012, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0332, 0.0332, 0.0270, 0.0185, 0.0128, 0.0062, 0.0040, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0248, 0.0248, 0.0241, 0.0184, 0.0138, 0.0070, 0.0083, -0.0004</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0234, -0.0234, -0.0191, -0.0135, -0.0097, -0.0029, -0.0006, -0.0004</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>DATA (((TCM(1),J,K), 1=1,8), J=1,13, K=1,1) /</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 0</td>
<td>0.0034, -0.0031, -0.0120, -0.0252, -0.0340, -0.0257, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0008, -0.0026, -0.0143, -0.0242, -0.0344, -0.0232, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0006, 0.0010, -0.0113, -0.0167, -0.0132, -0.0149, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0030, -0.0025, -0.0014, -0.0156, -0.0155, -0.0081, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0003, -0.0004, -0.0091, -0.0083, -0.0108, -0.0099, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0057, -0.0112, -0.0104, -0.0071, -0.0061, -0.0077, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0025, -0.0010, -0.0070, -0.0036, -0.0096, -0.0131, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0023, -0.0072, -0.0072, -0.0101, -0.0058, -0.0090, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0020, -0.0052, -0.0080, -0.0035, -0.0166, -0.0171, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0061, -0.0094, -0.0121, -0.0210, -0.0231, -0.0230, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0005, -0.0067, -0.0175, -0.0233, -0.0205, -0.0260, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0034, -0.0076, -0.0165, -0.0272, -0.0304, -0.0250, -0.0123</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0034, -0.0051, -0.0120, -0.0252, -0.0340, -0.0257, -0.0123</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>DATA (((TCM(1),J,K), 1=1,8), J=1,13, K=2,2) /</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 0</td>
<td>-0.0066, -0.0014, -0.0012, -0.0058, -0.0116, -0.0098, -0.0012</td>
</tr>
<tr>
<td>* 0</td>
<td>0.0018, 0.0015, 0.0005, 0.0047, 0.0116, 0.0070, -0.0012</td>
</tr>
<tr>
<td>* 0</td>
<td>-0.0035, -0.0052, -0.0035, -0.0057, -0.0060, -0.0063, -0.0012</td>
</tr>
</tbody>
</table>

462
* 0., 0.0024, -0.0014, -0.0005, -0.0044, -0.0049, -0.0008, -0.0012,
* 0., 0.0056, 0.0032, -0.0032, -0.0003, 0.0002, 0.0012, -0.0012,
* 0., 0.0040, 0.0063, -0.0047, -0.0014, 0.0016, 0.0132, -0.0012,
* 0., 0.0051, 0.0013, 0.0002, 0.0017, 0.0028, 0.0001, -0.0012,
* 0., 0.0000, -0.0024, -0.0026, -0.0036, -0.0019, 0.0000, -0.0012,
* 0., -0.0001, 0.0004, -0.0008, 0.0008, -0.0027, -0.0001, -0.0012,
* 0., 0.0010, 0.0004, -0.0016, -0.0062, -0.0061, -0.0031, -0.0012,
* 0., 0.0013, 0.0010, 0.0071, -0.0083, -0.0078, -0.0055, -0.0012,
* 0., -0.0036, -0.0004, -0.0038, -0.0016, -0.0109, -0.0113, -0.0012,
* 0., -0.0066, -0.0014, -0.0012, -0.0058, -0.0116, -0.0098, -0.0012

C DATA (TETAI, I=1,8) / 0., 5., 15., 30., 45., 60., 75., 90. /
C DATA (TETAI, I=1,13) / 0., 30., 60., 90., 120., 150., 180., 210.,
   240., 270., 300., 330., 360. /
C DATA (TPC(I), I=1,2) / 5., 75. /
C **** CALCULATE THE AERO COEFFICIENTS ****
C CX = TBLU3(BETA, ALPHA, PC, TETAI, TALPHA, TPC, TCX,
   1,1,1,8,13,2,8,13,2)
CY = TBLU3(BETA, ALPHA, PC, TETAI, TALPHA, TPC, TCY,
   1,1,1,8,13,2,8,13,2)
CZ = TBLU3(BETA, ALPHA, PC, TETAI, TALPHA, TPC, TCZ,
   1,1,1,8,13,2,8,13,2)
CL = TBLU3(BETA, ALPHA, PC, TETAI, TALPHA, TPC, TCL,
   1,1,1,8,13,2,8,13,2)
CM = TBLU3(BETA, ALPHA, PC, TETAI, TALPHA, TPC, TCN,
   1,1,1,8,13,2,8,13,2)
CN = TBLU3(BETA, ALPHA, PC, TETAI, TALPHA, TPC, TCN,
   1,1,1,8,13,2,8,13,2)
C RETURN
END
SUBROUTINE COSUIR(ANG, DCOS)
DIMENSION ANG(3), DCOS(3,3)
C
CALCULATES THE EULER ANGLES FROM THE DIRECTION COSINE MATRIX
C
ANG(1) = ARTAN2(DCOS(1,2), DCOS(1,1))
C
ANG(2) = ASIN(-DCOS(1,3))
C
ANG(3) = ARTAN2(DCOS(2,3), DCOS(3,3))
C
RETURN
END
FUNCTION DET3(D11, D12, D13, D21, D22, D23, D31, D32, D33)

C

FUNCTION FOR COMPUTING THE VALUE OF A 3 x 3 DETERMINANT

C

DET3 = D11 * (D22 * D33 - D23 * D32) - D12 * (D21 * D33 - D23 * D31) + D13 * (D21 * D32 - D22 * D31)

RETURN

END
SUBROUTINE DIRCOS (DCOS, ANG)
DIMENSION DCOS(3,3), ANG(3)
C
C DESIGNED BY C. L. WESI
C LAST MODIFIED - DECEMBER 6, 1980
C
C CALCULATES THE DIRECTION COSINE MATRIX FROM THE EULER ANGLES
C
SINPSI = SIN(ANG(1))
COSPSI = COS(ANG(1))
SINTHE = SIN(ANG(2))
COSTHE = COS(ANG(2))
SINPHI = SIN(ANG(3))
COSPHI = COS(ANG(3))
C
DCOS(1,1) = COSTHE * COSPSI
DCOS(1,2) = COSTHE * SINPSI
DCOS(1,3) = -SINTHE
DCOS(2,1) = SINPHI * SINTHE * COSPSI -
* COSPHI * SINPSI
DCOS(2,2) = SINPHI * SINTHE * SINPSI +
* COSPHI * COSPSI
DCOS(2,3) = SINPHI * COSTHE
DCOS(3,1) = COSPHI * SINTHE * COSPSI +
* SINPHI * SINPSI
DCOS(3,2) = COSPHI * SINTHE * SINPSI -
* SINPHI * COSPSI
DCOS(3,3) = COSPHI * COSTHE
C
RETURN
END
C SUBROUTINE DISECT — ENTRY POINT OF COMPASS PROG PICKER
IDENT PICKER
LIST L,R,G,W
ENTRY DISECT
USE /WORD/

ITRCOP bss 4
ISQUAD bss 1
USE 0

DISECT bss 1
SB1 -15
SB2 3
SA2 ISQUAD
MX3 -15

LOOP BX7 -X2+X3
BX6 -X7
SA6 ITROOP+B2
LX2 X2,B1
SB2 B2-1
GE B2,LOOP
EQ DISECT
END
SUBROUTINE KARATE (EADOT, WBODY, EULER)
DIMENSION EADOT(3), WBODY(3), EULER(3)
DATA PSID / 0 /

C CALCULATES THE EULER ANGLE RATES FROM THE BODY AXIS ANGULAR VELOCITY VECTOR
C
C ******************* CALLING SEQUENCE *******************
C
C ** OUTPUT **
C EADOT(3) - EULER ANGLE RATES -- YAW, PITCH, ROLL -- (RAD/SEC)
C
C ** INPUT **
C WBODY(3) - X, Y, Z BODY AXIS ANGULAR VELOCITY COMPONENTS (RAD/SEC)
C EULER(3) - EULER ANGLES (RAD)
C
C CP = COS(EULER(2))
SP = SIN(EULER(2))
CR = COS(EULER(5))
SR = SIN(EULER(3))

EADOT(2) = WBODY(2)*CR - WBODY(3)*SR
IF(CP.NE.0.) PSID = (WBODY(2)*SR + WBODY(3)*CR)/CP
EADOT(1) = PSID
EADOT(3) = WBODY(1) + PSID*SP
C
RETURN
END
FUNCTION FSW(A,B,C,D)

THIS FUNCTION IS DESIGNED AS FOLLOWS -

FSW = B IF A IS LESS THAN ZERO
FSW = C IF A IS EQUAL TO ZERO
FSW = C IF A IS GREATER THAN ZERO

10 IF (A) 10,20,30
   FSW=B
   GO TO 40
20 FSW=C
   GO TO 40
30 FSW=D
40 RETURN
END
SUBROUTINE LAG (CSDOT,CSCOM,CSP0S,CSTRM,TC,TIMC,TO)
C
C RESPONSE OF A FIRST ORDER LAG FUNCTION TO A CONTROL SURFACE STEP
C INPUT. TO MAY BE USED TO MECHANIZE A TIME DELAY, WITH THE CONTROL
C SURFACE REMAINING AT ITS TRIM POSITION UNTIL TIME TO.
C
C DEFINITION OF CALLING ARGUMENTS
C
CSOCT - CONTROL SURFACE RATE (DEG/SEC) --- OUTPUT ---
C CSCOM = CONTROL SURFACE COMMANDED POSITION (DEG)
C CSP0S - DEFLECTION OF THE CONTROL SURFACE FROM ITS
C TRIM POSITION (DEG)
C CSTRM - CONTROL SURFACE TRIM POSITION (DEG)
C TC = TIME CONSTANT (SEC)
C TIME = SIMULATION TIME (SEC)
C TO = TIME DELAY AFTER WHICH THE CONTROL SURFACE RATE IS
C CALCULATED (SEC)
C
IF(TIME-TO.GE.0) GO TO 10
CSODT = 0
GO TO 20
C
10 CSDOT = (CSCOM - (CSP0S+CSTRM))/TC
C
20 RETURN
END
SUBROUTINE PCAERO (FLIFT, FDRAG, FMDOT, SCT,
   *   SW, XPC, UPC, TLS, DTI, TDU, VOL, UVL,
   *   CT, CN, CM, FD, B, STI, RFS, FLA, TLA, TEM)

THIS ROUTINE DETERMINES PARACHUTE AERODYNAMIC FORCES ACTING ON
THE PARACHUTE

********** PCAERO OUTPUTS **********

FLIFT(3) - X,Y,Z EARTH SYSTEM LIFT COMPONENTS ACTING ON
THE PARACHUTE (LB)
FDRAG(3) - X,Y,Z EARTH SYSTEM DRAG COMPONENTS ACTING ON
THE PARACHUTE (LB)
FMDOT(3) - X,Y,Z EARTH SYSTEM MASS ACQUISITION FORCE
COMPONENTS ACTING ON THE PARACHUTE (LB)
SCT - COMPUTED TANGENTIAL DRAG AREA (FT**2)

********** PCAERO INPUTS **********

SW - FLAG TO INDICATE AERODYNAMIC CALCULATION MODE
1 = FROM PARACHUTE LAUNCH TO LINESRETCH
2 = DURING INFLATION
3 = DURING REEFING
4 = AFTER REEFING
5 = PARACHUTE INFLATED

XPC(3) - X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE
PARACHUTE (FT)

UPC(3) - X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE
PARACHUTE (FT/SEC)

TLS - TIME AT LINESRETCH (SEC)

DTI - THE TIME DURATION OF PARACHUTE CANOPY INFLATION (SEC)

TDL - REEFING DURATION (SEC)

VOL - VOLUME OF THE FILLED CANOPY (FT**3)

CT(3) - CONSTANTS USED IN THE EQUATION THAT CALCULATES THE
TANGENTIAL DRAG AREA

CN(3) - CONSTANTS USED IN THE EQUATION THAT CALCULATES THE
NORMAL DRAG AREA

CM(2) - CONSTANTS USED IN THE MACH EFFECTS EQUATION

FD - MAE TO FREE STREAM RATIO

B - CONSTANT USED IN THE EQUATION FOR CALCULATING
SCD OF THE PARACHUTE

SII - INFLATED PARACHUTE DRAG AREA (FT**2)

RFS - PRODUCT OF REFERENCE AREA AND TANGENT FORCE

COEFFICIENT WHEN REEFED (FT**2)

FLA - PARACHUTE MODE FLAG

5 = LINES SEVERED

TLA - PARACHUTE LAUNCH TIME / LINE SEVERING TIME (SEC)

TEM - TIME DURATION FOR PARACHUTE EMERGENCE / LINE
SEVERENCE (SEC)

CALLING SEQUENCE DIMENSIONS .......

DIMENSION FLIFT(3), FDRAG(3), FMDOT(3), XPC(3), UPC(3), UVL(3),
   *   CT(3), CN(3), CM(2)

INTERNAL DIMENSIONS .......

471
DIMENSION UO(3), UW(3), TEMP1(3), TEMP2(3), UVV(3), UVLIFT(3)

COMMON /CTIME/ TIME
COMMON /CIOA/ IREAD, IWRITE, IDIAG

DATA PI02 / 1.57080 /

ZERO THE MASS ACQUISITION FORCE .......

DO 5 I = 1, 3
5 FMOUT(I) = 0

------ DETERMINE THE AERODYNAMIC PARAMETERS ------

CALL ATMUS (VS, RHO, XPC(3), UW, 0, 0, 0)
UO(1) = UPC(1) - UW(1)
UO(2) = UPC(2) - UW(2)
UO(3) = UPC(3) - UW(3)
VBAR = SQRT(UO(1)**2 + UO(2)**2 + UO(3)**2)
VMACH = VBAR / VS

------ CALCULATE ALPHA (the angle whose cosine IS THE CHUTE VELOCITY UNIT VECTOR DOTTED ONTO THE LINE UNIT VECTOR) ------

DO 10 I = 1, 3
10 UVV(I) = UO(I) / VBAR

IF THE LINES HAVE BEEN SEVERED .......

FACTOR = 1.
IF (PLANE .LE. 5.) GO TO 15
IF (ILA .EQ. 0) TLA = TIME
FLIFT(1) = FLIFT(2) = FLIFT(3) = ALPHA = SINA = 0
COSA = 1.
DELTA = TIME - TLA
IF (DELTA .GE. TEM) GO TO 90
FACTOR = SIN(DELTA*PI02/TEM)
GO TO 60

OUT THE VELOCITY UNIT VECTOR ONTO THE LINE UNIT VECTOR .......

15 CALL UDIPRD (CALPHA, UVV, UVL, 3)
1F (ABS(CALPHA) .GE. 1.0) CALPHA = SIGN(1.0, CALPHA)

CALPHA = ABS(CALPHA)
SINA = SIN(CALPHA)
COSA = COS(CALPHA)

------ CALCULATE THE MASS ACQUISITION FORCE ------

CALCULATE THE MASS ACQUISITION FORCE IF SW = 2 OR 4 ......

IF (SW .NE. 2. .AND. SW .NE. 4.) GO TO 40

RHUS = RHO*(((1.0 + 0.5*VMACH**2)**2.5)
PCNIF = ((TIME - T00) - TLS)/DT1

472
PLNT = PCNTF
IF (PCNT.GT.0.5) PCNT = 0.5
DOTM = 0.01*PCNT*VOL*KMOS/DTl
DO 30 I=1,3
30 FMUOT(I) = -DOTM*UG(I)

C *******************************************
C ** LOGIC TO CHOOSE THE PROPER EQUATIONS **
C *******************************************

40 GO TO (50,60,70,80,90), SW

C ---- EQUATIONS USED PRIOR TO LINESTRETCH ----
C AND AFTER THE LINES ARE SEVERED
C
50 SCl = 6 * STI
SCN = 0.0
GO TO 90

C ---- EQUATIONS USED WHEN THE CHUTE IS INFLATING ----
C CALCULATE THE WAKE TO FREE STREAM RATIO .......
C
60 FC = FD
IF(VMACH.GT.1.0) FC = (1.0+(CM(1)+CM(2)*(VMACH-1.0))*(VMACH-1.0))*FD
C CALCULATE THE VARIABLES USED IN DETERMINING THE NORMAL AND TANGENTIAL
C LIFT AREAS DURING CHUTE INFLATION .......
C
SCIA = STI + ((CT(3)*ALPHA+CT(2))*ALPHA*CT(1))*ALPHA
SCOLS = 8 * SCITIA
C
SL = SCDLS + (SCIA-SCOLS)*PCNTF*FC
SCN = ((CN(3)*ALPHA+CN(2))*ALPHA+CNI)*ALPHA*PCNTF*FC
GO TO 90

C ---- EQUATIONS USED WHEN THE CHUTE IS REELED ----
C
70 SCl = RFS
SCN = 0.0
GO TO 90

C ---- EQUATIONS USED WHEN THE CANOPY IS FILLED ----
C
80 SCl = STI + ((CT(3)*ALPHA+CT(2))*ALPHA*CT(1))*ALPHA
SCN = ((CN(3)*ALPHA+CN(2))*ALPHA+CN(1))*ALPHA
C
C *******************************************
C ** CALCULATE THE LIFT AND DRAG AREAS **
C *******************************************

90 SLL = ABS(SCN*COSA - SCT*SINA)
SCG = ABS(SCN*SINA + SCT*COSA)
C
C *******************************************
C ** CALCULATE THE ZAXTH AXIS LIFT COMPONENTS **
C *******************************************
COMPUTE THE UNIT VECTOR IN THE DIRECTION OF LIFT ......

IF(FLA.EQ.5.) GO TO 120
CALL CRSPRO (TEMP1, UO, UVL)
CALL CRSPRO (TEMP2, UO, TEMP1)
RESULT = SQRT(TEMP2(1)**2 + TEMP2(2)**2 + TEMP2(3)**2)
DO 100 I=1,3
100 UVLIFT(I) = TEMP2(I)/RESULT
E = 0.5*RHO*SCL*VBAR*VBAR
DO 110 I=1,3
110 FLIFT(I) = -E * UVLIFT(I)

** CALCULATE THE EARTH AXIS DRAG COMPONENTS **

120 E = 0.5*RHO*SCD*VBAR*VBAR
DO 130 I=1,3
130 FDAMG(I) = -E * UVM(I) * FACTOR
RETURN
END
SUBROUTINE RATIO (NV, GV, TV, RAT, NC)
DIMENSION TV(1)

C

IF (NV .EQ. 1) GO TO 10
IF (GV - TV(1) .GT. 0) GO TO 30
10 NC = 1
20 RAT = 0
GO TO 60
30 DO 40 NCNT = 2, NV
   NC = NCNT
   IF (GV - TV(NC)) 50, 60, 40
40 CONTINUE
GO TO 20
50 RAT = (TV(NC) - GV) / (TV(NC) - TV(NC-1))
60 RETURN
END
FUNCTION RLIM(AA,bb,CC)

FUNCTION WHICH LIMITS THE VALUE OF VARIABLE AA TO WITHIN A RANGE DEFINED BY VARIABLES BB AND CC

IF(AA.LT.BB)GO TO 10
IF(AA.GT.CC)GO TO 20
RLIM=AA
GO TO 30

10 RLIM=BB
GO TO 30

20 RLIM=CC

30 RETURN

END
SUBROUTINE ROTATE1 (BMI, BPI, DC)
DIMENSION BMI(3), BMIT(3), BPI(3), BPIT(3), DC(3,3)

TRANSFORMS INERTIA PROPERTIES FROM ONE AXIS SYSTEM
TO ANOTHER THROUGH A DIRECTION COSINE MATRIX......

TRANSFORM THE MOMENTS OF INERTIAS......

DO 10 I=1,3
  BMIT(I) = DC(1,1)*BMI(I) + DC(1,2)*BMI(2) +
            DC(1,3)*BMI(3) - (DC(1,1)*DC(1,2)*BPI(I) +
            DC(1,3)*DC(1,2)*BPI(2) + DC(1,2)*DC(1,3)*BPI(3))*2.0
  10 CONTINUE

TRANSFORM THE PRODUCTS OF INERTIA......

  BPI(1) = -DC(1,1)*DC(2,1)*BMI(1) - DC(1,2)*DC(2,2)*BMI(2) -
           DC(1,3)*DC(2,3)*BMI(3) + (DC(1,1)*DC(2,2) +
           DC(1,2)*DC(2,1)*BPI(1) + (DC(1,1)*DC(2,3) +
           DC(1,3)*DC(2,2)*BPI(2) + (DC(1,2)*DC(2,3) +
           DC(1,3)*DC(2,3))*BPI(3)
  BPI(2) = -DC(1,1)*DC(3,1)*BMI(1) - DC(1,2)*DC(3,2)*BMI(2) -
           DC(1,3)*DC(3,3)*BMI(3) + (DC(1,1)*DC(3,2) +
           DC(1,2)*DC(3,1)*BPI(1) + (DC(1,1)*DC(3,3) +
           DC(1,3)*DC(3,1)*BPI(2) + (DC(1,2)*DC(3,3) +
           DC(1,3)*DC(3,3))*BPI(3)
  BPI(3) = -DC(2,1)*DC(3,1)*BMI(1) - DC(2,2)*DC(3,2)*BMI(2) -
           DC(2,3)*DC(3,3)*BMI(3) + (DC(2,1)*DC(3,2) +
           DC(2,2)*DC(3,1)*BPI(1) + (DC(2,1)*DC(3,3) +
           DC(2,3)*DC(3,1)*BPI(2) + (DC(2,2)*DC(3,3) +
           DC(2,3)*DC(3,3))*BPI(3)

DO 20 I=1,3
  BMI(I) = BMIT(I)
  20 BPI(I) = BPI(I)

RETURN
END
FUNCTION TBLU3(X1,Y1,Z1,X,Y,Z,F3,NDX,NDY,NDZ,NX,NY,NZ,MX,MY,MZ)

PURPOSE
TBLU3 performs table search and Lagrange polynomial interpolation of user-defined degree on 3 independent variables.

USAGE
DIMENSION X(NX),Y(NY),Z(NZ),F3(MX,MY,MZ)
V = TBLU3(X1,Y1,Z1,X,Y,Z,F3,NDX,NDY,NDZ,NX,NY,NZ,MX,MY,MZ)

INPUT PARAMETERS
X1,Y1,Z1 - POINT TO INTERPOLATE FOR
X,Y,Z - ARRAYS OF INDEPENDENT VARIABLES
F3 - 3D ARRAY OF DEPENDENT VARIABLE
NDX,NDY,NDZ - DEGREE OF INTERPOLATION FOR EACH DIMENSION
NX,NY,NZ - IABS OF EACH IS THE NUMBER OF DATA POINTS IN THE RESPECTIVE X, Y OR Z ARRAY. IF NEGATIVE, NEAREST END POINT IS TO BE USED UPON EXTRAPOLATION
MX,MY,MZ - DIMENSIONAL CONSTANTS FOR F3 ARRAY

OUTPUT PARAMETERS
V - RESULT OF TABLE SEARCH AND INTERPOLATION
SUCCESS V = INTERPOLATED VALUE
ERROR V = INDEFINITE VALUE WHERE RIGHTMOST DIGIT DEFINES THE ERROR DETECTED
1 DATA VALUES WITHIN X, Y OR Z ARE NOT DISTINCT
2 ONE OF NDX, NDY OR NDZ IS LESS THAN ZERO
3 ONE OF NX, NY OR NZ IS ZERO
4 EITHER MX,LT,IAABS(NX) OR MY,LT,IAABS(NY)

TEST FOR USER ERRORS
TBLU3 = 0
IF (NDX,LT,0) OR (NDY,LT,0) OR (NDZ,LT,0) TBLU3 = ERR2
IF (NX,EQ,0) OR (NY,EQ,0) OR (NZ,EQ,0) TBLU3 = ERR3
IF (MX,LT,IAABS(NX)) OR (MY,LT,IAABS(NY)) TBLU3 = ERR4
IF (TBLU3,NL,0) GO TO 50

SET UP INITIAL PARAMETERS
X2 = X1
Y2 = Y1
Z2 = Z1
MNX = NDX
MNY = NDY
MNZ = NDZ

SEARCH FOR X1, Y1 AND Z1 IN TABLES
IX = SEARCH(X2,X,MNX,NX,1)
IY = SEARCH(Y2,Y,MNY,NY,1)
IZ = SEARCH(Z2,Z,MNZ,NZ,1)

TEST FOR EXACTNESS IN 1 OR MORE DIMENSIONS
IW = IX+IY+IZ
IF (IW,EQ,1) GO TO 40
IF (IW,NL,1) GO TO 10
TBLU3 = F3(IW,J,K)
GO TO 50
10 IF (IX.EQ.0) GO TO 20
   X2 = X(I)
   MDX = 0
20 IF (IY.EQ.0) GO TO 30
   Y2 = Y(J)
   MDY = 0
30 IF (IZ.EQ.0) GO TO 40
   Z2 = Z(K)
   MDZ = 0
C INTERPOLATE
40 TBLU3 = TERP3(X2,Y2,Z2,X,Y,Z,F3,MDX,MDY,MDZ,FX,MY,MZ,I,J,K)
50 RETURN
END
SUBROUTINE TLU(NX,NY,NZ,ROW,COLM,PAGE,XG,YG,ZG,ANS,NTAB)
DIMENSION IB(1),ROW(NX),COLM(NY),PAGE(NZ),ANS(6)
FIRSTF (X,Y,Z) = X - L*(X - Y)

WHAT BALL PARK IS THE POINT IN ........

CALL RATIO (NX,XG,KOM,RATX,1)
CALL RATIO (NY,YG,COM,RATY,J)
CALL RATIO (NZ,ZG,PAGE,RATZ,K)

IT'S JUST PRIOR TO THE 'Ith' ROW, 'Jth' COLUMN AND THE 'Kth' PAGE ....

NOTE - IF ONE OF THE INCOMING ARGUMENTS IS OUT OF THE TABLE BOUNDS,
THE APPROPRIATE VALUE OF RATX, RATY, RATZ, WILL BE ZERO ....

WHAT IS THE LOCATION OF THE NEXT HIGHER POINT ....

NYX = NX*NY
NP = I + NX*(J-1) + NYX*(K-1)

LET'S INTERPOLATE FROM AS MANY AS NTAB TABLES......

DO 50 L=1,NTAB

C = E = 0

WHERE IS THE POINT BETWEEN ROWS ....

CALL UNPACK (NP,IB,C1,C2)
A = FIRSTF (C2,C1,RATX)

IF WE ARE IN THE FIRST COLUMN JUMP TO STATEMENT 10 ....

IF (J.EQ.1) GO TO 10

JUMP TO THE NEXT LOWER COLUMN ......

NP = NP - NX

BETWEEN ROWS IN THE ADJACENT COLUMN ......

CALL UNPACK (NP,IB,C1,C2)
B = FIRSTF (C2,C1,RATX)

JUMP TO THE NEXT LOWER PAGE.......

IF (J.EQ.1) GO TO 40

NP = NP - NY

IF WE ARE NOT ON THE FIRST PAGE JUMP TO STATEMENT 20 ....

IF (K.NE.1) GO TO 20

IF WE ARE IN THE FIRST COLUMN, JUMP TO STATEMENT 40 ......

IF (J.EQ.1) GO TO 40

JUMP TO THE NEXT HIGHER COLUMN ......
C

NP = NP + NX
GO TO 40

C
IF WE ARE IN THE FIRST COLUMN, JUMP TO STATEMENT 30 .......

C
40 IF (J.EQ.1) GO TO 30

C
BETWEEN ROWS ......

C
CALL UNPACK (NP,IB,CL,C2)
C = FIRSTF (C2,CL,RAI)

C
JUMP TO THE NEXT HIGHER COLUMN ......

C
NP = NP + NX

C
BETWEEN ROWS AGAIN IN THE NEXT HIGHER COLUMN ....

C
50 CALL UNPACK (NP,IB,CL,C2)
D = FIRSTF (C2,CL,RAI)

C
BETWEEN COLUMNS .......

C
E = FIRSTF (D,CL,RAT)
40 F = FIRSTF (A,B,RAT)

C
NOW BETWEEN PAGES ....

C
AMS(I) = FIRSTF (F,E,RAT)

C
MOVE TO THE BEGINNING OF THE NEXT TABLE ....

C
50 NP = NP + NXY + NZ*NXY

C
THAT'S IT, LET'S GO HOME ....

C
RETURN

END
SUBROUTINE UNPACK (NP, IB, WORD1, WORD2)
DIMENSION IB(1), JWORD(6)
COMMON /WORD/ ITROOP(4), ISQUAD
DATA FOUR /4.0/, BN /16383./, CMIN /-1.5/, RANGE /3.0/
NPRIOR = 1
PNIS = NP
WORD = PNTS/FOUR
NWORD = WORD + 0.1
NSUBWRD = (WORD - FLOAT (NWORD)) * FOUR
IF (NSUBWRD.EQ.0) NPRIOR = 2
IF (NPRIOR .EQ. 1) GO TO 20
ISQUAD = IB(NWORD)
CALL OISECT
DU 10 I=1,4
10 JWORD(I) = ITROOP(I)
20 ISQUAD = IB(NWORD + 1)
IF (NSUBWRD .EQ. 0) ISQUAD = IB(NWORD)
CALL OISECT
DO 30 I=1,4
30 JWORD(I+4) = ITROOP(I)
IF (NSUBWRD .EQ. 0) NSUBWRD = 4
IWORD1 = JWORD(NSUBWRD + 3)
IWORD2 = JWORD(NSUBWRD + 4)
WORD1 = CMIN + (FLOAT(IWORD1)/BN)*RANGE
WORD2 = CMIN + (FLOAT(IWORD2)/BN)*RANGE
RETURN
END
SUBROUTINE VECAYZ (TRANS, VEC, ORIGIN, DC, IOPT)
DIMENSION TRANS(3), VEC(3), ORIGIN(3), DC(3,3), DIFF(3)
C
C TRANSFORMS VECTORS FROM ONE REFERENCE FRAME INTO ANOTHER ......
C
***** CALLING ARGUMENTS *****
C
TRANS(3) - TRANSFORMED VECTOR (OUTPUT)
VEC(3) - INPUT VECTOR
ORIGIN(3) - SECONDARY SYSTEM ORIGIN IN THE PRIMARY SYSTEM
DC(3,3) - DIRECTION COSINE MATRIX
IOPT - FLAG TO DETERMINE TYPE OF TRANSFORMATION
       1 = FROM PRIMARY TO SECONDARY
       2 = FROM SECONDARY TO PRIMARY
C
IF(IOPT.EQ.2) GO TO 20
C
DO 10 I=1,3
10 DIFF(I) = VEC(I) - ORIGIN(I)
CALL MATMPY (TRANS, DC, DIFF, 3, 3, 1)  
GO TO 40
C
20 CALL MATMPY (TRANS, DC, VEC, 3, 3, 1)
DO 30 I=1,3
30 TRANS(I) = ORIGIN(I) + TRANS(I)
C
40 RETURN
END
SUBROUTINE VELXYZ (U,USEC,XPT,WSEC,DSI)
DIMENSION U(3),USEC(3),XPT(3),WSEC(3),UPTSEC(3),
 TEMP(3),DSI(3,3)

COMPUTES THE EARTH SYSTEM VELOCITY VECTOR OF A POINT
DISPLACED FROM THE ORIGIN OF A SECONDARY COORDINATE SYSTEM

********** CALLING ARGUMENTS **********

U(3) - X,Y,Z EARTH SYSTEM VELOCITY VECTOR OF A POINT
DISPLACED FROM THE ORIGIN OF A SECONDARY SYSTEM
(FT/SEC) — OUTPUT —
USEC(3) - X,Y,Z BODY AXIS VELOCITY VECTOR OF THE SECONDARY
SYSTEM (FT/SEC)
XPT - X,Y,Z BODY AXIS POSITION VECTOR OF THE DISPLACED
POINT IN THE SECONDARY SYSTEM (FT)
WSEC - X,Y,Z BODY AXIS ANGULAR VELOCITY VECTOR OF THE
SECONDARY SYSTEM (RAD/SEC)
DSI(3,3) - SECONDARY TO EARTH SYSTEM DIRECTION COSINE MATRIX

CALCULATE WSEC X XPT .......

CALL CRSRDK (TEMP,WSEC,XPT)

DETERMINE USEC + (WSEC X XPT) .......

DO 10 I=1,3
  10 USEC(I) = USEC(I) + TEMP(I)

TRANSFORM THE VELOCITY VECTOR FROM THE SECONDARY TO THE
EARTH SYSTEM .......

CALL MATMPY (U,DSI,UPTSEC,3,3,1)

RETURN
END
This Appendix contains the FILOAD Input Data. FILOAD is a program which creates a random access file from input data that defines the variable names in the calling sequence for each standard component. This random access file is employed by the Model Generation program to build the model defined by the user in the Model Generation input data.
NEW FILE
FILE NAME=EASIEST
AGINPT = 8
MT  = AMI 3 API 3 FAB 3 TAB 3 FAU 3 TAU 3 TRM 3
ABOUTP = 4
UAD  = SXAB 3 SWAB 3 SEAB 3 S
ABTABS = 0
SYMBOL, AD = 101
ATINPT = 29
AW  = b C S XCP ZLP AMI 3 API 3
THR  = AIL ELE RUD XEN 3 END 3 TAL TVE
FRA  = 1 RRA 3 1 FCA 3 1 TCA 3 1 FDA 3 1 TDA 3 1 FRA 3 2 TMA 3 2 CPF
FCA  = 2 TCA 3 2 FDA 3 2 TUA 3 2 CPF
AEOUTP = 9
UAP  = 3 SXAP 3 SWAP 3 SEAP 3 STRM 4 SALP BET VM
ALT  = ALTABS = 0
SYMBOL, AE = 101
AFINPT = 6
CJO  = C1 C2 C3 C4 C5
AFOUTP = 1
S  = 2
AFTABS = 0
SYMBOL, AF = 101
AGINPT = 5
H  = WIN 3 BP TE SW
AGOUTP = 2
V5  = RH0
AGTABS = 0
SYMBOL, AG = 101
AMINPT = 14
FL  = PRT EXP GAP GXN GYL GZL DRP
URN  = KDL DR GX GY GZ
AMOUTP = 4
URE  = RAD PTS PTI
AMTABS = 0
SYMBOL, AM = 101
APIINPT = 11
UP  = XFC 3 PA EPL 3 ZEM SKP 3 UST 3 EST 3
WST  = 3 XAP 3 LAP 3
APOUTP = 6
F2  = 3 1 T2 3 1 SW ALP CX CZ
APTABS = 2
TCA  = 20 1
TLE  = 20 1
SYMBOL, AP = 101
ASINPT = 19
OFF = UP ZWS XEM 3 CODX ECX ECY ECZ
CLP  = CM4 CNR S SRP 3 UST 3 EST 3 WST 3
OSA  = 3 SRA 3 RON
ASSOUTP = 17
F2  = 3 1 T2 3 1 ALP BET VM Q CX CY
CZ  = CC CM CN EXL EXA GEN 3 TCZ 2U
HU  = 0
ASTABS = 1
TAE  = 20 1
SYMBOL, AS = 101
AVINPT = 14

486
<table>
<thead>
<tr>
<th>U</th>
<th>3</th>
<th>W</th>
<th>3</th>
<th>ALT</th>
<th>EA</th>
<th>3</th>
<th>ID</th>
<th>1</th>
<th>VS</th>
<th>ALS</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>VV</td>
<td>WW</td>
<td>PW</td>
<td>QW</td>
<td>RW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVOUTP</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UG</td>
<td>3</td>
<td>WG</td>
<td>3</td>
<td>ID</td>
<td>2</td>
<td>QM</td>
<td>2</td>
<td>RW</td>
<td>2</td>
<td>CAL</td>
<td>SAL</td>
</tr>
<tr>
<td>ALP</td>
<td>VI</td>
<td>BE</td>
<td>WP</td>
<td>UP</td>
<td>EU</td>
<td>3</td>
<td>SIG</td>
<td>QC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**AVIABs = 0**

**SYMBOL, AV = 101**

**CL1NPT = 16**

| SW | PC | CW | CM1 | 3 | CPI | 3 | CLP | CMQ | CRN |
| XSP | 3 | FAB | 3 | TAB | 3 | FDO | 3 | TOO | 3 | FAU | 3 | TAU | 3 | TRM | 3 |

**CL2OUTP = 29**

| UCP | 3 | SSCP | 3 | SWCP | 3 | SECP | 3 | SSCO | SSG | SGX | GY |
| UG | DR | FAD | 3 | TAD | 3 | WT | S | B | C |
| CIN | 4 | CAX | CYY | C 2 | CL | CM | CN | ALP |
| BCP | VM | Q | ALT | SEP |   |   |   |   |   |    |   |

**CETABS = 0**

**SYMBOL, CL = 101**

**CL2INPT = 20**

| M | 3 | TR | TMI | AD | DA | GR | BDZ | CS |
| CA | P1D | D6 | GFI | DR | DRS | GB | CBS |
| DUL | UG | UG | US | H |   |   |   |   |
| CLOUTP | 6 |   |   |   |   |   |   |   |

**CETABS = 0**

**SYMBOL, CG = 101**

**CL1NPT = 10**

| LOA | TCA | TDA | COE | TCE | TDE | COR | TCR |
| TOR | TRM | 4 |

**CL2OUTP = 3**

| ALL | SELE | SHUO | S |

**CETABS = 0**

**SYMBOL, CS = 101**

**CL3INPT = 31**

| SW | UP | SAP | 3 | AAP | 3 | UCL | CSK | VI | PA |
| PT | CBP | C | CI | PMW | SK | CK | GAM |   |
| TF | CI | C2 | B | BXP | TI | TDE | SRP | 3 |
| UT | 3 | EST | 3 | MST | 3 | XAP | 3 | UAP | 3 | EAP | 3 | WAP | 3 |
| C1OUTP | 19 |   |   |   |   |   |   |   |   |    |   |

**EF | SCL | SMK | SMB | SFL | FON | FCA | 3 | TCA | 3 |
| FL | 3 | 1 | TI | 3 | 1 | CP | CEX | CV | TLO | PC | R |
| CVH | TSO | FS0 |   |   |   |   |   |   |   |    |   |

**C1TABS = 1**

**ICP = 20**

**SYMBOL, CI = 101**

**DE1INPT = 3**

| S | N | 1 | TAU | N | 1 | 1 |
| DFOUTP | 2 |   |   |   |   |   |
| S | N | 2AD1 | N | S |
| UETABS | 0 |   |   |   |   |   |
| SYMBOL, CE = -99 |

**MODES = BE**

**DF1INPT = 9**

| S | N | 1Z0 | N | 21 | N | 22 | N | PU | N | PI | N | TAU | CPU |
| N | 1 | 1 |
| DFOUTP | 8 |   |   |   |   |   |   |
| S | N | 2AD1 | N | SD2 | N | SAG | N | AL | N | AZ | N | B1 | N | B2 | N |

**UFTABS = 0**

487
SYMBOL, DF = -99
MODES = DF
INIT = 3
S N 1 S N 3 N 1 L
OUT = 1
S N 2
ITABS = 0
SYMBOL, DI = 101
MODES = DI
DRINPT = 6
GAP 3 DBA 3 XAP 3 EAP 3 SRP 3 EST 3
GROUP = 7
FZ 3 1 T2 3 1 FOA 3 1 TDA 3 1 DLL 1 DBF 1 Sw
URTABS = 1
B 20 1
SYMBOL, DR = 101
ENINPT = 6
TCO 3 THR 3 GAX 3 GAZ 3 X0 3 20
ENOUT = 3
TH SF S T 3
ENTABS = 0
SYMBOL, EN = 101
FMINPT = 5
T 3 GA1 3 DMP 3 WN 3 CX 3
FMOUT = 2
A SAD S
FMAB = 0
SYMBOL, FA = 101
FPINPT = 3
A AD CW S
FPDOUT = 2
EA 3 2 W 3
FPITABS = 0
SYMBOL, FP = 101
FOUT = 2
S 1 AN
S 2
FOUTAB = 1
FIA 46 1
SYMBOL, FIA = 101
FIOUT = 4
S 1 S 3 AN 8N
FIVOUT = 1
S 2
FIVTABS = 1
FIA 174 2
SYMBOL, FIA = 101
FIOUT = 6
S 1 S 3 S 4 ANX ANY 3 ANZ
FMOUT = 1
S 2
FMOUTAB = 1
FIA 242 3
SYMBOL, FW = 101
FIOUT = 11
EQI EQI AQI AQI AB RNL RS1 RW
XW XC WO
<table>
<thead>
<tr>
<th>Column</th>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
<th>Column 6</th>
<th>Column 7</th>
<th>Column 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Y0</td>
<td>YP</td>
<td>YR</td>
<td>YDR</td>
<td>YDA</td>
<td>LB</td>
<td>LBD</td>
<td>101</td>
</tr>
<tr>
<td>2</td>
<td>LP</td>
<td>LR</td>
<td>LDK</td>
<td>LDA</td>
<td>MB</td>
<td>NBO</td>
<td>NP</td>
<td>NR</td>
</tr>
<tr>
<td>3</td>
<td>MA</td>
<td>MDA</td>
<td>RUD</td>
<td>AIL</td>
<td>UD</td>
<td>WD</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td>BE</td>
<td>BU</td>
<td>XP</td>
<td>ID</td>
<td>CAL</td>
<td>SAL</td>
<td>UO</td>
<td>WO</td>
</tr>
<tr>
<td>5</td>
<td>LOOUTP</td>
<td>2</td>
<td>GO</td>
<td>2</td>
<td>ITZ</td>
<td>2</td>
<td>ITZ</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>LDA</td>
<td>101</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

490
SYM BoL, L1 = 101
LI NPT = 5
S N 1 TC1 N TC2 N GAI N N 11
LLOUTP = 2
AI N SS N 2
LLTABS = 0
SYMBOL, LL = 101
MODES = LL
LQINPT = 35
AU XA XU ADE ZO ZA ZAD ZQ
ZU ZDE MO MAL MAD MQ MU MDE
MA 1 C XP 1 ID CAL SAL F 3 T 3
FLE AL ALP UO 3 UP WP VT QS
MO 3 UM EU 3
LLOUTP = 7
FX 2 FZ 2 TY 2 UD WD MA 2 XP 2
LDTABS = 0
SYMBOL, LO = 101
LZINPT = 6
AUI AQI RS2 RL RNL XL XC WD
LZOUTP = 5
LEO SEQO SAOS SAQS SRTL
LETABS = 0
SYMBOL, LZ = 101
MAINPT = 4
S N 1 C1 N C2 N N 11
MAOUTP = 1
S N 2
MATABS = 0
SYMBOL, MA = 101
MODES = MA
MLEINPT = 6
S N 1 L S N 3 S N 4 C1 N C2 N C3 N C4 N N 1
MLOUTP = 1
S N 2
MCELABS = 0
SYMBOL, ML = 101
MODES = MC
MDINPT = 5
S N UMP N WN N GAI N N 11
MDOUTP = 3
Q N SQD N SQDD N
MDTABS = 0
SYMBOL, MD = 101
MODES = MD
MEINPT = 3
F 3 UCN N 3 N 11
MEOUTP = 1
S N 2
METABS = 0
SYM BoL, ME = 101
MODES = ME
MFINPT = 5
S N 1 S N 3 S N 4 S N 5 N 11
MFOUTP = 1
S N 2
MFTABS = 0
SYMBOL, MF = 101

491
MUSES = MF
MINT = 2
PLG 4 1M
MOUTH = 1
AMG 4
MTABS = 0
SYMBOL, MG = 101
MAIN = 5
Q N QU N QOD N PCM 3 N N 1 1
MOUTH = 3
X 3 XD 3 XDD 3
MTABS = 0
SYMBOL, MM = 101
MDES = MM
MIN = 32
SW YL 3 EA 3 XR XD ER 3 ED 3 UV 3
CSK VI PA PT CBP C CI PMW
GAM TF CL C2 B BXP TI TDE
SKP 3 UST 3 EST 3 WST 3 XPP 3 UPP 3 EPP 3 WPP 3
MOUTH = 19
BF SEL SMK SWB SFL FL 3 1 TL 3 1 FPP 3
IPP 3 FM EXM VM TLO PC R CVH
ISD FSU TRM 3
MPTABS = 1
MPI = 20 1
SYMBOL, MP = 101
MINT = 4
IC AL GTM TMS
MOUTH = 1
ITR
MTABS = 1
COE 29 1
SYMBOL, MR = 101
MI = 3
S N 1 S N 3 N 1 1
MOUTH = 1
S N 2
MTABS = 0
SYMBOL, MI = 101
MDES = MT
MI = 4
MOUTH = 4
PDV ALP MAC FLG
MTABS = 0
SYMBOL, MA = 101
MI = 17
FLG PDV CX LP NM LP L MQ
M NK MMP KRU KFL RFA MA IXX
IVY
MI = 19
U 3 SEA 3 SW 3 SFX FY F2 TX TY
T2 XD YU ZD X Y 2 RPM
PIT YAW TLL
MTABS = 0
SYMBOL, MZ = 101
MINT = 0
MOUTH = 0
492
OCTAB = 0
SYMBOL, OC = 400
PFLINP = 30
STI  RSC  RFD  RFS  b  CI  CT  3
CH  FD  PMI  PPI  TEM  CSP
CPG  FLA  FLP  FPP  TPP  VAP  UVL  3
VCG  PG  CWT  TPE  TRM  3
PMOUTP = 19
UPP  SXPP  SWPP  SEPP  SUPC  SXPC  SPMA  SW
FLI  FDR  FMA  RM  VOL  TLA  TLS  TOS
DII  TDU  TRP
PLTABS = 0
SYMBOL, PC = 101
PFIINP = 14
ED  EQ  AD  AQ  X1  X2  X3  X4
FR  AB  VB  CMA  G1
POUTP = 9
B1  SB2  SARO  AIO  AT  VPF  PFL  FIN
FG
PFTABS = 0
SYMBOL, PF = 101
PMINP = 10
F  3  MA  LA  LO  ALT  TI  DA  VEL
AZI  GAM
PMTOUTP = 2
K  3  SRD  S  3
PMITABS = 0
SYMBOL, PM = 101
POINP = 5
R  3  RD  A  3  T1  DA
POOUTP = 6
LA  LO  ALT  AZI  GAM  EA  3
POJABS = 0
SYMBOL, PO = 101
RAINP = 0
RADOUTP = 4
NU  NV  NW  NP
KATABS = 0
SYMBOL, KA = 101
RGINP = 4
W  3  1  SL  DMP  WN
RGOUTP = 2
W  3  2SWX  S
KGITABS = 0
SYMBOL, KG = 101
RGINP = 4
FON  XRN  3  YAW  PIT
KKOUTP = 6
PHA  RON  FST  3  TST  3  FR  TIG
KKTABS = 1
TRF  20.  1
SYMBOL, RK = 101
RKLINP = 26
BL1  BL2  3  BL3  BL4  3  BL5  3  BL6  3  UP  RLR
ARR  RLL  XRL  ERL  3  SPR  2  DPG  2  SBF  ZTS
AVS  CPI  3  SRP  UST  3  EST  3  WST  3  XAP  3  UAP  3
LAP  3  WAP  3
KLDOUTP = 12

493
| F2  | 3   | T2  | 3   | FRA | 3   | TRA | 3   | FL   | 3   | FTS  |   | TTS  | OFF |
|-----|-----|-----|-----|-----|-----|-----|-----|------|-----|------|   |      |     |
| USA | 3   | SRA | 3   | DIS | TM  | 3   |      |      |      |      |    |      |     |
| RLtabs| = | 0   |      |      |      |      |      |      |      |      |    |      |     |
| SYMBOL, RL = | 101 |
| RHinpt = | 3   |      |      |      |      |      |      |      |      |      |    |      |     |
| AX   | S1G | MN  |      |      |      |      |      |      |      |      |    |      |     |
| KMOutP = | 1   |      |      |      |      |      |      |      |      |      |    |      |     |
| S    |      |      |      |      |      |      |      |      |      |      |    |      |     |
| KNTABS = | 0   |      |      |      |      |      |      |      |      |      |    |      |     |
| SYMBOL, RN = | 101 |
| RSInpt = | 15  |      |      |      |      |      |      |      |      |      |    |      |     |
| FL   | XYZ | 3   | EA  | 3   | XPB | 3   | UPB | 3   | EPB  | 3   | WPB  | 3   | XAB  |     |
| QAD  | 3   | EAD | 3   | WAD | 3   | AR  | 3   | XD   | 3   | EK   | 3   | EQ   | 3   |
| KSOUTP = | 5   |      |      |      |      |      |      |      |      |      |    |      |     |
| FPB  | 3   | TPB | 3   | FAB | 3   | TAB | 3   | TRM  | 3   |      |    |      |     |
| KStabs = | 0   |      |      |      |      |      |      |      |      |      |    |      |     |
| SYMBOL, KS = | 101 |
| SAINPT = | 6   |      |      |      |      |      |      |      |      |      |    |      |     |
| s    | N   | 1   | C1  | N   | C2  | N   | C3  | N   | C4   | N   | C5   | N   | C6   | N   | 1   |
| SAOUTP = | 1   |      |      |      |      |      |      |      |      |      |    |      |     |
| s    | N   | 2   |     |     |     |     |     |     |     |     |    |      |     |
| SATABS = | 0   |      |      |      |      |      |      |      |      |      |    |      |     |
| SYMBOL, SA = | 101 |
| MUTES = | 5A  |      |      |      |      |      |      |      |      |      |    |      |     |
| SCLNPT = | 17  |      |      |      |      |      |      |      |      |      |    |      |     |
| SW   | UP  | SAP | 3   | AAP | 3   | UCL | 3   | CSK  | 3   | SK   | 3   | CK   |     |
| IDE  | SRP | 3   | UST | 3   | EST | 3   | WST | 3   | XAP  | 3   | UAP  | 3   | EAP  | 3   |
| WAP  | 3   |      |      |      |      |      |      |      |      |      |    |      |     |
| SGOUTP = | 12  |      |      |      |      |      |      |      |      |      |    |      |     |
| FL   | FUN | FCA | 3   | TCA | 3   | FCS | 3   | TCS  | 3   | CF   | 3   | CEX  |     |
| CV   | TCT | TSO | 3   | P50 | 3   |      |      |      |      |    |      |     |
| SCTABS = | 1   |      |      |      |      |      |      |      |      |      |    |      |     |
| TLP  |   | 2G  | 1   |      |      |      |      |      |      |    |      |     |
| SYMBOL, SC = | 101 |
| SCUNPT = | 12  |      |      |      |      |      |      |      |      |      |    |      |     |
| UD   | MU  | TX  | 3   | TY  | 3   | TL  | 3   | IXX  | 3   | IYY | 3   |      |     |
| IZL  | 1ZL | IY  | 3   | 1Y  | 3   |      |      |      |      |    |      |     |
| SUOUTP = | 6   |      |      |      |      |      |      |      |      |      |    |      |     |
| U    | 3   | SW  | 3   | SEA | 3   | SXD | 3   | YD   | 3   | ALR  | 3   | ALT  | 3   | SWD | 3 |
| SUtabs = | 0   |      |      |      |      |      |      |      |      |      |    |      |     |
| SYMBOL, SU = | 101 |
| SCUNPT = | 37  |      |      |      |      |      |      |      |      |      |    |      |     |
| F1   | 3   | F1  | 3   | T1  | 3   | T1  | 3   | T1   | 3   | T1   | 3   | T1   | 3   | 7   | 3 |
| F2   | 3   | F2  | 3   | T2  | 3   | T2  | 3   | T2   | 3   | T2   | 3   | T2   | 3   | 7   | 3 |
| T2   | 3   | T2  | 3   | T2  | 3   | T2  | 3   | T2   | 3   | T2   | 3   | T2   | 3   | 7   | 3 |
| CM   | CML | 3   | CPI | 3   | TM  | 3   |      |      |      |      |    |      |     |
| SCOUTP = | 11  |      |      |      |      |      |      |      |      |      |    |      |     |
| UST  | 3   | SSMP | 3 | SWST | 3   | SUST | 3   | SSID | 3   | SSC  | 3   | SGX  | 3   | GY  | 3 |
| GL  | UK  | 3   |     |     |     |     |     |     |     |     |    |      |     |
| SCLNPT = | 0   |      |      |      |      |      |      |      |      |      |    |      |     |
| SYMBOL, SC = | 101 |
| SCUNPT = | 4   |      |      |      |      |      |      |      |      |      |    |      |     |
| FRI  | FRI | AM1 | AM2 |      | 3   | LUF | AMP | 3   |      |      |    |      |     |
| S    | F   |      |      |      |      |      |      |      |      |      |    |      |     |
| SCtabs |   | 0   |      |      |      |      |      |      |      |      |    |      |     |
| SYMBOL, SG = | 101 |

494
SLINPT = 2
UD = 3
WD = 3
SLOUTP = 4
UAP = 3
SAAP = 3
SWAP = 3
SEAP = 3
SLTAB = 0
SYMOL, SL = 101
SPINPT = 20
SL = YPR
AVW
WM1
SM1
RI1
RIF
XR
= 3
UV = GSA
GSF
SPR
DPG
FMT
TMX
TFN
TOS = TSU
GMA
WST = 3
SPOUTP = 8
MG = SESG
3
SESR
3
SPHA
FL
= 1
3
1
T1
3
1
TIN
ECA
SPITABS = 3
TR1 = <0.1
TMA = <0.1
TST = <0.1
SYMOL, SP = 101
SNTP = 9
FGN
PLG
= 3
EA
= 3
XRN
= 3
YAW
= 3
PIT
= 3
PL
= 3
POD
= 3
PLU
= 15
SR = SPHA
RQN
FL
= 3
1
T1
3
1
X
3
1
BM
3
1
BP
3
FR = PWI
SPI
RHO
VWI
TM1
3
TIG
STTABS = 1
TR = <0.1
SYMOL, SR = 101
SINPT = 1
S = 7
SSOUTP = 7
SL = SGI
SSAV
CAV
GAN
PHS
CPU
STTABS = 0
SYMOL, SI = 101
SUINPT = 4
T = 3
1
T
3
1
F
3
3
T
3
3
SUOUTP = 2
T = 3
2
1
3
2
STTABS = 0
SYMOL, SU = 101
SNTP = 0
F = 3
1
T
3
1
F
3
3
T
3
3
SOUTP = 2
F = 3
< 1
3
2
STTABS = 0
SYMOL, SV = 101
SNTP = 0
N = 1
S
N
3
SM1
TC1
TC2
N
1
1
SUOUTP = 1
N = 0
STTABS = 0
SYMOL, SN = 101
SNTP = 0
STTABS = 0
SUOUTP = 1
N = 0
SYMOL = SN
SNTP = 0

495
S  N  1 >  '4  3  S  N  5  S  N  6  SW1  TC1  TC2  N  1
SXOUTP =  2
S  N  2  S  N  4
SXSTABS =  0
SYMBOL, SX =  101
MODES = SX
SYNPI =  10
S  N  1 >  '4  3  S  N  5  S  N  6  S  N  7  S  N  9  SW1  TC1
TC2  N  1  1
SYOUTP =  3
S  N  2  S  N  4  S  N  8
SYTAB =  0
SYMBOL, SY =  101
MODES = SY
TAINPT =  0
TADOUTP =  4
S  2  S  3  S  4  S  5
STABS =  4
AZT  39  1
BZT  39  1
CZT  39  1
DZT  39  1
SYMBOL, TA =  101
TBINPT =  0
TBOUTP =  2
S  2  S  3
STABS =  4
AZT  39  1
BZT  39  1
CZT  39  1
DZT  39  1
SYMBOL, TB =  101
TDBINPT =  4
T =  3  1XX  IYY  LZZ
TDOOUTP =  3
W  3  STA  3  SWU  3
TDSTABS =  0
SYMBOL, TU =  101
TFINPT =  6
S  2  N  1  21  N  PU  N  P1  N  N  1  1
TFOUTP =  2
AL  N  5  S  N  2  S
TFSTABS =  0
SYMBOL, TF =  101
MODES = TF
TGINPT =  5
TH  3G  3  X  3
THOUTP =  2
T =  3  1  3
TSTABS =  0
SYMBOL, TG =  101
TSINPT =  1
AL  ALU  GTA  CM  CN  CHP  CMH  CMF
WH  WMH  WMF
TROUTP =  1
VO
TVSTABS =  0
SYMBOL, TK =  101
TSINPT =  2
T  3  1  T  3  3
TSOUTP = 1
ITABS = 0
SYMBOL, TS = 101
TINPT = 3
I 3 L 3 3 3 4
TIOUTP = 1
IT = 2
ITABS = 0
SYMBOL, IT = 101
USINPT = 10
LMP M M MS1 M M WKK M STF M M THR 3 1 LMN 3 1 GNF M 1 DLM 2
N 1 1 M 1 1
USUOUTP = 10
UWW 3 SPWR 3 SSD1 2 SSD2 2 SOLED 2 SFXD N SSL1 2 SSL2 2
UXT 2 SFLX N S
USTABS = 0
SYMBOL, US = 101
MODES = US
UTINPT = 40
PWR 3 MS1 MS2 LS1 LS2 SPI 3 SP2 3 ME
LT EP 3 MSS IXX IYY IZZ IXY IXZ
LYE YEE ILE IYH IZH MM N PS1 2 N PS2 2
PL 2 N PEP 2 N WP1 WT1 WP2 WT2 WFX N WEP
X1 ZS1 ZS2 ZFX N ZEP ZET N 1 1 M 1
LUOUTP = 5
LMP M M MS1 M M STF M M MAS M M LQW M
UOUTABS = 0
SYMBOL, UT = 101
MODES = UT
VAINPT = 9
1 3 IN 3 3 LA 1 LU 1 TI 1 DA 1 ROL PIT
YAW
VAOUTP = 8
A 3 SW 4 SA 3 3 EA 3 LA 2 LU 2 TI 2 DA
VOUTABS = 0
SYMBOL, VA = 101
VoINPT = 15
VPP VL ED EW VRE G1 G2 K
T1 T2 T3 T4 CEX ED G3
VOUTP = 7
L2 SE4 SB5 SVO SEL E1 E3
VOUTABS = 0
SYMBOL, VO = 101
W1INPT = 27
Ab SW SX 3 SM 3 SP 3 W 1 X 3 1 BM 3
Op 3 1 W 2 X 3 2 BM 3 2 BP 3 2 W 3 X 3 3 BM 3
Op 3 3
WOUTP = 4
CL CCII 3 CM1 3 CPI 3
WOUTABS = 0
SYMBOL, WC = 101
W1INPT = 10
NU NV NW NP SLH SLV VS 1 SIM
SLV 5
W1OUTP = 11
CW SYW SVX SWW SWX SPW SXA SJW
KA SRW VS 2
WMATA$ = 0
SYMODL, WM = 101
XPINPT = 2
W 3 1 TRN 3 3
XPOUTP = 1
W 3 2
XPIATS = 0
SYMODL, XP = 101
XTINPT = 2
T 3 1 TRN 3 3
XTOUTP = 1
T 3 2
XTIATS = 0
SYMODL, XT = 101
EASIEST F-4E MANEUVERING COEFFICIENTS

This appendix contains a listing of the EASIEST F-4E airplane maneuvering coefficients formatted for the EASIEST airplane modeled by component AE.
<table>
<thead>
<tr>
<th>LOCATION OF INDEPENDENT VARIABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>C20(A,M)</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>0.22</td>
</tr>
<tr>
<td>0.55</td>
</tr>
<tr>
<td>0.975</td>
</tr>
<tr>
<td>CL M=0.2</td>
</tr>
<tr>
<td>CL M=0.6</td>
</tr>
<tr>
<td>CL M=0.7</td>
</tr>
<tr>
<td>CL M=0.8</td>
</tr>
<tr>
<td>CL M=0.9</td>
</tr>
<tr>
<td>CL M=1.0</td>
</tr>
<tr>
<td>CL M=1.1</td>
</tr>
<tr>
<td>CL M=2.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCATION OF INDEPENDENT VARIABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CZAD(M)</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>2.25</td>
</tr>
<tr>
<td>2.35</td>
</tr>
<tr>
<td>-1.25</td>
</tr>
<tr>
<td>CLAD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCATION OF INDEPENDENT VARIABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>C20(M1)</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>3.9</td>
</tr>
<tr>
<td>3.7</td>
</tr>
<tr>
<td>4.1</td>
</tr>
<tr>
<td>5.32</td>
</tr>
<tr>
<td>CLQ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCATION OF INDEPENDENT VARIABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CZJE(M,A)</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>0.0113</td>
</tr>
<tr>
<td>0.0090</td>
</tr>
<tr>
<td>0.0078</td>
</tr>
<tr>
<td>CLDS A=</td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>0.0109</td>
</tr>
<tr>
<td>0.0102</td>
</tr>
<tr>
<td>0.0005</td>
</tr>
<tr>
<td>0.0002</td>
</tr>
</tbody>
</table>

### Location of Independent Variables

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0</td>
<td>Location of independent variables (10.0)</td>
</tr>
<tr>
<td>5.2</td>
<td>Location of independent variables (10.0)</td>
</tr>
<tr>
<td>6.0</td>
<td>Location of independent variables (10.0)</td>
</tr>
<tr>
<td>6.5</td>
<td>Location of independent variables (10.0)</td>
</tr>
</tbody>
</table>

### Variation of C2g with Aileron Position

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0005</td>
<td>Variation of C2g with aileron position (10.0)</td>
</tr>
<tr>
<td>0.0002</td>
<td>Variation of C2g with aileron position (10.0)</td>
</tr>
<tr>
<td>6.0</td>
<td>Variation of C2g with aileron position (10.0)</td>
</tr>
<tr>
<td>6.5</td>
<td>Variation of C2g with aileron position (10.0)</td>
</tr>
</tbody>
</table>

### X Axis Bias Coefficient for Trim

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0315</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.0485</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.077</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.13</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.18</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.26</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.315</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.0485</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.077</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.13</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.16</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.26</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.315</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.0485</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.077</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.13</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.16</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
<tr>
<td>0.26</td>
<td>X axis bias coefficient for trim (10.0)</td>
</tr>
</tbody>
</table>

501
<table>
<thead>
<tr>
<th>Z</th>
<th>CM0(M)</th>
<th>CMQ(M)</th>
<th>CMU(M)</th>
<th>CMU(M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>28.</td>
<td>-0.012</td>
<td>-0.027</td>
<td>-0.042</td>
<td>-0.055</td>
</tr>
<tr>
<td>30.</td>
<td>-0.017</td>
<td>-0.0149</td>
<td>-0.0149</td>
<td>-0.0121</td>
</tr>
<tr>
<td>28.</td>
<td>-0.016</td>
<td>-0.0156</td>
<td>-0.0156</td>
<td>-0.0127</td>
</tr>
<tr>
<td>28.</td>
<td>-0.0145</td>
<td>-0.0145</td>
<td>-0.0145</td>
<td>-0.0115</td>
</tr>
<tr>
<td>28.</td>
<td>-0.0064</td>
<td>-0.0055</td>
<td>CM M=2</td>
<td></td>
</tr>
</tbody>
</table>

LOCATION OF INDEPENDENT VARIABLES

CM M=2

502
<table>
<thead>
<tr>
<th>Location</th>
<th>CMDA (M1)</th>
<th>Variation of CM0 with Aileron Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 1</td>
<td>12 31</td>
<td>CMDA (M1)</td>
</tr>
<tr>
<td>-0.115</td>
<td>-0.117</td>
<td>-0.118</td>
</tr>
<tr>
<td>-0.044</td>
<td>-0.004</td>
<td>0.013</td>
</tr>
<tr>
<td>0.106</td>
<td>0.106</td>
<td>-0.108</td>
</tr>
<tr>
<td>0.012</td>
<td>-0.098</td>
<td>-0.095</td>
</tr>
<tr>
<td>14 6</td>
<td>14 2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>CYB (M1)</th>
<th>Variation of CY with Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 2</td>
<td>14 40</td>
<td>CYB (M1)</td>
</tr>
<tr>
<td>-0.06</td>
<td>-0.065</td>
<td>-0.065</td>
</tr>
<tr>
<td>-0.33</td>
<td>-0.03</td>
<td>0.23</td>
</tr>
<tr>
<td>0.235</td>
<td>0.24</td>
<td>0.24</td>
</tr>
<tr>
<td>-0.32</td>
<td>-0.07</td>
<td>0.07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>CYP (M1)</th>
<th>Variation of CY with Roll Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 2</td>
<td>15 4</td>
<td>CYP (M1)</td>
</tr>
<tr>
<td>-0.06</td>
<td>-0.065</td>
<td>-0.065</td>
</tr>
<tr>
<td>-0.33</td>
<td>-0.03</td>
<td>0.23</td>
</tr>
<tr>
<td>0.245</td>
<td>0.24</td>
<td>0.24</td>
</tr>
<tr>
<td>-0.32</td>
<td>-0.07</td>
<td>0.07</td>
</tr>
</tbody>
</table>

503
### Variation of CY with Yaw Rate

<table>
<thead>
<tr>
<th>CYR</th>
<th>A=0</th>
<th>A=240</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.735</td>
<td>0.78</td>
<td>0.80</td>
</tr>
<tr>
<td>0.74</td>
<td>0.78</td>
<td>0.80</td>
</tr>
<tr>
<td>0.74</td>
<td>0.78</td>
<td>0.80</td>
</tr>
<tr>
<td>0.74</td>
<td>0.78</td>
<td>0.80</td>
</tr>
</tbody>
</table>

### Variation of CY with RUDDER Position

<table>
<thead>
<tr>
<th>CYDR</th>
<th>A=0</th>
<th>A=240</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00219</td>
<td>0.00219</td>
<td>0.00219</td>
</tr>
<tr>
<td>0.00126</td>
<td>0.00126</td>
<td>0.00126</td>
</tr>
<tr>
<td>0.00126</td>
<td>0.00126</td>
<td>0.00126</td>
</tr>
<tr>
<td>0.00126</td>
<td>0.00126</td>
<td>0.00126</td>
</tr>
<tr>
<td>0.00126</td>
<td>0.00126</td>
<td>0.00126</td>
</tr>
</tbody>
</table>

### Variation of CY with AILERON Position

<table>
<thead>
<tr>
<th>CYDA</th>
<th>A=0</th>
<th>A=240</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000145</td>
<td>0.000145</td>
<td>0.000145</td>
</tr>
<tr>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
</tr>
<tr>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0002</td>
</tr>
</tbody>
</table>

### Variation of CL with Beta

<table>
<thead>
<tr>
<th>CLB</th>
<th>A=0</th>
<th>A=240</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0045</td>
<td>-0.0045</td>
<td>-0.0045</td>
</tr>
<tr>
<td>-0.0019</td>
<td>-0.0019</td>
<td>-0.0019</td>
</tr>
<tr>
<td>-0.0019</td>
<td>-0.0019</td>
<td>-0.0019</td>
</tr>
<tr>
<td>-0.0019</td>
<td>-0.0019</td>
<td>-0.0019</td>
</tr>
</tbody>
</table>

504
<table>
<thead>
<tr>
<th>Location</th>
<th>CLP (m, A)</th>
<th>Location</th>
<th>CLP (m, A)</th>
<th>Location</th>
<th>CLP (m, A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2.</td>
<td>3.</td>
<td>4.</td>
<td>5.</td>
<td>6.</td>
</tr>
<tr>
<td>19</td>
<td>19.2</td>
<td>19.66</td>
<td>19.02</td>
<td>1.022</td>
<td>1.027</td>
</tr>
<tr>
<td>19.2</td>
<td>19.2</td>
<td>19.2</td>
<td>19.2</td>
<td>19.2</td>
<td>19.2</td>
</tr>
<tr>
<td>19.02</td>
<td>19.02</td>
<td>19.02</td>
<td>19.02</td>
<td>19.02</td>
<td>19.02</td>
</tr>
<tr>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
</tr>
<tr>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
</tr>
<tr>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
</tr>
<tr>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
</tr>
<tr>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
</tr>
<tr>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
</tr>
<tr>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
</tr>
<tr>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
</tr>
<tr>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
</tr>
</tbody>
</table>

**CLP (m, A) VARIATION OF CL WITH ROLL RATE**

<table>
<thead>
<tr>
<th>Location</th>
<th>CLP (m, A)</th>
<th>Location</th>
<th>CLP (m, A)</th>
<th>Location</th>
<th>CLP (m, A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2.</td>
<td>3.</td>
<td>4.</td>
<td>5.</td>
<td>6.</td>
</tr>
<tr>
<td>19</td>
<td>19.2</td>
<td>19.02</td>
<td>1.022</td>
<td>1.027</td>
<td>1.028</td>
</tr>
<tr>
<td>19.2</td>
<td>19.2</td>
<td>19.2</td>
<td>1.022</td>
<td>1.027</td>
<td>1.028</td>
</tr>
<tr>
<td>19.02</td>
<td>19.02</td>
<td>19.02</td>
<td>1.022</td>
<td>1.027</td>
<td>1.028</td>
</tr>
<tr>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
</tr>
<tr>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
</tr>
<tr>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
</tr>
<tr>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
</tr>
<tr>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
</tr>
<tr>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
</tr>
<tr>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
</tr>
<tr>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
</tr>
<tr>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
</tr>
</tbody>
</table>

**CLP (m, A) VARIATION OF CL WITH YAW RATE**

<table>
<thead>
<tr>
<th>Location</th>
<th>CLP (m, A)</th>
<th>Location</th>
<th>CLP (m, A)</th>
<th>Location</th>
<th>CLP (m, A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2.</td>
<td>3.</td>
<td>4.</td>
<td>5.</td>
<td>6.</td>
</tr>
<tr>
<td>19</td>
<td>19.2</td>
<td>19.02</td>
<td>1.022</td>
<td>1.027</td>
<td>1.028</td>
</tr>
<tr>
<td>19.2</td>
<td>19.2</td>
<td>19.2</td>
<td>1.022</td>
<td>1.027</td>
<td>1.028</td>
</tr>
<tr>
<td>19.02</td>
<td>19.02</td>
<td>19.02</td>
<td>1.022</td>
<td>1.027</td>
<td>1.028</td>
</tr>
<tr>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
</tr>
<tr>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
</tr>
<tr>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
</tr>
<tr>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
</tr>
<tr>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
</tr>
<tr>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
</tr>
<tr>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
</tr>
<tr>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
</tr>
<tr>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
</tr>
</tbody>
</table>

**CLP (m, A) VARIATION OF CL WITH RUDDER POSITION**

<table>
<thead>
<tr>
<th>Location</th>
<th>CLP (m, A)</th>
<th>Location</th>
<th>CLP (m, A)</th>
<th>Location</th>
<th>CLP (m, A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2.</td>
<td>3.</td>
<td>4.</td>
<td>5.</td>
<td>6.</td>
</tr>
<tr>
<td>19</td>
<td>19.2</td>
<td>19.02</td>
<td>1.022</td>
<td>1.027</td>
<td>1.028</td>
</tr>
<tr>
<td>19.2</td>
<td>19.2</td>
<td>19.2</td>
<td>1.022</td>
<td>1.027</td>
<td>1.028</td>
</tr>
<tr>
<td>19.02</td>
<td>19.02</td>
<td>19.02</td>
<td>1.022</td>
<td>1.027</td>
<td>1.028</td>
</tr>
<tr>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
<td>1.022</td>
</tr>
<tr>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
<td>1.027</td>
</tr>
<tr>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
<td>1.028</td>
</tr>
<tr>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
<td>1.029</td>
</tr>
<tr>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
<td>1.03</td>
</tr>
<tr>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
<td>1.031</td>
</tr>
<tr>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
<td>1.032</td>
</tr>
<tr>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
<td>1.033</td>
</tr>
<tr>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
<td>1.034</td>
</tr>
</tbody>
</table>
LOCATION OF INDEPENDENT VARIABLES

VARIATION OF CL WITH AILERON POSITION

VARIATION OF CN WITH BETA

VARIATION OF CN WITH ROLL RATE
<table>
<thead>
<tr>
<th>0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>CNP A=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.035</td>
<td>-0.045</td>
<td>-0.040</td>
<td>-0.032</td>
<td>-0.032</td>
<td>-0.040</td>
<td>-0.055</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.017</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.0023</td>
<td>-0.0056</td>
<td>-0.005</td>
<td>-0.042</td>
<td>-0.042</td>
<td>-0.042</td>
<td>-0.10</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.04</td>
<td>-0.003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.20</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.068</td>
<td>-0.008</td>
<td>-0.064</td>
<td>-0.055</td>
<td>-0.055</td>
<td>-0.075</td>
<td>-0.125</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.07</td>
<td>-0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.20</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.056</td>
<td>-0.0068</td>
<td>-0.064</td>
<td>-0.055</td>
<td>-0.055</td>
<td>-0.075</td>
<td>-0.125</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.005</td>
<td>-0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.20</td>
<td>CNP A=0</td>
</tr>
</tbody>
</table>

### Location of Independent Variables

#### Variation of CN with Yaw Rate

<table>
<thead>
<tr>
<th>0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>+0.005</th>
<th>CNP A=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.301</td>
<td>-0.32</td>
<td>-0.325</td>
<td>-0.34</td>
<td>-0.357</td>
<td>-0.374</td>
<td>-0.38</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.392</td>
<td>-0.365</td>
<td>-0.365</td>
<td>-0.375</td>
<td>-0.375</td>
<td>-0.380</td>
<td>-0.380</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.340</td>
<td>-0.370</td>
<td>-0.382</td>
<td>-0.395</td>
<td>-0.403</td>
<td>-0.403</td>
<td>-0.403</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.31</td>
<td>-0.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.20</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.375</td>
<td>-0.40</td>
<td>-0.42</td>
<td>-0.425</td>
<td>-0.425</td>
<td>-0.425</td>
<td>-0.425</td>
<td>CNP A=0</td>
</tr>
<tr>
<td>-0.31</td>
<td>-0.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.20</td>
<td>CNP A=0</td>
</tr>
</tbody>
</table>

#### Variation of CN with Rudder Position

<table>
<thead>
<tr>
<th>0.0013</th>
<th>-0.014</th>
<th>-0.0126</th>
<th>-0.0124</th>
<th>-0.0116</th>
<th>-0.0111</th>
<th>-0.0111</th>
<th>CNR A=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0002</td>
<td>-0.0031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CNR A=0</td>
</tr>
<tr>
<td>-0.0013</td>
<td>-0.013</td>
<td>-0.0126</td>
<td>-0.0124</td>
<td>-0.0118</td>
<td>-0.0111</td>
<td>-0.0111</td>
<td>CNR A=0</td>
</tr>
<tr>
<td>-0.00052</td>
<td>-0.0031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CNR A=0</td>
</tr>
<tr>
<td>-0.0012</td>
<td>-0.0121</td>
<td>-0.0118</td>
<td>-0.0115</td>
<td>-0.0111</td>
<td>-0.0103</td>
<td>-0.0103</td>
<td>CNR A=0</td>
</tr>
<tr>
<td>-0.00045</td>
<td>-0.00645</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CNR A=0</td>
</tr>
<tr>
<td>-0.00111</td>
<td>-0.00111</td>
<td>-0.00108</td>
<td>-0.00106</td>
<td>-0.00101</td>
<td>-0.00095</td>
<td>-0.00095</td>
<td>CNR A=0</td>
</tr>
<tr>
<td>-0.00036</td>
<td>-0.00036</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CNR A=0</td>
</tr>
</tbody>
</table>

#### Variation of CN with Aileron Position

<table>
<thead>
<tr>
<th>0.000126</th>
<th>-0.00005</th>
<th>-0.000073</th>
<th>-0.000089</th>
<th>-0.000108</th>
<th>-0.000123</th>
<th>CNDA A=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.000049</td>
<td>-0.000049</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000079</td>
<td>0.000079</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0002</td>
<td>0.0002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000212</td>
<td>0.000212</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MACH NUMBER</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
<td>1.0</td>
</tr>
<tr>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>MACH 1</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>MACH 2</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>MACH 3</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>MACH 4</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
<tr>
<td>MACH 5</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
</tr>
<tr>
<td>MACH 6</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALPHA TABLE</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>16</th>
<th>20</th>
<th>24</th>
<th>28</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA 1</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
</tr>
<tr>
<td>ALPHA 2</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
<tr>
<td>ALPHA 3</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
</tr>
<tr>
<td>ALPHA 4</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
</tr>
<tr>
<td>ALPHA 5</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
</tr>
<tr>
<td>ALPHA 6</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
</tr>
<tr>
<td>ALPHA 7</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
</tr>
<tr>
<td>M</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. 8. 12. 16. 24.</td>
<td>ALPHA 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37 0 37 0 NAB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. 12. 16. 20. 24.</td>
<td>ALPHA 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37 1 37 5 Alpha Table 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. 38 0 38 0 NAB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36 1 36 4 Alpha Table 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. 39 0 39 0 NAB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39 1 39 5 Alpha Table 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. 40 0 40 0 NAB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 1 40 4 Alpha Table 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. 41 0 41 0 NAB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41 1 41 4 Alpha Table 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. 42 0 42 0 NCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42 1 42 7 CL Table (Independent)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10^{10.0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. 0.4 0.6 0.8 0.9 1.0 CL Table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

509
This appendix contains input and output tables for the EASY5, (not EASIEST), standard components. Descriptive figures are also presented for the more complex components.
### ANALYTIC FUNCTION GENERATOR

**AF**

![AF Diagram]

### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COD</td>
<td></td>
<td>Specifies which analytic function is calculated. (See equations below for use of these inputs)</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Equations:**

1. \( S_2 = C_1 + C_2 \sin(C_3t + C_4) \)
2. \( S_2 = C_1 + C_2 \cos(C_3t + C_4) \)
3. \( S_2 = C_1 + e^{-C_5t} \sin(C_3t + C_4) \)
4. \( S_2 = C_1 + e^{-C_5t} \cos(C_3t + C_4) \)
5. \( S_2 = C_1 + C_2t \)
6. \( S_2 = C_1 + C_2e \)

**Note:** \( t \) = TIME

### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>2</td>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>PHYSICAL QUANTITY NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>U(3)</td>
<td></td>
<td>X, Y, Z BODY AXIS LINEAR VELOCITIES</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>W(3)</td>
<td></td>
<td>X, Y, Z BODY AXIS ANGULAR RATES</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>ALT</td>
<td></td>
<td>ALTITUDE ABOVE SEA LEVEL</td>
<td>FT</td>
</tr>
<tr>
<td>EA(3)</td>
<td></td>
<td>PITCH, ROLL, YAW EARTH TO BODY AXIS ANGLES</td>
<td>DEG</td>
</tr>
<tr>
<td>ID</td>
<td>1</td>
<td>INDICATOR FUNCTION FOR AERO COMPONENTS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = BODY AXIS, DIMENSIONAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = BODY AXIS, NON-DIMENSIONAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = STABILITY AXIS, DIMENSIONAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = STABILITY AXIS, NON-DIMENSIONAL</td>
<td></td>
</tr>
<tr>
<td>VS</td>
<td></td>
<td>STEADY STATE (TRIM) AIRSPEED</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>ALS*</td>
<td></td>
<td>STEADY STATE (TRIM) ANGLE OF ATTACK</td>
<td>DEG</td>
</tr>
<tr>
<td>S</td>
<td></td>
<td>REFERENCE AREA</td>
<td>FT$^2$</td>
</tr>
<tr>
<td>UW, VW, WM*, PW*</td>
<td></td>
<td>X, Y, Z BODY AXIS WIND VELOCITIES</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>QW, RW*</td>
<td>1</td>
<td>X, Y, Z BODY AXIS WIND ANGULAR RATES</td>
<td>DEG/SEC</td>
</tr>
</tbody>
</table>

*DEFAULT VALUES = 0
<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UO(3)</td>
<td></td>
<td>X, Y, Z BODY AXIS VELOCITIES INCLUDING WIND</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>WO(3)</td>
<td></td>
<td>X, Y, Z BODY AXIS ANGULAR RATES WITH WIND</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>ID</td>
<td>2</td>
<td>INDICATOR FUNCTION = ID1</td>
<td></td>
</tr>
<tr>
<td>WW(3)</td>
<td>2</td>
<td>ANGULAR RATE DUE TO GUSTS</td>
<td>DEG/SEC</td>
</tr>
<tr>
<td>CAL, SAL</td>
<td></td>
<td>DIRECTION COSINES FOR STABILITY AND BODY AXES</td>
<td></td>
</tr>
<tr>
<td>AL, ALP</td>
<td></td>
<td>ANGLE OF ATTACK IN BODY AND STABILITY AXES</td>
<td>DEG</td>
</tr>
<tr>
<td>VT</td>
<td></td>
<td>TRUE AIRSPEED</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>BE</td>
<td></td>
<td>SIDESLIP ANGLE</td>
<td>DEG</td>
</tr>
<tr>
<td>WP, UP</td>
<td></td>
<td>Z &amp; X STABILITY AXIS VELOCITIES (DIMENSIONAL)</td>
<td>FT/SEC</td>
</tr>
<tr>
<td>EU(3)</td>
<td></td>
<td>X, Y, Z BODY AXIS ACCEL. TERMS FOR U, V, W SOLUTIONS</td>
<td>FT/SEC^2</td>
</tr>
<tr>
<td>SIG</td>
<td></td>
<td>STANDARD ATMOSPHERE AIR DENSITY RATIO</td>
<td></td>
</tr>
<tr>
<td>QC</td>
<td></td>
<td>COMRESSIBLE DYNAMIC PRESSURE</td>
<td>LBS/FT^2</td>
</tr>
<tr>
<td>QS</td>
<td></td>
<td>DYNAMIC PRESSURE TIMES REFERENCE AREA</td>
<td>LBS</td>
</tr>
<tr>
<td>MAC</td>
<td></td>
<td>MACH NUMBER</td>
<td></td>
</tr>
</tbody>
</table>
VECTOR DEFINITIONS

\[
\begin{align*}
U(3) &= \begin{pmatrix} U \\ V \\ W \end{pmatrix}, \\
W(3) &= \begin{pmatrix} P \\ Q \\ R \end{pmatrix}, \\
E(3) &= \begin{pmatrix} PIT \\ ROL \\ YAW \end{pmatrix}, \\
UO(3) &= \begin{pmatrix} UO \\ VO \\ WO \end{pmatrix}, \\
WO(3) &= \begin{pmatrix} PO \\ QO \\ RO \end{pmatrix}, \\
WW(3) &= \begin{pmatrix} QW \\ RW \end{pmatrix}, \\
E(3) &= \begin{pmatrix} EU \\ EV \\ EW \end{pmatrix}.
\end{align*}
\]

AERODYNAMIC VARIABLE EQUATIONS

\[
\begin{align*}
CAL &= \begin{cases} \cos(\text{ALS}) & \text{id} = 2,3 \\ 1 & \text{id} = 0,1 \end{cases} \\
SAL &= \begin{cases} \sin(\text{ALS}) & \text{id} = 2,3 \\ 0 & \text{id} = 0,1 \end{cases} \\
UO &= U - U \cdot \hat{w} \\
VO &= V - V \cdot \hat{w} \\
WO &= W - W \cdot \hat{w} \\
PO &= (P + PW) \cdot CAL + (R + RW) \cdot SAL \\
QP &= Q + QW \\
RO &= (R + RW) \cdot CAL - (P + PW) \cdot SAL \\
AL &= \tan^{-1}(WO / UO) \\
ALP &= AL - ALS \\
VT &= (UO^2 + VO^2 + WO^2)^{1/2} \\
BE &= \sin^{-1}(VO / VT) \\
WP &= WO \cdot CAL - UO \cdot SAL \\
UP &= \begin{cases} UO \cdot CAL + WO \cdot SAL & \text{id} = 0,2 \\ UO \cdot VS \cdot \cos(\text{ALS}) / VS & \text{id} = 1 \\ UO \cdot CAL \cdot \sin(\text{ALS}) / VS & \text{id} = 3 \end{cases} \\
EU &= -Q \cdot W + R + V - G \cdot \sin(PIT) \\
EV &= -R \cdot U + P + W + G \cdot \cos(PIT) \cdot \sin(ROL) \\
EW &= -P + V + Q + U + G \cdot \cos(PIT) \cdot \cos(ROL) \\
\end{align*}
\]

where \( P = P \cdot \pi / 180 \), \( Q = Q \cdot \pi / 180 \), \( R = R \cdot \pi / 180 \)

\[
\begin{align*}
\text{SIG} &= \text{SIG}(\text{ALT}) \text{ and } A = A(\text{ALT}) \text{ obtained by table lookup} \\
DPS &= \frac{1}{2} PO \cdot \text{SIG} \cdot (VT)^2 \\
QS &= DPS \cdot S \\
\text{MAC} &= VT / A \\
\end{align*}
\]

\[
\begin{align*}
\text{QC} &= \begin{cases} DPS \cdot (1 + (1 + \text{MAC}^2 / 40) \cdot \text{MAC}^2 / 10) \cdot \text{MAC}^2 / 4) & \text{MAC} \leq 1 \\ DPS \cdot (1.839 - .772 / \text{MAC}^2 + .164 / \text{MAC}^4 + .035 / \text{MAC}^6) & \text{MAC} > 1 \end{cases}
\end{align*}
\]
### CONTROL MOMENT GYRO

**CMG ANGULAR RATES**

- Motor Torque
- **CG**

**CMG TORQUES**

- **CG**
- Motor & Gimbal Angles and Rates

---

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>W(3)</td>
<td></td>
<td>CMG Angular Rates; P, Q, R</td>
<td>rad/sec</td>
</tr>
<tr>
<td>TR</td>
<td></td>
<td>Motor Torque</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>TMI</td>
<td></td>
<td>Torque Motor Inertia</td>
<td>slug-ft²</td>
</tr>
<tr>
<td>AD</td>
<td></td>
<td>Torque Motor Rate Damping Limit</td>
<td>rad/sec</td>
</tr>
<tr>
<td>DA</td>
<td></td>
<td>Torque Motor Damping</td>
<td>ft-lb/rad/sec</td>
</tr>
<tr>
<td>GR</td>
<td></td>
<td>Gear Ratio</td>
<td>-</td>
</tr>
<tr>
<td>BDZ</td>
<td></td>
<td>Gear Backlash Deadzone</td>
<td>rad</td>
</tr>
<tr>
<td>CS</td>
<td></td>
<td>Gear Train Compliance</td>
<td>ft-lb/rad/sec</td>
</tr>
<tr>
<td>CA</td>
<td></td>
<td>Preload Spring Compliance</td>
<td>ft-lb/rad/sec</td>
</tr>
<tr>
<td>PLD</td>
<td></td>
<td>Preload Deadzone</td>
<td>rad</td>
</tr>
<tr>
<td>DG</td>
<td></td>
<td>Damping</td>
<td>ft-lb/rad</td>
</tr>
<tr>
<td>GFI</td>
<td></td>
<td>Gimbal Inertia</td>
<td>slug-ft²</td>
</tr>
<tr>
<td>DR</td>
<td></td>
<td>Gimbal Damping Coefficient</td>
<td>ft-lb/rad/sec</td>
</tr>
<tr>
<td>DRS</td>
<td></td>
<td>Gimbal Damping Saturation Limit</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>CB</td>
<td></td>
<td>Gimbal Friction Spring Term</td>
<td>ft-lb/rad/sec</td>
</tr>
<tr>
<td>CBS</td>
<td></td>
<td>Gimbal Friction Compliance Limit</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>DDZ</td>
<td></td>
<td>Gimbal Damping Deadzone</td>
<td>rad</td>
</tr>
<tr>
<td>DF</td>
<td></td>
<td>Gimbal Damping Equivalent Spring</td>
<td>-</td>
</tr>
<tr>
<td>DS</td>
<td></td>
<td>Gimbal Viscous Friction</td>
<td>ft-lb/rad/sec</td>
</tr>
<tr>
<td>H</td>
<td></td>
<td>Angular Momentum</td>
<td>ft-lb-sec</td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>*AL</td>
<td></td>
<td>Torque Motor Angle</td>
<td>rad</td>
</tr>
<tr>
<td>ALD</td>
<td></td>
<td>Torque Motor Rate</td>
<td>rad/sec</td>
</tr>
<tr>
<td>*AX</td>
<td></td>
<td>Torque Motor Intermediate State</td>
<td>rad/sec</td>
</tr>
<tr>
<td>*SG</td>
<td></td>
<td>Relative Gimbal Angle</td>
<td>rad</td>
</tr>
<tr>
<td>SGD</td>
<td></td>
<td>Relative Gimbal Angle Rate</td>
<td>rad/sec</td>
</tr>
<tr>
<td>*SGI</td>
<td></td>
<td>Inertial Gimbal Angle</td>
<td>rad</td>
</tr>
<tr>
<td>*SF</td>
<td></td>
<td>Gimbal Friction Spring Term</td>
<td>-</td>
</tr>
<tr>
<td>T(3)</td>
<td></td>
<td>CMG X, Y, Z Axis Torques</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>

*These outputs are states*
DISCRETE DELAY

\[ S \rightarrow 1 \rightarrow \frac{1}{Z} \rightarrow 2 \rightarrow S \]

\((N \times 1)\) \hspace{1cm} \((N \times 1)\)

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>TAU</td>
<td></td>
<td>Sample period</td>
<td>seconds</td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Delay output (Delay state)</td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS:

\[ S_2(N) = Z^{-1} [S_1(N)] \]

\[ Z^{-1} [ ] = \text{Discrete delay operator of TAU seconds} \]

**NOTE:** N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0.
DIGITAL FILTER

\[ \frac{Z_2 S^2 + Z_1 S + Z_0}{S^2 + P_1 S + P_0} \]

INPUTS

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input Quantity</td>
<td></td>
</tr>
<tr>
<td>Z0(N)</td>
<td></td>
<td>Numerator coefficient (S-plane)</td>
<td></td>
</tr>
<tr>
<td>Z1(N)</td>
<td></td>
<td>Numerator coefficient (S-plane)</td>
<td></td>
</tr>
<tr>
<td>Z2(N)</td>
<td></td>
<td>Numerator coefficient (S-plane)</td>
<td></td>
</tr>
<tr>
<td>P0(N)</td>
<td></td>
<td>Denominator coefficient (S-plane)</td>
<td></td>
</tr>
<tr>
<td>P1(N)</td>
<td></td>
<td>Denominator coefficient (S-plane)</td>
<td></td>
</tr>
<tr>
<td>TAU</td>
<td></td>
<td>Sample period</td>
<td>sec</td>
</tr>
</tbody>
</table>

OUTPUTS

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output quantity (Sample)</td>
<td></td>
</tr>
<tr>
<td>D1(N)</td>
<td></td>
<td>Intermediate output (Delay)</td>
<td></td>
</tr>
<tr>
<td>D2(N)</td>
<td></td>
<td>Intermediate output (Delay)</td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS:

\[
D_2 = Z^2 A_2 S + D_2 - B_2 S
\]
\[
D_1 = Z^1 D_2 + A_1 S + B_1 S
\]
\[
S_2 = A_0 S + D_1
\]
\[
Z^{-1} = \text{discrete delay operator}
\]

\[
A_0 \sim A_2 \text{ and } B_0 \sim B_1 \text{ are related to S-plane coefficients by applying prewarping and bilinear transformation;}
\]
\[
\frac{W_i}{\tau} \left( \frac{1 - \Delta}{1 + \Delta} \right)
\]

to each of the singularities, \( W_i \),

of the numerator and denominator.

Note: \( N \) may be specified at Model Generation time to allow inputs and outputs to be \( N \) dimensional vectors. Default value of \( N \) is 1.0
DIVIDER

\[
\begin{align*}
S2 &= \frac{S1}{S3} \\
S &= \text{INPUT} \\
S &= \text{OUTPUT}
\end{align*}
\]

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Numerator</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>3</td>
<td>Denominator</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Quotient</td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS:

\[
S2(N) = \frac{S1(N)}{S3(N)}
\]

NOTE: N may be specified at Model Generation time, to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0
FIRST ORDER LAG ENGINE MODEL

\[
\begin{align*}
&\text{THR} \quad \frac{1}{TCO \cdot s + 1} \quad \text{TH} \\
&\text{BODY AXIS TRANSFORMATION} \quad F(3) \\
&\text{GAX, GAZ, XO, ZO}
\end{align*}
\]

<table>
<thead>
<tr>
<th>PHYSICAL</th>
<th>PORT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUANTITY</td>
<td>NO.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td></td>
<td>Engine time constant</td>
<td>sec</td>
</tr>
<tr>
<td>THR</td>
<td></td>
<td>Required thrust level</td>
<td>lbs</td>
</tr>
<tr>
<td>TCO</td>
<td></td>
<td>X, Z body axis direction cosines</td>
<td></td>
</tr>
<tr>
<td>GAX, GAZ</td>
<td></td>
<td>X, Z thrust location components</td>
<td>ft</td>
</tr>
<tr>
<td>XO, ZO</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PHYSICAL</th>
<th>PORT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUANTITY</td>
<td>NO.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td></td>
<td>Thrust output - state</td>
<td>lbs</td>
</tr>
<tr>
<td>TH</td>
<td></td>
<td>X, Y, Z body axis forces</td>
<td>lbs</td>
</tr>
<tr>
<td>F(3)</td>
<td></td>
<td>Axis torques (pitching moment)</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS
\[
\begin{align*}
\text{TH} &= \frac{\text{THR} - \text{TH}}{\text{TCO}} \\
\text{F}(1) &= \text{TH} \cdot \text{GAX} \\
\text{F}(2) &= 0 \\
\text{F}(3) &= \text{TH} \cdot \text{GAZ} \\
\text{T}(1) &= 0 \\
\text{T}(2) &= (\text{Z} \cdot \text{F}X - \text{X}) \cdot \text{FZ} \\
\text{T}(3) &= 0 \\
\text{TCO} &= \text{Yields} \\
\text{TH} &= \text{THR}
\end{align*}
\]
TORQUES-TO-FLEXIBLE MODE AMPLITUDE AND RATE 

\[
\begin{align*}
T(1) & \quad \rightarrow \quad \text{CX} \\
T(2) & \quad \rightarrow \quad \text{CY} \\
T(3) & \quad \rightarrow \quad \text{CZ}
\end{align*}
\]

\[
\begin{align*}
\frac{Gw_n^2}{s^2 + 2\zeta w_n s + w_n^2}
\end{align*}
\]

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T(3)</td>
<td></td>
<td>X, Y, Z body axis torques</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>GAI</td>
<td></td>
<td>Mode gain, G</td>
<td>rad/ft-lb-sec</td>
</tr>
<tr>
<td>DMP</td>
<td></td>
<td>Mode damping, (\zeta)</td>
<td></td>
</tr>
<tr>
<td>WN</td>
<td></td>
<td>Mode natural frequency, (w_n)</td>
<td>rad/sec</td>
</tr>
<tr>
<td>CX(3)</td>
<td></td>
<td>X, Y, Z body axis coefficients to convert</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>*A</td>
<td></td>
<td>Mode amplitude - state</td>
<td>rad</td>
</tr>
<tr>
<td>*AD</td>
<td></td>
<td>Mode rate - state</td>
<td>rad/sec</td>
</tr>
</tbody>
</table>

EQUATIONS OF MOTION:

\[
\begin{align*}
\dot{A}D &= ((GAI \cdot (CX(1) \cdot T(1) + CX(2) \cdot T(2) + CX(3) \cdot T(3)) - A) \cdot WN - 2 \cdot DMP \cdot AD) \cdot WN \\
\dot{\dot{A}} &= AD
\end{align*}
\]

NOTE:

This component is used with FP to produce angular rates due to flexible structure.
FLEXIBLE MODE AMPLITUDE-TO-DEFLECTIONS AND RATES

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>Mode amplitude</td>
<td>rad</td>
</tr>
<tr>
<td>AD</td>
<td></td>
<td>Mode rate</td>
<td>rad/sec</td>
</tr>
<tr>
<td>CW(3)</td>
<td></td>
<td>X, Y, Z body axis coefficients to convert mode amplitude to body axis rates</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

| PHYSICAL QUANTITY NAME | PORT NO. | DESCRIPTION | UNITS | |
|------------------------|----------|-------------|-------|
| EA(3)                  |          | X, Y, Z body axis angular deflections | rad   |
| W(3)                   |          | X, Y, Z body axis rates | rad/sec |

EQUATIONS:

ROL = CP*A
PIT = CQ*A
YAW = CR*A
P = CP*AD
Q = CQ*AD
R = CR*AD

VECTOR DEFINITIONS:

\[
CW(3) = \begin{pmatrix} CP \\ CQ \\ CR \end{pmatrix},
EA(3) = \begin{pmatrix} ROL \\ PIT \\ YAW \end{pmatrix},
W(3) = \begin{pmatrix} P \\ Q \\ R \end{pmatrix}
\]

NOTE: This component is used with FM to produce angular deflections and rates due to flexible structure.
### FUNCTION GENERATOR

#### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>AN</td>
<td></td>
<td>Degree of interpolation (AN&lt;0 prevents extrapolation)</td>
<td></td>
</tr>
<tr>
<td>FTA</td>
<td></td>
<td>Tabular values of function</td>
<td></td>
</tr>
</tbody>
</table>

#### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>2</td>
<td>Output quantity</td>
<td></td>
</tr>
</tbody>
</table>

**EQUATION:**

\[ S_2 = \text{FTA}(S_1) \]

**NOTE:** A maximum of 18 points is allowed in the table.
TWO-DIMENSIONAL FUNCTION

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>3</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>AN</td>
<td></td>
<td>Degree of interpolation for S1*</td>
<td></td>
</tr>
<tr>
<td>BN</td>
<td></td>
<td>Degree of interpolation for S3*</td>
<td></td>
</tr>
<tr>
<td>FTA</td>
<td></td>
<td>Table of functional relationships</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>2</td>
<td>Output quantity</td>
<td></td>
</tr>
</tbody>
</table>

EQUATION:

\[ S_2 = FTA(S_1, S_3) \]

* A negative value for AN or BN prevents extrapolation beyond the table boundaries
THREE-DIMENSIONAL FUNCTION

\[ S_2 = f(S_1, S_3, S_4) \]

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>1</td>
<td>INPUT QUANTITY</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>3</td>
<td>INPUT QUANTITY</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>4</td>
<td>INPUT QUANTITY</td>
<td></td>
</tr>
<tr>
<td>ANX</td>
<td></td>
<td>DEGREE OF INTERPOLATION FOR S1*</td>
<td></td>
</tr>
<tr>
<td>ANY</td>
<td></td>
<td>DEGREE OF INTERPOLATION FOR S3*</td>
<td></td>
</tr>
<tr>
<td>ANZ</td>
<td></td>
<td>DEGREE OF INTERPOLATION FOR S4*</td>
<td></td>
</tr>
<tr>
<td>FTA</td>
<td></td>
<td>TABLE OF FUNCTIONAL RELATIONSHIPS</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>2</td>
<td>OUTPUT QUANTITY</td>
<td></td>
</tr>
</tbody>
</table>

EQUATION: \( S_2 = FTS(S_1, S_3, S_4) \)

* A NEGATIVE VALUE PREVENTS EXTRAPOLATION BEYOND THE TABLE BOUNDARIES.
### FEEDER AND CIRCUIT BREAKER

#### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>FIGURE 3.0-1 NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDI</td>
<td>edl</td>
<td>Load Voltage, D-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>EQI</td>
<td>eql</td>
<td>Load Voltage, Q-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>ADI</td>
<td>id</td>
<td>Generator Current, D-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>AQI</td>
<td>iq</td>
<td>Generator Current, Q-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>AB</td>
<td>Ib</td>
<td>Base Value of Current, Peak</td>
<td>amps</td>
</tr>
<tr>
<td>RNL</td>
<td>RNL</td>
<td>No-Load Shunt Resistance</td>
<td>p.u.</td>
</tr>
<tr>
<td>RS1</td>
<td>Rs1</td>
<td>Simulated Breaker Open Circuit Resistance</td>
<td>p.u.</td>
</tr>
<tr>
<td>RW</td>
<td>Rw</td>
<td>Feeder Resistance</td>
<td>p.u.</td>
</tr>
<tr>
<td>XW</td>
<td>Xw</td>
<td>Feeder Resistance</td>
<td>p.u.</td>
</tr>
<tr>
<td>XC</td>
<td>Xc</td>
<td>No-Load Shunt Capacitive Reactance</td>
<td>p.u.</td>
</tr>
<tr>
<td>WO</td>
<td>w0</td>
<td>Base Frequency (WO = Wzero)</td>
<td>rads/sec</td>
</tr>
</tbody>
</table>

#### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>FIGURE 3.0-1 NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>*EDO</td>
<td>ed</td>
<td>Generator Terminal Voltage, D-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>*EQQ</td>
<td>eq</td>
<td>Generator Terminal Voltage, Q-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>*ADO</td>
<td>idl</td>
<td>Load Current, D-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>*AQO</td>
<td>q1</td>
<td>Load current, Q-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>ARO</td>
<td>Ip</td>
<td>Real Current</td>
<td>amps</td>
</tr>
<tr>
<td>AIO</td>
<td>Iq</td>
<td>Imaginary Current</td>
<td>amps</td>
</tr>
<tr>
<td>AT</td>
<td>T</td>
<td>Total Line Current, RMS</td>
<td>amps</td>
</tr>
<tr>
<td>PF</td>
<td></td>
<td>Power Factor</td>
<td></td>
</tr>
<tr>
<td>RTF</td>
<td></td>
<td>Intermediate Quantity</td>
<td>p.u.</td>
</tr>
<tr>
<td>PHI</td>
<td>δL</td>
<td>Load Voltage D-Q Angle</td>
<td>radians</td>
</tr>
</tbody>
</table>

* This output quantity is a state.
EQUATIONS:

\[
RTF = RS1 + RW
\]

\[
PHI = ATAN(EDI/EQI)
\]

\[
ARO = (AB/1.4142) \cdot (1/SQRT(EDI + EQI - EQI)) \cdot (EDI + ADO + EQI - AQO)
\]

\[
AIO = (AB/1.4142) \cdot (1/SQRT(EDI + EQI - EQI)) \cdot (ADO + EQI - EDI + AQO)
\]

\[
AT = SQRT(ARO \cdot ARO + AIO \cdot AIO)
\]

\[
PF = \cos(\text{ATAN}(AIO/ARO))
\]

\[
EDO = (WO/RNL) \cdot (-EDO \cdot XC + EQO \cdot RNL + ADI \cdot XC \cdot RNL - ADO - XC \cdot RNL)
\]

\[
EQO = (WO/RNL) \cdot (-EQO \cdot XC - EDO \cdot RNL - AEO \cdot XC \cdot RNL + AQI \cdot XC \cdot RNL)
\]

\[
ADO = (WO/XW) \cdot (-ADO \cdot RTF + ADO \cdot XW + EDO - EDI)
\]

\[
AQO = (WO/XW) \cdot (-AQO \cdot RTF - ADO \cdot XW - EQI + EQO)
\]
### GENERATOR - EXCITER

#### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>FIGURE 3.0-2 NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 i_d</td>
<td></td>
<td>Generator Current, D-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>A2 i_q</td>
<td></td>
<td>Generator Current, Q-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>A3 i_f</td>
<td></td>
<td>Generator Field Current</td>
<td>p.u.</td>
</tr>
<tr>
<td>*A4</td>
<td></td>
<td>Component of D-Axis Amotisseur Flux</td>
<td>p.u.</td>
</tr>
<tr>
<td>*A5</td>
<td></td>
<td>Component of Q-Axis Amotisseur Flux</td>
<td>p.u.</td>
</tr>
<tr>
<td>A7</td>
<td></td>
<td>Generator Saturation Correction Current</td>
<td>p.u.</td>
</tr>
<tr>
<td>A9 i_ef</td>
<td></td>
<td>Exciter Field Current</td>
<td>amps</td>
</tr>
<tr>
<td>E3 e_e</td>
<td></td>
<td>Exciter Output Voltage</td>
<td>volts</td>
</tr>
<tr>
<td>E4 e_e</td>
<td></td>
<td>Exciter Output - A.C. Voltage</td>
<td>volts</td>
</tr>
<tr>
<td>E5 e_e</td>
<td></td>
<td>Voltage Behind Exciter Transient Reactance</td>
<td>volts</td>
</tr>
<tr>
<td>E6 e_f</td>
<td></td>
<td>Generator Main Field Voltage</td>
<td>p.u.</td>
</tr>
<tr>
<td>*E7 i_o</td>
<td></td>
<td>Internal Parameter</td>
<td></td>
</tr>
<tr>
<td>*SD ψ_d</td>
<td></td>
<td>Armature Flux, D-Axis</td>
<td>p.u.</td>
</tr>
<tr>
<td>*SQ ψ_q</td>
<td></td>
<td>Armature Flux, Q-Axis</td>
<td>p.u.</td>
</tr>
<tr>
<td>*SMC</td>
<td></td>
<td>Internal Parameter</td>
<td></td>
</tr>
<tr>
<td>SM ψ_md</td>
<td></td>
<td>Mutual Flux, D-Axis</td>
<td>p.u.</td>
</tr>
<tr>
<td>SN</td>
<td></td>
<td>Input to Saturation Table, FA1</td>
<td>p.u.</td>
</tr>
<tr>
<td>TD T_D</td>
<td></td>
<td>Generator Output Torque</td>
<td>p.u.</td>
</tr>
</tbody>
</table>

*This output quantity is a state.*
### GENERATOR - EXCITER

#### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>FIGURE NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFB</td>
<td>Ifb</td>
<td>Exciter Current Base Value</td>
<td>amps</td>
</tr>
<tr>
<td>EFB</td>
<td>Efb</td>
<td>Exciter Voltage Base Value</td>
<td>volts</td>
</tr>
<tr>
<td>E1</td>
<td>ed</td>
<td>Generator Terminal Voltage, D-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>E2</td>
<td>eq</td>
<td>Generator Terminal Voltage, Q-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>E8</td>
<td>ef</td>
<td>Voltage Into Exciter Field</td>
<td>volts</td>
</tr>
<tr>
<td>FA1</td>
<td>f1</td>
<td>Generator Saturation Function (Table)</td>
<td></td>
</tr>
<tr>
<td>AM*</td>
<td>fef</td>
<td>Degree of Interpolation for Table FA1</td>
<td></td>
</tr>
<tr>
<td>FE4</td>
<td>fef</td>
<td>Exciter Saturation Function (Table)</td>
<td></td>
</tr>
<tr>
<td>AN*</td>
<td>k3</td>
<td>Exciter Current Rectification Constant</td>
<td></td>
</tr>
<tr>
<td>K2</td>
<td>k3</td>
<td>Exciter Voltage Rectification Constant</td>
<td></td>
</tr>
<tr>
<td>PSM</td>
<td>PSM</td>
<td>1/Time Constant</td>
<td>rad/sec</td>
</tr>
<tr>
<td>Default Value = 10000.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>Rkd</td>
<td>Amortisseur Resistance, D-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>R2</td>
<td>Rkq</td>
<td>Amortisseur Resistance, Q-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>R3</td>
<td>Rf</td>
<td>Generator Field Resistance</td>
<td>p.u.</td>
</tr>
<tr>
<td>R4</td>
<td>Ra</td>
<td>Armature Resistance Per Phase</td>
<td>p.u.</td>
</tr>
<tr>
<td>R5</td>
<td>Ref</td>
<td>Exciter Field Resistance</td>
<td>ohms</td>
</tr>
<tr>
<td>T1</td>
<td>Te</td>
<td>Exciter Field Open-Circuit Time Constant</td>
<td>secs</td>
</tr>
<tr>
<td>W0</td>
<td>wo</td>
<td>Base Frequency (WO = W_zero)</td>
<td>rad/sec</td>
</tr>
<tr>
<td>W</td>
<td>w</td>
<td>Input Speed</td>
<td>p.u.</td>
</tr>
<tr>
<td>X1</td>
<td>Xfl</td>
<td>Generator Field Leakage Reactance @ WO</td>
<td>p.u.</td>
</tr>
<tr>
<td>X2</td>
<td>Xmd</td>
<td>Mutual Reactance, D-Axis @ WO</td>
<td>p.u.</td>
</tr>
<tr>
<td>X3</td>
<td>Xmq</td>
<td>Mutual Reactance, Q-Axis @ WO</td>
<td>p.u.</td>
</tr>
<tr>
<td>X4</td>
<td>Xkd1</td>
<td>Amortisseur Leakage Reactance, D-Axis @ WO</td>
<td>p.u.</td>
</tr>
<tr>
<td>X5</td>
<td>Xkq1</td>
<td>Amortisseur Leakage Reactance, Q-Axis @ WO</td>
<td>p.u.</td>
</tr>
<tr>
<td>X6</td>
<td>Xa1</td>
<td>Armature Leakage Reactance @ WO</td>
<td>p.u.</td>
</tr>
<tr>
<td>X7</td>
<td>Xed</td>
<td>Synchronous Reactance, Exciter D-Axis</td>
<td>ohms</td>
</tr>
<tr>
<td>X8</td>
<td>Xed</td>
<td>Transient Reactance, Exciter D-Axis</td>
<td>ohms</td>
</tr>
<tr>
<td>K3</td>
<td>K3</td>
<td>Saturation Function Adjustment PSI-MD</td>
<td></td>
</tr>
<tr>
<td>Default Value = 1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K4</td>
<td>K4</td>
<td>Saturation Function Adjustment ED</td>
<td></td>
</tr>
<tr>
<td>Default Value = 0.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* A negative value prevents extrapolation beyond the table boundaries.
The input quantity is monitored during a SIMULATE analysis. When time reaches TMAX, a histogram is produced with 16 intervals that span the range from FUP to FLO. The histogram is drawn on page of the output history.
## HYSTERESIS

![Hysteresis Diagram]

### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>GAI(N)</td>
<td></td>
<td>Gain</td>
<td></td>
</tr>
<tr>
<td>DEL(N)</td>
<td></td>
<td>1/2 Hysteresis Band Width</td>
<td></td>
</tr>
</tbody>
</table>

### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output quantity</td>
<td></td>
</tr>
<tr>
<td>SL(N)</td>
<td></td>
<td>Previous input value</td>
<td></td>
</tr>
<tr>
<td>CU(N)</td>
<td></td>
<td>Curve number</td>
<td></td>
</tr>
<tr>
<td>TL</td>
<td></td>
<td>Previous time</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td></td>
<td>Precalculation indicator</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** N specifies the number of modes and is specified at Model Generation time. The default value of N is 1.0. This component is used in conjunction with ME, MM, MT, and MF.
**INTEGRATOR**

![Integrator Diagram]

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>GKI(N)</td>
<td></td>
<td>Integration gain</td>
<td></td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>*S(N)</td>
<td>2</td>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>

**EQUATIONS:**

\[
\hat{S}_2 = GKI \cdot S_1
\]

*This output is a state*

**NOTE:** N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0
INTEGRATOR WITH SATURATION

\[ \dot{s}_2 = GKI(s_1 - GKL(s_2 - AMA)), \text{ if } s_2 > AMA \]
\[ \dot{s}_2 = GKI \cdot s_1, \text{ if } AMI \leq s_2 \leq AMA \]
\[ \dot{s}_2 = GKI(s_1 - GKL(s_2 - AMI)), \text{ if } s_2 < AMI \]

* This output is a state

NOTE: \( N \) may be specified at Model Generation time to allow inputs and outputs to be \( N \) dimensional vectors. Default value of \( N \) is 1.0.
INFINITE BUS

\[ \frac{2}{\omega_0} \quad \omega \quad \frac{1}{\rho} \quad \delta(TA) \text{ radians} \]

\[ e_{q_1} \quad \cos \delta \]

\[ e_{d_1} \quad \sin \delta \overrightarrow{e_{\text{infinite bus}}} \]

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>FIGURE NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER</td>
<td>E INFINITE BUS</td>
<td>VOLTAGE AT INFINITE BUS</td>
<td>PU</td>
</tr>
<tr>
<td>WI</td>
<td>(\omega) INFINITE BUS</td>
<td>FREQUENCY AT INFINITE BUS</td>
<td>RAD/SEC</td>
</tr>
<tr>
<td>W0</td>
<td>(\omega_0)</td>
<td>BASE FREQUENCY</td>
<td>PU</td>
</tr>
<tr>
<td>W1</td>
<td>(\omega_1)</td>
<td>SHAFT ROTATION RATE</td>
<td>RAD/SEC</td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>FIGURE NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>(e_{d1})</td>
<td>DIRECT VOLTAGE</td>
<td>PU</td>
</tr>
<tr>
<td>E2</td>
<td>(e_{q1})</td>
<td>QUADRATURE VOLTAGE</td>
<td>PU</td>
</tr>
<tr>
<td>WA</td>
<td>(\omega)</td>
<td>GENERATOR FREQUENCY</td>
<td>RAD/SEC</td>
</tr>
<tr>
<td>*TA</td>
<td>(\delta(TA))</td>
<td>TORQUE ANGLE</td>
<td>RADIANS</td>
</tr>
<tr>
<td>TX</td>
<td>-</td>
<td>TORQUE ANGLE</td>
<td>DEGREES</td>
</tr>
</tbody>
</table>

* THIS OUTPUT QUANTITY IS A STATE
FIRST ORDER LAG TRANSFER FUNCTION

\[
\begin{array}{c}
\frac{GAI}{TC \cdot S + 1} \\
\end{array}
\]

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>GAI(N)</td>
<td></td>
<td>Gain</td>
<td></td>
</tr>
<tr>
<td>TC(N)</td>
<td></td>
<td>Time constant</td>
<td>seconds</td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS:
\[ S2 = \frac{(GAI \cdot S1 - S2)}{TC} \]

NOTE: D.C. gain = GAI
and time constant = TC, seconds

\begin{align*}
\text{infinite freq. gain} &= 0 \\
\text{pole location} &= \frac{1}{TC} \text{ rad/sec}
\end{align*}

* This output is a state

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0
<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>YB, YBD</td>
<td></td>
<td>Side force coefficients:*&lt;br&gt;<strong>Beta and Beta dot coeff. (nondim.)</strong>&lt;br&gt;<strong>V and V dot coeff. (dim.)</strong></td>
<td>lb-sec/ft</td>
</tr>
<tr>
<td>YP, YR</td>
<td></td>
<td>P and R angular rate coefficients</td>
<td>lb-sec/deg</td>
</tr>
<tr>
<td>YDR, YDA</td>
<td></td>
<td>Rudder and aileron coefficients</td>
<td></td>
</tr>
<tr>
<td>LB, LBD</td>
<td></td>
<td>Rolling moment coefficients:*&lt;br&gt;<strong>Beta and Beta dot coeff. (nondim.)</strong>&lt;br&gt;<strong>V and V dot coeff. (dim.)</strong></td>
<td>lb-sec</td>
</tr>
<tr>
<td>LP, LR</td>
<td></td>
<td>P and R angular rate coefficient</td>
<td>ft-lb-sec/deg</td>
</tr>
<tr>
<td>LDR, LOA</td>
<td></td>
<td>Rudder and aileron coefficients</td>
<td>ft-lb/deg</td>
</tr>
<tr>
<td>NB, NBD</td>
<td></td>
<td>Yawing moment coefficients:*&lt;br&gt;<strong>Beta and Beta dot coeff. (nondim.)</strong>&lt;br&gt;<strong>V and V dot coeff. (dim.)</strong></td>
<td>lb-sec</td>
</tr>
<tr>
<td>NP, NR</td>
<td></td>
<td>P and R angular rate coefficients</td>
<td>ft-lb-sec/deg</td>
</tr>
<tr>
<td>NDR, NDA</td>
<td></td>
<td>Rudder and aileron coefficients</td>
<td>ft-lb/deg</td>
</tr>
<tr>
<td>RUD, AIL</td>
<td></td>
<td>Control Surfaces:*&lt;br&gt;Rudder and aileron deflections</td>
<td></td>
</tr>
<tr>
<td>UD, WD</td>
<td></td>
<td>Longitudinal accelerations:*&lt;br&gt;X and Z body axis acceleration</td>
<td>ft/sec²</td>
</tr>
<tr>
<td>F(3)</td>
<td></td>
<td>External forces*</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td></td>
<td>External torques*</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>UO(3)</td>
<td></td>
<td>X, Y, Z body axis velocities</td>
<td>ft/sec</td>
</tr>
<tr>
<td>WO(3)</td>
<td></td>
<td>X, Y, Z body axis angular rates</td>
<td>deg/sec</td>
</tr>
<tr>
<td>BE</td>
<td></td>
<td>Sideslip angle</td>
<td>deg</td>
</tr>
<tr>
<td>EV</td>
<td></td>
<td>Y body axis acceleration term for VO</td>
<td>ft/sec²</td>
</tr>
<tr>
<td>VT</td>
<td></td>
<td>True airspeed</td>
<td>ft/sec</td>
</tr>
<tr>
<td>QS</td>
<td></td>
<td>Dynamic pressure x reference area</td>
<td>lbs</td>
</tr>
<tr>
<td>RW</td>
<td></td>
<td>Y body axis angular rate gust</td>
<td>deg/sec</td>
</tr>
</tbody>
</table>

VECTOR DEFINITIONS:

\[
F(3) = \begin{pmatrix} FX \\ FY \\ FZ \end{pmatrix} \quad T(3) = \begin{pmatrix} TX \\ TY \\ TZ \end{pmatrix} \quad UO(3) = \begin{pmatrix} UO \\ VO \end{pmatrix} \quad WO(3) = \begin{pmatrix} PO \\ QO \end{pmatrix}
\]

* Small Beta angle approximation
### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td></td>
<td>Indicator function for coefficients&lt;br&gt;0 = body axis, dim.&lt;br&gt;1 = body axis, nondim.&lt;br&gt;2 = stability axis, dim.&lt;br&gt;3 = stability axis, nondim.&lt;br&gt;Direction cosines for body or stability axes, depending on ID</td>
<td></td>
</tr>
<tr>
<td>CAL, SAL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FY</td>
<td>2</td>
<td>Y body axis force sum</td>
<td>lbs</td>
</tr>
<tr>
<td>VD</td>
<td></td>
<td>Y body axis acceleration</td>
<td>ft/sec²</td>
</tr>
<tr>
<td>TX, TZ</td>
<td>2</td>
<td>X and Z axis (ROLL and YAW) moments</td>
<td>ft-lb</td>
</tr>
</tbody>
</table>

### CONSTANTS

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA</td>
<td></td>
<td>Rigid body mass</td>
<td>slugs</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>Wing span</td>
<td>ft</td>
</tr>
<tr>
<td>XP*</td>
<td></td>
<td>X axis c.p. - c.g.</td>
<td>ft</td>
</tr>
</tbody>
</table>

---

537
LATERAL-AERODYNAMIC FORCES AND MOMENTS (Implicit form)

DIMENSIONAL EQUATIONS:

\[
F_{Y_{aero}} = YB\cdot VO + YBD\cdot(V+W) + YP\cdot PO + YR\cdot RO + YDR\cdot RUD + YDA\cdot AIL,
\]

where

\[
\dot{V} = VD = FY2/MA + EW
\]

\[
\dot{W} = RW\cdot VT\cdot\pi/180
\]

\[
T_{X_{aero}} = LB\cdot VO + LBD(\dot{V}+\dot{W}) + LP\cdot PO + LR\cdot RO + LDR\cdot RUD + LDA\cdot AIL
\]

\[
T_{Z_{aero}} = NB\cdot VO + NBD(\dot{V}+\dot{W}) + NP\cdot PO + NR\cdot RO + NDR\cdot RUD + NDA\cdot AIL
\]

NONDIMENSIONAL EQUATIONS:

\[
F_{Y_{aero}} = QS\cdot(YB\cdot BE + (YBD\cdot BETA + YP\cdot \hat{P} + YR\cdot \hat{R})\cdot B/(2\cdot VT) + YDR\cdot RUD + YDA\cdot AIL),
\]

where

\[
BETA = \dot{V}\cdot(1 - BE^2)/VT - BE\cdot(UO\cdot UD + WO\cdot WD)/VT^2 + RW*
\]

\[
\hat{BE} = BE\cdot\pi/180, \text{ etc. for } \hat{P}, \hat{R}, \hat{RUD}, \hat{AIL}, \hat{RW}
\]

\[
T_{X_{aero}} = QS\cdot B\cdot(LB\cdot BE + (LBD\cdot BETA + LP\cdot \hat{P} + LR\cdot \hat{R})\cdot B/(2\cdot VT) + LDR\cdot RUD + LDA\cdot AIL)
\]

\[
T_{Z_{aero}} = QS\cdot B\cdot(NB\cdot BE + (NBD\cdot BETA + NP\cdot \hat{P} + NR\cdot \hat{R})\cdot B/(2\cdot VT) + NDR\cdot RUD + NDA\cdot AIL)
\]

FORCE AND TORQUE SUM:

\[
FY_2 = FY_{aero} + FY_1
\]

\[
TX_2 = \left\{ \begin{array}{l}
T_{X_{aero}} + TX_1 \\
T_{X_{aero}} \cdot CAL - T_{Z_{aero}} \cdot SAL + TX_1
\end{array} \right. \quad \text{ID} = 0, 1, 2, 3
\]

\[
TZ_2 = \left\{ \begin{array}{l}
T_{Z_{aero}} + TZ_1 + XP\cdot FY_{aero} \\
T_{Z_{aero}} \cdot CAL + T_{X_{aero}} \cdot SAL + XP\cdot FY_{aero}
\end{array} \right. \quad \text{ID} = 0, 1, 2, 3
\]

*Small Beta angle approximation*
## FIRST ORDER LEAD-LAG FUNCTION

### LE

\[ \frac{GAI(S + ZO)}{S + PO} \]

### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>GAI(N)</td>
<td></td>
<td>Infinite frequency gain</td>
<td></td>
</tr>
<tr>
<td>ZO(N)</td>
<td></td>
<td>Numerator coefficient</td>
<td>rad/sec</td>
</tr>
<tr>
<td>PO(N)</td>
<td></td>
<td>Denominator coefficient</td>
<td>rad/sec</td>
</tr>
</tbody>
</table>

### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X(N)</td>
<td></td>
<td>Intermediate quantity</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output quantity - variable</td>
<td></td>
</tr>
</tbody>
</table>

### EQUATIONS:

\[
S^2 = GAI \cdot S + X1
\]

\[
X1 = GAI \cdot S1 \cdot ZO - S2 \cdot PO
\]

### NOTE:

- d.c. gain = \( \frac{GAI \cdot ZO}{PO} \)
- zero location = -ZO
- infinite frequency gain = GAI
- pole location = -PO

### NOTE:

N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0.
FIRST ORDER LAG TRANSFER FUNCTION

\[ \frac{Z_0}{S + P_0} \]

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>ZO(N)</td>
<td></td>
<td>Numerator coefficient</td>
<td>rad/sec</td>
</tr>
<tr>
<td>PO(N)</td>
<td></td>
<td>Denominator coefficient</td>
<td>rad/sec</td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output quantity (state)</td>
<td></td>
</tr>
</tbody>
</table>

**EQUATION:**

\[ S_2 = Z_0 \cdot S_1 - P_0 \cdot S_2 \]

**NOTE:**

- d.c. gain = \( \frac{Z_0}{P_0} \)
- time constant = \( \frac{1}{P_0} \)
- infinite frequency gain = 0
- pole location = -P_0

**NOTE:** N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0
LEAD-LAG TRANSFER FUNCTION

\[
\frac{GAI(TC1 \cdot S + 1)}{TC2 \cdot S + 1}
\]

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>TC1(N)</td>
<td></td>
<td>Numerator time constant</td>
<td>sec</td>
</tr>
<tr>
<td>TC2(N)</td>
<td></td>
<td>Denominator time constant</td>
<td>sec</td>
</tr>
<tr>
<td>GAI(N)</td>
<td></td>
<td>Gain</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>*X1(N)</td>
<td></td>
<td>Intermediate quantity (state)</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output quantity (variable)</td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS:

\[
S_2 = (X_1 + S_1 \cdot TC1 \cdot GAI) / TC2
\]

\[
\dot{X}_1 = GAI \cdot S_1 - S_2
\]

NOTE:

d.c. gain = GAI
infinite gain = \( \frac{GAI \cdot TC1}{TC2} \)
zero location = \(-\frac{TC1}{1}\), rad/sec
pole location = \(-\frac{TC2}{1}\), rad/sec

*This output quantity is a state

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0
<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X axis force coefficients:*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XO</td>
<td></td>
<td>Bias coeff. for trim</td>
<td>lbs</td>
</tr>
<tr>
<td>XA</td>
<td></td>
<td>Alpha coeff. (nondim.)</td>
<td></td>
</tr>
<tr>
<td>ZA</td>
<td></td>
<td>Z axis velocity coeff. (dim.)</td>
<td>lb-sec/ft</td>
</tr>
<tr>
<td>XU</td>
<td></td>
<td>X axis velocity coeff.</td>
<td>lb-sec/ft</td>
</tr>
<tr>
<td>XDE</td>
<td></td>
<td>Elevator coefficient</td>
<td>lb/deg</td>
</tr>
<tr>
<td>Z axis force coefficients:*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZO</td>
<td></td>
<td>Bias coeff. for trim</td>
<td>lbs</td>
</tr>
<tr>
<td>ZA, ZAD</td>
<td></td>
<td>Alpha and Alpha dot coeff. (nondim.)</td>
<td></td>
</tr>
<tr>
<td>ZQ</td>
<td></td>
<td>Z axis velocity</td>
<td>lb-sec/ft</td>
</tr>
<tr>
<td>ZU</td>
<td></td>
<td>X axis velocity coeff.</td>
<td>lb-sec/ft</td>
</tr>
<tr>
<td>ZDE</td>
<td></td>
<td>Elevator coeff.</td>
<td>lb/deg</td>
</tr>
<tr>
<td>Pitching moment coefficients:*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MO</td>
<td></td>
<td>Bias coeff. for trim</td>
<td>ft-lb</td>
</tr>
<tr>
<td>MAL, MAD</td>
<td></td>
<td>Alpha and Alpha dot coeff. (nondim.)</td>
<td></td>
</tr>
<tr>
<td>MQ</td>
<td></td>
<td>Q angular rate coeff.</td>
<td>ft-lb-sec/deg</td>
</tr>
<tr>
<td>MU</td>
<td></td>
<td>X axis velocity coeff.</td>
<td>lb-sec</td>
</tr>
<tr>
<td>MDE</td>
<td></td>
<td>Elevator coeff.</td>
<td>ft-lb/deg</td>
</tr>
<tr>
<td>Constants:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MA</td>
<td>1</td>
<td>Rigid body mass</td>
<td>slugs</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>Mean and aerodynamic chord</td>
<td>ft</td>
</tr>
<tr>
<td>XP*</td>
<td>1</td>
<td>X axis distance: c.p. - c.g.</td>
<td>ft</td>
</tr>
<tr>
<td>ID</td>
<td></td>
<td>Indicator function for coefficients</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = body axis, dim.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = body axis, nondim.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = stability axis, dim.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = stability axis, nondim.</td>
<td></td>
</tr>
</tbody>
</table>

*Default values = 0
### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAL, SAL</td>
<td></td>
<td>Direction cosines for stability, body axes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>External forces and moments:*</td>
<td></td>
</tr>
<tr>
<td>F(3)</td>
<td></td>
<td>X, Y, Z body axis forces</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td></td>
<td>Body axis (pitching) moment</td>
<td>ft-lb</td>
</tr>
<tr>
<td>Aero-Variables:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELE*</td>
<td></td>
<td>Elevator deflection</td>
<td>deg</td>
</tr>
<tr>
<td>AL, ALP</td>
<td></td>
<td>Alpha in body and stability axes</td>
<td>deg</td>
</tr>
<tr>
<td>UO</td>
<td></td>
<td>X body axis velocity</td>
<td>ft/sec</td>
</tr>
<tr>
<td>UP, WP</td>
<td></td>
<td>X and Z perturbation velocities (nondim.)</td>
<td></td>
</tr>
<tr>
<td>VT</td>
<td></td>
<td>X and Z stability axes velocities (dim.)</td>
<td>ft/sec</td>
</tr>
<tr>
<td>QS</td>
<td></td>
<td>True airspeed</td>
<td>ft/sec</td>
</tr>
<tr>
<td>QO, QW</td>
<td></td>
<td>Dynamic pressure x reference area</td>
<td>lbs</td>
</tr>
<tr>
<td>EU(3)</td>
<td></td>
<td>Y body axis angular rate, rate gust</td>
<td>deg/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X, Y, Z axis accel. terms for UD, WD</td>
<td>ft/sec^2</td>
</tr>
</tbody>
</table>

### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FX, FZ</td>
<td>2</td>
<td>X and Z body axis force sum</td>
<td>lbs</td>
</tr>
<tr>
<td>TY</td>
<td>2</td>
<td>Y body axis (pitching) moment</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>UD, WD</td>
<td>2</td>
<td>X and Z body axis acceleration</td>
<td>ft/sec^2</td>
</tr>
<tr>
<td>MA</td>
<td>2</td>
<td>Rigid body mass</td>
<td>slugs</td>
</tr>
<tr>
<td>XP</td>
<td>2</td>
<td>X axis distance: c.p. - c.g.</td>
<td>ft</td>
</tr>
</tbody>
</table>

*Default value = 0

VECTOR DEFINITIONS:

\[
F(3) = \begin{pmatrix} FX \\ FY \\ FZ \end{pmatrix}, \quad T(3) = \begin{pmatrix} TX \\ TY \\ TZ \end{pmatrix}, \quad EU(3) = \begin{pmatrix} EU \\ EV \\ EW \end{pmatrix}
\]
LONGITUDINAL AERO - FORCES AND MOMENTS
(Implicit Form)

DIMENSIONAL EQUATIONS

\[ F_{x,aero} = x_0 + x_a \cdot w + x_u \cdot u + x_{de\cdot e} \]
\[ F_{z,aero} = z_0 + z_a \cdot w + z_{ad}(\dot{w} + \dot{w}) + z_q \cdot q + z_u \cdot u + z_{de\cdot e} \]

where
- \( \dot{w} = w_d - u \cdot s_al \cdot w \)
- \( \dot{w} = -q \cdot v + v \)
- \( t_{y,aero} = m_0 + m_al \cdot w + m_{ad}(\dot{w} + \dot{w}) + m_q \cdot q + m_u \cdot u + m_{de\cdot e} \)

NONDIMENSIONAL EQUATIONS

\[ F_{x,aero} = q_s \cdot (x_0 + x_a \cdot \alpha_p + x_u \cdot u + x_{de\cdot e}) \]
\[ F_{z,aero} = q_s \cdot (z_0 + z_a \cdot \alpha_p + (z_{ad} - \alpha - q \cdot w) + z_q \cdot q) \cdot c/(2 \cdot v + z_u \cdot u + z_{de\cdot e}) \]

where
- \( \alpha = (w_d - \alpha_l \cdot u)/(u^2) \)
- \( \alpha_l = \alpha_p \cdot m/180 \), etc., for \( e \cdot e, \cdot q, q, \cdot a \)
- \( t_{y,aero} = q_s \cdot c \cdot (m_0 + m_al \cdot \alpha_p + (m_{ad} - \alpha - q \cdot w) + m_q \cdot q) \cdot c/(2 v) \cdot m_u \cdot u + m_{de\cdot e} \)

FORCE AND TORQUE SUM

\[ F_{x,\text{sum}} = F_{x,aero} + F_{x1} \cdot \text{cal} + F_{z1} \cdot \text{sal} \]
\[ F_{z,\text{sum}} = F_{z,aero} + F_{z1} \cdot \text{cal} - F_{x1} \cdot \text{sal} \]
\[ F_{x2} = F_{x,\text{sum}} \cdot \text{cal} - F_{z,\text{sum}} \cdot \text{sal} \]
\[ F_{z2} = F_{z,\text{sum}} \cdot \text{cal} + F_{x,\text{sum}} \cdot \text{sal} \]
\[ t_{y2} = t_{y,aero} + t_{y1} - \chi \cdot (F_{z,aero} \cdot \text{cal} + F_{x,aero} \cdot \text{sal}) \]

ACCELERATIONS

\( u_d = F_{x2} / m_a + e_u \)
\( w_d = F_{z2} / m_a + e_w \)

*Small alpha angle approximation.
### LOAD INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>FIGURE 3.0-3 NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADI</td>
<td>$i_d$</td>
<td>Load Current, D-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>AQI</td>
<td>$i_q$</td>
<td>Load Current, Q-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>RS2</td>
<td>$R_s$</td>
<td>Linear Load, Simulated Open-Circuit Resistance</td>
<td>p.u.</td>
</tr>
<tr>
<td>RL</td>
<td>$R_L$</td>
<td>Linear Load Resistance</td>
<td>p.u.</td>
</tr>
<tr>
<td>RNL</td>
<td>$R_{NL}$</td>
<td>No-Load Shunt Resistance</td>
<td>p.u.</td>
</tr>
<tr>
<td>XL</td>
<td>$X_L$</td>
<td>Linear Load Reactance</td>
<td>p.u.</td>
</tr>
<tr>
<td>XC</td>
<td>$X_C$</td>
<td>No-Load Shunt Capacitive Reactance</td>
<td>p.u.</td>
</tr>
<tr>
<td>WO</td>
<td>$\omega_0$</td>
<td>Base Frequency ($W_0 = \omega_{\text{zero}}$)</td>
<td>rad/sec</td>
</tr>
</tbody>
</table>

### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>FIGURE 3.0-3 NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>*EDO</td>
<td>$e_d$</td>
<td>Load Voltage, D-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>*EQO</td>
<td>$e_q$</td>
<td>Load Voltage, Q-Axis Component</td>
<td>p.u.</td>
</tr>
<tr>
<td>*ADS</td>
<td>$f_d$</td>
<td>Intermediate Quantity (State)</td>
<td>p.u.</td>
</tr>
<tr>
<td>*AQS</td>
<td>$f_q$</td>
<td>Intermediate Quantity (State)</td>
<td>p.u.</td>
</tr>
<tr>
<td>RTL</td>
<td></td>
<td>Intermediate Quantity</td>
<td>p.u.</td>
</tr>
</tbody>
</table>

EQUATIONS:

- $E_D = W_0 \cdot X_C \cdot (-E_D/R_{NL} + A_D + A_S + E_Q/O/X_C)$
- $E_Q = W_0 \cdot X_C \cdot (-E_Q/R_{NL} + A_Q - A_S - E_D/O/X_C)$
- $A_D = -A_D \cdot W_0 \cdot R_T/2 + E_D \cdot W_0 \cdot X_L + A_S \cdot W_0$
- $A_S = -A_S \cdot W_0 \cdot R_T/2 + E_Q \cdot W_0 \cdot X_L - A_D \cdot W_0$
- $R_T = R_S + R_L$

*This output quantity is a state.*
MULTIPLY AND ADD

\[ S_2 = C_1 \cdot S_1 + C_2 \]

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>C1(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>C2(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output quantity</td>
<td></td>
</tr>
</tbody>
</table>

EQUATION:
\[ S_2 = C_1 \cdot S_1 + C_2 \]

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1
MULTIPLY AND ADD

\[ S_2 = C_1S_1 + C_2S_3 + C_3S_4 + C_4 \]

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>3</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>4</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>C1(N)</td>
<td></td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>C2(N)</td>
<td></td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>C3(N)</td>
<td></td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>C4(N)</td>
<td></td>
<td>Input quantity</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output quantity</td>
<td></td>
</tr>
</tbody>
</table>

EQUATION:

\[ S_2 = C_1S_1 + C_2S_3 + C_3S_4 + C_4 \]

**NOTE:** \( N \) may be specified at Model Generation time to allow inputs and outputs to be \( N \) dimensional vectors. Default value of \( N \) is 1.0.

547
STRUCTURAL MODE DYNAMICS

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Mode excitation</td>
<td></td>
</tr>
<tr>
<td>DMP(N)</td>
<td></td>
<td>Mode damping</td>
<td></td>
</tr>
<tr>
<td>WN(N)</td>
<td></td>
<td>Mode natural frequency - ( \omega )</td>
<td>rad/sec</td>
</tr>
<tr>
<td>GAI(N)</td>
<td></td>
<td>Mode gain at scale factor</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q(N)</td>
<td></td>
<td>Mode position (state)</td>
<td>1/sec</td>
</tr>
<tr>
<td>QD(N)</td>
<td></td>
<td>Mode velocity (state)</td>
<td>1/sec</td>
</tr>
<tr>
<td>QDD(N)</td>
<td></td>
<td>Mode acceleration</td>
<td>1/sec²</td>
</tr>
</tbody>
</table>

EQUATIONS:

\[
\begin{align*}
QDD(I) &= GAI \cdot S(I) - WN(I) \cdot (WN(I) \cdot Q(I) + 2 \cdot DMP(I) \cdot QD(I)) \\
QD(I) &= QDD(I) \\
Q(I) &= QD(I) \quad I = 1, 2, ..., N
\end{align*}
\]

Freezing \( Q(I) \) causes \( QD(I) \) to be frozen and \( QDD(I) \) to be set to zero, thus removing all effects of that mode from the model.

\( N \) specifies the number of modes, and is specified at Model Generation time. The default value of \( N \) is 1.0. This component is used in conjunction with ME, MM, MT, and MF.
STRUCTURAL MODE EXCITATION

\[ S = \text{DCM} \times F \]

**INPUTS**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F(3)</td>
<td></td>
<td>Disturbance Force or Torque</td>
<td>lbs or ftlbs</td>
</tr>
<tr>
<td>DCM(N, 3)</td>
<td></td>
<td>Disturbance coefficient matrix</td>
<td></td>
</tr>
</tbody>
</table>

**OUTPUTS**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Mode excitation</td>
<td>–</td>
</tr>
</tbody>
</table>

N specifies the number of modes and is specified at Model Generation time. The default value of N is 1.

This component is used in conjunction with MD, MM, MT, and MF.
MF

FOUR VECTOR SUM

\[ S_2 = S_1 + S_3 + S_4 + S_5 \]

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ S(N) ] 1</td>
<td>[ S(N) ]</td>
<td>1</td>
<td>Input quantity</td>
<td>–</td>
</tr>
<tr>
<td>[ S(N) ] 3</td>
<td>[ S(N) ]</td>
<td>3</td>
<td>Input quantity</td>
<td>–</td>
</tr>
<tr>
<td>[ S(N) ] 4</td>
<td>[ S(N) ]</td>
<td>4</td>
<td>Input quantity</td>
<td>–</td>
</tr>
<tr>
<td>[ S(N) ] 5</td>
<td>[ S(N) ]</td>
<td>5</td>
<td>Input quantity</td>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OUTPUTS</th>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ S(N) ] 2</td>
<td>[ S(N) ]</td>
<td>2</td>
<td>Output quantity</td>
<td>–</td>
</tr>
</tbody>
</table>

N specifies the number of modes and is specified at Model Generation time. The default of N is 1.

This component is used in conjunction with MD, MM, MT—MF.
**EARTH'S MAGNETIC FIELD MODEL**

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLG(4)</td>
<td></td>
<td>Position in local geographic coordinates</td>
<td></td>
</tr>
<tr>
<td>IM</td>
<td></td>
<td>Flag ( { ) = 0 Magnetic field in TESLA ( } ) = 1 Magnetic field in gauss</td>
<td></td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMG(4)</td>
<td></td>
<td>Magnetic field data</td>
<td></td>
</tr>
</tbody>
</table>

VECTOR DESCRIPTION:

- PLG(1) = Distance from geocenter, earth radii - dimensionless
- PLG(2) = Co-latitude = \( \pi/2 \) - Geographic north latitude, radians
- PLG(3) = Geographic east longitude, radians
- PLG(4) = \( i \), orbit inclination measured at ascending node, radians
- AMG(1) = Magnitude of magnetic field, tesla or gauss
- AMG(2) = Magnetic field along line of flight, tesla or gauss
- AMG(3) = Magnetic field perpendicular to orbit plane, tesla or gauss
- AMG(4) = Magnetic field along local vertical, tesla or gauss
STRUCTURAL MODE MOTION

\[ X = \text{PCM} \times Q \]
\[ XD = \text{PCM} \times QD \]
\[ XDD = \text{PCM} \times QDD \]

INPUTS

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q(N)</td>
<td></td>
<td>Mode position</td>
<td>-</td>
</tr>
<tr>
<td>QD(N)</td>
<td></td>
<td>Mode velocity 1/sec</td>
<td>1/sec</td>
</tr>
<tr>
<td>QDD(N)</td>
<td></td>
<td>Mode acceleration 1/sec^2</td>
<td>1/sec^2</td>
</tr>
<tr>
<td>PCM(3,N)</td>
<td></td>
<td>Position coefficient matrix</td>
<td>*u</td>
</tr>
</tbody>
</table>

OUTPUTS

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X(3)</td>
<td></td>
<td>Position due to modes</td>
<td>*u</td>
</tr>
<tr>
<td>XD(3)</td>
<td></td>
<td>Velocity due to modes</td>
<td>*u/sec</td>
</tr>
<tr>
<td>XDD(3)</td>
<td></td>
<td>Acceleration due to modes</td>
<td>*u/sec^2</td>
</tr>
</tbody>
</table>

*Units depend on units used in PCM.

N specifies the number of modes and is specified at Model Generation time. The default value of N is 1.

This component is used in conjunction with ME, MD, MT, and MF.
### MOTOR RIPPLE

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC</td>
<td></td>
<td>Torque motor command</td>
<td>ma</td>
</tr>
<tr>
<td>AL</td>
<td></td>
<td>Torque motor angle</td>
<td>rad</td>
</tr>
<tr>
<td>GTM</td>
<td></td>
<td>Torque motor gain</td>
<td>ft-lb/ma</td>
</tr>
<tr>
<td>TMS</td>
<td></td>
<td>Torque motor saturation limit</td>
<td>ma</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>Array of Ripple Model coefficients and frequencies (See below)</td>
<td></td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR</td>
<td></td>
<td>Motor torque</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>

**EQUATIONS:**

\[
TCS = GTM \cdot \text{Satur} (TC, TMS)
\]

\[
TR = TCS \cdot (1 + C(4) \cdot \sin(C(14) \cdot AL) + C(5) \cdot \cos(C(15) \cdot AL) + C(6) \cdot \sin(C(16) \cdot AL) + C(7) \cdot \cos(C(17) \cdot AL) + C(8) \cdot TCS \cdot TCS + C(9) \cdot \text{ABS}(TCS) + C(10) \cdot \sin(C(18) \cdot AL) + C(11) + \sin(C(21) \cdot AL) + C(12) \cdot \cos(C(22) \cdot AL) + C(13) \cdot \sin(C(23) \cdot AL))
\]

**RIPPLE MODEL COEFFICIENTS & FREQUENCIES:**

<table>
<thead>
<tr>
<th>RIPPLE MODEL COMPONENT</th>
<th>COEFFICIENT</th>
<th>FREQUENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hall probe null</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>Common node</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>Hall probe placement</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>Unequal gains</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>Magnetic field</td>
<td>8, 9, 10</td>
<td>18</td>
</tr>
<tr>
<td>Offset currents</td>
<td>11, 12</td>
<td>21, 22</td>
</tr>
<tr>
<td>Reluctance (Cogging)</td>
<td>13</td>
<td>23</td>
</tr>
</tbody>
</table>
TWO VECTOR SUM

\[ S_2 = S_1 + S_3 \]

INPUTS

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>3</td>
<td>Input quantity</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUTS

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output quantity</td>
<td></td>
</tr>
</tbody>
</table>

N specifies the number of modes and is specified at Model Generation time. The default value of N is 1. This component is used in conjunction with MT—ME.
OPTIMAL CONTROLLER

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>All optimal controller inputs are defined via the O.C. INPUTS command in the EASY Model Generation Program.</td>
<td></td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>All optimal controller outputs are defined via the O.C. OUTPUTS command in the EASY Model Generation Program.</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** Due to its very general nature, the O.C. component is specified by a special set of Model Generation and Analysis commands, which all start with the letters O.C. (See Section 4.13)
# Power Factor Controller

## Input

<table>
<thead>
<tr>
<th>Physical Quantity Name</th>
<th>Figure Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>ED</td>
<td>ED</td>
<td>D Axis Voltage</td>
<td>Per Unit</td>
</tr>
<tr>
<td>EQ</td>
<td>EQ</td>
<td>Q Axis Voltage</td>
<td>Per Unit</td>
</tr>
<tr>
<td>AD</td>
<td>AD</td>
<td>D Axis Current</td>
<td>Per Unit</td>
</tr>
<tr>
<td>AQ</td>
<td>AQ</td>
<td>Q Axis Current</td>
<td>Per Unit</td>
</tr>
<tr>
<td>X1</td>
<td>X1</td>
<td>Lead Time Constant</td>
<td>Sec</td>
</tr>
<tr>
<td>X2</td>
<td>X2</td>
<td>Lead Time Constant</td>
<td>Sec</td>
</tr>
<tr>
<td>X3</td>
<td>X3</td>
<td>Integral Gain (Inverse)</td>
<td>-</td>
</tr>
<tr>
<td>X4</td>
<td>X4</td>
<td>Lag Time Constant</td>
<td>Sec</td>
</tr>
<tr>
<td>PFR</td>
<td>PFR</td>
<td>Power Reference Factor</td>
<td></td>
</tr>
<tr>
<td>AB</td>
<td>AB</td>
<td>Base Line Current</td>
<td>Amps</td>
</tr>
<tr>
<td>VB</td>
<td>VB</td>
<td>Base Line Voltage</td>
<td>(See Code)</td>
</tr>
<tr>
<td>CMA</td>
<td>CMA</td>
<td>Output Limiter (Max)</td>
<td>Per Unit</td>
</tr>
<tr>
<td>CMI</td>
<td>CMI</td>
<td>Output Limiter (Min)</td>
<td>Per Unit</td>
</tr>
<tr>
<td>G1</td>
<td>G1</td>
<td>Saturation Slope</td>
<td></td>
</tr>
</tbody>
</table>

## Output

<table>
<thead>
<tr>
<th>Physical Quantity Name</th>
<th>Figure Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>* B1</td>
<td>-</td>
<td>Intermediate State</td>
<td></td>
</tr>
<tr>
<td>* B2</td>
<td>-</td>
<td>Intermediate State</td>
<td></td>
</tr>
<tr>
<td>ARO</td>
<td>ARO</td>
<td>Real Current</td>
<td>Amps</td>
</tr>
<tr>
<td>A11</td>
<td>A11</td>
<td>Reactive Current</td>
<td>Amps</td>
</tr>
<tr>
<td>AT</td>
<td>AT</td>
<td>Total Current</td>
<td>Amps</td>
</tr>
<tr>
<td>VPF</td>
<td>VPF</td>
<td>Output to Voltage Regulator (See Code)</td>
<td></td>
</tr>
<tr>
<td>PFL</td>
<td>PFL</td>
<td>Line Power Factor</td>
<td>-</td>
</tr>
<tr>
<td>FIN</td>
<td>FIN</td>
<td>Error Input</td>
<td>Per Unit</td>
</tr>
<tr>
<td>FO</td>
<td>FO</td>
<td>Lead Lag Output</td>
<td>Per Unit</td>
</tr>
</tbody>
</table>

* These output quantities are states
**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F(3)</td>
<td></td>
<td>External Force Vector, inertial axes</td>
<td>lbs</td>
</tr>
<tr>
<td>MA</td>
<td></td>
<td>Mass</td>
<td>slugs</td>
</tr>
<tr>
<td>*LA</td>
<td></td>
<td>Initial latitude</td>
<td>deg</td>
</tr>
<tr>
<td>*LO</td>
<td></td>
<td>Initial longitude</td>
<td>deg</td>
</tr>
<tr>
<td>ALT</td>
<td></td>
<td>Initial altitude</td>
<td>feet</td>
</tr>
<tr>
<td>**TI</td>
<td></td>
<td>Initial time</td>
<td>hour</td>
</tr>
<tr>
<td>***DA</td>
<td></td>
<td>Initial date - Julian day</td>
<td>day</td>
</tr>
<tr>
<td>VEL</td>
<td></td>
<td>Initial velocity</td>
<td>ft/sec</td>
</tr>
<tr>
<td>*AZI</td>
<td></td>
<td>Initial horizontal flight path angle (azimuth)</td>
<td>deg</td>
</tr>
<tr>
<td>*GAM</td>
<td></td>
<td>Initial vertical flight path angle</td>
<td>deg</td>
</tr>
</tbody>
</table>

* Default values of zero are provided for these quantities

** Default value of 12 is provided for TI

*** Default value of 80 is provided for DA

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R(3)</td>
<td></td>
<td>Position vector, inertial axes</td>
<td>ft</td>
</tr>
<tr>
<td>RD(3)</td>
<td></td>
<td>Velocity vector, inertial axes</td>
<td>ft/sec</td>
</tr>
</tbody>
</table>

First point of ARIES

557
**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R(3)</td>
<td></td>
<td>Position vector, inertial axes</td>
<td>ft</td>
</tr>
<tr>
<td>RD(3)</td>
<td></td>
<td>Velocity vector, inertial axes</td>
<td>ft/sec</td>
</tr>
<tr>
<td>A(3,3)</td>
<td></td>
<td>Inertial to Body Axis Transformation Matrix</td>
<td></td>
</tr>
<tr>
<td>TI</td>
<td></td>
<td>Initial time</td>
<td>hours</td>
</tr>
<tr>
<td>DA</td>
<td></td>
<td>Initial date - Julian days</td>
<td>days</td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td></td>
<td>Latitude</td>
<td>day</td>
</tr>
<tr>
<td>LO</td>
<td></td>
<td>Longitude</td>
<td>day</td>
</tr>
<tr>
<td>ALT</td>
<td></td>
<td>Altitude</td>
<td>ft</td>
</tr>
<tr>
<td>AZI</td>
<td></td>
<td>Azimuth angle, 0 = North &amp; clockwise</td>
<td>deg</td>
</tr>
<tr>
<td>GAM</td>
<td></td>
<td>Vertical flight path angle, + = pitch up</td>
<td>deg</td>
</tr>
<tr>
<td>EA(3)</td>
<td></td>
<td>Euler angles - Local Horizontal to Body Axes</td>
<td></td>
</tr>
</tbody>
</table>

**EQUATIONS:**

\[
\begin{align*}
L_O &= \tan^{-1} \left( \frac{R(2)}{R(1)} \right) + \frac{360}{365} (80 - D_A) - \frac{360}{12} \left( T_I - 12 - \frac{T_I}{3600} \right) \\
A_L &= |R| - 20927491.6 \\
L_A &= \sin^{-1} \left( \frac{R(3)}{|R|} \right)
\end{align*}
\]
Vehicle Attitude relative to Local Horizontal Transformation from Initial to Body Axes is given:

\[ T_{BI} = T_1(\phi) T_2(\theta) T_3(\psi) T_2(-90 - \phi) T_3(\alpha - \lambda + \lambda) \]

Separate Transformation from Local Horizontal to Body Axes

\[ D_{BH} = T_1(\phi) T_2(\theta) T_3(\psi) = T_{BI} T_3(\lambda - \alpha - \Lambda) T_2(90 + \phi) \]

\[ \theta = \sin^{-1}\left(-d_{13}\right) \]

\[ \phi = \tan^{-1}\left(\frac{d_{23}}{d_{33}}\right) \]

\[ \psi = \tan^{-1}\left(\frac{d_{12}}{d_{11}}\right) \]
Calculation of Flight Path Angle and Euler Angles relating Body Axes to Local Horizontal Axes:

Flight Path Angles

Given:
- \( \mathbf{\hat{R}}_I \) - velocity vector inertial coordinates
- \( \phi \) - latitude
- \( \lambda \) - longitude
- \( \alpha \) - time angle
- \( \lambda \) - date angle

Transform velocity vector into Local Horizontal Axes

\[
\mathbf{R}_H = T_2 (-90 - \phi) T_3 (\alpha - \lambda + \lambda) \mathbf{\hat{R}}_I
\]

\( \mu \) = azimuth - horizontal flight path angle

\( \gamma \) = vertical flight path angle

\[
AZI = \mu = \tan^{-1} \left( \frac{\mathbf{R}_H (2)}{\mathbf{R}_H (1)} \right)
\]

\[
GAM = \gamma = \tan^{-1} \left( \frac{-\mathbf{R}_H (3)}{\left( \mathbf{R}_H^2 (1) + \mathbf{R}_H^2 (2) \right)^{\frac{1}{2}}} \right)
\]
**Noise Generator for Wind Model**

RA

![Diagram](image)

**Output**

<table>
<thead>
<tr>
<th>Physical Quantity Name</th>
<th>Port No.</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>NU, NV, NW</td>
<td>NP</td>
<td>Noise samples for U, V, W gust velocities</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Noise sample for P angular rate gust</td>
<td></td>
</tr>
</tbody>
</table>

**Method:**

Call RN(VAR, DUM, SIG, AMN)

where

- **VAR** = Gaussian random output variable
- **DUM** = Internal variable to start RN
- **SIG** = Standard deviation of VAR = $\sqrt{2.5/\Delta}$; where
  - $\Delta$ = integrator stepsize
- **AMN** = Var mean value = 0

**Note:** RA can only be used with the fixed step integrator which is specified by the command: INT MODE = 3 or 4
RATE GYRO PACKAGE

INPUT

\[ W(3) \]

\[ s^2 + 2 \cdot WnS + Wn^2 \]

\[ SL \]

\[ Wn \]

OUTPUT

\[ W(3) \]

\[ WX(3) \]

EQUATIONS:

\[
FB = W2(I)
\]

\[
IF(\|W2(I)\|>SL), FB = 100 \cdot (W1(I) - \text{SIGN}(SL, W2(I))) + \text{SIGN}(SL, W2(I))
\]

\[
WX(I) = (W1(I) - FB) \cdot WN
\]

\[
W2(I) = (WX(I) - 2 \cdot \text{DMP} \cdot FB) \cdot WN \quad I = 1, 2, 3
\]

Saturation of output state is accomplished by increasing feedback gain by 100 if output exceeds saturation limit.

VECTOR DEFINITIONS:

\[
W1(3) = \begin{pmatrix} P1 \\ Q1 \\ R1 \end{pmatrix} \quad W2(3) = \begin{pmatrix} P2 \\ Q2 \\ R2 \end{pmatrix} \quad WX(3) = \begin{pmatrix} PX \\ QX \\ RX \end{pmatrix}
\]

NOTE: Component XP should be used to convert to and from body axes to gyro axes.

562
RN generates a normally distributed random number each time it is called. The seed, AX, should be an odd number greater than one. This component is automatically disabled for all analyses except SIMULATE.
SATURATION FUNCTION

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td>Slope 0 &lt; S1 &lt; C3</td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td>Slope S1 &gt; C3</td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td>Positive saturation intercept</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td>Slope 0 &gt; S1 &gt; C6</td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td>Slope S1 &lt; C6</td>
<td></td>
</tr>
<tr>
<td>C6</td>
<td></td>
<td>Negative saturation intercept</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>2</td>
<td>Output quantity</td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS:

\[ S2 = C1 \cdot C3 + C2 \cdot (S1 - C3) \text{ if } S1 > C3 \]
\[ S2 = C1 \cdot S1 \text{ if } 0 < S1 < C3 \]
\[ S2 = C4 \cdot S1 \text{ if } 0 > S1 > C6 \]
\[ S2 = C4 \cdot C6 + C5 \cdot (S1 - C6) \text{ if } S1 < C6 \]
SIX-DEGREE-OF-FREEDOM RIGID BODY DYNAMICS

ACCELERATIONS, VELOCITIES, ANGLES & POSITIONS

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UD, VD, WD</td>
<td></td>
<td>X, Y, Z body axis linear accelerations</td>
<td>ft/sec^2</td>
</tr>
<tr>
<td>TX, TY, TZ</td>
<td></td>
<td>X, Y, Z body axis torques</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>IXX, IYY, IZZ</td>
<td></td>
<td>X, Y, Z body axis moments of inertia</td>
<td>slug-ft^2</td>
</tr>
<tr>
<td>IXZ, IXY, IYZ</td>
<td></td>
<td>Cross products of inertia</td>
<td>slug-ft^2</td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>U(3)</td>
<td></td>
<td>X, Y, Z body axis linear velocities</td>
<td>ft/sec</td>
</tr>
<tr>
<td>W(3)</td>
<td></td>
<td>X, Y, Z body axis angular rates</td>
<td>deg/sec</td>
</tr>
<tr>
<td>EA(3)</td>
<td></td>
<td>Euler angles, body to inertial axes</td>
<td>deg</td>
</tr>
<tr>
<td>XD, YD</td>
<td></td>
<td>Horizontal position rates</td>
<td>ft/sec</td>
</tr>
<tr>
<td>*ALT</td>
<td></td>
<td>Vertical altitude from sea-level</td>
<td>ft</td>
</tr>
<tr>
<td>WD(3)</td>
<td>2</td>
<td>X, Y, Z body axis angular accelerations</td>
<td>deg/sec^2</td>
</tr>
</tbody>
</table>

ASSUMPTIONS:

1. Constant gravity, flat-earth model.

*These output quantities are states. X, u, L, p, M, q, y, v, Altitude = -Z
SIX DEGREE OF FREEDOM EQUATIONS OF MOTION

LINEAR VELOCITY EQUATIONS

\[
\begin{align*}
\dot{U} &= U(1) = U(1) \\
\dot{V} &= U(2) = U(2) \\
\dot{W} &= U(3) = U(3)
\end{align*}
\]

ANGULAR VELOCITY EQUATIONS

\[
\begin{align*}
\dot{\theta} &= \psi_{xy} = (\psi_{xy}(Q_{1} + P_{1} + Q_{2}) + \psi_{yz} = (\psi_{yz}(P_{1} + Q_{1} + P_{2}) + \psi_{xz} = (\psi_{xz}(P_{2} + Q_{2} + P_{1})) \\
\dot{\psi} &= \chi_{yz} = (\chi_{yz}(Q_{1} + P_{1} + Q_{2}) + \chi_{xz} = (\chi_{xz}(P_{1} + Q_{1} + P_{2}) + \chi_{xy} = (\chi_{xy}(P_{2} + Q_{2} + P_{1})) \\
\dot{\kappa} &= \psi_{xz} = (\psi_{xz}(Q_{1} + P_{1} + Q_{2}) + \psi_{xy} = (\psi_{xy}(P_{1} + Q_{1} + P_{2}) + \psi_{yz} = (\psi_{yz}(P_{2} + Q_{2} + P_{1}))
\end{align*}
\]

DETERMINANTS

\[
\begin{align*}
\det &= XX_{1}((\psi_{xz}(Q_{1} + P_{1} + Q_{2}) + \psi_{xy} = (\psi_{xy}(P_{1} + Q_{1} + P_{2}) + \psi_{yz} = (\psi_{yz}(P_{2} + Q_{2} + P_{1})) \\
\det &= XX_{2}((\psi_{xy}(Q_{1} + P_{1} + Q_{2}) + \psi_{yz} = (\psi_{yz}(P_{1} + Q_{1} + P_{2}) + \psi_{xz} = (\psi_{xz}(P_{2} + Q_{2} + P_{1})) \\
\det &= XX_{3}((\psi_{xz}(Q_{1} + P_{1} + Q_{2}) + \psi_{xy} = (\psi_{xy}(P_{1} + Q_{1} + P_{2}) + \psi_{yz} = (\psi_{yz}(P_{2} + Q_{2} + P_{1}))
\end{align*}
\]

ANGULAR POSITION EQUATIONS

\[
\begin{align*}
\theta &= \kappa_{xy} = (\kappa_{xy}(Q_{1} + P_{1} + Q_{2}) + \kappa_{yz} = (\kappa_{yz}(P_{1} + Q_{1} + P_{2}) + \kappa_{xz} = (\kappa_{xz}(P_{2} + Q_{2} + P_{1})) \\
\dot{\psi} &= \chi_{yz} = (\chi_{yz}(Q_{1} + P_{1} + Q_{2}) + \chi_{xz} = (\chi_{xz}(P_{1} + Q_{1} + P_{2}) + \chi_{xy} = (\chi_{xy}(P_{2} + Q_{2} + P_{1})) \\
\dot{\kappa} &= \psi_{xz} = (\psi_{xz}(Q_{1} + P_{1} + Q_{2}) + \psi_{xy} = (\psi_{xy}(P_{1} + Q_{1} + P_{2}) + \psi_{yz} = (\psi_{yz}(P_{2} + Q_{2} + P_{1}))
\end{align*}
\]

566
LINEAR POSITION EQUATIONS

\[
XD = CY(CP*U(1)) + (-SY*CR + CY*SPSR)*U(2) + (SY*SR + CY*SPCR)*Y(3)
\]

\[
YD = SY*CP*U(1) + (CY*CR + SY*SPSR)*U(2) + (-CY*SR + SY*SPCR)*U(3)
\]

\[
ZD = SP*U(1) - CP*SR*U(2) - CP*CR*U(3)
\]

The following abbreviations are used in these equations:

\[
SR = \text{SIN}(ROL) \quad CR = \text{COS}(ROL)
\]

\[
SP = \text{SIN}(PIT) \quad CP = \text{COS}(PIT)
\]

\[
SY = \text{SIN}(YAW) \quad CY = \text{COS}(YAW)
\]

\[
SPSR = SP*SR \quad SPCR = SP*CR
\]

VECTOR DEFINITIONS:

\[
UD(3) = \begin{pmatrix}
\dot{U} \\
\dot{V} \\
\dot{W}
\end{pmatrix} \quad U(3) = \begin{pmatrix}
U \\
V \\
W
\end{pmatrix}
\]

\[
WD(3) = \begin{pmatrix}
\dot{P1} \\
\dot{Q1} \\
\dot{R1}
\end{pmatrix} \quad W(3) = \begin{pmatrix}
P1 \\
Q1 \\
R1
\end{pmatrix} \quad EA(3) = \begin{pmatrix}
ROL \\
PIT \\
YAW
\end{pmatrix}
\]
SERVO ANALYZER SIGNAL GENERATOR

Inputs

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR1</td>
<td></td>
<td>Initial, (lower), frequency</td>
<td>hertz</td>
</tr>
<tr>
<td>FR2</td>
<td></td>
<td>Final, (upper), frequency</td>
<td>hertz</td>
</tr>
<tr>
<td>AM1</td>
<td></td>
<td>Initial, (lower), amplitude</td>
<td>-</td>
</tr>
<tr>
<td>AM2</td>
<td></td>
<td>Final, (upper), amplitude</td>
<td>-</td>
</tr>
</tbody>
</table>

Outputs

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td>Test Si Test signal</td>
<td>-</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>Test signal frequency</td>
<td>hertz</td>
</tr>
<tr>
<td>LGF</td>
<td></td>
<td>Log of test signal frequency</td>
<td>-</td>
</tr>
<tr>
<td>AMP</td>
<td></td>
<td>Amplitude of test signal</td>
<td>-</td>
</tr>
</tbody>
</table>

Equations:

\[ S = \text{AMP} \sin (2Ft) \]

Frequency scan occurs if:

\[ \text{FR2} > \text{FR1} \]

Amplitude scan occurs if:

\[ \text{FR2} \leq \text{FR1} \text{ or FR2} = .99999 \]

***WARNING***

This component operates only with fixed step Huen integrator INTMODE = 3. Only one SG Component can appear in a given model.

For frequency scans the following guidelines have been found useful in selecting simulation duration and step size.

\[ T_{MAX} \geq \frac{130}{\text{FR1}} \times \text{(No. decades scanned)} \]

\[ T_{INC} < \frac{1}{30 \times \text{FR2}} \]
SS

SERVO ANALYZER

(This component is used with component SG)

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAMES</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td>Test system output signal</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAMES</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAN</td>
<td></td>
<td>Gain</td>
<td>db.</td>
</tr>
<tr>
<td>PHS</td>
<td></td>
<td>Phase</td>
<td>degrees</td>
</tr>
<tr>
<td>SI *</td>
<td></td>
<td>sine integrator</td>
<td></td>
</tr>
<tr>
<td>CI *</td>
<td></td>
<td>cosine integrator</td>
<td></td>
</tr>
<tr>
<td>SAV</td>
<td></td>
<td>sine (in phase) average value</td>
<td></td>
</tr>
<tr>
<td>CAV</td>
<td></td>
<td>cosine (quad phase) average value</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td></td>
<td>signal used to initialize component</td>
<td></td>
</tr>
</tbody>
</table>

*These quantities are states

Equations:

\[
GAN = 20 \log \left( \frac{2(SAV^2 + CAV^2)^{1/2}}{AMP^2} \right)
\]

\[
PHS = \tan^{-1} \left( \frac{CAV}{SAV} \right)
\]

Several SS components can be used simultaneously with one SS signal generator.
STATISTICAL ANALYSIS

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>1</td>
<td>Input quantity to be monitored</td>
<td></td>
</tr>
<tr>
<td>STR</td>
<td></td>
<td>Parameter to utilize calculations (Default provided)</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MN</td>
<td></td>
<td>Running mean of input quantity</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td></td>
<td>Maximum value of input quantity</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td></td>
<td>Minimum value of input quantity</td>
<td></td>
</tr>
<tr>
<td>SIG</td>
<td></td>
<td>Running standard deviation of input quantity - rms</td>
<td></td>
</tr>
</tbody>
</table>

The measure of mean standard deviation, maximum, and minimum will start at the beginning of each SIMULATE analysis.
SUM TWO SETS OF 3-AXIS FORCES AND TORQUES

\[ F(3) \quad 1 \quad SU \quad 2 \quad F(3) \]

\[ T(3) \quad 3 \quad SU \quad 2 \quad T(3) \]

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F(3)</td>
<td>1</td>
<td>X, Y, Z body axis input forces, port 1</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td>1</td>
<td>X, Y, Z body axis input torques, port 1</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>F(3)</td>
<td>3</td>
<td>X, Y, Z body axis input forces, port 3</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td>3</td>
<td>X, Y, Z body axis input torques, port 3</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F(3)</td>
<td>2</td>
<td>X, Y, Z body axis output forces, port 2</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td>2</td>
<td>X, Y, Z body axis output torques, port 2</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>

**EQUATIONS:**

\[ F2(I) = F1(I) + F3(I) \]
\[ T2(I) = T1(I) + T3(I) \]

\( I = 1, 2, 3 \)
SUM THREE SETS OF 3-AXIS FORCES AND TORQUES

\[ F(3) \rightarrow SV \rightarrow T(3) \]
\[ F(3) \rightarrow SV \rightarrow T(3) \]
\[ F(3) \rightarrow SV \rightarrow T(3) \]

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F(3)</td>
<td>1</td>
<td>X, Y, Z body axis input forces, port 1</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td>1</td>
<td>X, Y, Z body axis input torques, port 1</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>F(3)</td>
<td>3</td>
<td>X, Y, Z body axis input forces, port 3</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td>3</td>
<td>X, Y, Z body axis input torques, port 3</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>F(3)</td>
<td>4</td>
<td>X, Y, Z body axis input forces, port 4</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td>4</td>
<td>X, Y, Z body axis input torques, port 4</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F(3)</td>
<td>2</td>
<td>X, Y, Z body axis output forces, port 2</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td>2</td>
<td>X, Y, Z body axis output torques, port 2</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>

EQUATIONS:

\[ F2(I) = F1(I) + F3(I) + F4(I) \]
\[ T2(I) = T1(I) + T3(I) + T4(I) \quad I = 1, 2, 3 \]
The switching operation may be controlled by either time or the input parameter SW1. The time dependence may be eliminated by setting TC1 = 10^36

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input to switch</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>3</td>
<td>Input to switch</td>
<td></td>
</tr>
<tr>
<td>SW1</td>
<td></td>
<td>Switch control parameter</td>
<td></td>
</tr>
<tr>
<td>TC1</td>
<td></td>
<td>Time for first switching</td>
<td>sec</td>
</tr>
<tr>
<td>TC2</td>
<td></td>
<td>Time for second switching</td>
<td>sec</td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output from switch</td>
</tr>
</tbody>
</table>

**EQUATIONS:**

\[
S2 = S1 \text{ if } SW1 = 1 \text{ and } t < TC1 \text{ or } t > TC2 \text{ or if } SW1 = 0 \text{ and } TC1 < t < TC2
\]

\[
S2 = S3 \text{ if } SW1 = 0 \text{ and } t < TC1 \text{ or } t > TC2 \text{ or if } SW1 = 1 \text{ and } TC1 < t < TC2
\]

where: \( t = \text{TIME, seconds} \)

N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0
TWO POLE SWITCH

NOTE: See SW for switch control logic.

### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input to switch 1</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>3</td>
<td>Input to switch 1</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>5</td>
<td>Input to switch 2</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>6</td>
<td>Input to switch 2</td>
<td></td>
</tr>
<tr>
<td>SW1</td>
<td></td>
<td>Switch control parameter</td>
<td></td>
</tr>
<tr>
<td>TC1</td>
<td></td>
<td>Time for first switching</td>
<td>sec</td>
</tr>
<tr>
<td>TC2</td>
<td></td>
<td>Time for second switching</td>
<td>sec</td>
</tr>
</tbody>
</table>

### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output from switch 1</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>4</td>
<td>Output from switch 2</td>
<td></td>
</tr>
</tbody>
</table>

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0.
NOTE: See SW for switch control logic.

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input to switch 1</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>3</td>
<td>Input to switch 1</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>5</td>
<td>Input to switch 2</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>6</td>
<td>Input to switch 2</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>7</td>
<td>Input to switch 3</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>9</td>
<td>Input to switch 3</td>
<td></td>
</tr>
<tr>
<td>SW1</td>
<td></td>
<td>Switch control parameter</td>
<td></td>
</tr>
<tr>
<td>TC1</td>
<td></td>
<td>Time for first switching</td>
<td>sec</td>
</tr>
<tr>
<td>TC2</td>
<td></td>
<td>Time for second switching</td>
<td>sec</td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output from switch 1</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>4</td>
<td>Output from switch 2</td>
<td></td>
</tr>
<tr>
<td>S(N)</td>
<td>8</td>
<td>Output from switch 3</td>
<td></td>
</tr>
</tbody>
</table>

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0.
FOUR TABULAR FUNCTIONS OF TIME

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2T</td>
<td></td>
<td>Tabular data describing $S_2$ vs. time</td>
<td></td>
</tr>
<tr>
<td>B2T</td>
<td></td>
<td>Tabular data describing $S_3$ vs. time</td>
<td></td>
</tr>
<tr>
<td>C2T</td>
<td></td>
<td>Tabular data describing $S_4$ vs. time</td>
<td></td>
</tr>
<tr>
<td>D2T</td>
<td></td>
<td>Tabular data describing $S_5$ vs. time</td>
<td></td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_2$</td>
<td>2</td>
<td>Output quantity</td>
<td></td>
</tr>
<tr>
<td>$S_3$</td>
<td>3</td>
<td>Output quantity</td>
<td></td>
</tr>
<tr>
<td>$S_4$</td>
<td>4</td>
<td>Output quantity</td>
<td></td>
</tr>
<tr>
<td>$S_5$</td>
<td>5</td>
<td>Output quantity</td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS:

$S_2 = A2T(t)$
$S_3 = B2T(t)$
$S_4 = C2T(t)$
$S_5 = D2T(t)$

NOTE: 15 points are allowed per table. Linear interpolation is used between points. The last point in the table is used for values of time outside the table range.
TWO TABULAR FUNCTIONS OF TIME

\[
S_2 = A_2T(t) \\
S_3 = B_2T(t)
\]

NOTE: 15 points are allowed per table. Linear Interpolation is used between points. The last point in the table is used for values of time outside the table range.
THREE-DEGREE-OF-FREEDOM RIGID BODY

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T(3)</td>
<td></td>
<td>X, Y, Z body axis torques</td>
<td>ft-lb</td>
</tr>
<tr>
<td>IXX, IYY, IZZ</td>
<td></td>
<td>X, Y, Z body axis moments of inertia</td>
<td>slug-ft²</td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>W(3)</td>
<td></td>
<td>X, Y, Z body axis angular rates</td>
<td>rad/sec</td>
</tr>
<tr>
<td>EA(3)</td>
<td></td>
<td>Euler angles, body to fixed axes</td>
<td>rad</td>
</tr>
<tr>
<td>WD(3)</td>
<td></td>
<td>X, Y, Z body axis angular accelerations</td>
<td>rad/sec²</td>
</tr>
</tbody>
</table>

ASSUMPTIONS:
1. Body axes are principal axes, i.e., products of inertia = 0
2. Body moments of inertia are constant
3. Euler angle sequence, body to fixed axes = roll, pitch, yaw.
THREE DEGREE OF FREEDOM RIGID BODY

Angular Velocity Equations
\[
\begin{align*}
\dot{P} &= PD = (TX - Q*R(ZZI - YYI))/XXI \\
\dot{Q} &= ZD = (TY - P*R*(XXI - ZZI))/YYI \\
\dot{R} &= RD = (TZ - Q*P*(YYI - XXI))/ZZI
\end{align*}
\]

Angular Position Equations
\[
\begin{align*}
\dot{\text{PIT}} &= Q*COS(ROL) - R*SIN(ROL) \\
\dot{\text{YAW}} &= (Q*SIN(ROL) + R*COS(ROL))/COS(\text{PIT}) \\
\dot{\text{ROL}} &= P + \dot{\text{YAW}}*SIN(\text{PIT})
\end{align*}
\]

Vector Definitions:
\[
T(3) = \begin{pmatrix} TX \\ TY \\ TZ \end{pmatrix}, \quad W(3) = \begin{pmatrix} P \\ Q \\ R \end{pmatrix}, \quad EA(3) = \begin{pmatrix} \text{ROL} \\ \text{PIT} \\ \text{YAW} \end{pmatrix}, \quad WD(3) = \begin{pmatrix} PD \\ QD \\ RD \end{pmatrix}
\]
### Transfer Function

\[
\frac{Z_1S + Z_0}{S^2 + P_1S + P_0}
\]

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>1</td>
<td>Input quantity</td>
<td></td>
</tr>
<tr>
<td>Z0(N)</td>
<td></td>
<td>Numerator coefficient</td>
<td></td>
</tr>
<tr>
<td>Z1(N)</td>
<td></td>
<td>Numerator coefficient</td>
<td></td>
</tr>
<tr>
<td>P0(N)</td>
<td></td>
<td>Denominator coefficient</td>
<td></td>
</tr>
<tr>
<td>P1(N)</td>
<td></td>
<td>Denominator coefficient</td>
<td></td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(N)</td>
<td>2</td>
<td>Output quantity (state)</td>
<td></td>
</tr>
<tr>
<td>X1(N)</td>
<td></td>
<td>Intermediate state (state)</td>
<td></td>
</tr>
</tbody>
</table>

**EQUATIONS:**

\[
\begin{align*}
\dot{X}_1 &= Z_0S - P_0S \\
\dot{S}_2 &= X_1 + Z_1S - P_1S
\end{align*}
\]

**NOTE:**
- d.c. gain = \( \frac{Z_0}{P_0} \)
- Infinite frequency gain = 0

**NOTE:** N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0.
ENGINE THRUST BODY AXIS TRANSFORM

\[
\text{TG}
\]

\[
\text{TH} \rightarrow \text{BODY AXIS TRANSFORMATION} \rightarrow \text{F(3), T(3)}
\]

\[
\text{GAM(3)} \rightarrow \text{X(3)}
\]

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH</td>
<td></td>
<td>Engine thrust</td>
<td>lbs</td>
</tr>
<tr>
<td>GAM(3) X(3)</td>
<td></td>
<td>X, Y, Z body axis direction cosines</td>
<td>lbs</td>
</tr>
<tr>
<td>X(3)</td>
<td></td>
<td>X, Y, Z thrust location components</td>
<td>ft</td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>F(3)</td>
<td></td>
<td>X, Y, Z body axis forces</td>
<td>lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td></td>
<td>X, Y, Z body axis torques</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>

EQUATIONS:

\[ F(I) = TH \times \text{GAM}(I) \]
\[ T = \bar{X} \times \bar{F} \text{ (vector cross product)} \]
\[ I = 1, 2, 3 \]
TACHOMETER RIPPLE EFFECTS

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td></td>
<td>Shaft angle</td>
<td>rad</td>
</tr>
<tr>
<td>ALO</td>
<td></td>
<td>Shaft rate</td>
<td>rad/sec</td>
</tr>
<tr>
<td>GTA</td>
<td></td>
<td>Tachometer gain</td>
<td>volt/rad/sec</td>
</tr>
<tr>
<td>CH</td>
<td></td>
<td>Hall probe null coefficient</td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td></td>
<td>Common node coefficient</td>
<td></td>
</tr>
<tr>
<td>CHG</td>
<td></td>
<td>Unequal gain coefficient</td>
<td></td>
</tr>
<tr>
<td>CMF</td>
<td></td>
<td>Magnetic field coefficient</td>
<td></td>
</tr>
<tr>
<td>WH</td>
<td></td>
<td>Hall probe frequency</td>
<td>rad/sec</td>
</tr>
<tr>
<td>WMF</td>
<td></td>
<td>Magnetic field frequency</td>
<td>rad/sec</td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VO</td>
<td></td>
<td>Tachometer output voltage</td>
<td>volt</td>
</tr>
</tbody>
</table>

EQUATIONS:

\[
WHAL = WH \cdot AL \\
WHPAL = WHP \cdot AL \\
VO = GTA \cdot ALO \cdot (1 + CH \cdot \sin(WHAL) + CN \cdot \cos(WHAL) \\
+ CHG \cdot \sin(WHPAL + CHG \cdot \cos(WHPAL) + CMF \cdot \sin(WMF \cdot AL))
\]
SUM TWO SETS OF 3-AXIS TORQUES

INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T(3)</td>
<td>1</td>
<td>X, Y, Z body axis input torques, port 1</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>T(3)</td>
<td>3</td>
<td>X, Y, Z body axis input torques, port 3</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>

OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T(3)</td>
<td>2</td>
<td>X, Y, Z body axis output torques, port 2</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>

EQUATIONS:

\[ T_2(I) = T_1(I) + T_3(I) \quad I = 1, 2, 3 \]
SUM THREE SETS OF 3-AXIS TORQUES

DESCRIPTION: Same as TS, except with one additional port.
IUS Vehicle with 6 Degrees of Freedom, Fuel Sloshing, Structural Flexibility, and Tail-wag-dog Engine Dynamics.
(This component must be used with component UT to form complete vehicle model.)

## INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMP(M,M)</td>
<td></td>
<td>Damping matrix</td>
<td></td>
</tr>
<tr>
<td>MSI(M,M)</td>
<td></td>
<td>Inverse Mass Matrix</td>
<td>Formed by component UT</td>
</tr>
<tr>
<td>WRK(M)</td>
<td></td>
<td>Work vector</td>
<td></td>
</tr>
<tr>
<td>THR(3)</td>
<td></td>
<td>Engine thrust vector in body coordinates</td>
<td>lb</td>
</tr>
<tr>
<td>LMN(3)</td>
<td></td>
<td>Spacecraft torque vector due to engine thrust</td>
<td>in-lb</td>
</tr>
<tr>
<td>DLM(2)</td>
<td></td>
<td>Moment exerted by actuator on engine nozzle about yaw and pitch axes</td>
<td>in-lb</td>
</tr>
<tr>
<td>GNF(N)</td>
<td></td>
<td>Generalized forces due to thrust exerted on flexing modes</td>
<td></td>
</tr>
</tbody>
</table>

## OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UVW(3)*</td>
<td></td>
<td>Rigid body translational velocity vector</td>
<td></td>
</tr>
<tr>
<td>PQR(3)*</td>
<td></td>
<td>Rigid body rotational velocity vector</td>
<td></td>
</tr>
<tr>
<td>SD1(2)*</td>
<td></td>
<td>Slosh dynamics velocity vector (1st tank)</td>
<td></td>
</tr>
<tr>
<td>SD2(2)*</td>
<td></td>
<td>Slosh dynamics velocity vector (2nd tank)</td>
<td></td>
</tr>
</tbody>
</table>

585
<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLD(2)*</td>
<td></td>
<td>Nozzle attitude velocity vector</td>
<td></td>
</tr>
<tr>
<td>FXD(N)*</td>
<td></td>
<td>Body flex modes velocity vector</td>
<td></td>
</tr>
<tr>
<td>SL1(2)*</td>
<td></td>
<td>Fuel slosh position vector (1st tank)</td>
<td></td>
</tr>
<tr>
<td>SL2(2)*</td>
<td></td>
<td>Fuel slosh position vector (2nd tank)</td>
<td></td>
</tr>
<tr>
<td>DLT(2)*</td>
<td></td>
<td>Nozzle attitude vector</td>
<td></td>
</tr>
<tr>
<td>FLX(N)*</td>
<td></td>
<td>Body flex mode position vector</td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS:
\[ \ddot{\mathbf{x}} = \mathbf{MSI} \left[ -\mathbf{DMP} \mathbf{\dot{x}} - \mathbf{STF} \mathbf{x} + \mathbf{f} \right] \]

MSI, DMP, and STF are M x M Matrices formed by standard component UT.
* These quantities are continuous states.
N must be specified as the number of structural flexibility modes
M must be specified as 12 + N
The above represent the spacecraft state vector and the vector of forces due to engine thrust and nozzle actuator, respectively.

This formulation follows Boeing Document D2-84124-4 (pg 70) except that some of the components of $X$ have been permuted for programming convenience.
IUS Vehicle with 6 Degrees of Freedom, Fuel Sloshing, Structural Flexibility, and Tail-wag-dog Engine Dynamics. (This component must be used with component US to form complete Vehicle Model.)

### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PQR(3)</td>
<td></td>
<td>Rigid body rotational velocity vector</td>
<td>rad/sec</td>
</tr>
<tr>
<td>MS1</td>
<td></td>
<td>Sloshing masses for tanks 1 and 2</td>
<td>lb-sec²/in</td>
</tr>
<tr>
<td>MS2</td>
<td></td>
<td>Sloshing pendulum arm lengths</td>
<td>inch</td>
</tr>
<tr>
<td>LS1</td>
<td></td>
<td>Nominal position of sloshing tanks 1 and 2 in body coordinates</td>
<td></td>
</tr>
<tr>
<td>LS2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP1(3)</td>
<td></td>
<td>Mass of engine nozzle</td>
<td>inch</td>
</tr>
<tr>
<td>LE</td>
<td></td>
<td>Distance from hinge point to nozzle center of gravity</td>
<td>inch</td>
</tr>
<tr>
<td>EP(3)</td>
<td></td>
<td>Position of nozzle center of gravity in body coordinates when nozzle is in undeflated position</td>
<td>inch</td>
</tr>
<tr>
<td>MSS</td>
<td></td>
<td>Mass of entire spacecraft</td>
<td>lb-sec²/in</td>
</tr>
<tr>
<td>IXX</td>
<td></td>
<td>Spacecraft moments of inertia</td>
<td>in-lb-sec²</td>
</tr>
<tr>
<td>IYY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IZZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IYE</td>
<td></td>
<td>Nozzle moments of inertia about nozzle center of gravity</td>
<td>in-lb-sec²</td>
</tr>
<tr>
<td>IZE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHYSICAL QUANTITY NAME</td>
<td>PORT NO.</td>
<td>DESCRIPTION</td>
<td>UNITS</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>IXY</td>
<td></td>
<td>Spacecraft products of inertia</td>
<td>in-lb-sec²</td>
</tr>
<tr>
<td>IXZ</td>
<td></td>
<td>Spacecraft products of inertia</td>
<td>in-lb-sec²</td>
</tr>
<tr>
<td>IYZ</td>
<td></td>
<td>Spacecraft products of inertia</td>
<td>in-lb-sec²</td>
</tr>
<tr>
<td>MN(N)</td>
<td></td>
<td>Flex deflection coefficients</td>
<td></td>
</tr>
<tr>
<td>PS1(2,N)</td>
<td></td>
<td>Flex deflection coefficients at tanks 1 and 2</td>
<td></td>
</tr>
<tr>
<td>PS2(2,N)</td>
<td></td>
<td>Flex deflection coefficients at nozzle</td>
<td></td>
</tr>
<tr>
<td>PE(2,N)</td>
<td></td>
<td>Flex deflection coefficients at nozzle</td>
<td></td>
</tr>
<tr>
<td>PEP(2,N)</td>
<td></td>
<td>Flex rotation coefficients at nozzle</td>
<td></td>
</tr>
<tr>
<td>WP1</td>
<td></td>
<td>Natural frequencies at sloshing modes</td>
<td>rad/sec</td>
</tr>
<tr>
<td>WT1</td>
<td></td>
<td>Natural frequencies at sloshing modes</td>
<td>rad/sec</td>
</tr>
<tr>
<td>WP2</td>
<td></td>
<td>about yaw and pitch axes</td>
<td>rad/sec</td>
</tr>
<tr>
<td>WT2</td>
<td></td>
<td>about yaw and pitch axes</td>
<td>rad/sec</td>
</tr>
<tr>
<td>WFX(N)</td>
<td></td>
<td>Natural frequencies of flex modes 1, ..., n</td>
<td>rad/sec</td>
</tr>
<tr>
<td>WEP</td>
<td></td>
<td>Natural frequencies of nozzle</td>
<td>rad/sec</td>
</tr>
<tr>
<td>WET</td>
<td></td>
<td>Natural frequencies of nozzle</td>
<td>rad/sec</td>
</tr>
<tr>
<td>ZS1</td>
<td></td>
<td>Damping ratios of sloshing modes</td>
<td></td>
</tr>
<tr>
<td>ZS2</td>
<td></td>
<td>Damping ratios of sloshing modes</td>
<td></td>
</tr>
<tr>
<td>ZFX(N)</td>
<td></td>
<td>Damping ratios of flexing modes</td>
<td></td>
</tr>
<tr>
<td>ZEP</td>
<td></td>
<td>Damping ratios of flexing modes</td>
<td></td>
</tr>
<tr>
<td>ZET</td>
<td></td>
<td>Linear damping ratio for nozzle</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>about yaw and pitch axes</td>
<td></td>
</tr>
</tbody>
</table>
OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMP(M,M)</td>
<td></td>
<td>Damping matrix</td>
<td></td>
</tr>
<tr>
<td>MSI(M,M)</td>
<td></td>
<td>Inverse mass matrix</td>
<td></td>
</tr>
<tr>
<td>STF(M,M)</td>
<td></td>
<td>Stiffness matrix</td>
<td></td>
</tr>
</tbody>
</table>

EQUATIONS:

See document D2-84124-4, page 70.

N must be specified as the number of structural flexibility modes
M must be specified as 12 + N
### VEHICLE ATTITUDE

**VA**

\[ T(3) \rightarrow VA \rightarrow W(3), Q(4) \]

\[ IN(3,3) \]

\[ A(3,3), EA(3) \]

#### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>*LA</td>
<td>1</td>
<td>Initial latitude</td>
<td>deg</td>
</tr>
<tr>
<td>*LO</td>
<td>1</td>
<td>Initial longitude</td>
<td>deg</td>
</tr>
<tr>
<td>**TI</td>
<td>1</td>
<td>Initial time</td>
<td>hours</td>
</tr>
<tr>
<td>***DA</td>
<td>1</td>
<td>Initial date - Julian days</td>
<td>days</td>
</tr>
<tr>
<td>*ROL</td>
<td></td>
<td>Initial roll - relative to local horizontal axes</td>
<td>deg</td>
</tr>
<tr>
<td>*PIT</td>
<td></td>
<td>Initial pitch - relative to local horizontal axes</td>
<td>deg</td>
</tr>
<tr>
<td>*YAW</td>
<td></td>
<td>Initial yaw - relative to local horizontal axes</td>
<td>deg</td>
</tr>
<tr>
<td>T(3)</td>
<td></td>
<td>External torques, body axes</td>
<td>ft-lb</td>
</tr>
<tr>
<td>IN(3,3)</td>
<td></td>
<td>Inertia matrix, body axes</td>
<td>slug-ft²</td>
</tr>
</tbody>
</table>

* Default values of zero are provided for these quantities

** Default value of 12 is provided for TI

*** Default value of 80 is provided for DA

#### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>W(3)</td>
<td></td>
<td>Angular rates - body axes</td>
<td>deg/sec</td>
</tr>
<tr>
<td>Q(4)</td>
<td></td>
<td>Quaternians - inertial to body axes</td>
<td></td>
</tr>
<tr>
<td>A(3,3)</td>
<td></td>
<td>Direction cosine matrix - inertial to body axes</td>
<td></td>
</tr>
<tr>
<td>EA(3)</td>
<td></td>
<td>Euler angle - inertial to body axes</td>
<td>deg</td>
</tr>
<tr>
<td>LA</td>
<td>2</td>
<td>Initial latitude</td>
<td>deg</td>
</tr>
<tr>
<td>LO</td>
<td>2</td>
<td>Initial longitude</td>
<td>deg</td>
</tr>
<tr>
<td>TI</td>
<td>2</td>
<td>Initial time</td>
<td>hours</td>
</tr>
<tr>
<td>DA</td>
<td>2</td>
<td>Initial date - Julian days</td>
<td>days</td>
</tr>
</tbody>
</table>

591
## VOLTAGE REGULATOR

### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY</th>
<th>FIGURE</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPF</td>
<td>VPF</td>
<td>INPUT FROM POWER FACTOR CONTROLLER</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>VL</td>
<td>VL</td>
<td>LINE VOLTAGE</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>ED</td>
<td>ED</td>
<td>D AXIS VOLTAGE FROM GEN</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>EQ</td>
<td>EQ</td>
<td>Q AXIS VOLTAGE FROM GEN</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>VRE</td>
<td>VRE</td>
<td>VOLTAGE REFERENCE</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>G1</td>
<td>G1</td>
<td>LAG GAIN</td>
<td>-</td>
</tr>
<tr>
<td>G2</td>
<td>G2</td>
<td>LEAD LAG GAIN (FEEDBACK)</td>
<td>-</td>
</tr>
<tr>
<td>G3</td>
<td>G3</td>
<td>FEEDBACK GAIN</td>
<td>-</td>
</tr>
<tr>
<td>T1</td>
<td>T1</td>
<td>LAG TIME CONSTANT</td>
<td>-</td>
</tr>
<tr>
<td>T2</td>
<td>T2</td>
<td>LEAD LAG TIME CONSTANT (FEEDBACK)</td>
<td>SEC</td>
</tr>
<tr>
<td>T3</td>
<td>T3</td>
<td>LEAD LAG TIME CONSTANT (FEEDBACK)</td>
<td>SEC</td>
</tr>
<tr>
<td>T4</td>
<td>T4</td>
<td>LAG TIME CONSTANT</td>
<td>SEC</td>
</tr>
<tr>
<td>CEX</td>
<td>CEX</td>
<td>LIMITER MAX</td>
<td>SEC</td>
</tr>
<tr>
<td>EB</td>
<td>EB</td>
<td>LAG GAIN (PER UNIT CONVERSION)</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>G3</td>
<td>G3</td>
<td>SATURATION SLOPE</td>
<td>-</td>
</tr>
</tbody>
</table>

### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>FIGURE NAME</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2</td>
<td>E2</td>
<td>INTERNAL STATE LAG OUTPUT</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>E4</td>
<td>E4</td>
<td>INTERNAL STATE</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>E5</td>
<td>E5</td>
<td>INTERMEDIATE STATE</td>
<td></td>
</tr>
<tr>
<td>v0</td>
<td>v0</td>
<td>OUTPUT TO GEN/EXCITER</td>
<td>VOLTS</td>
</tr>
<tr>
<td>EL</td>
<td>EL</td>
<td>RSS OF EQ AND ED</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>E1</td>
<td>E1</td>
<td>ERROR SUM</td>
<td>PER UNIT</td>
</tr>
<tr>
<td>E3</td>
<td>E3</td>
<td>LIMITER OUTPUT</td>
<td>PER UNIT</td>
</tr>
</tbody>
</table>

* THESE OUTPUT QUANTITIES ARE STATES
### RANDOM WIND GUST MODEL

**WM**

![Diagram showing the flow of variables](image)

**Transfer Functions:**
- SLH, SLV, VS, SIH, SIV, B

#### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NU, NV, NW, NP</td>
<td>1</td>
<td>Random noise inputs for UW, VW, WW</td>
<td>ft</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Random noise input for PW angular rate</td>
<td></td>
</tr>
<tr>
<td>SLH, SLV, VS</td>
<td>1</td>
<td>Horizontal and vertical scales*</td>
<td>ft</td>
</tr>
<tr>
<td>SIH, SIV, B</td>
<td>1</td>
<td>Steady state airspeed input</td>
<td>ft/sec</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Horizontal and vertical RMS gust intensity*</td>
<td>ft/sec</td>
</tr>
</tbody>
</table>

#### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UW, VW, WW</td>
<td>2</td>
<td>X, Y, Z body axis wind velocity states</td>
<td>ft/sec</td>
</tr>
<tr>
<td>VX, WX</td>
<td>2</td>
<td>Y, Z axis intermediate states</td>
<td>ft/sec</td>
</tr>
<tr>
<td>QX, RX</td>
<td>2</td>
<td>Y, Z body axis wind angular rate states</td>
<td>deg/sec</td>
</tr>
<tr>
<td>PW, QW, RW, VS</td>
<td>2</td>
<td>X, Y, Z body axis wind angular rate outputs</td>
<td>deg/sec</td>
</tr>
<tr>
<td>VS</td>
<td>2</td>
<td>Steady state airspeed</td>
<td>ft/sec</td>
</tr>
</tbody>
</table>

*Default values:
- SLH = SLV = 1750
- SIH = SIV = 0

In general, choose SIH and SIV such that \[
\frac{(SIH)^2}{SLH} = \frac{(SIV)^2}{SLV}
\]
WIND MODEL TRANSFER EQUATIONS

**UW:**
\[ \frac{G_v}{1 + L_H^{'}} \]

- \[ L_H^{' \prime} = \frac{SLH}{VS} \]
- \[ G_u = \sin(2 + L_H^{'}/\pi)^{1/2} \]
- \[ U_W = \frac{(G_v \times NU - UW)/L_H^{'}}{\frac{G_v}{1 + L_H^{'}}} \]

**VW:**
\[ \frac{G_v(1 + 3 + L_H^{'})}{(1 + L_H^{'})^{2}} \]

- \[ G_v = \sin((L_H^{'}/\pi)^{1/2}) = G_u/\sqrt{2} \]
- \[ V_X = \frac{(G_v \times NV - VW)/(L_H^{'})^{2}}{1} \]
- \[ V_W = V_X + \frac{\sqrt{3} \times G_v \times NV - 2 \times VW)/L_H^{'}}{1} \]

**WW:**
\[ \frac{G_w(1 + \sqrt{3} + L_H^{'})}{(1 + L_H^{'})^{2}} \]

- \[ L_V = \frac{SLV}{VS} \]
- \[ G_w = \sin((L_V/\pi)^{1/2}) \]
- \[ W_X = \frac{(G_w \times NW - WW)/(L_V^{'})^{2}}{1} \]
- \[ W_W = W_X + \frac{\sqrt{3} \times G_w \times NW - 2 \times WW)/L_V^{'}}{1} \]

**PW:**
\[ \frac{G_p(180/\pi)}{1 + C_H \times S} \]

- \[ C_H = \frac{4 \times B/(\pi \times VS)}{1} \]
- \[ G_p = 1/\sin((0.8 \times (\pi \times SLV)/(4 \times B))^{1/3} / (SLV \times VS))^{1/2} \]
- \[ P_W = ((G_P \times NP - PW/C_H) \times 180/\pi) \]

**RW:**
\[ \frac{-S(180/\pi)}{VS(1 + C_V \times S)} \]

- \[ C_V = 3 \times B/(\pi \times VS) = 0.75 \times C_H \]
- \[ RW = RX - 180/\pi \times VW/(VS \times C_V) \]
- \[ RX = -RW/C_V \]

**QW:**
\[ \frac{S(180/\pi)}{VS(1 + C_H \times S)} \]

- \[ Q_W = Q_X + 180/\pi \times WW/(VS \times C_H) \]
- \[ Q_X = -Q_W/C_H \]

594
**STATIC TRANSFORMATION OF ANGULAR RATES**

![Diagram showing static transformation of angular rates XP](image)

**INPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>W(3)</td>
<td>1</td>
<td>Input angular rates - initial coordinates</td>
<td>rad/sec</td>
</tr>
<tr>
<td>TRN(3,3)</td>
<td></td>
<td>3 x 3 transformation matrix</td>
<td></td>
</tr>
</tbody>
</table>

**OUTPUT**

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>W(3)</td>
<td>2</td>
<td>Output angular rates - final coordinates</td>
<td>rad/sec</td>
</tr>
</tbody>
</table>

**EQUATIONS:**

\[ W_2 = \text{TRN} \cdot W_1 \text{ (Matrix Multiply)} \]

**ASSUMPTIONS:**

TRN contains the direction cosines required to transform from the initial coordinate system. TRN is input as follows:

- TRNXP = TRNXP
- R(1,1) \( a_{11} \), \( a_{12} \), \( a_{13} \)
- R(2,1) \( a_{21} \), \( a_{22} \), \( a_{23} \)
- R(3,1) \( a_{31} \), \( a_{32} \), \( a_{33} \)

595
## Static Transformation of Torques

### INPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T(3)</td>
<td>1</td>
<td>Input torques - initial coordinates</td>
<td>ft-lbs</td>
</tr>
<tr>
<td>TRN</td>
<td></td>
<td>3 x 3 transformation matrix</td>
<td></td>
</tr>
</tbody>
</table>

### OUTPUT

<table>
<thead>
<tr>
<th>PHYSICAL QUANTITY NAME</th>
<th>PORT NO.</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T(3)</td>
<td>2</td>
<td>Output torques - final coordinates</td>
<td>ft-lbs</td>
</tr>
</tbody>
</table>
This appendix is a reproduction of Section 4.4 of reference 1. It presents a description of the mathematical methods used in each of the analyses available in the EASY Analysis Program.
4.3.2 Scalar Data

Scalar data, i.e., parameter values, error controls and initial conditions, should be loaded by data cards immediately following any tabular data cards. All of these scalar values should be specified before any analysis is requested. However, to prevent the loss of an analysis run due to the omission of one or more parameter values, error controls, or initial conditions, all parameter values are initialized to a default value of 0.99999, all error controls to 0.1*, and all initial conditions to 0. Sections 4.2.2 and 4.2.3 describe the program commands and formats used to specify scalar data. Once the parameter values, error controls, and initial conditions have been specified, other program commands may be issued to request analyses. The values of any of the scalar data can be modified between analysis requests by using the same commands described in Sections 4.2.2 and 4.2.3.

4.4 ANALYSIS DESCRIPTIONS

This section contains a description of the mathematical methods used in each of the analyses available in the EASY Analysis Program. Further details of each analysis can be found in the Section 6.

4.4.1 Simulation Calculations

One of the most used and well known numerical integration rules is the classical explicit fourth order Runge-Kutta method (Reference 1). The method is easy to implement, has nice truncation error properties, and combined with an error control (step size adjustment) is a good standard integration method for systems with eigenvalues (of the Jacobian) all relatively the same size. For this reason, the 4th order Runge-

* See Section 4.5.3 for special default values provided by the EASY Model Generation program for states whose name starts with the letters P or T.
Kutta method is included as one integrator available in EASY. It is not the default integrator, however, because of its stability properties. A short discussion of integration rule stability follows.

For most integration algorithms, truncation error (the error incurred due to a finite order approximation to the exact solution) is directly related to the step size raised to a power equal to the order of the method. By controlling the step size, the single step error can theoretically be maintained at any desired level. This assumes that sufficient precision is used so that round off effects (error due to approximating numbers by a finite number of bits or digits) does not become a factor. Most integration algorithms thus contain some error measurement calculation and a step size adjuster so that single step error is below a specified limit.

The question now arises of what happens to such systems when the actual value of the truncation error becomes very small due to the actual solution approaching a steady or slowly varying value. The normal logic in most algorithms indicates that the step size should be increased. As the step size is increased a phenomenon related to integration rule stability occurs. That is, even though the solution and the resultant error are well below the specified error limit, increasing the step size will eventually cause errors to increase over the limit. This is due to the fact that every integration rule has a region of stability (a function of step size) where given a stable (non-increasing) system it will compute a non-increasing solution. Outside of that region, even though the solution should decrease, it will compute an increasing solution. This region is normally described as a function of the time step h times the complex eigenvalues of the system. Thus if one were to plot the region in a complex plane modified by the step size, the 4th order Runge-Kutta would have a region that appears as shown in Figure 42.

This means that if any stable mode (represented by a eigenvalue \( \lambda_i \)) is large enough that \( h \lambda_i \) lies outside the shaded area, then for that
Figure 42. Region of Absolute Stability of Fourth Order Explicit Runge-Kutta Method

Figure 43. Regions of Absolute Stability for Stiffly Stable Methods of Orders One Through Three. Methods Are Stable Outside of Closed Contours.

Figure 44. Regions of Absolute Stability for Stiffly Stable Methods of Orders Four Through Six
mode and step size $h$, an increasing solution will be computed even though the actual solution is decreasing. For this reason, even when truncation error is reduced to a very small value and the solution mode is in steady state, the step size for the Runge-Kutta method is limited to approximately

$$h_{\text{max}} \leq \frac{2.7}{|\lambda_{\text{max}}|}$$

in order to prevent the computed solution from diverging.

For systems that have wide ranges of eigenvalues, this limitation can cause unreasonably long computation times. Thus, one seeks integration rules which have more desirable stability properties.

The integration rule implemented in EASY is a "stiffly stable" method developed by Gear and published both in his book and in the Communications of the Association for Computing Machinery, Vol. 14, No. 3, March 1971. This is a variable step size, variable order method which has regions of stability outside of the contours in Figures 43 and 44. For these regions, it is noted that large magnitude eigenvalues with negative real parts that are large fall well inside the region of stability. Thus as truncation error becomes small during the integration process, the method is not restricted from using large step sizes.

Note that part of the right hand plane is stable even though the actual system would be unstable. All this means is that if the step size were unchanged, the integrator would output a decreasing solution. Error control, however, would detect the difference and decrease the step size until a correct solution to the specified accuracy was obtained.

Since the algorithm is well documented in Gear's book in Chapter 9, the theoretical exposition is not repeated here. The modifications made to the data structure so that storage is by column and not by row. Theoretically this is of no importance but practically it is better due to the manner in which FORTRAN stores and computes indices in arrays.
Only the stiff integrator that computes partial derivatives by numerical differencing is retained. The process of solving a linear system of equations by matrix inversion is replaced by the more efficient and accurate direct Gaussian elimination method. The method order is restricted to 5 or less because of stability considerations.

The process of integration is controlled by a master subroutine which keeps track of time and the necessary reporting sequence. Further, this routine recognizes when a new call is made to the integrator for the first time and uses a special start up procedure. This procedure essentially uses the standard 4th order Runge-Kutta for 100 steps (picked by the step size controller) to let initial transients settle out before handing the problem over to Gear's method. Since the Gear method must start out with a 1st order integration rule, large initial transients can cause problems. Thus using another 4th order rule to integrate over small intervals of large transient behavior allows the Gear method to start in a smoother region of the solution. This external integration process will occur whenever large transients cause the Gear method to fail.

Minimum step size is set at $10^{-5}$ seconds or TINC/10000, whichever is the smaller value. The maximum step size is set equal to the print interval and is often attained. The error test used is based on relative error with respect to the maximum value computed for a particular variable. The current value is set at 5 significant figures maintained over a single step.

At the start of each simulation run, the time variable is set equal to zero; the state vector of the system model is set equal to the initial condition vector, (values input via the INITIAL CONDITION command); and the state variable time derivatives (rates), are set equal to zero. The rates are set equal to zero as part of the procedure that allows individual states to be frozen.
For frozen states, the rates are not recalculated by the system model. Thus, since the rates are set to zero these states remain "frozen" at their initial values.

Integration of the system model equations continues until the value of time equals the value of TMAX specified by the analyst. If it is desired to have a simulation stop for some condition, before time reaches TMAX, a test on this condition can be added to the system model, (in subroutine EQMO), and TIME set equal to TMAX should this condition occur. An example of this sort was shown in Example 3.3.

4.4.2 Steady State Calculations

The STEADY STATE option allows the steady state of a stable system dynamic model to be quickly determined. This is accomplished by modifying the dynamic characteristics of the system so that all eigenvalues are near -1. This allows the system transient to be quickly integrated to reach steady state.

The nonlinear simulation model can be defined as:

\[ \dot{x} = f(x, t) \]  \hspace{1cm} (4.4-1)

where:
- \( \dot{x} \) = n dimensional vector of state variable derivatives
- \( x \) = n dimensional vector of state variables
- \( f \) = n dimensional vector of nonlinear functions relating state variables and time to state variable derivatives.

The steady state of this system is defined as that value, \( x_{ss} \), of the system state vector, \( x \), that causes \( \dot{x} \) to equal zero. Thus:

\[ 0 = f(x_{ss}, t) \]  \hspace{1cm} (4.4-2)

Let a linear approximation for the nonlinear system, as described in Section 4.5.3, be given by:

\[ \dot{x} = Ax \]  \hspace{1cm} (4.4-3)
Where $A = nxn$ stability matrix (Jacobian) of the system model.

The major objection to integrating the given nonlinear system of (4.4-1) to obtain the steady state is that many small integration steps are required over a long transient duration to reach steady state. As discussed in Section 4.4.1 this problem is related to a large range of eigenvalue magnitudes of the system stability matrix, $A$. If the objective is to rapidly reach steady state, the ideal dynamic system would have all of its eigenvalues concentrated in a very small range. This can be accomplished, if one is not interested in the accuracy of the transient calculation, for a stable system with a negative definite $A$ by premultiplying the system matrix by $-A^{-1}$. The modified state will be designated by $x'$.

\[
\dot{x}' = -A^{-1} A x' \\
= -I x'
\]

4.4-4
4.4-5

The modified system of equation (4.4-5) has the desired feature that all of its eigenvalues are in a small range, i.e., all equal minus one. Thus, by pre-multiplying the given system function by $-A^{-1}$, we may obtain a modified system with all eigenvalues near -1. Applying this modification to equation 4.4-1 we obtain

\[
\dot{x}' = -A^{-1} f (x', t) 
\]

4.4-6

Since the transformation $A^{-1}$ is nonsingular, the only solution to the modified steady state equation

\[
0 = -A^{-1} f (x_{ss}, t) 
\]

4.4-7

is that shown in equation (4.4-2). Thus the system of equations given in (4.4-6) has the same steady state solution as the original system, (4.4-1) but has an eigenvalue range that greatly reduces the number of integration steps required to reach steady state. This approach to solving for the steady state may also be viewed as a multi-dimensional
version of Newton's Method for solving the nonlinear algebraic equation of (4.4-2). The numerical method proceeds as follows:

The system rates and stability matrix are evaluated at the initial state, $x_i$.

\[ \dot{x}_i = \frac{\partial f(x_i, t)}{\partial x} \quad 4.4-8 \]

\[ A_i = \frac{\partial f(x_i, t)}{\partial x} \bigg|_{x=x_i} \quad 4.4-9 \]

Rather than premultiply by the inverse matrix, as indicated in (4.4-6), the equation

\[ -A_i \dot{x}_i = f(x_i, t) \quad 4.4-10 \]

is solved for $\dot{x}_i$, given $A_i$ and $f(x_i, t)$ by the Gaussian elimination method.

The Euler forward difference approximation, for a time difference of 1, is then used to represent $\dot{x}_i$

\[ \dot{x}_i = x_{i+1} - x_i \quad 4.4-11 \]

Solving for $x_{i+1}$ we obtain

\[ x_{i+1} = x_i + \dot{x}_i \quad 4.4-12 \]

The process of solving equations (4.4-8) through (4.4-12) is repeated until the norm of the residual vector, $\dot{x}$, becomes less than $10^{-4}$ or more than SS ITERATIONS occur. As implemented, the system stability matrix $A$ is not completely recalculated each iteration and a step size less than 1 second is used if the method encounters difficulty in converging.
Should this method fail to reach a steady state from a given initial condition, the less efficient, but more stable simulation approach can be used. Of course, for some nonlinear systems a steady state can not be reached from certain regions of the state space, (initial conditions). In these cases, it will be necessary to vary the initial conditions to find a steady state by either the STEADY STATE, or the SIMULATE commands.

At the final state reached by the steady state analysis, a linear model of the system is generated and its eigenvalues are calculated and printed. These should be examined to assure that there are no non-negative real parts which would indicate an unstable system. It is usually of interest to know the eigenvalues of the system at each steady state operating point. Also, in rare cases, the steady state method can converge to an unstable equilibrium point such as point X2 in Figure 45.

4.4.3 Linear Analysis Calculations

**Stability Matrix Calculation**

The LINEAR ANALYSIS option allows linear approximations to the nonlinear system model to be generated at any given operating point. This analysis calculates the stability matrix (i.e. Jacobian) of the nonlinear system model and the eigenvalues of that matrix. This analysis can be described as follows. The nonlinear system model can be defined as:

\[
\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)
\]  

4.4-13

where:
- \( \mathbf{x} \) = n dimensional vector of state variable derivatives
- \( \dot{\mathbf{x}} \) = n dimensional vector of state variables
- \( \mathbf{f} \) = n dimensional vector of nonlinear functions relating state variables and time to state variable derivatives.

A linear model of this nonlinear system can be expressed as:

\[
\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}
\]  

4.4-14

where \( \mathbf{A} = n \times n \) system stability matrix
Figure 45. Nonlinear Stability Example
The $ij^{th}$ element of $A$, $a_{ij}$, is related to the partial derivative of the elements of $f$ with respect to the elements of $x$, at the operating point $x_0$ as

$$a_{ij} = \frac{\partial f_i(x,t)}{\partial x_j}$$

$x = x_0$

The eigenvalues of the stability matrix are a set of $n$ complex numbers that characterize the dynamic behavior of the system in a region about the chosen operating point, $x_0$. Eigenvalues with non-negative real parts indicate that the system is unstable in the region about, $x_0$.

It must be kept in mind that for highly nonlinear systems, this simple measure of stability is not a necessary or sufficient condition for stable operation. This can be demonstrated with a simple first order system as shown in Figure 45. For this example, the state derivative $\dot{x}$ is shown as a highly nonlinear function of the single system state variable, $x$. The arrows on the plot of the function show the trajectory the state, and state derivative would follow from any initial state $x$. For the values of $x$ shown, there is a stable region, and an unstable region. Initial values of $x$ in the stable region will result in the system reaching the steady state operating point, $x_{ss}$. Initial values of $x>x_2$ will result in $x$ diverging to large positive values.

The eigenvalue of this simple system is the partial derivative, $\frac{3x}{3x}$. We see that the simple criteria of a negative real eigenvalue for stability specifies that the system is unstable in the region $x_1$ to $x_2$, while for this example, it will converge to the steady state point, $x_{ss}$. In the region $x_3$ to $x_4$ the eigenvalue criteria would indicate that the system was stable, while in fact it will diverge from this region.

This example is presented to illustrate the hazards that exist when using eigenvalues to measure system stability at points other than steady state operating points. However, much useful information and
insights into system behavior can be obtained from such linear analyses. Especially since they can be easily verified by the nonlinear simulation capabilities of the EASY Analysis program.

The numerical method used to calculate the stability matrix is as follows: The values of the state variable derivatives, (rates) are calculated at the given operating point, \( x_0 \)

\[
\dot{x} = f(x_0, 0)
\]

where: 
- \( \dot{x}_0 \) = n dimensional vector of state derivatives at operating point
- \( x_0 \) = n dimensional vector of state variables which specifies the operating point.
- \( f \) = n dimensional vector of nonlinear functions relating state variables to state derivatives.

These values are printed and should be examined to determine if the operating point is a steady state operating point, i.e. \( \dot{x}_0 = 0 \).

Non zero elements of \( \dot{x}_0 \), (rates), indicate the sign and magnitude of unbalance at the chosen operating point.

The \( j^{th} \) element of the operating point vector is perturbed by adding the \( j^{th} \) element of the error vector, \( e_j \). This perturbed operating point is used to recalculate the state variable derivatives, \( \dot{x}_j \). The \( j^{th} \) column of the stability matrix, \( A_j \), is then calculated as:

\[
(A)_j = \frac{\dot{x}_j - \dot{x}_0}{e_j}
\]

* Note: this is the same vector that is used for integration error control. It’s values are furnished to the program via the ERROR CONTROL commands.
where: $\dot{x}_j = n$ dimensional vector of state derivatives at the operating point, perturbed by adding $j^{th}$ element of error vector to $x_0$.

$e_j = j^{th}$ element of the error vector.

$(A)_j = j^{th}$ column of the system stability matrix.

This process is repeated for all $n$ columns of $A$.

As a measure of the validity of the linear approximation, the stability matrix calculation described above is repeated using perturbations one half those used in the initial calculation.

The ratios of the derivatives calculated with the two step sizes are evaluated and placed in an array, RATIO. If the results of measuring all derivatives with both step sizes are equal, all elements of RATIO will equal one.

The elements of RATIO are compared to one and the number of elements differing from one by more than ten percent noted. If one or more such elements is found, the count of such elements is recorded on the printer along with a list of the elements of RATIO that exceed the tolerance of ten percent.

Figure 46 shows an example of how the values in the array RATIO may be used to measure the local linearity of the system model.

**Eigenvalue Calculation**

The method used to compute the eigenvalues of the system stability matrix consists of three basic steps. The first step is the conditioning of the matrix prior to the application of the normal transformation process.

The conditioning process is divided into two steps of reduction and scaling. Reduction is the process whereby through row and column interchange the matrix is transformed into upper block triangular form. This means that the diagonal blocks can be treated independently for the
RATIO (2, 1) = \frac{S_1}{S_2} \approx 1.

RATIO (5, 3) = \frac{S_1}{S_2} \neq 1.

Figure 46. Linearity Measure Example

Figure 47. Equivalent Stability Margin System
purpose of eigenvalue calculation. This reduction naturally occurs whenever openloop or feed forward systems are described. The algorithm used for reduction is described in detail in Appendix A under the title of the McCreight algorithm. For hisotrical perspective, an earlier method proposed by Harary is given. The second phase of the conditioning process is scaling. Since the errors in all the transformation algorithms used subsequent to the conditioning process are related to the norm of the matrix, scaling is used to reduce the norm. Historically it was though that the need for scaling was eliminated when the transition form analog to digital computers was made. Modern numerical analysis indicates that this is not true and that proper scaling is important to minimize loss of significance in computed results. The scaling algorithm used is one developed by E. E. Osborne in 1960 and consists of a sequence of diagonal transformations to minimize the Euclidean norm of a irreducible matrix. Since the reduction process was performed first, each diagonal block is irreducible and the scaling algorithm applies. Details of the algorithm are explained in Appendix A.

The second process in the computing of eigenvalues is to transform the scaled diagonal blocks determined in the first step into upper Hessenburg form. This form, where all the elements below the first sub-diagonal are zero, is most convenient and efficient for further calculation. In Appendix A, two methods are discussed with the "direct reduction with interchanges" being the method implemented.

The final step in the computation of eigenvalues is the actual determination of the eigenvalues for each diagonal block (now scaled and in Hessenburg form). The algorithm used is the QR algorithm developed by Francis in the early 1960's and described in Appendix A. The algorithm uses a series of unitary transformations to drive elements of the subdiagonal of the Hessenburg form to effective zero values. As the subdiagonal elements approach zero, the diagonal elements approach the
the desired eigenvalues. The algorithm is very efficient and quite suitable for problems of moderate size (less than 100-200 order).

Appendix A, which is comprised of notes from a series of lectures, presents the basic mathematics of each of the above processes along with numerical examples to demonstrate the actual computing sequence.

4.4.4 Stability Margin Calculations

The method that is used to determine stability margins is a frequency domain technique of Bode. This technique has been found to be numerically superior to other approaches, such as the Routh array approach and much faster than the direct approach of repeated eigenvalues determination.

The parameter K for which the stability margin is to be calculated can be thought of as providing a single loop feedback around the system model as shown in Figure 47.

The characteristic equation of the above system with nominal parameter \( K = K_n \), is:

\[
N(s) = P(s) - K_n Z(s)
\]

Note that the sign of the feedback is determined by the sign of K and is not assumed to be negative as is often the case in textbooks. The roots of \( N(s) \) are the eigenvalues of the nominal system, and the roots of \( P(s) \) are the eigenvalues of the system with \( K = 0 \).

To concentrate the analysis on the stability boundary of the complex plane, i.e. the imaginary axis, we may set \( s = j\omega \) in Equation 4.4-18. The polynomials \( P(j\omega) \) become complex quantities for real values of \( \omega \).

We are interested in determining those real values of \( K, K_o \), which will cause \( N(j\omega) = 0 \). Such values of \( K \) will result in roots of the characteristic equation on the imaginary axis of the complex plane.
Solving equation (4.4-18) for such values of \( K \) we obtain:

\[
0 = P(j\omega) - K_0 Z(j\omega) \quad 4.4-19
\]

\[
K_0 = \frac{P(j\omega)}{Z(j\omega)} \quad 4.4-20
\]

Since we are interested in only real values of \( K_0 \) that satisfy 4.4-19, we need consider only those values of \( \omega \) which cause the phase of \( P(j\omega)/Z(j\omega) \) to equal \( 0^\circ \) or \( 180^\circ \). Further, if the nominal parameter \( K_n < 0 \), only values of \( 180^\circ \) need to be considered, and if the nominal parameter \( K_n > 0 \), only the values of \( \omega \) that produces \( 0^\circ \) phase need be considered.

The approach that will be taken to determine \( K_0 \) will be as follows. The roots of \( N(s) \) and \( P(s) \) of 4.4-18 can be calculated as the eigenvalues of the nominal system, and the eigenvalues with \( K = 0 \) respectively, and will be designated as:

\[
N_i \quad i = 1, 2, \ldots, n \quad K = K_n
\]

\[
P_i \quad i = 1, 2, \ldots, n \quad K = 0
\]

Thus \( N(s) \) and \( P(s) \) can be stated in terms of their roots as:

\[
P(s) = \prod_{i=1}^{n} (s - P_i) \quad 4.4-22
\]

\[
N(s) = \prod_{i=1}^{n} (s - N_i)
\]

Solving 4.4-18 for the open loop transfer function in terms of \( K_n \), \( N(s) \) and \( P(s) \) we obtain:
\[
\frac{Z(s)}{P(s)} = \frac{1}{K_n} \left[ 1 - \frac{N(s)}{P(s)} \right] \quad 4.4-23
\]

\[
= \frac{R(s)}{K_n}
\]

Where:

\[
R(s) = 1 - \frac{N(s)}{P(s)}
\]

If \( K_n > 0 \), the phase of \( \frac{Z(s)}{P(s)} \) is the phase of \( R(s) \). If \( K_n < 0 \), the phase of \( \frac{Z(s)}{R(s)} \) is the phase of \( R(s) \) minus 180°. Thus, the method simplifies to a search for the frequencies that cause the phase of \( R(s) \) to be 0°, regardless of the sign of \( K_n \).

Substituting \( s = j\omega \) into 4.4-22 and 4.4-22 into 4.4-23 we obtain

\[
\frac{Z(j\omega)}{P(j\omega)} = \frac{1}{K_n} \left[ 1 - \prod_{i=1}^{n} \frac{(j\omega - N_i)}{(j\omega - P_i)} \right] \quad 4.4-24
\]

\[
= \frac{R(j\omega)}{K_n} \quad 4.4-25
\]

A range of \( 0 < \omega < \omega_{\text{max}} \) will be searched to find those values of \( \omega \) at which the phase of \( R(j\omega) \) is zero. At this frequency, \( \omega_o \), the limiting value of \( K \), \( K_o \), can be calculated by substituting 4.4-25 into 4.4-20:

\[
K_o = \frac{K_n}{||R(j\omega_o)||} \quad 4.4-26
\]

Magnitudes of \( R(j\omega) > 1 \) result in lower \( K \) limits. Magnitudes of \( R(j\omega) < 1 \) determine upper \( K_o \) limits. The usual definition of stability...
margin is the ratio of maximum $K_r$ to nominal, $K_n$ is obtained from 4.4-26 to be:

$$\frac{K_0}{K_n} = \frac{1}{R(j\omega_0)}$$ \hspace{1cm} 4.4-27

**Search for Zero Phase**

A range of $\omega$ from 0 to $\omega_{\text{max}}$ must be searched for zero crossings of $R(j\omega)$. $\omega_{\text{max}}$ is arbitrarily established as 2 times the magnitude of the largest eigenvalue of the nominal system. Zero frequency is included since a real divergence is indicated by a zero phase of $R(0)$. After $\omega = 0$ has been checked, the search begins at some low frequency $\omega_{\text{min}}$. Since we are interested in phase angles near 0, small angle approximations may be used for the phase of $R(j\omega)$. By this approach it will be possible to avoid time consuming trigonometric calculations. Thus phase angle of $R(j\omega)$ will be approximated as:

$$\arg R(j\omega) \approx \frac{\text{Im } R(j\omega)}{\text{Re } R(j\omega)}$$ \hspace{1cm} 4.4-28

The search proceeds with geometric steps from $\omega_{\text{min}}$. When a zero crossing occurs, the search switches to a dichotomous mode until the error is reduced to some tolerance $\epsilon$, i.e.

$$|\arg R(j\omega)| < \epsilon = 0.00001 \text{ radian}$$ \hspace{1cm} 4.4-29

A further condition is included in this search strategy. That is that the phase angles determined on two subsequent geometric search steps should not differ by more than one quadrant. This condition is included to prevent the search from not detecting a zero crossing in a region of rapidly changing phase.

The mode of the search can be easily related to the standard quadrant designations of the phase angles as described below.
The absolute value of the difference of the quadrant numbers of the current and previous phase angle is calculated. If this value is less than two, the geometric search is continued. If this value is equal to two, a small step backward is taken, since a change of two quadrants has occurred and a zero crossing may have been overlooked. If this value is greater than two, the phase angle has passed from the first to fourth (or visa-versa), quadrant and a dichotomous search is started to locate the value of frequency that produces zero phase.

When such a value of frequency is determined, the value, \( \omega_0 \), and the stability margin, \( \frac{1}{R(j\omega_0)} \), are stored in arrays, and the search continues in the geometric fashion until \( \omega_{max} \) is reached.

At this point in the analysis, there are two arrays of k elements \( \Omega(i) \) and \( GM(i) \) that contain the frequencies \( \omega_0 \) and the corresponding magnitudes \( \frac{1}{R(j\omega_0)} \) respectively. The lower stability limit is determined by the maximum value of \( \frac{1}{R(j\omega_0)} \) which is less than 1.

The upper stability limit is determined by the minimum value of \( \frac{1}{R(j\omega_0)} \) which is greater than 1. The k elements of \( GM(i) \) are searched to determine these values. Any remaining elements of \( GM(i) \) and \( \Omega(i) \) indicate parameter values and divergence frequencies which exceed the critical stability limits, but at which another oscillation would occur if the parameter were increased beyond the critical stability bounds. If such values exist in the searched region, they will be printed out by the program as noncritical stability limits.

4.4.5 Transfer Function Calculations

The method that is used to calculate transfer functions is very similar to that used to calculate stability margins. In each case, the eigen-
values of the nominal system, and the eigenvalues of a related system, are calculated and used to obtain the desired results. Since the eigenvalues of a linearized system can be calculated quite efficiently and accurately, this approach provides an efficient and accurate method of obtaining specified transfer functions.

The transfer function from any point R to any point C in the system model can be represented as shown in Figure 48. The transfer function between points R and C is composed of the ratio of rational polynomials Z(s) and P(s).

\[
\begin{align*}
\frac{C(s)}{R(s)} &= \frac{Z(s)}{P(s)} \\
\end{align*}
\]

where:  
R(s) - the specified input quantity.  
C(s) - the specified output quantity.  
Z(s) - transfer function numerator polynomial.  
P(s) - transfer function denominator polynomial.  
s - Laplace complex frequency variable.

The roots of the denominator, P(s) can be obtained by forming a linear representation of the system and calculating the nominal system eigenvalues, as discussed in Section 4.4-3. If the equivalent transfer function system of Figure 49 is modified by adding a feedback path from the specified output quantity to the input quantity, we obtain a new dynamic system whose transfer function is:

\[
\begin{align*}
\frac{C'(s)}{R(s)} &= \frac{Z(s)}{Z(s) + P(s)} \\
\end{align*}
\]

Let the roots of P(s), the nominal system eigenvalues, be designated as P₁, and the roots of Z(s) + P(s), the modified system eigenvalues, be designated N₁, i = 1, 2, ..., n where n is the system order.
Figure 48. Equivalent Transfer Function

\[
\frac{C(S)}{R(S)} = \frac{Z(S)}{P(S)}
\]

Figure 49. Modified Equivalent Transfer Function System

\[
\frac{C'(S)}{R(S)} = \frac{Z(S)}{Z(S) + P(S)}
\]
\[ P(s) = \prod_{i=1}^{n} (s-P_i) \]  \hspace{1cm} 4.4-32

\[ Z(s) + P(s) = \prod_{i=1}^{n} (s-N_i) \]  \hspace{1cm} 4.4-33

The desired transfer function, \( \frac{Z(s)}{P(s)} \), can be obtained in terms of the two sets of eigenvalues \( P_i \) and \( N_i \) by dividing equation 4.4-33 by 4.4-32.

\[ \frac{Z(s)}{P(s)} = \frac{\prod_{i=1}^{n} (s-N_i)}{\prod_{i=1}^{n} (s-P_i)} - 1 \]  \hspace{1cm} 4.4-34

Since we are interested in the steady state frequency response, we will confine our attention to the imaginary axis of the S plane, by replacing \( s \) with \( j\omega \).

\[ \frac{Z(j\omega)}{P(j\omega)} = \frac{\prod_{i=1}^{n} (j\omega-N_i)}{\prod_{i=1}^{n} (j\omega-P_i)} - 1 \]  \hspace{1cm} 4.4-35

Equation 4.4-35 gives the desired transfer function in terms of the eigenvalues of the nominal system, and that system modified by a single loop closure. Since \( N_i \) and \( P_i \) are, in general, complex quantities, and the \( j\omega \) terms are pure imaginary quantities, the transfer function will be a complex function of \( \omega \).
The numerical methods that are used to calculate the nominal system stability matrix and eigenvalues are described in Section 4.4.3. The modified system stability matrix is calculated as follows: First, the nominal value of the specified output quantity, $C_0$, is determined. At each step of the stability matrix calculation, after a $j$th state variable has been perturbed, the difference between the resulting value of $C$, $C_j$, and the nominal value $C_0$ is subtracted from the current value of the input quantity, $R_j$.

\[ R'_j = R_j - (C_j - C_0) \]  

where:  
\begin{align*}  
R'_j & \quad \text{input quantity modified by -1 loop closure from C.} \\
R_j & \quad \text{input quantity without -1 loop closure from C.} \\
C_0 & \quad \text{nominal value of output quantity.} \\
C_j & \quad \text{output quantity value resulting from perturbing jth state variable.} 
\end{align*}

The system model is then re-evaluated from the point in the model equations at which $R$ appears. In this way the effect of a -1 loop closure from output to input is simulated. Note, that this technique fails if the output quantity is a direct, algebraic function of the input quantity. In such a case, the change in $C$ would cause a change in $R$ via (4.4-36), which would cause a further change in $C$, etc. A test for such "algebraic loops" is performed before the transfer function analysis is allowed to proceed. This situation only occurs in those cases in which the transfer function numerator polynomial and denominator polynomial are of the same order. This situation is fortunately quite uncommon in most physical dynamic systems.

### 4.4.6 Root Locus Calculation

A root locus analysis provides the locus of the system eigenvalues as a function of some specified parameter. The EASY Analysis program allows a root locus analysis to be performed as a function of any operating
point value, as well as any system parameter.

The root loci are calculated by forming the stability matrix for the system for each specified value of the root locus parameter. The eigenvalues of each stability matrix are calculated to give the root loci.

The methods described in Section 4.4.3 are used to calculate the system stability matrices and eigenvalues. However, the calculation of the linearity measure, RATIO, is omitted for two different* values of the root locus parameter. A comparison of the elements of these stability matrices is made to determine which elements are affected by changes in parameter. Subsequent stability matrix calculations only re-evaluate those elements which were modified by the first two values of the root locus parameter. Due to storage limitations, a limit has been placed on the number of elements that can be modified by the root locus parameter. This limit is 400 elements of the stability matrix. If more than 400 elements of the stability matrix are modified by the root locus parameter, the program reverts to the less efficient process of evaluating all elements of the stability matrix for each value of the root locus parameter.

4.4.7 Eigenvalue Sensitivity Calculations

An eigenvalue sensitivity analysis provides a measure of the sensitivity of system eigenvalues to changes in a specified system parameter. The eigenvalue sensitivity measure is the ratio of the percentage change in the parameter for which the sensitivity is to be measured. This is stated

* The two different values are the nominal parameter value and the RL START value. Therefore RL START should not equal the nominal parameter value.
mathematically as:

\[ S_{\sigma i} = \frac{1 - \sigma_i}{\left| 1 - \frac{\sigma_i}{P'} \right|} \quad 4.4-37 \]

\[ S_{\omega i} = \frac{1 - \omega_i}{\left| 1 - \frac{\omega_i}{P'} \right|} \quad 4.4-38 \]

Where:

- \( S_{\sigma i} \) = Sensitivity measure of real part of \( i^{th} \) eigenvalue to change in parameter \( P \).
- \( S_{\omega i} \) = Sensitivity measure of imaginary part of \( i^{th} \) eigenvalue to changes in parameter \( P \).
- \( \sigma_i \) = Nominal value of real part of \( i^{th} \) eigenvalue
- \( \omega_i \) = Nominal value of imaginary part of \( i^{th} \) eigenvalue
- \( P \) = Nominal value of parameter for which sensitivity measure is being calculated
- \( \prime \) = Prime indicates perturbed values of parameters and eigenvalues
- \( i = 1, 2, ..., n \quad n = \text{model order} \)

This sensitivity measure has the following properties:

a. It is dimensionless which allows the relative sensitivities of parameters with different units to be compared.

b. Sensitivity measure of one indicates equal percentage change in eigenvalue per unit change in the parameter.

c. Positive sensitivity indicate eigenvalue motion toward the right half plane, i.e., destabilizing and lower frequencies.

d. Negative sensitivities indicate eigenvalue motion toward the left half plane, i.e., stabilizing, and higher frequencies.
4.4.8 Function Scan Calculations

Function scan calculations begin by setting the system state variable to the current operating point values, and all state variable derivatives to zero.

The system model equations are then evaluated. The specified independent variable, INDEP1, is then set to its initial value, START1, and the model equations are re-evaluated. If the independent variable is a state variable or parameter, the model equations are completely re-evaluated. However, if the independent variable is a variable or rate, which would normally be calculated by the model equations, the re-evaluation begins at the statement immediately following the normal calculation of the variable or rate. In this way, the effect of the variable or rate on the model is determined for the specified, rather than the normal value calculated by the model. This process of re-evaluation is repeated as the independent variable is scanned from START1 to STOP1. After each re-evaluation the value of the specified dependent variable DEPEN is recorded.

If a second independent variable, INDEP2, is specified, this variable is set to its specified value before each scan of INDEP1 and the model is re-evaluated. This places a constraint on the relationship of INDEP1 to INDEP2:

If INDEP2 is a variable or rate, INDEP1 must be a variable or rate that is calculated below INDEP2 in the model calculation sequence.

If this constraint is violated, INDEP2 will not scan its specified values, but will merely take on its nominal model calculated values. Such a conflict can always be resolved by interchanging INDEP1 and INDEP2. If this form of plots is not desired, the desired family of curves can be obtained by repeated use of the SCAN1 option with INDEP2 varied using the PARAMETER VALUES command.
APPENDIX M

OPTIMAL CONTROLLER DESIGN WITH THE EASY PROGRAM

This appendix is a reproduction of Section 4.5 of reference 1. It presents a description of the optimal controller designs performed by the EASY Analysis Program.
4.5 OPTIMAL CONTROLLER DESIGN

The optimal controller designs performed by the EASY Analysis program are based on the linear optimal regulator theory and linear filter theory of Kalman. By allowing the designer to specify the model order and optimal controller order he wishes to use it is possible to apply the theory to large system models and to obtain reasonable sized practical controller designs.

The design process is shown in Figure 50 where the dashed line indicates engineering feedback needed until the design obtained is acceptable by some criterion. The basic flow indicates the linearization about a desired operating point, the reduction of the linear model, and then the calculation of the optimal gain and filter matrices via linear optimal regulator theory. The initial reduction of order in the linear description is permitted in order to reduce computational and storage requirements in the subsequent controller calculations. Likewise, before leaving the design process, the complexity of the calculated controller can be reduced to any prescribed level to facilitate practical realizations and analysis. The final tasks of preliminary linear analysis (eigenvalues of resultant system with reduced controller) and subsequent simulation of full nonlinear systems with reduced controller are needed to assess the real performance of the design. Based on this, the designing engineer can adjust design parameters to effect more desirable behavior.

Section 4.5.1 considers the model linearization. The method for reduction of the order of linear systems is delayed until Section 4.5.12. The factors affecting the design parameters are considered in Section 4.5.2 where the basic problem definition is given. Section 4.5.3 treats model considerations, including the calculation of default values for design parameters. In Section 4.5.4 the theory for the optimal gain matrix calculation is given. The detailed calculation process is given in
CHOOSE OPERATING POINT

LINEARIZE MODEL

REDUCE LINEAR MODEL

CHOOSE DESIGN PARAMETERS

CALCULATE GAIN MATRIX

CALCULATE OPTIMAL FILTER

FORM FULL CONTROLLER

REDUCE AND ADJUST CONTROLLER

PRELIMINARY LINEAR ANALYSIS

SIMULATION

Figure 50. Central Design Process

Tasks performed by Design O.C. Command
Section 4.5.5 while Section 4.5.6 indicates what analysis information is generated as a result of the calculation process. Section 4.5.7 parallels the development for the Kalman filter with Section 4.5.8 giving the detailed calculation process and Section 4.5.9 the analysis information. Section 4.5.10 then covers the controller formation and subsequent reduction and adjustment. Section 4.5.11 considers the reduction theory with 4.5.12 giving the detailed calculation sequence. Finally, Section 4.5.13 considers the use of the designed controller in the nonlinear system simulation.

4.5.1 Linear Model Generation

The design process starts with the generation of a complete linear model of the system at the specified operating point. This non-linear system model can be expressed by Equations 4.5-1 through 5.4-3.

\[
\begin{align*}
\dot{x} &= f(x,u,t) & 4.5-1 \\
y_s &= f_s(x,t) & 4.5-2 \\
y_c &= f_c(x,u,t) & 4.5-3
\end{align*}
\]

where

\[
\begin{align*}
\dot{x} &= n_x \text{ dimensional state vector} \\
u &= n_u \text{ dimensional control vector} \\
y_s &= n_s \text{ dimensional sensor vector} \\
y_c &= n_c \text{ dimensional criteria vector} \\
f &= n_x \text{ dimensional vector of nonlinear functions relating state variable, inputs, and time to the state variable derivatives} \\
f_s &= n_s \text{ dimensional vector of nonlinear functions relating state variables to sensed quantities}
\end{align*}
\]
\( f_c = n_c \) dimensional vector of nonlinear functions relating state variables, inputs, and time to criteria quantities.

A linear model of this system is obtained by numerically taking the partial derivatives of \( f, f_s, \) and \( f_c \) with respect to \( x \) and \( u \) as described in Section 4.4.3. The equations thus obtained are:

\[
\begin{align*}
\dot{x} &= Ax + Bu + I_x d \\
Y_s &= H_s x + I_s v \\
Y_c &= H_c x + D_c u
\end{align*}
\]

where:

- \( A = n_x \) by \( n_x \) system stability matrix
- \( B = n_x \) by \( n_u \) system input matrix
- \( H_s = n_s \) by \( n_x \) system sensor matrix
- \( H_c = n_c \) by \( n_x \) criteria matrix
- \( D_c = n_c \) by \( n_u \) criteria input disturbance matrix
- \( I_x = n_x \) by \( n_x \) identity matrix
- \( I_s = n_s \) by \( n_s \) identity matrix
- \( d = n_x \) dimensional state disturbance vector
- \( v = n_s \) dimensional sensor disturbance vector

Note that it is assumed that the control vector, \( u \), of actuator input does not directly effect the sensed quantities, \( Y_s \). The control quantities do effect the sensed quantities via their effect on the system states.
4.5.2 Design Formulation

The state vector $x$ represents deviations from a desired set point and the control vector $u$ represents perturbations about the control level at the set point. The vector $d$ is a disturbance vector for the state derivatives and for this problem is considered to be a zero mean white noise process with a covariance matrix given by a diagonal matrix $C_d$. Likewise $v$ is a zero mean white noise process affecting the sensors and has a diagonal covariance matrix $C_v$. With this description, it is to be noted that all set point levels for the state, control, and noise $v$ vectors have been removed. Further, all noise correlation is assumed to be included through additional states representing filtered white noise. Details of this procedure are treated in a later section. The theory presented does not require this limited disturbance description and the design programs can easily be altered to include non-diagonal covariance matrices and a more general multiplier (instead of the identity matrix). The choice was made to facilitate understanding of the design procedure and to reduce both storage requirements and required input data. Further, the chosen level of generality is sufficient for most all design problems considered in the preliminary design and analyses stages.

The design criterion is given by a cost functional

$$ J = \frac{1}{2} \int_{-\infty}^{\infty} (y' \quad Q \quad y_c + u' \quad R \quad u) \, dt $$

where $Q$ is a positive semi-definite weighting matrix relating the relative importance of the various criteria variables and is assumed diagonal (any off diagonal weighting can be accounted for by a redefinition of the variables in the vector $y_c$). The control weighting matrix $R$ is a positive definite matrix and for convenience assumed diagonal (little physical interpretation can be given to off diagonal terms).

The design problem of interest is to obtain a description of $u$ as a
function of the sensor outputs given by $y_s$ that causes the cost functional of Equation 4.5-7 to be minimized given any initial displacement.

4.5.3 Modeling Considerations

Model Assumptions

Several assumptions are made in the problem description just given for the sake of ease of computing and storage. The zero-mean value assumption for both the state and sensor equations is made knowing that non-zero-mean quantities are included in the set point values. Realizing that equations 4.5-5 through 4.5-6 are for deviations about set point values, the disturbance descriptions are for deviations about their mean values.

The assumption that each state derivative is affected by white noise uncorrelated with that affecting other states seems more restrictive. In practice, however, if one defines band limiting filter equations and accounts for the correlation through the output of the filter entering into the equations for the affected state derivatives, most cases can be approximately treated. The theory that follows does not require this limitation and the computer programs implementing the algorithms can be modified to include the more general form of the disturbance function. With the limitation, however, the amount of data input and internal storage is reduced.

Design Default Value

From the problem description, the design parameters are the $Q$ and $R$ vectors for the gain calculation and the $C_d$ and $D_v$ vectors for the filter calculation. The defining equations for the criteria variables are also part of the design specification but are more likely to remain fixed for any given problem whereas the $Q$ and $R$ vectors are varied to effect different performing systems. The choice of the elements of $Q$ and $R$ are relative to each other and not absolute (doubling all the
elements of each does not change the problem). Since \( R \) must be positive definite a logical default value for any element of \( R \) less than or equal to zero is unity. Likewise for \( Q \) which must be positive semi-definite, default values are unity for any element less than zero. The above two sets of default values do not take into account any relative sizes of criteria or control variables but only assure the sign definite requirements of the problem formulation.

Default values for the noise covariance matrices (assumed diagonal) used in the calculation of the Kalman Filter require more computation in that they are less likely to be input by design engineers due to less familiarity—especially in the initial stages of the problem. To get some physical interpretation, if one assumes that noise causes errors (both in the state derivatives and in the measurements) that are normally distributed about the correct value with 95% of the errors within a bound \( \pm \alpha \), then the appropriate choice for the variance \( (\sigma^2) \) is given by

\[
\sigma^2 = \frac{\alpha^2}{3.8416}
\]

4.5-8

This equation is derived through the use of the erf function as

\[
2 \operatorname{erf} \left( \frac{\alpha}{\sigma} \right) = .95
\]

4.5-9

or

\[
\frac{\alpha}{\sigma} = 1.96
\]

4.5-10

which is obtained from a table for the erf function. Equation 4.5-8 is then a direct result of Equation 4.5-10.

To get some bounds on the errors in the calculation of state derivatives due to both external disturbances and model inaccuracy, a measure of the relative size of each state is needed. In the EASY program, this is provided by the ERROR vector. Thus to obtain uncertainty bounds for the
state derivatives, the following equation is used for limit values $L^i$.

$$L^i = 10 \sum_{j=1}^{n_x} |a_{ij} \cdot \text{ERROR}(j)|$$ \hspace{1cm} 4.5-11

which indicates the sum of all the absolute state minimum perturbation sizes weighted by the multiplier in the system matrix $A$. The 10 multiplier is artificial and used to account for model inaccuracy in general and to force the resulting design to favor current measurements rather than historical information (which will happen if the model is assumed more accurate than the measurements) the actual covariance matrix elements is then computed as

$$\sigma_{11}^2 = \frac{L_1^2}{3.8416}$$ \hspace{1cm} 4.5-12

The noise covariance matrix for the measurements is computed in a similar manner where the limits $L^i$ are computed as

$$L^i = \sum_{j=1}^{n_x} |(H_s)_{ij} \cdot \text{ERROR}(j)|$$ \hspace{1cm} 4.5-13

which weights the measurements relative to the minimum perturbations in the states. This is not ideal but suffices in the absence of any other data.

It is anticipated that these default values will help get a design started but that as experience is gained with the model and with resulting controllers better values can be input to more fully effect the "best" design.
4.5.4 Gain Matrix Calculation

The separation theorem of linear optimal control states that the optimal controller is composed of a linear feedback gain matrix $G$ operating on an optimal estimate of the state obtained through the use of a Kalman filter. The feedback matrix $G$ is computed as if no noise disturbances were present and as if all the states are available for feedback. The following section outlines the procedure for calculating the optimal feedback gain matrix $G$.

Substitution of the expression for $y_c$ in equation 4.5-6 into the cost functional of equation 4.5-7 yields

$$ J = \frac{1}{2} \int (H_c x + D_c u)'Q(H_c x + D_c u) + u' R u) dt $$

Following a procedure using the Minimum Principal of Pontryagin (Ref.2) one forms the Hamiltonian for this system as

$$ H = \frac{1}{2} \{ x' H_c 'Q H_c x + u'D_c 'Q H_c x + x'H_c 'Q D_c u + u'(R + D_c 'Q D_c )u \} $$

$$ + p' A x + p' B u $$

where $p$ is now the costate vector. The differential equation for $p$ is given by

$$ \dot{p} = - \frac{\partial H}{\partial x} = - (H_c 'Q H_c x + H_c 'Q D_c u + A' p) $$

634
A necessary condition for an optimal solution is given by

\[
\frac{\partial H}{\partial u} = 0 = D_c'QH_c x + (R + D_c'QD_c)u + B'p
\]

4.5-17

which implies

\[
u = -(R + D_c'QD_c)^{-1}(D_c'QH_c x + B'p).
\]

4.5-18

Therefore, substitutions of the expression for \( u \) into the differential equations for \( x \) and \( p \) yields

\[
\dot{x} = x - B (R + D_c'QD_c)^{-1}(D_c'QH_c x + B'p)
\]

4.5-19

\[
\dot{p} = -A'p - H_c'QH_c x + H_c'QD_c (R + D_c'QD_c)^{-1}(D_c'QH_c x + B'p)
\]

4.5-20

or in matrix form

\[
\begin{bmatrix}
\dot{x} \\
\dot{p}
\end{bmatrix}
= \begin{bmatrix}
-A - B(R + D_c'QD_c)^{-1}D_c'QH_c & -B(R + D_c'QD_c)^{-1}B' \\
-H_c'(Q - QD_c(R + D_c'QD_c)^{-1}D_c'Q)H_c & -A' + H_c'QD_c(R + D_c'QD_c)^{-1}B'
\end{bmatrix}
\begin{bmatrix}
x \\
p
\end{bmatrix}
\]

4.5-21
where:

\[
\begin{align*}
\tilde{A} &= A - B(R + D_c 'QD_c)^{-1}D_c 'QH_c \quad 4.5-22 \\
\tilde{R} &= (R + D_c 'QD_c) \quad 4.5-23 \\
\tilde{Q} &= Q - Q D_c (R + D_c 'QD_c)^{-1}D_c 'Q \quad 4.5-24
\end{align*}
\]

Since $R$ was assumed positive definite and $Q$ positive semi-definite, it can be shown that $\tilde{R}$ is also positive definite and $\tilde{Q}$ is positive semi-definite.

A second condition termed the transversality condition requires that

\[
p(t) \big|_{t=\infty} = 0. \quad 4.5-25
\]

When the initial condition for $x(t)$ is considered, it is seen that equations 4.5-21 and 4.5-25 pose a two point boundary value problem. In order to solve for $p(t)$ and $x(t)$ which are needed to determine the control $u(t)$, consider a change of variable

\[
\tau = \infty - t \quad 4.5-26
\]

which when used in equations 4.5-21 and 4.5-25 results in

\[
\begin{bmatrix}
\dot{x}(\tau) \\
\dot{p}(\tau)
\end{bmatrix} =
\begin{bmatrix}
-A & -1 \\
BR & B'
\end{bmatrix}
\begin{bmatrix}
x(\tau) \\
p(\tau)
\end{bmatrix} \quad 4.5-27
\]

\[
p(\tau) \big|_{\tau=0} = 0 \quad 4.5-28
\]

\[
x(\tau) \big|_{\tau=\infty} = x. \quad 4.5-29
\]
Now let $\Omega$ be the fundamental* matrix for the system matrix in equation 4.5-27. Partition $\Omega$ into quadrants corresponding to the partition in equation 4.5-27 to obtain

$$\begin{bmatrix} x(\tau) \\ p(\tau) \end{bmatrix} = \begin{bmatrix} \Omega_{11}(\tau) & \Omega_{12}(\tau) \\ \Omega_{21}(\tau) & \Omega_{22}(\tau) \end{bmatrix} \begin{bmatrix} x(\tau) | \tau = 0 \\ p(\tau) | \tau = 0 \end{bmatrix}$$

4.5-30

Now using the condition of Equation 4.5-28

$$\begin{bmatrix} x(\tau) \\ p(\tau) \end{bmatrix} = \begin{bmatrix} \Omega_{11}(\tau) \end{bmatrix} \begin{bmatrix} x(\tau) | \tau = 0 \\ 0 \end{bmatrix}$$

4.5-31

$$p(\tau) = \Omega_{21}(\tau) \begin{bmatrix} x(\tau) | \tau = 0 \end{bmatrix}$$

4.5-32

from which one obtains

$$p(\tau) = \Omega_{21}(\tau) \Omega_{11}^{-1}(\tau) x(\tau)$$

4.5-33

providing $\Omega_{21}(\tau)$ is non singular. Since $\Omega_{11}(\tau)$ is equal to the identify matrix at $\tau$ equal to zero and is a fundamental matrix, it is nonsingular for all $\tau$.

Drawing on some results by J. J. O'Donnell, (Ref. 3), it is known that the system matrix of equation 4.5-27 has eigenvalues symmetric with respect to both the real and imaginary axis of the complex plane. This is shown by using a linear transformation

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

4.5-34

which when applied to the system matrix of equation 4.5-27 indicates it is similar to a matrix whose eigenvalues are the negative of its own.

* Also referred to as the state transition matrix.
The conditions of \( R \) and \( Q \) being positive definite and semidefinite is sufficient to insure all eigenvalues with zero real parts are of multiplicity 2. Using these facts, let \( W \) be a transformation such that

\[
W^{-1} \begin{bmatrix}
\hat{A} & BR & B' \\
H_C QH_C & A'
\end{bmatrix} W = \begin{bmatrix}
\Lambda & 0 \\
0 & -\Lambda'
\end{bmatrix}
\]

where all the eigenvalues of \( \Lambda \) have non-negative real parts and complex eigenvalues occur in conjugate pairs. Thus

\[
\Omega(\tau) = W \begin{bmatrix}
 e^{\Lambda \tau} & 0 \\
0 & e^{\Lambda' \tau}
\end{bmatrix} W^{-1}
\]

Let

\[ U = W^{-1} \]

and partition \( U \) and \( W \) to obtain

\[
\Omega_{11}(\tau) = W_{11} e^{\Lambda \tau} U_{11} + W_{12} e^{\Lambda' \tau} U_{21}
\]
\[
\Omega_{21}(\tau) = W_{21} e^{\Lambda \tau} U_{11} + W_{22} e^{\Lambda' \tau} U_{21}
\]

Then equation 4.5-33 reduces to

\[
p(\tau) = \left[ W_{21} e^{\Lambda \tau} U_{11} + W_{22} e^{\Lambda' \tau} U_{21} \right] \left[ W_{11} e^{\Lambda \tau} U_{11} + W_{12} e^{\Lambda' \tau} U_{21} \right]^{-1} x(\tau)
\]
Since we are interested in the control law in the time frame of $t$ near zero, we must look at $p(t)$ as $t$ approaches $\infty$. If $\Lambda$ has all eigenvalues with positive real parts (not just non-negative) then as $\tau$ becomes large the terms with $e^{\Lambda \tau}$ must become small with the result that for large $\tau$

$$p(\tau) = \left[ W_{21} e^{\Lambda \tau} U_{11} \right] \left[ W_{11} e^{\Lambda \tau} U_{11} \right]^{-1} x(\tau)$$

which assuming non singularity of $W_{11}$ and $U_{11}$ yields

$$p(\tau) = W_{21} e^{\Lambda \tau} U_{11} \left[ U_{11}^{-1} (e^{\Lambda \tau})^{-1} W_{11}^{-1} \right] x(\tau)$$

$$= W_{21} W_{11}^{-1} x(\tau)$$

as $\tau$ approaches $\infty$. Thus for $t$ near zero, from equation 4.5-18 we obtain

$$u(t) = -R^{-1} (D_c Q + B' W_{21} W_{11}^{-1}) x(t).$$

The condition that causes the indicated inverses $W_{11}$ and $U_{11}$ not to exist is the existence of a unstabilizable mode in the original system equations. If the mode has eigenvalues with zero real parts, the assumption that $e^{\Lambda \tau}$ terms in equation 4.5-40 become small with respect to $e^{\Lambda \tau}$ terms is incorrect. If the mode has eigenvalues with positive real parts, then $W_{11}$ will be singular. To see this consider a system of equations

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \end{bmatrix} u$$

4.5-44
in which $A_1$ has eigenvalues with positive real parts.

The resulting matrix for equation 4.5-27 is

$$\begin{bmatrix}
  \dot{x}_1 (\tau) \\
  \dot{x}_2 (\tau) \\
  \dot{p}_1 (\tau) \\
  \dot{p}_2 (\tau)
\end{bmatrix} =
\begin{bmatrix}
  -A_1 & 0 & 0 \\
  0 & -A_2 & B_2 R^{-1} B_2 \\
  Q_{11} & Q_{12} & A_1 \\
  Q_{12} & Q_{22} & A_2
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  p_1 \\
  p_2
\end{bmatrix}
\quad \tau = 0$$  

4.5-45

Note now that as one computes the eigenvectors corresponding to eigenvalues with positive real parts the only portion of the eigenvector that can be non-zero is that corresponding to the third partition. Thus $W_{11}$ would have columns of zeros corresponding to each variable in $x_1$.

The conclusion of this section is that if one is able to partition the eigenvalues as indicated in Equation 4.5-35, and if none of the eigenvalues have zero real parts, and if the inverse of $W_{11}$ exists, then the solution given in 4.5-43 is the correct solution. In practice, the program used to implement the procedure require that the matrix in Equation 4.5-35 be diagonalizable so that if $W_{11}$ is singular, it might also be the result of this restriction not being satisfied.

4.5.5 Solution Process

The numerical process for computing the gain matrix is given by:

1. Form the matrix for the system and adjoint equations as in Equation 4.5-27 with definitions 4.5-22, 4.5-23 and 4.5-24.
2. Compute the eigenvalues of the matrix formed. If any eigenvalues have zero (with the precision of the computation) real parts, this indicates that the system is unstabilizable and that no solution exists. (See Appendix A).
3. Partition the eigenvalues into two groups with all eigenvalues with positive real parts in the first group.
4. Compute eigenvectors for each eigenvalue with a positive real part. (See Appendix A).
5. Partition the eigenvectors computed into matrices \( W_{11} \) and \( W_{21} \).
6. Solve for \( B'W_{21}W_{11}^{-1} \) where \( W_{11}^{-1} \) exists. If \( W_{11} \) is singular (within precision limitations) indicate that either the original system had an unstabilizable (unstable and uncontrollable mode or that the rare event of a non-diagonalizable system + adjoint matrix occurred.
7. Compute the gain matrix

\[
G = -R^{-1}(D'QH + B'W_{21}W_{11}^{-1})
\]

4.5.6 Closed Loop Eigenvalues

Computing the optimal feedback matrix in this manner yields information on the resulting closed loop linear control system. From equation 4.5-35

\[
\tilde{A}W_{11} + BR^{-1}B'W_{21} = W_{11}A
\]

Where \( A \) contained the eigenvalues with positive real parts.

Postmultiplying by \( -W_{11}^{-1} \) one obtains

\[
\tilde{A} - BR^{-1}B'W_{21}W_{11}^{-1} = W_{11}A W_{11}^{-1}
\]

or when \( \tilde{A} \) and \( R \) are substituted as in Equation 4.5-22 and 4.5-23

\[
A - B(R+D'_c QD_c)^{-1}(D'_c QH_c + B'W_{21}W_{11}^{-1}) = W_{11}(-\Lambda)W_{11}^{-1}
\]
Recognizing the second term as B times the optimal gain matrix G computed in Equation 4.5-46, one obtains
\[ A + BG = W_{11} (-\Lambda) W_{11}^{-1} \] 4.5-50

which indicates that the optimal closed loop system given by A+BG has the eigenvalues of -\Lambda. For -\Lambda in a diagonal form W_{11} is the set of eigenvectors. Note that as \Lambda was chosen as all the eigenvalues with positive real parts, -\Lambda must have all eigenvalues with negative real parts. Thus A+BG must be stable.

4.5.7 Kalman Filter Calculation

In this section the filter portion of the total controller is considered. Using the notation of Section 4.5.1 and the results of Theorem 7.1 in the book by Meditch, (Ref. 4), the optimal filtered estimate for the system described in Equations 4.5.4 through 4.5.6 is given by
\[ \dot{\hat{x}}(t) = A\hat{x}(t) + S(t) \left[ Y_s(t) - H_s \hat{x}(t) + B u(t) \right] \] 4.5-51

where
\[ \hat{x}(0) = 0 \]
and where
\[ S(t) = \dot{P}(t) H_s' C_v^{-1} \] 4.5-53

and where P(t) satisfies the differential equation
\[ \dot{P}(t) = A P(t) + P(t) A' - P(t) H_s' C_v \] \[ H_s \] \[ P(t) + C_d \] 4.5-54

with
\[ P(0) = E \left[ x(0) x'(0) \right] \] 4.5-55

642
where the term on the right side of Equation 4.5-55 is the covariance of the state at time zero. Although \( P(t) \) and thus \( S(t) \) are in general time varying, it is undesirable from an implementation point of view to design time variable controllers. More realistically if one assumes that the covariance of the filtered estimate is at steady state which is obtained as the limiting value of \( P(t) \) as \( t \) becomes large in Equation 4.5-54, then \( S \) given in Equation 4.5-53 becomes a constant matrix with the result that the filter equations are linear and time-invariant.

In order to solve

\[
A p + P A' + PH_s C_v^{-1} H_s P + C_d = 0
\]  
4.5-56

can use the eigenvector approach reported by Potter (Reference 5) and by O'Donnell (Reference 3) which states that

\[
P = W_{21} W_{11}^{-1}
\]  
4.5-57

\[
\begin{bmatrix}
-A' & H_s C_v^{-1} H_s \\
C_d & A
\end{bmatrix}
\begin{bmatrix}
W_{11} \\
W_{21}
\end{bmatrix} =
\begin{bmatrix}
W_{11} \\
W_{21}
\end{bmatrix}^\Lambda 
\]  
4.5-58

and where \( \Lambda \) is the set of eigenvalues of the matrix on the left hand side of equation 4.5-58 that have positive real parts. Then \( W_{11} \) and \( W_{21} \) are partitions of the set of eigenvectors corresponding to eigenvalues with positive real parts. The solution is analogous to that computed for the gain matrix in the optimal regulator problem and the conditions that all unobservable modes are stable along with \( C_v \) positive definite and \( C_d \) positive semi-definite insure the existence of \( W_{11}^{-1} \) and a solution.

Having calculated \( W_{21} \) and \( W_{11} \), the \( S \) matrix defined in Equation 4.5-53 can be evaluated from the expression
\[ S = W_{21} W_{11}^{-1} H_s' C_v^{-1} \]

by first solving the linear system of equations for \( W_{11}^{-1} H_s' C_v^{-1} \) and then premultiplying by \( W_{21} \). The dynamic equations for the Kalman filter can now be written (from equation 4.5-51) noting that \( u(t) \) is to be given by

\[ u = G \hat{x} \]

and

\[ \dot{\hat{x}} = (A+BG-SH_s)\hat{x} + SY_s \]

Equations 4.5-60 and 4.5-61 now form the description of the full controller with \( x_s \) the input and \( u \) the output.

4.5.8 Kalman Filter Solution Process

The numerical process for computing the filter matrix \( S \) is given by:

1. Form the \( 2n_x \) by \( 2n_x \) matrix of system and adjoined equations given by the left hand side of Equation 4.5-58.

2. Compute the eigenvalues of the matrix formed. If any of the eigenvalues have zero real parts, this is an indication that the system is unobservable and that no solution exists. (See Appendix A for computational details).

3. Partition the eigenvalues into two groups with all the eigenvalues with positive real parts in the first group.

4. Compute the eigenvectors (or real combinations of eigenvectors in the case of complex conjugate eigenvalues) for each eigenvalue in the first group. (See Appendix A for computational details).

5. Partition the matrix computed into \( W_{11} \) and \( W_{21} \).

6. Solve for \( W_{11}^{-1} H_s' C_v^{-1} \) with a standard linear equation solver routine. Should \( W_{11} \) be singular (or badly conditioned), this indicates that either the original system had an unstable un-
observable mode or that the rare event that the matrix formed in step 1 was undiagonlizable occurred. With the calculation process used, multiple eigenvalues with independent eigenvectors will not cause the method to fail except in extremely rare cases.

7. Compute $S$ as the product of $W_{21}$ with the above solution.

4.5.9 System Eigenvalues Using Kalman Filter

As in the case of the gain matrix calculation where the eigenvalues (obtained by partitioning) with negative real parts were the optimal closed loop eigenvalues for the system using the computed feedback matrix, the eigenvalues computed in the solution process for the Kalman filter have significance.

Using Equations 4.5-60 and 4.5-61 as the description of the full Kalman filter/controller and the original system equations given in 4.5-4 and 4.5-5, one obtains the equations for the total closed loop system as

\[
\begin{bmatrix}
\dot{x} \\
\dot{\hat{x}}
\end{bmatrix} =
\begin{bmatrix}
A & BG \\
SH_s & A+BG-SH_s
\end{bmatrix}
\begin{bmatrix}
x \\
\hat{x}
\end{bmatrix}
\]

4.5-62

Consider now a transformation $J$ where

\[
J = \begin{bmatrix}
I & 0 \\
I & I
\end{bmatrix}
\]

4.5-63

and where the I's are identity matrices of order $n_x$.

Then
which indicates that the total closed loop system has eigenvalues corresponding to \((A+BG)\) which are the eigenvalues computed during the calculation of the optimal gain matrix and corresponding to \((A-SH_S)\). It will now be shown that these eigenvalues are the ones computed during the calculation of the Kalman filter. From Equation 4.5-58

\[
-j^{-1} \begin{bmatrix}
A & BG \\
SA_S & A+BG - SH_S
\end{bmatrix} j
\]

\[
= \begin{bmatrix}
A+BG & BG \\
0 & A - SH_S
\end{bmatrix}
\] 4.5-64

which postmultiplying by \(W_{11}^{-1}\) yields

\[
-A' W_{11} + H_s' C_v^{-1} H_s W_{21} = W_{11} \Lambda
\] 4.5-65

Since \(P\) from Equation 4.5-56 is symmetric and equal to \(W_{21} W_{11}^{-1}\) the use of Equation 4.5-59 yields

\[
A' - H_s' S' = W_{11} (-\Lambda) W_{11}^{-1}
\]

which indicates that the negative of the eigenvalues calculated in the solution process for the \(S\) matrix are indeed the eigenvalues of \(A-SH_S\) since eigenvalues are invariant under transformation. Thus the \(2n_x\) eigenvalues of the total closed loop system are the eigenvalues calculated as part of the gain matrix and optimal filter solution process.
By more manipulation the eigenvectors for the system described in Equation 4.5-62 can be described in terms of the $W_{11}$ matrices (and inverse) calculated for both the gain matrix and Kalman filter. No attempt is made to exploit this information as the real subsequent analysis hinges on a reduced controller operating with the nonlinear system.

4.5.10 Controller Formation, Adjustment, and Reduction

The formation of the controller is straightforward when no initial system reduction took place. That is, from equations 4.5-60 and 4.5-61, the controller input is $Y_s$, the output is the actuator signal $u$, and the representative block diagram given in Figure 51.

Now the above controller is of the same order ($n_x$) as the original system description. Since this controller is now just another linear dynamic system, it is natural to ask if a lower order approximation can be made. The input $Y_s$ and output $u$ would have to remain the same but the dynamics describing $\hat{x}$ would be reduced. Section 4.5.11 gives the theory and calculation necessary to reduce this system. For now it suffices to state that a new reduced system of the form shown in Figure 52 results.

Note that in Figure 52 the input and output have not changed. The matrices $S_K$, $G_K$, $A_K$, are now of reduced dimensions ($z$ is not as large as $\hat{x}$) and a new block represented by $F_K$ is added. This is a controller feedforward block and represents a direct gain from the measurements (inputs to the controller) to the control signal (the output from the controller). Intuitively this addition is needed in that when fast dynamics are ignored, their effect is essentially an instantaneous response to the input. Also, the classical methods in control design allow a feedforward controller (i.e. a simple feedback gain) so that this reduction process that results in the $F_K$ term seems most reasonable.
Figure 51. Full Controller Block Diagram

Figure 52. Reduced Controller Block Diagram
Controller formation and reduction for the case when initial system reduction took place is more complicated. If one proceeds in a logical manner for the reduction of order in the initial description, a feedforward term in the expression for the sensor output $Y_s$ results, that is

$$Y_s = H_s x + D_s u$$

In block diagram form, the initial system (reduced) appears in Figure 53.

This term due to $D_s$ does not have any effect during the calculation of the optimal gain matrix and can be ignored during the calculation of the optimal filter. That is, the optimal filter is predicted on an input $H_s x$ which is now really $(Y_s - D_s u)$. Thus to form a controller with input $Y_s$ and output $u$, one has to subtract the $D_s u$ term so that the correct input to the Kalman filter results. This is shown in Figure 54.

The total controller is now the dynamics between points P1 and P2. Several alternatives now exist for the reduction of this controller. Since the feedback term involving $D_s$ has no dynamics associated, order reduction can be accomplished either before or after simplification by elimination of the feedback path. Elimination before results in a system shown in Figure 55.

This system is now just like the one shown in Figure 51 except for the extra term in the system matrix and can be similarly reduced. Another approach would be to take the dynamic system between points P2 and P3 in Figure 54 which is now just that of Figure 51 and reduce it to obtain the system shown in Figure 52. If this is done, and the reduced system substituted between points P2 and P3 in Figure 56, the following block diagram results.
Figure 53. Reduced System Block Diagram

Figure 54. Block Diagram of Controller When Initial System Had Feedforward Term
Figure 55. Total Controller Incorporating Static Feedback

Figure 56. Reduced Controller Before Static Feedback Elimination
In the block diagram of Figure 56 there is both algebraic feed-forward and feedback that must be accounted for. Noting that for the case shown in Figure 56.

\[ u = F_K (Y_s - D_s u) + G_K z \] 4.5-69

Then one can solve for \( u \) as

\[ u = (I + F_K D_s)^{-1}(F_K Y_s + G_K z) \] 4.5-70

providing the inverse exists (an assumption rarely violated).

From this one can define a modified \( F_K \) and \( G_K \) as

\[ F_K = (I + F_K D_s)^{-1}F_K \] 4.5-71
\[ G_K = (I + F_K D_s)^{-1}G_K \] 4.5-72

with

\[ u = F_K Y_s + G_K z. \] 4.5-73

The expression for the dynamic portion then becomes

\[ \dot{z} = A_K z + S_R (Y_s - D_s u) \] 4.5-74

which through the use of Equation 4.5-73 becomes

\[ \dot{z} = A_K z + S_R Y_s - S_R D_s F_K Y_s - S_R D_s G_K \] 4.5-75
\[ = (A_K - S_R D_s G_K) z + (S_R - S_R D_s F_K) Y_s \]

which indicate the modified \( A_K \) and \( S_K \) required to eliminate the static feedback. By using this second technique, the linear analysis of the resulting system becomes simple as
For this reason and because little advantage is seen for either technique over the other, the second method is the one implemented.

### 4.5.11 Linear System Reduction Theory

The problem of interest is to approximate a high order linear dynamic system by one of lower order in such a manner that the output responses due to various inputs are "close". The value of a low order approximation lies in the reduced computational and storage requirements for analysis and design and in the reduced complexity for implementation. Consider the linear description in the normal form,

\[
\dot{x} = Ax + Bu
\]

\[
y = Hx
\]

Where \(X\) is a \(n_x\) dimensional state vector, \(u\) is a \(n_u\) dimensional control vector and \(Y\) is a \(n_s\) dimensional measurement vector.

The lower order approximation sought is of the form

\[
\dot{z} = A_R z + B_R u
\]

\[
y = H_R z + D_R u
\]

where \(u\) and \(y\) are as defined above and \(z\) is a \(n_R\) dimensional reduced state vector with

\[
n_R \leq n_x
\]

This description differs from many reported in the literature in that the feedforward term accounted for in \(D_R\) is permitted. In some cases this may be a disadvantage but for most, especially for the simplification of controllers, it leads to a natural and appealing reduction.
The proposed approach is a classical one of retaining the lowest frequency modes and neglecting the dynamics associated with the higher stable modes.

Consider a transformation $T$ where $T$ is nonsingular and

\[ x = T w \]  

and

\[ T^{-1} \Lambda T = \Lambda = \begin{bmatrix} \Lambda_L & 0 \\ 0 & \Lambda_H \end{bmatrix} \]

Where $\Lambda$ is block diagonal with 1 by 1 blocks for real eigenvalues and 2 by 2 blocks for complex conjugate pairs of eigenvalues. For this discussion and for the implementation, it is assumed that $A$ is diagonalizable (any multiple eigenvalues have as many independent eigenvectors). Further it is assumed that $\Lambda$ is partitioned into $\Lambda_L$ and $\Lambda_H$ where all unstable and the lowest magnitude stable eigenvalues are in $\Lambda_L$ and the large magnitude stable eigenvalues and in $\Lambda_H$.

The resulting equations for a similarly partitioned $w$ are

\[
\begin{bmatrix}
\dot{w}_L \\
\dot{w}_H
\end{bmatrix} =
\begin{bmatrix}
\Lambda_L & 0 \\
0 & \Lambda_H
\end{bmatrix}
\begin{bmatrix}
w_L \\
w_H
\end{bmatrix}
+ \begin{bmatrix}
(T^{-1} B)_L \\
(T^{-1} B)_H
\end{bmatrix} u
\]

\[ y = [(H T)_L (H T)_H] \begin{bmatrix} w_L \\ w_H \end{bmatrix} \]

To neglect the dynamics associated with $w_H$ is to assume that $w_H$ responds instantaneously to any input. Thus $\dot{w}_H$ should be zero resulting in
\[
\dot{w}_H = \Lambda_H w_H + (T^{-1} B)_H u = 0 \quad 4.5-86
\]
\[
w_H = -\Lambda_H^{-1} (T^{-1} B)_H u \quad 4.5-87
\]
which is the algebraic relation desired. Equations 4.5-84 and 4.5-85 can then be written eliminating \(w_H\) as
\[
\dot{w}_L = \Lambda_L w_L + (T^{-1} B)_L u \quad 4.5-88
\]
\[
y = (H^T)_L w_L - (H^T)_H \Lambda_H^{-1} (T^{-1} B)_H u \quad 4.5-89
\]
with the terms identified as
\[
A_R = \Lambda_L \quad 4.5-90
\]
\[
B_R = (T^{-1} B)_L \quad 4.5-91
\]
\[
H_R = (H^T)_L \quad 4.5-92
\]
\[
D_R = -(H^T)_H \Lambda_H^{-1} (T^{-1} B)_H \quad 4.5-93
\]
The needed assumption is that \(\Lambda_H^{-1}\) exists which will be the case when \(\Lambda_H\) contains large stable eigenvalues. Note also that \(n_R\) can be prespecified as long as \(n_R\) is greater than the number of unstable eigenvalues. Further, it may be necessary to adjust \(n_R\) one integer less to insure that \(\Lambda_L\) is partitioned such that both of complex conjugate eigenvalues are included or excluded.

For this reduction technique, the reduced model is asymptotically correct for any input level. As the eigenvalues in \(\Lambda_H\) become separated from those in \(\Lambda_L\), the approximation naturally becomes more exact.

4.5.12 Reduction Calculation Sequence

The numerical process for computing the reduced linear system consists of:
1. Compute the eigenvalues of the full \( A \) matrix.

2. Sort the eigenvalues according to real parts with most positive at the top. Count unstable eigenvalues to insure retention.

3. Compute eigenvectors for sorted list.

4. Compute \( T^{-1} B \) and \( H T \) and partition.

5. Set \( A_R \) as block diagonal matrix mode from computed eigenvalues at top of list.

6. Set \( B_R \) and \( H_R \) as the top partitions in \( T^{-1} B \) and \( H T \) respectively.

7. Compute \( D_R = -(H T)_H A_H^{-1} (T^{-1} B)_H \)

4.5.13 Controller Use In Simulation

The controller designed is returned to the simulation program as a linear system described by the four matrices \( F_K, A_K, G_K, \) and \( S_K \). It must be remembered, however, that all the design analysis was performed about an operating point defined by \( u_0 \) and \( y_0 \). For a total controller, these quantities must be added back in. A total controller block diagram is thus given in Figure 57.

If several controllers are designed around several operating points, it may be necessary to "gain schedule" by changing controllers and set points as a function of operating point measured or commanded. These and other decisions on the value of the designed controller must now be based on the results of the simulation.

4.6 WARNING MESSAGES

One or more of the following warning messages will occur if the program encounters difficulty in interpreting analysis instructions or performing an analysis. These messages will be preceded by: ***WARNING***.

The symbols xxx, zzz, or nnn are used to indicate phrases from the analysis description that are included as part of the warning message. The following messages are listed in alphabetical order:
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM, VOLUME II

C L WEST, B R UMMEL, R F YURCZYK

AFWAL-TR-80-3014-VOL-1
APPENDIX N

EASIEST EXAMPLE

This appendix presents a supplementary ejection seat analysis example. This example utilizes the following components which were not included in the ejection seat simulation example in Section VI.

- AE  Airplane
- CS  Airplane control surfaces
- DR  Dart
- AP  Aerodynamic plate

A simplified thrust vector control system is also included in this model.
MODEL DESC * EASIEST EXAMPLE WITH FLUIDIC THRUST VECTOR CONTROL

LOCATION=010 AB

LOCATION=164 FORT

ADD VARIABLES=NIFLAG

ADD PARAMETERS=CTTIME

FORTRAN STATEMENTS

IF (TIME,NF,CTTIME) NIFLAG=1

LOCATION=134 CT

INPUT(6SP+6SP, UST+UST, EST+EST, WST+WST)

AE (SP+AP, UAP+UAP, EAP+EAP, WAP+WAP)

LOCATION=138 DA

INPUT(6SP+6SP, UST+UST, EST+EST, WST+WST)

AE (SP+AP, UAP+UAP, EAP+EAP, WAP+WAP)

LOCATION=132 BD

INPUT(6SP+6SP, UST+UST, EST+EST, WST+WST)

AE (SP+AP, UAP+UAP, EAP+EAP, WAP+WAP)

LOCATION=136 AP

INPUT(6SP+6SP, UST+UST, EST+EST, WST+WST)

AE (SP+AP, UAP+UAP, EAP+EAP, WAP+WAP)

LOCATION=153 RL

INPUT(6SP+6SP, UST+UST, EST+EST, WST+WST)

AE (SP+AP, UAP+UAP, EAP+EAP, WAP+WAP)

LOCATION=152 CB

INPUT(6SP+6SP, UST+UST, EST+EST, WST+WST)

AE (SP+AP, UAP+UAP, EAP+EAP, WAP+WAP)

LOCATION=153 AE

INPUT(IL+1, CT+1, DR+1)

LOCATION=150 A6

INPUT(6SP+6SP, UST+UST, EST+EST, WST+WST)

AE (SP+AP, UAP+UAP, EAP+EAP, WAP+WAP)

LOCATION=153 E6

INPUT(6S+6S, DR+1, RL+1, AS+1)

AP (E6)

LOCATION=153 FORT

INPUT=ROKER, WSTR, 82 IN

ADD VARIABLES=KI, XIOUT

ADD PARAMETERS=KI

FORTRAN STATEMENTS

**** RATES AS MEASURED BY THE FVVC HARDWARE ****

C IC

S1 * POLE ASSOCIATED WITH PSEUDO INTEGRATOR

C IC

G1 * PSEUDO INTEGRATOR DC GAIN

C XDOT * PSEUDO INTEGRATOR RATE (DEG/SEC)

C ZS * PSEUDO INTEGRATOR STATE (DEG)

C WTSR * LENS PITCH RATE (DEG/SEC)

C (IF XFST.NE.83)

.G1 = (1+ALORI1)+XORT(400, 01, 01, 31/30) + (1+ALORI1)+XORT(11, 11, 11)

ARARM + BX + BY + BX + BX

C XDOT $1 = XORT(1, 11) + XORT(12, 31) + XIOUT

SIG1 = XIOUT + XIOUT + XIOUT + XIOUT

C IBYOUT = BI

LOCATION=151 IN

INPUT=FORT(1, XDOT) B1

C LOCATION=152 FORT

INPUT=N1, XIOUT, RODER, 82 IN, 82 ITS, WSTR

ADD VARIABLES=1200, G3, G3

ADD PARAMETERS=K1, G3, G3, G3, X3, X3, K1

FORTRAN STATEMENTS

C **** SUSTAINER ROCKET NOZZLE DEFLECTION ****

C IC

AI * ZERO ASSOCIATED WITH PSEUDO INTEGRATOR

C IC

B2 * POLE ASSOCIATED WITH PSEUDO INTEGRATOR

C IC

BS * RATE GAIN

C IC

SR * ATTITUDE GAIN

C EQC * CONTROL SYSTEM DC GAIN

C XDOT * SUSTAINER ROCKET DEFLECTION RATE (DEG/SEC)

C ZS * ITS = SUSTAINER ROCKET DEFLECTION (DEG)

C XIOUT * PITCH ANGLE (DEG)
----- PARAMETERS FOR COMPONENT "CT" (CATAPULT) ----- 
CTTIME= .406
CACP= 85.0 - 9.0
UC= 348.0
CUT= 18.9
PA CT= 18.9
PT CT= 18.9
CBFCT= 8533
C TF= 000000
C T= 000000
C C= 000000
C CT= 000000
C C= 000000
C CT= 000000
CT= 000000
C T= 000000
PA CT= 18.9
----- PARAMETERS FOR COMPONENT "SR" (SUSTAINER ROCKET) ----- 
SRTIM= 32.0
SCH= 27.0 - 1.0
Sa= 0.0 - 0.0
Ph= 0.0 - 0.0
Trade= 0.0
--- Parameters for component "DF" (DART) ---
DAP= 16.6 - 0.0
--- Parameters for component "AP" (Attached Plate) ---
APERS= 0.0
--- Parameters for component "RL" (Rails) ---
RLL= 0.0 - 0.0
--- Parameters for component "CS" (control surfaces) ---
CSCS= 0.0
--- Parameters for component "AE" (Airplane) ---
AE= 0.0
--- Parameters for component "AS" (Aerodynamics) ---
AS= 0.0
--- Parameters for component "BB" (beat) ---
BB= 0.0
--- Parameters for the thrust vector control ---
TH= 100 4 92X120 836=1 1018=12
### Sustainer Rocket Off at Time = 0.608 sec

#### Integrator Step Size Limiting Counts

<table>
<thead>
<tr>
<th>TIME</th>
<th>0.608</th>
<th>1.056</th>
<th>1.504</th>
<th>1.952</th>
<th>2.400</th>
<th>2.848</th>
<th>3.296</th>
<th>3.744</th>
<th>4.192</th>
<th>4.640</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
</tr>
<tr>
<td>ASCII</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
</tr>
<tr>
<td>ASCII</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
</tr>
<tr>
<td>ASCII</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
</tr>
<tr>
<td>ASCII</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
</tr>
<tr>
<td>ASCII</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
</tr>
<tr>
<td>ASCII</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
</tr>
<tr>
<td>ASCII</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
<td>1073</td>
</tr>
</tbody>
</table>

---

CPU seconds were required for the previous analysis: 23.9120
LMED
8