AIR-TO-AIR COMBAT MODEL
PROGRAM & APPENDICES
TECHNICAL DETAILS

November 1967

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

DEPUTY FOR DEVELOPMENT PLANNING
AERONAUTICAL SYSTEMS DIVISION
WRIGHT-PATTERSON AFB, OHIO
ATAC-2: Single Search and Double Search

Terrell J. Harris
Barton J. Jacobs
Walter J. Strauss

November 1967

Contract Number AF33(615)-3707
Project Number 5101-57

prepared for

[Signature]
Systems Engineering Group (ASD)
Air Force Systems Command
United States Air Force
Wright-Patterson Air Force Base, Ohio 45433

The Table of Contents and page numbers of the original edition are preserved here.
This report is published in four volumes. Volume I, Model Description, presents an overall view of the model and its two major submodels, the ENGAGEMENT Model and DATA PROCESSING Model. Volume II, Model Development, contains the rationale for the development and discussion of details, together with the derivations of all equations. Flow charts and program listings appear in Volume III, Program. Volume IV, Appendices, contains discussions of certain model concepts in detail.

The entire report is UNCLASSIFIED.

This report supersedes the original ATAC-2 document [Ref. 1]. The many changes and modifications made in the evolutionary development of the model, based on the analysis of many computer runs, have rendered the earlier version outdated. The program of the model as reported here was used for production runs in June, 1967.

Certain modifications which allow either aircraft to detect initially are reported separately in the document "Fighter Vs. Fighter Combat: ATAC-2 Model: Double Search," [Ref. 2].
ATAC-2 is a simulation model designed to help evaluate fighters in air-to-air combat. The model treats the one vs. one dogfight which arises from a random search situation. Both aircraft in the combat are (usually) aggressive. The two principal outputs from the model are the probability a given aircraft is killed in the fight and the expected number of enemy aircraft an aircraft kills over its useful life. Combat is restricted to a fixed altitude. The maneuvers are dynamic in that each aircraft responds to the situation at each moment in a duel depending on the information it has about an opponent's activities.

Inputs include, for each aircraft, search and tracking radar characteristics, passive radar sensors, optical capability, IFF, energy-maneuverability data, weapon loadings, weapon characteristics, and weapon kill probabilities.

The rationale for the model specifics are presented. Flow charts and program listings are included. The model has been run repeatedly on an IBM 7094.
ACKNOWLEDGMENT

It is a pleasure to acknowledge the technical contribution of the personnel of the Directorate of Operations Research with whom and for whom this model was developed. In particular, it is noted that many helpful comments, suggestions, and criticisms were made by Lawrence Boyd, Duane Dunlap, Walter Grady, Thomas Haney, Lt. Kenneth Pierscalla, and Ralph Schwab.

It is also a pleasure to acknowledge the many suggestions and ideas of our colleagues at CSA. In particular, it is worthy to single out Thomas Caywood, James Cerone, James Hall, Thomas Morton, James Rust, Donald Schiller, and Ronald Swanson.

The model was programmed by Jeannette Filsinger and Pet Young of the Systems Engineering Group, ASD, at Wright-Patterson Air Force Base. Their patience and response to the frequent changes and modifications made in the flow charts are appreciated.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE</td>
<td>i</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>iii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
</tbody>
</table>

## VOLUME I - MODEL DESCRIPTION

1. INTRODUCTION                      | 1    |
2. ENGAGEMENT Model                 | 4    |
   1.2.1 Initiation                  | 7    |
   1.2.2 Maneuvers                   | 8    |
   1.2.3 Firing Conditions           | 10   |
   1.2.4 Termination                | 11   |
3. DATA PROCESSING Model            | 11   |
4. Inputs                           | 12   |
   1.4.1 Aircraft Characteristics   | 12   |
   1.4.2 Sensors                    | 13   |
   1.4.3 Weapons                    | 15   |
   1.4.4 Miscellaneous              | 15   |
5. Output of ENGAGEMENT Model       | 17   |
6. Output of DATA PROCESSING Model  | 18   |
7. Example of an ATAC-2 Run         | 20   |

## VOLUME II - MODEL DEVELOPMENT

2. MODEL CHARACTERISTICS AND ASSUMPTIONS | 1    |
3. POINTS OF EMPHASIS IN ATAC-2        | 2    |
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Engagement Model Development</td>
<td>4</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>4</td>
</tr>
<tr>
<td>4.1.1 Layout of Development</td>
<td>5</td>
</tr>
<tr>
<td>4.1.2 Symbols</td>
<td>7</td>
</tr>
<tr>
<td>4.1.3 Coordinate Systems</td>
<td>8</td>
</tr>
<tr>
<td>4.2 Search</td>
<td>9</td>
</tr>
<tr>
<td>4.3 Engagement Geometry</td>
<td>16</td>
</tr>
<tr>
<td>4.3.1 Equations of Motion</td>
<td>20</td>
</tr>
<tr>
<td>4.3.2 Equation Close In</td>
<td>25</td>
</tr>
<tr>
<td>4.4 Weapon Firings</td>
<td>26</td>
</tr>
<tr>
<td>4.4.1 Launch Envelopes</td>
<td>27</td>
</tr>
<tr>
<td>4.4.2 Lethality</td>
<td>29</td>
</tr>
<tr>
<td>4.4.3 IFF</td>
<td>31</td>
</tr>
<tr>
<td>4.4.4 Oxygen Debt</td>
<td>31</td>
</tr>
<tr>
<td>4.4.5 Firing Rates</td>
<td>32</td>
</tr>
<tr>
<td>4.4.6 Tail Guns</td>
<td>32</td>
</tr>
<tr>
<td>4.5 Attacker’s Course</td>
<td>33</td>
</tr>
<tr>
<td>4.5.1 DEL Pursuit Course</td>
<td>33</td>
</tr>
<tr>
<td>4.5.2 Preferred Firing Positions (R*) and Closing Speed Doctrine</td>
<td>36</td>
</tr>
<tr>
<td>4.6 Turning Rate ((\dot{\theta})) and Acceleration ((\dot{a})) Tactics</td>
<td>37</td>
</tr>
<tr>
<td>4.6.1 Doctrine’s Desired (\dot{\theta}) and (\dot{a})</td>
<td>37</td>
</tr>
<tr>
<td>4.6.1.1 Turning Rate</td>
<td>37</td>
</tr>
<tr>
<td>4.6.1.2 Criterion for Acceleration</td>
<td>38</td>
</tr>
<tr>
<td>4.6.2 Limitations on Maneuvers</td>
<td>39</td>
</tr>
<tr>
<td>4.6.2.1 Specific Power, (P_S)</td>
<td>39</td>
</tr>
<tr>
<td>4.6.2.2 General Rule for Handling (P_S) Limitations</td>
<td>41</td>
</tr>
<tr>
<td>4.6.2.3 Other Limitations</td>
<td>43</td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS (CONT.)

<table>
<thead>
<tr>
<th>Section</th>
<th>Details</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7</td>
<td>Details of Tactics</td>
<td></td>
</tr>
<tr>
<td>4.7.1</td>
<td>Pursuit Doctrine</td>
<td>44</td>
</tr>
<tr>
<td>4.7.1.1</td>
<td>Steady State</td>
<td>52</td>
</tr>
<tr>
<td>4.7.2</td>
<td>Non-Pursuit Doctrine with Good Information</td>
<td>54</td>
</tr>
<tr>
<td>4.7.3</td>
<td>Tactics with Poor Information</td>
<td>55</td>
</tr>
<tr>
<td>4.7.4</td>
<td>Passive Information</td>
<td>55</td>
</tr>
<tr>
<td>4.7.5</td>
<td>Lack of Active Information</td>
<td>57</td>
</tr>
<tr>
<td>4.7.6</td>
<td>Evasion</td>
<td>58</td>
</tr>
<tr>
<td>5.</td>
<td>ENGAGEMENT MODEL FLOW CHART DISCUSSION</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>60</td>
</tr>
<tr>
<td>5.2</td>
<td>General Conventions</td>
<td></td>
</tr>
<tr>
<td>5.2.1</td>
<td>Diagrams of Angles</td>
<td>61</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Flow Chart Conventions</td>
<td>68</td>
</tr>
<tr>
<td>5.3</td>
<td>The EXECUTIVE Routine</td>
<td>69</td>
</tr>
<tr>
<td>5.4</td>
<td>The GRID PREPARATION Routine</td>
<td>71</td>
</tr>
<tr>
<td>5.5</td>
<td>The COMBAT Routine</td>
<td>80</td>
</tr>
<tr>
<td>5.5.1</td>
<td>The GRID Routine</td>
<td>82</td>
</tr>
<tr>
<td>5.5.2</td>
<td>The INITIALIZE FLIGHT Routine</td>
<td>87</td>
</tr>
<tr>
<td>5.5.3</td>
<td>NAVIGATIONAL SYSTEMS</td>
<td>89</td>
</tr>
<tr>
<td>5.5.3.1</td>
<td>The INFO Routine</td>
<td>94</td>
</tr>
<tr>
<td>5.5.3.2</td>
<td>The ACTIVE Routine</td>
<td>95</td>
</tr>
<tr>
<td>5.5.3.3</td>
<td>The PASSIVE Routine</td>
<td>95</td>
</tr>
<tr>
<td>5.5.3.4</td>
<td>The R((r_1)) Function</td>
<td>97</td>
</tr>
<tr>
<td>5.5.3.5</td>
<td>The AWARE Routine</td>
<td>98</td>
</tr>
<tr>
<td>5.5.3.6</td>
<td>The G_1(x) Routine</td>
<td>98</td>
</tr>
<tr>
<td>5.5.3.7</td>
<td>The g_1(x) Routine</td>
<td>99</td>
</tr>
<tr>
<td>5.5.3.8</td>
<td>The P_i(V_1, (\delta_i)) Function</td>
<td>99</td>
</tr>
<tr>
<td>5.5.3.9</td>
<td>The (S(x)) Function</td>
<td>99</td>
</tr>
<tr>
<td>5.5.4</td>
<td>ADVANCE RELATIVE COORDINATES</td>
<td>99</td>
</tr>
<tr>
<td>5.5.5</td>
<td>TRANSFORM TO INERTIAL COORDINATES Routine</td>
<td>101</td>
</tr>
<tr>
<td>5.5.6</td>
<td>The FIND G_1 Routine</td>
<td>103</td>
</tr>
<tr>
<td>5.5.7</td>
<td>CHECK WEAPONS Routine</td>
<td>104</td>
</tr>
<tr>
<td>5.5.7.1</td>
<td>The (R_{MIS}, R'_{MIS}) Routine</td>
<td>107</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (CONT.)

5.5.8 The PRINT Routine.................................................. 109
5.5.9 The OVER Routine................................................... 110
5.5.10 The RESULTS Routine............................................. 110
5.5.11 General Purpose Routines......................................... 112

5.5.11.1 The PV(x) Function............................................. 112
5.5.11.2 The SGN(x) Function........................................... 112
5.5.11.3 The Q(x) Function............................................. 113

6. DATA PROCESSING MODEL DISCUSSION.................................. 114

6.1 Introduction......................................................... 114
6.2 Engagement Probabilities of Kill.................................... 115
6.3 Encounter Probabilities............................................... 121
6.4 A Measure of Effectiveness........................................... 129
6.5 Computational Considerations........................................ 131

7. DEFINITIONS............................................................. 7-1

7.1 Introduction.......................................................... 7-1
7.2 Definition List....................................................... 7-1
7.3 Input Considerations.................................................. 7-18

7.3.1 Parameter Guidance................................................ 7-18
7.3.2 Model Logic.......................................................... 7-20
7.3.3 Flow Chart Restrictions.......................................... 7-21
7.3.4 Program.............................................................. 7-22

VOLUME III - PROGRAM

7. DEFINITIONS (Repeated from Volume II).................................. 7-1

7.1 Introduction.......................................................... 7-1
7.2 Definition List....................................................... 7-1
7.3 Input Considerations.................................................. 7-18

7.3.1 Parameter Guidance................................................ 7-18
7.3.2 Model Logic.......................................................... 7-20
7.3.3 Flow Chart Restrictions.......................................... 7-21
7.3.4 Program.............................................................. 7-22

8. FLOW CHARTS........................................................... 23
### TABLE OF CONTENTS (CONT.)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9. PROGRAM LISTING</td>
<td>46</td>
</tr>
<tr>
<td>10. AN EXAMPLE OF INPUT AND OUTPUT</td>
<td>138</td>
</tr>
</tbody>
</table>

#### VOLUME IV - APPENDICES

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPENDIX A - PROBABILITY OF DETECTION</td>
<td>1</td>
</tr>
<tr>
<td>A.1 Stationary Target</td>
<td>2</td>
</tr>
<tr>
<td>A.2 Non-Stationary Target</td>
<td>2</td>
</tr>
<tr>
<td>APPENDIX B - GEOMETRIC CONSIDERATIONS</td>
<td>5</td>
</tr>
<tr>
<td>B.1 Inertial vs. Relative Coordinates</td>
<td>5</td>
</tr>
<tr>
<td>B.2 Deriving Relative Coordinates from Inertial Coordinates</td>
<td>7</td>
</tr>
<tr>
<td>B.3 The Equations of Relative Motion</td>
<td>7</td>
</tr>
<tr>
<td>APPENDIX C - THE DEL PURSUIT COURSE</td>
<td>10</td>
</tr>
<tr>
<td>C.1 Rationale</td>
<td>10</td>
</tr>
<tr>
<td>C.2 Formulation</td>
<td>14</td>
</tr>
<tr>
<td>C.3 Implications of the DEL Pursuit Course</td>
<td>18</td>
</tr>
<tr>
<td>C.4 Derivation of $\Psi^*$</td>
<td>21</td>
</tr>
<tr>
<td>APPENDIX D - TURNING RATE</td>
<td>23</td>
</tr>
<tr>
<td>APPENDIX E - SPECIFIC POWER FUNCTION</td>
<td>26</td>
</tr>
<tr>
<td>APPENDIX F - STEADY STATE CONDITIONS</td>
<td>32</td>
</tr>
</tbody>
</table>
# List of Figures

## Volume I

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7-1</td>
<td>Plot of an Engagement</td>
<td>21</td>
</tr>
</tbody>
</table>

## Volume II

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1-1</td>
<td>ATAC-2 Master Flow</td>
<td>6</td>
</tr>
<tr>
<td>4.2-1</td>
<td>Typical Radar Pattern</td>
<td>10</td>
</tr>
<tr>
<td>4.2-2</td>
<td>Typical Optical Pattern</td>
<td>10</td>
</tr>
<tr>
<td>4.2-3</td>
<td>Bombers in Random Direction</td>
<td>12</td>
</tr>
<tr>
<td>4.2-4</td>
<td>Bombers in Same Direction</td>
<td>14</td>
</tr>
<tr>
<td>4.2-5</td>
<td>Results of Increasing Fighter Speed</td>
<td>14</td>
</tr>
<tr>
<td>4.2-6</td>
<td>Search Flow</td>
<td>15</td>
</tr>
<tr>
<td>4.2-7</td>
<td>Head On Case</td>
<td>17</td>
</tr>
<tr>
<td>4.2-8</td>
<td>Bomber Comes In from Behind</td>
<td>17</td>
</tr>
<tr>
<td>4.2-9</td>
<td>Fighter Detects While Bomber is in Rear</td>
<td>17</td>
</tr>
<tr>
<td>4.2-10</td>
<td>Grid For Given c</td>
<td>18</td>
</tr>
<tr>
<td>4.3-1</td>
<td>Defining Relative Positions</td>
<td>19</td>
</tr>
<tr>
<td>4.3-2</td>
<td>Establishing Inertial Position of Bomber</td>
<td>23</td>
</tr>
<tr>
<td>4.3-3</td>
<td>Establishing Inertial Position of Fighter</td>
<td>24</td>
</tr>
<tr>
<td>4.4-1</td>
<td>Weapon Firing Envelope</td>
<td>28</td>
</tr>
<tr>
<td>4.4-2</td>
<td>Handling Missiles with Varying Lethality</td>
<td>30</td>
</tr>
<tr>
<td>4.5-1</td>
<td>Decrease in Lag as Pursuer Swings Behind Target</td>
<td>34</td>
</tr>
<tr>
<td>4.6-1</td>
<td>Geometric Description of S(V')</td>
<td>40</td>
</tr>
<tr>
<td>4.6-2</td>
<td>Typical Specific Power Function</td>
<td>42</td>
</tr>
<tr>
<td>4.7-1</td>
<td>Tactical Cases (m states)</td>
<td>45</td>
</tr>
<tr>
<td>4.7-2</td>
<td>Steady State Geometry</td>
<td>53</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES (CONT'D)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7-3</td>
<td>Passive Receivers.</td>
<td>56</td>
</tr>
<tr>
<td>5.2-1</td>
<td>Line of Sight Segments</td>
<td>61</td>
</tr>
<tr>
<td>5.2-2</td>
<td>Defining Relative Positions.</td>
<td>62</td>
</tr>
<tr>
<td>5.2-3</td>
<td>Orientation of $\hat{\beta}_4$ with respect to the Inertial Coordinate System.</td>
<td>64</td>
</tr>
<tr>
<td>5.2-4</td>
<td>A/C Sensor Half-Angles</td>
<td>65</td>
</tr>
<tr>
<td>5.2-5</td>
<td>Orientation of $\phi_B$ for Firing Tables.</td>
<td>66</td>
</tr>
<tr>
<td>5.2-6</td>
<td>Definition of Angles in the GRID (X,Y) Coordinate System</td>
<td>67</td>
</tr>
<tr>
<td>5.4-1</td>
<td>Logical Basis For GRID PREPARATION and GRID Routines</td>
<td>73</td>
</tr>
<tr>
<td>5.4-2</td>
<td>GRID PREP Cases.</td>
<td>75</td>
</tr>
<tr>
<td>5.4-3</td>
<td>Determination of $\nu$</td>
<td>77</td>
</tr>
<tr>
<td>5.5-1</td>
<td>Determination of $R$ and $\alpha_F$ from Arc Gridpoint</td>
<td>84</td>
</tr>
<tr>
<td>5.5-2</td>
<td>Determination of $R$ and $\alpha_F$ Radial Gridpoint</td>
<td>86</td>
</tr>
<tr>
<td>5.5-3</td>
<td>Determination of $\phi_B$</td>
<td>87</td>
</tr>
<tr>
<td>5.5-4</td>
<td>Information ($k$) states</td>
<td>90</td>
</tr>
<tr>
<td>5.5-5</td>
<td>Determination of $\phi_B$ and $\phi_R$</td>
<td>100</td>
</tr>
<tr>
<td>6.3-1</td>
<td>Arc Portion Represented by $\epsilon$'s</td>
<td>125</td>
</tr>
</tbody>
</table>

### VOLUME IV

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1-1</td>
<td>Inertial and Relative Geometry</td>
<td>6</td>
</tr>
<tr>
<td>C.1-1</td>
<td>Defining Function of the DEL Pursuit Course</td>
<td>12</td>
</tr>
<tr>
<td>C.1-2</td>
<td>A DEL Pursuit Course</td>
<td>13</td>
</tr>
<tr>
<td>C.2-1</td>
<td>Lead and Lag Positions</td>
<td>15</td>
</tr>
<tr>
<td>C.2-2</td>
<td>Singular Cases</td>
<td>16</td>
</tr>
<tr>
<td>C.3-1</td>
<td>DEL Paths for Different Values of $\phi_{MAX}$</td>
<td>19</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>D-1</td>
<td>A Duel</td>
<td>24</td>
</tr>
<tr>
<td>E-1</td>
<td>A Specific Power Function</td>
<td>29</td>
</tr>
<tr>
<td>F-1</td>
<td>Steady State Parameters</td>
<td>34</td>
</tr>
<tr>
<td>F-2</td>
<td>Steady State Velocity as Function of Angle-off</td>
<td>35</td>
</tr>
<tr>
<td>F-3</td>
<td>Steady State Range as Function of Angle-off</td>
<td>36</td>
</tr>
</tbody>
</table>
SECTION 7
DEFINITIONS

7.1 Introduction

Section 7 is intended primarily for reference. Section 7.2 defines all symbols used in the flow charts and the text, although the symbols are also defined where introduced in the text. The units associated with a variable are included, both for clarification and for use in setting input values.

An "I" after a definition means that this variable is an input to the model. A "C" means the variable is internally computed. This can be useful when reading the flow charts, as some variables which abstractly seem to be "inputs" are actually computed from other values. Variables used only in the text for model discussion have a 1) after their definition. This list is repeated in Volume III, Section 7.

Also included in this list is the FORTRAN symbol associated with each variable where appropriate. These symbols do not include the subscripts of subscripted FORTRAN symbols nor do they include the arguments of FORTRAN functions.

Section 7.3 discusses some input restrictions.

7.2 Definition List
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>An index with values of 0 and 1 indicating respectively that the bomber was unaware or aware during an engagement. Also used in text for the area within which the bomber flies during the search. (C)</td>
<td>1A</td>
</tr>
<tr>
<td>a</td>
<td>The area swept out by the fighter's detection pattern during its search. ( \text{ft}^2 )</td>
<td></td>
</tr>
<tr>
<td>( a_i )</td>
<td>The acceleration of aircraft ( i ). ( \text{ft/} \text{sec}^2 ) (C)</td>
<td>A</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>A logical variable that takes values of YES and NO (or, equivalently, TRUE and FALSE) indicating whether or not an aircraft has active information from optical or detection radar. (C)</td>
<td>ACT</td>
</tr>
<tr>
<td>( a_{\text{DEC}}(i) )</td>
<td>The input deceleration of aircraft ( i ). ( a_{\text{DEC}}(i) ) must be input as a negative number. ( \text{ft/} \text{sec}^2 ) (I)</td>
<td>ADEC</td>
</tr>
<tr>
<td>( A_{\text{M}}(i) )</td>
<td>The input parameter of the decreasing lag course function of aircraft ( i ). This is the angle that aircraft ( i ) will try to lag by when its enemy is flying pure pursuit and ( \lambda_i = 0 ). (deg) (I)</td>
<td>AM</td>
</tr>
<tr>
<td>B</td>
<td>An index identifying the aircraft designated as &quot;bomber,&quot; ( B ) always equals 2. (I)</td>
<td>IBMR</td>
</tr>
<tr>
<td>( B_i )</td>
<td>An index with values 0, 1 indicating the pilot's sickness state. (C)</td>
<td>BINDEX</td>
</tr>
<tr>
<td>C</td>
<td>A flow chart symbol used to indicate the general maneuver of Circle.</td>
<td></td>
</tr>
<tr>
<td>CL</td>
<td>A flow chart symbol denoting the general Circle Lost maneuver; i.e., lost information.</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>The distance traveled by the bomber during the fighter's search. (ft) (I)</td>
<td>D</td>
</tr>
<tr>
<td>DIV</td>
<td>A temporary computation used in the Data Processing Model. (C)</td>
<td>DIV</td>
</tr>
</tbody>
</table>

1) Used in text only.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>D(k, i)</td>
<td>An index indicating the general maneuver to be performed by aircraft i when in information state k; 0 = evasive maneuver, 1 = aggressive maneuver, (I)</td>
<td>D</td>
</tr>
<tr>
<td>e</td>
<td>The base of the natural logarithm = 2.71828 ....</td>
<td>EXP</td>
</tr>
<tr>
<td>EKB, EKT</td>
<td>The expected number of bombers killed by a fighter over its useful life. (C)</td>
<td>EKB</td>
</tr>
<tr>
<td>EXF</td>
<td>The expected number of fighters killed by a bomber over its useful life. (C)</td>
<td>EXF</td>
</tr>
<tr>
<td>ESB</td>
<td>A temporary calculation of the expected number of sorties completed by the fighter. (C)</td>
<td>ESB</td>
</tr>
<tr>
<td>FC</td>
<td>A flow chart symbol used to indicate the general maneuver of Evade by Circle.</td>
<td></td>
</tr>
<tr>
<td>FL</td>
<td>A flow chart symbol used to indicate the general maneuver of Evade Linearly.</td>
<td></td>
</tr>
<tr>
<td>FS</td>
<td>Specific energy.  (^1) (ft)</td>
<td></td>
</tr>
<tr>
<td>FS(n)</td>
<td>The expected number of sorties completed in at most ( n ) attempts.  (^1)</td>
<td></td>
</tr>
<tr>
<td>ENV SW(MIS, i)</td>
<td>A switch which when ON, or TRUE, indicates that aircraft i has fired a weapon of type MIS; otherwise the weapon type has not been fired and the variable has a value of OFF, or FALSE. (C)</td>
<td>ENV SW</td>
</tr>
<tr>
<td>F</td>
<td>The segment ( Y^* ) times the ratio of the bomber's velocity to the relative velocity; ( F ) is used in the computation of ( P_0(t) ), (ft) (C)</td>
<td>FSMALL</td>
</tr>
<tr>
<td>F</td>
<td>An index identifying the aircraft designated &quot;fighter.&quot; ( F ) always equals 1. (C)</td>
<td>IFTR</td>
</tr>
<tr>
<td>f_1 , f_2</td>
<td>Oxygen flow leaving and returning to pilot's brain. Functional notation in text is made explicit in flow chart, (^1) (sec-(^1))</td>
<td></td>
</tr>
<tr>
<td>e_1 , e_2</td>
<td>An arbitrary number of g's pulled.  (^1)</td>
<td></td>
</tr>
<tr>
<td>G_1 , G_2 , G_3</td>
<td>The three levels of target's total g's for which the weapon envelopes are input. (I)</td>
<td>GT</td>
</tr>
<tr>
<td>GB(i)</td>
<td>An index with values of 0, 1 indicating respectively that the degree of oxygen debt of the pilot of aircraft i will not or will affect the maneuverability of his aircraft by limiting the g's of his aircraft. (I)</td>
<td>GB</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
<td>Fortran</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>$G_1$</td>
<td>The number of total g's being sustained by aircraft $i$. (C)</td>
<td>$G$</td>
</tr>
<tr>
<td>$G_1(V)$</td>
<td>The structural or aerodynamic limit of total g's for aircraft $i$ as a function of its velocity. (I)</td>
<td>GBIG</td>
</tr>
<tr>
<td>$g_1(V)$</td>
<td>The total g function of velocity for aircraft $i$ at which the specific power function is zero. Note if $g_1(V) &gt; G_1(V)$ then for that $V$ $g_1(V)$ is unattainable and $G_1(V)$ should be inputted. (I)</td>
<td>GMXT</td>
</tr>
<tr>
<td>$SNIS(i)$</td>
<td>The level of total g's for aircraft $i$ above which weapon type MIS cannot be fired from aircraft $i$. (I)</td>
<td>CMIS</td>
</tr>
<tr>
<td>$G_{p}(i)$</td>
<td>The maximum number of total g's that the pilot of aircraft $i$ is able to sustain. (I)</td>
<td>GP</td>
</tr>
<tr>
<td>h</td>
<td>The altitude of the simulated engagements; used as an identifier only. (ft) (I)</td>
<td>H</td>
</tr>
<tr>
<td>i, j</td>
<td>Indices that take on values of F and E and do not have the same value. These symbols always indicate an aircraft and nothing else. (C)</td>
<td>I, J</td>
</tr>
<tr>
<td>IA(i)</td>
<td>The aircraft that fired the $i$th weapon in an engagement. (C)</td>
<td>IAFIRI</td>
</tr>
<tr>
<td>ICAN(i)</td>
<td>An index with values 1, 0 indicating respectively that aircraft $i$'s firing of any weapon is or is not being delayed in an engagement so as to get a better position at the time of firing. (C)</td>
<td>ICAN</td>
</tr>
<tr>
<td>ID</td>
<td>Identifying titles of the combatants for printing purposes. (I)</td>
<td>ID</td>
</tr>
<tr>
<td>ISHIFT</td>
<td>An index with values 1, 2 and 3 indicating that the first, second or third value of $t_{D}(\cdot)$ is assigned to $At$. (C)</td>
<td>ISHIFT</td>
</tr>
<tr>
<td>ITEMP(i)</td>
<td>An index with values 1, 0 indicating respectively that aircraft $i$ does or does not have IFF. (C)</td>
<td>ITEMP</td>
</tr>
<tr>
<td>k</td>
<td>An index with values 1, 2, ..., 11 indicating the information state of an aircraft. (Also used throughout as an arbitrary index with integer values; when used as such it is defined in context.) (C)</td>
<td>K</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
<td>Fortran</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>$k_i$</td>
<td>The $k$ state of aircraft $i$.</td>
<td>KPRT</td>
</tr>
<tr>
<td>$X_T, X_B, X$</td>
<td>Symbols used to describe the slope of the DEL Pursuit Course function.</td>
<td></td>
</tr>
<tr>
<td>$k(\text{MIS}, i)$</td>
<td>A counter of the number of weapons of type MIS fired from aircraft $i$.</td>
<td>KOUNTR</td>
</tr>
<tr>
<td>$L$</td>
<td>One greater than the total number of weapons that may be fired by both aircraft; $L = 1 + k_i f_{\text{MIS}} N(\text{MIS}, i)$</td>
<td>LCAP</td>
</tr>
<tr>
<td>$L$</td>
<td>A flow chart symbol used to indicate a Linear course.</td>
<td>LLITL</td>
</tr>
<tr>
<td>$l$</td>
<td>An index giving the order in which weapons were fired in an engagement, $l &lt; L - 1$.</td>
<td></td>
</tr>
<tr>
<td>$m$</td>
<td>An index with values 1, 2 indicating whether firing by one or both aircraft is permitted. Used in $F_j^m$, $PK_j^m$, etc.</td>
<td>M1</td>
</tr>
<tr>
<td>$M(i)$</td>
<td>The MIS identification number of the $i$th weapon fired in an engagement.</td>
<td>MISTP1</td>
</tr>
<tr>
<td>MIS</td>
<td>The identification number assigned to a weapon type, MIS takes values of 1, 2, ..., $n_{\text{M}}(i)$.</td>
<td>MIS</td>
</tr>
<tr>
<td>$m_i$</td>
<td>An index identifying the position, velocity and information ($k$) state of aircraft $i$.</td>
<td>LMSTAT</td>
</tr>
<tr>
<td>$N$</td>
<td>The number of grid-points or points of initialization for each $e$.</td>
<td>N</td>
</tr>
<tr>
<td>$n$</td>
<td>An index with values 1, 2, ..., $N$ indicating the grid-point number under consideration. Also used in text for other purposes but always so identified.</td>
<td>IGRIDP</td>
</tr>
<tr>
<td>NUIND</td>
<td>An index with values of 0, 1, 2 indicating the mode of operation of the tracking radar of each aircraft.</td>
<td>NUIND</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
<td>Fortran</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>n_m(i)</td>
<td>The number of distinct weapon types carried by aircraft i. (I)</td>
<td>N M</td>
</tr>
<tr>
<td>N_u(c)</td>
<td>The number of grid-points in which the bomber was unaware for a given c. (C)</td>
<td>NU</td>
</tr>
<tr>
<td>N_e</td>
<td>The total number of c values used in an encounter. (C)</td>
<td>NEPS</td>
</tr>
<tr>
<td>n'</td>
<td>An index used to compute the variables x and y. (C)</td>
<td>NT</td>
</tr>
<tr>
<td>n^a</td>
<td>The maximum number of sorties by each aircraft, for the calculation of EKB and EKF. (I)</td>
<td>YN</td>
</tr>
<tr>
<td>N(MIS, i)</td>
<td>The total number of weapons of type MIS on aircraft i. (C)</td>
<td>NUMIS</td>
</tr>
<tr>
<td>O_i</td>
<td>A variable indicating the amount of &quot;oxygen debt&quot; of the pilot of aircraft i incurred by pulling g's over a period of time, (C)</td>
<td>OXDEBT</td>
</tr>
<tr>
<td>P</td>
<td>A flow chart symbol used to indicate the general maneuver of Pursuit.</td>
<td></td>
</tr>
<tr>
<td>PASSIVE</td>
<td>A logical variable that takes values of YES and NO or TRUE and FALSE indicating whether or not an aircraft has passive information. (C)</td>
<td>PASS</td>
</tr>
<tr>
<td>PKB</td>
<td>The probability for an encounter that the bomber is detected and killed. (C)</td>
<td>PKB</td>
</tr>
<tr>
<td>PKBGD</td>
<td>The probability that the bomber is killed given detection for the encounter. (C)</td>
<td>PKBGD</td>
</tr>
<tr>
<td>PKE</td>
<td>The probability that the bomber is detected and killed at or before the time it becomes aware for an encounter. (C)</td>
<td>PKE</td>
</tr>
<tr>
<td>PKF</td>
<td>The probability for an encounter that the fighter detects the bomber and the fighter is killed. (C)</td>
<td>PKF</td>
</tr>
<tr>
<td>PKFGD</td>
<td>The probability that the fighter is killed, given detection for the encounter. (C)</td>
<td>PKFGD</td>
</tr>
</tbody>
</table>

7-6
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKG</td>
<td>The probability that the bomber is killed after it becomes aware, given that it survived to the time of its awareness for an encounter. (C)</td>
<td>PKG</td>
</tr>
<tr>
<td>PKL</td>
<td>The probability that the bomber is killed after it becomes aware and it survived to the time of its awareness for an encounter. (C)</td>
<td>PKL</td>
</tr>
<tr>
<td>P_D</td>
<td>The probability of detection of the bomber. (C)</td>
<td>PDD</td>
</tr>
<tr>
<td>P_S</td>
<td>Specific power,1) (ft/sec). Also probability of survival of an aircraft.1)</td>
<td>PDD</td>
</tr>
<tr>
<td>P_SB</td>
<td>The probability for an encounter that the bomber survives. (C)</td>
<td>PSB</td>
</tr>
<tr>
<td>P_SF</td>
<td>The probability for an encounter that the fighter survives and detects the bomber and the bomber is aware of the fighter. (C)</td>
<td>PS</td>
</tr>
<tr>
<td>P_u</td>
<td>The probability that the bomber is unaware for an encounter. (C)</td>
<td>PUU</td>
</tr>
<tr>
<td>P(I)</td>
<td>The probability that the target is dead just after the Ith weapon hits its target in some engagement. (C)</td>
<td>P</td>
</tr>
<tr>
<td>p(I)</td>
<td>The probability an aircraft is killed by the Ith weapon only.1)</td>
<td>P</td>
</tr>
<tr>
<td>PS(I)</td>
<td>An index indicating the capability of the passive receiver of aircraft 1; 0 implies no capability, 1 implies the ability to detect the presence of another aircraft but not the position, and 2 implies the capability of 1 with the ability to distinguish the hemisphere of the source. (I)</td>
<td>IPS</td>
</tr>
<tr>
<td>P_1(J)</td>
<td>An index that when set to zero will require aircraft j to turn its tracking radar on for one time pulse only when launching a weapon. (I)</td>
<td>P1</td>
</tr>
<tr>
<td>P_2(I)</td>
<td>An index with values 1 and 0 indicating respectively that aircraft 1 has or has not activated its tracking radar. (I)</td>
<td>P2</td>
</tr>
<tr>
<td>PV(x)</td>
<td>A function that gives the principal value of its angular argument. (rad) (C)</td>
<td>PV</td>
</tr>
<tr>
<td>P(x, y)</td>
<td>Probability distribution of bomber's (x, y) coordinates during search.1)</td>
<td>P</td>
</tr>
</tbody>
</table>

7-7
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{U/D} )</td>
<td>The probability that the bomber is unaware, given that it is detected for an encounter. (C)</td>
<td></td>
</tr>
<tr>
<td>( P_D(e) )</td>
<td>The probability of detection of the bomber by the fighter for some ( e ). (C)</td>
<td></td>
</tr>
<tr>
<td>( P_U(e) )</td>
<td>The probability that the bomber is unaware of the fighter for some value of ( e ). (C)</td>
<td></td>
</tr>
<tr>
<td>( P_4(V, \dot{\theta}) )</td>
<td>The specific power function of aircraft ( i ) at velocity ( V ) and turning rate ( \dot{\theta} ) (ft/sec) (I)</td>
<td>PEEU</td>
</tr>
<tr>
<td>( PK_B^1 )</td>
<td>The probability that the bomber is killed, given that the bomber is detected and unaware for an encounter. (C)</td>
<td>PK2B</td>
</tr>
<tr>
<td>( PK_j^1 )</td>
<td>The encounter probability that aircraft ( j ) is killed, given that the bomber is detected and aware and that aircraft ( j ) does not fire. (C)</td>
<td>PKK</td>
</tr>
<tr>
<td>( PK_j^2 )</td>
<td>The encounter probability that aircraft ( j ) is killed, given that the bomber is detected and aware. (C)</td>
<td>PKK</td>
</tr>
<tr>
<td>( P_B(e) )</td>
<td>The probability for a given ( e ) that the bomber is killed, given that it is detected and unaware.</td>
<td>PZ</td>
</tr>
<tr>
<td>( P_j^1(e) )</td>
<td>The probability for a given ( e ) that aircraft ( j ) is killed, given that the bomber is detected, aware and that aircraft ( j ) does not fire. (C)</td>
<td>PK</td>
</tr>
<tr>
<td>( P_j^2(e) )</td>
<td>The probability for a given ( e ) that aircraft ( j ) is killed, given that the bomber is detected and aware. (C)</td>
<td>PK</td>
</tr>
<tr>
<td>( P_C(e) )</td>
<td>The probability that the bomber is killed at or before the time it becomes aware for some ( e ). (C)</td>
<td>PCCEPS</td>
</tr>
<tr>
<td>( P_B(e, n) )</td>
<td>The probability for a given ( e ) and grid-point ( n ) that the bomber is killed, given that it is detected and unaware. (C)</td>
<td>PPZ</td>
</tr>
<tr>
<td>( P_j^1(e, n) )</td>
<td>The probability for a given ( e ) and grid-point ( n ) that aircraft ( j ) is killed, given that the bomber is detected and aware and that ( j ) does not fire. (C)</td>
<td>PP</td>
</tr>
<tr>
<td>( P_j^2(e, n) )</td>
<td>The probability for a given ( e ) and grid-point ( n ) that aircraft ( j ) is killed, given the bomber is detected and aware. (C)</td>
<td>PP</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
<td>Fortran</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>( P_c(e, n) )</td>
<td>The probability that in an engagement the bomber is killed at or before the time it becomes aware. (C)</td>
<td>PCCEPN</td>
</tr>
<tr>
<td>( P_k(MI, i) )</td>
<td>The probability of kill of weapon type MIS on aircraft ( i ). (I)</td>
<td>PKP</td>
</tr>
<tr>
<td>( P_k(MI, i) )</td>
<td>The probability of kill of weapon type MIS on aircraft ( i ); set to the input value, ( P_k(MI, i) ), or to zero. (C)</td>
<td>PK</td>
</tr>
<tr>
<td>( Q_B )</td>
<td>The quadrant of the point ( B ) on the fighter's detection pattern. (C)</td>
<td>QB</td>
</tr>
<tr>
<td>( Q_C )</td>
<td>The quadrant of the point ( C ) on the fighter's detection pattern. (C)</td>
<td>QC</td>
</tr>
<tr>
<td>( q(l) )</td>
<td>The probability both aircraft are alive at time ( T(l), 1 ) ( ^* )</td>
<td></td>
</tr>
<tr>
<td>( Q(X) )</td>
<td>A function that gives the quadrant of the angle ( X ). (C)</td>
<td>Q</td>
</tr>
<tr>
<td>( R )</td>
<td>The range between the two aircraft. (ft) (C)</td>
<td></td>
</tr>
<tr>
<td>( r )</td>
<td>The range of the detection capability of the fighter; ( r ) is set to ( RDET(F) ) if this is not zero and to ( RQPT(F) ) otherwise. Also used in text for the ( Y^* ) projection against a stationary target. (ft) (C)</td>
<td>RSIALL</td>
</tr>
<tr>
<td>( R_1 )</td>
<td>A variable used to indicate whether the tracking radar is turned on. (ft) (C)</td>
<td>R1</td>
</tr>
<tr>
<td>( R_1, \beta_2 )</td>
<td>Distances used in describing steady state, ( ^* ) (ft)</td>
<td></td>
</tr>
<tr>
<td>( r_{in} )</td>
<td>An override initial range that will act so as to shrink the detection range to ( r_{in} ). (ft) (I)</td>
<td>RANGE</td>
</tr>
<tr>
<td>( R'(1, i) )</td>
<td>The first and second values respectively that will be assigned to ( R^8(1) ), i.e., before and after the opponent becomes aware. (ft) (I)</td>
<td>RPRIME</td>
</tr>
<tr>
<td>( R'(2, i) )</td>
<td>( R(2, i) )</td>
<td></td>
</tr>
<tr>
<td>( RFLOOR(MI, i) )</td>
<td>A superimposed minimum boundary of weapon type MIS such that the weapon type may not be fired from aircraft ( i ) whenever the range is less than ( RFLOOR(MI, i) ). (ft) (I)</td>
<td>RFLOOR</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
<td>Fortran</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>RNOV(1)</td>
<td>The range at which aircraft i may commence to fire at an unaware enemy; against an unaware target, firings by aircraft i are postponed until within a range of $R \leq \text{RNOV}(i)$. (ft) (C)</td>
<td>RNOV</td>
</tr>
<tr>
<td>RTEST(1)</td>
<td>The ranges at which At will change values from $t_D(1)$ to $t_D(2)$ and from $t_D(2)$ to $t_D(3)$ respectively. (ft) (I)</td>
<td>RTEST</td>
</tr>
<tr>
<td>RTEST(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPAS(1)</td>
<td>The range of the passive detection capability of aircraft i . (ft) (I)</td>
<td>RPAS</td>
</tr>
<tr>
<td>RTRK(1)</td>
<td>The range of the tracking radar of aircraft i . (ft) (I)</td>
<td>RTRK</td>
</tr>
<tr>
<td>ROPT(1)</td>
<td>The range of the optical capability of aircraft i . (ft) (I)</td>
<td>ROPT</td>
</tr>
<tr>
<td>RDET(1)</td>
<td>The range of the detection radar of aircraft i . (ft) (I)</td>
<td>RDET</td>
</tr>
<tr>
<td>RIFF(1)</td>
<td>The range of the IFF capability of aircraft i . (ft) (I)</td>
<td>RIFF</td>
</tr>
<tr>
<td>RMIN</td>
<td>A minimum range which will terminate an engagement; $R &lt; \text{RMIN}$ causes termination of an engagement. (ft) (I)</td>
<td>RMIN</td>
</tr>
<tr>
<td>RMIS</td>
<td>The minimum range of some weapon type. (ft) (C)</td>
<td>RMIS</td>
</tr>
<tr>
<td>RMIS</td>
<td>The maximum range of the weapon envelope of some weapon type. (ft) (C)</td>
<td>RMIS</td>
</tr>
<tr>
<td>RF1T(V_a, o_b, MIS, 1)</td>
<td>The outer weapon envelope of weapon type MIS on aircraft i associated with a velocity of $V_a$ and an angle-off of $o_b$ for a value of target $G_1$. (ft) (I)</td>
<td>RF1T</td>
</tr>
<tr>
<td>RF2T(V_a, o_b, MIS, 1)</td>
<td>The same as $R_1(V_a, o_b, MIS, 1)$ but for a target $G_2$. (ft) (I)</td>
<td>RF2T</td>
</tr>
<tr>
<td>RF3T(V_a, o_b, MIS, 1)</td>
<td>The same as $R_1(V_a, o_b, MIS, 1)$ but for a target $G_3$. (ft) (I)</td>
<td>RF3T</td>
</tr>
<tr>
<td>RF1PT(V_a, o_b, MIS, 1)</td>
<td>The inner envelope limit of weapon type MIS on aircraft i for an average velocity of $V_a$ and angle-off of $o_b$ for $G(1)$ total target $G_1$'s. (ft) (I)</td>
<td>RF1PT</td>
</tr>
<tr>
<td>RF2PT(V_a, o_b, MIS, 1)</td>
<td>The same as $R_1(V_a, o_b, MIS, 1)$ but for a target $G_2$. (ft) (I)</td>
<td>RF2PT</td>
</tr>
<tr>
<td>RF3PT(V_a, o_b, MIS, 1)</td>
<td>The same as $R_1(V_a, o_b, MIS, 1)$ but for a target $G_3$. (ft) (I)</td>
<td>RF3PT</td>
</tr>
</tbody>
</table>

7-10
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>The rate of change of the range; (ft/sec) (C)</td>
<td>RDOT</td>
</tr>
<tr>
<td>$L$</td>
<td>The range which aircraft $i$ will attempt to attain when in the rear of its enemy, (ft) (C)</td>
<td>RSTAR</td>
</tr>
<tr>
<td>$R(i)$</td>
<td>The range of the tail gun of aircraft $i$, (ft) (I)</td>
<td>RANGUN</td>
</tr>
<tr>
<td>$R(\phi_j)$</td>
<td>The maximum range at which aircraft $j$ can fire any weapon when approaching an unaware target from the rear, based on the speeds of the aircraft and tracking angle of the pursuer. It assumes the target flies linearly. This is also used as the name of the routine that calculates $R(\phi_j)$, (ft) (C)</td>
<td>RPHIJ</td>
</tr>
<tr>
<td>$S_{1}$</td>
<td>The expected fraction of $t_{\text{MAX}}$ that the pilot of aircraft $i$ will spend in a &quot;sick&quot; condition, $0 \leq S_{1} \leq 1$, for an encounter. Also used to represent the total amount of time that the pilot of aircraft $i$ spends in a sick condition. (In latter case: sec) (C)</td>
<td>TINSIC</td>
</tr>
<tr>
<td>$S_{1}(c)$</td>
<td>The expected fraction of $t_{\text{MAX}}$ that the pilot of aircraft $i$ will spend in a &quot;sick&quot; condition for some value of $c$. (C)</td>
<td>SICEPS</td>
</tr>
<tr>
<td>$S_{1}(c, n)$</td>
<td>The fraction of $t_{\text{MAX}}$ that the pilot of aircraft $i$ spends in a &quot;sick&quot; condition during an engagement defined by $c$ and $n$. (C)</td>
<td>SICTIM</td>
</tr>
<tr>
<td>$ST(i)$</td>
<td>An index indicating the general maneuver of aircraft $i$; (C)</td>
<td>IST</td>
</tr>
<tr>
<td></td>
<td>0 — circle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 — linear flight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 — pursuit course</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 — circle, lost information</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 — evade</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 — evade, lost information</td>
<td></td>
</tr>
<tr>
<td>$S$</td>
<td>An estimate of the amount of change in range when decelerating at a constant rate from some velocity to $V_0$ at a rate of $a_{\text{DEC}}(i)$. (ft) (C)</td>
<td></td>
</tr>
<tr>
<td>$\text{sgn}(x)$</td>
<td>The signature function of the argument $x$; (C)</td>
<td>SCN</td>
</tr>
<tr>
<td></td>
<td>$\text{sgn}(x) = \begin{cases} 1, &amp; \text{if } x \geq 0 \ -1, &amp; \text{Otherwise.} \end{cases}$</td>
<td></td>
</tr>
</tbody>
</table>

7-11
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_f(i)$</td>
<td>An index with values of 0, 1 indicating respectively that the degree of oxygen debt of the pilot of aircraft $i$ will not or will effect the firing of weapons by retarding such firings. (I)</td>
<td>$S_f$</td>
</tr>
<tr>
<td>$S(MIS, i)$</td>
<td>An index with values of 1 and 0 respectively indicating that aircraft $i$ has or has not fired weapon type MIS. (C)</td>
<td>$S$</td>
</tr>
<tr>
<td>$T$</td>
<td>Represents &quot;target&quot; in text (often used as subscript). (I)</td>
<td>$T$</td>
</tr>
<tr>
<td>$t$</td>
<td>The amount of time since initialization of an engagement. (sec) (C)</td>
<td></td>
</tr>
<tr>
<td>$t_f$</td>
<td>The time of flight of all weapons. (sec) (I)</td>
<td>$T_f$</td>
</tr>
<tr>
<td>$t'$</td>
<td>An arbitrary time prior to $T(f)$. Also used in text as a time prior to detection. (sec)</td>
<td></td>
</tr>
<tr>
<td>$t_{AWARE}$</td>
<td>The time at which the bomber became aware of the presence of the fighter. (sec) (C)</td>
<td>$T_{AWARE}$</td>
</tr>
<tr>
<td>$T_{LAST}$</td>
<td>The duration of time of an engagement. (sec) (C)</td>
<td>$T_{LAST}$</td>
</tr>
<tr>
<td>$t_{\text{max}}$</td>
<td>The maximum amount of combat time allowed for a single engagement. (sec) (I)</td>
<td>$T_{\text{max}}$</td>
</tr>
<tr>
<td>$t_{\text{min}}$</td>
<td>The amount of time required to elapse after which a loss of information by both combatants will terminate an engagement. (sec) (I)</td>
<td>$T_{\text{min}}$</td>
</tr>
<tr>
<td>$t_{\text{PRT}}$</td>
<td>The amount of time until the next printout of each aircraft's relevant parameters. (sec) (C)</td>
<td>$T_{\text{PRT}}$</td>
</tr>
<tr>
<td>$t_{p}(1), t_{p}(2), t_{p}(3)$</td>
<td>The first, second and third values assigned at at various transitions during an engagement. (sec) (I)</td>
<td>$T_{DELT}$</td>
</tr>
<tr>
<td>$t_{C}(1)$</td>
<td>The maximum amount of combat time allowed for aircraft $i$. (sec) (I)</td>
<td>$T_{C}$</td>
</tr>
<tr>
<td>$t_{MIS}(1)$</td>
<td>The time at which aircraft $i$ fired the first weapon of type MIS. (sec) (C)</td>
<td>$T_{MIS}$</td>
</tr>
<tr>
<td>$t(MIS, i)$</td>
<td>The time of the last firing of a weapon of type MIS from aircraft $i$. (sec) (C)</td>
<td>$T_{LASTF}$</td>
</tr>
<tr>
<td>$t^\ast$</td>
<td>The amount of time between printouts of the combatants' relevant variables. (sec) (I)</td>
<td>$T^\ast$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
<td>Fortran</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>$T(t)$</td>
<td>The time at which the $i^{th}$ firing took place during an engagement. (sec) (C)</td>
<td>TFIREI</td>
</tr>
<tr>
<td>$V_a$</td>
<td>Aircraft speed used in the input tables for launch envelopes and for energy-maneuverability. (ft/sec) (C)</td>
<td>VATAB, VATAG</td>
</tr>
<tr>
<td>$V_o$</td>
<td>The speed for aircraft $i$ which would make the range rate, $R$, equal zero, subject to $V_o \geq V_c(i)$. (ft/sec) (C)</td>
<td>VZERO</td>
</tr>
<tr>
<td>$V'$</td>
<td>An arbitrary speed used in the distance function $S$. (ft/sec)</td>
<td></td>
</tr>
<tr>
<td>$V^*$</td>
<td>The speed of one aircraft relative to the other at the time of initialization. (ft/sec) (C)</td>
<td>VSTAR</td>
</tr>
<tr>
<td>$V_i$</td>
<td>The speed of aircraft $i$. (ft/sec) (C)</td>
<td>V</td>
</tr>
<tr>
<td>$V_o(i)$</td>
<td>The initial speed of aircraft $i$. (ft/sec) (I)</td>
<td>VZ</td>
</tr>
<tr>
<td>$V^*(i)$</td>
<td>The speed at which the sustainable turning rate of aircraft $i$ is an absolute maximum. (ft/sec) (C)</td>
<td>VSTR</td>
</tr>
<tr>
<td>$V_c(i)$</td>
<td>The minimum sustainable speed of aircraft $i$. (ft/sec) (C)</td>
<td>VC</td>
</tr>
<tr>
<td>$V_{max}(i)$</td>
<td>The maximum sustainable speed of aircraft $i$. (ft/sec) (I)</td>
<td>VMAX</td>
</tr>
<tr>
<td>$W_i$</td>
<td>The weight of aircraft $i$. (lbn) (I)</td>
<td>W</td>
</tr>
<tr>
<td>$X$</td>
<td>One axis of the moving coordinate system used in the initiation phase. (ft)</td>
<td></td>
</tr>
<tr>
<td>$XPHI$</td>
<td>A multiplier with values 0 and 1 to change the computed angle $\theta^*$ to 0 or to leave it as is. (I)</td>
<td>XPHI</td>
</tr>
<tr>
<td>$x_i$</td>
<td>The $x$ position of aircraft $i$ in the $(x, y)$ inertial coordinate system. (ft) (C)</td>
<td>X</td>
</tr>
<tr>
<td>$x$</td>
<td>A symbol used as the argument of various functions. Also used for one axis of the inertial coordinate system. (I) (ft in latter case)</td>
<td>CAPX</td>
</tr>
<tr>
<td>$X$</td>
<td>Used for temporary computations. (C)</td>
<td></td>
</tr>
</tbody>
</table>

7-13
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>A temporary calculation of the maximum turning rate of an aircraft when pulling; the minimum of $g_i(V_i)$ and $G_i(V_i)$ total g's. (rad/sec)$^2$</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>A temporary calculation of the maximum turning rate of an aircraft when pulling; $G_i(V_i)$ total g's. (rad/sec)$^2$</td>
<td></td>
</tr>
<tr>
<td>$y$</td>
<td>A temporary calculation of the probability that both aircraft are alive at the time of firing of the $i^{th}$ weapon. (C)</td>
<td></td>
</tr>
<tr>
<td>$x_1^* , y_1^*$</td>
<td>Temporary computations of the future position of aircraft 1. (ft)$^2$</td>
<td></td>
</tr>
<tr>
<td>$y$</td>
<td>One axis of the moving coordinate system used in the initiation phase. (ft)</td>
<td></td>
</tr>
<tr>
<td>$y$</td>
<td>One axis of the inertial coordinate system. (ft)</td>
<td></td>
</tr>
<tr>
<td>$y_B$</td>
<td>The relative Y coordinate of the bomber at the beginning of the engagement. (ft) (C)</td>
<td></td>
</tr>
<tr>
<td>$y_C$</td>
<td>The normal projection of the fighter's detection pattern onto the Y axis in the $(X, Y)$ coordinate system at the beginning of one engagement. (ft) (C)</td>
<td></td>
</tr>
<tr>
<td>$y_{MIN}$</td>
<td>The lower limit of the segment $Y^*$ in the $(X, Y)$ coordinate system (used in initiating engagements). (ft) (C)</td>
<td></td>
</tr>
<tr>
<td>$y_{MAX}$</td>
<td>The upper limit of the segment $Y^*$ in the $(X, Y)$ coordinate system (used in initiating engagements). (ft) (C)</td>
<td></td>
</tr>
<tr>
<td>$y^*$</td>
<td>The steady state speed. (ft/sec) (C)</td>
<td></td>
</tr>
<tr>
<td>$y^*$</td>
<td>A temporary calculation of the probability that aircraft $j$ survived any weapons that hit between the firing and arrival of the $i^{th}$ weapon. (C)</td>
<td></td>
</tr>
<tr>
<td>$y_{NEW}$</td>
<td>A parameter used in determining the detection contour for some $e_i$. (ft) (C)</td>
<td></td>
</tr>
<tr>
<td>$y_{OLD}$</td>
<td>A parameter used in determining the detection contour for some $e_i$. (ft) (C)</td>
<td></td>
</tr>
</tbody>
</table>

2) Used in flow chart only.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_i$</td>
<td>The $y$ position of aircraft $i$ in the $(x, y)$ inertial coordinate system, (ft) (C)</td>
<td>$Y$</td>
</tr>
<tr>
<td>$z_K^l , z_K^r$</td>
<td>Temporary calculations used to compute the weapon launch envelopes, (ft$^2$)</td>
<td></td>
</tr>
<tr>
<td>$z_L^l , z_L^r$</td>
<td>Used to describe DEL course, $^1)$ (rad)</td>
<td></td>
</tr>
<tr>
<td>$a_1 , a_2 , a_3 , a_4$</td>
<td>The rate of change of $a_k$; (rad/sec) (C) $a_i = da_i/dt$</td>
<td>ALPDOT</td>
</tr>
<tr>
<td>$a_{KB}^l (i)$</td>
<td>The half-angle defining the tail gun capability of aircraft $i$ measured from the tail of the aircraft, (deg) (I)</td>
<td>ALPGUN</td>
</tr>
<tr>
<td>$a_{MAX}^l (i)$</td>
<td>The internally computed parameter of the DEL function of aircraft $i$, (rad) (C)</td>
<td>ALEMIX</td>
</tr>
<tr>
<td>$a_{MIS}^l (i)$</td>
<td>The half-angle of the cone in which aircraft $i$ must be tracking its enemy in order to fire weapon type MIS, (deg) (I)</td>
<td>ALPMIS</td>
</tr>
<tr>
<td>$a_{DET}^l (i)$</td>
<td>The tracking angle of aircraft $i$, measured from the inner line of sight to the velocity vector of the aircraft, (rad) (C)</td>
<td>ALPDET</td>
</tr>
<tr>
<td>$a_{IFF}^l (i)$</td>
<td>The half-angle of the IFF capability of aircraft $i$, (deg) (I)</td>
<td>ALPIFF</td>
</tr>
<tr>
<td>$a_{OPT}^l (i)$</td>
<td>The half-angle of the optical capability of aircraft $i$, (deg) (I)</td>
<td>ALPOPT</td>
</tr>
<tr>
<td>$a_{PAS}^l (i)$</td>
<td>The half-angle of the passive detection capability of aircraft $i$ measured off the tail of the aircraft, (deg) (I)</td>
<td>ALPPAS</td>
</tr>
<tr>
<td>$a_{TRK}^l (i)$</td>
<td>The half-angle of the tracking radar cone of aircraft $i$, (deg) (I)</td>
<td>ALPTRK</td>
</tr>
<tr>
<td>$\beta_i$</td>
<td>The turning rate of aircraft $i$, (rad/sec) (C)</td>
<td>BETDOT</td>
</tr>
<tr>
<td>$\beta_{i}$</td>
<td>The angle of aircraft $i$'s heading measured from the $x$ axis, (rad) (C)</td>
<td>BETA</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>An angle used in defining the initial positions of the aircraft, (rad) (C)</td>
<td>GAMMA</td>
</tr>
</tbody>
</table>

$^2)$ Used in flow chart only.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \delta )</td>
<td>The density of fighters assigned to an area for search purposes; used in computing the probability of detection, ( P_d(\epsilon) ). (ft(^{-2})) (I)</td>
<td>DELTA</td>
</tr>
<tr>
<td>( \Delta a )</td>
<td>The change in ( a ) in one time pulse,(^{1)} ) (rad)</td>
<td></td>
</tr>
<tr>
<td>( \Delta t )</td>
<td>The incremental time slice of simulation -- a time for which rate of change, ( \dot{A}, \dot{\theta}, \dot{e}, ) etc., are assumed to be constant in the integration of the equations of motion. (sec) (C)</td>
<td>DELTAT</td>
</tr>
<tr>
<td>( \Delta y_g )</td>
<td>The change in the value of ( Y_g ) for different grid-points. (ft) (C)</td>
<td>DELTYG</td>
</tr>
<tr>
<td>( \Delta \epsilon )</td>
<td>The change between ( \epsilon ) values for various sets of grid-points. If input to a value greater than 180(^{\circ}) the program will execute only one value of ( \epsilon ). (deg) (I)</td>
<td>DELEPS</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>The angle between the velocity vectors of the combatants measured from the fighter to the bomber at the time of initialization. Also used to indicate the index of a particular value of ( \epsilon ) in summations, (In special cases this may be an input.) (deg) (C)</td>
<td>EPSLON</td>
</tr>
<tr>
<td>( \epsilon_1, \epsilon_2 )</td>
<td>The first and last values respectively of ( \epsilon ), (deg) (C)</td>
<td>EPSLON</td>
</tr>
<tr>
<td>( \eta )</td>
<td>The desired tracking angle of an aircraft as computed by the DEL course function, ( \theta_a ) (rad) (C)</td>
<td>ETA</td>
</tr>
<tr>
<td>( \lambda_i )</td>
<td>The constant deviate angle flown by aircraft ( i ) when inside the ( \psi^* ) cone of its enemy. (deg) (I)</td>
<td>LAMBDA</td>
</tr>
<tr>
<td>( \theta_B )</td>
<td>The angle between the X-axis and the ray from (0, 0) to the point ( B ) of the fighter's detection pattern, (rad) (C)</td>
<td>THETR</td>
</tr>
<tr>
<td>( \theta_C )</td>
<td>The angle between the X-axis and the ray from (0, 0) and the point ( C ) on the fighter's detection pattern, (rad) (C)</td>
<td>THEIC</td>
</tr>
<tr>
<td>( \dot{\theta} )</td>
<td>The turning rate of the line of sight, (rad/sec) (C)</td>
<td>THEDOT</td>
</tr>
<tr>
<td>( \mu )</td>
<td>The angle between the vector ( V^e ) and the heading of the fighter at initialization, (rad) (C)</td>
<td>IHU</td>
</tr>
<tr>
<td>\pi</td>
<td>3.14159 ... (C)</td>
<td>PI</td>
</tr>
</tbody>
</table>

7-16
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>ρ</td>
<td>The half-angle of the fighter's detection capability; ( ρ ) is set to ( ρ_{DET}(β) ) if this is not zero and to ( ρ_{OPT}(β) ) otherwise. (rad) (C)</td>
<td>RHO</td>
</tr>
<tr>
<td>δ</td>
<td>An angle indicating the angle-off times the sign of the direction of turn of the target. (rad) (I)</td>
<td>SIGB</td>
</tr>
<tr>
<td>σ(MIS, i)</td>
<td>A counter of the amount of time that aircraft ( i ) has spent in the envelope of weapon type MIS since the last firing of weapon type MIS. (sec) (C)</td>
<td>SIGMIS</td>
</tr>
<tr>
<td>t(_i)</td>
<td>The approximate amount of time that aircraft ( i ) will need to fire a weapon after its tracking radar is activated; used to compute distance from ( R(φ_i) ) at which to activate its tracking radar. (sec) (I)</td>
<td>TAU</td>
</tr>
<tr>
<td>τ(MIS, i)</td>
<td>A counter of the total amount of time that aircraft ( i ) has spent in the envelope of weapon type MIS. (sec) (C)</td>
<td>TAUNIS</td>
</tr>
<tr>
<td>φ(_i)</td>
<td>The angle off the target of aircraft ( i ); measured from the outer line of sight to the heading of aircraft ( j ), ( j \neq i ). (rad) (C)</td>
<td>PHI</td>
</tr>
<tr>
<td>( \dot{φ}_i )</td>
<td>The rate of change of ( φ_i ); ( \text{rad/sec) (C) PHIDOT, \dot{φ}_i = \frac{dφ_i}{dt} )</td>
<td></td>
</tr>
<tr>
<td>( α^* )</td>
<td>The half-angle of the cone in the rear of a target aircraft within which pure pursuit is the navigation doctrine, if ( α^*_i = 0 ). (rad) (C)</td>
<td></td>
</tr>
</tbody>
</table>
7.3 Input Considerations

While ATAC-2 was developed with a general user in mind, completely arbitrary inputs are, of course, impossible to handle. Limitations exist and considerations must be given to the values of inputs. These considerations are necessary, in some instances, due to the nature of air-to-air combat and in other instances due to the specifics of the program as it exists. Below are listed some of these considerations and limitations under various headings.

7.3.1 Parameter Guidance

(1) In the program the value of the internal variable $R^*(F)$ is initially set to the input $R'_{(1, F)}$. This is the range which the fighter tries to attain off the tail of an unaware bomber, before firing any weapons. If the fighter can maintain surprise $R'_{(1, F)}$ will be the range at which the bomber becomes aware by being fired upon. Two considerations, therefore, should be observed when selecting a value for $R'_{(1, F)}$.

Firstly, this range should be such as to allow firings of the more lethal weapons of the fighter. Secondly, the range should be such as to allow the fighter to stay behind the bomber when the latter becomes aware and begins to maneuver. This last consideration is very difficult to quantify. In general, the ability of the fighter to stay behind a maneuvering bomber is a function of both velocities, both turning rates and both deceleration rates. However, the closer the fighter is to its steady state range and its associated velocity (see Appendix F), if it exists, the easier it will be to stay behind the bomber.
(2) The range $R^*(i)$, input as $R'(2, i)$, represents the range at which aircraft $i$ would like to be off the tail of its enemy so as to be able to fire its shortest range weapons, usually guns, while the enemy is maneuvering. The selection of this parameter is influenced by the capability of this shortest range weapon. However, the maneuverability of both aircraft must be accounted for. In Appendix $P$ it is shown that selection of $R^*(i)$ (or a steady state range) has implications on other parameters as well, namely the velocity and angle-off associated with $R^*(i)$. Thus, if possible these considerations (the values of $R^*(i)$ that allow weapon firings and the values of $R^*(i)$ that allow steady state conditions to obtain) should be combined to arrive at a realizable $R^*(i)$ whenever possible.

It should be noted that $R^*(i)$ originally takes on an input value $R'(1, i)$ which in the case of the fighter, is the range at which it may begin firing. Once this range is achieved, it then closes to this close range $R'(2, i)$.

(3) The length of the time pulse $At$ influences the running time of the computer program. The running time is inversely proportional to $At$. However, as $At$ increases some singularities occur. For example, the path of an aircraft may cross a launch envelope from one pulse to the next, without being in the launch envelope at the beginning of a pulse. A possible firing will be missed. Also, since the method of numerical integration in the model assumes that the time derivatives of the relative parameters are constant for a period of time $At$, the length of $At$ should be kept fairly small. The error due to this assumption is inversely proportional to $R^2$. The assumption is therefore worse for
smaller values of range; hence the capability in the program to decrease
the length of At as the range gets smaller. Typical values for At are:

\[ t_D(1) = 5.0 \text{ sec.} \quad R_{\text{TEST}}(1) = 100,000 \]
\[ t_D(2) = 2.0 \text{ sec.} \quad R_{\text{TEST}}(2) = 70,000 \]
\[ t_D(3) = .25 \text{ sec.} \]

7.3.2 Model Logic

The following are logical restrictions of the model, i.e., the
violation of them will affect the logic of the model:

1. All calculations involving \( a_{\text{DEC}}(i) \) assume an associated
   negative value; hence \( a_{\text{DEC}}(i) \) must be inputted as less than
   or equal to zero.

2. IFF is necessary before firings can take place; \( a_{\text{IFF}}(i) \) and
   \( R_{\text{IFF}}(i) \) must, therefore, be non-zero if aircraft \( i \) is to
   fire its weapons.

3. Each aircraft may obtain active information only from either
   its detection radar or optical system. An extension of tracking
   radar coverage over that of the detection coverage adds no
   capability. To avoid confusion and false interpretation
   the tracking pattern should be contained within the detection
   pattern: \( a_{\text{TRK}}(i) \leq a_{\text{DET}}(i) \) and \( R_{\text{TRK}}(i) \leq R_{\text{DET}}(i) \).
4. The speeds for which the specific power function is inputted should be restricted to allow the aircraft at least to maintain altitude at the assumed power setting and altitude. Both \( g \) functions of speed \( g_4(V) \) and \( G_4(V) \) must be greater than or equal to 1 for all possible speeds. The specific power function must be greater than or equal to zero at 1-total \( g \) (zero turning rate) for all possible speeds.

7.3.3 Flow Chart Restrictions

The following restrictions are implied by the flow charts:

1. There are three \((3)\) \( g \) values for which the weapon launch envelopes are input, namely \( g_1, g_2, g_3 \).

2. In the tables for \( g_4(V), G_4(V) \), and weapon envelopes the minimum and maximum speeds should extend at least to the possible speed values of \( V_c(i) \) and \( V_{\text{max}}(i) \), respectively. Otherwise when a routine uses a table to find a functional value for a speed outside the tabular range, the routine linearly extrapolates to find the required functional value.

3. The time, \( t^* \), between printings of the positional parameters of the aircraft should be a multiple of the time pulse \( \Delta t \). This avoids round-off errors and ensures a constant time between printings.

4. The initial velocity, \( V_0(i) \), of aircraft \( i \) must, of course, be within the velocity region \( V_c(i) \) to \( V_{\text{max}}(i) \).
5. Whenever \( g_j(V) \) is unattainable due to the value of \( G_j(V) \) being prohibitive, any value greater than or equal to \( G_j(V) \) may be used for \( g_j(V) \) at that \( V \). The program will select the minimum of the two when appropriate.

6. The tail gun of aircraft \( i \) is assumed to be weapon number \( n_m(i) \), the last weapon. Further \( n_m(i) \) may not be zero. The capability and hence the effect of the tail gun may be negated by setting \( R_g(i) \) and \( q_g(i) \) to zero, or setting \( N(n_m(i), i) \) to zero.

7.3.4 Program Restrictions

The following are restrictions that exist in the current program. They arise from computer space allocation considerations. To change them, however, may require significant programming effort.

1. The number of weapon types must be no more than six: \( n^m(i) \leq 6 \).

2. The number of grid-points for one \( \epsilon \) must be limited by twenty: \( N \leq 20 \).

3. The number of speed values for which the weapon launch envelopes are input must be either 2 or 3.

4. The number of angular values for which the weapon launch envelopes are input must be between two and fifteen inclusive.

5. Since the number of \( \epsilon \) values considered will be \( 180^\circ/\Delta\epsilon \) or \( 360^\circ/\Delta\epsilon \), \( \Delta\epsilon \) must be greater than or equal to 30° if \( PS (F \text{ or } B) = 1 \) and greater than or equal to 15°, otherwise.
SECTION 8
FLOW CHARTS

In this section the flow charts of the various ATAC-2 routines are presented. They are intended for use with the relevant discussion of Volume II. The EM flow charts are discussed in Section 5, Volume II. The discussion of the routines of the DATA PROCESSING Model is given in Section 6.5, Volume II.
THE EXECUTIVE ROUTINE-EM

START

INPUT

ΔE > π

NO

YES

PS(i) = 1
PS(j) = 1

ε = π

ε = 0

R_{ij}(P) = 0

YES

NO

r = R_{max}(P)
ρ = k_{max}(P)

r = min(r; \rho)

GRID PRIOR

COMBAT

ε = Estab

ε > π

NO

YES

STOP

- 24 -
THE GRID PREPARATION ROUTINE

\[ V^0 = [V(F) - V(B)]^2 - 2V(F)V(B) \text{ for } V > 0 \]

\[ V = 0 \]

\[ A = 0 \]

\[ \lambda = \cos^{-1} \left( \frac{V(F)}{V(B)} \right) \]

\[ E > 0 \]

\[ \lambda = -\lambda \]

\[ Q_x = P(V(x,y)) \]

\[ Q_x < P(V(x,y)) \]

\[ Y_0 = \sin \lambda \theta \]

\[ Z_0 = \cos \lambda \theta \]

\[ Y_{x+1} = Y_x + Y_0 \]

\[ Z_{x+1} = Z_x + Z_0 \]

\[ Q_x = 0 \]

\[ Q_x = 1 \]

\[ Q_x = 2 \]

\[ Q_x = 3 \]

\[ Q_x = 4 \]

\[ Y_x = 0 \]

\[ Z_x = 0 \]

\[ Y_{x+1} = Y_x + Y_0 \]

\[ Z_{x+1} = Z_x + Z_0 \]

\[ Q_x = 5 \]

\[ Q_x = 6 \]

\[ Q_x = 7 \]

\[ Q_x = 8 \]

\[ Y_x = 0 \]

\[ Z_x = 0 \]

\[ Y_{x+1} = Y_x + Y_0 \]

\[ Z_{x+1} = Z_x + Z_0 \]

\[ Q_x = 9 \]

\[ Q_x = 10 \]

\[ Q_x = 11 \]

\[ Q_x = 12 \]
THE COMBAT ROUTINE

START

GRIDPOINT = 1
GRID

INITIALIZE FLIGHT

L = F
D = B

NAVIGATIONAL SYSTEM

L = D
D = F

NAVIGATIONAL SYSTEM

ADVANCE RELATIVE COORDINATES

TRANSFORM TO INERTIAL COORDINATES

FIND Gt for SYSTEM

L = F
D = B

CHECK WEAPON

L = D
F = F

CHECK WEAPON

PRINT

OVER

RESULT

YES

GRIDPOINT = N

GRIDPOINT = GRIDPOINT + 1

RETURN

- 26 -
THE GRID ROUTINE

```
EVENT

Y3 = Y3 - AY

Y = sin(3/2)

Y2 = Y2 + Y3

Y2 < Y2

R = Y

Y3 = Y3

Y2 = Y2 - Y3

R = sin((A - Y3))/Y3

Y3 = Y3 + 1

R = sin(Y3)
```

- 27 -
THE INITIALIZE FLIGHT ROUTINE

START
\[ t_0 = 10^9 \]
\[ R_0 = R_{ini,0} \]
\[ \Delta t = t_0 (1) \]
\[ t_{next} = 2 \]
\[ R = R_{ini,0} \]
\[ \Delta t = t_0 (1) \]
\[ t_{next} = 3 \]
\[ h_{next} = \Delta t \]

Set to zero for \( \text{HIS} = 1, 2, \ldots, n \)
and \( t = F, B \):
- \( \beta_1, \Omega_1, B_1, t_2(1), P_1 (1), t(his,1), \sigma_{(mes,3)}, k_{(mes,1)}, s_{(mes,1)}, M_1, \sigma_{(mes,3)} \)
- \( \text{ENK, SW, } \text{MEK, } \text{MKE} \)

\( t(N) = O \)
\( \text{IAD} = O \)
\( H_2(N) = O, t \)
\( \text{a} + 1 = 2, \ldots, L \)

\[ \text{KOWN}(N) = R(N) \]
\[ \text{R}(t) = \text{KOWN}(t) \]
\[ \text{STG} = L \]
\[ \text{A} = X(t) \]
\[ \text{a} = F, B \]

\[ [\text{if } \text{a} = F, \text{b} = \text{p} ] \]
\[ [\text{h} \times \text{a} \times \text{p} ] \]
\[ \text{N} = \text{a} \times \text{p} \]
\[ \text{p} \times \text{p} \times \text{p} \]
\[ \text{k} \times \text{p} \times \text{p} \]
\[ \text{a} \times \text{p} \times \text{p} \]

\[ \text{m} + \text{p} = \text{p} + \text{p} \]
\[ \text{p} \times \text{p} = \text{p} + \text{p} \]
\[ \text{a} \times \text{p} = \text{p} + \text{p} \]

\[ \text{a} = \text{a} + \text{p} \]
\[ \text{p} = \text{p} + \text{p} \]

\[ M = \text{a} \times \text{p} \]
\[ \text{a} = \text{a} + \text{p} \]

RETURN
NAVIGATIONAL SYSTEMS

\[ q = A \phi (x(v; \xi, \eta); e_{0}) \]

\[ q \in A \Phi (v) \]

Diagram A

Diagram B

Diagram C

Diagram D

Diagram E

Diagram F
THE ACTIVE ROUTINE

START

NO

RETURN

NO

AVARATE ROUTINE

YES

THE PASSIVE ROUTINE
THE $R(\theta)$ FUNCTION

**START**

Define $R(\theta_j)$ as the maximum range at which aircraft $j$ may fire a weapon based on zero target $g_j$'s, the current velocity of the target, $g_j$'s and tracking angle of aircraft $j$.

**RETURN**

THE AWARE ROUTINE

**START**

$\text{MAX} \begin{cases} \text{t}_{\text{MAX}} = 1,000 \quad \text{if} \quad l = 3 \end{cases} \rightarrow \text{RETURN}$

$\text{t}_{\text{MAX}} = t$

$\text{ICAN}(\theta) = 0$

$R^o(\theta) = R(\theta, \theta) \quad R(\theta) = R(\theta, \theta)$

**RETURN**
THE $G_i(x)$ ROUTINE

Linearly interpolate across input velocity values to find $G_i(0)$

\[
\begin{align*}
&\text{if } B_i = 1 \text{ and } G_i(0) = 1, \\
&\quad \text{go to step 10} \\
&\text{if not, } \\
&\quad G_i(0) = \min\{G_i(0); 1.5\} \\
&\quad \text{return}
\end{align*}
\]

THE $G_i(x)$ ROUTINE

Linearly interpolate across input velocity values to find $G_i(0)$

\[
\begin{align*}
&\text{if } B_i = 1 \text{ and } G_i(0) = 1, \\
&\quad \text{go to step 10} \\
&\text{if not, } \\
&\quad G_i(0) = \min\{G_i(0); 1.5\} \\
&\quad \text{return}
\end{align*}
\]
THE $p(M,\beta)$ FUNCTION

\[ p = \left[ 1 + \left( \frac{V/M}{100} \right)^2 \right]^{1/2} \]

Linearly interpolate across dimensions of velocity (V is desired) and total g's ($\beta$ is desired) in the input specific power table.

RETURN

THE $\beta(x)$ FUNCTION

\[ \beta(x) = \frac{x}{V} \left( x^2 - 1 \right)^{1/2} \]

RETURN
TRANSFORM TO INERTIAL COORDINATES ROUTINE

\[ x' = x_0 - V_x a t + \omega y a t \cos \alpha \]
\[ y' = y_0 + V_y a t \sin \alpha \]
\[ V_x' = V_x + B_x a t \]
\[ V_y' = V_y + B_y a t \]
\[ \text{RETURN} \]

THE FIND \( G_i \) ROUTINE

\[ G_i [(\cos \beta \sin \beta \cos \gamma + 1)^{i/2}] \]
\[ O_i, O_i + 0.0001, O_i + 0.0005 \]
\[ O_i < 1 \]
\[ O_i > 1 \]
\[ S_i = S_i + \Delta t \]
\[ B_i = 1 \]
\[ \text{RETURN} \]
\[ G_i = G_i + 0.0005 \]
\[ S_i = S_i + \Delta t \]
\[ \text{RETURN} \]
\[ O_i = O_i + 0.01 \]
\[ B_i = 0 \]
\[ O_i = 0 \]
\[ \text{RETURN} \]
CHECK WEAPONS ROUTINE

START

1. IF TEMP(0) = 1 THEN
   2. IF R > Rmax(1) or |m(1)| > mmax(1) THEN
      3. RETURN
   4. IF ICAN(1) = 1 THEN
      5. RETURN
   6. MIS = nmax(1)

2. IF R > Rmax(1) or |m(1)| > mmax(1) THEN
   3. RETURN

4. IF MIS = nmax(1) THEN
   5. C(1) = M + n(1)
   6. CALL R_MIS_ROUTINE
   7. IF MIS < 3 and R < RFLOW(MIS,1) THEN
      8. R12 = R & R12
      9. tSHEE(0) = tmax(1) + at
         10. tPASS(1) = tPASS(1) + at
      11. ENV. Tw(max,1) = t0
      12. S(1) = 1
      13. Q(1) = 1 - exp [-s(tmax,1) - 1/tau]
THE $R_{MIS}$, $R'_{MIS}$ ROUTINE

Select terminal $q_0$, nearest current $q_1$

$G_1 \leq \sqrt{G_0 - 1} \leq G_2$

Yes

$K = 1$, $L = 2$

No

$G_1 \leq \sqrt{G_0 - 1} \leq G_3$

Yes

$K = 2$, $L = 3$

No

$\sqrt{G_0 - 1} < G_1$

Yes

$G_1 = \sqrt{G_0 + 1}$

$K = 2$, $L = 3$

No

$G_1 = \sqrt{G_0 + 1}$

$K = 1$, $L = 2$

Select for the $q_0$ chosen, plus for the particular $G_0 (MIS)$ and the particular $j$:

$R_{M_1}(V_{C_0}), R_{M_2}(V_{C_0}), R_{M_3}(V_{C_0}), R_{M_4}(V_{C_0})$

$R_{M_5}(V_{C_0}), R_{M_6}(V_{C_0}), R_{M_7}(V_{C_0})$, where $V_{C_0}$ is the minimum $V_0$ with $V_0 \geq (V_0 + V_I) (0)$ and $V_{M_0}$ is the maximum $V_0$ with $V_0 \leq (V_0 + V_I) (0)$.

$Z_x = R_{M_0}(V_{C_0}) - \frac{V_{C_0} - V_I}{V_{C_0}} [R_{M_0}(V_{C_0}) - R_{M_1}(V_{C_0})]$  

$Z'_x = R_{M_0}(V_{C_0}) - \frac{V_{C_0} - V_I}{V_{C_0}} [R_{M_0}(V_{C_0}) - R_{M_2}(V_{C_0})]$  

$R_{MIS} = Z_x - \frac{\sqrt{G_0 - 1} - G_2}{G_1 - G_2} [2Z_x - Z_x]$

$R'_{MIS} = Z'_x - \frac{\sqrt{G_0 - 1} - G_2}{G_1 - G_2} [2Z'_x - Z'_x]$  

RETURN

- 38 -
THE PRINT ROUTINE

START

\( t_{\text{ref}} < 0 \) NO

\( t_{\text{ref}} - t^* \)

Print: \( t, R, X, Y, \theta_1, \beta_1, \nu_1 \), \( E_l, \theta(E_l), \theta_l, \nu_l, \nu_h \)
for \( l = F \& B \)

\( t_{\text{ref}} - t^* - \Delta t \)

RETURN

THE OVER ROUTINE

START

BEGIN: \( L = 0 \) and \( ST(BM) = L \) or \( CL \)

YES

NO

RETURN

BEGIN: \( E < E_{\text{ref}} \)

YES

\( t_{\text{ref}} + t \)

RETURN

NO

YES

RETURN

- 39 -
THE RESULTS ROUTINE

THE PV(W) FUNCTION
THE SGN(x) FUNCTION

START

x > 0

YES

NO

sgn(x) = 1

RETURN

sgn(x) = -1

RETURN

THE Q(x) FUNCTION

START

x > \frac{7}{2}

YES

NO

x > 0

YES

NO

x > -\frac{7}{2}

YES

NO

Q(x) = 3

Q(x) = 4

Q(x) = 1

Q(x) = 2

RETURN

41
THE EXECUTIVE ROUTINE-DPM

START

INPUT 1

INPUT 2

$E = E_1$

CALCULATE $E$
DATA

CALCULATE UNCONDITIONAL
DATA

OUTPUT

$E = E_2$

If $E_2 < \epsilon$
ELSE

$E = E + \delta E$

STEP
THE $P^*_j$ ROUTINE

START

$P(0), P(1), \ldots, P(r, 1) = 0$

$x = 1$

$C = P \iff P(x) > 0$

$T(x) = 0 \iff x = 1$

$T(x) = 0 \iff x = 1$

$X = [x - 1]$

$P(x) = P(x - 1)$

$P(x) = P(x - 1)$

RETURN

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$

$X = x + 1$

$T(x) = 0 \iff x = 1$
THE CALC. UNCOND. DATA ROUTINE

START

\[ R_{0} = \sum_{k=1}^{n} r_{k} / \left( n \times (n-1) \right) \]

\[ R_{2} = \sum_{k=1}^{n-1} r_{k} \sum_{k=1}^{n} r_{k} / \left( n \times (n-1) \right) \]

\[ R_{4} = \sum_{k=1}^{n-1} r_{k} \sum_{k=1}^{n} r_{k} / \left( n \times (n-1) \times (n-2) \right) \]

\[ R_{6} = \frac{R_{0}}{n} \quad \text{DIV}=n-1 \]

\[ R_{8} = \frac{R_{2}}{n} \quad \text{DIV}=n-1 \]

\[ R_{0} = \frac{R_{8}}{n} \quad \text{DIV}=n-1 \]

\[ P_{E} = \frac{R_{6}}{n} \quad \text{DIV}=n-1, 2 \]

\[ P_{L} = \frac{1 - P_{E}}{n} \quad \text{DIV}=n-1, 2 \]

\[ P_{K} = P_{L} \cdot E_{K} \]

\[ E_{K} = 1 - P_{E} \quad \text{DIV}=n-1, 2 \]

\[ E_{B} = E_{K} \]

\[ E_{E} = P_{L} \]

\[ P_{K} = 1 - P_{E} \quad \text{DIV}=n-1, 2 \]

RETURN

- 45 -
SECTION 9

PROGRAM LISTING

In this section the FORTRAN IV program of ATAC-2 is presented. For the EM and DPM each, the program consists of a main program (corresponding to the EXECUTIVE Routine) and various FORTRAN subroutines all of which correspond to the previously presented routines.
SUBROUTINE NAVSYS
COMMON/MAXALP/AV(2)
COMMON/EXEC/RSMALL(2),RHO(4),IFTR,IBMR,FM(2),BSMALL(2),
COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHI,ETA,ALPMAX(2),XLAMDA(2)
*
COMMON/XPHIX/BETTAB(27*2),VSTR(2)
COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
10FLEPS,DELTAT,FSLON,SBGB,G37(3),RMIN,H*
4,NM(2),IPS(2),VID,
1V(2),V2(2),V3(2),VA(2),W(2),TSTARS(2),
2ADEC(2),RDET(2),RIFF(2),ROPT(2),
3RPS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPDET(2),VMAX(2),GP(2),ACC(2),
5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
9NUGRC(2)
COMMON/VANUV/IMSTAT(2)
COMMON/G3DSC/GAMMA,ALPHA(2),R
COMMON/INITL,T,PRT,BEDOT(2),ENVSW(6,2),
1ST(2),V(2),JI,JS,TM,S(5,2),PHI(2),Y(2),X(2),
2DX(2),DY(2),TAWAP,G(2),LEFTSW
COMMON/SLOW/DECL
COMMON/NAVSY/GLAT(2),VZERO,S,ASMLL,RC1,
1FLAVDA*M,XC(2),YD(2)
COMMON/RVARY/KCAP,KLITLH,TATPR(6,2),THERE(2),MISPR(6,2),TOLER,
*?PRIME(6,2)
COMMON/CﻮMA/I+J
COMMON/ORC/R prolific
COMMON/C_ptrs/PRIME(2)*K,D(11,2),GCAP(2)
COMMON/PW3IT/KOPT(7)
LOGICAL ACT,LEFTSW,PASS
DIMENSION GMAX(2)
PI 1,14159265
GCAP(1)=GCAP(V(I),I)
GMAX(I) = SMAXF(V(I), I)
BETMAX(I) = 0.
CALL INFO
ALP^vAX(I) = AM(I)
IF (I .NE. IFTR) GOTO 52
IF (TAW .NE. IF .1000.) GOTO 52
ALP^vAX(I) = ALPDEF(I)
CONTINUE
KPRT(I) = K
IF (K .EQ. 1) GO TO 700
IF (D(K-I) .NE. 0.) GO TO 100
IF (K .EQ. 2 .OR. (K .GE. 8)) GOTO 12
IF (ABS(PHI(I)) .GT. PI/2.) GOTO 12
IST(I) = 5
IF (V(I) .GE. VSTR(I)) GOTO 10
CAPX = AMIN1(GCAP(I), GMAX(I))
BETDOT(I) = -SGN(BETDOT(J)) * BFUNCT(CAPX, V(I), I)
**(V(I) / VSTR(I))
IMSTAT(I) = 12
GOTO 13
CONTINUE
CAPX = AMIN1(GCAP(I), GMAX(I))
BETDOT(I) = -SGN(BETDOT(J)) * BFUNCT(CAPX, V(I), I)
A(I) = 32.2 / V(I) * PEEU(V(I), BETDOT(I), I)
A(I) = AMIN1(A(I), (VMAX(I) - V(I)) / DELTAT, (VMAX(I) - V(I)) / DELTAT)
IMSTAT(I) = 11
RETURN
BETDOT(I) = 0.
IMSTAT(I) = 13
IST(I) = 4
A(I) = 32.2 / V(I) * PEEU(V(I), BETDOT(I), I)
A(I) = AMIN1(A(I), (VMAX(I) - V(I)) / DELTAT)
RETURN
IF ((K .NE. 10) .AND. (K .NE. 11)) GOTO 125
IST(I) = 3
IF (V(I) .LE. VSTR(I)) GOTO 106
BETDOT(I) = SGN(BETDOT(I)) * BFUNCT(GCAP(I), V(I), I)
A(I)=AMIN1(32.2/V(I),PPEU(V(I),BETDOT(I),I),ADEC(I))
A(I)=AMAX1(A(I),V(I)-VSTR(I))/DELTAT)
IMSTAT(I)=7
RETURN
106 CONTINUE
BETDOT(I)=SGN(BETDOT(I))*BFUNCT(AMIN1(GCAP(I),GMAX(I)),V(I),I)
**(V(I)/VSTR(I))
A(I)=AMIN1(32.2/V(I),PPEU(V(I),BETDOT(I),I),(VSTR(I)-V(I))/DELTAT)
IMSTAT(I)=8
RETURN
125 IF ((I.EQ.2).OR.(I.EQ.8).OR.(I.EQ.9)) GOTO 600
THEDOT=(VINFTR)*SIN(A.PHA(IFTR))-V(BMR)*SIN(PHI(IFTR)))/R
PHISTR=0.
IF(V(I).LT.2.*V(J)) PHISTR=ARCOS(V(I)/(2.*V(J))
PHISTX=PHISTR
IF (ABS(PHI(I)).GT.PI) ETA=SGN(ALPHA(I))*(ALPMAX(I)/(PHISTR-PI)*((ABS(PHI(I))-PHISTR)-
*XLAMDA(I))
IF (ABS(PHI(I)).EQ.PI) ETA=SGN(ALPHA(I))*(ALPMAX(I)+XLAMDA(I))
3 CONTINUE
BETDOT(I)=THEDOT+(ALPHA(I)-ETA)/DELTAT
IF (ABS(PHI(I)).LE.PI/2.) GO TO 200
CAPX=AMIN1(GCAP(I),GMAX(I))
CAPX=BFUNCT(CAPX,V(I),I)
S=((V(I)-VSTR(I))**2*VSTR(I)**2(VSTR(I)-2.*V(J)**COS(PHI(I))/(1-COS(PHI(I)**2)))/
*(-2.*ADEC(I))
IF (R.LT.S) GO TO 126
IMSTAT(I)=5
IF (V(I).LT.VSTR(I)) CAPX=CAPX*V(I)/VSTR(I)
A(I)=32.2/V(I)*PPEU(V(I),BETDOT(I),I)
GO TO 127
126 CONTINUE
A(I)=ADEC(I)
IMSTAT(I)=6
127 IF(ABS(BETDOT(I)).LT.CAPX) GOTO 128
BETDOT(I) = SGN(BETDOT(I)) * CAPX

IF (IMSTAT(I) .EQ. 5) A(I) = 32.2 / V(I) * PEEU(V(I)) * BETDOT(I) * I
IST(I) = 0
GOTO 129

128 IST(I) = 2
129 CONTINUE
A(I) = A(N) = AMAX1(A(I), (V(I) + V(J)) / DELTAT, (VMAX(I) - V(I)) / DELTAT)
RETURN

200 CONTINUE
XX = ALPHA(I)
IF (ABS(XX) .GE. PI / 2.0) XX = SGN(XX) * PI / 2.0
VZERO = AMAX1(V(I), V(J)) * COS(PHI(I)) / COS(XX)
IF (V(I) .LT. VZERO) GO TO 800
RYNEW = V(I) + ((-2.0 * V(I)) * PEEU(V(I)) * 0.0) * DELTAT
S = (YNFW - VZERO) * (YNFW + VZERO - 2.0 * V(J) * COS(PHI(I))) / (-2.0 * ADEC(I))
IF (RSTAR(I) .GT. RSTAR(I)) A(I) = 0.0
RETURN

250 CONTINUE
IST(I) = 2
IF (ABS(BETDOT(I)) .LT. BFUNCT(GCAP(I) * V(I)) * V(I)) GOTO 225
BETDOT(I) = SGN(BETDOT(I)) * BFUNCT(GCAP(I) * V(I)) * V(I)
IST(I) = 0

225 A(I) = AMIN1(ADEC(I), 32.2 / V(I) * PEEU(V(I)) * BETDOT(I) * I)
A(I) = AMIN1(A(I), (VMAX(I) - V(I)) / DELTAT)
IF (((V(I) + A(I)) * DELTAT) .LT. VZERO) AND (RSTAR(I) .GT. RSTAR(I)) A(I) = 0.0
RETURN

300 CONTINUE
CAPX = BFUNCT(AMIN1(GCAP(I), GMAX(I)) * V(I)) * V(I) * I
IMSTAT(I) = 2
IST(I) = 2
IF (ABS(BETDOT(I)) .LT. CAPX) GOTO 310
BETDOT(I) = SGN(BETDOT(I)) * CAPX
IST(I) = 0

310 CONTINUE
A(I) = (32.2 / V(I)) * PEEU(V(I)) * BETDOT(I) * I
A(I) = AMIN1(A(I), (VMAX(I) - V(I)) / DELTAT)
IF (ABS(BETDOT(I)) .EQ. CAPX) .AND. (V(I) .GE. VSTR(I)) A(I) = 0.
RETURN
600 CONTINUE
BETDOT(I) = BFUNCT(AMIN1(GCAP(I), GMAX(I)), V(I), I) * (V(I) / VSTR(I))
IMSTAT(I) = 10
A(I) = AMIN1(32.2 / V(I) * PEEU(V(I)), BETDOT(I), I) * (VSTR(I) - V(I)) / DELTAT
IF (V(I) .GT. VSTR(I)) BETDOT(I) = BFUNCT(GCAP(I), V(I), I)
IF (V(I) .LE. VSTR(I)) GO TO 604
A(I) = AMIN1(32.2 / V(I) * PEEU(V(I)), BETDOT(I), I) * ADEC(I)
IMSTAT(I) = 9
604 CONTINUE
IF (IPS(I), EQ, 1) GOTO 602
IF (ALPHA(I)) 602 * 601 * 603
IF (PHI(I) .LE. 0.) GOTO 603
BETDOT(I) = -BETDOT(I)
IMSTAT(I) = 0
RETURN
700 CONTINUE
A(I) = 0.
IST(I) = 1
BETDOT(I) = 0.
IMSTAT(I) = 0
RETURN
800 CAPX = BFUNCT(AMIN1(GCAP(I), GMAX(I)), V(I), I) * AMIN1(V(I) / VSTR(I), 1.)
IMSTAT(I) = 3
IF (R .GT. RSTAR(I)) GO TO 802
CAPX = BFUNCT(AMIN1(GCAP(I), GMAX(I)), V(I), I)
IMSTAT(I) = 4
YNEW = ((RSTAR(I) * BETDOT(J) / V(J)) ** 2
IF (YNEW .GT. 1.) GO TO 850
YNEW = V(J) * SQRT(1. - YNEW)
IF (YNEW .LT. VC(I)) YNEW = VC(I)
IF (ABS(BETDOT(I)) .GT. BFUNCT(AMIN1(GCAP(I), GMAX(I)), YNEW, I)) * GO TO 850
IF (YNEW .LT. V(I)) GO TO 825
A(I) = AMIN1(32.2 / V(I) * PEEU(V(I)), CAPX, I) * (YNEW - V(I)) / DELTAT)
GO TO 802
C

825 A(I)=AMAX1(I),YNEW-V(I),/DELTAT)
850 IF (ABSTDOT(I).LE.CAPX) GO TO 852
851 A(I)=AMIN1(I),YNEW-V(I),/DELTAT)
852 YNEW-V(I) = VSTR(I)
802 CONTINUE
804 CONTINUE
805 RETURN
END
SUBROUTINE RPHI

COMMON /Radar/ PI(2), TAU(2), P2(2), S(6,2), RI*RPHI, NUIND

COMMON /INPUT/ TMAX, TMIN, TSTAR, VA,

* N, NV(2), IPS(2), NID,

IV1(2), V2(2), V3(2), V4(2), W(2), TSTARS(2),

2AFC2(2), PDET(2), P2FF(2), POPT(2),

3RAS(2), RTRK(2), VC(2), V2(2), RSTAR(2), ALPDET(2),

4ALP1FF(2), ALP2FF(2), ALP3FF(2), VMAX(2), GP(2), ACC12,

5RF3T(15,6,2), RF2T(15,6,2), RF1T(15,6,2), R1PT(15,6,2),

6RF2PT(15,6,2), RF3PT(15,6,2), X1D(72), GMIS(6,2), VATAB(10),

7TALPH(6,2), MMAX(27,2), SIGTAB(15), NVA, NSIGMA, VATAG(27,2),

8NVAGC(2), RC(27,8,2), GBIG(27,2), TC(2), GOFRC(8,2),

9RF17(5,15,6,2), RF2T(5,15,6,2), RF3T(5,15,6,2),

10RF17P(5,15,6,2), RF2P(5,15,6,2), RF3P(5,15,6,2),

11XID(72), GMIS(6,2), VATAB(10),

12FIRE!T25, MISTPI(25), IAFTI(25), TFIRES(25,10), MISTPI(25,10),

* RP(25,10), LCA, LILTL,

 COMMON /INIFLT/T1, TPI, BTDOT(2), FNSW(6,2),

 LIST(2), V(2), I, J, M(6,2), R(2), X(2),

 DX(2), DZ(2), TAMARF, G(2), LEFSW,

 COMMON/GND/ GAMMA, ALPHA(2),

 COMMON/COMRA/I, J,

 COMMON/XOFTIR/ICAN(2), RNW(2),

 NEED INPUT COMMON /INIFLT/RADAR

 PI3, 14159265

 IF (ABS(ALPHA(J)-LT.PI/2.) AND (ABS(RPHI(J)-LT.PI/2.)) GOTO 50

 CONTINUE

 TRPHI =

 DIFF = ABS(RPHI(J) - SIGSTAR(LIST)),

 GO 10 LIST = 2, NSIGMA

 IF (ABS(RPHI(J) - SIGTAB(LIST)) < DIFF) GOTO 10

 TRPHI = LIST

 DIFF = ABS(RPHI(J) - SIGSTAR(LIST))

 CONTINUE

 VTST = (V(1) + V(J)) / 2.

 DO 20 LIST = 2, NVA

 ---
IF (VATAB(LIST).LT.VTEST) GO TO 20
IVAP=LIST
IVAM=LIST-I
GO TO 21
20 CONTINUE
IVAP=NVA
IVAM=NVA-1
DIMENSION TEST(6)
21 NMIS=NM(J)-1
DO 25 K=1,NMIS
ARGM=RFIT(IVAM.IPHI.K.J)
ARGP=RFIT(IVAP.IPHI.K.J)
25 TEST(K)=ARGX+(ARGP-ARGM)*(VTEST-VATAB(IVAM))/
            *(VATAB(IVAP)-VATAB(IVAM))
IF (NMIS.EQ.1) GO TO 40
RPHIJ=0.
DO 28 K=1,NMIS
IF (NUMIS(K,J).EQ.0) GO TO 28
IF (RPHIJ.GE.TEST(K)) GO TO 28
IJ=K
RPHIJ=TEST(K)
28 CONTINUE
IF (ABS(ALPHA(J)).GE.ALPMIS(IJ+J)) RPHIJ=0.
IF (G(J).GE.GMIS(IJ+J)) RPHIJ=0.
IF(RPHIJ.NE.0.) ICAM(J)=0
RETURN
40 RPHIJ=TEST(1)
IF (ABS(ALPHA(J)).GE.ALPMIS(1+J)) RPHIJ=0.
IF (G(J).GE.GMIS(1+J)) RPHIJ=0.
IF(RPHIJ.NE.0.) ICAM(J)=0
RETURN
50 CONTINUE
IF (R.LE.RNOW(J)) GO TO 5
RPHIJ=0.
RETURN
END
SUBROUTINE PASSIVCPASS
COMMON/DBLLNC/SIGMIS(6,2),KOUNTR(6,2),NUMIS(6,2),TAUMIS(6,2)
*TFIREI(25),MISTPI(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI
*RP(25,10),LCAP,LLITL
COMMON/RIFFC/IITEMO(2)
COMMON/ADRECO/RDOT,PHIDOT(2),ALPDOT(2),GPR
COMMON/RADA?/PI(2),TAU(2),P2(2),S(6,2),R1,RPHIJ,NUIND
COMMON/INPUTV/TMAX,TMIN,TSTAR,VA
1DELEPS=DELTAT,EPSON,SIGB,GT(3),RMIN,H
*NUM(2),IPS(2),NID,
IVI(2),V(2),V3(2),V4(2),W(2),TSTARS(2),
2ADEC(2),RDRT(2),RIFF(2),ROPT(2),
3RPAS(2),RTIK(2),VC(2),V(2),RSTAR(2),ALPD(2),
4ALPFF(2),ALPOPT(2),ALPPAS(2),ALPKG(2),VMAX(2),GP(2),AC(2),
5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(6,2),VATAB(10),
7ALPMX(6,2),GMATX(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
8NVAG(2),RC(27,9,2),GBIG(27,2),TC(2),GOFRC(8,2),
9NULOC(2)
COMMON/GRIDSO/GAMMA,ALPHA(2),R
COMMON/INIFLT/T,TPRT,BETDOT(2),ENVSW(6,2),
1IST(2),V(2),I,J,TMS(6,2),PHI(2),Y(2),X(2),
2DX(2),DY(2),TAWARE,G2(2),LEFTSW
COMMON/COMBA/1,J
COMMON/EXEC/BSMALL(2),RHO(4),IFTR,IBMR,FM(2),BSMALL(2),
LOGICAL LEFTSW,PASS
LOGICAL ENVSW
R1=TPK(J)
IF((J.EQ.1BM),AND.,(TAWARE.EQ.1000.)) GO TO 35
IF((J.EQ.1BM)) GO TO 40
IF((NUIND.EQ.0)) GO TO 40
IF((NUIND.EQ.1)) GO TO 5
RI=C
IF(IITEMP(J),E0.0) GO TO ?
CONTINUE
IF((PI(J),NE.1.)) GO TO 30
DO 25 L = 1, NM1
   IF (SIGMIS(L).GT.DELTAT) GO TO 25
   S(L) = -1.

25 CONTINUE
PASS = .FALSE.
RETURN

30 IF (P2(J).NE.1.) GO TO 31
   R1 = RTRK(J)
   GO TO 40
31 CALL RPHI
   IF (RPHI .LE. 0.) GO TO 35
   IF (RDOT .GE. 0.) GO TO 36
   1 = AMIN1(RTRK(J), RPHI - TAUC(J) * RDOT)
   GO TO 40
35 PASS = .FALSE.
   RETURN
36 IF (RPHI .LT. R) GO TO 35
   P1 = RTRK(J)

40 IF ((R .LE. RMAX) .AND. (R .LE. R1) .AND. (ABS(ALPHA(J)) .LE. ALPTRK(J)) .AND. (ABS(°H(J)) .LE. ALPPAS(J))) GO TO 10
9 PASS = .FALSE.
   RETURN
10 IF (IPS(I).EQ.1.) GO TO 15
   IF (IPS(I).NE.2) GO TO 9
73 CALL AWAPF
   PASS = .TRUE.
   P2(J) = 1.
   RETURN
15 LEFTSW = .TRUE.
   GO TO 13
END
GO TO 10
19 IF ((PI-ABS(ALPHA(I)))*LE. ALPGUN(I)) AND (R:LE. RANGUN(I))) GO TO 21
   RETURN
21  ITEMP(I)=1
         GO TO 12
11  MIS=NM(I)
      GO TO 10
END
*TFTC MAIN XR7, DECK
COMMON/NOWFIR, ICAN(2), RNOW(2)
COMMON/XPLTIM/
* EPSPLT(5), IGRPLT(5), PLOTIM, NOWPLT, KOUNT, XPLT(300+4), YPLT(300
*+4), XN, XX, YN, YXL, XRL, YLL, YUL, NP(4), IPLOT, TIPLOT, IERR
COMMON/COWARD/VPRIME(2) K D(11, 2) GCAP(2)
COMMON/INPUTV/TMAX, TMIN, TSTAR, VA
1DELEPS, DELTAT, EPSLON, RANGE, GT(3), RMIN, H
* N, NX(2), IPS(2), NID,
1V1(2), V2(2), V3(2), V4(2), W(2), TSTARS(2),
2ADEC(2), RINF(2), RROPT(2),
3PPAS(2), RTPK(2), VC(2), VZ(2), RSTAP(2), ALPDET(2),
4ALPIFF(2), ALPPOPT(2), ALPPAS(2), ALPTRK(2), VMAX(2), GP(2), AC(2),
5RPTT(5, 15, 6, 2), RF2T(5, 15, 6, 2), RF3T(5, 15, 6, 2), RF1PT(5, 15, 6, 2),
6RPTT(5, 15, 6, 2), RF2PT(5, 15, 6, 2), XI, D(T), GMIS(6, 2), VATAB(10),
7ALPMIS(6, 2), GMAXTB(2), 5GTAB(15), NVA, NSIGMA, VATA(27, 2),
8NVAG(2), RC(27, 2), GBIG(27, 2), TC(2), GOFCR(8, 2),
9USRC(2),
COMMON/EXEC/P, SMALL, RHO, IFTR, IBMR, FM(2), BSML(2),
COMMON/CMP, ICOMP
COMMON/PURSUF/A(2), BETMAX(2), THEDOT, PHISTR, ETA, ALPMAX(2), XLAMDA(2),
XPHEL
COMMON/D BLNRC SIGMIS(6, 2), KOUNTR(6, 2), NUMIS(6, 2), TAUMIS(6, 2),
* TFIRE(25), MISTPI(25), IAIFI1(25), TFIREP(25, 10), MISTPP(25, 10), IAIFI
PP(25, 10), LCAP, LITL
COMMON/SLOWUP/DECLF
COMMON/RADAR/P1(2), TAU(2), P2(2), S(6, 2), R1, RPHI, J, N1,
COMMON/REARG/RANGUN(2), ALPGUN(2)
COMMON/BEFIX/BETAB(27, 2), VSTR(2),
COMMON/COMP/ODCT/P(2), BINDEX(2), GB(2), SICKF(2), OX(2),
COMMON/TOTIME/TDELTS(3), RTEST(2),
COMMON/RVARY/KCAP, KLITL, TATRP(16, 2), TTHERE(2), MISPRP(16, 2), TOLER,
* RPRIME(6, 2)
COMMON/MAXALP/AM(2),
NAME LIST/SLODWN/DECLF
MAKE LIST/PURS/ALPMAX, XLAMDA, XPHI
NAME LIST/FIREM, TAUMIS, NUMIS, LCAP
CONTINUE
101 FORMAT(19X,110)
102 FORMAT(19X,5F10.0)
103 FORMAT(19X,5110)
READ (5,PURS)
READ (5,SLODON)
READ (5,FIRIME)
READ (5,TAIL)
READ (5,IMSIK)
READ (5,STAPUT)
ALPMAX(1)=ALPMAX(1)*RADIUS
ALPMAX(2)=ALPMAX(2)*RADIUS
XLANDA(1)=XLANDA(1)*RADIUS
XLANDA(2)=XLANDA(2)*RADIUS
ALPGUN(1)=ALPGUN(1)*RADIUS
ALPGUN(2)=ALPGUN(2)*RADIUS
WRITE (6,WRITAL)
SDFEPS=DEFLEPS
EPSLON=EPSLON+DEFLEPS*RADIUS
SIGN=SIGN*RADIUS
EPSLON=EPSLON*RADIUS
EPSLON=EPSLON+DEFLEPS*RADIUS
IF (DEFLEPS.GT.PIE) GO TO 10
IF ((EPSLON*EPSLON).LT.1.0) OR ((EPSLON*EPSLON).G.E.1.0) GO TO 20
EPSLON=0.
GO TO 10
EPSLON=-PIE
10 CONTINUE
NTEPS=(PIE-EPSLON)/DELZPS+1.
50 FORMAT(19X,110)
WRITE (11,50) HRUN,NTEPS,N,NM(1),NM(2)
WRITE (11,40) XID(ID),ID=1,MID
WRITE (11,60) ICONP
40 FORMAT(14X,12/(1PA6))
IF (RDET(IFTR).LT.0.0) GO TO 11
RSMALL=RDET(IFTR)
RHO=ALPDFT(IFTR)
GO TO 10
10   SMALL=ROPT(IFTR)
RHO=ALPDPT(IFTR)
15 CONTINUE
SMALL=MIN(SMALL,RANGE)
EPS=EPS+1
AM(1)=ALPMAX(1)
AM(2)=ALPMAX(2)
201 CONTINUE
CALL GRIDR
CALL CONTAM(EPS)
EPSLON=EPSLON+DELEPS
IF(EPSLON.LE.PF) GO TO 201
DO 25 11=1.2
ALPDFT(11)=ALPDFT(11)*RTD
ALPPF(11)=ALPPF(11)*RTD
ALPPRT(11)=ALPPRT(11)*RTD
ALPPAS(11)=ALPPAS(11)*RTD
ALPPTRK(11)=ALPPTRK(11)*RTD
ALPGUN(2)=ALPGUN(2)*RTD
ALPGUN(1)=ALPGUN(1)*RTD
DELEPS=DELEPS*RTD
EPSLON=EPSLON*RTD
RTD=SIGN(1)*RTD
ALPMAX(1)=ALPMAX(1)*PTD
ALPMAX(2)=ALPMAX(2)*PTD
XLMAX(1)=XLMAX(1)*RTD
XLMAX(2)=XLMAX(2)*RTD
DELEPS=DELEPS
EPSLON=EPSLON
WRITE(6,90)
IF (ICOND.EQ.0) GO TO 250
ROPT(IFTR)=ROPT(IFTR)/2.
ROPT(IFTR)=ROPT(IFTR)/2.
RSMALL=RSMALL*2.
250 CONTINUE
SUBROUTINE COMBAT(I EPS)

COMMON/INPUT/TMAX,TMIN,TSTAR,V1
COMMON/TAU/ICLC
COMMON/SICL,TIMS IC(2),SICTIM(2),25)
COMMON/PLTJNF/,EPSPLT(5),IGRPLT(5),PLOTIM,NOWPLT,KOUNT,XPLT(300)
COMMON/CX/RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2)
COMMON/NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2)
COMMON/NAVSY/GLAT(2),V7,FRO,ASMALL,RC1

LOGICAL AGAIN

NY1=NM(1)
NM2=NM(2)
10 CALL SOGRID
   DO 15 ADD=1,26
   TFIREP(IADD,IGRIDP)=0
   MISTP(IADD,IGRIDP)=0
   IAFIRP(IADD,IGRIDP)=0
15 CONTINUE
   ITMP(1)=0
   ITMP(2)=0
   IPRINT=0
   NOXPLT=0
   TIMPLT=0.
   KOUNT=0
   IF (IPLX.EQ.0) GO TO 13
   DO 11 I1=1,IPLX
   IF (EPSLON.NE.EPSPLT(I1)) GO TO 11
11 CONTINUE
   IF (IGRIDP.NE.IGRPLT(I1)) GO TO 13
13 CONTINUE
   ISHIFT=1
   DELTAT=TDELT(1)
   CALL INITFL
20 I=IFTR
   J=IRMX
   CALL NAVSYS
   VZEPRT(I)=VZERO
   SPRT(I)=5
   I=IRMX
   J=IFTR
   CALL NAVSYS
   VZEPRT(I)=VZERO
   SPRT(I)=5
   I=IFTR
   J=IRMX
   CALL CKWEAP
I=IBM
J=IFTR
CALL CKWFAO
CALL PRIFME
CALL ADRELC
CALL TIMERC
1=IFTR
CALL FINDO
2=IBM
CALL FINDO
CALL OVR(AGAIN)
IF (NOT, AGAIN) GO TO 20
CALL RESULT(IGRIDP, IEPS).
IF (IGRIDP, EQ, N) GO TO 12
IGRIDP=IGRIDP+1
GO TO 10
WRITE (11, 100) IEPS(IEPS), (TLIT(IP), AWARET(IP), FPR(IP), IP=1, N),
* ((TMIPI(MISS, 1, NGP), MISS=1, NM2), NGP=1, N),
* ((TMIPI(MISS, 2, NGP), MISS=1, NM2), NGP=1, N),
WRITE (11, 100) (TMIPI(MISS, 1, NGP), MISS=1, NM1), NGP=1, N),
*(TMIPI(MISS, 2, NGP), MISS=1, NM2), NGP=1, N),
DO 50 NGP=1, N
IF(NGP, EQ, 0) GO TO 50
IF(IFZER, EQ, 0) GO TO 46
IF(IFIZEREQ, IF, 1) GO TO 44
WRITE (11, 200) NGP, T Fit(1, WRIT, NGP), MISTPP(WRIT, NGP),
1 AFIR(1, WRIT, NGP), IFIZER
3 CONTINUE
50 CONTINUE
WRITE (11, 100) (SICIPI(1, NGP), SICTIM(2, NGP), NGP=1, N)
WRITE (11, 300) (IE(NGP), NGP=1, N)
200 FORMAT (1H, 1015)
200 FORMAT (1H, 15, E15.7, 3110)
100 FORMAT (1H, 4E16.8/((17X, 3E16.8))
RETURN
END
*%9F9C INITL XR7,DFCK
SUBROUTINE INITFL
COMMON/MULP/TLASTF(6,2)
COMMON/EXFC/RSMALL,RHQ,IFTR,IBMR,FM(2),BSMALL(2)
COMMON/CORPS/INDERT(2),BINDEX(2)
COMMON/RADAR/PI(2),TAU(2),P2(2),S1(6,2),R1,RPHIJ,NUMIND
COMMON/PRINT/PRINT
COMMON/TINOQP/XP(2),YP(2),BETA(2)
COMMON/GRAPH/JTOP,YTOP,YBOT
COMMON/CWFA/SIG(2),ANS(2,2),MIS
COMMON/SICLV/TIMIC(2),SICTM(2,25)
COMMON/NWRT/KNRT(2)
COMMON/VMOUT/VMRTAT,ALP,ALPOPT,ALPASC,ALPTPK(2)
COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
DELTA,DELTA,EPSON,SIGB,GT(3),RMINH
FNN,NN(2),EPS(2),NID,
1V1(2),VZ(2),V3(2),V4(2),X(2),TSTARS(2),
2DEFC,RDFT(2),RIEF(2),POPT(2),
2PPAS(2),RTPK(2),VC(2),VZ(2),MSTAR(2),ALPDET(2),
4ALPIFF(2),ALPOFF(2),ALPAS(2),ALPK(2),VMAX(2),GP(2),ACC(2),
5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF4T(5,15,6,2),RF5T(5,15,6,2),RF6T(5,15,6,2),
6RFPT(5,15,6,2),RF8T(5,15,6,2),RF9T(5,15,6,2),RF10T(5,15,6,2),RF11T(5,15,6,2),RF12T(5,15,6,2),
7ALPAS(6,2),GKAT(27,27,2),SIGTA(15,27,2),NVAR1,NSIGA,VATAG(27,2),
8NVAG(2),AC(27,2),SIG(27,2),TC(2),GOFRC(8,2),
9UNIFC(2),
COMMON/NIVAL/T,TERR,INTERT(2),ENVSW(6,2),
1ST(2),V1(2),I,JJ,TMIST(6,2),PHI(2),Y(2),X(2),,
2V(1),VY(2),TAMIFG(2),LEFTSW,
COMMON/CMPMJ,J
COMMON/DIRILNC/SIGM(6,2),KOUNTR(6,2),NUMR(6,2),TAUMIS(6,2),
*TFIRE(25),MSTP(25),IAFIRI(25),TFIREP(25,10),MSTPP(25,10),IAFI
*PAR(25,10),LCP,LJITL
COMMON/DIRLE/GAMMA,ALPHA(2),R
COMMON/VXEP/IPCAN(2,2),SNOW(2)
COMMON/TOT H/DELTAT(3)*TEST(2)

C0NT1NU "15HIFT=3

IF (R .LE. RTEST(2)) I5HIFT=3

IF (R .LE. "TEST"(2)) DELTAT=DELTAT(3)

TPRT=DELTAT

WRITE (6,200)

WRITE (3,293)

WRITE (2,190) I5HIFT=3

WRITE (3,293)

IF (R .LE. RTEST(2)) I5HIFT=3

IF (R .LE. "TEST"(2)) DELTAT=DELTAT(3)

TPRT=DELTAT

WRITE (6,200)

WRITE (3,293)

WRITE (2,190) I5HIFT=3

WRITE (3,293)

IF (R .LE. RTEST(2)) I5HIFT=3

IF (R .LE. "TEST"(2)) DELTAT=DELTAT(3)

TPRT=DELTAT

WRITE (6,200)
SIGMIS(IMIS,IK)=0.
KOUNTR(IMIS,IK)=0
70 FVSW(IMIS,IK)=.FALSE.
    LLITL=1
    DO 55 K=1,2
    NMIS=NM(K)
    P2(K)=0.
    DO 55 L=1,NMIS
    55 S(L,K)=0.
    DO 80 IK=1,25
    FIF(IK)=0.
    MISTP(IK)=0
    IAFIRI(IK)=0
    PANFIR(IK)=0.
    ALPFI(IK)=0.
    PHIFI(IK)=0.
    CONTINUE
70 IST(IFT)=1
    IST(IDMR)=1
    V(I[F])=VZ(IFTP)
    V(IDMR)=VZ(IDMR)
    !1=1BM
    JJ=IFTP
    DO 71 IK=1,2
    IFTP(IK)=0
    IMSTAT(IK)=0
    G(IK)=0.
    DO 60 K=1,24
    60 ANS(K+IK)=0.
    DO 61 K=1,24,4
    61 ANS(K+IK)=200.
    L=NM(K)
    DO 71 NMIS=1,L
    71   FVS(NMIS,IK)=0.
    DO 80 IK=1,2
    J1=2
    IF(I1.EQ.2) J1=1
IF(ABS(ALPHA(I1)) .GT. PI)  ALPHA(I1) = SGN(ALPHA(I1)) * (ABS(ALPHA(I1)) - PI)

90 IF(ALPHA(I1) .GT. 0.) GO TO 96

PHI(I1) = ALPHA(I1) + PI
GO TO 100

96 PHI(I1) = ALPHA(I1) - PI

100 CONTINUE

X(IBMR) = 0.
Y(IBMR) = 0.
Y(IFTR) = R * COS(PHI(IFTR) + PI/2.)
Y(IFTR2) = - R * SIN(PHI(IFTR) + PI/2.)
IPRINT = 1
DX(IFTR) = 0.
DY(IFTR) = V(IFTR) * DELTAT
SX(IBMR) = V(IBMR) * DELTAT * X*EPSLON
SY(IBMR) = V(IBMR) * DELTAT * Y*EPSLON
TAXFTE = 1000.
G(IFTR) = 1.
G(IBMR) = 1.
BETA(IBMR) = PI/2.
BETA(IFTR) = PI/2. - ALPHA(IFTR) + PHI(IFTR)
XTOP = MAX(X(IBMR) + X(IFTR))
XTOP = MIN(X(IBMR) + X(IFTR))
YTOP = MAX(Y(IBMR) + Y(IFTR))
YBOT = MIN(Y(IBMR) + Y(IFTR))
RETURN
END
IF (ABS(TMPLT).GT.0.001) GO TO 15
TMPLT=TMPLT+DELTA
IF (KOUNT.GE.300) GO TO 20
KOUNT=KOUNT+1
XPLT(KOUNT+1)=X(IBMR)/FTNM
YPLT(KOUNT+1)=Y(IBMR)/FTNM
GO TO 20
15 TMPLT=TMPLT-DELTA
20 CONTINUE
IF (TSTAR.EQ.0.) RETURN
TRT=TRT-DELTA
IF (TRT.GE.0.001) GO TO 10
TRT=TSTAR
XMACR=V(IBMR)/W(1)
XMACF=V(IFTR)/W(1)
YGR=INSTAT(IBMR)
YFR=INSTAT(IFTR)
CL(IFTR)/SG(IFTR)*SF(IFTR)*((2.**(DENS*V(IFTR)**2))**G(IFTR)
CL(IBMR)/SG(IBMR)*SF(IBMR)*((2.**(DENS*V(IBMR)**2))**G(IBMR)
ALPH=ALPHA(IBMR)*RPD
ALPF=ALPHA(IFTR)*RPD
PH12=PH1(IFTR)*RPD
PHIF=PH1(IFTR)*RPD
WRITE (6,100) T,X,Y,F
*F(RX(IBMR),Y(IBMR),V(IBMR),IMB,G(IBMR),*
*RDOT(IBMR),ALPHA*IST(IFTR),PHIF,KPRT(IBMR),*
*X(IFTR),Y(IFTR),V(IFTR),IMF,G(IFTR),*
*RDOT(IFTR),ALPF*IST(IFTR),PHIF,KPRT(IFTR)
100 FORMAT(150,F15.11)
   F7.0$I*F8.0$I*IX*F8.0$I*IX*F5.0*I*IX*13.1X*
   *F4.2*IX*F5.2*IX*F6.1*IX*11*IX*F6.1*IX*12*IX*
   *F8*0.1*IX*F8.0*1*IX*F5.0*1*IX*14*IX*F4.2*IX*F5.2*IX*F6.1*IX*11*IX*F6.1*
   *IX*12)
ETADEG=ETA*RPD
TFADG=THEDOT*RPD
SUBROUTINE CKFWP
COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
1DFPS,DELTAT,DOLON,SIGMA,GT(3),RMIN,H,
*NX(N),IP(N),NID,
1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
2ADEC(2),PDFT(2),RIPF(2),ROPT(2),
3RPAS(2),RTK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
4ALPIFF(2),ALPOLT(2),ALPPAS(2),ALPRTK(2),VMAX(2),GP(2),ACC(2),
5RF1T(5,15,6,2),RF2T(5,15,6,2),RF1PT(5,15,6,2),RF2PT(5,15,6,2),
6RF2PT(5,15,6,2),RF2PT(5,15,6,2),R2PT(15,6,3),GMIS(6,2),VATAB(10),
7ALPMIS(6,2),GMAT(7,2),SIGTAB(15),VNA,NSIGMA,VTAG(27,2),
8NVAG(2),NC(27,8,2),GR1G(27,2),TC(2),GOFRC(18,2),
*N:GRC(2)
COMMON/ALLNC/SIGMIS(6,2),KOUNTR(6,2),NUMIS(6,2),TAUMIS(6,2),
M:THRE(25),MSTPI(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI
*2(25,10),LCAP,LLITL
COMMON/GR1S,GAUGE,ALPHA(2),
COMMON/EXC/SMALL/RHO1,EMR,FM(2),RSMALL(7)
COMMON/INFLT/IXR,RT,S1F,RF1RT,RF2RT,RF1PT,RF2PT,ENVSW(6,2),
1ST(2),V(2),V1,1JJ,TMIS(6,2),TH1(2),V2,1X(2),
2X(2),DY(2),TAKARE,G1(2),LETFW
COMMON/CONNA/1,
*COMMON/ENRIR/ENRIR,ENRIR
COMMON/CKWEA/SIG(2),ANS(24,2)
*VIS
COMMON/RIEFFK/ITEFV(2)
COMMON/FRAP,FRANG,FRANG(2),FRANG(2),
COMMON/GRAND/OXDET(2),GRINDEX(2),GR(2),SIKEF(2),OX(2)
COMMON/SADAP/P(2),TAR(2),P2(2),SLT(2),R1,PHIJ,NUIND
COMMON/MULP/TLAST(6,2)
COMMON/FRWR/PANFIR(25),ALPFR(25),PHIFIR(25)
COMMON/NOKIP/IKAN(2),NOKIP(2)
COMMON/VARY/KCAP,KLITLP,TATRPR(6,2),THERE(2),MISPR(6,2),TOLER,
*PRIME(6,2)
DIMENSION RELOR(6)
LOGICAL ENVSW
DATA PI=3.1415927
RTD=67.2357705
RFLOOR(I)=3000.
RFLOOR(2)=4000.
IF (RDIF(I) .LE. 0.) GO TO 5
IF (GF.PPOP(I)) GO TO 11
IF (ABS(ALPHA(I)) .GE. ALPOPT(I)) GO TO 11
GO TO 6
5 CONTINUE
IF (RDIF(I) .LE. 0.) GO TO 11
IF (ABS(ALPHA(I)) .GE. ALPOPT(I)) GO TO 11
6 CONTINUE
IF (TFMP(I,k).EQ.1) GO TO 40
IF (TIFF(I,L).EQ.0) RETURN
IF(ALP(I,L).EQ.1) RETURN
IF (TFMP(I,k).EQ.1) RETURN
10 IF(ALPMIS(VIS*I).LT.ABS(ALPHA(I))) GO TO 20
IF(GMIS(MIS,I).LE.GM(I)) GO TO 20
IF(KOWNTR(MIS,I).GT.NUMIS(MIS,I)) GO TO 20
IF(MIS,FO,N1)) GO TO 19
SIGNAL=SGM(RET(I))+PH(I)
CALL PM5QP
IF (MIS,LT,3).AND.(R,LF,RFLOOR(MIS))) GO TO 20
IF(R,LF,PM) GO TO 20
IF(R,LF,PM) GO TO 20
12 CONTINUE
TW(MIS,I)=TW(MIS,I)+DELTAT
OX(I)=OX(I)-OXMFR(I)
SIGN(MIS,I)=SIGN(MIS,I)+DELTAT
IF(ENVGK(MIS,I)) GO TO 14
IF (SICKEF(I,LS).LT.1) GO TO 13
IF (OX(I,L).LT.EXP((TASTF(MIS,I)-T)/TAUMIS(MIS,I))) GO TO 20
13 CONTINUE
ENVSW(MIS, I) = .TRUE.
IMIS=4*MIS-3
ANS(I[MIS+1]) = T
ANS(I[MIS+2]) = ALPHA(I)*RTD
ANS(I[MIS+3]) = PHI(I)*RTD
P2(J) = 1.
ITEMP(J) = 1
IF(I.NE.IFR) GOTO 14
IF(TAWARE*NF.1000.) GOTO 14
TAWARE=T
TCAM(J) = 0
@STAR(I) = RPRIMF(2,J)
RSTAP(I) = RPRIMF(2,J)
14 CONTINUE
TOP=LASTF(MIS, I)-T
IF(KOUNTR(MIS, I)) .EQ. 0) GO TO 15
IF((T-LASTF(MIS, I)) .LT. TAUMIS(MIS, I)) GO TO 20
15 CONTINUE
IF(SICF(I, NE. 1.) GO TO 16
IF(CY(I, LT. EXP(TOP/TAUMIS(MIS, I))) GO TO 20
16 CONTINUE
IF(KOUNTR(MIS, I)) .GE. NUMIS(MIS, I)) GO TO 20
LASTF(MIS, I) = T
KOUNTR(MIS, I) = KOUNTR(MIS, I)+1
TFIRE[I, LLITL] = T
MSTPI(LLLTL) = MIS
IAFIR[I, LLLTL] = I
PHIFIR[I, LLLTL] = R
ALPFIR[I, LLLTL] = ALPHA(I)*RTD
PHIFIR[I, LLLTL] = PHI(I)*RTD
SIGMIS[MIS, I] = 0
LLITL=LLITL+1
20 IF(MIS .EQ. NUM(I)) RETURN
MIS=MIS+1
CALL OVER(AGAIN)
IF(.NOT. AGAIN) GO TO 20
CALL RESULT(IGRIDP,IEPS)
IF (NOWPLT.EQ.0) GO TO 14
DIMENSION MODE(4),ISYM(4)
DATA MODE(1),MODE(2),MODE(3),MODE(4),ISYM(1),ISYM(2),ISYM(3),ISYM(4)
14/3,3,4,4,0,0,1,2/
CALL PREDPT
IF (IFERR.GT.0) GO TO 14
CALL PLOT(XPLT,YPLT,2,4,NP,MODE,ISYM,XN,XX,YN,YX,XLL,XRL,YLL,*YUL,1,1,1)
14 CONTINUE
IF (IGRIDP.EQ.N) GO TO 12
IGRIDP=IGRIDP+1
GO TO 10
12 WRITE(11,100)EPS(IEPS),(TLINE(IP),AWARET(IP),FPR(IP),IP=1,N),
* (TMISP(MISS+1,NGP),MISS=1+NM1,NGP=1,N),
* (TMISP(MISS+2,NGP),MISS=1+NM2,NGP=1,N)
WRITE (11,100) ((TMISP(MISS+1,NGP),MISS=1+NM1,NGP=1,N),
* (TMISP(MISS+2,NGP),MISS=1+NM2,NGP=1,N)
DO 50 NGP=1,N
IFZERO=0
DO 45 IWRT=1,LCAP
IF(IFIREP(IWRT,NGP).GT.0) GO TO 44
IF(IFZERO.NE.0) GO TO 46
IFZERO=1
44 WRITE (11,200)NGP,IFIREP(IWRT,NGP),MISTPP(IWRT,NGP),
* IAIFIRP(IWRT,NGP),IFZERO
45 CONTINUE
46 CONTINUE
50 CONTINUE
WRITE (11,100) (SICTIM(1,NGP),SICTIM(2,NGP),NGP=1,N)
700 FORMAT(14,4*E15.7,3*16)
100 FORMAT(14*4F16.8/17X,3*16.8))
RETURN
END
SUBROUTINE RMISRP
COMM/INPUTV/ITMAX,TMIN,TSTAR,VA,
1DELS,DELTA,EPSLON,SIGB,G(3),RMIN,H,
<,NW(2),IP5(2),NID>,
1V(2),V(2),V(2),V(2),W(2),TSTARS(2),
2ADEC(2),RDEF(2),RIFF(2),ROPT(2),
3RPS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
4ALDIFF(2),ALPORT(2),ALPAS(2),ALPRK(2),VMAX(2),GP(2),ACC(2),
5DFT(5),15,6,2),RF2T(5),15,6,2),RF3T(5),15,6,2),RF1PT(5),15,6,2),
6RF2PT(5),15,6,2),RF3PT(5),15,6,2),XID(72),GMIS(6,2),VATAB(10),
7ALPMIS(6,2),GMAXT(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(18,2),
9NUGR(2)
COMM/INITL/T,TPRT,ETD3T(2),ENVSW(6,2),
LIST(2),V(2),11,JJ,TMIS(6,2),PHI(2),Y(2),X(2),
2DX(2),DY(2),TAWARE,G(2),LEFTSW
COMM/COMTA/1,J
COMM/RMISP/PMIS,RMISP
COMM/CKWFA/ SIG(2)*ANS(24,2)
*VIS
ISIG=1
DIFF=ABS(SIG(1)-SIGTAR(1))
DO 10 LIST=2,NVA
IF(ABS(SIG(1)-SIGTAB(LIST))*GE,D1FF) GO TO 10
ISIG=LIST
DIFF=ABS(SIG(LIST)-SIGTAR(LIST))
10 CONTINUE
VTEST=(V(1)+V(J))/2.
DO 20 LIST=2,NVA
IF (VATAB(LIST)*LT,VTEST) GO TO 20
IVAP=LIST
IVAP=LIST-1
GO TO 21
20 CONTINUE
IVAP=NVA
IVAP=NVA-1
21 TEST=SORT(G(J)**2-1.)
   IF(ABS(TEST-GL(1.))>LE.*001) GOTO 50
   IF (TEST<LT.GL(1.)) GO TO 25
   IF(TEST<LE.GL(2.)) GO TO 31
   IF (TEST<LE.GL(3.)) GO TO 26
25 IF (TEST<LT.GL(1.)) GO TO 30
   TEST=GL(3.)
   GK=GL(2.)
   RFKVAP=RF2T(IVAP,ISIG,MIS*I)
   RFLVAP=RF2T(IVAM,ISIG,MIS*I)
   RFLVAM=RF3T(IVAM,ISIG,MIS*I)
   RFLVPV=RF3PT(IVAM,ISIG,MIS*I)
   RFLPVV=RF3PT(IVAM,ISIG,MIS*I)
   GO TO 40
30 TEST=GL(1.)
31 GK=GL(1.)
   GL=GL(2.)
   RFKVAP=RF1T(IVAP,ISIG,MIS*I)
   RFLVAP=RF1T(IVAM,ISIG,MIS*I)
   RFLVAM=RF2T(IVAM,ISIG,MIS*I)
   RFLVPV=RF2PT(IVAM,ISIG,MIS*I)
   RFLPVV=RF2PT(IVAM,ISIG,MIS*I)
   GO TO 40
40 DIV=(VATR-IVAP)/(IVAM-VATAB(IVAM))
   ZK=RFKVAP-DIV*RFKVAM-REKVP
   ZKP=RFKPV-M-DIV*RFKPV-REKVP
   ZL=RFLVAM-DIV*RFLVAM-RFLVP
   ZLP=RFLVV-DIV*RFLVV-RFLPV
   MIS=ZK-(1.TEST-GK)/(GL-GK)*(ZK-ZL)
   MIS=ZKP-(1.TEST-GK)/(GL-GK)*(ZK-ZLP)
   RETURN
SUBROUTINE ADRCOR
COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
1DFEPS,DELTA,TPSLON,STBG,T(1),RMIN,H,
*V,M(2),IPS(2),*T,
1V1(2),V2(2),V3(2),V4(2),X(2),TSTAR(2),
2ADEC(2),ROET(2),RFF(2),ROPT(2),
3RPA5(2),RTRK(2),VCL(2),VZ(2),RSTAR(2),ALPDET(2),
4ALPRT(2),ALPAS(2),ALPTRK(2),MAX(2),GP(2),ACC(2),
5DFRPT(5,15,6,2),RF3P(5,15,6,2),RF1P(5,15,6,2),
6RF2P(5,15,6,2),RF3P(5,15,6,2),XID(2),GMIS(6,2),VATAB(10),
7ALPM(6,2),GMAX(27,2),SIGMA(15),NSIGMA,VATAG(27,2),
8NYAG(2),RG(27,R),GEIG(27,R),TC(2),GOFRC(3,2),
9NXAGC(2),
COMMON/PURSU/A(2),FETA(MAX(2),TETHD,T,ETAB,ALPMAX(2),XLAMDA(2),
X,XPH,
COMMON/ADRES/A,PHIDOT(2),ALPDOT(2),GPUR,
COMMON/GINDO/GAMMA,ALPHA(2),R,
COMMON/INFL/T,TPRT,BETDOT(2),ENVSW(6,2),
12S(2),V(2),I,J,T,MS(6,2),PHI(2),Y(2),X(2),
27X(2),DY(2),TAAMAR,G(2),LEFTSK,
COMMON/EXEC/BSMALL,R/K,IFTR,IFTR,FM(2),BSMALL(2),
COMMON/TING/FP(2),XP(2),BETA(2),
COMMON/TOTIME/DELT/3),RTEST(2),
COMMON/TW/T,ISHIT,
DATA R(2),IFTR,26/,
IF (DELTAT EQ TDELT(2)) GO TO 6,000
IF (DELTAT EQ TDELT(3)) GO TO 6,
IF (R .GT. RTEST(1)) GO TO 6
DELTAT=TDELT(2),
ISHIT=2,
5 CONTINUE,
IF (R .LE. RTEST(2)) ISHIF=3,
6 CONTINUE,
IF (R .LE. V(1)+IFTR) DELTAT=DELTAT(1) GO TO 100
\[
\begin{align*}
R &= \sqrt{(X(\text{IBMR}) - X(\text{IFTR}))^2 + (Y(\text{IBMR}) - Y(\text{IFTR}))^2} \\
\text{ALPHA} &= \text{ATAN2}(Y, X) - \text{BETA} \quad \text{END}
\end{align*}
\]
SUBROUTINE ACTIVE (ACT)

COMMON INPUTV/TMAX, TMIN, TSTAR, VA,

IDLP5, DELTAT, FSLOM, SIGRA, GT(3), RMIN, H,

N, V(2), IPS(2), NID,

V1(2), V2(2), V3(2), V4(2), W(2), TSTART(2),

2ADEC(2), RDF(2), RIF(2), ROPT(2),

3PAS(2), TRK(2), VC(2), VH(2), RSTAR(2), ALPDET(2),

4ALP(2), ALOPT(2), ALPPAS(2), ALPPAS(2), VMAX(2), GP(2), ACC(2),

5RF(5, 15, 6, 2), RF(5, 15, 6, 2), RF(5, 15, 6, 2), RFPT(5, 15, 6, 2),

6RFPT(5, 15, 6, 2), RFPT(5, 15, 6, 2), XID(72), GMIS(6, 2), VATAB(10),

7ALPAS(6, 2), GMAX(27, 2), SIGTAB(15), NVA, NSIGMA, VATAG(27, 2),

8NVAG(2), RC(27, 2), GBIG(27, 2), TC(2), GOFRC(8, 2),

9NUGRC(2),

COMMON/GP, DSO/GAMMA, ALPHA(2), R

COMMON/COMPA/I, J

LOGICAL ACT

IF((R, LE, RDF(1)) .AND. (ABS(ALPHA(I)) .LE. ALPDET(I))) GO TO 10

IF((R, LE, ROPT(1)) .AND. (ABS(ALPHA(I)) .LE. ALPDET(I))) GO TO 10

ACT = .FALSE.

RETURN

10 CALL AWARE

ACT = .TRUE.

RETURN

END
**STRETCH AWARE**  XR7, DECK

**SUBROUTINE AWARE**

**COMMON/PVARY/KCLP, KLTLP, TATRP(6,2)*THERE(2)*MISRPR(6,2)*TOLER**

**PPRIVS(4,2)**

**COMMON/INPUT/VMAX,TMIN,TSTAR,VA**

**1DELES,DELAT,EPSLON,SGB,GT(3),RMIN,H**

**SN,XT(2),IF5(2),NIF**

**1V1(2),V2(2),V3(2),V4(2),H(2),TSTARS(2)**

**2APR(2),RDE(2),RIF7(2),ROPT(2)**

**3LX(2),STK(2),VLC(2),V2(2),RSTAR(2),ALPF(2)**

**4APR(2),ALP(2),ALP(2),ALP(2),VMAX(2),GP(2),ACC(2)**

**5RTH(15,6,2),RTH(15,6,2),RF3T(15,6,2),RF1PT(15,6,2)**

**6RF2P(15,6,2),RF3PT(15,6,2),XID(72),GMIS(6,2),VATA(10)**

**7ALPMIS(6,2),GMAX(27,2),SIGTAR(15),VNA,NSIGMA,VATAG(27,2)**

**8NVAG(2),RA(27,8,2),GBIG(27,2),TC(2),GOFRC(6,2)**

**RNGRC(2)**

**COMMON/NCHFIP/ICAN(2),CNNO(2)**

**COMMON/EXEC/PBSmall, RHO, IFTR, IBMR, FM(2), BSMA(2)**

**COMMON/INFLT/1, 1PT, 1BEE, 1LEN(6), 1ENVSH(6)**

**LIST(2), V12(2), IJJ, TMIS(6,2), PHI(2), Y12(2), X12(2)**

**2DX(12),DY(12), TAWAKE(2), LEFTSW**

**COMMON/COM/, I,J**

**1F(1,1,1,1) RETURN**

**1F(TAMPAR,NF,1000) RETURN**

**TAWAF=T**

**PSTAR(1)=PPRIVS(1,1)**

**8PSTAR(1)=PPRIVS(2)**

**ICAN(J)=0**

**RETURN**

**END**
SUBROUTINE INFO
COMMON /INPUTV/TMAX,TMIN,TSTAR,VA,
1DFLPS,DELTA,TPSLON,SG,GT(9),AVMIN,H,
*V,NX(2),YPS(2),XID,
1V(2),VP(2),VP(2),V(2),W(2),TSTARS(2),
2APCE(2),RFRT(2),R1FF(2),ROPT(2),
3?PAS(2),RTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
4ALPIFF(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
5RFIT(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XID(72),GMIS(62),VATA(10),
7ALPMIS(62),GMXT(72),SIGTAR(15),NVA,NSIGMA,VATAG(27,2),
8NVAG(2),RC(27,2),GBIG(27,2),TC(2),GOFRC(8,2),
9MGRNC(2),
COMMON /INFLT/T,TPRT,BETDOT(2),ENSVW(6,2),
1ST(2),V(2),J,J,MT(36,2),PHI(2),Y(2),X(2),
2DX(2),DY(2),TAWAFG,GT(2),LEFTSV,
3COMMON/EXEC/RSMALL,RH0,IFTR,IPMR,FM(2),BSMALL(2),
4COMMON/COPBA/I,J,
COMMON /OLLNC, /SIGMIS(62),KOUNTR(6,2),NULIS(6,2),TAUMIS(6,2),
*IFIFI(25),VISIT(25),IAFIRI(25),TFIREP(25,10),MISTPP(25,10),IAFI,
*PP,10,10,1LCAP,LLITL,
COMMON/GETFIT/ETTAR(27,2),VSTR(2),
COMMON/3FFCE/ITEMP(2),
COMMON/COWARD/VPRI(2),K,D(11,2),GCAP(2),
*ISUMK=0,
*ISUMN=0,
MUYDO=N(1),
DO 1 LEAP=1,NMUYDO,
ISUMK=ISUMK+KOUNTR(LEAP),1,
ISUMN=ISUMN+NUMIS(LEAP),1,
CONTINUE,
LOGICAL ACT,PASS,
IF (.NOT.ACT) GO TO 10
CALL ACIVL(ACT),
IF (.NOT.ACT) GO TO 10
IF (ITEMP(1),EQ=1) GO TO 3
CALL PASSIV(PASS)
K=6
IF(PASS) RETURN
K=4
RETURN
2 CALL ACTIVE(ACT)
K=3
IF(ACT) RETURN
K=2
RETURN
3 CALL PASSIV(PASS)
K=7
IF(PASS) RETURN
K=5
RETURN
10 CALL PASSIV(PASS)
IF(.NOT.PASS) GOTO 11
K=0
IF(I TEMP(I).EQ.1) RETURN
K=8
RETURN
11 K=1
IF(I.EQ.IBMR).AND.(TAWARE.EQ.1000.) RETURN
K=10
BETAMI=(32.2*SORT(ARRAY(VSTR(I),I)**2-1))/VSTR(I)
BETAMJ=(32.2*SORT(ARRAY(VSTR(J),J)**2-1))/VSTR(J)
IF(BETAMI.GT.BETAMJ).AND.(I TEMP(I).NE.1)) RETURN
K=11
RETURN
END.
FUNCTION FUNCT(GARG, VARG, INDEX)

IF (GARG > VARG)
  RETURN

BFUNCT = 0.

BFUNCT = 32.2 / VARG * SQRT(GARG^2 - 1.)

END
SUBROUTINE OVER(Again)

COMMON/GRIDSO/GAMMA, ALPHA(2), R
COMMON/GRDSO/TLAST
COMMON/INFLT/T, TPT, BETDOT(2), ENVSW(6, 2)
1ST(2) * V(2), * T, J, TMIS(6, 2), PHI(2), Y(2), X(2)
2DX(2), 2DY(2), TAMAP, G(2), LFFTSW
COMMON/INPUTV/TMAX, TMIN, TSTAR, VA,
1NFLPS, DELTAT, EPSILON, SIGMA, GT(3), RMIN, H,
*VIMM(2), IPS(2), NIC,
1V1(2), V2(2), V3(2), V4(2), W(2), TSTARS(2),
2ADEC(2), RDET(2), RFF(2), ROPT(2),
3RPAS(2), RTRK(2), VC(2), VZ(2), RSTAR(2), ALPDET(2),
4ALPIFF(2), ALPOPT(2), ALPPAS(2), ALPTRK(2), VMAX(2), GP(2), ACC(2),
5RF1T(5, 15, 6, 2), RF2T(5, 15, 6, 2), RF3T(5, 15, 6, 2), RF1PT(5, 15, 6, 2),
6RF2PT(5, 15, 6, 2), RF3PT(5, 15, 6, 2), XID(72), GMIS(6, 2), VATAB(10),
7ALPMIS(6, 2), GMAXT(27, 2), SIGTAB(15), NVA, NSIGMA, VATAG(27, 2),
8NVAG(2), RC(27, 8, 2), GBIG(27, 2), TC(2), GOFRC(8, 2),
9MIGRC(2)
COMMON/EXEC/RSMALL, RHO, IFTR, IBMP, FM(2), BSMAILL(2)
COMMON/GRAPHO/XTOP, XBOT, YTOP, YBOT
LOGICAL AGAIN
IF (.MOD.(1ST(IFTR), 2) .EQ. 0) GO TO 5
IF (.MOD.(1ST(IBMP), 2) .NE. 0) GO TO 20
5 CONTINUE
IF (T .GT. TMAX) GO TO 10
11 CONTINUE
IF (R .LT. RMIN) GO TO 10
AGAIN = .FALSE.
RETURN
10 TLAST = T
AGAIN = .TRUE.
WRITE (6, 100) 1ST(IFTR), 1ST(IBMP), R
100 FORMAT (1H0, 4HFFT=*12, 10X, 4HBMRR=12, 10X, 2HR=* E15.7)
WRITE (6, 200) XBOT, XTOP, YBOT, YTOP
200 FORMAT (1H0, 8HX RANGE(*, F8.0, 1H, F8.0, 11H), Y RANGE(*
*F8.0, 1H, F8.0, 11H))
RETURN
IF(T.GT.TMIN) GO TO 10
GO TO 11
END
FUNCTION GCAPF(V1)
COMMON/CORPS/UXDEBT(2),BINDEX(2),GB(2),SICKF(2),OX(2)
COMMON/INPUTX/TVIP,TMIN,TSTAR,VA,
1DELPS,DELTAT,EPSSLO,SIGR,GT(3),RMIN,H,
*NM(2),IPS(2),NID,
1V1(2),V2(2),V3(2),V4(2),X(2),TSTARS(2),
2AVC(2),BDET(2),RFF(2),ROPT(2),
3RPAS(2),RPTRK(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
4RPAS(2),ALPT(2),ALPPAS(2),ALPBT(2),WMAX(2),GP(2),ACC(2),
5RFIT(5,15,6,2),RFIT(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
6RFINT(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),XID(72),GMIS(4,2),VATAG(10),
7ALPMIS(6,2),GMAXT(71,2),SIGTAB(11),NVAG,NSIGMA,VATAG(71,2),
8NVAG(2),RC(27,3,2),GBIG(27,2),TC(2),GOFRC(8,2),
9MUGRC(2)
NVADO=NVAG(1)
DO 11 I=1,NVADO
J1=11
IF(VATAG(I+1)-V) 1+6.2
1 CONTINUE
GCAPF=GRIG(NVADO+1)
GO TO 10
6 GCAPF=GRIG(J1+1)
GOTO 10
2 IF (J1+1.GT.1) GOTO 4
GCAPF=GRIG(J1+1)
GOTO 10
4 GCAPF=GBIG(J1-1,I)+GBIG(J1-1,I)-GBIG(J1-1,I)*
11V-VATAG(J1-1,I)/VATAG(J1-1,I)-VATAG(J1-1,I)
10 GCAPF=AQIN(GCAPF,GP(1))
IF (BINDEX(I).NE.1) RETURN
IF (GR(1).NE.1) RETURN
GCAPF=AQIN(GCAPF,1.5)
RETURN
END
*BFTC GFIND XR7,DECK
SUBROUTINE FIND
COMMON/PURSUE/A(2),BETMAX(2),THEDOT,PHISTR,ETA,ALPMAX(2),XLAMDA(2)
* XPHI
COMMON/CORPUS/OXDEBT(2),INDEX(2),GB(2),SICF(2),OX(2)
COMMON/CORPVX/A(1,J)
COMMON/INPUTV/TMAX,TMIN,TSTAV
10 DELEPS,DELTAT,EPSLON,SIGB,GT(3),RMIN,H
* N.XX(2),IPS(2),MID
IV(1),V(2),W3(2),V4(2),W12),TSTARS(2),
2ADFC(2),RFDT(2),RIFT(2),ROPT(2),
3RPA(2),TRK(2),VCI(2),V2(2),RSTAR(2),ALPDE(2),
4LPIFF(2),ALPTDP(2),ALPHA(2),ALPTR(2),VMAX(2),GP(2),ACC(2),
5RFT(5),15,6,2),RF3T(5,15,6,2),RF3T(5,15,6,2),RF3PT(5,15,6,2),
6RF3PT(5,15,6,2),RF3PT(5,15,6,2),XI(72),GMIS(6,2),VATAB(10),
7ALPM(6,2), جانب(27,2),SIGTAB(15),NYA,NSIGMA,VATAG(27,2),
8NVAG(2),RC(27,8,2),GBIG(27,2),TC(2),GOFRC(8,2),
9OGUROC(2),
COMMON/ADRECO/RDOT,PHIDOT(2),ALPDOT(2),GPRU
COMMON/INFILT/T,TRT,BETDOT(2),ENVSW(6,2)
1ST(2),V(2),J,X,MT(6,2),PHT(2),V(2),X(2),
2DX(2),DY(2),TAWAF,G(2),LEFT5W
COMMON/SCILV/TIMSC(2),SCTIM(2,2),
G(I)=SCTM((V(I)),RFDTDOT(1)),32,2,31+1)
IF ((G(I).LT.0.0.) AND (G(I).LT.0.0.)) GO TO 10
IF ((G(I).LE.0.0.) AND (G(I).LE.0.0.)) GO TO 5
IF (G(I).LE.0.0.) GO TO 5
IF (OXDEBT(1).GE.1.0.) TIMSC(I)=TIMSC(I)+DELTAT
RETURN
5 OXDEBT(I)=OXDEBT(I)+0.0167*DELTAT
IF (OXDEBT(I).GE.1.0.) TIMSC(I)=TIMSC(I)+DELTAT
IF (OXDEBT(I).LT.0.0.) RETURN
INDEX(I)=0
OXDEBT(I)=0
RETURN
10 OXDEBT(I)=OXDEBT(I)+DELTAT/(-23.*ALOG(1,25*G(I)-1.1))
IF (OXDEBT(I).LT.1.0.) RETURN
C

C

C

tmsic(1) = tmsic(1) + deltat
return
end
<table>
<thead>
<tr>
<th>Step</th>
<th>Command/Function</th>
<th>Input/Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>2</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>3</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>4</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>5</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>6</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>7</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>8</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>9</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>10</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>11</td>
<td>VATAG(J1,J1)-V1</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>12</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>13</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>14</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>15</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>16</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>17</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>18</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>19</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>20</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>21</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>22</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>23</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>24</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>25</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>26</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>27</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>28</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>29</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>30</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>31</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>32</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>33</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>34</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>35</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>36</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>37</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>38</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>39</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
<tr>
<td>40</td>
<td>VATAG(J1,J1)</td>
<td>VATAG(J1,J1)</td>
</tr>
</tbody>
</table>

*The above table represents a list of commands and functions with their respective inputs and variables. It appears to be a part of a computer program or algorithm.*
FUNCTION PFFij(VARG,RFTARG)  
COMMON/COWP,IPRIME(2),K,PI(11,2),GAPA(2)  
COMMON/INPUTK/TMAX,TMIN,STAR,VA,  
I,DELTAT,EPSON,SIGB,GT(3),RMIN,  
K,NM(2),PS(2),NID,  
V(2),V3(2),V4(2),W(2),TSTAR(2),  
2ADEC(2),REDF(2),ROPT(2),  
3PPAS(2),RTPK(2),VC(2),VZ(2),RSTAR(2),ALPDE(2),  
4ALP(2),ALPOPT(2),ALPPAS(2),ALPPTK(2),VMAX(2),GP(2),ACC(2),  
5REPT(5,15,6,2),RF2T(5,15,6,2),RE3T(5,15,6,2),RF1PT(5,15,6,2),  
6REPT(5,15,6,2),RE3PT(5,15,6,2),XID(7),GMIS(6,2),VATAR(10),  
7ALPMIS(6,2),GMXAT(27,2),SIGTAR(15),NVARNSIGMA,VATAG(27,2),  
BNVG(2),RC(27,8,2),GRIG(27,2),TC(2),GOFR(8,2),  
NUGRC(2)  
GARG=SVRT(1+(VARG*RETAIL)/2.2)**2  
NVADO=SVATG(1)  
NUMDO=NUGRC(1)  
DO 2 LIST=2,NVADO  
IF(VATAG(LIST,1),LT,VARG) GOTO 7  
IVAPL=LIST  
IVAMI=LIST-1  
GOTO 3  
2 CONTINUE  
IVAPL=NVADO  
IVAMI=NVADO-1  
3 DO 4 LIST=2,NUMDO  
IF(GOFR(1,LIST,1),LT,GARG) GOTO 4  
IGPL=LIST  
IGMI=LIST-1  
GOTO 5  
4 CONTINUE  
IGL=NUGDO  
IGMI=NUGDO-1  
5 TXXP=RC(IVAMI,IGMI,1)-(RC(IVAMI,IGPL,1)-RC(IVAMI,IGMI,1))  
**=(GARG-GOFR(IGMI,1),(GOFR(IGPL,1)-GOFR(IGMI,1)))  
TXXPP=RC(IVAPL,IGMI,1)-(RC(IVAPL,IGPL,1)-RC(IVAPL,IGMI,1))
XV7.DECK

*STFTC  BLK

BLOCK DATA
COMMON/EXEC/RSMALL+PH1+1FTR+1MAR+FM(2)+RSMALL(2)
DATA RSMALL+PHO+0+0+/ COMMON/INFLT/T+TPRT+BDEXT(2)+ENVSK(6+2)+
LIST(2)+V(2)+I+JJ+TIMS(6+2)+PHI(2)+Y(2)+X(2)+
2DX(2)+DY(2)+T+AMRE+G(2)+LEFTSW
DATA T+TPRT+BDEXT(1)+BDEXT(2)+V(1)+V(2)+PHI(1)+PHI(2)+Y(1)+Y(2)+
*X(1)+X(2)+DX(1)+DX(2)+DY(1)+DY(2)+1660+/ COMMON /ADREC/DPOT+PHIDOT(2)+ALPDOT(2)+GPUR
COMMON /INCOO/XP(2)+YP(2)+PM(n)+S
DATA XP(1)+XP(2)+YP(1)+YP(2)+PTA(2)+BETA(2)+660+/ COMMON /GPIDS/GAMMA+ALPHA(2)+R
DATA GAMMA+ALPHA(1)+ALPHA(2)+660+/ COMMON /NAVSY/GLAT(2)+VZERO+S+ASMALL+RC+FLAMDA+M+XC(2)+YC(2)
DATA GLAT(1)+GLAT(2)+VZERO+S+ASMALL+RC+FLAMDA+XC(1)+XC(2)+YC(1)+YC
1(2)+
*/160+/
END
SUBROUTINE GRIDP

COMMON/INPUTV/TMAX,TMIN,TSTAR,VA,
.DELPS,DELTAT,EPSLON,SIGMA,GT(3),RMIN,H,
*N,NS(2),IPS(2),NID,
1V1(2),V2(2),V3(2),V4(2),W(2),TSTARS(2),
2ADEC(2),RDET(2),RF(2),ROPT(2),
3RANS(2),RTRANS(2),VC(2),VZ(2),RSTAR(2),ALPDET(2),
4ALPJP(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
5RF1T(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
6RF2PT(5,15,6,2),RF3PT(5,15,6,2),XIDT2,GMIS(6,2),VATAG(10),
7ALPHIS(6,2),SMAXT(2,2),SIGMAM(15),NVA,NSIGMA,VATAG(27,2),
8AVAG(2),RC(27,8,2),SBG(27,2),TC(2),50FRC(8,7),
9NUCRC(2)

COMMON/GRIDP/VSTAR,FMU,THETB,THETC,9,C,
1FMX,FMIN,GR,QC,DELT,FSMALL,F

COMMON/EXEC/RSMALL,RHO,IFTR,IMR,FM(2),BSMALL(2)

COMMON/COMP/ICOMP

COMMON/GRIDSO/GAMMA,ALPHA(2),P

IF (ICOMP.EQ.0) GO TO 1

XN=N

DELT=(2.*RHO)/XN

GAMMA=RHO+.5*DFLTF

RRETURN

1 CONTINUE

VSTAR=SQRT(VZ(1IFTR)**2+VZ(1IMR)**2-2.*VZ(1IFTR)*
*VZ(1IMR)*COS(EPSLON))

IF (VSTAR.EQ.0.0) GO TO 10

FMU=ACOS((VZ(1IFTR)-VZ(1IMR)*COS(EPSLON))/VSTAR)

GO TO 11

10 FMU=0.

11 CONTINUE

IF (EPSLON.LE.0) FMU=-FMU

THETB=PV(FMU+RHO)

THETC=PV(FMU-RHO)

B=SMALL*XSIN(THETB)

C=SMALL*XSIN(THETC)
IF(OC.EQ.2.) GO TO TC 242
IF(OC.EQ.3.) GO TO 243
IF(B.GT.C) GO TO 244
FMIN=B
GO TO 260
241 FMIN=B
GO TO 260
242 FMAX=C
FMIN=B
GO TO 260
243 IF(R.GT.C) GO TO 260
FMAX=0.
FMIN=B
C=0.
GO TO 260
244 FMIN=C
GO TO 260
250 IF(OC.EQ.1.) GO TO 260
IF(OC.EQ.2.) GO TO 251
IF(OC.EQ.3.) GO TO 252
IF(B.LE.C) GO TO 260
FMAX=0.
FMIN=C
C=0.
GO TO 260
251 FMAX=C
GO TO 260
252 FMAX=0.
C=0.
260 DFLTF=(FMAX-FMIN)/FLOAT(N)
FSMALL=FMAX+.5*DFLTF
F=(FMAX-FMIN)*VSTAT/VZ19MR
RETURN
END
SUBROUTINE PREPPT
COMMON /PLTINF/
  EPSPLT(5), IGRPLT(5), PLOTIM, NOWPLT, KOUNT, XPLT(300*4), YPLT(300*4), XN*XN, YN*YN, X*X, XL*XL, YL*YL, NP(4), IPLOT, TIMPLT, IERR
XMAX=XPLT(1:1)
YMAX=YPLT(1:1)
DO 4 I=1*KOUNT
   DO 4 J=1*KOUNT
   IF (XMAX.GE.XPLT(I,J)) GO TO 3
   XMAX=XPLT(I,J)
3   IF (YMAX.GE.YPLT(I,J)) GO TO 4
   YMAX=YPLT(I,J)
4 CONTINUE
   IF (XMIN.LT.0.) GO TO 5
   XMIN=0.
   LXMIN=0
   GO TO 6
5   LXMIN=XMIN-1.
   XWIN=LXMIN
   IF (YMIN.LT.0.) GO TO 7
   YMIN=0.
   LYMIN=0
   GO TO 8
6   IF (YMIN.LT.0.) GO TO 7
   YMIN=0.
   LYMIN=0
   GO TO 8
7   LYMIN=YMIN-1.
   YMIN=LYMIN
   IF (XMAX.GE.XMIN+1) GO TO 50
   IF (XMAX.GE.XMIN+18) GO TO 100
   IF (XMAX.GE.XMIN+36) GO TO 200
   IF (XMAX.GE.XMIN+180) GO TO 2000
   IF ((LYMIN/10):EQ:0.LXMIN) GO TO 41
EQ.LXMAX) GO TO 44
GO TO 2000
LXM!N=LXMIN-1
GO TO 40
4!   IF (<LXMAX/10)*10.E0.LXMAX> GO TO 42
LXVAX=LX^AX+1
GO TO AT
4?   IF ( (LYMIN/in)*10 .FO.LYMIN) GO TO 43
LYMIN=LYMIN-1
GO TO 42
43 IF ((LYMAX/10)*30
LYMAX=LYMAX+1
GO TO 43
44 XMAX-LXMAX
XMlNaLXMIN
YMAX^LYMAX
YVlMaLYMlN
IF ((YMAX-YMIN).GT.250")
IF ((XMAX-XMIN).GT.180.)
LXWAX=LXMAX+130-(LXMAX-LXMIM)
LYN!AX = LYMAX+250-.LYMAX-LYMlN)
XMAX=LXMAX
YMAX=LY^AX
GO TO 500
IF (YRANGE.GT.50.) GO TO 40
IF ((LXMIN/2)*2.NE.LXMIN) LXMIN=LXMIN-1
IF ((LXMAX/2)*2.NE.LXMAX) LXMAX=LXMAX+1
XMAX=LXMAX
XMIN=LXMIN
IF ((XMAX-XMIN).GT.36.) GO TO 40
IF ((LYMIN/2)*2.NE.LYMIN) LYMIN=LYMIN-1
IF ((LYMAX/2)*2.NE.LYMAX) LYMAX=LYMAX+1
YMAX=LYMAX
YMIN=LYMIN
IF ((YMAX-YMIN).GT.50.) GO TO 40
LXMAX=LXMAX+6-(LXMAX-LXMIN)
LYMAX=LYMAX+50-(LYMAX-LYMIN)
XMAX=LXMAX
YMAX=LYMAX

WRITE (6,150) XMAX,XMIN,YMAX,YMIN
DRAW AXIS
DIMENSION XAXIS(2,2),YAXIS(2,2),NPAX(2),MODAX(2),ISYMAX(2)
IFERR=0
XAXIS(1,1)=0.
XAXIS(2,1)=0.
YAXIS(1,1)=YMIN
YAXIS(2,1)=YMAX
XAXIS(1,2)=XMIN
XAXIS(2,2)=XMAX
YAXIS(1,2)=0.
YAXIS(2,2)=0.
DATA NPAX(1),NPAX(2),MODAX(1),MODAX(2),ISYMAX(1),ISYMAX(2)
/7,2,*'3.1.4.7/"
CALL PLOT(XAXIS,YAXIS,2,2,NPAX,MODAX,ISYMAX,XMIN,XMAX,
*YMIN,YMAX+1.0,18.0,25.0,1.0,0.0,12:)
J=0
DO 180 J=1,KOUNT+10
J=J+1
XPLT(J,3)=XPLT(I1+1)
YPLT(J,4)=YPLT(I1+2)
YPLT(J,3)=YPLT(I1+1)
YPLT(J+1)=YPLT(J+2)

CONTINUE

NP(3)=J
NP(4)=J
YN=YMIN
YX=YMAX
XM=XMIN
XX=XMAX
XLL=0.
XRL=IR.
YLL=0.
YUL=25.
RETURN

1900 IERR=1
WRITE (6,182)

182 FORMAT (1HO,10X,37HPLLOT INFORMATION OUT OF PREDETERMINED,
*6H SCALE)

150 FORMAT (1HO,20X,16HPLLOT INFORMATION/21X,6HMAX X=, E15.7,5X,6HMIN X=,
*E15.7,5X,6HMAX Y=, E15.7,5X,6HMIN Y=, E15.7)
WRITE (6,151) XMAX,XMIN,YMAX,YMIN

151 FORMAT (1H,20X,6HMAX X=, E15.7,5X,6HMIN X=, E15.7,5X,6HMAX Y=, E15.7
*6HMIN Y=, E15.7)
RETURN
RETURN
END
SUBROUTINE INPUT
COMM/TIME/TDELTS(3),RTEST(2)
COMM/PURSUE/T(2),BETMAX(2),THEDOT,PHISTR, ETA,ALPMAX(2),XLAMDA(2)
COMMON/TOTIMEMAX,MINSTAR,DENS,
1DELEPS,DELTAT, EPSLON,RANGE,GT(3),RMIN,H,
*N, NM(7),IP5(5),NIP,
1TA(2),TR(2),SF(2),SFN(2),W(2),WGT(2),
2ADF(2),RDF(2),RIF(2),ROPT(2),
2RAP(2),RPTK(2),V(2),V(2),RSTAR(2),ALPMFT(2),
4ALPFI(2),ALPOPT(2),ALPPAS(2),ALPTRK(2),VMAX(2),GP(2), ACC(2),
5RT(5),15,6,7,5RT(5),15,6,7,15RT(5),15,6,7,15RT(5),15,6,7,
6RF2PT(5),15,6,7,RF2PT(5),15,6,7,XID(72),GMIS(5),VATAB(10),
7ALPMTS(6),15,GMAX(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
8NSAVG(2),RC(27,6,2),GIG(27,6,2),TC(2),GOFRC(6,2),
9NURG(2),
COMMON /COMP/ I COMP
COMMON /ETFLX/BFTTA8(27,2),VSTR(2)
COMMON /OWAND/VRIME(2),K,D(11,2),GCAP(2)
COMMON /ADAR/ P(2),TAU(2),P2(2),S(6,2),R1,RPHI,NUIND
COMMON /EXFC/SMALL,RHO,ITFR,IFMP,FM(2),BSMALL(2)
DIMENSION VAR(15),IVAP(6),TAB(50),FNAME(13)
1,X,NAMF(5),TANAME(50),TYNAME(3)
*LATG(2),INTS(6)
EQUIVALENT(TMAX,VAR),(N,IVAR),(TA,TAB2)
DATA(FNAME(1),I=1,13)/6HTMAX
*6HTMIN,6HTSTAR,6HDENS,6DELEPS,6DELTAT,
6HSLOAN,6HSRANGE,6HG1,6HG2,6HG3,6HRMIN,6HH,
6HTPS,6HST,6HTA,6HTB,6HTC,6HTD,6HTF
6HSR,6HSAP,6HSF,6HR,6HGT,6HADEC,6HRDE,6HRDEFF,
6HRDEPP,6HRDEP,6HRDEPP,6HRDEPP,6HRDEPP,
6HRDEPP,6HRDEPP,6HRDEPP,
DATA (INTS(1),I=1,15)/6HT1,6HT2,6HT3,6HT4
NAME LST/NAME/TAP,F1,NUIND
NAME LST/NAME/TAP/F1,NUIND
NAME LIST/GTAB/GMAXT/latg/vatag/nvag/rc/tc/GBIG
   *GOFRC, NUGRC, D
C READ 2*3 DIMENSION TABLES
DATA (TYNAMF(I), I=1,3)/6HFL0PT, 6HINVAR, 6HTAB1 /
   RFAD (5,117), NM(1), NM(2)
116 FORMAT(10X$I, 157)
   DO 180 I=1,2
      J1=NM(I)
180 RFAD (5,171) (ALPMIS(J, I), J=1, J1)
   DO 181 I=1, 2
      J1=NM(I)
181 RFAD (5,171) (GMIS(J, I), J=1, J1)
C READ TABLES5
READ (5,117) NVA, NSIGMA
READ (5,171) (VATAR(I), I=1, NVA)
READ (5,171) (SIGTAR(I), I=1, NSIGMA )
READ (5,117) TRMIS
DO 1544 I=1, 2
   J1=NM(I)-1
   DO 1544 J=1, J1
   DO 1544 K=1, NSIGMA
   IF (I.EQ.1) GO TO 154
   IF (IBMIS.EQ.0) GO TO 154
   DO 1544 L=1, NVA
1541 RF1T(L,K,J,Z)=RF1T(L,K,J,1)
   GO TO 1544
154 READ(5,171) (RF1T(L,K,J,1), L=1, NVA)
1544 CONTINUE
171 FORMAT(10X$I, 3F10.0)
   DO 1554 L=1, 7
      J1=NM(I)-1
      DO 1554 J=1, J1
      DO 1554 K=1, NSIGMA
      IF (1.EQ.1) GO TO 155
      IF (IBMIS.EQ.0) GO TO 155
      DO 1554 L=1, NVA
1551 RF2T(L,K,J,Z)=RF2T(L,K,J,1)
GO TO 155A
155 READ(5,*171)(RF2T(L,K,J,I),L=1,NVA)
1554 CONTINUE
DO 1564 I=1,2
   J1=NM(I)-1
   DO 1564 J=1+J1
   DO 1564 K=1,NSIGMA
   IF (I.EQ.*1) GO TO 156
   IF (IVMS*EQ.*0) GO TO 156
   DO 1561 L=1,NVA
   1561 RF3T(L,K,J+2)=RF3T(L,K,J+1)
   GO TO 1564
156 READ(5,*171)(RF3T(L,K,J,I),L=1,NVA)
1564 CONTINUE
DO 1574 I=1,2
   J1=NM(I)-1
   DO 1574 J=1+J1
   DO 1574 K=1,NSIGMA
   IF (I.EQ.*1) GO TO 157
   IF (IVMS*EQ.*1) GO TO 157
   DO 1571 L=1,NVA
   1571 RF1PT(L,K,J+2)=RF1PT(L,K,J+1)
   GO TO 1574
157 READ(5,*171)(RF1PT(L,K,J,I),L=1,NVA)
1574 CONTINUE
DO 1584 I=1,2
   J1=NM(I)-1
   DO 1584 J=1+J1
   DO 1584 K=1,NSIGMA
   IF (I.EQ.*1) GO TO 158
   IF (IVMS*EQ.*2) GO TO 158
   DO 1581 L=1,NVA
   1581 RF2PT(L,K,J+2)=RF2PT(L,K,J+1)
   GO TO 1584
158 READ(5,*171)(RF2PT(L,K,J,I),L=1,NVA)
1584 CONTINUE
DO 1594 I=1,2

J1=MM(1)-1
DO 1504 J=1,J1
DO 1504 K=1,N5IGMA
IF (L.EQ.1) GO TO 159
IF (IRMS.EQ.0) GO TO 159
DO 1591 L=1,NVA
1591 RF3PT(L*K,J,2) =RF3PT(L*K,J,1)
GO TO 1594
1594 CONTINUE
117 FORMAT(19X,2116)
READ (5,171) PTEST(1),PTEST(2),(TDFLTS(M),M=1,3)
ENTRY INPUT
READ (5,115), NID,MID, I=1,NID)
115 FORMAT(19X,15/(12A6))
C SECOND ENTRY READ ID
READ (5,110) NUTYP
IF(NUTYP.EQ.0) GO TO 301
NO 300 M=1,NUTYP
READ (5,111) TYPE,NUM
110 FORMAT(19X,110)
111 FORMAT(AA,13X,110)
DO 200 L=1,3
IF(TYPE.NE.TYNAME(L))GO TO 200
NGO=L
GO TO 201
200 CONTINUE
WRITE(6,202) TYPE
202 FORMAT(1H0,10X,12HNUMBER TYPE *A6,14H IS NOT INCLUDED IN THE ATAC-2 INPUT ROUTINE.*
221PROGRAM DISCONTINUED*)
STOP
201 GO TO (205,225,250);NGO
C205 FLOATING PT* VARIABLES
205 DO 220 L=1,NV
READ (5,112) VARNAM,VALUE
112 FORMAT(A6,13X,510,0)
DO 215 K = 1, N
  IF (VAR(K) .NE. FNAME(K)) GO TO 215
  CONTINUE

GO TO 200

WRITE (6, 201) VAR(K), VALUE
1 FOR A (1) OF X
2 FOR A (2) OF X
3 FOR A (3) OF X
4 FOR A (4) OF X
5 FOR A (5) OF X
6 FOR A (6) OF X
7 FOR A (7) OF X
8 FOR A (8) OF X
9 FOR A (9) OF X
10 FOR A (10) OF X
11 FOR A (11) OF X
12 FOR A (12) OF X
13 FOR A (13) OF X
14 FOR A (14) OF X
15 FOR A (15) OF X
16 FOR A (16) OF X
17 FOR A (17) OF X
18 FOR A (18) OF X
19 FOR A (19) OF X
20 FOR A (20) OF X
21 FOR A (21) OF X
22 FOR A (22) OF X
23 FOR A (23) OF X
24 FOR A (24) OF X
25 FOR A (25) OF X
26 FOR A (26) OF X
27 FOR A (27) OF X
28 FOR A (28) OF X
29 FOR A (29) OF X
30 FOR A (30) OF X
31 FOR A (31) OF X
32 FOR A (32) OF X
33 FOR A (33) OF X
34 FOR A (34) OF X
35 FOR A (35) OF X
36 FOR A (36) OF X
37 FOR A (37) OF X
38 FOR A (38) OF X
39 FOR A (39) OF X
40 FOR A (40) OF X
41 FOR A (41) OF X
42 FOR A (42) OF X
43 FOR A (43) OF X
44 FOR A (44) OF X
45 FOR A (45) OF X
46 FOR A (46) OF X
47 FOR A (47) OF X
48 FOR A (48) OF X
49 FOR A (49) OF X
50 FOR A (50) OF X
51 FOR A (51) OF X
52 FOR A (52) OF X
53 FOR A (53) OF X
54 FOR A (54) OF X
55 FOR A (55) OF X
56 FOR A (56) OF X
57 FOR A (57) OF X
58 FOR A (58) OF X
59 FOR A (59) OF X
60 FOR A (60) OF X
61 FOR A (61) OF X
62 FOR A (62) OF X
63 FOR A (63) OF X
64 FOR A (64) OF X
65 FOR A (65) OF X
66 FOR A (66) OF X
67 FOR A (67) OF X
68 FOR A (68) OF X
69 FOR A (69) OF X
70 FOR A (70) OF X
71 FOR A (71) OF X
72 FOR A (72) OF X
73 FOR A (73) OF X
74 FOR A (74) OF X
75 FOR A (75) OF X
76 FOR A (76) OF X
77 FOR A (77) OF X
78 FOR A (78) OF X
79 FOR A (79) OF X
80 FOR A (80) OF X
81 FOR A (81) OF X
82 FOR A (82) OF X
83 FOR A (83) OF X
84 FOR A (84) OF X
85 FOR A (85) OF X
86 FOR A (86) OF X
87 FOR A (87) OF X
88 FOR A (88) OF X
89 FOR A (89) OF X
90 FOR A (90) OF X
91 FOR A (91) OF X
92 FOR A (92) OF X
93 FOR A (93) OF X
94 FOR A (94) OF X
95 FOR A (95) OF X
96 FOR A (96) OF X
97 FOR A (97) OF X
98 FOR A (98) OF X
99 FOR A (99) OF X
WRITE(6,204) VAR,NAM, VAL1, VAL2
204 FORMAT(1HC,10X,36H THE FOLLOWING FLT. PT. TABLE IS NOT 
137H INCLUDED IN THE ATAC-2 INPUT ROUTINE/20X, 
2A6*13X*2F10.7/20X, 21H PROGRAM DISCONTINUED./) 
STOP 
275 CONTINUE 
300 CONTINUE 
301 CONTINUE 
READ(5,110) ICOMP 
C WRITE OUT INPUT DATA 
WRITE(6,130) 
130 FORMAT(54X,25H INPUT FOR THE ATAC-2 MODEL) 
WRITE(6,119) (XID(I),I=1,NID) 
119 FORMAT(1HC,10X,12A6) 
WRITE(6,122) XNAMEF(1),TVAR(1),XNAMEF(3),TVAR(2), 
* TVAR(3),XNAMEF(5),TVAR(4),TVAR(5) 
122 FORMAT(1HC,4X,A6,3X,I12/(49X,A6,3X,2I12)) 
WRITE(6,120) (FNAMEF(I),VAR(I),I=1,13) 
120 FORMAT(1HC,4X,A6,3H = &F12.3/(49X,A6,3H = &F12.3)) 
WRITE(6,121) (NAMEF(I),TAB2(I),TAB2(I+1),I=1,46,2) 
121 FORMAT(1HC,4X,A6,3X,2F12.3) 
C WRITE TABLES 
DO 160 I=1,2 
160 WRITE(6,123)I,J,(SIGTAR(K),K=1,NSIGMA) 
DO 80 J=1,N1 
80 WRITE(6,80) 
160 WRITE(6,124)VA1TAB(L),RFIT(L,K,J,I),K=1,NSIGMA 
123 FORMAT(1HC,5X,13HRF1(VA,SIGMA)/52X,RHA/C NO. II,5X, 
111HWEAPON NO. I1/9H SIGMAS,15FR.2) 
124 FORMAT(1HC,16FR.0) 
125 FORMAT(1HC,5X,13HRF2(VA,SIGMA)/52X,RHA/C NO. II,5X, 
111HWEAPON NO. I1/9H SIGMAS,15FR.2) 
126 FORMAT(1HC,5X,13HRF3(VA,SIGMA)/52X,RHA/C NO. II,5X, 
111HWEAPON NO. I1/9H SIGMAS,15FR.2)
127 FORMAT(1HO,5X,14H'REF1-(VA,SIGMA)/$X,8HA/C NO. 1L,5X*)
111WEAPON NO. *11/0H SIGMAS *15FB*?1
128 FORMAT(1HO,5X,14H'REF2-(VA,SIGMA)/$X,8HA/C NO. 1L,5X*)
111WEAPON NO. *11/0H SIGMAS *15FB*?1
129 FORMAT(1HO,5X,14H'REF3-(VA,SIGMA)/$X,8HA/C NO. 1L,5X*)
111WEAPON NO. *11/0H SIGMAS *15FB*?1
DO 160 I=1,2
J1=NM(I)-1
DO 161 J=1,J1
WRITE (6,125) I,J,(SIGTAB(K) *K=1*NSIGMA)
WRITE(6*80)
DO 161 L=1,NVA
161 WRITE (6,124) VATAR(L),(RF2T(L,K,J)+1) *K=1*NSIGMA)
DO 162 I=1,2
J1=NM(I)-1
DO 162 J=1,J1
WRITE (6,126) I,J,(SIGTAB(K) *K=1*NSIGMA)
WRITE(6*80)
DO 162 L=1,NVA
162 WRITE (6,124) VATAR(L),(RF3T(L,K,J)+1) *K=1*NSIGMA)
DO 163 I=1,2
J1=NM(I)-1
DO 163 J=1,J1
WRITE (6,127) I,J,(SIGTAB(K) *K=1*NSIGMA)
WRITE(6*80)
DO 163 L=1,NVA
163 WRITE (6,124) VATAR(L),(RF1PT(L,K,J)+1) *K=1*NSIGMA)
DO 164 I=1,2
J1=NM(I)-1
DO 164 J=1,J1
WRITE (6,128) I,J,(SIGTAB(K) *K=1*NSIGMA)
WRITE(6*80)
DO 164 L=1,NVA
164 WRITE (6,124) VATAR(L),(RF2PT(L,K,J)+1) *K=1*NSIGMA)
DO 165 I=1,2
J1=NM(I)-1
DO 165 J=1,J1
WRITE (6,129) I,J,(SIGTAB(K),K=1,NSIGMA)
WRITE (6,80)
DO 165 L=1,NVA
165 WRITE (A,124) VATAB(L),(RF3PT(L,K,J,I),K=1,NSIGMA)
READ (5,RADON)
READ (5,NUTAB)
IF(MORE.EQ.0) GO TO 350
READ (5,GTAB)
DO 500 I=1,7
NVAD=NVAG(I)
IF(LATS(I).EQ.0) GO TO 450
DO 400 J=1,NVAD
400 GMAXT(J,I)=SORT(GMAXT(J,I)**2+1)
450 DO 470 J=1,NVAD
IF (GMAXT(J,I) .GT. 8.) GMAXT(J,I)=8.
BETTAB(J,I)=0.
TEMP=AMINS(GMAXT(J,I),GBIG(J,I))
IF (TEMP.LE.1.) GO TO 470
BETTAB(J,I)=(32.*SORT(TEMP**2-1.))/(VATAG(J,I)**W(I))
470 VATAG(J,I)=W(I)*VATAG(J,I)
VSTR(I)=VATAG(J,I)
TABMAX=BETTAB(J,I)
DO 480 J=2,NVAD
IF (BETTAB(J,I).LE.TABMAX) GO TO 480
VSTR(I)=VATAG(J,I)
TABMAX=BETTAB(J,I)
480 CONTINUE
VPRIME(I)=0.
TABMAX=0.
DO 490 J=1,NVAD
490 BETG=0.
IF (GBIG(J,I).GT.1.) BETG=(32.*SORT(GBIG(J,I)**2-1.))/VATAG(J,I)
IF (BETG.LE.TABMAX) GO TO 490
TABMAX=BETG
VPRIME(I)=VATAG(J,I)
CONTINUE
CONTINUE
450 CONTINUE
101 FORMAT(1HO*6X*36HENERG~S MANUVERABILITY TABLES FOR AC=12/
*56X*17HG-5 FOR RC TABLES/41X*8F10.3)
107 FORMAT(1H4*12F10.3)
103 FORMAT(1HO*3X*ZHVA*8X*5HGMAXT*3X*5HBETDOT*5X*4HGBIG*18X*4HRC=S)
DO 550 I=1+2
NVAD0=NVAG(I)
NGDO=NGRC(I)
WRITE (6,101) I,(GOFRC(J1+I),J1=1+NGDO)
WRITE (6,102) I,GOFRC(J1+I)
DO 550 I1=1+NVADO
WRITE (6,102) VATAG(I1+I),GMAXT(I1+I),BETTAB(I1+I),
= (I1+I)*(?C(I1+J1)+J1=1+NGDO)
550 CONTINUE
WRITE (6,574)
DO 575 I=1+2
J1=NM(I)
575 WRITE (6,109) I,(ALPMIS(J1)+J=1+J1)
574 FORMAT (1HO*57X*19HALPMIS(A/C*MISSILE))
109 FORMAT (7X*11+6X*10F12.7)
WRITE (6,109)
102 FORMAT (1HO*57X*17HGMIS(A/C*MISSILE))
108 FORMAT (7X*11+6X*6F11.2)
J1=NM(I)
IF(NY(I)+LT+NM(2))J1=NM(2)
WRITE (6,106) (IKTS(I1)+I1=1+J1)
106 FORMAT (4X*7HMISSEP*6X*6(I1+10X))
WRITE (6,107)
107 FORMAT (4H A/C)
DO 10 I=1+2
J1=NM(I)
10 WRITE (6,108) I,(GMIS(J1)+J=1+J1)
WRITE (6,105)
105 FORMAT (1H1)
RETURN
END
SUBROUTINE S0GRTD
COMMON/EXEC/RSMALL,RHO,IFTR,IR,FM(2),BSMALL(2)
COMMON/GRIDQ/GAMMA,ALPHA(2),R
COMMON/GRIDP/VSTAR,FMU,THETB,THETC,B+C
IFMAX,FMIN,GB,GC,DELTIF,FSMALL,F
COMMON/INPUTV/TMAX,TMIN,TSTAR,VA+
1DELPFS,DELTAT,FPSLON,SIGB,GT(3),RMIN,H,
N,NM(2),TPS(2),NSD,
TV(2),V2(2),V3(2),VA(2),W(2),TSTARS(2),
2ADEC(2),RDIF(2),RIFF(2),ROPT(2),
3RPAS(2),RTK(2),VC(2),VZ(2),RSTAR(2),ALPDE(2),
4ALPFF(2),ALPOPT(2),ALPAS(2),ALPTRK(2),VMAX(2),GP(2),ACC(2),
5RFIT(5,15,6,2),RF2T(5,15,6,2),RF3T(5,15,6,2),RF1PT(5,15,6,2),
6RF2PT(5,15,6,2),RF3PT(5,15,6,2),X1D(72),GMI(6,2),VATAB(10),
7ALPMIS(6,2),GMAXC(27,2),SIGTAB(15),NVA,NSIGMA,VATAG(27,2),
8NVAG(2),RC(27,8,2),G61G(27,2),TC(2),GOFRC(8,2),
9NUGRC(2)
COMMON/COMP/ICOMP
DATA PT/3.14159265/
DATA TOL/0.000001/
IF ICOMP.EQ.0 GO TO 1
GAMMA=GAMMA-DFLTF
ALPHA(IFTR)=GAMMA
R=RS0MALL
GO TO 10?
1 CONTINUE
FSMALL=FSMALL-DFLTF
GAMMA=ASIN(FSMALL/RSMALL)
IF1((FSMALL-C),GT,TOL) GO TO 100
IF(FSMMALL-B),TOL,LT,0) GO TO 100
IF(FSMAINL,GT,0) GO TO 102
ALPHA(IFTR)=RHO
GO TO 102
100 ALPHA(IFTR)=PV(FMU-GAMMA)
R=RSMALL
GO TO 103
FUNCTION SGN(X)
  IF (X.LT.0) SGN=-1.
  RETURN
END
FUNCTION PV(THETA)
PI=3.14159265
PV=THETA
IF(ABS(THETA).GT.PI) GO TO 1
RETURN
1 IF(THETA.GT.0.) GO TO 2
PV=THETA+2.*PI
RETURN
2 PV=THETA-2.*PI
RETURN
END
PROBTC MYSIN XR7.DECK
FUNCTION XSIN(X)
DATA TOL/.00001/
XSIN=SIN(X)
IF(ABS(XSIN).GT.TOL) GO TO 1
XSIN=0.
RETURN
1 IF(ABS(ABS(XSIN)-1.).GT.TOL) GO TO 2
XSIN=SIGN(1.,XSIN)
2 RETURN
END
$1RFTC MYCOS XR7,DECK
FUNCTION XCOS(X)
DATA TOL/0.00001/
XCOS=COS(X)
IF (ABS(XCOS).GT.TOL) GO TO 1
XCOS=0.
RETURN
1 IF (ABS(ABS(XCOS)-1.).GT.TOL) GO TO 2
XCOS=SIGN(1.,XCOS)
2 RETURN
END
SUBROUTINE EPSCAL
COMMON/EPSCA/PD(73),PU(73),P(2,2,73),PZ(2,73),GRIDP
COMMON/EXEC/NFPS,IFTR,IBMR,TMAXBF
*+TMAX
COMMON/COMP/ICOMP
COMMON/INPUTC/D,RHOTC(2),M(2),IW(6,2),
1PKP(6,2),N,XRUMC,NDPMD,DPMD(72)
COMMON/PCGCAL/PCL,PKG,PKE,PCCEPSN(73,25),PCCEPS(73)
COMMON/INPUTC/NRUN,NID,XID(72),NTEPS,EPSON(73)
1NN,TSMALL(25),TAMARE(25),F(25),T(6,2,25)
COMMON/GRIDD/1,J,LIM,J,LIMC,1U,A PK(6,2)
COMMON/PJUC/PV(2,2,73,25),PPZ(2,73,25)
COMMON/SICLV/TIM5IC(2),SICTIM(2,25),SICEPS(2,5)
DO 1 I=1,2
SICEPS(12,NEPS)=0.
212,NEPS)=0.
DO 1 J=1,2
P(12,J,NEPS)=0.
NU=0
DO 11 I=1,N
GRIDP=J
PD(NEPS)=1.
IF (ICOMP.EQ.0) PD(NEPS)=1.-EXP(-F(GRIDP)*RHO*D)
CALL GRIDDC
SICEPS(1,NEPS)=SICEPS(1,NEPS)+SICTIM(1,NEPS)/TSMALL(NEPS)
SICEPS(2,NEPS)=SICEPS(2,NEPS)+SICTIM(2,NEPS)/TSMALL(NEPS)
IF (A.EQ.0) GO TO 10
P(2,IBMR,NEPS)=P(2,IBMR,NEPS)+PP(2,IBMR,NEPS,GRIDP)
P(1,IBMR,NEPS)=P(1,IBMR,NEPS)+PP(1,IBMR,NEPS,GRIDP)
P(1,IFTR,NEPS)=P(1,IFTR,NEPS)+PP(1,IFTR,NEPS,GRIDP)
P(1,IFTR,NEPS)=P(1,IFTR,NEPS)+PP(1,IFTR,NEPS,GRIDP)
PCCEPS(NEPS)=PCCEPS(NEPS)+PCCEPS(NEPS,GRIDP)
GO TO 11
10 NU=NU+1
PZ(1,IBMR,NEPS)=PZ(1,IBMR,NEPS)+PPZ(1,IBMR,NEPS,GRIDP)
11 CONTINUE
PU(NEPS) = FLOAT(NU) / FLOAT(N)
XN = X
PCCEPS(NEPS) = PCCEPS(NEPS) / XN
SICEPS(1+NEPS) = SICEPS(1+NEPS) / XN
SICEPS(2+NEPS) = SICEPS(2+NEPS) / XN
IF (NU .EQ. 0) GO TO 14
PZ(IBM R,NEPS) = PZ(IBM R,NEPS) / FLOAT(NU)
IF (NU .NE. N) GO TO 15
P(1+IBM R,NEPS) = 2.
P(2+IBM R,NEPS) = 2.
P(1+IFTR,NEPS) = 2.
P(2+IFTR,NEPS) = 2.
RETURN
14  PZ(IBM R,NEPS) = ?.
15  FNNU = NU - NU
     P(1+IBM R,NEPS) = P(1+IBM R,NEPS) / FNNU
     P(2+IBM R,NEPS) = P(2+IBM R,NEPS) / FNNU
     P(1+IFTR,NEPS) = P(1+IFTR,NEPS) / FNNU
     P(2+IFTR,NEPS) = P(2+IFTR,NEPS) / FNNU
RETURN
END
SUBROUTINE UNCOND
COMMON/EXEC/NEPS,IFTR,IBMR,TMAXBF,*

COMMON/EXEC/NEPS,IFTR,IBMR,TMAXBF

COMMON/PKG/PKG,PKE,PCPEPS(73),PCPCEPS(73)
 COMMON/INPUT/NRT,NID,XID(72),NTEPS,EPSLON(73),*

1IN,TSNALL(25),TAWARE(25),T(25),T(25),T(25),T(25),T(25)

COMMON/INPUT/PK,PKE,PKDP,PKE(2,2)

COMMON/SICLIV/TIMSCI(1),SIMSTIM(1,25),SIMCPS(2,5)

COMMON/INPUT/PK,
COMMON/INPUT/PK,LIDE(25),PKGDE(25),PKGD,PKEGD,

NEPSM=1:NEPSM=1

PPUJ=5(*)PU(1)+PD(NEPS)*PU(NEPS))

PPD=5(*)PD(1)+PD(NEPS))

PKZ*=5(*)PD(1)+PD(NEPS)*PU(NEPS)

IFPZ=1:IFPZ=1)

PKE=5(*)PD(1)+PD(NEPS)*PCPEPS(1)*PD(NEPS)*PCPEPS(NEPS))

TIMSCI(1)=-5(*)PD(1)+PD(NEPS)*SICPS(1,1,1)+PD(NEPS)*SICPS(1,1,1,1))

TIMSCI(2)=-5(*)PD(1)+PD(NEPS)*SICPS(2,1,1)+PD(NEPS)*SICPS(2,1,1,1)

DO 5 J=1,2

5 PKK(I,J)=0

DO 10 I=1,NEPSM

TIMSCI(1)=-TIMSCI(1)+PD(1)+SICPS(1,1,+EPS)

TIMSCI(2)=-TIMSCI(1)+PD(2)+SICPS(2,1,EPS)

PPUJ=PPUJ+PD(1)+SICPS(1,EPS)

PPD=PPD+PD(1,EPS)

PKZ=PKZ+PD(1,EPS)*PPU(1,EPS)*PZ(1,IBMR,1,EPS)

PKE=PKE+PD(1,EPS)*PCPEPS(1,EPS)

CONTINUE

PPD=PPD/XEPS

PPUJ=PPUJ/XEPS

TIMSCI(1)=TIMSCI(1)/(PD(1)*XEPS)

10 CONTINUE
TMYsIC(2) = TMYsIC(2) / (PDD * XEPS)
PUPD = PYY / PDD
PKZB = PKZB / (XEPS * PUU)
PKE = PKE / XEPS
DIV = XEPS * (PDD - PPU)
DO 20 I1 = 1, 2
DO 20 J = 1, 2
PKK(J, I1) = 5 * PDD(1) * (1 - PU(1)) * P(J, I1, 1)
DO 15 K = 2, NTEPS1
PKK(J, I1) = PKK(J, I1) + PD(K) * (1 - PU(K)) * P(J, I1, K)
15 CONTINUE
DO 20 I1 = 1, 2
20 PKK(J, I1) = (PKK(J, I1) + 5 * PD(NTEPS) * (1 - PU(NTEPS)) * P(J, I1, 1) / DIV
PS = 1 - PKK(2, IFTR) * PDD * (1 - PUPD)
PKB = PKK(2, IFTR) * PDD * (1 - PUDP) + PKZB * PUU
IF (PS * EQ. 1) GO TO 30
EXT = (1 - PS * YN) / (1 - PS)
GO TO 31
30 EKT = YN
31 EKT = PKB * EKT
PKF = (1 - PUDP) * PDD * PKK(2, IFTR)
PSB = 1 - PKF
IF (PSB * EQ. 1) GO TO 40
ESB = (1 - PSB * YN) / (1 - PSB)
GO TO 41
40 ESB = YN
41 ESB = PKF * ESB
PKL = PKB - PKE
PKG = PKL / (1 - PKE)
PKBGD = PKB / PDD
PKFGD = PKF / PDD
DO 25 K = 1, NTEPS
PKBGD(K) = PZ(IBMR, K) * PU(K) + P2(IBMR, K) * (1 - PU(K))
PKFGD(K) = P2(IFTR, K) * (1 - PU(K))
25 CONTINUE
RETURN
END
SINFCT PJU. XR7, DECK

SUBROUTINE PJU

COMMON/EXEC/NEPS,IFTR,IBMR,TMAXBF

*,TMAX

COMMON/INPUTT/NRUN,NID,XID(72),NTEPS,EPSON(73),
1NN,TSMALL(25),TAKARF(25),F(5,2,25)

COMMON/INPUTT/N,RHO,TC(2)*M(2)+INK(6,2)+
1PKP(6,2)+NRUNCH,NDPMID,DPMID(72)

COMMON/PSCA/POD(73),PI(73),PI(2,2,73)+PZ(2,2,73)+GRIDP

COMMON/GRIDDD/I,J,LM3,LIM3+LIMC+IU+A+K(6,2)

COMMON/ORDERF/TT(6,2,7)+TMAB(6,2)

COMMON/PJUC/PP(2,2,73,25),PZ(2,73,25)

COMMON/PGKCAL/PGK,PGK,PCEPN(73,25)+PCEPS(73)

COMMON/MULTPL/TFIREP(25,10),MISTPP(25,10)+IAFIRP(25,10)

COMMON/SIMULT/TT

M1=V(1)
M2=M1

DO 1 1=1,M1

KW1=1K(11,1)

1 PK(KW1,1)=PKP(KW1,1)

DO 4 T=1,M2

KW2=1K(I2,2)

4 PK(KW2,2)=PKP(KW2,2)

ENTRY PJU1

LITLE=1

P=0.

IF (1.EQ.IFTR) PCEPN(NEPS,GRIDP).EQ.0.

5 IF (TFIREP,LLTL,GRIDP).EQ.0. GO TO 35

IAL=IAFIRP(LLTL,GRIDP)

IF (TFIREP(LLTL,GRIDP).GT.TC(IAL)) GO TO 30

7 LITLE=LLTL+1

GO TO 5

10 IF (LLTL.GT.1) GO TO 12

IF (TFIREP(LLTL,GRIDP).GT.TC(1)) GO TO 30

K=MISTP(1,GRIDP)

P=PK(K,1)
GO TO 30
12  K/TINY=LL/LTL-1
14  IF(TFIREP(LL/LTL,IGRIDP)=TF.GT.TFIREP(K/TINY,IGRIDP))GO TO 20
15  IF(K/TINY.EQ.1) GO TO 16
16  K/TINY=K/TINY-1
17  GO TO 14
18  IF(TFIREP(LL/LTL,IGRIDP).GT.TC(I)) GO TO 30
19  K=MISTP(LL/LTL,IGRIDP)
20  P=P+(1-P)*PK(K,I)
21  GO TO 30
22  CONTINUE
23  NT=K/TINY+1
24  Y=1
25  IF(NT.GT.L.L/LTL-1) GO TO 25
26  K=MISTP(NT,IGRIDP)
27  IF(TFIREP(NT,IGRIDP).LE.TC(IANT)) Y=Y*(1-PK(K,I))
28  NT=NT+1
29  GO TO 21
30  NT=1
31  X=1
32  CONTINUE
33  MINT=MISTP(NT,IGRIDP)
34  IANT=IAFIREP(NT,IGRIDP)
35  IF(TFIREP(NT,IGRIDP).LE.TC(IANT)) X=X*(1-PK(MINT,IANT))
36  NT=NT+1
37  IF(NT.LE.K/TINY) GO TO 26
38  K=MISTP(LL/LTL,IGRIDP)
39  P=P*X*PK(K,I)
40  CONTINUE
41  IF(I*IFTFEP) GO TO 7
42  IF(ABS(TFIREP(LL/LTL,IGRIDP)-TWARE(IGRIDP)).GT.0.01) GO TO 7
43  PCCEPN(NEPS,IGRIDP)=P
44  GO TO 7
45  CONTINUE
46  IF(I*EQ.0) GO TO 37
47  P.P(IU,J,NEPS,IGRIDP)=P
48  RETURN
SUBROUTINE OUTPUT

COMMON/EXEC/EPS, IFTR, IBMR, TMAXBF

COMMON/PCGC/PK, PKG, PKK, PK, PCEPN(73, 25), PCCPS(73)

COMMON/FPK, EFC, PC(73), P(2, 2), P(2, 73), PZ(2, 73)

COMMON/PKBG, PK, P, P(73), P(2, 2), P(2, 73), PZ(2, 73)

COMMON/PJUC, PP(2, 2), PP(2, 73), PPZ(2, 73)

COMMON/INPUTC/D, RHO, TC(2), M(2), IP*, IP, IP(9)

COMMON/UNCON, PNP(2, 73), PN(2, 2, 73), PNPZ(2, 73)

COMMON/XPUTT/NRUN, ND*ID(72), NTEPS, EPSLON(72)

COMMON/STCLV/TC(2), TIMSCF(2), STCTIM(2, 72), SICEP(5, 72)

COMMON/NUPROB, PKBGO, PKFGDE(2, 25), PKBGO(2, 25), PKE(2, 2, 72)

DATA (TR, D(10)) = (1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0)

FORMAT (2*10E12)

WRITE (6*1) (X1D(IJ)*IJ, NDPMID)

WRITE (6*1) (X1D(IJ)*IJ, NDPMID)

IF (N.GT.10) GO TO 105

IF (N.GT.20) GO TO 120

IF (N.GT.30) GO TO 120

FORMAT (2*10E12)

DATA (TR, D(20)) = (1.0, 2.0, 3.0, 4.0, 5.0, 6.0)

FORMAT (2*10E12)

DATA (TR, D(30)) = (1.0, 2.0, 3.0, 4.0, 5.0, 6.0)

FORMAT (2*10E12)
105 KOUNT=3
106 LIMUP(1)=10
107 LIMUP(2)=20
108 LIMUP(3)=M
120 CONTINUE
\[\text{WRITE}(6,3)\]
3 FORMAT(11H0,32X,24HNO. OF INITIAL ANGLES = 12*10X,
\[\text{130 NO. OF GRID POINTS PER ANGLE = 12)}\]
\[\text{DO 20 I=1,NTEPS}\]
20 EPSLON(I)=EPSLON(I)*57.2957795
\[\text{DO 130 L=1,KOUNT}\]
\[\text{N1=L+10-0}\]
\[\text{N2=LIMUP(L)}\]
\[\text{WRITE}(6,4)(Z8*,(ICOUNT(IJ)+IJ=N1+N2))\]
\[\text{WRITE}(6,26)\]
26 FORMAT(1X,6HEPSLON)\]
\[\text{DO 21 I=1,NTEPS}\]
21 WRITE(6,5) EPSLON(I) \((PPZ(1BMNR+I*J)+J=N1+N2)\)
130 CONTINUE
5 FORMAT(14X,8E3,11X*10(F7.5*3X))
\[\text{DO 140 L=1,KOUNT}\]
\[\text{N1=L+10-0}\]
\[\text{N2=LIMUP(L)}\]
\[\text{WRITE}(6,4) (Z8*,(ICOUNT(IJ)+IJ=N1+N2))\]
\[\text{WRITE}(6,26)\]
\[\text{DO 22 J=1,NTEPS}\]
22 WRITE(6,5) EPSLON(I) \((PP(1,1BMNR+I*J)+J=N1+N2)\)
140 CONTINUE
\[\text{DO 150 L=1,KOUNT}\]
\[\text{N1=L+10-0}\]
\[\text{N2=LIMUP(L)}\]
\[\text{WRITE}(6,4) (Z8*,(ICOUNT(IJ)+IJ=N1+N2))\]
\[\text{WRITE}(6,26)\]
\[\text{DO 22 J=1,NTEPS}\]
23 WRITE(6,5) EPSLON(I) \((PP(2,1BMNR+I*J)+J=N1+N2)\)
150 CONTINUE
DO 160 L=1, COUNT
   N1=L=10-9
   N2=LIMUP(L)
   WRITE (6,4) BP1F*(I=COUNT(IJ)+1=10,N)
   WRITE (6,26)
   DO 24 I=1, NTEPS
   24 WRITE (6,5) EPSLON(I);(PP(1,IFTR,I)+J=10,N)
  160 CONTINUE
   DO 170 L=1, COUNT
   N1=L=10-9
   N2=LIMUP(L)
   WRITE (6,4) BP2F*(I=COUNT(IJ)+1=10,N)
   WRITE (6,26)
   DO 25 I=1, NTEPS
   25 WRITE (6,5) EPSLON(I);(PP(2,IFTR,I)+J=10,N)
  170 CONTINUE
   WRITE (6,8)
   8 FORMAT(15H0,19X,2HPU,11X,2HPD,11X,3HPZB,10X,3HP1B+10X,3HP2B,10X,3HP
*1F,10X,3HP2F,2X,5HPKBD,8X,5HPKFGD,5X,6HEPSLON)
   DO 10 I=1, NTEPS
   10 WRITE (6,9) EPSLON(I),PU(I),PD(I),PK(IBM+1),PK(IBM+1)
   1P(2,IBM+1),PK(IBM+1),PK(IBM+1)
   2*PK(IBM+1),PK(IBM+1)
   9 FORMAT(4X,FA,3,9,6X,F7.5))
   WRITE (6,6) PPUB,PPD,PKZB,PKK(IBM),PKK(IBM),PKK(IBM)
   1PK(IBM),PK(IBM),PK(IBM)
   6 FORMAT(1HA,4X,7HPD/D,F7.5,5X,5HPD=F7.5,5X,7HPKZB=F7.5,5X,17HPK1B=F7.5,5X,7HPK2B=F7.5,5X,27HPK2F=F7.5)
   WRITE (6,200) EKT, EK, PKBD, PKFGD
   200 FORMAT(HA,4X,HEKB=F9.5,4X,HEKF=F9.5,4X,6HPKBD=F9.5,4X,6HPKFGD=F9.5,4X,6HPKFGD=1F9.5)
   WRITE (6,500) PKB
   500 FORMAT(1HA,4X,6HPKBD=F9.5)
   WRITE (6,300) (COUNT(IJ)+1=10,N)
   300 FORMAT(1HA,2X,9HPD/EPSN,11X,4HGRID,11X,6HPPOINTS,10(8X,12))
   WRITE (6,76)
DO 350 I=1,NEPS
250 WRITE (6,5) EPSLON(I)+PCCEPN(I+J)+J=1,N
WRITE (6,301)
201 FORMAT(1HO+21X+8PC(IEPS)/5X+6HEPSLON)
CC 351 :=1*NEPS
351 WRITE (6,9) EPSLON(I)+PCCEPS(I)
WRITE (6,302) PKL,PKG,PKF
302 FORMAT(1HO+12X+4HPKL=F7.5+5X+4HPKG=F7.5+5X+4HPKE=F7.5)
WRITE (6,303) TIMSIC(1),TIMSIC(2)
303 FORMAT(1HO,10X+5HS(1)=F5.3,20X+5HS(2)=F5.3)
RETURN
END
SUBROUTINE GRIDOC
COMMON/EX3C/WR.PS,IFTR,IBM.R,TMAXBE
* TMAX
COMMON/INPUTC/?,RHO,TC(2),M(2),IWK(6,2),1PKP(6,2),N,NSNC,NDPMID,DPMD(72)
COMMON/OUTPUTC/MPSLXID(72),RPSLOX(73),
1MN,TSMALL(25),TANARE(25),F(25),T(6,2,25)
COMMON/GRDG0/,J,LIMA,LMR,LMC,IM+1,A,PK(6,2)
COMMON/PRGS/PO(73),PO(73),P(2,2,73),P(2,73),IGRIDP
COMMON/MULTPL/TFIREP(25,10),MSTPP(25,10),IAFIRP(25,10)
I=IBM.R
J=IFTR
LIMA=V(I)
1M=IWK(1,1FTR)
TMAX=T(IM+1,1GRIDP)
IF (LIMA,Eq.1) GO TO 6
DO J=1M-2,LIMA
1M=IWK(1M,1FTR)
IF (T(1M+1,1GRIDP),Eq.1000,1) GO TO 5
IF (TMAX,LT,T(V+1,1GRIDP)) TMAX=T(V+1,1GRIDP)
CONTINUE
6 CONTINUE
IF (TAWARE(IGRIDP),GE,A.MIN(TC(1,TMAX))) GO TO 20
A=1
I=IBM.R
J=IFTR
1=2
CALL PJU
1=1
1=4(MJ)
DO 10 11=1,12
11=1M(J)
PK(KW,J)=0
CALL PJU
1=IFTR
J=IBM.R

SUBRTC MAIN
XR7, DECK
COMMON/PKCCAL/PKL, PKG, PKF, PCRCPN(73,25), PCCEPS(73)
COMMON/FSFC/NEPS, ITTP, ITMR, TMAXRF
*, TMAX
COMMON/INPUT/D/M, RHO/TC(2), M(2), IWK(6, 2)
1PK(6, 2), X, NEPC, NPMID, DPMID(72)
COMMON/INPUT/INPUT, NID, XID(72), NEPS, EPSLON(73)
1MK, TSMALL(25), TAWARN(25), F(75), T(6, 2, 25)
COMMON/PCUC/PCUP(2, 7, 25), PCZ(2, 73, 25)
REWIND 11
1
1FCR=1
ITMR=2
DO 2 K1=1, 73
DO 2 K2=1, 73
DO 2 K3=1, 72
PDZ(K1, K2, K3)=0.
DO 2 K4=1, 72
PPP(K1, K4, K2, K3)=0.
WRITE(6, 11)
FORMAT(111)
CALL INPUT
TMAXRF=AMAX1(FC(1), FC(2))
DO 10 N=1, 11, N1EP
NEPS=11
PCCEPS(NEPS)=0
CALL INPUT?
CALL FSICAL
CONTINUE
CALL UNCOND
CALL OUTPUT
REWIND 11
GO TO 1
END
S19FC INPUT. XR7.DECK

SUBROUTINE INPUT
COMMON/COMP/ICOMP
COMMON/EXK/EKX,YN,FKF
COMMON/EXFC/NEPS,IFTR,IBM,TFMAXRF

*TVMAX
COMMON/INPUTC/D,PHO,TC,TF,MDW,A1XW(6+2)
1PKP(6+2),M,NDPMNC,NDPMID,NDPMID(72)
COMMON/INPUT/YRN,MID,XID(72),MTEPS,EPSONL(73)
1N,M,SMALL(25),TAMARE(25),F(25),L(6+2+5)
COMMON/TIME/TXIMI(6+2+5)
COMMON/MULTP/TIREP(25+10),MISTPP(25+10),IAFRP(25+10)
COMMON/SICLW/TIMIC(2),SICTIM(2,25),SICEPS(2,5)
COMMON/SIMUL/TFS
DIMENSION ISTIME(6+2)
DIMENSION NTOT(2)
READ(5,1)NRUNC=N(M+10J+1J),1J=1,N2)
1 FORMAT(14X,11/14X=1/1AX,2/15)
M=M(1)
M=M(2)
READ(5,2),(IWK(K+1),K=1,M1),(IWK(L+1),L=1,N2)
2 FORMAT(14X,11/14X=1/1AX,2/15)
READ(5,40),(NDPMID(I),I=1,NDPMID)
40 FORMAT(14X,17/12AX=1/1AX,2/15)
READ(5,3),(PHO,TC(I),TC(2)
2 FORMAT(14X,17/12AX=1/1AX,2/15)
DO 15 K1=1,2
15 CONTINUE

16 FORMAT(14X,F13.0)
DO 17 I=1,N1
17 READ(YK,0)PKP(UK,1)
READ(5,16)NP2
READ(5,16)US
READ(5,16)LCAP
READ(5,16)TF
5 READ (11,50) NRUN, NTEPS, NN, NM1, NM2
50 FORMAT (1H0*515)
   READ (11,40) NID, (XID(ID), ID=1, NID)
   READ (11,50) COMP
   IF (NRUNG.EQ.0 .AND. NRUN).GO TO 4
   DO 10 I=1, NTEPS
      READ (11,52) EPSLON(I), (TSMALL(IJ), TAWARE(IJ), F(IJ)),
      IJ=1, NN)
      (TMIS1, NGP1, MIS1=1, NM1), NGP1=1, NN)
      2((TMIS2+2, NGP2, MIS2=1, NM2), NGP2=1, NN)
   READ (11,52) ((TMIS(MIS1, NGP1, MIS1=1, NM1), NGP1=1, NN),
     *((TMIS(MIS2, NGP2, MIS2=1, NM2), NGP2=1, NN)
   DO 1002 IDISC=1, NN
   DO 1000 IMNUTS=1, LCAP
   READ (11,200) NGP, TFIREP(IMNUTS, NGP), MISTPP(IMNUTS, NGP),
   *!AFIRP(IMNUTS, NGP), IFZERO
   IF (IFZERO.GE.0) GO TO 1001
1000 CONTINUE
1001 CONTINUE
1002 CONTINUE
   READ (11,52) (SICRIM(I, NGP1), SICRIM(2, NGP1), NGP1=1, NN)
10 CONTINUE
   GO TO 5
   ENTRY INPUT2
   READ (11,52) EPSLON(NEPS), (TSMALL(IJ), TAWARE(IJ), F(IJ),
      IJ=1, NN)
      (TMIS1, NGP1, MIS1=1, NM1), NGP1=1, NN)
      2((TMIS2+2, NGP2, MIS2=1, NM2), NGP2=1, NN)
   READ (11,52) ((TMIS(MIS1, NGP1, MIS1=1, NM1), NGP1=1, NN),
     *((TMIS(MIS2, NGP2, MIS2=1, NM2), NGP2=1, NN)
52 FORMAT (1H0*416, A/17X, 3E16.8))
   EPSLD=EPSLON(NEPS) + 5.2957795
   WRITE (6,30) EPSLD
30 FORMAT (1H0*57X, 'HEPSLON = ', F7.3)
   WRITE (6,31)
31 FORMAT (1H0*27X, 'GRIDPOINT >= 20X, 1HT + 21X, 6HTWREST, 22X, 1HF')
   DO 32 IJ=1, N
32 WRITE (6,32) IJ, TSMALL(IJ), TAWARE(IJ), F(IJ)
32 FORMAT (26X, 12*4X, 3(5X, F10.3))
WRITE(6,34)
FORMAT(1HO*37X*7MISSILE*7X*3HA/C,6X*3HGRIDPOINT,3X,
12HT(MIS=A/C,GRIDPOINT)*6X*12HTIME IN ENV.)
NTOT(11)=NM1
NTOT(2)=NM2
DO 35 II=1,N
MIS=NTOT(II)
DO 35 II=1,N
35 WRITE (6,36) I2=1+13,T=(I2=II+I3)*TTMIS(I2=II+I3)
FORMAT(41X*I1,10X,I1,10X,I2,13X,F8.3,12X,F8.3)
DO 100 II=1,N
MIS=NTOT(II)
DO 100 II=1,N
TOTIME(I2=II)=0.
DO 100 II=1,N
TOTIME(I2+II)=TOTIME(I2+II)+TTMIS(I2+II+I3)
CONTINUE
WRITE (6,37) (TOTIME(I2=II),II=1,NM1)
WRITE (6,38) (TOTIME(I2=II),II=1,NM1)
FORMAT(1HO*31X*13HTOTAL(K+1)=6(F8.3,5X)
FORMAT(1HO*31X*13HTOTAL(K+2)=6(F8.3,5X)
WRITE (6,202)
FORMAT(1HO*3X*10GRID_POINT*6X*8HAIRCRAFT*7X*7MISSILE,
*AX,10HTIME FIRED)
DO 200 ISAVE=1,NN
DO 200 INUNTS=1,LCAP
READ (11,200) NGP,TFIREP(INNUNTS,NGP),MISTPP(INNUNTS,NGP),
IAFIRP(INNUNTS,NGP),IFZERO
WRITE (6,201) NGP,IAFIRP(INNUNTS,NGP),MISTPP(INNUNTS,NGP),
TFIREP(INNUNTS,NGP)
IF (IFZERO*NGP=0) GO TO 2001
CONTINUE
WRITE (6,201) NGP,IAFIRP(INNUNTS,NGP),MISTPP(INNUNTS,NGP),
TFIREP(INNUNTS,NGP)
IF (IFZERO*NGP=0) GO TO 2001
CONTINUE
READ(11,52) (SCTIM1=NGP1,SCTIM2,NGP1),NGP1=1,NN
FORMAT(1H*15,C15.7,3I10)
201 FORMAT(1H,'3J15,F10.3)
RETURN
4 WRITE(6,7)
7 FORMAT(41X,'INPUT FOR THE ATAC-2 DATA PROCESSING MODEL')
WRITE(6,6)(XID(IJ),IJ=1,NID)
6 FORMAT(1HO,29X,12A6)
WRITE(6,20)NRUNC,N,M(1),M(2),NTEPS
20 FORMAT(1HO,31X,'HRUN NUMBER',I3,5X,'HN',I3,5X,'NTEPS',I3)
WRITE(6,21)(1WK(K+1),K=1,M1)
21 FORMAT(1HO,31X,'HRUN (K) = ',I3,5X)
WRITE(6,22)(1WK(K+2),K=1,M2)
22 FORMAT(1HO,31X,'K2 = ',I3,5X)
WRITE(6,23)D,RHO,TC(1),TC(2)
23 FORMAT(1HO,29X,'D = F8.3',5X,'RHC = F8.3',5X,'TC(1) = F8.3',5X,'TC(2) = F8.3')
DO 25 I=1,M1
25 WRITE(6,26)KW1,PKP(KW1)
26 FORMAT(55X,'KW1 = IWK(17,1)')
DO 27 J=1,M2
27 WRITE(6,28)KW2,PKP(KW2)
28 FORMAT(55X,'KW2 = IWK(17,2)')
WRITE(6,60)TF
60 FORMAT(1HO,54X,'TF=F8.3')
RETURN
END
SECTION 10
AN EXAMPLE OF INPUT AND OUTPUT

In this section an example of the computer printout for the simulation of one engagement is presented and discussed. The inputs and the printout for this run are shown on pages 144 through 150.

10.1 Inputs

On page 144 and the top of page 145 are shown some of the inputs of the program. The combatants are coded on top of page 144 as aircraft 10 (fighter) and 11 (bomber) for identification purposes. The range of the fighter's IFF is given as 12,000 feet. The altitude associated with the performance data of each aircraft is 10,000 feet. Then data are listed which in most cases are self-explanatory. For example, $t_{\text{MAX}}$, the limit on the time of each engagement is 300 seconds. Moving down to the double columns of data (aircraft 10 is on the left, aircraft 11 on the right) it is seen that the avionics data for the aircraft are identical. Each has a detection radar with range of 140,000 ft, and half-angle 60°, etc.

On pages 145 and 146 some of the weapon envelopes are shown. The entries of the big blocks of data on pages 145 and 146 give the ranges of a weapon envelope as a function of the angle-off and target velocity. The first weapon number, MIS, is 1 and it is on aircraft 1 (aircraft 10). The first tables are for RFL, the outer range of the envelopes when the target pulls $G_1$ lateral g's. $G_1$, $G_2$, $G_3$ in the input list correspond to the FORTRAN names $GT(1)$, $GT(2)$, $GT(3)$, respectively. Page 144 shows $G_1 = 0$, which means the target is non-maneuvering.

1) I.e., in this particular example, aircraft number 1 is aircraft design number 10, and aircraft number 2 is design number 11.
The last block of data shown at the bottom of page 146 gives the inner range of MIS number 5 on aircraft 2 (aircraft 11) associated with $G_3$ ($= 4.9$) lateral g's.

On page 147 the all important specific power and $g_4$ functions of each aircraft are shown. At the extreme left the speeds associated with all the other entries are listed. The second column headed $G_{MAXT}$ gives the $g_4(V)$ function of aircraft 1. For example, at 1185.8 feet per second, aircraft 1 is able to sustain 4.3 total g's without having to decelerate. The third column, which is not an input, but is computed by the program gives the turning rate $g_1$, at $g_4(V)$ total g's. The fourth column, headed $GBIG$, contains the $G(V)$ function of aircraft 1. This is the upper bound to the total g's of the aircraft. For example, at 754.6 feet per second aircraft 1 may sustain no more than 6.2 total g's. Finally, columns 5 through 11 describe the specific power as a function of speed (by row) and total g's (by column). At, say, a speed of 1401.4 feet per second and 3 total g's this aircraft has a specific power of 220.5 feet per second. Also at a speed 1,078 feet per second and, say, 6.0 total g's the aircraft must decelerate, for its specific power is - 99.3 feet per second. Finally, the last two blocks of data on page 147 give the $a_{MIS}(i)$ and $G_{MIS}(i)$ limits, respectively.

On page 148 some more input data are shown. At the top of the page is shown $a_{max}(i)$ for both aircraft as .26 radians. The angle is actually inputted as 15°, but is converted to radians by the computer prior to printing out. Further down, opposite the word $N(MIS, i)$, is shown the number of weapons $N(MIS, i)$ of type MIS carried by aircraft 1. Here it is seen that each aircraft carries two of weapon types 1 and 2, and eight of weapon type 5. VSTR corresponds to $V_4^*$ and is computed by the program.
Finally, pages 149 and 150 show the printout (results) of one engagement. The whole table is ordered on time, shown in seconds in the first column under $T$. The second column gives the range, $R$, between the aircraft. Then the variables peculiar to the bomber are given. They are from left to right:

- $XB$ the $x$ coordinate of the bomber,
- $YB$ the $y$ coordinate of the bomber,
- $VB$ the speed of the bomber,
- $MB$ the $m$-state of the bomber,
- $GB$ the total g's of the bomber,
- $BB$ the turning rate of the bomber,
- $ALPB$ the tracking angle of the bomber,
- $I$ the ST(1) indicator of the bomber,
- $PHIB$ the bomber's angle off the fighter,
- $KB$ the information state of the bomber.

The right side of the page gives these same variables for the fighter.

At the bottom of page 150 the summary of firings of each aircraft is given. The first block of data gives firings of all weapon types. The entry of 1,000.0 for the time of firing indicates that the weapon was not fired. The TMIS column gives the time spent in that weapon's envelope, and the remaining columns give the relative parameters at the time of firing: the range, tracking angle and angle-off of the firing aircraft. The last block of data also lists the variables $IA(I), NI(I), T(I)$, which are inputs to the DFM. For example, at 83.5 seconds into the engagement aircraft 1 fired two missiles at aircraft 2, weapons number 1 and 2. Then aircraft 2 fired missile type 1 at 163 seconds followed by another at 199.5 seconds. One observes that the first firing of aircraft 1 elapsed at 140.
as evidenced by $t_{\text{Aware}} = 83.5$ seconds. Finally, the last line gives the
nick time of each aircraft. This is the amount of time that each aircraft
spent in a condition in which $O_1 > 1$.

It may be interesting and instructive to observe these data in greater
detail and thereby better understand some of the notions previously
developed. The reader is cautioned to consider this as only one example
and as such it cannot present a general view of the model.

Firstly, for 83.5 seconds the bomber is totally unaware of the fighter;
it is moving up the $y$-axis in linear flight at a constant velocity of 916
feet per second. During this time the fighter is afforded the opportunity
to gain a very favorable position. The fighter initially has active
information ($KF = 4$). It also notices that the bomber is unaware and
hence the fighter sets its lag angle, $\phi_{\text{max}}(F)$, to 60° and thereby tries
to get behind the bomber. Also, in the initial 50 seconds the fighter
accelerates to press the attack. Then as the fighter acquires the tail
of its enemy it lags less and less till it reaches $\phi^*$, and thereafter
flies a pure pursuit course, $\text{ALPF} = \phi_x = 0$. As the data indicate, $\phi^*$
is a little less than 57.8°. It is $\cos^{-1} \left( \frac{v_y}{2v_x} \right)$, or about $\cos^{-1}
(1350/2 \cdot 916) = 57.8$. From this point until the awareness of the bomber,
the fighter is in the $\phi^*$ cone of the bomber and flies a pure pursuit
course. Another point of interest in this portion of the engagement is
seen in the $n$-state of the bomber. Between 40 and 50 seconds this index
changes from 2 to 1. As previously noted this signals that the fighter
goes from a state of acceleration, $n = 2$, to a state of deceleration,
$n = 1$. Of course, this is clearly indicated by the speed printout as well.
In this portion of the engagement the fighter gains IFF. The range of the
IFF gear of the fighter is 12,000 feet (see inputs). And this range is attained somewhere between 50 and 60 seconds of combat. Verification of the acquisition of IFF by the fighter is seen in its information state KF. This index changes from 4 (active information) to 5 (active with IFF). These are some of the notable events in the early portions of the combat.

All during the early pursuit the fighter holds fire so as to get to a good range off the tail of the enemy, $R^*$. Against an unaware enemy $R^*(F)$ is set to $R'(1, F)$ which in this case is 4,100 feet. This range is attained shortly after 80 seconds, for at 80 seconds the range is 4,265 feet. Of course, due to the time slice nature of the simulation 4,100 feet will not be hit exactly. However, the fighter attains this range within 5 feet, for it is seen in the firing summary that the fighter launched its first 2 missiles at a range of 4,095 feet. It then alerted the bomber by these firings. And from 83.5 seconds on, both aircraft are aware of each other.

At 90 seconds the bomber is in information state 9: Passive and IFF. This is because the fighter is illuminating the bomber with its tracking radar. It has IFF because it was fired upon. In this state the bomber turns as hard as possible to acquire its enemy. In so doing it loses the passive information and goes into a lost information state, $KB = 11$. Therein (100 to 150 seconds) it continues to turn hard and decelerates to $V^* = 755$ feet per second. This is the speed at which its sustainable turning rate is best. In so doing the bomber finally, at about 150 seconds, acquires the fighter and gets active information, $KB = 5$. The maneuver also causes the fighter to lose information, $KF = 11$. 
Through the rest of the engagement each aircraft turns hard to acquire
the other. However, it is seen that the bomber, in fact, does a better
job of this than the fighter, for from 150 seconds until the end, the bomber
has active information most of the time, while the fighter is in a lost or
passive information state. Further, the bomber is able to launch two weapons
at the maneuvering fighter, one at 163 seconds and the other at 190 seconds.
This superiority of the bomber over the fighter is, in part, clear from the
inputs. From the functions for each aircraft, (see page 147) it is
seen that $V*$ for both is 754.6 feet per second. But at this speed the
bomber can out-turn the fighter since $\dot{\beta}_B = .274$ radians per second, while
$\dot{\beta}_F = .261$. Thus by the arguments presented in Appendix D, superiority of the
bomber over the fighter in the latter half of the engagement is expected.
<table>
<thead>
<tr>
<th>ALFTEN</th>
<th>0.20</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>WALK</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>GP</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALFTEN</th>
<th>0.20</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>WALK</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>GP</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALFTEN</th>
<th>0.20</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>WALK</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>GP</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALFTEN</th>
<th>0.20</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>WALK</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>GP</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALFTEN</th>
<th>0.20</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>WALK</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>GP</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>W4</td>
<td>GAP%</td>
<td>EFFECT</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td>322.400</td>
<td>1.000</td>
<td>0.00</td>
</tr>
<tr>
<td>431.700</td>
<td>2.000</td>
<td>0.10</td>
</tr>
<tr>
<td>536.000</td>
<td>3.00</td>
<td>0.17</td>
</tr>
<tr>
<td>646.000</td>
<td>4.00</td>
<td>0.21</td>
</tr>
<tr>
<td>765.000</td>
<td>5.00</td>
<td>0.22</td>
</tr>
<tr>
<td>862.000</td>
<td>6.00</td>
<td>0.29</td>
</tr>
<tr>
<td>962.000</td>
<td>7.00</td>
<td>0.21</td>
</tr>
<tr>
<td>1062.000</td>
<td>8.00</td>
<td>0.23</td>
</tr>
<tr>
<td>1145.000</td>
<td>9.00</td>
<td>0.14</td>
</tr>
<tr>
<td>1214.000</td>
<td>10.00</td>
<td>0.09</td>
</tr>
<tr>
<td>1360.000</td>
<td>11.00</td>
<td>0.07</td>
</tr>
<tr>
<td>1492.000</td>
<td>12.00</td>
<td>0.02</td>
</tr>
</tbody>
</table>

**ENERGY MANOUVERABILITY TABLES FOR AC 2**

<table>
<thead>
<tr>
<th>W4</th>
<th>GAP%</th>
<th>EFFECT</th>
<th>GSPG</th>
<th>BEE%</th>
</tr>
</thead>
<tbody>
<tr>
<td>103.000</td>
<td>1.000</td>
<td>0.06</td>
<td>1.000</td>
<td>218.400</td>
</tr>
<tr>
<td>373.400</td>
<td>1.000</td>
<td>0.04</td>
<td>2.000</td>
<td>247.500</td>
</tr>
<tr>
<td>431.700</td>
<td>2.000</td>
<td>0.16</td>
<td>3.000</td>
<td>451.400</td>
</tr>
<tr>
<td>546.000</td>
<td>3.000</td>
<td>0.18</td>
<td>4.000</td>
<td>556.000</td>
</tr>
<tr>
<td>646.000</td>
<td>4.000</td>
<td>0.29</td>
<td>5.000</td>
<td>654.000</td>
</tr>
<tr>
<td>765.000</td>
<td>5.000</td>
<td>0.45</td>
<td>6.000</td>
<td>762.000</td>
</tr>
<tr>
<td>862.000</td>
<td>6.000</td>
<td>0.47</td>
<td>7.000</td>
<td>874.200</td>
</tr>
<tr>
<td>962.000</td>
<td>7.000</td>
<td>0.34</td>
<td>8.000</td>
<td>972.000</td>
</tr>
<tr>
<td>1062.000</td>
<td>8.000</td>
<td>0.29</td>
<td>9.000</td>
<td>1072.000</td>
</tr>
<tr>
<td>1145.000</td>
<td>9.000</td>
<td>0.18</td>
<td>10.000</td>
<td>1172.000</td>
</tr>
<tr>
<td>1214.000</td>
<td>10.000</td>
<td>0.09</td>
<td>11.000</td>
<td>1272.000</td>
</tr>
<tr>
<td>1360.000</td>
<td>11.000</td>
<td>0.07</td>
<td>12.000</td>
<td>1364.000</td>
</tr>
<tr>
<td>1492.000</td>
<td>12.000</td>
<td>0.02</td>
<td>13.000</td>
<td>1464.000</td>
</tr>
</tbody>
</table>

**APLUS (AC/PISSILE)**

<table>
<thead>
<tr>
<th>AP/C</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20.00</td>
<td>20.00</td>
<td>20.00</td>
<td>20.00</td>
<td>20.00</td>
<td>20.00</td>
</tr>
<tr>
<td>2</td>
<td>20.00</td>
<td>20.00</td>
<td>20.00</td>
<td>20.00</td>
<td>20.00</td>
<td>20.00</td>
</tr>
</tbody>
</table>

**PISLIE (AC/PISSILE)**

<table>
<thead>
<tr>
<th>AP/C</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
</tr>
<tr>
<td>2</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
<td>8.00</td>
</tr>
<tr>
<td>TANAME</td>
<td>32.5</td>
<td>TLAST</td>
<td>3CC.E</td>
<td>F</td>
<td>10T151</td>
<td>EPSLRH</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>---</td>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td>I</td>
<td>H15</td>
<td>T</td>
<td>TPIS</td>
<td>R</td>
<td>ALPHA</td>
<td>PHI</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>31.50</td>
<td>0.25</td>
<td>4090.04</td>
<td>-C</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>7</td>
<td>33.50</td>
<td>0.25</td>
<td>4090.04</td>
<td>-C</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1CC.CC</td>
<td>0</td>
<td>0</td>
<td>-C</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1CC.CC</td>
<td>0</td>
<td>0</td>
<td>-C</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1CC.CC</td>
<td>0</td>
<td>0</td>
<td>-C</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>1</td>
<td>1CC.CC</td>
<td>0</td>
<td>0</td>
<td>-C</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>163CC</td>
<td>2.78</td>
<td>7239.95</td>
<td>-16.41</td>
<td>72.95</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>7</td>
<td>1CC.CC</td>
<td>0</td>
<td>0</td>
<td>-C</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>1CC.CC</td>
<td>0</td>
<td>0</td>
<td>-C</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>1</td>
<td>1CC.CC</td>
<td>0</td>
<td>0</td>
<td>-C</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>1</td>
<td>1CC.CC</td>
<td>0</td>
<td>0</td>
<td>-C</td>
</tr>
<tr>
<td>CRIC PT</td>
<td>AIRCRAFT</td>
<td>P15</td>
<td>TYPF</td>
<td>T/IPE</td>
<td>FIRED</td>
<td>RANGE</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>81.500</td>
<td>4075.034</td>
<td>-0</td>
<td>0.186</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>81.500</td>
<td>4075.034</td>
<td>-0</td>
<td>0.186</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1</td>
<td>163.000</td>
<td>7239.95</td>
<td>-16.407</td>
<td>72.954</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1</td>
<td>106.500</td>
<td>7211.751</td>
<td>-10.494</td>
<td>61.960</td>
</tr>
</tbody>
</table>

SICK TIME(1)=225.25
SICK TIME(2)=212.25
APPENDIX A

THE PROBABILITY OF DETECTION

Since the ATAC-2 model initiates all engagements with the fighter detecting the bomber and since the relevant probabilities of kill are conditional on detection, the probability of detection as a function of the initial relative heading angle, \( \theta \), is of importance. This section contains the derivation of that probability. The scenario is one that is consistent with the random search of an area with no prior knowledge of the presence of an enemy. This situation is also consistent with the method of initialization of the combatants by the ENGAGEMENT Model.

Suppose then that the fighter is searching some large area \( A \) in which the bomber exists but its position is unknown to the fighter. Let the bomber's position in this area be a random variable with bivariate distribution function

\[
P(x, y) = \{ \Pr X < x, Y < y \},
\]

where \( X \) and \( Y \) are the random variables defining the bomber's position in \( A \). Let the bomber's velocity vector be parallel to the \( x \)-axis. Now \( A \) is such that the bomber's position in \( A \) is invariant with time, i.e., the area \( A \) moves with the bomber. Hence,

\[
\int_A dP(x, y) = 1.
\]

(\( P(x, y) \) will later be taken to be uniform for simplicity although this need not be the case.)
A.1 Stationary Target

Suppose for the moment that the fighter is searching the area $A$ for a stationary target. The velocity $\overline{V_B}$ is zero and the target does not move. In an amount of time $t$ the fighter's detection pattern will have swept out an area $a = r V_F t$ where $r$ is the normal projection of the fighter's detection pattern onto the normal to the fighter's velocity vector. Further it may be said that the stationary target will be detected in time $t$ if and only if the target is in the area $a$ at time $t' \leq t$. But the target is in the area $a$ with probability

$$\int_a dP(x, y)$$

And if $P(x, y)$ is a uniform distribution, then this probability is just

$$\frac{a/A}{a/A} = \frac{r V_F t}{A}$$

The above may now be generalized to account for the moving target.

A.2 Non-Stationary Target

Suppose now, as is generally the case, that the bomber has a velocity $\overline{V_B}$ while the fighter is searching for the target with velocity $\overline{V_F}$. Put in relative coordinates the fighter searches the area $A'$ for a stationary target with velocity $\overline{V_S} = \overline{V_F} - \overline{V_B}$, the relative velocity. With this in mind the situation is completely analogous to the stationary target case above. Again the fighter sweeps out an area in $A'$ equal to $a'$. By analogy then the area $a'$ is $a' V_S t$ where $r'$ is the normal projection onto the
normal to the velocity vector with which the fighter searches for a stationary target, namely $V^*$. Then, if $P_D(e)$ is the probability of detection at some initial relative heading angle $e$ and $P(x, y)$ is uniform over $A$, \[ P_D(e) = \frac{Y^* V^* e}{A} \]

$V^*$ is obtained from $V_F$, $V_B$ and $e$ by the Law of Cosines, i.e., \[ V^* = \left[ V_F^2 + V_B^2 - 2 V_F V_B \cos e \right]^{1/2} \]

Let $D$ be the distance that the target travels in time $t$, analogous to a penetration distance, then \[ P_D(e) = \frac{(1/A) Y^* V^* D}{V_B} \]

where \[ F = \frac{Y^* V^*}{V_B} \]

The probability of no detection is $1 - P_D(e)$. And if $n$ fighters search the area $A$ independently with the same relative velocity, then the probability that no detection occurs is \[ \left[ 1 - P_D(e) \right]^n \]

while the probability that at least one detects the target is \[ 1 - \left[ 1 - P_D(e) \right]^n \]
An approximation to this quantity is given by the exponential function:

\[ [1 - P_D(c)]^n = \frac{1}{n} - e^{-(n/A) F D} \]

Then letting \( \delta = n/A \) be the density of fighters in the area \( A \) and taking the above approximation

\[ P_D(c) = 1 - e^{\delta F D} \]

It is well to note some of the tacit assumptions throughout the derivation of \( P_D(c) \). For example, it was assumed that the fighter remained within the area \( A \) for the total time \( t \). For small values of \( t \) or large \( A \) this assumption is not unreasonable. Also it was assumed that no detection can take place in zero time; detection is accomplished by relative motion between the aircraft. Therefore, if \( \vec{V}_0 \) the relative velocity vector is zero, then \( P_D(c) \) will also be zero. (For non-zero velocities of the aircraft this only occurs at \( \vec{V}_F = \vec{V}_B \) and \( c = 0 \).)
APPENDIX B
GEOMETRIC CONSIDERATIONS

This Appendix defines the connection between the inertial and relative coordinate systems. Also, the equations of relative motion are derived.

B.1 Inertial Vs. Relative Coordinates

Consider then two aircraft labeled F and B with respective speeds $V_F$ and $V_B$ at some fixed altitude, see Figure B.1-1. The inertial positions of the aircraft are defined by $(x_F, y_F)$ and $(x_B, y_B)$, while the directions of their headings are given by $\beta_F$ and $\beta_B$, measured as shown. Let $\theta$ be the direction of the line of sight between the two aircraft. As previously stated $\alpha_i$ is the tracking angle of aircraft $i$ measured from the inner line of sight to the heading of aircraft $i$, while $\phi_i$, the angle-off, is measured from the outer line of sight to the heading of aircraft $j \neq i$.

The following conventions are adopted:

1. All angles are between $-\pi$ and $\pi$ inclusive,
2. Angles measured in a counterclockwise manner are positive, while angles measured in a clockwise manner are negative.

Now the inertial coordinate system uniquely defines the positions of the aircraft with respect to any observer. Seven parameters are used -- $x_F$, $x_B$, $y_F$, $y_B$, $\beta_F$, $\beta_B$, and $\theta$. In this system $\theta$ is redundant since it may be obtained from the others; by observation

$$\theta = \tan^{-1} \left( \frac{y_B - y_F}{x_F - x_B} \right) \quad (B.1-1)$$
This may also be taken as a definition of $\Theta$. However, the relative coordinate system makes use of five parameters -- $R$, $\alpha_F$, $\phi_F$, $\phi_B$, and $\phi_B$ and does not uniquely define the aircraft's inertial position. One of the angles $\alpha_F$, $\phi_F$ and one of the angles $\alpha_B$, $\phi_F$ are redundant. Again by observation

$$\alpha_B = \phi_F - \pi \text{ sgn}(\phi_F)$$

and

$$\phi_B = \alpha_F - \pi \text{ sgn}(\alpha_F).$$

Thus, only three parameters are needed to define the relative coordinate system; a distance (range) and two angles.
Yet these variables define the positions for an observer that is fixed with respect to an aircraft. Indeed, translations and rotations of the inertial system give rise to the same relative position. The object of the above is to show that the relative coordinates may be uniquely determined by the inertial coordinates. The reverse is, of course, not true. The following gives this correspondence between the inertial and relative coordinate system explicitly.

B.2 Deriving the Relative Coordinates From the Inertial Coordinates

Suppose that the relative coordinates as a function of the inertial coordinates are desired. The range $R$ is given by

$$R = \left[ (x_B - x_F)^2 + (y_B - y_F)^2 \right]^{1/2} \quad (B.2-1)$$

Also by reference to Figure B.1-1 it is seen that

$$\alpha_F = \theta - \beta_F \quad (B.2-2)$$

and

$$\phi_F = \theta - \beta_B \quad (B.2-3)$$

Thus the relative coordinate system is defined from the inertial system.

B.3 The Equations of Relative Motion

In this section the equations of relative motion are derived. These equations are expressions for the time derivatives of the relative parameters in terms of the relative parameters.

Firstly, consider the closure rate or rate of change of the range, $\dot{R}$. From (B.2-1) differentiating with respect to time gives
\( p \frac{dp}{dt} = (x_B - x_F) \frac{d}{dt}(x_B - x_F) \\
+ (y_B - y_F) \frac{d}{dt}(y_B - y_F) \).

But

\( \frac{d}{dt}(x_B - x_F) = v_B \cos \beta_B - v_F \cos \beta_F \)

and

\( \frac{d}{dt}(y_B - y_F) = v_B \sin \beta_B - v_F \sin \beta_F \).

Also

\( (x_B - x_F)/R = \cos \theta \)

and

\( (y_B - y_F)/R = \sin \theta \).

Thus

\( \frac{dR}{dt} = \dot{R} = v_B(\cos \beta_B \cos \theta - \sin \beta_B \sin \theta) \\
- v_F(\cos \beta_F \cos \theta - \sin \beta_F \sin \theta) \).

Since \( \alpha_F = 0 - \beta_F \) and \( \phi_F = \theta - \beta_B \), \( \dot{R} \) is finally given by

\( \dot{R} = v_B \cos \phi_F - v_F \cos \alpha_F \). \( \text{(B.3-1)} \)

From (B.2-2) and (B.2-3) the angular rates \( \dot{\alpha}_F \) and \( \dot{\phi}_F \) are easily obtained as

\( \dot{\alpha}_F = 0 - \beta_F \). \( \text{(B.3-2)} \)

and

\( \dot{\phi}_F = \theta - \beta_B \). \( \text{(B.3-3)} \).
Now, \( \dot{\beta}_T \) and \( \dot{\beta}_B \) are the turning rates of the respective aircraft, which are at the control of the respective pilots. However, \( \dot{\theta} \), the turning rate of the line of sight, must be obtained. To do so note that

\[ x_B - x_F = R \cos \theta \]

Then

\[ \frac{d}{dt}(x_B - x_F) = \dot{R} \cos \theta - R \dot{\theta} \sin \theta \]

\[ = V_B \cos \beta_B - V_F \cos \beta_F \]

With this and the expression for \( \dot{R} \), (B.3-1), rearrangement gives

\[ R \dot{\theta} \sin \theta = V_F (\cos \beta_F - \cos \alpha_F \cos \theta) \]

\[ - V_B (\cos \beta_B - \cos \phi_F \cos \theta) \]

But \( \cos \beta_B = \cos(\theta - \phi_F) \) and \( \cos \beta_F = \cos(\theta - \alpha_F) \); hence

\[ R \dot{\theta} \sin \theta = V_F \sin \theta \sin \alpha_F - V_B \sin \theta \sin \phi_F \]

and finally the turning rate of the line of sight is given by

\[ \dot{\theta} = \frac{V_F \sin \alpha_F - V_B \sin \phi_F}{R} \quad \text{(B.3-4)} \]
APPENDIX C
THE DEL PURSUIT COURSE

The Decreasing Lag Pursuit Course forms one of the basic tactical considerations of the ATAC-2 Model. The course dictates the manner in which each aircraft pursues its enemy; the method of attack. This section describes and defines this course. Also, some of the implications of the course are shown.

C.1 Rationale

The development of the DEL Pursuit Course was motivated by the assumed objectives of an attacking aircraft engaged in air-to-air combat. These objectives are two-fold; it is assumed that in general a pursuer desires

1) to avoid a nose-on attack, and
2) to attack the enemy from behind.

The nose-on attack is avoided to help prevent the enemy from firing its weapons. Further, the pursuer's weapons are in general not as effective when fired at an aircraft traveling towards it. Also, being in the front hemisphere of an enemy while traveling towards it is an unstable situation. On the other hand, a steady state situation off the tail of an enemy at a range and angle-off at which its more lethal weapons may be launched is a very desirable situation. Of course, counter examples to the desirability of these objectives under more specific situations may be pointed out; these assumptions are for a rather general case.
A pursuit course that helps an aircraft accomplish these objectives is one that tends to avoid head-on attacks while helping acquire the tail of the enemy. Such a course is the DEL Pursuit Course. This course dictates that a pursuer point more and more towards the enemy when approaching a tail chase.

The DEL Pursuit Course specifies the amount by which an aircraft will deviate from a pure pursuit course (pointing at the target) as a function of where it is with respect to the target. When in the front hemisphere of its enemy the DEL Pursuit Course requires the pursuer to deviate by a certain amount. Then as the pursuer swings around, decreasing the angle-off, this deviation decreases. Finally, as a tail chase is approached the deviation goes to zero, so that a pure pursuit course is flown. The DEL Pursuit Course is defined by specifying $\alpha_1$ (the tracking angle) as a function of $\phi_1$, the angle-off. Figure C.1-1 gives the general form of the function.

There are two parameters in the DEL Pursuit Course function; $\alpha_{\text{MAX}}$ and $\phi^*$. The function decreases between $\pi$ and $\phi^*$ for decreasing $|\phi_1|$ and is zero for $|\phi_1| \leq \phi^*$. Thus, there is a cone, with half-angle $\phi^*$, off the tail of the target in which the pursuer will fly a pure pursuit course (setting $\alpha_1 = 0$). Further, $\alpha_{\text{MAX}}$ is the largest deviation and this value occurs only at $\phi_1 = \pm \pi$ or when the enemy points directly at the aircraft flying a DEL Pursuit Course. A graphic description is shown in Figure C.1-2.
Figure C.1-1 Defining Function of the DEL Pursuit Course
The angle $\alpha_i$ decreases linearly from $\alpha_{\text{MAX}}$ to 0 between $\phi_i = \pi$ and $\phi_i = \phi^*$. 

Figure C.1-2 A DEL Pursuit Course
C.2 Formulation

To specify the general form of the DEL Pursuit Course it is necessary to define some terms. In general, the term lag implies pointing to the rear of the target while the term lead implies pointing ahead. With this in mind consider Figure C.2-1. It is seen in (a) that F is pointing ahead of B or leading B while in (b) that F is pointing behind or lagging B. These positions are generalized by the following definitions:

1) a lag position is one in which the velocity vectors lie on opposite sides of the line of sight, and

2) a lead position is one in which both velocity vectors lie on the same side of the line of sight.

Lag or lead courses are courses in which the positions at all times are lag or lead, respectively. This is interpreted symbolically as follows: A lag position implies that \( \text{sgn} (a_{1}) \neq \text{sgn} (\phi_{1}) \); a lead position implies that \( \text{sgn} (a_{1}) = \text{sgn} (\phi_{1}) \).

Now it will be noticed, again in Figure C.2-1 that in (a) B also lags F while in (b) B leads F. Indeed, this is the general case as the definition implies. That is, if F lags/leads B, then correspondingly B lags/leads F. F cannot lag B while B leads F, or vice-versa. It is well to note four singularities at this point. They occur when \( a_{1} = 0 \) or \( \phi_{1} \), and \( \phi_{1} = 0 \) or \( \phi_{1} \) (see Figure C.2-2). Here both velocity vectors lie on the same line. And in any of these cases it is said that both aircraft lead and lag at the same time.

To give a general expression for the tracking angle of an aircraft flying a DEL Pursuit Course it is necessary to define the desired tracking angle \( \theta \) as the angle to which \( a_{1} \) will be set whenever aircraft \( 1 \) is flying a
Figure C.2-1 Lead (a) and Lag (b) Positions
Figure C.2-2 Singular Cases
DEL Pursuit Course. And symbolically expressing Figure C.1-1 in slope-intercept form gives

\[
\eta = \begin{cases} 
\text{sgn}(\phi_i) \frac{a_{\text{MAX}}(i) (|\phi_i| - \phi^*)}{\phi^* - \pi}, & \text{if } |\phi_i| > \phi^* , \\
0, & \text{if } |\phi_i| \leq \phi^*. 
\end{cases}
\]

(C.2-1)

Now since \(0 < \phi^* < \pi/2\) (see derivation in Section C.4) the term \(\text{sgn}(\phi_i) (|\phi_i| - \phi^*)/(\phi^* - \pi)\) will always have the opposite sign of \(\phi_i\).

Then the final sign of \(\eta\) will be determined by the sign of \(a_{\text{MAX}}(i)\).

If \(a_{\text{MAX}}(i) > 0\), then \(\text{sgn}(\eta) \neq \text{sgn}(\phi_i)\), whereas if \(a_{\text{MAX}}(i) < 0\), then \(\text{sgn}(\eta) = \text{sgn}(\phi_i)\). Also, if \(\text{sgn}(\eta) = +1\) a lag course will be called for, while if \(\text{sgn}(\eta) = -1\) a lead course will be called for. Again due to the fact that F's leading B implies B leads F, it is impossible to require one aircraft to lead and one to lag. Hence, a requirement is that \(a_{\text{MAX}}(F)\) and \(a_{\text{MAX}}(B)\) have the same sign for a lag.

Equation (C.2-1) is an abbreviated form. For purposes of generality the capability of specifying a constant deviation angle was introduced. This is accomplished by subtracting the quantity \(\text{sgn}(\phi_i) \lambda_i\) from both forms of the expression given in (C.2-1), where \(\lambda_i\) is the desired constant lag (lead) angle of aircraft \(i\).

Thus the final form of the DEL Pursuit Course is given by
\[
\eta = \begin{cases} 
\text{sgn}(\phi_1) \left( \frac{\alpha_{\text{MAX}}(i) (|\phi_1| - \phi^*)}{\phi^* - \pi} - \lambda_1 \right), & \text{if } |\phi_1| > \phi^* \\
- \text{sgn}(\phi_1) \lambda_1, & \text{otherwise}
\end{cases}
\]

(C.2-2)

Note that in the generalized form, the largest lag or lead angle is not
\[|\alpha_{\text{MAX}}(i)|\] but \[|\alpha_{\text{MAX}}(i) + \lambda_1|\].

C.3 Implications of the DEL Pursuit Course

The final form of the DEL Pursuit Course is rather general. For example, a pure pursuit course will be flown by aircraft \(i\) if \(\alpha_{\text{MAX}}(i)\) and \(\lambda_1\) are both set to zero, thus giving \(\eta\) as zero. Also, a deviated pursuit course (with constant deviate angle) may be specified by setting \(\alpha_{\text{MAX}}(i)\) to zero and \(\lambda_1\) to the desired deviate angle. However, the usual situation is to set \(\lambda_1 = 0\) so that at, say, \(\phi_1 = \pm \pi\) the aircraft will deviate by setting \(|\alpha_1| = \alpha_{\text{MAX}}(i)\) and tend to swing around behind its enemy, linearly decreasing its tracking angle down to zero at \(\phi_1 = \pm \phi^*\). Thereafter aircraft \(i\) will fly a pure pursuit course, setting \(\alpha_1 = 0\).

Since \(\alpha_{\text{MAX}}(i)\) is a parameter of the function defining the tracking angle at all points along the DEL Pursuit Course, its magnitude influences how well the attacker will get to the rear of its enemy; whether or not the objective is obtained. In Figure C.3-1 various DEL Pursuit Course paths are drawn against a non-maneuvering target, for different values of \(\alpha_{\text{MAX}}(i)\). These paths are drawn in relative \((r, \phi)\) coordinates where \(r\) corresponds to the range between the aircraft and \(\phi\) the angle-off. Of course, the initial point and relative velocities influence the shape; also, the
Figure C.3-1  DEL Pursuit Course Paths for Different Values of $a_{\text{MAX}}(i)$
conclusion is that increasing the value of $\alpha_{MAX}(i)$ tends to force the path out and away from the target. The path is longer and takes more time to execute, but at any given time, points on the paths with larger values of $\alpha_{MAX}(i)$ are further from the target.

An interesting implication of the DEL Pursuit Course is seen as one considers what happens when both aircraft fly such courses. Suppose then that they do both fly DEL Pursuit Courses specified by (C.2-1) for $i = F, B$. The notation may be abbreviated to require that $\alpha_i$ be a linear function of $\phi_i$ for $i = F, B$. Then

$$\alpha_F = K_F \phi_F + C_F$$

and

$$\alpha_B = K_B \phi_B + C_B$$

where $K_F$, $C_F$, $K_B$, and $C_B$ are constants. Then differentiating with respect to time gives

$$\dot{\alpha}_F = K_F \dot{\phi}_F$$

and

$$\dot{\alpha}_B = K_B \dot{\phi}_B$$

Since $\alpha_i$ and $\phi_i$ are negative supplements of each other

$$\dot{\alpha}_B = \dot{\phi}_F$$

and

$$\dot{\alpha}_F = \dot{\phi}_B$$

- 20 -
hence

\[ \dot{\alpha}_F = K_F \alpha_B = K_F K_B \dot{\alpha}_B \]

\[ = K_F K_B \dot{\alpha}_P \]

And from this it is concluded that either \( K_F K_B = 1 \) or \( \dot{\alpha}_P = \dot{\alpha}_B = 0 \).

But \( K_F \) and \( K_B \) are chosen. Hence, whenever \( K_F K_B \neq 1 \) then \( \dot{\alpha}_P = \dot{\alpha}_B = 0 \) and the aircraft fly at constant tracking angles whenever they are both flying DEL Pursuit Courses. Thus, if both aircraft have the turning capability to fly the DEL Pursuit course, both aircraft are thwarted from reducing their angles-off.

C.4 Derivation of \( \phi^* \)

This section contains a brief derivation of the angle \( \phi^* \). This angle is used as a parameter of the DEL Pursuit Course function. It is defined as the angle of maximum lateral g's of a pursuer flying pure pursuit. It assumes a constant pursuer velocity and a linear non-maneuvering target with constant velocity greater than half the pursuer's velocity.

Suppose \( F \) is pursuing \( B \) with pure pursuit at constant velocity \( V_F < 2 V_B \). Also suppose that \( \beta_B = 0 \). Since \( F \) is on pure pursuit \( \dot{\alpha}_F = \alpha_P = 0 \). Hence

\[ \dot{\theta} = \frac{-V_B}{R} \sin \phi_F \]

\[ \dot{R} = V_B \cos \phi_F - V_F \]

and

\[ \ddot{\beta}_F = \dot{\phi}_F \]

Further, since \( F \) is at constant velocity, the point of maximal lateral g's is also the point of maximal turning rate, \( \beta_F \). And a necessary
condition for \( \beta_F \) to be maximal is that \( \beta_F = 0 \). And

\[
\beta_F = \dot{\theta} = \frac{\partial}{\partial R} \left( \frac{d\theta}{dt} \right) \frac{d\theta}{dt} + \frac{\partial}{\partial \phi_F} \left( \frac{d\theta}{dt} \right) \frac{d\theta}{dt}
\]

\[
= \frac{V_B}{R^2} \sin \phi_F \left( V_B \cos \phi_F - V_P \right) + \frac{V_B^2}{R^2} \sin \phi_F \cos \phi_F
\]

\[
= \frac{V_B}{R^2} \sin \phi_F \left( 2V_B \cos \phi_F - V_P \right)
\]

And by definition

\[
\frac{d\Phi}{dt} \bigg|_{\phi_F = \phi^*} = 0
\]

hence there are two roots; namely \( \phi^* = 0 \) and \( \phi^* = \cos^{-1} \left( \frac{V_P}{2V_B} \right) \).

The point of \( \phi_F = 0 \) is also the point of zero range, except for the special case in which the fighter started pursuit at \( \phi_F = 0 \) -- an uninteresting situation. Thus, the applicable non-zero root is

\[
\phi^* = \cos^{-1} \left( \frac{V_P}{2V_B} \right)
\]

Note that only necessary conditions have been given; sufficient conditions may be demonstrated by showing that this is an absolute maximum. Also, the case of \( V_P > 2V_B \) has not been dealt with. For this condition it can be shown that the lateral \( \gamma^* \)'s of the pursuer increase without bound. Hence, no \( \phi^* \) exists for this case. In that case, \( \phi^* \) is set to zero.
APPENDIX D

TURNING RATE

The turning rate of an aircraft, \( \dot{\beta} \), is the time rate of change of the heading angle, the direction of travel. As such it is related to the g's and speed of the aircraft. From equation (4.6-2) an aircraft at speed \( V \) pulling \( G \) total g's will turn at an angular rate of

\[
\dot{\beta} = \left( \frac{32.2}{V} \right) (G^2 - 1)^{1/2}
\]

But the maximum g's an aircraft can sustain are in general a function of speed described in Appendix E. Hence, the maximum turning rate that an aircraft can sustain is also a function of speed. Now, throughout the development of ATAC-2 various considerations have led to the conclusion that this \( \dot{\beta} \) function of speed is a better measure of performance and capability than is the g function of speed. Some of these considerations are now given.

To show the importance of the turning rate capability of an aircraft, a rather contrived situation is considered first. This is an extreme example. Suppose two dissimilar aircraft have agreed to a duel under the "old code". They start back to back, fly linearly for say 10 seconds, (at possibly differing speeds) then at the same time start to turn, and finally fire their missiles, (see Figure D-1). Suppose also that each has the same weapon. The weapon may be fired at any range but with an angular constraint. The firing aircraft must point at its target to within 5 degrees to fire. Further suppose that each weapon has a zero time of flight and constant kill probability. Then, since each aircraft
starts its turn at the same time, the aircraft that turns the quickest fires first and thereby wins the duel. It matters little how many g's each pulls but rather the time it takes to turn through about 180 degrees.

Indeed this is a contrived situation. However, consider what happens without the agreement to duel by the code. What happens after each aircraft has passed its opponent head-on? Each aircraft tries to turn around and fire at its opponent before being fired upon. Then the distance between the centers of the combatant's circles of turn has decreased from the above example. However, the situation is quite analogous. And again the objective of each combatant is to turn as fast as possible so as to point near its opponent first. To be sure the situations are not completely similar. However, closer analysis will show that for a vast range of relative parameters, it is the aircraft that turns the quickest that points at its enemy first. This then is one reason turning rate is an important consideration of performance. An aircraft desires to turn as fast as possible,
irrespective of from what initial angle it started, so as to point near its target and be able to fire weapons.

Another reason for the desire of high turning rate is general maneuvering. In Appendix F it is shown that one of the necessary conditions for one aircraft to follow another in steady state while on pure pursuit is that it turns as fast as the target - that $\dot{\beta}$ of the pursuer equals $\beta$ of the evader. This requirement holds even if the pursuer is allowed to fly a deviated pursuit course (one in which its tracking angle is a constant not necessarily zero). It must still be able to turn as fast as its opponent in order to hold its range and angle-off constant. Further, it is seen that the pursuer in steady state on pure pursuit pulls less g's than the evader at all non-zero steady state ranges. From the evader's standpoint it is also the case that high turning rate helps. For the greater the disparity between the turning rate of the evader over the pursuer, the sooner will the evader extricate itself from this unhealthy position of having the enemy on its tail. Merely pulling more g's than the pursuer does not necessarily help its situation.

Thus, it is for these and other reasons that one is led to consider high turning rate capability as beneficial and the $\beta$ function of speed as an important indicator of performance. And this then is the justification for many of the tactics of the model that require an aircraft in certain situations to tend to $V^*$, the speed at which the sustainable turning rate is an absolute maximum.
APPENDIX E

SPECIFIC POWER FUNCTION

In general, the performance of an aircraft is thought to be its ability to climb and dive, to turn, to speed up and slow down as well as its maximum speed, etc. Further, most all of these abilities are interrelated; the ability to climb is a function of lateral g level, the ability to turn is a function of speed, the maximum speed is a function of altitude, and so on. Yet there is a concept which helps unify many of these abilities; this is the concept of specific power, $P_s$. The literature abounds with various treatments of specific power, see for example Boyd [Ref. 4], so that only a brief description will be given here. The method in which ATAC-2 makes use of the concept of specific power, however, is discussed in detail.

The specific power of an aircraft is defined to be the time rate of change of the specific energy level. This includes the energy due to altitude (potential) and the energy due to speed (kinetic). Thus, for an aircraft at altitude $h$ and velocity $V$ the specific energy is given by

$$E_s = h + \frac{V^2}{2 \cdot 32.2}$$

then

$$P_s = \frac{dE_s}{dt}$$

$$= \frac{dh}{dt} + \frac{V}{32.2} \frac{dV}{dt} \tag{E-1}.$$ 

1) Throughout the term specific is with respect to the weight of the aircraft.
Then specific power is composed of two components; the power of altitude change and the power of speed change. An aircraft at a positive specific power level will increase its energy level. It may do this by linearly accelerating \( \frac{dh}{dt} = 0 \) at a rate of \( \frac{dV}{dt} = P_s \left( \frac{32.2}{V} \right) \), for example. Or the aircraft could accomplish this by climbing at constant speed \( \frac{dV}{dt} = 0 \) at a rate of \( \frac{dh}{dt} = P_s \). In fact the aircraft could combine the two (climbing and increasing its speed) in any manner that is consistent with (E-1). A completely analogous situation applies for a negative specific power. Here the aircraft will lose energy. This may be accomplished by losing altitude in any combination with losing speed again consistent with (E-1). Thus, the specific power of an aircraft specifies the rate of change of the energy level of the aircraft. The rate of change of the altitude and of the speed at a given speed.

By definition then the specific power of an aircraft has units of power (ft-lbs/sec) divided by weight (lbs) or ft/sec. This is reflected in (E-1) for \( \frac{dh}{dt} \) and \( \frac{V}{32.2} \frac{dV}{dt} \) also have units of feet per second.

One might logically question then how the specific power function of an aircraft is characterized. How is it specified? [Ref. 4] shows that for an aircraft with thrust \( T \), drag \( D \), weight \( w \) and speed \( V \) the level of specific power is

\[
P_s = \frac{(T - D) V}{w} \quad \text{ (E-2)}
\]

Thus to specify the specific power of an aircraft, knowledge of engine and aerodynamic properties at various speeds is required. Now the thrust...
of an aircraft is a function of the speed, altitude and power setting. Also the drag of an aircraft is a function of speed, altitude and lateral g's. Consequently, the specific power of an aircraft is a function of nearly every other possible state parameter, namely, speed, power setting, altitude and lateral g's. At a fixed power setting and a fixed altitude specific power is a function of speed and lateral g'a. This is built into the ATAC-2 model. The aircraft is assumed to be at a fixed power setting and at a constant altitude. Henceforth, then the specific power function is taken to be a function of speed and lateral g's.

An example of the specific power function of a fictitious, though not atypical, aircraft is shown in Figure E-1. Each line shown gives the value of the specific power function along the ordinate for the associated speed along the abscissa and total g level labeled. (The total g's of an aircraft are the lateral g's plus the g due to gravity, added vectorially.) Also associated with this diagram is a fixed altitude and power setting. Now for a 1g level the function is positive at all speeds shown. Thus, the aircraft will accelerate at this constant altitude. However, for higher g levels, say about 3.5 g's, the specific power function is negative for all speeds indicated. Thus, the aircraft will decelerate. From equation (E-1) it is seen that the acceleration of the aircraft at this constant altitude is given by $32.2 \frac{P_g}{V}$. And if the function is negative the aircraft will decelerate, while it will gain speed if the function is positive.

More information than just the acceleration capability of an aircraft may be obtained from this diagram. The maximum speed $V_{max}$ is just the limit of the lines indicated on the right in Figure E-1. This limit at
Specific Power Function of Velocity and g's for an Aircraft at Constant Altitude and Power Setting.
this altitude is a heat limit. At some other altitude, however, the
maximum speed may be a power limit. The minimum speed $V_c$ is given by
the extent of the lines on the left as shown. Also, the maximum $g$'s that
the aircraft may sustain is indicated by the extent of the $g$ lines. This
is the limit to which the aircraft's structure is able to sustain $g$'s.
In this example the limit is given by 4 $g$'s. Also, more noteworthy
information is available from this specific power function. The dotted
line at the left indicates the limit of the speed an aircraft may fly at
while at a given $g$ level. This is the stall limit; if the aircraft flies
at less than this speed, it begins to lose its lift, starts to buffet and
finally stalls. Yet the concept may be viewed from the other way also,
namely, as the maximum amount of $g$'s the aircraft may sustain at a given
speed without stalling. Then the whole lower line (dotted on the left, solid
otherwise) may be viewed as the maximum total $g$'s the aircraft may attain.
This $g$ limit is itself a function of speed as indicated. The symbol used
for this function is $G(V)$.

There is one important concept which has yet to be discussed.
This is the concept of the $g(V)$ function. To illustrate this concept
an example is useful. Consider an aircraft at some speed and 1 total $g$.
Suppose the aircraft begins to pull more and more $g$'s so that it continues
to lose specific power (see E-1). As the aircraft continues to pull more
$g$'s it passes a singular transition point. This is the point at which
the specific power function is zero; the point where the function changes
sign. Clearly an aircraft that pulls more $g$'s at that speed will lose speed.
At the constant speed vertical line this value is about 3.0 $g$'s. At other
velocities the relevant value is, of course, different. Thus, it is a
function of velocity. Hence $g(V)$ is the total $g$'s as a function of speed,
V, at which the specific power function is zero. This may be viewed as the sustainable g's at the given speed; for the aircraft will neither accelerate nor decelerate at \( g(V) \) total g's. Then, if an aircraft wishes to pull its maximum g's at, say, speed \( V \), and not decelerate, it can pull no more than \( g(V) \) total g's.

In summary then the specific power function, indeed, provides a great deal of knowledge about the performance of the aircraft. The value of the function itself gives the rate at which the aircraft will accelerate at some g level and velocity. Also, the limits of the function give the maximum velocity and the maximum total g's the aircraft may achieve. Finally, the \( g(V) \) function is obtained from the specific power function; it is the g level at which the velocity \( V \) is just maintained.
This appendix gives the derivation of the necessary steady state conditions. Also, the applicability of the steady state conditions to ATAC-2 is discussed.

These conditions are applicable when one aircraft follows another that turns. One aircraft tries to maneuver by turning while the other tries to stay behind it. Under this situation then, suppose \( P \) is following \( B \) in the steady state defined to be the condition of constant range, \( R = 0 \), and constant angle-off \( \Phi_p = 0 \). Since the fighter is assumed or pure pursuit, \( \Phi_p = 0 \). Then from (B.3-2), (B.3-3), and (B.3-4) it is seen that

\[
\frac{\dot{\Phi}_p = 0}{\dot{\Phi}_p} \implies \frac{\dot{\alpha}_p = 0}{\dot{\alpha}_p} \implies \frac{\dot{\Phi}_F = \dot{\Phi}_F}{\dot{\Phi}_B = \dot{\Phi}_B} \quad (P-1)
\]

and

\[
R = \frac{V_B \sin \Phi_F}{\dot{\Phi}_B} \quad (P-2)
\]

From (B.3-1),

\[
\dot{R} = 0 \implies \dot{V}_F = V_B \cos \Phi_F \quad (P-3)
\]

These then are the necessary conditions for \( P \) to follow \( B \) in steady state on pure pursuit.
To observe some of the singularities of this situation consider Figure F-1. Suppose that $F$ desires to follow $B$ in steady state at a range $R'$ and angle-off $\phi'$ and a speed $V_F'$. Then among the three parameters $V_F'$, $\phi'$, and $R'$ the fighter is free to pick only one of these, for the other two will be determined by this choice. For example, suppose $\phi'$ is chosen. Then (F-3) gives the $V_F'$ associated with this $\phi'$ while (F-2) gives the $R'$; assuming, of course, that the bomber continues at $V_B$ and $\dot{\theta}_B$.

Yet even this choice of one of the three parameters is limited. It is seen from Figure F-2 that the fighter's speed has an upper bound of $V_B$; it can fly no faster than the target in steady state. Also, if $V_C(F)$ is the minimum speed of the fighter then there is a definite upper bound to $\phi'$ given by $\cos^{-1}\left(V_C(F)/V_B\right)$. And it is seen from Figure F-3 that an upper bound on $R'$ is the bomber's radius $r_B = V_B/\dot{\theta}_B$. So there are definite limits to the choice of $R'$, $\phi'$ and $V_F'$.

Further considerations of the above conditions influence the tactics of the model. Since $\dot{\theta}_p$, the turning rate of the fighter, is bounded by the maximum turning rate (determined by the speed), the bomber can prevent the fighter from obtaining a steady state condition by making $\dot{\theta}_B$ larger than this bound. For example, suppose two identical aircraft engage each other. Also suppose that the implied $\dot{\theta}$ function of speed for both aircraft has an absolute maximum at $V^*$. Then if the bomber flies at speed $V^*$, turning at its maximum rate, the fighter would not be able to obtain a steady state at any non-zero range.
Figure F-1 Steady State Parameters
Figure F-2  Steady State Velocity as Function of Angle-Off
Figure F-3  Steady State Range as Function of Angle-Off
Of course, similar considerations may be made with dissimilar aircraft by comparing their $\dot{\phi}$ functions of speed. In fact, one can determine before hand which, if either, of two combatants is able to follow another in the steady state. Further, the range of the parameters $R'$, $\phi'$, $V_p'$ which will accomplish a steady state can be evaluated, by noting the $\dot{\phi}$ function of speed, and applying equations (F-1), (F-2), and (F-3).

A word of caution may be in order here. To say that the steady state is unattainable for one aircraft or another is indeed not to imply that one is inferior. For the transient state or near steady state may be quite easily attained and is often quite sufficient in order to remain behind a target for a long time.
LIST OF REFERENCES

[1] Cerone, J. R., T. J. Harris, B. J. Jacobs, W. J. Strauss, 
Fighter Vs. Fighter Combat: ATAC-2 Model, Volume I, II, 

Fighter Combat: ATAC-2 Model: Double Search, CSA Report 

[3] Personal Communications, Dr. Hitchcock, U. S. Naval Air 

Air Force Systems Command, Eglin Air Force Base, APGC-TR-66-4, 
Volume 1, March, 1966 (Secret).

ATAC-2 is a simulation model designed to help evaluate fighters in air-to-air combat. The model treats the one vs. one dogfight which arises from a random search situation. Both aircraft in the combat are (usually) aggressive. The two principal outputs from the model are the probability a given aircraft is killed in the fight and the expected number of enemy aircraft an aircraft kills over its useful life. Combat is restricted to a fixed altitude. The maneuvers are dynamic in that each aircraft responds to the situation at each moment in a duel depending on the information it has about an opponent’s activities.(U)

Inputs include, for each aircraft, search and tracking radar characteristics, passive radar sensors, optical capability, IFF, energy-maneuverability data, weapon loadings, weapon characteristics, and weapon kill probabilities. (U)

The rationale for the model specifics are presented. Flow charts and program listings are included. The model has been run repeatedly on an IBM 7094. (U)
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPERATIONS RESEARCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMULATION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIR-TO-AIR COMBAT MODEL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COST EFFECTIVENESS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TACTICS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYNAMIC MODEL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:**

**COOP RANKING**

- [X]
- [X]
- [X]