MATH MODEL DESCRIPTION FOR THE VISUAL TECHNOLOGY RESEARCH SIMUL--ETC(U)

JAN 82  E N HOLLER

UNCLASSIFIED  NAVTRAEGUIPC-14-335
TECHNICAL REPORT: NAVTRAEEUICEN IM-335

MATH MODEL DESCRIPTION FOR THE VISUAL TECHNOLOGY RESEARCH SIMULATOR (VTRS)
CONVENTIONAL TAKEOFF AND LANDING (CTOL)
WEAPON DELIVERY VISUAL SYSTEM

Edward M. Holler
Visual Technology Research Simulator
Naval Training Equipment Center
Orlando, FL 32813

January 1982

Final Report

DoD Distribution Statement
Approved for Public Release
Distribution Unlimited

Naval Air Systems Command
Code 4133J
Washington, D.C. 20361

82 07 21 003
MATH MODEL DESCRIPTION FOR THE
VISUAL TECHNOLOGY RESEARCH SIMULATOR (VTRS)
CONVENTIONAL TAKEOFF AND LANDING (CTOL)
WEAPON DELIVERY VISUAL SYSTEM

Edward M. Holler
Code N-732
Advanced Simulation Concepts Laboratory
Naval Training Equipment Center
Orlando, Florida 32813

January 1982

GOVERNMENT RIGHTS IN DATA STATEMENT

Reproduction of this publication in
whole or in part is permitted for any
purpose of the United States Government.

Approved:

DAVID P. GLENN
Director, Research Department

A. M. SPOONER
Head, Advanced Simulation Concepts Laboratory

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813
**Title:** Math Model Description for the Visual Technology Research Simulator (VTRS) Conventional Take Off and Landing (CTOL) Weapon Delivery Visual System

**Authors:** Edward M. Holler

**MONITORING AGENCY NAME AND ADDRESS:**
Visual Technology Research Simulator
Naval Training Equipment Center
Orlando, FL 32813

**NUMBER OF PAGES:** 134

**DISTRIBUTION STATEMENT (of this Report):** Approved for Public Release; distribution unlimited.

**ABSTRACT:** This report provides a technical description of the software for weapon delivery simulation for the T-2C aircraft at the Visual Technology Research Simulator (VTRS) facility. A summary of the complete system is provided first and then each of the software modules is described in detail. System flow charts, axis system diagrams, math model vector diagrams, and logic tables are provided. The mathematical equations are done in Fortran. Pictures of visual scenes and moving models are also provided to illustrate system capability.

**Key Words:** Visual Technology Research Simulator, Flight Simulation, Weapon Delivery Simulation
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>7</td>
</tr>
<tr>
<td>INTRODUCTION.</td>
<td>7</td>
</tr>
<tr>
<td>FLIGHT AND VISUAL DISPLAY EQUIPMENT</td>
<td>7</td>
</tr>
<tr>
<td>COMPUTER IMAGE GENERATION EQUIPMENT</td>
<td>8</td>
</tr>
<tr>
<td>VTRC CTL SYSTEM</td>
<td>8</td>
</tr>
<tr>
<td>VISUAL SOFTWARE OVERVIEW.</td>
<td>13</td>
</tr>
<tr>
<td>MODULES</td>
<td>13</td>
</tr>
<tr>
<td>SUBROUTINES</td>
<td>13</td>
</tr>
<tr>
<td>VISUAL MODULE/SUBROUTINE SUMMARY.</td>
<td>15</td>
</tr>
<tr>
<td>FLIGHT/VISUAL INTERFACE</td>
<td>15</td>
</tr>
<tr>
<td>WINDOW/POINT OF INTEREST.</td>
<td>15</td>
</tr>
<tr>
<td>TARGET IMAGE PROJECTOR.</td>
<td>15</td>
</tr>
<tr>
<td>MOVING MODEL CONTROL.</td>
<td>15</td>
</tr>
<tr>
<td>WEAPON SCORING.</td>
<td>15</td>
</tr>
<tr>
<td>VISUAL/CIG INTERFACE.</td>
<td>15</td>
</tr>
<tr>
<td>MOVING MODEL SUBROUTINES.</td>
<td>16</td>
</tr>
<tr>
<td>II</td>
<td>17</td>
</tr>
<tr>
<td>FLIGHT/VISUAL INTERFACE MODULE.</td>
<td>17</td>
</tr>
<tr>
<td>FLIGHT/VISUAL INTERFACE MODULE FUNCTIONAL DESCRIPTION</td>
<td>17</td>
</tr>
<tr>
<td>FLIGHT/VISUAL FUNCTIONS</td>
<td>19</td>
</tr>
<tr>
<td>III</td>
<td>27</td>
</tr>
<tr>
<td>WINDOW/POINT OF INTEREST MODULE</td>
<td>27</td>
</tr>
<tr>
<td>WINDOW/POINT OF INTEREST FUNCTIONAL DESCRIPTION</td>
<td>27</td>
</tr>
<tr>
<td>IV</td>
<td>34</td>
</tr>
<tr>
<td>TARGET IMAGE PROJECTOR.</td>
<td>34</td>
</tr>
<tr>
<td>TARGET IMAGE PROJECTOR MODULE FUNCTIONAL DESCRIPTION.</td>
<td>34</td>
</tr>
<tr>
<td>V</td>
<td>42</td>
</tr>
<tr>
<td>MOVING MODEL CONTROL.</td>
<td>42</td>
</tr>
<tr>
<td>MOVING MODEL CONTROL MODULE FUNCTIONAL DESCRIPTION.</td>
<td>42</td>
</tr>
<tr>
<td>VI</td>
<td>46</td>
</tr>
<tr>
<td>WEAPON SCORING.</td>
<td>46</td>
</tr>
<tr>
<td>WEAPON SCORING MODULE FUNCTIONAL DESCRIPTION.</td>
<td>46</td>
</tr>
<tr>
<td>VII</td>
<td>52</td>
</tr>
<tr>
<td>VISUAL/CIG INTERFACE.</td>
<td>52</td>
</tr>
<tr>
<td>VISUAL/CIG INTERFACE MODULE FUNCTIONAL DESCRIPTION.</td>
<td>52</td>
</tr>
</tbody>
</table>

## APPENDICES

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>65</td>
</tr>
<tr>
<td>AIRCRAFT CARRIER DRIVER</td>
<td>65</td>
</tr>
<tr>
<td>CARRIER DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION</td>
<td>66</td>
</tr>
<tr>
<td>B</td>
<td>69</td>
</tr>
<tr>
<td>FLAK DRIVER SUBROUTINE</td>
<td>69</td>
</tr>
<tr>
<td>FLAK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION</td>
<td>70</td>
</tr>
<tr>
<td>C</td>
<td>79</td>
</tr>
<tr>
<td>FLOS DRIVER</td>
<td>79</td>
</tr>
<tr>
<td>FLOS DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION</td>
<td>79</td>
</tr>
<tr>
<td>D</td>
<td>89</td>
</tr>
<tr>
<td>GENERAL PURPOSE DRIVER</td>
<td>89</td>
</tr>
<tr>
<td>&quot;GENERAL&quot; DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION</td>
<td>89</td>
</tr>
<tr>
<td>E</td>
<td>93</td>
</tr>
<tr>
<td>AIR-TO-SURFACE ROCKET DRIVER SUBROUTINE</td>
<td>93</td>
</tr>
<tr>
<td>ROCKET DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION</td>
<td>93</td>
</tr>
</tbody>
</table>
APPENDICES (CONTINUED)

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>SURFACE-TO-AIR MISSILE (SAM) DRIVER SUBROUTINE.</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>SAM DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION.</td>
<td>97</td>
</tr>
<tr>
<td>G</td>
<td>TA4J DRIVER SUBROUTINE.</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>TA4J DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION</td>
<td>108</td>
</tr>
<tr>
<td>H</td>
<td>TANK DRIVER.</td>
<td>111</td>
</tr>
<tr>
<td></td>
<td>TANK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION</td>
<td>111</td>
</tr>
<tr>
<td>I</td>
<td>TRUCK DRIVER.</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>TRUCK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION</td>
<td>123</td>
</tr>
</tbody>
</table>
**LIST OF ILLUSTRATIONS**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VTRS CTOL System</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>CTOL Hardware Configuration</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>CTOL Cockpit and Dome</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>CTOL Projector System</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>Weapon Visual Software Flowchart</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>Flight Earth Axis System</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>CIG Earth Axis System</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>Body Axis System</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>Relationship Between Aircraft Body Axis and Eyepoint Axis Systems</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>Target Axis System</td>
<td>24</td>
</tr>
<tr>
<td>11</td>
<td>Point of Interest in Observer Axis</td>
<td>27</td>
</tr>
<tr>
<td>12</td>
<td>Point of Interest (POI) Offset in Observer Axis</td>
<td>28</td>
</tr>
<tr>
<td>13</td>
<td>Point of Interest in CIG Earth Axis System</td>
<td>29</td>
</tr>
<tr>
<td>14</td>
<td>Point of Interest Angles in Observer Axis</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>Target Image Projector Position Relative to Eyepoint, Side View</td>
<td>34</td>
</tr>
<tr>
<td>16</td>
<td>Screen Intersection Vectors in Display Axis System</td>
<td>35</td>
</tr>
<tr>
<td>17</td>
<td>The Azimuth Vectors in the Tip Axis System</td>
<td>38</td>
</tr>
<tr>
<td>18</td>
<td>Tip Azimuth Angles in Tip Axis System, Top View</td>
<td>39</td>
</tr>
<tr>
<td>19</td>
<td>Vector Diagram for Bomb Miss Distance</td>
<td>47</td>
</tr>
<tr>
<td>20</td>
<td>Vector Diagram for Rocket/Bullet Miss Distance</td>
<td>47</td>
</tr>
<tr>
<td>21</td>
<td>Vector Diagram for CPA Values Below the Ground</td>
<td>51</td>
</tr>
<tr>
<td>A-1</td>
<td>Carrier Landing Approach on VTRS System</td>
<td>65</td>
</tr>
<tr>
<td>A-2</td>
<td>Carrier Reference Point and Center of Gravity</td>
<td>66</td>
</tr>
<tr>
<td>B-1</td>
<td>Anti-Aircraft FLAK Burst</td>
<td>69</td>
</tr>
<tr>
<td>B-2</td>
<td>Location of FLAK and SAM Launch Pad in CIG Earth Axis System</td>
<td>70</td>
</tr>
<tr>
<td>B-3</td>
<td>FLAK Burst Position in CIG Earth Axis System</td>
<td>73</td>
</tr>
<tr>
<td>C-1</td>
<td>FLOLS Position on Carrier in Carrier Axis System</td>
<td>80</td>
</tr>
<tr>
<td>C-2</td>
<td>FLOLS Position and Attitude for Meridian Airport Data Base</td>
<td>81</td>
</tr>
<tr>
<td>C-3</td>
<td>FLOLS Meatball Position Relative to Aircraft Glideslope</td>
<td>84</td>
</tr>
<tr>
<td>C-4</td>
<td>FLOLS Datum Line and Meatball Position, Side View</td>
<td>86</td>
</tr>
<tr>
<td>E-1</td>
<td>Air-to-Surface Missiles (2.75&quot; Rockets)</td>
<td>93</td>
</tr>
<tr>
<td>E-2</td>
<td>Rocket Centroid Definition in Rocket Axis System</td>
<td>94</td>
</tr>
<tr>
<td>F-1</td>
<td>Surface-to-Air Missile (SAM) Launch</td>
<td>97</td>
</tr>
<tr>
<td>F-2</td>
<td>SAM Launch Pad Setup, CIG Earth Axis System</td>
<td>98</td>
</tr>
<tr>
<td>F-3</td>
<td>SAM Trajectory Vectors, CIG Earth Axis System</td>
<td>100</td>
</tr>
<tr>
<td>G-1</td>
<td>TAAA in Formation Flight</td>
<td>107</td>
</tr>
<tr>
<td>H-1</td>
<td>T-62 Tank</td>
<td>111</td>
</tr>
<tr>
<td>H-2</td>
<td>Tank Route on Gunnery Range</td>
<td>112</td>
</tr>
<tr>
<td>H-3</td>
<td>Tank Route on Twin Towns</td>
<td>115</td>
</tr>
<tr>
<td>H-4</td>
<td>Tank Route in River Valley</td>
<td>118</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS (CONTINUED)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-1</td>
<td>Truck Convoy</td>
<td>123</td>
</tr>
<tr>
<td>I-2</td>
<td>Truck Convoy Route on Gunnery Range</td>
<td>124</td>
</tr>
<tr>
<td>I-3</td>
<td>Truck Convoy Route in Twin Towns</td>
<td>126</td>
</tr>
<tr>
<td>I-4</td>
<td>Truck Convoy in River Valley</td>
<td>130</td>
</tr>
</tbody>
</table>
NAVTRAUEQPCEN IH-335

LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MOVING MODEL OPTIONS FOR DATABASE ENVIRONMENTS</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td>MOVING MODEL CONTROL FLAGS</td>
<td>43</td>
</tr>
<tr>
<td>3(A)</td>
<td>SEL TO DEC INTERFACE FLOATING POINT PARAMETERS 32 BIT WORDS</td>
<td>54</td>
</tr>
<tr>
<td>3(B)</td>
<td>SEL TO DEC INTERFACE CONTROL FLAGS</td>
<td>58</td>
</tr>
<tr>
<td>4</td>
<td>DEC TO SEL INTERFACE (A) FRAME RATE BUFFER</td>
<td>61</td>
</tr>
<tr>
<td>5</td>
<td>DEC TO SEL INTERFACE (B) DATA BASE INITIALIZATION BUFFER</td>
<td>62</td>
</tr>
</tbody>
</table>
SECTION I
INTRODUCTION

This math model report describes the visual software for the Conventional Take-off and Landing (CTOL) system at the Visual Technology Research Simulator (VTRS) facility. The VTRS CTOL system is a real-time flight simulator for the Navy T2C jet aircraft. This system includes a six degree of freedom motion platform, a G-seat, T2C cockpit, an experimenter operator station, and a full color wide angle visual system. Capabilities of this system include aircraft carrier takeoff and landings, airport takeoff and landings, formation flight, and air-to-ground combat. The air-to-ground capabilities include 2.75 inch rockets, 50 caliber machine guns and bombs which can be launched against six moving targets and eight fixed targets. Also included is anti-aircraft flak and surface-to-air missiles which can be launched against the T2C. Hits are scored and displayed for fixed and moving targets and against the pilot.

The visual software referenced in this report, except for Singer-Link executive, target projector routines, and utility programs, was designed, coded, implemented, and documented at the Naval Training Equipment Center (NTEC) as an in-house effort. The original system delivered to NTEC included a Singer-Link wide angle model board visual system and a T2C flight simulator that was limited to aircraft carrier takeoff and landings. NTEC later purchased a General Electric Computer Image Generation (CIG) system and modified the above Singer-Link model board flight simulator so that CIG images could be displayed to the pilot in the dome. Both the GE CIG system and the flight simulator were recently upgraded to permit research in the air-to-ground area. The CIG system upgrade was performed by GE and the flight simulator system upgrade was done by NTEC. This report addresses the new visual air-to-ground software that was developed by NTEC.

The hardware configuration for the VTRS CTOL simulator is illustrated in Figure 1. Note that the simulator is divided into two basic systems: (1) the flight/visual system; and (2) the Computer Image Generation (CIG) system. Photographs of the VTRS CTOL hardware are provided in Figures 2, 3 and 4. This equipment is summarized below:

FLIGHT AND VISUAL DISPLAY EQUIPMENT.

a. Flight cockpit, G-seat, and motion platform.

b. Visual screen (dome), and two-channel projection system.

(1) Target channel has a high resolution servo-driven projection system for target image display.

(2) Background channel has a low resolution wide angle fixed projection system for background scene display.
c. Experimenter operator station (EOS) to control flight and visual functions, initiate data recording, monitor pilot performance, etc.

d. High speed analog/digital interface and computer system drives the simulator and interfaces with the Computer Image Generation (CIG) system. The computer system includes three System Engineering Laboratories (SEL) 32/75 minicomputers, memory, and two control data 9762 eighty mega byte discs.

   (1) Central Processing Unit (CPU) 1 is the master computer for the visual and flight computers and drives the EOS.

   (2) CPU 2 receives data from flight, drives the visual projectors and inputs visual parameters to CIG for visual displays.

   (3) CPU 3 receives pilot parameters from cockpit and computes flight parameters for motion, G-seat, and visual computer.

COMPUTER IMAGE GENERATION EQUIPMENT.

a. Frame I equipment includes two DEC PDP 11/T55 computers and four RKOSJ disc units. One PDP 11/T55 interfaces with the SEL computers and provides inputs for Frame II equipment.

b. Frame II equipment is digital image processing hardware that converts three dimensional data into two dimensional data and performs distortion correction for images so that they will appear undistorted in the dome.

c. Frame III equipment is digital raster processing hardware that inputs Frame II data and puts it in raster format for display in the dome through the background and target projectors.

VTRS CTOL SYSTEM.

An overview of the VTRS CTOL system operation is now provided. The simulator's initial flight conditions, visual functions, CIG environment requests, data monitoring and data recording are controlled from the EOS, which is the primary control device for the simulator. The SEL executive computer processes EOS requests and is the master for the flight and visual SEL computers. The SEL flight computer processes pilot cockpit responses, computes the appropriate flight equations, and drives the cockpit, G-seat, and motion platform. The SEL visual computer inputs aircraft position, attitude, and other pertinent data from flight and computes the appropriate drive parameters for the visual projection hardware in the dome. The CIG system inputs all viewpoint and moving model positions and attitudes, environment requests, etc., from the SEL visual computer. The CIG Frame I computers process these parameters for input to the CIG Frame II hardware which converts three dimensional data into two dimensional data and performs distortion correction on this data so visual pictures will appear undistorted in the dome. Frame III equipment inputs the data from Frame II and puts it in raster
Figure 1. VTRS CTOL System.
Flight Simulator Cockpit and G Seat
Visual System Dome and Motion Platform

Figure 3. CTOL Cockpit and Dome.
Figure 4. CTOL Projector System.

**VISUAL SOFTWARE OVERVIEW.**

The visual software contains six software modules and nine subroutines. All modules and subroutines are written in Fortran. The software modules are called from the visual executive program and execute at a thirty Hertz rate. The visual executive program is written in SEL assembly language. The subroutines are called by the software modules at a thirty Hertz rate, but only when requested by the specified visual task. This significantly reduces visual program execution time. The weapon visual software flow chart is shown in Figure 5 and is broken down as follows:

**MODULES.**

a. ZM6SW#33: Flight/Visual Interface  
b. ZM6SW#36: Window/Point of Interest  
c. ZM6SW#37: Target Image Projector  
d. ZM6SMOVE: Moving Model Control  
e. ZM6SCORE: Weapon Scoring  
f. ZM6SWCGI: Visual/CIG Interface

**SUBROUTINES.**

a. WCARRIER: Carrier Driver  
b. WFLAK : FLAK Driver  
c. WFLOLS : FLOLS Driver  
d. WGENERAL: General Purpose Driver  
e. WROCKETS: Rocket Driver  
f. WSAM : Sam Driver
Figure 5. Weapon Visual Software Flowchart.
g. WTA4J : TA4J Driver 

h. WTANK : Tank Driver 

i. WTRUCK : Truck Driver 

VISUAL MODULE/SUBROUTINE SUMMARY.

The visual software is structured so that each module and subroutine performs a specific function. These functions are summarized below and a detail description is provided in Section II. All mathematical equations in this report are shown in Fortran IV format.

FLIGHT/VISUAL INTERFACE. This module interfaces VTRS flight and visual parameters. Flight parameters are computed by the flight computer (CPU #3) and passed via datapool to the visual computer (CPU #2). This module converts all flight parameters from the Flight Axis System to the CIG Axis System for visual processing.

WINDOW/POINT OF INTEREST. This module computes azimuth and elevation angles for the observer point of interest in the Observer Coordinate System. The point of interest is typically a target such as an aircraft carrier or tank. This module defines the window through which the observer looks to see this target.

TARGET IMAGE PROJECTOR. This module inputs the azimuth and elevation angles from Window/Point of Interest module and converts them into the Target Image Projector coordinates. These angles are then used to drive the target projector servos in the dome. This puts the target in the proper position in the dome.

MOVING MODEL CONTROL. This module looks at the loaded database and then activates the appropriate subroutine to drive the models that exist in that environment. Each database has its own unique models that must be driven by the weapons visual software. For example, in the aircraft carrier database, this module would activate the carrier and Fresnel Lens Optical Landing System (FLOLS) moving models.

WEAPON SCORING. This module scores weapon deliveries of bombs, bullets, and rockets shot from the T2C aircraft. Bombs are scored as ground miss distances. Rockets and bullets are scored as Closest Point of Approach (CPA) above the ground. If hits are scored on the tank or truck, this module sends a kill flag to the CIG system, which will display an explosion on the target.

VISUAL/CIG INTERFACE. This module gets the weapon visual parameters, converts them to PDP 11 computer format as required, and stores them in datapool for transfer to the CIG system. These parameters are used to drive the viewpoint and moving model position and attitudes in the CIG system. The CIG database
parameters from the CIG system are assigned to appropriate weapon visual parameters. These parameters are used to determine which database is loaded and which moving models should be activated.

MOVING MODEL SUBROUTINES. The nine moving model subroutines are listed in Appendixes A-I. These routines are called by the moving model control module. The function of these routines is to drive the CIG models as required by each database. For example, the Truck Driver routine must drive the truck on the road in the specified database. There are currently three different roadways in each of the three databases with trucks. This report includes a detailed section for each of these subroutines.
SECTION II

FLIGHT/VISUAL INTERFACE MODULE

This module interfaces flight parameters and visual parameters. All flight coordinates are converted from Flight to CIG coordinates in this module. All weapon visual parameters are in the CIG Coordinate System.

FLIGHT/VISUAL INTERFACE MODULE FUNCTIONAL DESCRIPTION.

The coordinate systems are the reference system through which the visual and flight world are defined. A complete understanding of these axis systems is required to comprehend the mathematical description in this report. The following coordinate descriptions should be thoroughly understood before proceeding any further in this section.

The Flight Earth Axis System is shown in Figure 6. It is a standard right-handed rectangular coordinate system with its axis and angles defined below. The preferred order of rotation is yaw, pitch, and then roll. The axis system is summarized as follows:

a. +X is pointed north.

b. +Y is pointed east.

c. +Z is pointed down.

d. +yaw is defined as a right-hand rotation about the Z axis.

e. +pitch is defined as a right-hand rotation about the Y axis.

f. +roll is defined as a right-hand rotation about the X axis.

\[ X_E (NORTH) \]
\[ Y_E (EAST) \]
\[ Z_E (DOWN) \]

Figure 6. Flight Earth Axis System.
The CIG Earth Axis System is shown in Figure 7. Its axis system differs from the Flight system as follows: (1) The X and Y axis are interchanged; (2) the Z axis is in the opposite direction. The preferred order of rotation of this axis system is also yaw, pitch, roll. This axis system is summarized as follows:

a. +Y is pointed north.
b. +X is pointed east.
c. +Z is pointed up.
d. + yaw is defined as a right-hand rotation about the Z axis.
e. + pitch is defined as a right-hand rotation about the X axis.
f. + roll is defined as a right-hand rotation about the Y axis.

![CIG Earth Axis System Diagram]

Figure 7. CIG Earth Axis System.

The body axis for both the Flight and CIG systems is shown in Figure 8. These axis systems are summarized as follows:

a. +X is out the nose of the aircraft.
b. +Y is out the right wing.
c. +Z is down.
d. + azimuth is rotated about the Z axis and to the right.
e. + elevation is rotated about the Y axis (nose up).
f. + roll is rotated about the X axis and to the right (right wing down).
The Target Axis System is exactly the same as the two body axis systems described above and is also shown in Figure 8.

\[ \begin{align*}
X_B & \text{(NOSE)} \\
Y_B & \text{(RIGHT WING)} \\
Z_B & \text{(DOWN)}
\end{align*} \]

Figure 8. Body Axis System.

FLIGHT/VISUAL FUNCTIONS.

The functional tasks of the Flight/Visual Interface Module are now described in detail.

The nine Flight direction cosines that define the body to earth transfer matrix are computed using a heading correction to compensate for the curvature of the earth. This is needed because the Visual Coordinate System is rectangular, while the earth is obviously spherical. The derivation is shown below:

The heading angle PSI is computed from flight inputs: (1) sine and cosine of heading; (2) angular heading bias resulting from the convergence of longitude as the aircraft nears the north or south poles.

\[
\text{PSI} = \arctan\left(\frac{\text{AFSPSI}}{\text{AFCPSI}}\right) + \text{ANGBIAS}
\]

\[
\text{CPSI} = \cos(\text{PSI})
\]

\[
\text{SPSI} = \sin(\text{PSI})
\]

Where:

- \text{PSI} is aircraft true heading in degrees
- \text{AFSPSI}, \text{AFCPSI} are sine and cosine of aircraft heading without angular heading bias
- \text{ANGBIAS} is aircraft angular heading bias in degrees
- \text{CPSI} is cosine of PSI
- \text{SPSI} is the sine of PSI

19
Next, the nine direction cosines that describe the axis system in Figure 6 are derived using the standard Euler angles PSI, THETA, and PHI:

\[ \begin{align*}
AVBEFL_{11} &= AFCTHT \times CPSI \\
AVBEFL_{21} &= AFCTHT \times SPSI \\
AVBEFL_{31} &= -AFSTHT \\
AVBELF_{12} &= AFSPHI \times AFSTHT \times CPSI - AFCPHI \times SPSI \\
AVBELF_{22} &= AFCPHI \times CPSI + AFSPHI \times AFSTHT \times SPSI \\
AVBELF_{32} &= AFSPHI \times AFCTHT \\
AVBELF_{13} &= AFSPHI \times CPSI + AFCPHI \times AFSTHT \times SPSI \\
AVDELF_{23} &= AFCPHI \times AFSTHT \times SPSI - AFSPHI \times CPSI \\
AVDELF_{33} &= AFCPHI \times AFCTHT
\end{align*} \]

Where: AVBEFL11-33 are the nine direction cosines which describe the orientation of the Flight Earth Axis System in Figure 6 with respect to the body axis in Figure 8.

AFCTHT, AFSTHT are sine and cosine of aircraft pitch.
AFSPHI, AFCPHI are sine and cosine of aircraft roll.
CPSI, SPSI are sine and cosine of true aircraft heading.

Next, the nine Flight direction cosines are converted into CIG direction cosines. By inspection of Figure 6 and 7, it is obvious that the X and Y axis are interchanged and Z axis is opposite between the two systems. Hence, to convert from Flight to CIG, the X and Y matrix components are swapped and the Z matrix components are negated.

Convert Flight body to earth direction cosines to CIG body to earth direction cosines:

\[ \begin{align*}
AVBDER_{11} &= AVBEFL_{21} \\
AVBDER_{21} &= AVBEFL_{11} \\
AVBDER_{31} &= -AVBEFL_{31} \\
AVBDER_{12} &= AVBEFL_{22} \\
AVBDER_{22} &= AVBEFL_{12} \\
AVBDER_{32} &= -AVBEFL_{32} \\
AVBDER_{13} &= AVBEFL_{23} \\
AVBDER_{23} &= AVBEFL_{13} \\
AVBDER_{33} &= -AVBEFL_{33}
\end{align*} \]

Where: AVBDER11-AVBDER33 are CIG body to earth direction cosines.

Since the aircraft position and attitude parameters in flight are defined in the Aircraft Body Axis System, which has its origin at the aircraft center of gravity (C.G.), the visual software must define the pilot's eyepoint position in the Eyepoint Axis System to get the correct picture in the dome. The Aircraft Body Axis and the Eyepoint Axis System have their X, Y, Z axis aligned, but are offset in position by the relationship shown in Figure 9. The translation from the Aircraft Body Axis to the Eyepoint Axis System is derived below:
The eyepoint position is computed relative to Aircraft Body Axis System (see Figure 9). Note that the aircraft body axis shifts as the C.G. changes with aircraft weight changes, but the eyepoint position remains fixed relative to the aircraft fuselage. This point is illustrated in Figure 8.

\[ \text{AVDXBEP} = 9.2875 - \text{AFLX} \ldots \text{all coordinates and distances are measured in feet} \]
\[ \text{AVDYBEP} = 0.0 \]
\[ \text{AVDZBEP} = -2.125 \]

Where: \( \text{AVDXBEP}, \text{AVDYBEP}, \text{AVDZBEP} \) are eyepoint coordinates in Body Axis System
\( \text{AFLX} \) is aircraft X axis center of gravity (C.G.) position
9.2875 is C.G. to eyepoint position
0.0 is C.G. to eyepoint Y axis distance
-2.125 is C.G. to eyepoint Z axis distance

The eyepoint position relative to the Body Axis System is now converted into the CIG Earth Axis System using the body to earth transfer matrix derived above. These vector components represent the position offset between the C.G. and eyepoint in the Earth Axis System:

\[ \text{AVDXEEP} = \text{AVDXBEP} \times \text{AVBDER11} + \text{AVDYBEP} \times \text{AVBDER12} + \text{AVDZBEP} \times \text{AVBDER13} \]
\[ \text{AVDYEEP} = \text{AVDXBEP} \times \text{AVBDER21} + \text{AVDYBEP} \times \text{AVBDER22} + \text{AVDZBEP} \times \text{AVBDER23} \]
\[ \text{AVDZEEP} = \text{AVDXBEP} \times \text{AVBDER31} + \text{AVDYBEP} \times \text{AVBDER32} + \text{AVDZBEP} \times \text{AVBDER33} \]

Where: \( \text{AVDXEEP}, \text{AVDYEEP}, \text{AVDZEEP} \) are eyepoint vector components in the CIG Earth Axis System
\( \text{AVDXBEP}, \text{AVDYBEP}, \text{AVDZBEP} \) are eyepoint vector components in the Body Axis System
\( \text{AVBDER11-33} \) are eyepoint to earth axis transfer matrix components

To get the eyepoint position in the CIG Earth Axis System, we simply added the eyepoint position offset to the C.G. position, since both positions are now defined in the CIG Earth Axis System.

\[ \text{AVPXCEP} = \text{ANAGEX} + \text{AVDXEEP} \ldots \text{all positions are measured in feet} \]
\[ \text{AVPYCEP} = \text{ANAGEY} + \text{AVDYEEP} \]
\[ \text{AVPZCEP} = \text{ANAGEZ} + \text{AVDZEEP} \]

Where: \( \text{AVPXCEP}, \text{AVPYCEP}, \text{AVPZCEP} \) are eyepoint position, CIG Earth Axis System
\( \text{ANAGEX}, \text{ANAGEY}, \text{ANAGEZ} \) are aircraft C.G. position, CIG Earth Axis System
\( \text{AVDXEEP}, \text{AVDYEEP}, \text{AVDZEEP} \) are eyepoint position offsets from aircraft C.G., CIG Earth Axis System
NOTE: AIRCRAFT C.G. WILL SHIFT ALONG BODY AXIS AS AIRCRAFT FUEL, ORDNANCE WEIGHTS VARY, BUT EYEPONT POSITION REMAINS FIXED RELATIVE TO AIRCRAFT FUSELAGE.

LEGEND:

AFLX = VARIATION IN C.G. POSITION RELATIVE TO AIRCRAFT FUSELAGE DUE TO AIRCRAFT WEIGHT VARIATION
C.G. = AIRCRAFT CENTER OF GRAVITY; POINT ABOUT WHICH AIRCRAFT BODY AXIS ROTATES
EYEPONT = POSITION OF PILOT'S EYEPONT AND ORIGIN OF EYEPONT AXIS SYSTEM
HOOK PIVOT POINT = POSITION ON AIRCRAFT FUSELAGE ABOUT WHICH TAILHOOK PIVOTS (ONLY USED AS REFERENCE POINT, HERE)

Figure 9. Relationship Between Aircraft Body Axis and Eypoint Axis Systems.

Since the Body and Eypoint Axis Systems have their X, Y, and Z axis aligned (Figure 9), their attitude angles in the CIG Earth Axis System will be the same. These are derived below from the sine and cosine of yaw, pitch, and roll which are available from the flight software.
NAVTRAEOIPCEM 1H-335

AVWVPYW = ATAN(SPSSI/CPSI)
AVWVPPT = ATAN(AFSTHT/AFCTHT)
AVWVPRL = ATAN(AFSPHI/AFCPHI)

Where: AVWVPYW, AVWVPPT, AVWVPRL are CIG earth axis yaw, pitch, and roll in degrees
SPSSI, CPSI are sine and cosine of yaw
AFSTHT, AFCTHT are sine and cosine of pitch
AFSPHI, AFCPHI are sine and cosine of roll

Next, extrapolations are computed for viewpoint position and attitude. These are required for the CIG System, which runs at a 60 Hertz rate, while visual runs at 30 Hertz. Hence, visual software must extrapolate ahead 16.7 milliseconds. A simple two-point forward extrapolation is used for both the viewpoint positions and angles.

Newton's Law is used as follows:

\[ S' = S + \frac{V \times T}{2} \] (predicted distance for 1/2 field (16.7 ms))
\[ S' = S + \frac{(S - S_0)}{2} \]
\[ S' = 1.5S - .5S_0 \]

Where:
S' is predicted distance traveled for 16.7 ms
S0 is previous position last frame (33 ms ago)
V is average velocity between this frame and last frame
S is position this frame
T is 1/30 (33 ms) = SEL frame (update period)
T/2 is 1/60 (16.7 ms) = CIG frame (update period)

The X, Y, Z position extrapolations are then computed:

AVELXEEP = 1.5*AVPXCEP - .5*AVPXCEP1
AVELYEEP = 1.5*AVPYCEP - .5*AVPYCEP1
AVELZEEP = 1.5*AVPZCEP - .5*AVPZCEP1

Save previous frame's position:

AVPXCEP1 = AVPXCEP
AVPYCEP1 = AVPYCEP
AVPZCEP1 = AVPZCEP

Where: AVELXEEP, AVELYEEP, AVELZEEP are 16.7 ms position extrapolations
AVPXCEP, AVPYCEP, AVPZCEP are eyepoint position, CIG Earth axis, this frame values
AVPXCEP1, AVPYCEP1, AVPZCEP1 are eyepoint position, CIG Earth axis, last frame values
Similarly, yaw, pitch, and roll angle extrapolations are computed.

\[
\begin{align*}
AVWVPRLD &= 1.5 \times AVWVPRL - 0.5 \times AVWVPRL1 \\
AVWVPPTD &= 1.5 \times AVWVPPT - 0.5 \times AVWVPPT1 \\
AVWPWYWD &= 1.5 \times AVWVPYW - 0.5 \times AVWVPYW1
\end{align*}
\]

Save previous frame's attitude:

\[
\begin{align*}
AVWVPRL1 &= AVWVPRL \\
AVWVPPT1 &= AVWVPPT \\
AVWVPYW1 &= AVWVPYW
\end{align*}
\]

Where: AVWVPRLD, AVWVPPTD, AVWPWYWD are 16.7 ms attitude extrapolations.

AVWVPRL, AVWVPPT, AVWVPYW are eyepoint attitude, CIG Earth axis, this frame values.

AVWVPRL1, AVWVPPT1, AVWVPYW1 are eyepoint attitude, CIG Earth axis, last frame values.

Now, the target parameters are processed. The CIG database has a total of 14 targets that can be displayed on the VTRS dome. There are a maximum of 6 moving targets (moving models) and 8 fixed targets that can be contained in a CIG database. However, typically, only 2 moving models are used as targets for any given database. Therefore, the flight software is only designed to handle 10 (2 moving and 8 fixed) targets at once. The positions of these 10 targets are passed from visual to flight to permit target locations to be displayed by the TACAN in the cockpit.

This module also computes the target to earth transfer matrix which is used by the Window/Point of Interest Module. The Target Axis System is shown in Figure 10. Note that this axis system is identical to the Body Axis System. Hence, the derivation is exactly the same as the body to earth transfer matrix.

![Figure 10. Target Axis System.](image)
First, attitude angles for the selected target (as explained above, 14 targets available) are used to compute sines and cosines required for the target to earth matrix derivation:

\[
\begin{align*}
AVCMROLL &= AVWMRL(IJPRTGI) \\
AVCMPITC &= AVWMPT(IJPRTGI) \\
AVHEADCR &= AVWMYW(IJPRTGI)
\end{align*}
\]

Compute sines and cosines for yaw, pitch, roll angles:

\[
\begin{align*}
AVWSINR &= \sin(AVCMROLL) \\
AVWCOSR &= \cos(AVCMROLL) \\
AVWSINP &= \sin(AVCMPITC) \\
AVWCOSP &= \cos(AVCMPITC) \\
AVWSINY &= \sin(AVHEADCR) \\
AVWCOSY &= \cos(AVHEADCR)
\end{align*}
\]

Where: AVCMROLL, AVCMPITC, AVHEADCR are target yaw, pitch, and roll in degrees
AVWMRL, AVWMPT, AVWMYW are selected target yaw, pitch, and roll angles in degrees
IJPRTGI is the target index (1-14)
AVWSINR, AVWCOSR are sine and cosine of roll
AVWSINP, AVWCOSP are sine and cosine of pitch
AVWSINY, AVWCOSY are sine and cosine of yaw

Next, the nine standard direction cosines that are used to convert from the Target Axis System (Figure 10) to the Flight Earth Axis System (Figure 6) are derived:

\[
\begin{align*}
AVBEFL11 &= AVWCOSP*AVWCOSY \\
AVBEFL21 &= AVWCOSP*AVWSINY \\
AVBEFL31 &= -AVWSINP \\
AVBEFL12 &= AVWSINR*AVWSINP*AVWCOSY - AVWCOSR*AVWSINY \\
AVBEFL22 &= AVWCOSR*AVWCOSY + AVWSINR*AVWSINP*AVWSINY \\
AVBEFL32 &= AVWSINR*AVWCOSP \\
AVBEFL13 &= AVWSINR*AVWSINY + AVWCOSR*AVWSINP*AVWCOSY \\
AVBEFL23 &= AVWCOSR*AVWSINP*AVWSINY - AVWSINR*AVWCOSY \\
AVBEFL33 &= AVWCOSR*AVWCOSP
\end{align*}
\]

The Flight target to earth direction cosines are converted to CIG target to earth direction cosines:

\[
\begin{align*}
AVBETS11 &= AVBEFL21 \\
AVBETS21 &= AVBEFL11 \\
AVBETS31 &= -AVBEFL31 \\
AVBETS12 &= AVBEFL22 \\
AVBETS22 &= AVBEFL12 \\
AVBETS32 &= -AVBEFL32
\end{align*}
\]
NAVTRAIEQUIPCENH-335

AVBETS13 = AVBEFL23
AVBETS23 = AVBEFL13
AVBETS33 = -AVBEFL33

Where: AVBETS11-33 are target to earth transfer matrix components in CIG System
AVBEFL11-33 are target to earth transfer matrix components in Flight System

Flight logicals are passed to appropriate visual parameters. Separate visual parameters are used so that they may be manipulated independently of flight parameters. This feature protects flight logicals from being corrupted by changes in the visual software.

Finally, the last task is a test driver that allows the visual software and simulator to be controlled from the EOS. In this mode, all flight inputs are ignored and the operator must manually input viewpoint, target positions and attitudes. Flags can also be manually set. This is a convenient checkout mode which permits a controlled test, since it permits changing one parameter at a time (cause and effect are more easily identified).
SECTION III

WINDOW/POINT OF INTEREST MODULE

This module computes the vector from the observer to the selected target in Observer Axis System. This vector is called the point of interest (POI) vector. This vector is defined by a horizontal (azimuth) angle and vertical (elevation) angle. The sine and cosine of these angles are computed and passed to Visual/CIG Interface Module.

WINDOW/POINT OF INTEREST FUNCTIONAL DESCRIPTION.

This module defines "window" through which the observer must look in order to see the desired point of interest (POI) (Figure 11) in the Observer Axis System. The POI directs the observer's vision to the desired spot on the screen. It is usually the centroid of the target, but may be offset from the target centroid if desired. The angles defined in this routine are used by the CIG System to generate a target channel image and by the Target Image Projector Module to correctly point the target projector in the dome.

The functional tasks of the Window/Point of Interest Module are as follows:

First, the POI for the selected target is determined. Each of the six moving targets has its own respective "offset" which is defined in its own Target Axis System. An example of the offset used in the carrier model is shown in Figure 12. Here, it is more desirable to aim the target projector at the carrier runway centerline rather than the carrier centroid, because it's extremely important to see the runway clearly when making carrier landings.
Figure 12. Point of Interest (POI) Offset in Observer Axis.

Next, after selecting the appropriate offset above, the components of this offset vector must be converted from the Target Axis System (in which they are defined) to the Earth Axis System.

Convert offset vector into earth axis vector components:

\[
\begin{align*}
AVPXPOI &= AVPXFLT1 \times AVBTES11 + AVPYFLT1 \times AVBTES12 + AVPZFLT1 \times AVBTES13 \\
AVPYPOI &= AVPXFLT1 \times AVBTES21 + AVPYFLT1 \times AVBTES22 + AVPZFLT1 \times AVBTES23 \\
AVPZPOI &= AVPXFLT1 \times AVBTES31 + AVPYFLT1 \times AVBTES32 + AVPZFLT1 \times AVBTES33
\end{align*}
\]

Where: AVPXPOI, AVPYPOI, AVPZPOI are offset vector components in the Earth Axis System, in units of feet

AVPXFLT1, AVPYFLT1, AVPZFLT1 are offset vector components in the Target Axis System, in units of feet

AVBTES11–AVBTES33 represents the body to earth transfer matrix

The vector from the observer to the point of interest (POI vector) is computed in the Earth Axis System (Figure 13). This vector is used to get the window angles and slant range (X, Y, Z distance) to the target. The POI vector is derived below:
Figure 13. Point of Interest in CIG Earth Axis System.

Define the sum of the total X, Y, Z vector components about a point for Figure 13.

AVPXCEP + AVXPOIPG - AVPXPOI - AVPXPOIR = 0 (sum of vectors = 0)
AVPYCEP + AVYPOIPG - AVYPYPOI - AVYPYPOIR = 0 (sum of vectors = 0)
AVPZCEP + AVZPOIPG - AVPZPOI - AVPZPOIR = 0 (sum of vectors = 0)

Where: AVPXCEP, AVPYCEP, AVPZCEP are eyepoint position vector components, Earth Axis in units of feet
AVXPOIPG, AVYPOIPG, AVZPOIPG are eyepoint to POI vector components, Earth Axis in units of feet
AVPXPOI, AVPYPOI, AVPZPOI are offset vector components in CIG Earth Axis in units of feet
AVPXPOIR, AVPYPOIR, AVPZPOIR are target centroid position vector components, CIG Earth Axis in units of feet
Next solve for POI vector X, Y, Z components:

\[
\begin{align*}
AVXPOIPG &= AVPXPOI + AVPXPOIR - AVPXCEP \\
AVYPOIPG &= AVPYPOI + AVPYPOIR - AVPYCEP \\
AVZPOIPG &= AVPZPOI + AVPZPOIR - AVPZCEP
\end{align*}
\]

...all vector components are in units of feet

Now compute the slant range:

\[
AVEORNGE = \sqrt{AVXPOIPG^2 + AVYPOIPG^2 + AVZPOIPG^2}
\]

Where: AVEORNGE is eyepoint to point of interest vector magnitude in feet

Then, the sine and cosine of the azimuth angle (in the X-Y plane) and elevation (in the XY-Z plane) of POI vector are computed. These angles completely define the orientation of the POI vector in the observer axis and are shown in Figure 14. The derivation is as follows:

![Diagram showing point of interest angles in observer axis](image)

**Legend:**

AZIMUTH ANGLE = EYEPOINT TO POI YAW ANGLE (AVOBTGAZ)
ELEVATION ANGLE = EYEPOINT TO POI PITCH ANGLE (AVOBTGEL)

Figure 14. Point of Interest Angles in Observer Axis.

The POI unit vector is computed:

\[
\begin{align*}
AVXPIPGN &= AVXPOIPG / AVEORNGE \\
AVYPPIPGN &= AVYPOIPG / AVEORNGE \\
AVZPIPGN &= AVZPOIPG / AVEORNGE
\end{align*}
\]

Where: AVXPIPGN, AVYPPIPGN, AVZPIPGN are POI unit vectors, CIG Earth Axis System
AVXPOIPG, AVYPOIPG, AVZPOIPG are eyepoint to POI vector components, CIG Earth Axis, in feet
AVEORNGE is eyepoint to POI vector magnitude in feet
The POI unit vectors are now converted to the Eyepoint Axis System:

\[
\begin{align*}
\text{AVXPOIVO} &= \text{AVXPIGN} \times \text{AVBDER11} + \text{AVYPOIVO} \times \text{AVBDER12} + \text{AVZPOIVO} \times \text{AVBDER13} \\
\text{AVYPOIVO} &= \text{AVXPIGN} \times \text{AVBDER21} + \text{AVYPOIVO} \times \text{AVBDER22} + \text{AVZPOIVO} \times \text{AVBDER23} \\
\text{AVZPOIVO} &= \text{AVXPIGN} \times \text{AVBDER31} + \text{AVYPOIVO} \times \text{AVBDER32} + \text{AVZPOIVO} \times \text{AVBDER33}
\end{align*}
\]

Where: \( \text{AVXPOIVO}, \text{AVYPOIVO}, \text{AVZPOIVO} \) are POI unit vectors in the Eyepoint Axis System,
\( \text{AVBDER11} - \text{AVBDER33} \) are eyepoint to earth axis transfer matrix components,
\( \text{AVXPIGN}, \text{AVYPIGN}, \text{AVZPIGN} \) are POI unit vectors, CIG Earth Axis System, in feet.

Next, the sine and cosine of azimuth and pitch are computed in the Eyepoint Axis System:

\[
\begin{align*}
\text{AVRPOIVP} &= \sqrt{\text{AVXPOIVO} \times \text{AVXPOIVO} + \text{AVYPOIVO} \times \text{AVYPOIVO}} \\
\text{AVSHDPOI} &= \text{AVYPOIVO} / \text{AVRPOIVP} \\
\text{AVCHDPOI} &= \text{AVXPOIVO} / \text{AVRPOIVP} \\
\text{AVSPHPOI} &= -\text{AVZPOIVO} \\
\text{AVCPHPOI} &= \text{AVRPOIVP}
\end{align*}
\]

Where: \( \text{AVRPOIVP} \) is the length of the POI vector projected on the X-Y plane,
\( \text{AVSHDPOI}, \text{AVCHDPOI} \) are the sine and cosine of the POI azimuth angle, Eyepoint Axis,
\( \text{AVSPHPOI}, \text{AVCPHPOI} \) are the sine and cosine of the POI elevation angle, Eyepoint Axis,
\( \text{AVXPOIVO}, \text{AVYPOIVO}, \text{AVZPOIVO} \) are POI unit vectors in the Eyepoint Axis System.

Then, extrapolations for these sines and cosines are computed for the CIG System. These are simple two-point extrapolations identical to the ones used for the viewpoint parameters in Section II. Note that the angle rather than the sines and cosines are extrapolated, because it was determined experimentally that angles give a smoother and more consistent extrapolation.

First, the sines and cosines of azimuth and elevation are converted into angles:

\[
\begin{align*}
\text{AVOBTGAZ} &= \text{ATAN}(\text{AVYPOIVO} / \text{AVXPOIVO}) \\
\text{AVOBTGEL} &= \text{ATAN}(-\text{AVZPOIVO} / \text{AVRPOIVP})
\end{align*}
\]

Where: \( \text{AVOBTGAZ}, \text{AVOBTGEL} \) are eyepoint to POI yaw and pitch angles in degrees,
\( \text{AVXPOIVO}, \text{AVYPOIVO}, \text{AVZPOIVO} \) are X, Y, Z vector components of eyepoint to POI vector, eyepoint axis system,
\( \text{AVRPOIVP} \) is the length of the POI vector projected on the X-Y plane.
Next, a 16.7 millisecond forward extrapolation for the azimuth and elevation angles is computed for the CIG system:

AVOBTAZ2=1.5*AVOBTGAZ - .5*AVOBTAZI
AVOBTEL2=1.5*AVOBTTEL - .5*AVOBTEL1

Where: AVOBTAZ2, AVOBTEL2 are extrapolated eyepoint to POI yaw and pitch angles, eyepoint axis, in degrees
AVOBTGAZ, AVOBTGEL are current eyepoint to POI yaw and pitch angles, in degrees
AVOBTAZI, AVOBTTEL1 are last frames yaw and pitch angles

Save the previous frame azimuth and elevation angles:

AVOBTAZI=AVOBTGAZ
AVOBTEL1=AVOBTEL

Convert the azimuth and elevation values into sines and cosines:

AVSHDPOB=SIN(AVOBTAZ2)
AVCHDPOB=COS(AVOBTAZ2)
AVSPHPOB=SIN(AVOBTEL2)
AVCPHPOB=COS(AVOBTEL2)

Where: AVSHDPOB, AVCHDPOB are the sine and cosine of the POI azimuth angle, field 2 value
AVSPHPOB, AVCPHPOB are the sine and cosine of the POI elevation angle, field 2 value

Finally, a flag is tested to determine if the aircraft is on the ground (weight on wheels flag). If so, then the point of interest is set straight ahead. This feature provides the correct picture for carrier and airport landings.

The modules that interface this module and their parameters are:

Input Module: (1) ZM6SWO33 - viewpoint position, CIG Earth Axis
- selected POI position, CIG Earth Axis
- target to earth transfer matrix
- body to earth transfer matrix
(2) ZMVISINT - target alignment position, CIG Target axis system
Output Module: (1) ZM6SWCGI - sine, cosine of POI elevation and azimuth angles (in degrees), CIG Body axis system
- sine, cosine of POI elevation and azimuth angles extrapolated for field 2 updates, CIG Body axis system
(2) ZM6SWG37 - sine, cosine of POI elevation and azimuth angles, CIG Body axis system
- POI to target vector magnitude (slant range) in feet
SECTION IV
TARGET IMAGE PROJECTOR

This module inputs azimuth and elevation angles from the Point of Interest Module and converts them into Target Image Projector (TIP) Axis System. These angles are then used to drive the TIP servos in the dome. This puts the target in the proper position in the dome. The target channel field of view is also controlled by this module through the use of a zoom ratio parameter which drives a zoom lens. This permits optimization of target resolution in the target channel.

TARGET IMAGE PROJECTOR MODULE FUNCTIONAL DESCRIPTION.

The TIP is physically offset 42.5 inches above and 6 inches behind the pilot's eyepoint in the dome (Figure 15). Also, the TIP pivots on a 6-inch arm to the right of the pilot's eyepoint. These offsets require that the eyepoint angles be converted into the TIP Axis System (Figure 16). These angles are translated into output commands that are sent to the servos to position the projector.

Figure 15. Target Image Projector Position Relative to Eyepoint, Side View.
Referring to Figure 16, the screen intersection point is computed relative to the TIP pivot point using direction cosines, the dot product, the law of cosines, and the quadratic equation.

First the direction cosines of the POI vector in the Observer Axis System are computed. These direction cosines define the unit vector from the eyepoint to screen intersection point (EPSIPU):

\[
\begin{align*}
AVCHDPOI &= X / \sqrt{X^2 + Y^2 + Z^2} \\
AVCPHPOI &= \sqrt{X^2 + Y^2 + Z^2} \\
AVSHDPOI &= Y / \sqrt{X^2 + Y^2 + Z^2} \\
AVSPHPOI &= Z / \sqrt{X^2 + Y^2 + Z^2}
\end{align*}
\]

Where: X, Y, Z are the X, Y, Z components of the vector from eyepoint to screen intersection point in the Observer Axis System

AVCHDPOI, AVSHDPOI are the cosine and sine of the POI azimuth angle in the Observer Axis System

AVCPHPOI, AVSPHPOI are the cosine and sine of the POI elevation angle in the Observer Axis System

Let AVXPOIOB = X; AVYPOIOB = Y; AVZPOIOB = Z.
By substituting for X, Y, and Z, the direction cosines of the unit vector EPSIPU can be expressed:

\[
\begin{align*}
\text{AVXPOI\text{OB}} &= \text{AVCHDPOI} \times \text{AVCPHPOI} \quad (\text{X component of EPSIPU}) \\
\text{AVYPOI\text{OB}} &= \text{AVSHDPOI} \times \text{AVCPHPOI} \quad (\text{Y component of EPSIPU}) \\
\text{AVZPOI\text{OB}} &= -\text{AVSPHPOI} \quad (\text{Z component of EPSIPU})
\end{align*}
\]

Where: AVXPOI\text{OB}, AVYPOI\text{OB}, AVZPOI\text{OB} are the direction cosines for the POI vector in the Observer Axis
AVCHDPOI, AVSHDPOI are the sine and cosine of the POI azimuth angle in the Observer Axis
AVCPHPOI, AVSPHPOI are the sine and cosine of the POI elevation angle in the Observer Axis

Next, the dot product between the dome center to observer vector (CEP) and the unit vector of observer to POI (EPSIPU) vector is computed (see Figure 16):

\[
\begin{align*}
\text{AVCPC20P} &= \text{CEP}.\text{EPSIPU} = A1*A2 + C1*C2 \\
&= \text{AVCPC2OP} = 6*\text{AVXPOI\text{OB}} + 24.5*\text{AVZPOI\text{OB}} \\
&= \text{AVCPC2OP} = 6*\text{AVXPOI\text{OB}} + 24.5*\text{AVZPOI\text{OB}}
\end{align*}
\]

Where: AVCPC2OP is the DOT product of CEP and EPSIP unit vector
CEP is the dome center to observer vector
EPSIPU is the unit vector of EPSIP
A1, A2 are X components of CEP, EPSIPU vectors
C1, C2 are Z components of CEP, EPSIPU vectors
AVXPOI\text{OB}, AVYPOI\text{OB} are the direction cosines for the POI vector in the Observer Axis
6.0 is dome center to eyepoint distance along the dome X axis, in inches
24.5 is distance from dome center to eyepoint along the dome Z axis, in inches

Then, the distance from the observer to the intersection point on the screen (EPSIP) is computed using the law of cosines (L.C.) and the quadratic equation (Q.E.):

\[
\begin{align*}
\text{CEP*CEP} &= \text{EPSIP*EPSIP} + \text{CSIP*CSIP} - 2*\text{EPSIP*EPSIP*COSA} \\
14400 &= \text{EPSIP*EPSIP} + 636.25 - 2*\text{EPSIP*CSIP*COSA} \\
\text{EPSIP*EPSIP} - 2*\text{CSIP*COSA*EPSIP} - 13763.25 &= 0 \\
\text{EPSIP} &= (2*\text{CSIP*COSA} + \sqrt{(2*\text{CSIP*COSA})^2 - 4*(-13763.75) \}/2} \\
\text{EPSIP} &= \text{CSIP*COSA} + \sqrt{(\text{CSIP*COSA} + \text{CSIP*COSA}) + 13763.75}
\end{align*}
\]

(arrange for substitution)
Substituting AVDSOBPI for EPSIP and AVPC2OP for CSIP*COSA, the eyepoint to screen distance is determined:

\[
AVDSOBPI = AVPC2OP + \sqrt{(AVCOC2OP \cdot AVCOC2OP + 13763.75)}
\]

Where:
- AVDSOBPI = EPSIP = eyepoint to screen intersection point distance
- AVPC2OP = CSIP*COSA = DOT product of the vector CEP and unit vector EPSIP
- 14400 = 120 * 120 = dome radius distance squared (units in square inches)
- 636.25 = 6 * 6 + 24.5 * 24.5 = CEP distance squared (units in square inches)
- 13763.75 = 14400 - 636.25

Now the screen intersection point is defined in terms of the TIP Axis System (located at the TIP pivot point) X, Y, Z components:

\[
\begin{align*}
AVXPOITP &= 6.0 + AVDSOBPI \cdot AVXPO1OB \\
AVYPOITP &= AVDSOBPI \cdot AVYPO1OB \\
AVZPOITP &= 42.5 + AVDSOBPI \cdot AVZPO1OB
\end{align*}
\]

Where:
- AVXPOITP, AVYPOITP, AVZPOITP are the screen intersection point coordinates in the TIP Axis System
- 6.0 = X component of TIP distance from the eyepoint (units in inches)
- 42.5 = the Z component of the TIP distance from the eyepoint (inches)
- AVDSOBPI*AVXPO1OB is X observer to POI on screen distance
- AVDSOBPI*AVYPO1OB is Y observer to POI on screen distance
- AVDSOBPI*AVZPO1OB is Z observer to POI on screen distance

Next, the vector AVRSCNTP (Figure 17) is computed. This is used in deriving the TIP azimuth angle:

\[
\begin{align*}
AVRSCNPI &= TSIPXY \cdot TSIPXY \\
AVRSCNPI &= AVXPOITP \cdot AVXPOITP + AVYTOITP \cdot AVYTOITP \\
AVRSCNTP &= \sqrt{AVRSCNPI - 36.0}
\end{align*}
\]

Where:
- AVRSCNPI is TSIPXY squared
- AVRSCNTP is TIP projection point to (AVXPOITP, AVYPOITP)
- AVXPOITP, AVYPOITP, AVZPOITP X, Y components of screen intersection point projection on TIP X-Y plane
- TSIPXY is projection of vector TSIP on TIP X-Y plane
- 36.0 = 6 * 6 = TIP pivot arm distance squared
Referring to Figure 18, the TIP azimuth angle in the TIP Axis System is derived using trigonometric relations as follows:

\[
\begin{align*}
\text{TIP azimuth} &= H = A + B - 90. \\
\sin(H) &= \sin(A+B-90.) = -\cos(A+B) = -\cos A \cos B + \sin A \sin B \\
\cos(H) &= \cos(A+B-90.) = \sin(A+B) = \sin A \cos B + \cos A \sin B \\
\sin(A) &= \frac{\text{AVYPOITP}}{\sqrt{\text{AVRSCNPI}}} \\
\cos(A) &= \frac{\text{AVXPOITP}}{\sqrt{\text{AVRSCNPT}}} \\
\sin(B) &= \frac{\text{AVRSCNTP}}{\sqrt{\text{AVRSCNPT}}} \\
\cos(B) &= \frac{6.}{\sqrt{\text{AVRSCNPI}}} \\
\sin(H) &= \frac{\text{AVYPOITP} \times 6.}{\text{AVRSCNPI} + \text{AVXPOITP} \times \text{AVRSCNTP} / \text{AVRSCNPI}} \\
\cos(H) &= \frac{\text{AVYPOITP} \times 6.}{\text{AVRSCNPI} - \text{AVXPOITP} \times \text{AVRSCNTP} / \text{AVRSCNPI}} \\
\cos(H) &= \text{AVCTPHDG} = \text{AVRSCNTP} \times \text{AVXPOITP} + \text{AVYPOITP} \times 6. \\
\sin(H) &= \text{AVSTPHDG} = \text{AVRSCNTP} \times \text{AVYPOITP} - \text{AVXPOITP} \times 6. \\
H &= \text{ATAN}(\text{AVSTPHDG} / \text{AVCTPHDG})
\end{align*}
\]
NAVTRAEEICPCEN IH-335

Where:  
A = angle between TIP X axis and vector TSIPXY  
B = angle between vector TIPXY and TIP pivot arm  
H = TIP azimuth angle  
AVRSCNPI = TSIPXY squared  
AVRSCNTP = TIP projection point to (AVXPOITP, AVYPOITP)  
AVXPOITP, AVYPOITP are X, Y components of vector TSIPXY  
6. = TIP pivot arm length (inches)

The TIP evaluation angle in the TIP Axis System is easily computed:

\[ \text{AVTIPPTH} = \text{ATAN}(-\text{AVZPOITP}/\text{AVRSCNTP}) \]

where:  
AVTIPPTH is the TIP elevation angle in degrees  
AVZPOITP is the Z component of vector TSIP  
AVRSCNTP is TIP projection point to (AVXPOITP, AVYPOITP)

The TIP zoom ratio is computed as a function of the slant range from the zoom schedule in "WTIPZOOM."

The TIP roll is computed from TIP heading and pitch commands. (It removes roll induced by TIP azimuth and elevation.)
TIP position is checked against excursion limits and drive commands limited and iris closed to protect hardware from hitting stops in dome.

a. "TIPSAME" determines if TIP was in stop area last frame.
   
   (1) Used to determine if iris should stay close.
   
   (2) Used in setting proper commanded headings based on current and previous commanded headings.

TIP elevation and azimuth commands are limited to 200 deg/sec by "LIMITEL" and "LIMITAZ" commands.

a. TIP velocity and acceleration for AZ and EL are computed and then commanded positions are limited for 200 deg/sec.

   (1) TIP, azimuth, elevation, roll commands are delayed one frame (33 millisec) to sync TIP hardware with CIG background video.
   
   (2) TIP, azimuth, elevation, roll, iris commands are scaled to drive hardware.

Extrapolations (field 2 updates) for TIP elevation, azimuth, roll angles are computed for the CIG System. A simple two-point extrapolation (see Section II) is used.

The extrapolated yaw, pitch, and roll angles are computed as follows:

\[
\begin{align*}
AVWTRL2 &= 1.5*AVTIPROL - .5*AVWTRL11 \\
AVWTPT2 &= 1.5*AVTIPPTH - .5*AVWTPT11 \\
AVWTYW2 &= 1.5*AVTIPHDG - .5*AVWTYW11 \\
\end{align*}
\]

Where: \(AVWTYW2, AVWTPT2, AVWTRL2\) are extrapolated TIP yaw, pitch, roll angles in TIP axis system
\(AVTIPHDG, AVTIPPTH, AVTIPROL\) are current TIP yaw, pitch, roll angles in TIP axis system
\(AVWTYW11, AVWTPT11, AVWTRL11\) are last frame's TIP yaw, pitch, roll angles in TIP axis system

Save last frame's TIP yaw, pitch, roll angles:

\[
\begin{align*}
AVWTRL11 &= AVTIPROL \\
AVWTPT11 &= AVTIPPTH \\
AVWTYW11 &= AVTIPHDG \\
\end{align*}
\]

The modules that interface this routine and their parameters are:
Input Module: ZM6SW#36 - sine, cosine of elevation, azimuth, POI angles, and CIG Body
WTIPZOOM - TIP zoom ratio (ratio varies from 1 to 0.1 to control target channel field of view)
LIMITAZ - commanded TIP, azimuth angle, TIP Axis
LIMITEL - commanded TIP elevation angle, TIP Axis
TIPSAME - TIP same side (T/F)

Output Module: ZM6SWCGI - elevation, azimuth, roll, zoom, TIP Axis
- field 2 updates, TIP Axis
This module looks at the database loaded and then activates the appropriate subroutines to drive the models that exist in that environment. Each database has its own unique models that must be driven by the weapons visual software. The user may utilize a variety of moving models for each database, but must know which models are available for each respective database. A default mode exists for each database so that the unknowledgeable user may perform limited operations with the system.

MOVING MODEL CONTROL MODULE FUNCTIONAL DESCRIPTION.

This module is the “brains” of the visual weapon system and must decide what subroutines are permitted to run. This decision is based on which database is loaded and which user request flags are set. These two inputs are explained below:

First, the environment is checked to determine which options (Table 1) the user is permitted to exercise. For the DAYFORRESTAL and FLOLS Meridian, there are no options. The drivers for these two databases come up automatically. The other databases, however, have the options listed in Table 1. In the default mode, the tank and truck drivers are activated automatically for the Gunnery Range, Twin Towns, and River Valley. The six basic environment types are listed below.

a. DAYFORRESTAL (aircraft carrier)
b. Standard Meridian (has TA4J for formation flight)
c. FLOLS Meridian (has carrier FLOLS on runway)
d. Gunnery range (has tank, truck, rockets, bombs, bullets)
e. Twin towns (has tank, truck, SAM, FLAK, rockets, bombs, bullets)
f. River valley (has tank, truck, FLAK, rockets, bombs, bullets)
TABLE 1. MOVING MODEL OPTIONS FOR DATABASE ENVIRONMENTS

<table>
<thead>
<tr>
<th>DATABASES</th>
<th>MOVING MODEL DRIVERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVA</td>
<td>FLOLS</td>
</tr>
<tr>
<td>DAYFORRESTAL</td>
<td>X</td>
</tr>
<tr>
<td>STANDARD MERIDIAN</td>
<td></td>
</tr>
<tr>
<td>FLOLS MERIDIAN</td>
<td></td>
</tr>
<tr>
<td>GUNNERY RANGE</td>
<td></td>
</tr>
<tr>
<td>TWIN TOWNS</td>
<td></td>
</tr>
<tr>
<td>RIVER VALLEY</td>
<td></td>
</tr>
</tbody>
</table>

LEGEND:
CVA = US FORRESTAL AIRCRAFT CARRIER
FLOLS = FRENSAL LENS OPTICAL LANDING SYSTEM
TA4J = TA4J SKYHAWK AIRCRAFT
ASM = AIR-TO-SURFACE MISSILE (2.75 INCH ROCKET)
TANK = RUSSIAN TANK
TRUCK = FIVE TRUCK CONVOY
SAM = SURFACE-TO-AIR MISSILE (SA-3, SA-6)
FLAK = ANTI-AIRCRAFT FLAK
GEN = GENERAL PURPOSE MOVING MODEL DRIVER

NOTE: VARIATIONS FOR EACH OF THESE DATABASES ARE AVAILABLE

Valid user requests for moving model drivers results in the activation of one or more of the nine weapon subroutine drivers illustrated in Table 1. For example, the user may desire to use only the tank driver in Twin Towns even though six drivers are available. To perform this task the user must set the tank flag in Table 2 to true and set the other applicable flags (refer to twin towns in Table 1) to false. The nine subroutine drivers and the conditions necessary to activate them are summarized below:

TABLE 2. MOVING MODEL CONTROL FLAGS

<table>
<thead>
<tr>
<th>DATABASES</th>
<th>CONTROL FLAGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVA</td>
<td>NA</td>
</tr>
<tr>
<td>FLOLS</td>
<td>NA</td>
</tr>
<tr>
<td>TA4J</td>
<td>EAFORFLT</td>
</tr>
<tr>
<td>ASM</td>
<td>EJBROB</td>
</tr>
<tr>
<td>TANK</td>
<td>LVTANK</td>
</tr>
<tr>
<td>TRUCK</td>
<td>LVTRUCK</td>
</tr>
<tr>
<td>SAM</td>
<td>EJBSAM</td>
</tr>
<tr>
<td>FLAK</td>
<td>EJBFALK</td>
</tr>
<tr>
<td>GEN</td>
<td>EGENERAL</td>
</tr>
</tbody>
</table>

1 NOTE: AUTOMATICALLY ACTIVATED WHEN DAYFORRESTAL IS LOADED
2 NOTE: AUTOMATICALLY ACTIVATED WHEN FLOLS MERIDIAN IS LOADED
a. WCARRIER drives a model of the USS FORRESTAL aircraft carrier. This subroutine is automatically activated whenever the DAYFORRESTAL database is loaded.

b. WFLOLS drives a model of the Fresnal Lens Optical Landing System (FLOLS). This subroutine is automatically activated whenever the DAYFORRESTAL or FLOLS Meridian database is loaded.

c. WTAAJ drives a model of the TA4J "SKYHAWK" aircraft. This subroutine is activated whenever the formation flight flag is set and standard Meridian database is loaded.

d. WROCKETS drives models of two Air-To-Surface (ASM) 2.75 inch rockets. This subroutine is activated whenever the ASM flag is set and Gunnery Range, Twin Towns, or River Valley databases are loaded.

e. WSAM drives a model of a Russian Surface-To-Air (SAM) missile (SA-3 or SA-6). This subroutine is activated whenever the SAM launch flag is set and the Twin Towns database is loaded.

f. WFLAK drives a model of anti-aircraft FLAK. This subroutine is activated whenever the FLAK flag is set and Twin Towns or River Valley databases are loaded.

g. WTANK drives a model of a Russian tank on each respective road system for Gunnery Range, Twin Towns, and River Valley. This routine is activated whenever the tank flag is set and one of the above three databases is loaded.

h. WTRUCK drives a model of a convoy of five trucks on each respective road system for Gunnery Range, Twin Towns, and River Valley. The trucks are driven on different roads than the tanks. This subroutine is activated whenever the truck flag is set and one of the above three databases is loaded.

i. WGENERAL is a general purpose driver available to the user for special applications when one of the above drivers fails to meet his requirements. This routine is activated whenever the general purpose flag is set and is independent of the database load.

The parameter relationship between input modules and the output module that ZM6SMOVE drives are as follows:

Input Module: ZM6SWCG1 - active environment set (tells what database is loaded)
ZM6SW@33 - load guns, bombs, rockets, flags set
Output Module:

WCARRIER - is called if carrier database loaded
WFLAK - called if FLAK launched and weapon* database is loaded
WFLOLS - called if carrier or FLOLS Meridian database is loaded
WGENERAL - is called if EGENERAL = true
WROCKETS - called if ASM launched and weapon database is loaded
WSAM - called if SAM launched and twin town database is loaded
WTA4J - called if EAFORFLT true and standard Meridian loaded
WTANK - called if weapon database loaded and LVTANK = true
WTRUCK - called if weapon database loaded and LVTRUCK = true

* Note: Weapon databases include Gunnery Range, Twin Towns, and River Valley
SECTION VI

WEAPON SCORING

This module scores weapon deliveries of bombs, bullets, and rockets shot from the T2-C aircraft. Bombs are scored as ground miss distances. Rockets and bullets are scored as closest point of approach (CPA) above the ground. Weapon visual effect flags are also set by this module. These include hit flags for the tank and trucks, and impact flags for bombs, rockets, and bullets.

WEAPON SCORING MODULE FUNCTIONAL DESCRIPTION.

This module scores bombs by computing the ground impact distance from the selected target. This is illustrated in Figure 19. Note that all scoring vector magnitudes are measured in feet. The following is a summary of this derivation:

As noted in Figure 19, the following parameters are passed to this module:

- a. Weapons release vector (PA)
- b. Target vector (PT)
- c. Weapon impact vector (B')

The ground miss distance (Q), then is readily computed.

Express vector relationship of B', Q, PT:

\[ B' - Q - PT = 0 \]  
...sum of vectors around a point = 0

Solve for target to impact vector Q:

\[ Q = B' - PT \]  
...solve for Q

AJTXIMP = AJPXIMP - AJTXPOS  
AJTYIMP = A JPYIMP - AJTYPOS  
AJTZIMP = AJPZIMP - AJTZPOS

Where:  
AJTXIMP, AJTYIMP, AJTZIMP are target to impact vector (Q) components, CIG Earth Axis
AJPXIMP, AJPYIMP, AJPZIMP are weapon impact vector (B') components, CIG Earth Axis
AJTXPOS, AJTYPOS, AJTZPOS are target vector (PT) components, CIG Earth Axis
Solve for magnitude of vector $Q$:

$$AJTLEGQ^2 = \sqrt{AJTXIMP^2 + AJTYIMP^2 + AJTZIMP^2}$$

Where: $AJTLEGQ^2 = Q =$ magnitude of target to impact vector

This module scores rockets and bullets by computing the closest point of approach (CPA) above ground. As shown in Figure 20, the CPA is the closest point the trajectory passes by the target. This is especially important for rockets and bullets, which tend to impact at low elevation angles (10-30 degrees). Hence, it is possible for a bullet to pass through a target, yet impact the ground hundreds of feet from the target.
NAVTRAEEQUIPCEN IH-335

Using the vectors PA, PT, B', PAT, Q, AND Y', the closest point of approach vector X' can be derived as follows.

Express vector relationship of PA, PAT, PT:

\[ \text{PA} + \text{PAT} - \text{PT} = 0 \]

...sum of vectors around point = 0

Solve for weapon to target vector PAT:

\[ \text{PAT} = \text{PT} - \text{PA} \]

...solve for PAT

\[ \text{AJATXPOS} = \text{AJTXPOS} - \text{AJAXPOS} \]

...define PAT, PT, PA in X, Y, Z vector components

\[ \text{AJATZPOS} = \text{AJTZPOS} - \text{AJAYPOS} \]

Where: AJATXPOS, AJATYPOS, AJATZPOS are weapon (aircraft) to target X, Y, Z vector components, CIG Earth Axis

AJTXPOS, AJTYPOS, AJTZPOS are target X, Y, Z position vector components, CIG Earth Axis

AJAXPOS, AJAYPOS, AJAZPOS are weapon (aircraft) X, Y, Z position vector components, CIG Earth Axis system

Solve for magnitude of PAT:

\[ \text{AJATRPOS} = \sqrt{\text{AJATXPOS}^2 + \text{AJATYPOS}^2 + \text{AJATZPOS}^2} \]

Where: AJATRPOS=PAT=magnitude of weapon to target vector

Express vector relationship of PA, B, B':

\[ \text{PA} + \text{B} - \text{B'} = 0 \]

Solve for weapon release point to impact vector B:

\[ \text{B} = \text{B'} - \text{PA} \]

...solve for B

\[ \text{AJAXIMP} = \text{AJPXIMP} - \text{AJAXPOS} \]

...define B, B', PA in X, Y, Z vector components

\[ \text{AJAYIMP} = \text{AJPYIMP} - \text{AJAYPOS} \]

\[ \text{AJAZIMP} = \text{AJPZIMP} - \text{AJAZPOS} \]

Where: AJAXIMP, AJAYIMP, AJAZIMP are X, Y, Z components of the weapon to impact vector

AJPXIMP, AJPYIMP, AJPZIMP are X, Y, Z components of impact position, CIG Earth Axis

AJAXPOS, AJAYPOS, AJAZPOS are X, Y, Z components of aircraft position, CIG Earth Axis

Solve for magnitude of vector B:

\[ \text{AJTLEGQ1} = \sqrt{\text{AJAXIMP}^2 + \text{AJAYIMP}^2 + \text{AJAZIMP}^2} \]
Where: \(AJTLEGQ1 = \text{magnitude of weapon to impact vector}\)

Solve for \(\cos(A)\) in Figure 20, using the DOT product relationship:

\[
Q \cdot B = [\text{ABS}(Q) \times \text{ABS}(B)] \times \cos(A) \quad \text{...DOT product definition}
\]

\[
AJTLEG = [AJTLEGQ2 \times AJTLEGQ1] \times AJCOSYQ \quad \text{...substitute VIS PAR's.}
\]

\[
AJTLEG = [AJTLEGQ2 \times AJTLEGQ1] \times AJCOSYQ \quad \text{...can be redefined in}
\]

\[
X, Y, Z \text{ components}
\]

\[
AJTLEG = [AJTLEGQ2 \times AJTLEGQ1] \times AJCOSYQ \quad \text{...(substituting and}
\]

\[
solving for AJCOSYQ)
\]

Where: \(Q \cdot B\) is DOT product of vectors \(Q\) and \(B\)

\(\text{ABS}(Q) \times \text{ABS}(B)\) is product of absolute values of \(Q\) and \(B\)

\(\cos(A)\) is cosine of angle \(A\) (see Figure 20)

\(AJTLEGQ2, AJTLEGQ1\) are vector magnitudes of \(Q\) and \(B\)

\(AJCOSYQ = \cos(A)\) is visual parameter for \(AJCOSYQ\)

\(AJTXIMP, AJTYIMP, AJTZIMP\) are target to impact vector

\(AJAXIMP, AJAYIMP, AJAZIMP\) are weapon to impact vector

\(AJCOSYQ = \cos(A)\) (redefined in \(X, Y, Z\) components)

\(CIG\) Earth Axis

\(AJAXIMP, AJAYIMP, AJAZIMP\) are target to impact vector

\(AJAXIMP, AJAYIMP, AJAZIMP\) are weapon to impact vector

\(AJTLEGQ1\) is the magnitude of vector \(B\)

Compute the unit vector components of \(B\) (Figure 20):

\[
AJUAXIMP = AJAXIMP / AJTLEGQ1 \quad \text{...divide X, Y, Z components of } B
\]

\[
AJUAYIMP = AJAYIMP / AJTLEGQ1 \quad \text{...by absolute value of } B
\]

\[
AJUAZIMP = AJAZIMP / AJTLEGQ1
\]

Where: \(AJUAXIMP, AJUAYIMP, AJUAZIMP\) are unit vector components

\(AJAXIMP, AJAYIMP, AJAZIMP\) are weapon to impact vector

\(AJAXIMP, AJAYIMP, AJAZIMP\) are target to impact vector

\(AJTLEGQ1\) is the magnitude of vector \(B\)

Now, the vector \(Y'\) (see Figure 20) can be derived:

\[
Y' = Q \times \cos(A) \quad \text{...solving for } Y'
\]

\[
AJYLEG = AJTLEGQ2 \times AJCOSYQ \quad \text{...substituting for VIS PAR's}
\]

Solving for \(X, Y, Z\) components of \(Y'\):

\[
AJYLEGX = AJYLEG \times AJUAXIMP \quad \text{...since vectors } B \text{ and } Y' \text{ have}
\]

\[same \text{ orientation, the unit vectors of } B \text{ define the unit vectors of } Y'
\]

\[
AJYLEGY = AJYLEG \times AJUAYIMP
\]

\[
AJYLEGZ = AJYLEG \times AJUAZIMP
\]

49
Where: \( AJYLEGX, AJYLEGY, AJYLEGZ \) are vector components of \( Y' \)
\( AJYLEG \) is the magnitude of vector \( Y' \)
\( AJUAXIMP, AJUAYIMP, AJUAZIMP \) are unit vectors of \( B \)
...direction of \( Y' \)

Finally, the closest point of approach vector \( X' \) (Figure 20) can be computed:

\[
Q - Y' - X' = 0
\]

...sum of vectors about point = 0

\[
X' = Q - Y'
\]
...solve for \( X' \) (CPA)

\[
AJXCPA = AJTXIMP - AJYLEGX
\]
...substitute VIS PAR's

\[
AJYCPA = AJTYIMP - AJYLEGY
\]

\[
AJZCPA = AJTZIMP - AJZLEGZ
\]

Where: \( AJXCPA, AJYCPA, AJZCPA \) are vector components of \( X' \),
CIG Earth Axis System

\( AJTXIMP, AJTYIMP, AJTZIMP \) are vector components of \( Q \),
CIG Earth Axis System

\( AJYLEGX, AJYLEGY, AJYLEGZ \) are vector components of \( Y' \),
CIG Earth Axis System

Solve for magnitude of \( X' \) which is the closest point of approach (CPA) that bullet/rocket passes by the target:

\[
AJCPA = \sqrt{AJXCPA^2 + AJYCPA^2 + AJZCPA^2}
\]
...sum of vectors \( X, Y, Z \)

Where: \( AJCPA \) is the magnitude of the CPA vector \( X' \)
\( AJXCPA, AJYCPA, AJZCPA \) are the vector components of \( X' \),
CIG Earth Axis System

Next, since bullets and rockets do not significantly bore holes below the ground, the \( Z \) (vertical) component of the CPA vector must be checked for negative values. If the \( Z \) component is negative, then the ground miss distance \( Q \) is used instead of the CPA for scoring. Figure 21 illustrates this point. The mathematical derivation is provided below:

\[
X'(Z) = AJZCPA
\]

If \( AJZCPA < 0 \), then CPA miss distance is invalid and the value of \( Q \) is used.
Otherwise, CPA miss distance is valid so the value of $X'$ can be used.

Figure 21. Vector Diagram for CPA Values Below the Ground.

Target hits are scored when the kill radius of the target and the weapon kill radius overlap. The kill radius of fixed targets are specified by the CIG System when a database is loaded. The kill radius of moving models are specified by the weapon visual software. Typically, moving models such as tanks and trucks have a small kill radius (30 feet), while fixed targets have a larger kill radius (100 feet, etc.). The kill radius of the weapons is listed below:

- Bullets: 0.0 feet (must hit target directly).
- Rockets: 10.0 feet
- Bombs: 50-300 feet

This module also sets weapon control flags. These flags control the visual weapon effects and also synchronize weapon trajectories with the flight software. These flags are summarized:

- Tank Explode Flag ... IVWMHT set to 3
- Truck Explode Flag ... IVWMHT set to 4
- Fixed Target Delete Flag ... IVWFHT set to 1, 2, 3, 4, 5, 6, 7, 8 to delete respective target
- Bullet Impact Flag ... KVGIF set to 2 for 2 bullet hits
- Bomb Impact Flag ... KVGIF set to 2 for 2 bomb hits
- Rockets Impact Flag ... LDWL set to true for explode
- Scoring Complete Flag ... EJSCOREF set to true when done
- Target Killed ... EJTGD set true when target hit

51
This module interfaces weapon visual parameters and CIG parameters on the CIG side (Figure 1). All weapon visual parameters required to drive the CIG System are converted into PDP-11/55T format and put into datapool for output to the PDP. All CIG parameters required for weapon visual software are obtained from datapool and assigned to the appropriate weapon visual parameters. These are primarily database parameters and are required to control the air-to-ground software.

VISUAL/CIG INTERFACE MODULE FUNCTIONAL DESCRIPTION.

All interface tasks done by this module are performed on parameters that reside in datapool. The actual data transfer between the DEC and SEL Systems is performed by another subroutine called WDECSEL. This subroutine merely takes these datapool parameters which have been set up by the Visual/CIG Interface Module and reads or writes to the CIG System. The parameters in Tables 3 and 4 are real-time parameters and are transferred at a 30 Hertz rate. The parameters in Table 5 are database initialization parameters and are only transferred when a new database is loaded. These tables will be discussed below.

There are three basic interface tasks performed by this module:

a. The first is the transfer of all weapon visual parameters required to drive the CIG Visual System. Note that all position and attitude coordinates are in CIG Earth Axis system (Figure 7) and are specified in feet and degrees respectively. These parameters are shown in Table 3 and are summarized below:

1. Viewpoint position field 1 and field 2 (X, Y, Z positions)
2. Viewpoint attitude field 1 and field 2 (yaw, pitch, roll)
3. Moving models 1-6 position field 1 and field 2
4. Moving models 1-6 attitude field 1 and field 2
5. FLOLS glide slope, roll angle, brightness, meatball position
6. TIP roll, pitch, yaw angles field 1 and field 2
7. TIP zoom ratio (values: .15-1.0)
8. Window attitude: sine, cosine of yaw, pitch angles for field 1 and field 2
(9) Bomb impact locations (X and Y positions for each impact)
(10) Gun impact locations (X and Y positions for each impact)
(11) Target control display flag (specifies what target you see in the target channel)
(12) Background control display flag (specifies what you see in the background channel).
(13) Weapons control flags:
   (a) FLOLS on flag (true=on)
   (b) FLOLS wave off flag (true=wave off)
   (c) FLOLS auxiliary wave off flag (true=wave off)
   (d) FLOLS cut light flag (true=cut)
   (e) Viewpoint crash flag (true=viewpoint crashed)
   (f) Scene freeze flag (true=scene frozen)
   (g) Bomb impact flag (0=no impact, 1=1 impact, 2=2 impacts)
   (h) Gun impact flag (0=no impact, 1=1 impact, 2=2 impacts)
   (i) ASM launch flag (true=launch rockets)
   (j) SAM launch flag (true=launch SAM)
   (k) AA active flag (true=flash anti-aircraft pad)
   (l) AA explode flag (true=explose FLAK)
   (m) Close hit flag (true=make sky turn transparent yellow)
   (n) Own hit flag (true=make sky turn solid yellow)
   (o) IC complete (true=initial condition complete)
   (p) New command (true=new CIG command is requested)
   (q) ASM explode (true=make rockets explode)
   (r) Heads up display select (0, 1, 2 for display 0, 1, 2)
### TABLE 3. SEL TO DEC INTERFACE
**(A) FLOATING POINT PARAMETERS**

**32 BIT WORDS**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>SYMBOL</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIEWPOINT, FIELD 1</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>Y</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>3</td>
</tr>
<tr>
<td>ROLL</td>
<td>ADWPR1</td>
<td>4</td>
</tr>
<tr>
<td>PITCH</td>
<td>ADWVT</td>
<td>5</td>
</tr>
<tr>
<td>YAW</td>
<td>ADWVPY</td>
<td>6</td>
</tr>
<tr>
<td>VIEWPOINT, FIELD 2</td>
<td>DX</td>
<td>7</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>DY</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>DZ</td>
<td>9</td>
</tr>
<tr>
<td>DROLL</td>
<td>ADWVRDL</td>
<td>10</td>
</tr>
<tr>
<td>DPITCH</td>
<td>ADWVPTD</td>
<td>11</td>
</tr>
<tr>
<td>DYAW</td>
<td>ADWVPYWD</td>
<td>12</td>
</tr>
<tr>
<td>MOVING MODEL #1, FIELD 1</td>
<td>X</td>
<td>13</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>Y</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>15</td>
</tr>
<tr>
<td>ROLL</td>
<td>ADWMI</td>
<td>16</td>
</tr>
<tr>
<td>PITCH</td>
<td>ADWMIPT</td>
<td>17</td>
</tr>
<tr>
<td>YAW</td>
<td>ADWMIY</td>
<td>18</td>
</tr>
<tr>
<td>MOVING MODEL #1, FIELD 2</td>
<td>DX</td>
<td>19</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>DY</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>DZ</td>
<td>21</td>
</tr>
<tr>
<td>DROLL</td>
<td>ADWMI1LD</td>
<td>22</td>
</tr>
<tr>
<td>DPITCH</td>
<td>ADWMIPTD</td>
<td>23</td>
</tr>
<tr>
<td>DYAW</td>
<td>ADWMIYWD</td>
<td>24</td>
</tr>
<tr>
<td>MOVING MODEL #2, FIELD 1</td>
<td>X</td>
<td>25</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>Y</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>27</td>
</tr>
<tr>
<td>ROLL</td>
<td>ADWMI2</td>
<td>28</td>
</tr>
<tr>
<td>PITCH</td>
<td>ADWMI2PT</td>
<td>29</td>
</tr>
<tr>
<td>YAW</td>
<td>ADWMI2Y</td>
<td>30</td>
</tr>
<tr>
<td>MOVING MODEL #2, FIELD 2</td>
<td>DX</td>
<td>31</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>DY</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>DZ</td>
<td>33</td>
</tr>
<tr>
<td>DROLL</td>
<td>ADWMI2LD</td>
<td>34</td>
</tr>
<tr>
<td>DPITCH</td>
<td>ADWMI2PTD</td>
<td>35</td>
</tr>
<tr>
<td>DYAW</td>
<td>ADWMI2YWD</td>
<td>36</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>SYMBOL</td>
<td>WORD</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>MOVING MODEL #3</td>
<td>X</td>
<td>ADWM3X 37</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>Y</td>
<td>ADWM3Y 38</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWM3Z 39</td>
</tr>
<tr>
<td>ROLL</td>
<td>ADWM3RL 40</td>
<td></td>
</tr>
<tr>
<td>PITCH</td>
<td>ADWM3PT 41</td>
<td></td>
</tr>
<tr>
<td>YAW</td>
<td>ADWM3YW 42</td>
<td></td>
</tr>
<tr>
<td>MOVING MODEL #3, FIELD 2</td>
<td>DX</td>
<td>ADWM3XD 43</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>DY</td>
<td>ADWM3YD 44</td>
</tr>
<tr>
<td></td>
<td>DZ</td>
<td>ADWM3ZD 45</td>
</tr>
<tr>
<td>DROLL</td>
<td>ADWM3RLD 46</td>
<td></td>
</tr>
<tr>
<td>DPITCH</td>
<td>ADWM3PTD 47</td>
<td></td>
</tr>
<tr>
<td>YAW</td>
<td>ADWM3YWD 48</td>
<td></td>
</tr>
<tr>
<td>MOVING MODEL #4</td>
<td>X</td>
<td>ADWM4X 49</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>Y</td>
<td>ADWM4Y 50</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWM4Z 51</td>
</tr>
<tr>
<td>ROLL</td>
<td>ADWM4RL 52</td>
<td></td>
</tr>
<tr>
<td>PITCH</td>
<td>ADWM4PT 53</td>
<td></td>
</tr>
<tr>
<td>YAW</td>
<td>ADWM4YW 54</td>
<td></td>
</tr>
<tr>
<td>MOVING MODEL #4, FIELD 2</td>
<td>DX</td>
<td>ADWM4XD 55</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>DY</td>
<td>ADWM4YD 56</td>
</tr>
<tr>
<td></td>
<td>DZ</td>
<td>ADWM4ZD 57</td>
</tr>
<tr>
<td>DROLL</td>
<td>ADWM4RLD 58</td>
<td></td>
</tr>
<tr>
<td>DPITCH</td>
<td>ADWM4PTD 59</td>
<td></td>
</tr>
<tr>
<td>YAW</td>
<td>ADWM4YWD 60</td>
<td></td>
</tr>
<tr>
<td>MOVING MODEL #5</td>
<td>X</td>
<td>ADWM5X 61</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>Y</td>
<td>ADWM5Y 62</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWM5Z 63</td>
</tr>
<tr>
<td>ROLL</td>
<td>ADWM5RL 64</td>
<td></td>
</tr>
<tr>
<td>PITCH</td>
<td>ADWM5PT 65</td>
<td></td>
</tr>
<tr>
<td>YAW</td>
<td>ADWM5YW 66</td>
<td></td>
</tr>
<tr>
<td>MOVING MODEL #5, FIELD 2</td>
<td>DX</td>
<td>ADWM5XD 67</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>DY</td>
<td>ADWM5YD 68</td>
</tr>
<tr>
<td></td>
<td>DZ</td>
<td>ADWM5ZD 69</td>
</tr>
<tr>
<td>DROLL</td>
<td>ADWM5RLD 70</td>
<td></td>
</tr>
<tr>
<td>DPITCH</td>
<td>ADWM5PTD 71</td>
<td></td>
</tr>
<tr>
<td>YAW</td>
<td>ADWM5YWD 72</td>
<td></td>
</tr>
</tbody>
</table>
## TABLE 3(A) (CONTINUED)

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>SYMBOL</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVING MODEL #6, FIELD 1</td>
<td>X</td>
<td>73</td>
</tr>
<tr>
<td>POSITION, ATTITUDE</td>
<td>Y</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>75</td>
</tr>
<tr>
<td>ROLL</td>
<td>ADWM6RL</td>
<td>76</td>
</tr>
<tr>
<td>PITCH</td>
<td>ADWM6PT</td>
<td>77</td>
</tr>
<tr>
<td>YAW</td>
<td>ADWM6YW</td>
<td>78</td>
</tr>
<tr>
<td>DX</td>
<td>ADWM6XD</td>
<td>79</td>
</tr>
<tr>
<td>DY</td>
<td>ADWM6YD</td>
<td>80</td>
</tr>
<tr>
<td>DZ</td>
<td>ADWM6ZD</td>
<td>81</td>
</tr>
<tr>
<td>DROLL</td>
<td>ADWM6RLD</td>
<td>82</td>
</tr>
<tr>
<td>DPITCH</td>
<td>ADWM6PTD</td>
<td>83</td>
</tr>
<tr>
<td>DYAW</td>
<td>ADWM6YWD</td>
<td>84</td>
</tr>
<tr>
<td>FLOLS</td>
<td>ADWFGS</td>
<td>85</td>
</tr>
<tr>
<td>HOOK-TO-EYE ROLL ANGLE</td>
<td>ADWFRA</td>
<td>86</td>
</tr>
<tr>
<td>BRIGHTNESS</td>
<td>ADWFBR</td>
<td>87</td>
</tr>
<tr>
<td>MEATBALL LOCATION</td>
<td>ADWMT</td>
<td>88</td>
</tr>
</tbody>
</table>
TABLE 3(A) (CONTINUED)

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>SYMBOL</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>TARGET IMAGE</td>
<td>FIELD 1:</td>
<td>ROLL</td>
</tr>
<tr>
<td>PROJECTOR</td>
<td></td>
<td>PITCH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YAW</td>
</tr>
<tr>
<td>FIELD 2:</td>
<td>ROLL</td>
<td>ADWTRL2 92</td>
</tr>
<tr>
<td></td>
<td>PITCH</td>
<td>ADWTPT2 93</td>
</tr>
<tr>
<td></td>
<td>YAW</td>
<td>ADWTYW2 94</td>
</tr>
<tr>
<td>ZOOM RATIO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET VIEW</td>
<td>FIELD 1:</td>
<td>SINE PITCH</td>
</tr>
<tr>
<td>WINDOW ATTITUDE</td>
<td></td>
<td>COSINE PITCH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SINE YAW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COSINE YAW</td>
</tr>
<tr>
<td>FIELD 2:</td>
<td>SINE PITCH</td>
<td>ADWWSP2 100</td>
</tr>
<tr>
<td></td>
<td>COSINE PITCH</td>
<td>ADWWCP2 101</td>
</tr>
<tr>
<td></td>
<td>SINE YAW</td>
<td>ADWWSY2 102</td>
</tr>
<tr>
<td></td>
<td>COSINE YAW</td>
<td>ADWWCY2 103</td>
</tr>
<tr>
<td>BOMB IMPACT LOCATIONS</td>
<td>X1</td>
<td>ADWBIY1 105</td>
</tr>
<tr>
<td></td>
<td>Y1</td>
<td>ADWBIY2 107</td>
</tr>
<tr>
<td></td>
<td>X2</td>
<td>ADWBIY2 106</td>
</tr>
<tr>
<td></td>
<td>Y2</td>
<td>ADWBIY2 107</td>
</tr>
<tr>
<td>GUNNERY IMPACT LOCATIONS</td>
<td>X1</td>
<td>ADWGIY1 109</td>
</tr>
<tr>
<td></td>
<td>Y1</td>
<td>ADWGIY2 111</td>
</tr>
<tr>
<td></td>
<td>X2</td>
<td>ADWGIY2 110</td>
</tr>
<tr>
<td></td>
<td>Y2</td>
<td>ADWGIY2 111</td>
</tr>
</tbody>
</table>
### TABLE 3. (CONTINUED)
**SEL TO DEC INTERFACE**

**B) CONTROL FLAGS**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>WORD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TARGET CHANNEL CS DISPLAY CONTROL</strong></td>
<td></td>
</tr>
<tr>
<td>LDWFL (FLOLS ON)</td>
<td></td>
</tr>
<tr>
<td>LDWWO (WAVE OFF)</td>
<td></td>
</tr>
<tr>
<td>LDWVOA (AUX WAVE OFF)</td>
<td></td>
</tr>
<tr>
<td>LDWCT (CUT LIGHTS)</td>
<td>112</td>
</tr>
<tr>
<td><strong>BACKGROUND CHANNEL CS DISPLAY CONTROL</strong></td>
<td></td>
</tr>
<tr>
<td>LDWVC (VP CRASH)</td>
<td></td>
</tr>
<tr>
<td>LDWSF (SCENE FREEZE)</td>
<td></td>
</tr>
<tr>
<td>IDWBIF (BOMB IMPACT)</td>
<td></td>
</tr>
<tr>
<td>IDWGF (GUN IMPACT)</td>
<td>113</td>
</tr>
<tr>
<td>LDWRLC (ASM LAUNCH)</td>
<td></td>
</tr>
<tr>
<td>LDWSLC (SAM LAUNCH)</td>
<td></td>
</tr>
<tr>
<td>LDWAA (AA ACTIVE)</td>
<td></td>
</tr>
<tr>
<td>LDWAE (AA EXPLODE)</td>
<td></td>
</tr>
<tr>
<td>LDWCH (CLOSE HIT)</td>
<td></td>
</tr>
<tr>
<td>LDWOH (OWN HIT)</td>
<td></td>
</tr>
<tr>
<td>LDWCC (IC COMPLETE)</td>
<td></td>
</tr>
<tr>
<td>LDWNC (NEW COMMAND)</td>
<td></td>
</tr>
<tr>
<td>ASM EXPLORE</td>
<td></td>
</tr>
<tr>
<td>LDWRLE (KDHUD)</td>
<td>114</td>
</tr>
<tr>
<td>FIXED MODEL HIT CONTROL</td>
<td></td>
</tr>
<tr>
<td>IDWFHT</td>
<td>115</td>
</tr>
<tr>
<td>MOVING MODEL HIT CONTROL</td>
<td></td>
</tr>
<tr>
<td>IDWMHT</td>
<td>116</td>
</tr>
<tr>
<td>REALTIME COMMAND BUFFER</td>
<td></td>
</tr>
<tr>
<td>IDWNCD</td>
<td>117</td>
</tr>
<tr>
<td></td>
<td>118</td>
</tr>
<tr>
<td></td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>124</td>
</tr>
<tr>
<td></td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>126</td>
</tr>
<tr>
<td></td>
<td>127</td>
</tr>
</tbody>
</table>
(s) MIL setting for gun sight (1=1 mil, etc.)

(14) Fixed model hit control (set to 0-15 to make target 0-15 disappear)

(15) Moving model hit control (set to 1-6 to make moving model 1-6 blow up [only 3 and 4 used])

(16) Realtime command buffer (set to valid CIG commands)

b. The second task is the transfer of all CIG parameters required for weapon visual software to the appropriate weapon visual parameters. Note that all position parameters are in CIG Earth coordinates (Figure 7). These parameters include DEC status flags, realtime commands, and database target parameters and are shown in Tables 4 and 5. These parameters are specified below:

DEC parameters updated at 30 Hertz:

(1) Input error code (I/O error: CIG numerical code)

(2) Buffer error offset (byte offset to first I/O error)

(3) Buffer error replacement (first value used by CIG to replace, by default, the first erroneous data item found)

(4) Collision detection flag (true=collision)

(5) DEC status flag (true=DEC is operating)

(6) DEC #2 status flag (true=DEC #2 is operating)

(7) CMD inactive (true=next time big buffer is expected)

(8) Realtime command buffer (legal CIG command)

DEC parameters updated when new database loaded (one shot):

(1) Active environment set verification (set to integer values 1-32) indicates which database is loaded.

(a) DAYFORRESTAL = 1

(b) Standard Meridian = 2

(c) FLOLS Meridian = 4

(d) Gunnery Range = 8

(e) Twin Towns = 16
(f) River Valley = 32

(2) Active coordinate sets verification (2-64) indicates which coordinate sets are being used at the time.
   (a) Moving Model 1 = 2
   (b) Moving Model 2 = 4
   (c) Moving Model 3 = 8
   (d) Moving Model 4 = 16
   (e) Moving Model 5 = 32
   (f) Moving Model 6 = 64

(3) Fixed model target control (0-7) indicates the presence of that target in the database loaded.
   (a) There can be up to 8 fixed targets in each database.
   (b) These targets can be deleted by setting the fixed model hit indicator to simulate target destruction.

(4) Fixed model target identifiers (32 bytes) indicates the name of the target (4 letters).
   (a) Example: tank, towr, etc.

(5) Fixed model target positions (X, Y, Z positions).
   (a) There can be a total of eight targets

(6) Fixed model target kill radius (kill radius in feet)

c. The third task is a test driver. In this mode, all weapon visual software inputs to CIG are bypassed except for those generated by the test driver. The test mode is used for Visual/CIG Interface debugging. The interface buffer size is 512 bytes which is equivalent to 128 SEL words (4 byte words) and 256 DEC words (2 byte words). The test driver has the following two test options:

(1) The first test option fills the SEL to DEC interface buffer with integers incremented from 1 to 256. This puts an easily identifiable number in each of the 256 DEC (16 bit words) in the buffer which is a handy checkout tool.

(2) The second test option fills the SEL to DEC interface buffer with a walking bit pattern. Bits 0 through 32 are sequentially set for the first 32 SEL words (32 bit words). This is used for checking bits.
TABLE 4. DEC TO SEL INTERFACE
(A) FRAME RATE BUFFER

<table>
<thead>
<tr>
<th>INPUT ERROR CODE</th>
<th>IDWERR 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFER ERROR OFFSET</td>
<td>IDWRPT 2</td>
</tr>
<tr>
<td>BUFFER ERROR REPLACEMENT</td>
<td>IDWRR 3</td>
</tr>
<tr>
<td>LDWCOL</td>
<td>LDWST DEC STATUS (1)</td>
</tr>
<tr>
<td>CMD INACTIVE</td>
<td></td>
</tr>
<tr>
<td>REALTIME IDWRSP</td>
<td>COMMAND</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>
### TABLE 5. DEC TO SEL INTERFACE
#### (B) DATA BASE INITIALIZATION BUFFER

<table>
<thead>
<tr>
<th>IDWAES</th>
<th>ACTIVE ENVIRONMENT SET VERIFICATION</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDWACS</td>
<td>ACTIVE COORDINATE SETS VERIFICATION</td>
<td>20</td>
</tr>
<tr>
<td>IDWFTC</td>
<td>FIXED MODEL TARGET CONTROL</td>
<td>21</td>
</tr>
<tr>
<td>IDWFTN</td>
<td>FIXED MODEL TARGET IDENTIFIERS</td>
<td>22</td>
</tr>
<tr>
<td>\textbf{ASCE NAME}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>\textbf{FIXED MODEL TARGET 1}</td>
<td>POSITION X</td>
<td>ADWT1X</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>ADWT1Y</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWT1Z</td>
</tr>
<tr>
<td></td>
<td>\textbf{RADIUS OF VULNERABILITY}</td>
<td>ADWT1S</td>
</tr>
<tr>
<td>\textbf{FIXED MODEL TARGET 2}</td>
<td>X</td>
<td>ADWT2X</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>ADWT2Y</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWT2Z</td>
</tr>
<tr>
<td></td>
<td>\textbf{RV}</td>
<td>ADWT2S</td>
</tr>
<tr>
<td>\textbf{FIXED MODEL TARGET 3}</td>
<td>X</td>
<td>ADWT3X</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>ADWT3Y</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWT3Z</td>
</tr>
<tr>
<td></td>
<td>\textbf{RV}</td>
<td>ADWT3S</td>
</tr>
<tr>
<td>\textbf{FIXED MODEL TARGET 4}</td>
<td>X</td>
<td>ADWT4X</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>ADWT4Y</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWT4Z</td>
</tr>
<tr>
<td></td>
<td>\textbf{RV}</td>
<td>ADWT4S</td>
</tr>
</tbody>
</table>
## TABLE 5. (CONTINUED)

<table>
<thead>
<tr>
<th>FIXED MODEL TARGET 5</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>ADWT5X</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>ADWT5Y</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWT5Z</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>RV</td>
<td>ADWT5S</td>
<td>49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FIXED MODEL TARGET 6</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>ADWT6X</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>ADWT6Y</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWT6Z</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>RV</td>
<td>ADWT6S</td>
<td>53</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FIXED MODEL TARGET 7</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>ADWT7X</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>ADWT7Y</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWT7Z</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>RV</td>
<td>ADWT7S</td>
<td>57</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FIXED MODEL TARGET 8</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>ADWT8X</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>ADWT8Y</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>ADWT8Z</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>RV</td>
<td>ADWT8S</td>
<td>61</td>
</tr>
</tbody>
</table>
This subroutine computes the drive parameters for the carrier. Position, heading, velocity, heave commands come from flight module ZM3SM00A. This subroutine is called by the visual module ZM6SMOVE when the carrier database is loaded at the Experimenter Operator Station (EOS). Figure A-1 illustrates a typical carrier landing approach for the VTRS system. The carrier is a CIG model of the USS Forrestal.
CARRIER DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The carrier position and attitude are defined relative to the Carrier Reference Point (C.R.P.) in the flight software figure (A-2). Since the visual software needs moving model positions relative to the center of Gravity (C.G.), the following conversion is needed.

Figure A-2. Carrier Reference Point and Center of Gravity Definition in Carrier Axis System.

Define the X,Y,Z offset between the C.R.P. and the C.G. in the Carrier Axis System:

\[ AVXCMGF2 = 300. \]
\[ AVYCMGF2 = 0. \]
\[ AVZCMGF2 = 0. \]

WHERE: \( AVXCMGF2, AVYCMGF2, AVZCMGF2 \) are the X,Y,Z offset between C.R.P. and C.G. in Carrier Axis System.

300. = The X Offset in feet
0. = The Y Offset in feet
0. = The Z Offset in feet

Convert the offset into the CIG Earth Axis System:

\[ AVPXFST = AVXCMGF2*AVBTES11+AVYCMGF2*AVBTES12+AVZCMGF2*AVBTES13 \]
\[ AVPYOFST = AVXCMGF2*AVBTES21+AVYCMGF2*AVBTES22+AVZCMGF2*AVBTES23 \]
\[ AVPZOFST = AVXCMGF2*AVBTES31+AVYCMGF2*AVBTES32+AVZCMGF2*AVBTES33 \]

WHERE: \( AVPXFST, AVPYOFST, AVPZOFST \) are X,Y,Z offset components in CIG Earth Axis System.
\( AVXCMGF2, AVYCMGF2, AVZCMGF2 \) are X,Y,Z offset components in Carrier Axis System.

66
AVBTES11-AVBTES33 represents the target to earth transfer matrix.

Add the X,Y,Z offset components to the carrier position (relative to C.R.P.) to get the C.G. position of the carrier in the CIG Earth Axis System:

\[
\begin{align*}
AVWMX(2) &= ANCGEX + AVPXOFST \\
AVWMY(2) &= ANCGEY + AVPYOFST \\
AVWMZ(2) &= ANCGEZ + AVPZPFST
\end{align*}
\]

WHERE: 
AVWMX(2), AVWMY(2), AVWMZ(2) are X,Y,Z position of Carrier C.G. in CIG Earth Axis System. (Note that subscript 2 indicates carrier as moving model 2.)
ANCGEX, ANCGEY, ANCGEZ are X,Y,Z position of Carrier C.R.P. in CIG Earth Axis System.
AVPXOFST, AVPYOFST, AVPZPFST are X,Y,Z offsets between C.R.P. and C.G. in CIG Earth Axis System.

B. Since there is only a position offset between flight and visual reference points to the carrier, the flight and visual angles correlate one to one and can be passed directly from flight to visual parameters.

Pass flight attitude angles to visual parameters:

\[
\begin{align*}
AVWMYW(2) &= ANCVAT & \text{... Pass Yaw Angle} \\
AVWMPT(2) &= ANTHEC & \text{... Pass Pitch Angle} \\
AVWMRL(2) &= ANPHIC & \text{... Pass Roll Angle}
\end{align*}
\]

WHERE: 
AVWMRL(2), AVWMPT(2), AVWMYW(2) are visual yaw, pitch, roll carrier angles, CIG Earth Axis.
ANPHIC, ANTHEC, ANCVAT are flight yaw, pitch, roll carrier angles, CIG Earth Axis.

C. Next, extrapolations are computed for carrier position attitude. These are required for the CIG System which, runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of carrier position 16.7 milliseconds from now:

\[
\begin{align*}
AVWMDX(2) &= 1.5 * AVWX(2) - 0.5 * AVWX1 \\
AVWMYD(2) &= 1.5 * AVWY(2) - 0.5 * AVWY1 \\
AVWMDZ(2) &= 1.5 * AVWZ(2) - 0.5 * AVWZ1
\end{align*}
\]

WHERE: 
AVWMDX(2), AVWMDY(2), AVWMDZ(2) are extrapolated X, Y, Z carrier position, CIG Earth Axis System.

AVWX(2), AVWY(2), AVWZ(2) are current X, Y, Z carrier position, CIG Earth Axis System.
AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z carrier position, CIG Earth Axis System.

Save the previous frame position:

AVWMX1=AVWMX(2)  
AVWMY1=AVWMY(2)  
AVWMZ1=AVWMZ(2)

WHERE:  AVWMX1, AVWMY1, AVWMZ1 are last frames carrier X, Y, Z position, CIG Earth Axis System.  
AVWMX(2), AVWMY(2), AVWMZ(2) are current carrier X, Y, Z position, CIG Earth Axis System.

Compute the extrapolated value of carrier attitude 16.7 milliseconds from now:

AVWMYWD(2)=1.5*AVWMYW(2)-.5*AVWMYW1  
AVWMPTD(2)=1.5*AVWMPT(2)-.5*AVWMPT1  
AVWMRLD(2)=1.5*AVWMRL(2)-.5*AVWMRL1

WHERE:  AVWMYWD(2), AVWMPTD(2), AVWMRLD(2) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.  
AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis.  
AVWMYW1, AVWMPT1, AVWMRL1 are last frames yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

AVWMYW1=AVWMYW(2)  
AVWMPT1=AVWMPT(2)  
AVWMRL1=AVWMRL(2)

WHERE:  AVWMYW1, AVWMPT1, AVWMRL1 are last frames yaw, pitch, roll angles, CIG Earth Axis System.  
AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis System.

D. The input module, output module, and parameters used by this subroutine are listed below:

Input Module:  (1) ZM6SMOVE - Calls This Routine Upon Request  
(2) ZM3SNOOA - Inputs carrier (C.R.P.) position, attitude

Output Module: (2) ZM6SWCGI - CVA X, Y, Z Pos CIG Earth, and extrapolations  
- CVA PT, YW, RL Ang, CIG Earth, and extrapolations
This module computes the drive parameters for anti-aircraft flak. This subroutine is called by the visual module ZM6SMOVE when either the Twin Town or River Valley database is loaded and the flak launch flag LJBFLAK is set true at the experimenter operator station (EOS). Figure B-1 illustrates a typical flak burst for the River Valley database.
FLAK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The first step in the flak driver algorithm is to initialize the flak launch parameters. This includes placement of the flak on the launch pad (Figure B-2) and resetting flak timers and hit flags. As shown in Figure B-2, the flak is placed at the center of the launch pad.

Figure B-2. Location of FLAK and SAM Launch Pad in CIG Earth Axis.
Initialize flak starting position:

AJFLAKX=AJFLAKXS
AJFLAKY=AJFLAKYS
AJFLAKZ=AJFLAKZS

...PAD X position coordinate = 15,910 feet
...PAD Y position coordinate = 139,000 feet
...PAD Z position coordinate = 0 feet

Initialize flak attitude angles:

AJFLAKYW=0.
AJFLAKPT=90.
AJFLAKRL=0.

...Yaw = 0 degrees
...Pitch = 90 degrees
...Roll = 0 degrees

WHERE: AJFLAKX, AJFLAKY, AJFLAKZ are flak X, Y, Z current position, CIG Earth Axis.
AJFLAKXS, AJFLAKYS, AJFLAKZS are X, Y, Z launch position of flak, CIG Earth Axis System.
AJFLAKYW, AJFLAKPT, AJFLAKRL are current flak yaw, pitch, roll angles.

Flak launch flags and timers are initialized:

LJBFLAK=.FALSE.
LJBFLKFLT=.TRUE.
LJWCH=.FALSE.
LJWCH=.FALSE.
LJWAAE=.FALSE.
IJFLAKHT=0
IJFLAKTM=0
IJFTIME=60
IJFBURST=0

WHERE: LJBFLAKFLT = True When Flak is in Flight (Active)
LJBFLAK = True to Launch Flak (Set False After First Pass)
LJWCH = Flak Passed Within 100 Ft When = True (Close Hit)
LJWCH = Flak Passed Within 50 Ft When = True (Own Hit)
LJWAAE = Flak Explode Flag For New Hit
IJFLAKHT = Flak Flight Timer, 30 Per Second
IJFLAKTM = Close/Own Hit Timer, 30 Per Second
IJFTIME = Flak Burst Time = 60 For 2 Second Burst (60 program loops = 2 seconds)
IJFBURST = Flak Burst Counter = Initialized to 0 to Start

8. The second step is to flash the flak launch pad for 5 seconds. This is done to simulate the acquisition phase of a flak launch. The flashing pad indicates to the pilot that the flak site has locked on to the T-2C and is preparing to launch a flak against him.
Let pad flash for 5 seconds, then go to flak trajectory phase:

LJWSLC=.TRUE. ... Flash Flak Launch Pad
IJFLAKTM=IJFLAKTM+1 ... Increment Flak Flight Counter
IF(IJFLAKTM.LT.150) GO TO 20 ... Bypass Flak Trajectory for
                         First 150 Loops (5 Seconds)
LJWAAA=.TRUE. ... Make AAA Gun Barrels Flash
LJWSLC=.TRUE. ... Make AAA Pad Flash

20 CONTINUE ... Pass Flak Launch Position
LJWAAA=.FALSE. ... Turn Off AAA Gun Flash
LJWSLC=.FALSE. ... Turn Off Launch Pad Flash

WHERE:  IJFLAKTM is flak subroutine timer.
        LJWAAA = True to Flash AAA Gun Barrels
        LJWSLC = True to Flash AAA Launch Pad (So Pilot Can See It)
150 = 5 Seconds of Pad Flash = 150 Program Loops

C. The third step is to compute the flak trajectory and explosions. Six flak bursts are randomly put in front of the pilot's aircraft at two second intervals. The distance in front of the aircraft and the radius of the dispersion pattern for the flak can be controlled at the EOS. Figure B-3 shows a typical flak pattern. Flak bursts will continue for 12 seconds (each burst lasts 2 seconds) unless pilot is hit (burst within 10 feet) or pulls High G evasive maneuver (200 deg/sec in yaw, pitch, or roll). The screen will flash a transparent yellow for a near hit (burst within 50 feet) or a solid yellow for a hit.

Set up flak timer for 2 second flak burst:

IJFTIME=IJFTIME + 1 ... Increment Burst Timer
IF(IJFTIME.LT.60) GO TO 200 ... Let Each Flak Burst Stay Put
                              For 2 Seconds (Lets Aircraft
                              Fly By Flak Bursts Suspended
                              In Air)

WHERE: IJFTIME=60 = 2 Seconds of Flak Burst = Burst Counter
       60 = 2 Seconds (2 * 30 Program Loops)
       200 Is Entry Point For Next Flak Position

Set up flak reference point in body axis system:

AJFLAKXS=320. ... Put Reference Point for Flak Position 320 Feet in Front of
AJFLAKYS=0.    Pilot's Eyepoint (320 Feet Out
AJFLAKZS=0.    +X Axis)
SIX RANDOM FLAK POSITIONS ARE COMPUTED AT 2 SECOND INTERVALS. EACH FLAK POSITION REMAINS FIXED IN SPACE WHILE AIRCRAFT FLIES AROUND/THROUGH EXPLOSION. THE DISPERSION DISTANCE CAN BE CONTROLLED AT THE EOS TO INCREASE/DECREASE PILOT AVOIDANCE TASK IF PILOT HITS FLAK (PASSES WITHIN 10 FEET), HE GETS SOLID YELLOW FIREBALL ON VISUAL SCREEN.

Figure B-3. FLAK Burst Position in CIG Earth Axis System.

73
WHERE:  AJFLAKXS, AJFLAKYS, AJFLAKZS is flak reference point, CIG Aircraft
Axis System.
320. = 320 Feet in the Aircraft X Axis
0 = 0 Feet in Aircraft Y Axis
0 = 0 Feet in Aircraft Z Axis

Get flak reference point in earth axis system.

AJFLAKXE=AJFLAKXS*AVBDER11+AJFLAKYS*AVBDER12+AJFLAKZS*AVBDER13
AJFLAKYE=AJFLAKXS*AVBDER21+AJFLAKYS*AVBDER22+AJFLAKZS*AVBDER23
AJFLAKZE=AJFLAKXS*AVBDER31+AJFLAKYS*AVBDER32+AJFLAKZS*AVBDER33

Compute random X, Y, Z position components for flak and add them to
flak reference point (320 feet in front of eyepoint):

AJFLAKX=AVPXCEP + AJFLAKXE
      + ZAROMA*AJVFMTS
AJFLAKY=AVPYCEP + AJFLAKYE
      + ZAROMA*AJVFMTS
AJFLAKZ=AVPZCEP + AJFLAKZE + ZAROMA*AJVFMTS

WHERE:  AJFLAKX, AJFLAKY, AJFLAKZ are random X, Y, Z flak positions, CIG
Earth Axis System.
AVPXCEP, AVPYCEP, AVPZCEP are X, Y, Z eyepoint positions, CIG Earth
Axis System.
AJFLAKXE, AJFLAKYE, AJFLAKZE are X, Y, Z flak reference point
positions, CIG Earth Axis.
ZAROMA is a random decimal number which varies between 0 and 1
produced by random number generator.
AJVFMTS is scale factor for random flak burst positions (this
controls magnitude of dispersion pattern).

The timer for the flak burst (length of each explosion) is reset and
flak counter (number of flak bursts) is incremented:

IJFTIME=0
IJFBURST=IJFBURST + 1

WHERE:  IJFTIME is length of flak bursts (6 for 2 seconds).
IJFBURST is number of flak bursts.

Next, scoring is determined for flak explosions. If the pilot
evades 200 deg/sec in yaw, pitch, or roll; or the flak misses by 50 feet, then
no hit is scored. If the flak miss is less than 50 feet a close hit occurs.
If the flak miss is less than 10 feet, a direct hit is scored:

200 CONTINUE        .. Entry Point for Flak Scoring
        IF(IJFLAKTM.GT.510) GO TO 225  .. Flak Driver Out of Time
        Reset parameters and Exit
Let flak miss if pilot evades 200 deg/sec in any attitude:

IF(AVFDPA.GT.200) GO TO 250  .. Roll Exceeds 200 Deg/Sec
IF(AVFDQA.GT.200) GO TO 250  .. Yaw Exceeds 200 Deg/Sec
IF(AVFDRA.GT.200) GO TO 250  .. Pitch Exceeds 200 Deg/Sec

WHERE: AVFDPA, AVFDQA, AVFDRA are aircraft roll, yaw, pitch rate in deg/sec
CIG Body Axis.

IJFLAKTM is FLAK subroutine counter (increments once per program loop)

510 = 5 seconds pad flash + 6 flak bursts (2 seconds each) =
   (5 + 6 x 2) x 30 = 510 loops
250 = Program entry point beyond flak hit/near hit code

Compute flak miss distance:

AJXMIS=AJFLAKX-AVPXCEP
AJYMIS=AJFLAKY-AVPYCEP
AJZMIS=AJFLAKZ-AVPZCEP
AJVFMIS=SQRT(AJXMIS*AJXMIS+ AJYMIS*AJYMIS+ AJZMIS*AMZMIS)

WHERE: AJXMIS, AJYMIS, AJZMIS are X, Y, Z flak miss distance components in
CIG Earth Axis System.
AJFLAKX, AJFLAKY, AJFLAKZ are X, Y, Z flak positions in CIG Earth
Axis System.
AVPXCEP, AVPYCEP, AVPZCEP are X, Y, Z eyepoint positions in CIG
Earth Axis System.
AJVFMIS is vector magnitude of flak miss distance.

Check for close hit and direct hit. If a close hit occurs, set close hit flag. If a direct hit occurs set direct hit flag and exit program after 5 seconds of solid yellow on screen:

IF(AJVFMIS.LT.50.) LJWCH=.TRUE.  .. If Close Hit, Set Flag True (transparent yellow on screen)
IF(AJVFMIS.GT.10.) GO TO 250  .. No Direct Hit, So Keep Shooting Flak
LJWOH=.TRUE.  Set "HIT" TRUE  .. Direct Hit, So Kill Pilot (Solid Yellow on Screen)

WHERE: AJVFMIS is magnitude of flak miss distance
LJWCH is close hit flag
LJWOH is hit flag
50 = 50 feet miss distance for close hit
10 = 10 feet miss distance for direct hit
Now the flak trajectory is complete, so the flak is now put out in the south forty and its attitude reset:

225 CONTINUE 
   .. Flak Termination Entry Point
   AJFLAKX=99999999.  .. Put X, Y, Z Flak Position in South Forty
   (removes flak from scene)
       AJFLAKY=99999999.
       AJFLAKZ=99999999.
   AJFLAKPT=90.0 
   .. Reset Flak Yaw, Pitch, Roll Angles
       AJFLAKYW=0.0
       AJFLAKRL=0.0

Turn off AA explode, close hit, own hit, flak flight flags. Let flak hit flags persist for 5 seconds:

   IJFLAKHT=IJFLAKHT+1   .. Increment Hit Timer
   IF(IJFLAKHT.LT.150) GO TO 250   .. Let Hits Persist for 5 Sec
       LJWAAE=.FALSE.   .. Reset AA Explode
       LJWCH=.FALSE.   .. Reset Close Hit Flag
       LJWOH=.FALSE.   .. Reset Own Hit Flag
       LJFLKFLT=.FALSE.   .. Reset Flak Flight Flag

WHERE:  IJFLAKHT is Flak Hit Timer = 150 for 5 Seconds
       LJWAAE is Flak Explode Flag = True for Explosion
       LJWCH is Flak Close Hit Flag = True for Close Hit
       LJWOH is Flak Own (Direct) Hit = True for Direct Hit
       LJFLKFLT is Flak Flight Timer = True for Active Flak

250 = program address that bypasses flak reset code

D. Next, extrapolations are computed for flak position attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

First, the flak explode timer is checked and turned off after position.

Compute the extrapolated value of flak position 16.7 milliseconds from now:

250 CONTINUE 
Pass flak position to visual:

   AVWMX(5)=AJFLAKX
   AVWMY(5)=AJFLAKY
   AVWMZ(5)=AJFLAKZ
   AVWMDX(5)=1.5*AVWMX(5) - .5*AVWMDX(5)
   AVWMDY(5)=1.5*AVWMDY(5) - .5*AVWMY(5)
   AVWMDZ(5)=1.5*AVWMDZ(5) - .5*AVWMZ(5)
76
WHERE: AVWMXD(5), AVWMMYD(5), AVWMZD(5) are extrapolated X, Y, Z flak position, CIG Earth Axis System.
AVWMX(5), AVWMMY(5), AVWMZ(5) are current X, Y, Z flak position, CIG Earth Axis System.
AVWMXI, AVWMMYI, AVWMZI are last frame's X, Y, Z flak position, CIG Earth Axis System.

Save the previous frame position.

AVWMXI=AVWMX(5)
AVWMMYI=AVWMMY(5)
AVWMZI=AVWMZ(5)

WHERE: AVWMXI, AVWMMYI, AVWMZI are last frame's flak X, Y, Z position, CIG Earth Axis System.
AVWMX(5), AVWMMY(5), AVWMZ(5) are current flak X, Y, Z position, CIG Earth Axis System.

Pass flak attitude angles to visual:

AVWMMYW(5)=AJFLAKYW
AVWMMPT(5)=AJFLAKPT
AVWMRL(5)=AJFLAKRL

Compute the extrapolated value of flak attitude 16.7 milliseconds from now:

AVWMMYW(5)=AVWMYW(5)
AVWMMPT(5)=AVWMPT(5)
AVWMRL(5)=AVWMRL(5)

WHERE: AVWMMYW(5), AVWMMPT(5), AVWMRL(5) are current yaw, pitch, roll angles, CIG Earth Axis.
AVWMMYW1, AVWMMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

AVWMMYW1=AVWMMYW(5)
AVWMMPT1=AVWMMPT(5)
AVWMRL1=AVWMRL(5)

WHERE: AVWMMYW1, AVWMMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
AVWMMYW(5), AVWMMPT(5), AVWMRL(5) are current yaw, pitch, roll angles, CIG Earth Axis System.
E. The input module, output module, and input/output parameters are listed below:

Input Module: (1) ZM6SWCGI - Flak X, Y, Z Start Pos, CIG Earth
               (2) ZM6SW033 - X, Y, Z Eyept Pos, CIG Earth
               - AC Attitude acceleration angles (yaw, pitch, roll)
               - Flak Launch, T/F

Output Module: (1) ZM6SWCGI - Flak X, Y, Z Pos, CIG Earth, extrapolations
                             - Flak Pt, Yw, Rl, CIG Earth, extrapolations
                             - Launch, Close Hit, Hit, T/F
APPENDIX C

FLOLS DRIVER

This subroutine computes the position and attitude of the fresnel lens optical landing system (FLOLS) and the FLOLS meatball position. The FLOLS is a landing aid used by the pilot to stay at the correct glideslope (elevation) angle when making landings on either the carrier or FLOLS Meridian databases. This subroutine is automatically called by the module ZM6SMOVE when one of the above data bases is loaded. Figure A-1 shows the CIG model of the FLOLS (black box with 2 horizontal white lines and white dot midway between the 2 lines) located to the left of the aircraft carrier runway. The two white lines represent the FLOLS datum line and the white dot is the FLOLS meatball. The pilot is on correct glideslopes when the meatball is level with the datum line. If the meatball is displaced upwards by a distance equal to twice the diameter of the meatball, the pilot is "2 balls" high (3/4 degree above glideslope). An equal negative displacement would be 2 balls low. The pilot must be within 2 balls displacement for a safe landing.

FLOLS DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION.

A. First, the FLOLS position and attitude is determined. There are two situations that must be addressed: (1) Aircraft Carrier FLOLS; (2) Meridian Airport FLOLS.

1. The FLOLS position and attitude for the Forrestal Aircraft Carrier is illustrated in Figure C-1. These parameters are defined in the carrier axis system and transformed into the CIG Earth Axis System:

```
AVPXFLCM = -28.0 feet
AVPYFLCM = -134.834 feet
AVPZFLCM = -63.028 feet

AVPXFLGT = AVPXFLCM * AVBETS11 + AVPYFLCM * AVBETS12 + AVPZFLCM * AVBETS13
AVPYFLGT = AVPXFLCM * AVBETS21 + AVPYFLCM * AVBETS22 + AVPZFLCM * AVBETS23
AVPZFLGT = AVPXFLCM * AVBETS31 + AVPYFLCM * AVBETS32 + AVPZFLCM * AVBETS33
```

WHERE: AVPXFLGT, AVPYFLGT, AVPZFLGT are FLOLS X, Y, Z position offsets from carrier C.G., CIG Earth Axis.
AVPXFLCM, AVPYFLCM, AVPZFLCM are FLOLS X, Y, Z position offsets from carrier C.G., Carrier Axis.
AVBETS11-33 are target to earth matrix elements.

Next, the above offsets are added to the carrier position in CIG Earth Axis System to get the FLOLS position:

```
AVWMX(5) = AVWMX(2) + AVPXFLGT
AVWMY(5) = AVWMY(2) + AVPYFLGT
AVWMZ(5) = AVWMZ(2) + AVPZFLGT
```
Figure C-1. FLOLS Position on Carrier in Carrier Axis System.

WHERE: AVWMX(5), AVWMY(5), AVWMZ(5) are X, Y, Z FLOLS positions in CIG Earth Axis System. (Note: The CIG system presently ignores these parameters and the FLOLS position on the ship is controlled by the CIG system.) AVWMX(2), AVWMY(2), AVWMZ(2) are X, Y, Z carrier positions in CIG Earth Axis System. AVPXFILGT, AVPYFILGT, AVPZFILGT are X, Y, Z FLOLS offset from carrier in CIG Earth Axis.

Compute the FLOLS attitude in the CIG earth axis system:

\[
\text{AVWMRL}(5) = \text{AVWMRL}(2) \quad \text{.. Let FLOLS Image Roll With Carrier}
\]
\[
\text{AVWMPT}(5) = \text{AVWMPT}(2) + \text{AvFLSELV} \quad \text{.. Pass FLOLS Elevation}
\]
\[
\text{AVWMYW}(5) = \text{AVWMYW}(2) - 10.5 \quad \text{.. Align FLOLS with carrier runway (rotated -10.5 degrees from carrier)}
\]

WHERE: AVWMRL(5), AVWMPT(5), AVWMYW(5) are FLOLS roll, pitch, yaws, CIG Earth Axis System. (Note: In the carrier database, these parameters are ignored by the CIG — it controls the FLOLS attitude.)
AVFLSELV is FLOLS elevation angle, CIG Earth Axis.
-10.5 is FLOLS yaw offset from carrier, CIG Earth Axis.

2. The FLOLS position and attitude for the FLOLS Meridian Airport is illustrated in Figure C-2. Note that the pitch and roll angles for the moving model inputs are different than the pitch and roll angles used for meatball calculations. This makes the FLOLS box appear upright (no roll, etc.) along the runway. Separate code is used to compute the meatball position which uses FLOLS roll and elevation angles to keep aircraft on correct glideslope. These parameters are defined in the CIG Earth Axis System, since the airport is defined in the CIG Earth Axis System.

AVWMX(5)=AVPXFLCW
AVWMY(5)=AVPYFLCW
AVWMZ(5)=AVPZFLCW
WHERE: AVWMX(5), AVWMY(5), AVWMZ(5) are X, Y, Z FLOLS positions in CIG Earth Axis System.
AVPXFLCW, AVPYFLCW, AVPZFLCW are X, Y, Z FLOLS positions variable inputs, CIG Earth Axis.

Get FLOLS yaw, pitch, roll angles for Meridian Airport:

AVWMLR(5)=AVFLSROL=0.
AVWMPT(5)=AVFLSELV=0.
AVWMMYW(5)=AVFLSHDG=8.864

FLOLS ATTITUDE:
YAW = +8.864°
PITCH = 0°
ROLL = 0°

FLOLS POSITION:
X = 4170'
Y = 8295'
Z = 4.5'

RUNWAY ROTATED 8.864° TO RIGHT

Figure C-2. FLOLS Position and Attitude for Meridian Airport Data Base.
WHERE: AVWMYW(5), AVWMPT(5), AVWMRL(5) are yaw, pitch, roll angles, CIG Earth Axis System.
AVFLSROL, AVFLSELV, AVFLSHDG are yaw, pitch, roll angle inputs for FLOLS in CIG Earth Axis.

B. Next, extrapolations are computed for FLOLS position attitude. These are required for the CIG System which, runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of FLOLS position 16.7 milliseconds from now:

\[ \begin{align*}
    AVWMXD(5) &= 1.5 \times AVWMX(5) - 0.5 \times AVWMX1 \\
    AVWMYD(5) &= 1.5 \times AVWMY(5) - 0.5 \times AVWMY1 \\
    AVWMZD(5) &= 1.5 \times AVWMZ(5) - 0.5 \times AVWMZ1
\end{align*} \]

WHERE: AVWMXD(5), AVWMYD(5), AVWMZD(5) are extrapolated X, Y, Z FLOLS position, CIG Earth Axis System.
AVWMX(5), AVWMY(5), AVWMZ(5) are current X, Y, Z FLOLS position, CIG Earth Axis System.
AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z FLOLS position, CIG Earth Axis System.

Save the previous frame position:

\[ \begin{align*}
    AVWMX1 &= AVWMX(5) \\
    AVWMY1 &= AVWMY(5) \\
    AVWMZ1 &= AVWMZ(5)
\end{align*} \]

WHERE: AVWMX1, AVWMY1, AVWMZ1 are last frame's FLOLS X, Y, Z position, CIG Earth Axis System.
AVWMX(5), AVWMY(5), AVWMZ(5) are current FLOLS X, Y, Z position, CIG Earth Axis System.

Compute the extrapolated value of FLOLS attitude 16.7 milliseconds from now:

\[ \begin{align*}
    AVWMYWD(5) &= 1.5 \times AVWMYW(5) - 0.5 \times AVWMYW1 \\
    AVWMPTD(5) &= 1.5 \times AVWMPT(5) - 0.5 \times AVWMPT1 \\
    AVWMRLD(5) &= 1.5 \times AVWMRL(5) - 0.5 \times AVWMRL1
\end{align*} \]

WHERE: AVWMYWD(5), AVWMPTD(5), AVWMRLD(5) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW(5), AVWMPT(5), AVWMRL(5) are current yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

AVWMYW1=AVWMYW(5)  
AVWMPT1=AVWMPT(5)  
AVWMRL1=AVWMRL(5)

WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw pitch, roll angles, CIG Earth Axis System.
AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis System.

C. Next, the FLOLS landing parameters are computed. These include: (1) FLOLS control flags; (2) FLOLS meatball position computation.

1. First, the FLOLS control flags are determined. The control flags drive the FLOLS cut lights, wave off lights, and auxiliary cut lights. These lights are used by the Navy Landing Signal Control Officer (LSO) to issue landing commands to the pilot. For example, to abort a landing attempt, the wave off lights would be turned on by the LSO.

Check the FLOLS control flags set at the experimenter operator station (EOS) and pass their values to the visual software:

\[
\begin{align*}
    \text{LVCELTLGT} &= \text{LWCTLGT} \\
    \text{LVAUXLGT} &= \text{.NOT. LWVAXLGT} \\
    \text{LWAVLGT} &= \text{LWWVLGT}
\end{align*}
\]

WHERE: LVCELTLGT, LWCTLGT are visual and EOS FLOLS cut light control flags (respectively).
LVAUXLGT, LWVAXLGT are visual and EOS FLOLS auxiliary wave off control flags (respectively).
LWAVLGT, LWWVLGT are visual and EOS FLOLS wave off control flags (respectively).

2. Second, the FLOLS meatball drive is computed. The meatball angle defines the position of the FLOLS meatball in reference to the FLOLS datum line (see Figure C-3). When the meatball is in the center of the datum line, the pilot is exactly on glideslope. If the pilot is 3/4 of a degree high or low, the meatball will be displaced by 2 balls (its maximum displacement) up or down, respectively. The meatball angle is derived below:
Compute the eyepoint to FLOLS vector in CIG Earth Axis:

\[
\begin{align*}
AVXEEPFL &= AVPXCEP - AVWMX(5) \\
AVYEEPFL &= AVPYCEP - AVWMY(5) \\
AVZEEPFL &= AVPZCEP - AVWMZ(5)
\end{align*}
\]

WHERE:  
AVXEEPFL, AVYEEPFL, AVZEEPFL are X, Y, Z components of eyepoint to FLOLS vector, CIG Earth Axis System.  
AVPXCEP, AVPYCEP, AVPZCEP are X, Y, Z components of eyepoint vector, CIG Earth Axis.  
AVWMX(5), AVWMY(5), AVWMZ(5) are X, Y, Z components of FLOLS position, CIG Earth Axis System.

**Figure C-3. FLOLS Meatball Position Relative to Aircraft Glideslope.**

Next, the sine and cosine of FLOLS yaw, pitch, and roll are computed so that the Earth to FLOLS transfer matrix can be determined:

\[
\begin{align*}
AVFLSHDG &= (AVWMYW(5) + 180.) \quad \text{.. Convert CIG FLOLS Yaw to True FLOLS Yaw (FLOLS X axis must swing around 180 degrees to point at aircraft)} \\
AVSIN1 &= \sin(AVFLSHDG) \quad \text{.. Get Sine of FLOLS Yaw} \\
AVCOS1 &= \cos(AVFLSHDG) \quad \text{.. Get Cosine of FLOLS Yaw} \\
AVSIN2 &= \sin(AVFLSROL) \quad \text{.. Get Sine of FLOLS Roll} \\
AVCOS2 &= \cos(AVFLSROL) \quad \text{.. Get Cosine of FLOLS Roll} \\
AVSIN3 &= \sin(AVFLSELV) \quad \text{.. Get Sine of FLOLS Pitch} \\
AVCOS3 &= \cos(AVFLSELV) \quad \text{.. Get Cosine of FLOLS Pitch}
\end{align*}
\]
WHERE: AVFLSHDG, AVFLSELV, AVFLSROL are true FLOLS yaw, pitch, roll angles in degrees
AVWMMYW(5) is FLOLS yaw modified for CIG FLOLS Model, since the backside must face pilot, it is 180 degrees out of phase with true FLOLS yaw.
AVWMRL(5) is FLOLS roll angle, CIG Earth Axis.
AVWMP7(5) is FLOLS pitch angle, CIG Earth Axis.

The earth to FLOLS axis transfer matrix is now defined: In the standard right-handed system (it will have to be converted to the CIG Earth Axis Coordinate System in the next step).

\[
\begin{align*}
AVSEFL_{11} &= \text{AVCOS3} \times \text{AVCOS1} \\
AVSEFL_{12} &= \text{AVCOS3} \times \text{AVSIN1} \\
AVSEFL_{13} &= -\text{AVSIN3} \\
AVSEFL_{21} &= \text{AVSIN2} \times \text{AVSIN3} \times \text{AVCOS1} - \text{AVCOS2} \times \text{AVSIN1} \\
AVSEFL_{22} &= \text{AVSIN2} \times \text{AVSIN3} \times \text{AVSIN1} + \text{AVCOS2} \times \text{AVCOS1} \\
AVSEFL_{23} &= \text{AVSIN2} \times \text{AVCOS3} \\
AVSEFL_{31} &= \text{AVCOS2} \times \text{AVSIN3} \times \text{AVCOS1} + \text{AVSIN2} \times \text{AVSIN1} \\
AVSEFL_{32} &= \text{AVCOS2} \times \text{AVSIN3} \times \text{AVSIN1} - \text{AVSIN2} \times \text{AVCOS1} \\
AVSEFL_{33} &= \text{AVCOS2} \times \text{AVCOS3}
\end{align*}
\]

WHERE: AVSIN1, AVCOS1 are sine and cosine of FLOLS yaw.
AVSIN2, AVCOS2 are sine and cosine of FLOLS roll.
AVSIN3, AVCOS3 are sine and cosine of FLOLS pitch.
AVSEFL_{11-33} are earth to FLOLS matrix elements in a standard right-handed system.

The earth axis to FLOLS is now converted into the CIG Earth Axis System (X and Y axis interchanged, Z axis inverted):

\[
\begin{align*}
AVERFL_{11} &= \text{AVSEFL21} \\
AVERFL_{22} &= \text{AVSEFL11} \\
AVERFL_{31} &= -\text{AVSEFL31} \\
AVERFL_{12} &= \text{AVSEFL22} \\
AVERFL_{22} &= \text{AVSEFL12} \\
AVERFL_{32} &= -\text{AVSEFL32} \\
AVERFL_{13} &= \text{AVSEFL13} \\
AVERFL_{23} &= \text{AVSEFL13} \\
AVERFL_{33} &= -\text{AVSEFL33}
\end{align*}
\]

WHERE: AVERFL_{11-33} are CIG earth to FLOLS matrix elements.
AVSEFL_{11-33} are standard right-handed earth to FLOLS matrix elements.

Next, the eyepoint position is converted into the FLOLS Axis Coordinate system so that the meatball angle can be computed:
AVFLSPXP = AVXEEPFL*AVERFL11 + AVYEEPFL*AVERFL12 + AVZEEPFL*AVERFL13
AVFLSPYP = AVXEEPFL*AVERFL21 + AVYEEPFL*AVERFL22 + AVZEEPFL*AVERFL23
AVFLSPZP = AVXEEPFL*AVERFL31 + AVYEEPFL*AVERFL32 + AVZEEPFL*AVERFL33

WHERE: AVFLSPXP, AVFLSPYP, AVFLSPZP are X, Y, Z components of FLOLS to eyepoint vector, FLOLS Axis.
AVXEEPFL, AVYEEPFL, AVZEEPFL are X, Y, Z components of FLOLS to eyepoint vector, CIG Earth Axis.
AVERFL11-33 are earth to FLOLS matrix elements.

Now the meatball angle is computed. Note in Figure C-4 that the meatball virtual image is 150. feet behind the datum line in the FLOLS Axis System (Figure C-3) and .4166 feet below the datum line:

\[ AVTNFVRT = \frac{.41666 - AVFLSZVP}{AVFLSXVP + 150.} \]
\[ AVMTBLAG = 57.29328 \times AVTNFVRT \]
\[ AVMEATBL = 1.2266667 \times AVMTBLAG \]

WHERE: AVTNFVRT is the tangent of the glideslope angle.
AVFLSXVP, AVFLSZVP are X, Z components of FLOLS to eyepoint.
AVMTBLAG is the meatball angle in degrees (this parameter is sent to CIG to drive FLOLS)
AVMEATBL is the drive parameter for the EOS FLOLS meatball indicator (scaled for hardware).

Figure C-4. FLOLS Datum Line and Meatball Position, Side View.
D. The input module, output module, and input/output parameters are listed below:

**Input Module:**
1. ZM6SMOVE - Calls This Routine if Carrier or FLOLS Meridian is Loaded
   - FLOLS Roll, Pitch Set at EOS
2. WCARRIER - CVA Position, Attitude Par's

**Output Module:**
1. ZM6SWCGI - FLOLS X, Y, Z Position, CIG Earth, extrapolations
   - FLOLS Roll, Pitch, Yaw Angles, extrapolations
   - Meatball Angle
This subroutine computes the drive parameters for any moving model selected by the user. This subroutine is called by the visual module ZM6SMOVE when the flag general is set true. This is a special user subroutine which allows total user control of the moving model specified. All the other drivers are bypassed in this mode.

"GENERAL" DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The "GENERAL" position and attitude are provided by the user through the experimenter operator station (EOS) or the visual initializer software. The selected target index is also set for values 1-6 to specify which model is to be driven. The user must know which moving models are available on which database to get successful results.

Pass flight position parameters to visual:

I=BJPRTGI ... Get Target Index
AVWMX(I)=AJGEXT(I) ... Pass X Position
AVWMY(I)=AJGEYT(I) ... Pass Y Position
AVWMZ(I)=AJGEZT(I) ... Pass Z Position

WHERE: AVWMX(I), AVWMY(I), AVWMZ(I) are visual X, Y, Z positions of selected target, CIG Earth Axis.
AJGEXT(I), AJGEYT(I), AJGEZT(I) are X, Y, Z flight target positions, CIG Earth Axis.
BJPRTGI is the flight target (moving model) index which should be set to values 1-6.

Pass flight attitude angles to visual parameters:

AVWMYW(I)=AJPSIT(I) ... Pass Yaw Angle
AVWMPT(I)=AJGPITCH ... Pass Pitch Angle
AVWMRL(I)=AJGROLL ... Pass Roll Angle

WHERE: AVWMRL(I), AVWMPT(I), AVWMYW(I) are visual yaw, pitch, roll "GENERAL" angles, CIG Earth Axis.
AJPSIT(I), AJGPITCH, AJGROLL are flight yaw, pitch, roll angles, CIG Earth Axis System.

B. Next, extrapolations are computed for "GENERAL" position and attitude. These are required for the CIG System which, runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.
Compute the extrapolated value of "GENERAL" position 16.7 milliseconds from now:

\[ \begin{align*}
AVWMXD(I) &= 1.5*AVWMX(I) - 0.5*AVWMX1 \\
AVWMYD(I) &= 1.5*AVWMY(I) - 0.5*AVWMY1 \\
AVWMZD(I) &= 1.5*AVWMZ(I) - 0.5*AVWMZ1 
\end{align*} \]

WHERE: \( AVWMXD(I), AVWMYD(I), AVWMZD(I) \) are extrapolated X, Y, Z "GENERAL" position, CIG Earth Axis System.
\( AVWMX(I), AVWMY(I), AVWMZ(I) \) are current X, Y, Z "GENERAL" position, CIG Earth Axis System.
\( AVWMX1, AVWMY1, AVWMZ1 \) are last frame's X, Y, Z "GENERAL" position, CIG Earth Axis System.

Save the previous frame position:

\[ \begin{align*}
AVWMX1 &= AVWMX(I) \\
AVWMY1 &= AVWMY(I) \\
AVWMZ1 &= AVWMZ(I) 
\end{align*} \]

WHERE: \( AVWMX1, AVWMY1, AVWMZ1 \) are last frame's "GENERAL" X, Y, Z position, CIG Earth Axis System.
\( AVWMX(I), AVWMY(I), AVWMZ(I) \) are current "GENERAL" X, Y, Z position, CIG Earth Axis System.

Compute the extrapolated value of "GENERAL" attitude 16.7 milliseconds from now:

\[ \begin{align*}
AVWMYWD(I) &= 0.5*AVWMYW(I) - 0.5*AVWMYW1 \\
AVWMPTD(I) &= 1.5*AVWMPT(I) - 0.5*AVWMPT1 \\
AVWMRLD(I) &= 1.5*AVWMRL(I) - 0.5*AVWMRL1 
\end{align*} \]

WHERE: \( AVWMYWD(I), AVWMPTD(I), AVWMRLD(I) \) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
\( AVWMYW(I), AVWMPT(I), AVWMRL(I) \) are current yaw, pitch, roll angles, CIG Earth Axis.
\( AVWMYW1, AVWMPT1, AVWMRL1 \) are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

\[ \begin{align*}
AVWMYW1 &= AVWMYW(I) \\
AVWMPT1 &= AVWMPT(I) \\
AVWMRL1 &= AVWMRL(I) 
\end{align*} \]

WHERE: \( AVWMYW1, AVWMPT1, AVWMRL1 \) are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
\( AVWMYW(I), AVWMPT(I), AVWMRL(I) \) are current yaw, pitch, roll angles, CIG Earth Axis System.
C. The input, output module, and the input/output parameters used in this subroutine are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request

Output Module: (2) ZM6SWCGI - "GENERAL" X, Y, Z Pos CIG Earth, extrapolations
- "GENERAL" PT, YW, RL Ang, CIG Earth, extrapolations
APPENDIX E

AIR-TO-SURFACE ROCKET DRIVER SUBROUTINE

This subroutine computes the drive parameters for the air-to-surface missiles (ASM's) launched from the T-2C. These ASM's are 2.75 inch rockets which are launched in pairs from launchers located on the T-2C. The release position and attitude come from flight module ZM3SAO03 and position updates from module ZM3SA004. This module is called by the visual module ZM6SMOVE when weapon data bases are loaded and rockets are fired. Figure E-1 illustrates a typical rocket launch delivery.

Figure E-1. Air-to-Surface Missiles (2.75" Rockets).

ROCKET DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The rocket position and attitude is defined for each of the two rockets launched in the flight software. The visual software treats the two rockets as a single moving model and needs the centroid position between the two rockets (see Figure E-2) to present the correct picture in the dome. Therefore, the two flight rocket positions must be converted into a single centroid position for the visual software. The attitude angles are the same and hence are passed directly from flight to visual.

Convert the left and right rocket positions to a single centroid value and pass to visual:

\[
\begin{align*}
AVWMX(2) &= \frac{(AJAXPOS+AJAXPOS)}{2} + AJBY & \text{Centroid Release} \\
AVWMY(2) &= \frac{(AJAYPOS+AJAYPOS)}{2} + AJBX & \text{Point + Update Value} \\
AVWMI(2) &= AJAZPOS - AJBZ
\end{align*}
\]
WHERE: AVWMX(2), AVWMY(2), AVWMZ(2) are visual X, Y, Z centroid position of rocket, CIG Earth Axis System.
AJAXPOS R, AJAXPOSL are flight X position of right, left rockets, CIG Earth Axis System. (Rockets are separated by 22.42 feet on aircraft--see Figure E-2.)
AJAYPOS R, AJAYPOSL are flight Y position of right, left rockets, CIG Earth Axis System.
AJAZPOS is flight altitude of rockets at launch, CIG Earth Axis System.
AJBX, AJBY, AJBZ are flight rocket X, Y, Z position update (both rockets), Flight Earth Axis System.

Figure E-2. Rocket Centroid Definition in Rocket Axis System.

Pass flight rocket attitude angles to visual:

\[
\begin{align*}
AVWMYW(2) &= AJBPHI \quad \text{... Pass Yaw Angle} \\
AVWMPT(2) &= AJBTHET \quad \text{... Pass Pitch Angle} \\
AVWMLR(2) &= AJBPSI \quad \text{... Pass Roll Angle}
\end{align*}
\]

WHERE: AVWMYW(2), AVWMP T(2), AVWMRL(2) are visual yaw, pitch, roll rocket angles, CIG Earth Axis.
AJBPSI, AJBPHI AJBPSI are flight yaw, pitch, roll rocket angles, CIG Earth Axis.

B. Next, extrapolations are computed for rocket position and attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.
Compute the extrapolated value of rocket position 16.7 milliseconds from now:

\[
\begin{align*}
AVWMXD(2) &= 1.5*AVWMX(2) - 0.5*AVWMX1 \\
AVWMYD(2) &= 1.5*AVWMY(2) - 0.5*AVWMY1 \\
AVWMZD(2) &= 1.5*AVWMZ(2) - 0.5*AVWMZ1 \\
\end{align*}
\]

WHERE: AVWMXD(2), AVWMYD(2), AVWMZD(2) are extrapolated X, Y, Z rocket position, CIG Earth Axis System. AVWMX(2), AVWMY(2), AVWMZ(2) are current X, Y, Z rocket position, CIG Earth Axis System. AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z rocket position, CIG Earth Axis System.

Save the previous frame position:

\[
\begin{align*}
AVWMX1 &= AVWMX(2) \\
AVWMY1 &= AVWMY(2) \\
AVWMZ1 &= AVWMZ(2) \\
\end{align*}
\]

WHERE: AVWMX1, AVWMY1, AVWMZ1 are last frame's rocket X, Y, Z position, CIG Earth Axis System. AVWMX(2), AVWMY(2), AVWMZ(2) are current rocket X, Y, Z position, CIG Earth Axis System.

Compute the extrapolated value of rocket attitude 16.7 milliseconds from now:

\[
\begin{align*}
AVWMYWD(2) &= 1.5*AVWMYW(2) - 0.5*AVWMYW1 \\
AVWMPTD(2) &= 1.5*AVWMPT(2) - 0.5*AVWMPT1 \\
AVWMRLD(2) &= 1.5*AVWMRL(2) - 0.5*AVWMRL1 \\
\end{align*}
\]

WHERE: AVWMYWD(2), AVWMPTD(2), AVWMRLD(2) are extrapolated yaw, pitch, roll angles, CIG Earth Axis. AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis. AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

\[
\begin{align*}
AVWMYW1 &= AVWMYW(2) \\
AVWMPT1 &= AVWMPT(2) \\
AVWMRL1 &= AVWMRL(2) \\
\end{align*}
\]

WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System. AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis System.
C. The input module, output module, and parameters used by this subroutine are listed below:

Input Module:
1. ZM6SMOVE - Calls This Routine Upon Request
2. ZM3SA003 - Gives ASM Pos, Att at Release
3. ZM3SA04 - Generates ASM Pos, Att, Rates

Output Module:
2. ZM6SWCGI - ASM X, Y, Z Pos CIG Earth, extrapolations
   - ASM PT, YW, RL Ang, CIG Earth, extrapolations
<table>
<thead>
<tr>
<th>UNCLASSIFIED</th>
<th>NAVTRAEQUIPC-1H-335</th>
<th>NL</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 x 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 x 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>123</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

END DATE FUND  ATIC
This module computes the drive parameters for the surface-to-air missile (SAM). This subroutine is called by the visual module ZM6SMOVE when the Twin Town database is loaded and the SAM launch flag LJBSAM is set true at the experimenter operator station (EOS). Figure F-1 shows a SAM launch.

Figure F-1. Surface-to-Air Missile (SAM) Launch.

SAM DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The first step in the SAM driver algorithm is to initialize the SAM launch parameters. This includes placement of the SAM on the launch pad (Figure F-2) and resetting SAM timers and hit flags. As shown in Figure F-2, the SAM is placed at the center of the launch pad and with a pitch angle of 90 degrees.
Figure F-2. SAM Launch Pad Setup, CIG Earth Axis System.

Initialize SAM starting position:

\[
\begin{align*}
\text{AJSAMXP} & = \text{AJSAMXSP} \quad \ldots \text{PAD X position coordinate} = 15,910 \text{ feet} \\
\text{AJSAMYP} & = \text{AJSAMYSP} \quad \ldots \text{PAD Y position coordinate} = 139,000 \text{ feet} \\
\text{AJSAMZP} & = \text{AJSAMZSP} \quad \ldots \text{PAD Z position coordinate} = 0 \text{ feet}
\end{align*}
\]

WHERE:  
AJSAMXP, AJSAMYP, AJSAMZP are SAM X, Y, Z current position, CIG Earth Axis.  
AJSAMXSP, AJSAMYSP, AJSAMZSP are X, Y, Z launch position of SAM, CIG Earth Axis System.

Initialize SAM starting attitude:

\[
\begin{align*}
\text{AJSAMYAW} & = 0. \\
\text{AJSAMPTH} & = 90. \\
\text{AJSAMROL} & = 0.
\end{align*}
\]

WHERE:  
AJSAMYAW, AJSAMPTH, AJSAMROL are SAM attitude angles in CIG Earth Axis System. 
90 = 90 Degrees Up (SAM Pointed Straight Up) 
0 = 0 Degrees of Yaw and Roll

SAM launch flags and timers are initialized:

\[
\begin{align*}
\text{LJSAMFLT} & = .\text{TRUE.} \\
\text{LJSAMSFT} & = .\text{FALSE.} \\
\text{LJWCH} & = .\text{FALSE.}
\end{align*}
\]
B. The second step is to flash the SAM launch pad for 5 seconds. This is done to simulate the acquisition phase of a SAM launch. The flashing pad indicates to the pilot that the SAM site has locked on the T-2C; and is preparing to launch a SAM against him.

Let pad flash for 5 seconds, then go to SAM trajectory phase:

LJWSLC=.TRUE. ... Flash SAM Launch Pad
IJSAMTIM=IJSAMTIM+1 ... Increment SAM Flight Counter
IF(IJSAMTIM.LT.150) GO TO 100 ... Bypass SAM Trajectory for First 150 Loops (5 Seconds)
LJWSLC=.FALSE. ... Turn Off Launch Pad Flash
(SAM Trajectory Program) ... Compute SAM Trajectory

100 CONTINUE ... Pass SAM Launch Position To Visual (Still on Pad)

C. The third step is to compute the SAM trajectory. The SAM trajectory has two phases. In the first phase, the SAM lifts off the pad vertically and gradually turns toward the aircraft, acquiring full turn capability (9 G's) after 5 seconds of flight. This phase simulates the initial SAM position and attitude alignment on the target immediately after launch. The second phase is full pursuit (9 G's) of the target until one minute of flight or pilot flys below 200 feet altitude. Refer to Figure F-3 for the trajectory description below.

Compute SAM to target vector components:

\[
\begin{align*}
XV &= AVPXCEP - AJSAMXP \\
YV &= AVPYCEP - AJSAMYP \\
ZV &= AVPZCEP - AJSAMZP
\end{align*}
\]

WHERE: \(XV, YV, ZV\) are \(X, Y, Z\) components of SAM to target vector, CIG Earth Axis System.
\(AVPXCEP, AVPYCEP, AVPZCEP\) are \(X, Y, Z\) components of eyepoint position, CIG Earth Axis System.
AJSAMXP, AJSAMYP, AJSAMZP are X, Y, Z components of SAM position, CIG Earth Axis System.

Compute SAM to target miss distance:

\[ \text{AJSAMSL} = \sqrt{X^V \times X^V + Y^V \times Y^V + Z^V \times Z^V} \]

**WHERE:**
- AJSAMSL is SAM to target vector magnitude in feet.
- XV, YV, ZV are X, Y, Z components of SAM to target vector, CIG Earth Axis System.

Save previous SAM velocity vector:

\[ V_1X = AJSAMVX \]
\[ V_1Y = AJSAMVY \]
\[ V_1Z = AJSAMVZ \]

**WHERE:**
- V1X, V1Y, V1Z are X, Y, Z components of previous SAM velocity vector, CIG Earth Axis System.
- AJSAMVX, AJSAMVY, AJSAMVZ are X, Y, Z components of current SAM velocity vector, CIG Earth Axis System.

Figure F-3. SAM Trajectory Vectors, CIG Earth Axis System.
Compute new requested SAM velocity vector:

\[ \begin{align*}
A_{\text{JSAMVX}} &= A_{\text{JSAMVEL}} \times X / A_{\text{JSAMSL}} \\
A_{\text{JSAMVY}} &= A_{\text{JSAMVEL}} \times Y / A_{\text{JSAMSL}} \\
A_{\text{JSAMVZ}} &= A_{\text{JSAMVEL}} \times Z / A_{\text{JSAMSL}}
\end{align*} \]

WHERE:  
\( A_{\text{JSAMVX}}, A_{\text{JSAMVY}}, A_{\text{JSAMVZ}} \) are \( X, Y, Z \) components of SAM velocity vector, CIG Earth Axis.  
\( A_{\text{JSAMVEL}} \) is SAM velocity in ft/sec (normally 1200 ft/sec).  
\( X / A_{\text{JSAMSL}} \) is \( X \) component of SAM to target unit vector.  
\( Y / A_{\text{JSAMSL}} \) is \( Y \) component of SAM to target unit vector.  
\( Z / A_{\text{JSAMSL}} \) is \( Z \) component of SAM to target unit vector.

Compute requested acceleration (cross vector):

\[ \begin{align*}
V_{3X} &= A_{\text{JSAMVX}} - V_{1X} \\
V_{3Y} &= A_{\text{JSAMVY}} - V_{1Y} \\
V_{3Z} &= A_{\text{JSAMVZ}} - V_{1Z} \\
A_{\text{JSAMVC}} &= \sqrt{V_{3X}^2 + V_{3Y}^2 + V_{3Z}^2}
\end{align*} \]

WHERE:  
\( V_{3X}, V_{3Y}, V_{3Z} \) are \( X, Y, Z \) components of requested acceleration (cross vector), CIG Earth Axis System.  
\( A_{\text{JSAMVX}}, A_{\text{JSAMVY}}, A_{\text{JSAMVZ}} \) are components of requested SAM velocity (to get to the target), CIG Earth Axis System.  
\( V_{1X}, V_{1Y}, V_{1Z} \) are \( X, Y, Z \) components of previous SAM velocity vector, CIG Earth Axis System.  
\( A_{\text{JSAMVC}} \) is magnitude of cross vector components \( V_{3X}, V_{3Y}, V_{3Z}. \)

Compute maximum velocity change for specified SAM G limit:

\[ (\text{Maximum velocity change per pass} = 32.2 \times G \times AQ\text{DELT2}) \]

\[ A_{\text{JSAMVM}} = 32.2 \times A_{\text{JSAMG}} \times AQ\text{DELT2} \]

WHERE:  
\( A_{\text{JSAMVM}} \) is maximum velocity change per pass in Ft/Sec  
32.2 = Acceleration of Gravity = 32.2 Ft/Sec*Sec  
\( A_{\text{JSAMG}} \) is maximum G's SAM can pull in maneuvering to target.  
\( AQ\text{DELT2} \) is visual CPU integration constant (1/30 sec) per pass through this subroutine.

Let SAM turn slowly for first 5 seconds (Phase 1 Guidance):

\[ \text{IF}(I_{\text{JSAMTIM.GT.}}(I_{\text{JSAMTM}}+I_{\text{JSAMLTM}}+150)) \text{ GO TO 18} \]

\[ A_{\text{JSAMVM}} = (I_{\text{JSAMTM}}-(I_{\text{JSAMLTM}}+150))/150 \times A_{\text{JSAMVM}} \text{...INCR G'S WITH TIME} \]

18 CONTINUE

WHERE:  
\( I_{\text{JSAMTIM}} \) is SAM flight counter (increments 1 per pass).  
\( I_{\text{JSAMLTM}} \) is SAM Phase 1 counter (normally set to 150), which controls Phase 1 trajectory time span.  
150 = 30*5 = 5 seconds of time for phase 1 trajectory.  
\( A_{\text{JSAMVM}} \) is maximum G's SAM can pull.
Next, the acceleration is checked to see if the maximum allowed has been exceeded. If it has, only the maximum value is used in the trajectory computation. Otherwise, the requested acceleration required to track the target is used.

Test Acceleration to see if it exceeds maximum value allowed:

\[
\text{IF}(\text{AJSAMVC.LT.AJSAMVM}) \text{ GO TO 20} \quad \ldots \text{ Bypass Rate Limit Code If G's Not Exceeded}
\]

WHERE:
- AJSAMVC is the requested velocity change (acceleration) to guide to target in Ft/Sec*Sec.
- AJSAMVM is the maximum velocity change (acceleration) allowed (computed using maximum G limit).

Requested guidance will exceed maximum acceleration allowed.

Compute limited velocity vectors to guide on the target:

\[
\begin{align*}
V_{3LX} &= V3X \times \frac{AJSAMVM}{AJSAMVC} \\
V_{3LY} &= V3Y \times \frac{AJSAMVM}{AJSAMVC} \\
V_{3LZ} &= V3Z \times \frac{AJSAMVM}{AJSAMVC}
\end{align*}
\]

WHERE:
- \(V_{3LX}, V_{3LY}, V_{3LZ}\) are X, Y, Z components of velocity change (acceleration limited) to track target.
- \(V3X, V3Y, V3Z\) are X, Y, Z components of SAM velocity components change requested to track target.
- AJSAMVM is the maximum velocity change allowed.
- AJSAMVC is the requested velocity change to guide on target.

Compute new commanded SAM velocity vector (G limited):

\[
\begin{align*}
AJSAMVX &= VIX + V_{3LX} \\
AJSAMVY &= VIY + V_{3LY} \\
AJSAMVZ &= VIZ + V_{3LZ}
\end{align*}
\]

20 CONTINUE \( \ldots \text{ Branch to Here if G's Not Limited}\)

WHERE:
- AJSAMVX, AJSAMVY, AJSAMVZ are new SAM X, Y, Z velocity vector components, CIG Earth Axis System.
- \(V_{3LX}, V_{3LY}, V_{3LZ}\) are X, Y, Z components of allowed velocity change to track target.

Next, the SAM position and attitude are computed. The position is computed using the velocity components determined above. The attitude is computed from the angles defined by the vector between the previous SAM position and current SAM position.
Save the previous SAM position:

\[
\text{AJSAMXP}_1 = \text{AJSAMXP} \\
\text{AJSAMYP}_1 = \text{AJSAMYP} \\
\text{AJSAMZP}_1 = \text{AJSAMZP}
\]

\text{WHERE:}
\text{AJSAMXP}_1, \text{AJSAMYP}_1, \text{AJSAMZP}_1 \text{ are } X, Y, Z \text{ components of SAM position, last frame.}
\text{AJSAMXP, AJSAMYP, AJSAMZP \text{ are } X, Y, Z \text{ components of SAM current position, CIG Earth Axis.}}

Compute new SAM position in CIG Earth:

\[
\begin{align*}
\text{AJSAMXP} &= \text{AJSAMXP} + \text{AJSAMVX} \times \text{AQDELT2} \\
\text{AJSAMYP} &= \text{AJSAMYP} + \text{AJSAMVY} \times \text{AQDELT2} \\
\text{AJSAMZP} &= \text{AJSAMZP} + \text{AJSAMVZ} \times \text{AQDELT2}
\end{align*}
\]

Pass SAM position to visual (CIG):

\[
\begin{align*}
\text{AVWMX}(1) &= \text{AJSAMXP} \\
\text{AVWMY}(1) &= \text{AJSAMYP} \\
\text{AVWMZ}(1) &= \text{AJSAMZP}
\end{align*}
\]

\text{WHERE:}
\text{AJSAMXP, AJSAMYP, AJSAMZP \text{ are components of new SAM Position, CIG Earth Axis System.}}
\text{AJSAMVX, AJSAMVY, AJSAMVZ \text{ are } X, Y, Z \text{ components of SAM velocity, CIG Earth Axis System.}}
\text{AQDELT2 is integration constant for visual computer.}
\text{AVWMX(1), AVWMY(1), AVWMZ(1) \text{ is the current SAM position passed to CIG moving model 1.}}

Compute vector from previous SAM position to current SAM position:

\[
\begin{align*}
\text{AJSAMXPV} &= \text{AJSAMXP} - \text{AJSAMXP}_1 \\
\text{AJSAMYPV} &= \text{AJSAMYP} - \text{AJSAMYP}_1 \\
\text{AJSAMZPV} &= \text{AJSAMZP} - \text{AJSAMZP}_1
\end{align*}
\]

\text{WHERE:}
\text{AJSAMXPV, AJSAMYPV, AJSAMZPV \text{ are } X, Y, Z \text{ components of SAM previous to current position vector, CIG Earth Axis System.}}
\text{AJSAMXP, AJSAMYP, AJSAMZP \text{ are } X, Y, Z \text{ components of SAM current position, CIG Earth Axis.}}
\text{AJSAMXP}_1, \text{AJSAMYP}_1, \text{AJSAMZP}_1 \text{ are } X, Y, Z \text{ components of SAM previous position, CIG Earth.}

Compute the projection of the SAM previous to present position vector on the X-Y CIG Earth Axis Plane. This will be used to compute the SAM Heading angle:

\[
\text{AJSAMXYR} = \sqrt{\text{AJSAMXPV} \times \text{AJSAMXPV} + \text{AJSAMYPV} \times \text{AJSAMYPV}}
\]

103
Compute new SAM attitude:

\[
\begin{align*}
AJSAMPTH &= \arctan(AJSAMZPV, AJSAMXYR) \\
AJSAMYAW &= \arctan(AJSAMXPV, AJSAMYPV) \\
AJSAMROL &= \arctan(AJSAMZPV, AJSAMXPV)
\end{align*}
\]

Pass SAM attitude to visual (CIG):

\[
\begin{align*}
AVWMYW(1) &= AJSAMYAW \\
AVWMPT(1) &= AJSAMPTH \\
AVWMRL(1) &= AJSAMROL
\end{align*}
\]

WHERE: \(AJSAMPTH, AJSAMYAW, AJSAMROL\) are yaw, pitch, and roll angles of SAM, CIG Earth Axis. \(AVWMYW(1), AVWMPT(1), AVWMRL(1)\) is current SAM attitude passed to CIG moving model 1. \(AJSAMXPV, AJSAMYPV, AJSAMZPV\) are \(X, Y, Z\) components of SAM previous to present position vector, CIG Earth Axis System. \(AJSAMXYR\) is projection of SAM previous to present position vector on \(X-Y\) plane of CIG Earth Axis.

D. Next, extrapolations are computed for SAM position attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of SAM position 16.7 milliseconds from now:

\[
\begin{align*}
AVWMXD(1) &= 1.5 \times AVWMX(1) - 0.5 \times AVWMX1 \\
AVWMYD(1) &= 1.5 \times AVWMY(1) - 0.5 \times AVWMY1 \\
AVWMZD(1) &= 1.5 \times AVWMZ(1) - 0.5 \times AVWMZ1
\end{align*}
\]

WHERE: \(AVWMXD(1), AVWMYD(1), AVWMZD(1)\) are extrapolated \(X, Y, Z\) SAM position, CIG Earth Axis System. \(AVWMX(1), AVWMY(1), AVWMZ(1)\) are current \(X, Y, Z\) SAM position, CIG Earth Axis System. \(AVWMX1, AVWMY1, AVWMZ1\) are last frame's \(X, Y, Z\) SAM position, CIG Earth Axis System.

Save the previous frame position.

\[
\begin{align*}
AVWMX1 &= AVWMX(1) \\
AVWMY1 &= AVWMY(1) \\
AVWMZ1 &= AVWMZ(1)
\end{align*}
\]

WHERE: \(AVWMX1, AVWMY1, AVWMZ1\) are last frame's SAM \(X, Y, Z\) position, CIG Earth Axis System. \(AVWMX(1), AVWMY(1), AVWMZ(1)\) are current SAM \(X, Y, Z\) position, CIG Earth Axis System.
Compute the extrapolated value of SAM attitude 16.7 milliseconds from now:

\[
\begin{align*}
AVWMYWD(1) &= 1.5 \times AVWMYW(1) - 0.5 \times AVWMYW1 \\
AVWMPTD(1) &= 1.5 \times AVWMPT(1) - 0.5 \times AVWMPT1 \\
AVWMRLD(1) &= 1.5 \times AVWMRL(1) - 0.5 \times AVWMRL1
\end{align*}
\]

WHERE: \(AVWMYWD(1), AVWMPTD(1), AVWMRLD(1)\) are extrapolated yaw, pitch, roll angles, CIG Earth Axis. \(AVWMYW(1), AVWMPT(1), AVWMRL(1)\) are current yaw, pitch, roll angles, CIG Earth Axis. \(AVWMYW1, AVWMPT1, AVWMRL1\) are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

\[
\begin{align*}
AVWMYW1 &= AVWMYW(1) \\
AVWMPT1 &= AVWMPT(1) \\
AVWMRL1 &= AVWMRL(1)
\end{align*}
\]

WHERE: \(AVWMYW1, AVWMPT1, AVWMRL1\) are last frame's yaw, pitch, roll angles, CIG Earth Axis System. \(AVWMYW(1), AVWMPT(1), AVWMRL(1)\) are current yaw, pitch, roll angles, CIG Earth Axis System.

E. The input modules, output modules, and input/output parameters used by this routine are summarized below:

Input Module: (1) ZM6SWCGI - SAM X, Y, Z Start Pos, CIG Earth
(2) ZM6SW033 - X, Y, Z Eyept Pos, CIG Earth
- SAM Launch, T/F

Output Module: (1) ZM6SWCGI - SAM X, Y, Z Pos, CIG Earth, extrapolations
- SAM PT, YW, RL, CIG Earth, extrapolations
- Launch, Close Hit, Hit, T/F
This subroutine computes the drive parameters for the TA4J. Position, heading, velocity commands come from flight module ZM3SM00A. The altitude is set at the EOS station. This subroutine is called by the visual module ZM6SMOVE when the flag EAFORFLT is set at the experimenter operator station (EOS). Figure G-1 shows the TA4J in a typical formation flight scene.

Figure G-1. TA4J in Formation Flight.
TA4J DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The TA4J position and attitude are defined relative to the center of gravity (C.G.) in both the flight and visual software. Therefore, no conversion is required for position and attitude values. Hence this subroutine merely passes flight position and attitude parameters directly to visual.

Pass flight position parameters to visual:

\[
\begin{align*}
AVWMX(2) &= ANCGEX \\
AVWMY(2) &= ANCGEY \\
AVWMZ(2) &= AVHEVECR
\end{align*}
\]

WHERE: \(AVWMX(2), AVWMY(2), AVWMZ(2)\) are visual X, Y, Z position of TA4J C.G., CIG Earth Axis System.

\(ANCGEX, ANCGEY\) are flight X, Y position of TA4J C.G., in CIG Earth Axis System.

\(AVHEVECR\) is the altitude of TA4J above sea level (set at EOS) in CIG Earth Axis System.

Pass flight attitude angles to visual parameters:

\[
\begin{align*}
AVWMYW(2) &= ANCVAT \\
AVWMPT(2) &= ANTHEC \\
AVWMRL(2) &= ANPHIC
\end{align*}
\]

WHERE: \(AVWMYW(2), AVWMPT(2), AVWMRL(2)\) are visual yaw, pitch, roll TA4J angles, CIG Earth Axis System.

\(ANCVAT, ANTHEC, ANPHIC\) are flight yaw, pitch, roll TA4J angles, CIG Earth Axis System.

B. Next, extrapolations are computed for TA4J position and attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of TA4J position 16.7 milliseconds from now:

\[
\begin{align*}
AVWMXD(2) &= 1.5*AVWMX(2) - .5*AVWMX1 \\
AVWMYD(2) &= 1.5*AVWMY(2) - .5*AVWMY1 \\
AVWMZD(2) &= 1.5*AVWMZ(2) - .5*AVWMZ1
\end{align*}
\]

WHERE: \(AVWMXD(2), AVWMYD(2), AVWMZD(2)\) are extrapolated X, Y, Z TA4J position, CIG Earth Axis System.

\(AVWMX(2), AVWMY(2), AVWMZ(2)\) are current X, Y, Z TA4J position, CIG Earth Axis System.
AVWMX1, AVWMMY1, AVWMZ1 are last frame's X, Y, Z TA4J position, CIG Earth Axis System.

Save the previous frame position:

AVWMX1 = AVWMX(2)
AVWMMY1 = AVWMMY(2)
AVWMZ1 = AVWMZ(2)

WHERE: AVWMX1, AVWMMY1, AVWMZ1 are last frame's TA4J X, Y, Z position, CIG Earth Axis System.
AVWMX(2), AVWMMY(2), AVWMZ(2) are current TA4J X, Y, Z position, CIG Earth Axis System.

Compute the extrapolated value of TA4J attitude 16.7 milliseconds from now:

AVWMMYWD(2) = 1.5*AVWMMYW(2) - .5*AVWMMYW1
AVWMPTD(2) = 1.5*AVWMPT(2) - .5*AVWMPT1
AVWMMRLD(2) = 1.5*AVWMMRL(2) - .5*AVWMMRL1

WHERE: AVWMMYWD(2), AVWMPTD(2), AVWMMRLD(2) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
AVWMMYW(2), AVWMPT(2), AVWMMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis.
AVWMMYW1, AVWMPT1, AVWMMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

AVWMMYW1 = AVWMMYW(2)
AVWMPT1 = AVWMPT(2)
AVWMMRL1 = AVWMMRL(2)

WHERE: AVWMMYW1, AVWMPT1, AVWMMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
AVWMMYW(2), AVWMPT(2), AVWMMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis System.

C. The input, output module, and the parameters used in this subroutine are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request
(2) ZM3SNOOA - Generates Flt Input Par's

Output Module: (2) ZM6SWCGI - TA4J X, Y, Z Pos CIG Earth, extrapolations
- TA4J PT, YW, RL Ang, CIG Earth, extrapolations

109/110
This subroutine computes the drive parameters for the tank. The road coordinates are determined based on the CIG database loaded and then the appropriate drive parameters are computed to make the tank move along that specified road. Figure H-1 shows the tank in a typical air-to-ground database scene.

![Figure H-1. T-62 Tank.]

**TANK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION**

A. The tank driver algorithm for the gunnery range is illustrated in Figure H-2. The tank drives back and forth on the roadway which is south of the diamond figure on the gunnery range. This roadway runs north and south between the diamond and arrow head figures on the gunnery range.

The tank position and attitude are set up for the initializer for the roadway shown in Figure H-2:
Figure H-2. Tank Route on Gunnery Range.

IF (IVWAES.NE.8) GO TO 90  ...Exit, if not Gunnery Range
XSTART3=145920.         .. Start Tank on South End of Roadway
YSTART3=-5502.
ZSTART3=0.0
XSTOP3=145920.           .. Stop Tank on North End of Roadway
YSTOP3=2010.
ZSTOP3=0.0
YWSTART3=0.0             .. Head Tank North
PTSTART3=0.
RLSTART3=0.
YWSTOP3=180.             .. Head Tank South
PTSTOP3=0.
RLSTOP3=0.
WHERE: IVWAES is the visual database identification parameter (8 for Gunnery Range).
XSTART, YSTART, ZSTART are X, Y, Z tank starting coordinates, CIG Earth Axis System.
XSTOP, YSTOP, ZSTOP are X, Y, Z tank stopping coordinates, CIG Earth Axis System.
YWSTART, PTSTART, RLSTART are X, Y, Z tank yaw, pitch, roll starting angles, CIG Earth Axis System.
YWSTOP, PTSTOP, RLSTOP are X, Y, Z tank yaw, pitch, roll return angles, CIG Earth Axis System.

Tank driver control flag is set up so tank is initialized at starting position only once for a given initial condition:

IF(LVTNKST) GO TO 10  .. Keep Tank Going if Already Moving
LVTNKST=.TRUE.  .. Reset Tank Control Flag For Next Pass

WHERE: LVTNKST is set true to keep driving tank.
Is set false to restart tank at starting position.

Initialize tank starting position and attitude:
X=XSTART3  .. Get Starting X, Y, Z Position For Tank
Y=YSTART3
Z=ZSTART3
ROL=RLSTART3  .. Get Starting Yaw, Pitch, Roll Angle for Tank
PTH=PTSTART3
YAW=YWSTART3
10 CONTINUE  .. Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current X, Y, Z tank position in CIG Earth Axis.
XSTART, YSTART, ZSTART are starting X, Y, Z tank position in CIG Earth Axis System.
YAW, PTH, ROL are current tank yaw, pitch, roll angles in CIG Earth Axis System.
RLSTART3, PTSTART3, YWSTART3 are starting yaw, pitch, roll angles in CIG Earth Axis System.

Determine if tank should be driven to the North or South:

IF((Y.LT.YSTART3).OR.(Y.LT.YSTOP3.AND.YD.GT.0.)) GO TO 20 (North)
GO TO 30 (South)

Drive tank to the North Building.
B. The tank driver algorithm for the twin town database is illustrated in Figure H-3. The tank drives around the rectangular roadway path shown in Figure H-3. The tank starts at the small town and makes a series of left hand turns at each intersection until this driver is terminated.

The tank position and attitude are set up for the initializer per the road coordinates indicated in Figure H-3:
Figure H-3. Tank Route on Twin Towns.

90 CONTINUE 
   .. Entry point for twin towns
   IF (IVWAES.NE.16) GO TO 190 
      .. Exit, if not twin towns
XSTART4=17950. 
   .. Southeast Roadway Intersection (Small Town)
YSTART4=128390. 
   (Tank Always Starts Here)
ZSTART4=0.
YWSTART4=0.0 
   .. Head Tank Towards Large Town (North)
PTSTART4=0.0 
   RLSTART4=0.0
X1=17950. 
   .. Northeast Roadway Intersection (Large Small)
Y1=150390. 
   (Tank Will Make a Left Hand Turn Here)
Z1=0.
YAW1=-90.0 
   .. Head Tank Left (West) to Northwest 
   Roadway Intersection
PTH1=0.
ROL1=0.
X2=-5980. 
   .. Northwest Roadway Intersection
Y2=150390. 
   Z2=0.
YAW2=180.0 
   .. Head Tank South
PTH2=0.
ROL2=0.

115
NAVTRAEOIPCEN IH-335

X3=-5983.  .. Southwest Roadway Intersection
Y3=128390.
Z3=0.0

YAW3=+90.0  .. Head Tank North to Small Town
PTH3=0.
ROL3=0.

WHERE:

XSTART4, YSTART4, ZSTART4 are X, Y, Z starting coordinates, CIG Earth Axis System.
YWSTART4, PTSTART4, RLSTART4 are X, Y, Z starting yaw, pitch, roll angles, CIG Earth Axis.
X1, Y1, Z1 are X, Y, Z coordinates of Northeast intersection, (Large Town), CIG Earth Axis System.
YAW1, PTH1, ROL1 are yaw, pitch, roll angles to make tank turn left (to West), CIG Earth Axis System.
X2, Y2, Z2 are X, Y, Z coordinates of Northwest intersection, CIG Earth Axis System.
YAW2, PTH2, ROL2 are yaw, pitch, roll angles to make tank turn left (to South), CIG Earth Axis System.
X3, Y3, Z3 are X, Y, Z coordinates of Southwest intersection, CIG Earth Axis System.
YAW3, PTH3, ROL3 are yaw, pitch, roll angles to make tank turn left (to East), CIG Earth Axis System.

Tank driver control flag is set up so tank is initialized at starting position only once for a given initial condition:

IF(LVTNKST2) GO TO 100  .. Keep Tank Going if Already in Motion
LVTNKST2=.TRUE.  .. Reset Tank Control Flag for Next Pass

WHERE:

LVTNKST is set to true to keep driving tank.
Is set to false to restart tank at starting position.

X=XSTART4  .. Get Starting X, Y, Z Position For Tank
Y=YSTART4
Z=ZSTART4

YAW=YWSTART4  .. Get Starting Yaw, Pitch, Roll Angle for Tank
PTH=PTSTART4
ROL=RLSTART4

100 CONTINUE  .. Entry Point if Initialization Bypassed

WHERE:

X, Y, Z are current tank position in CIG Earth Axis System.
XSTART4, YSTART4, ZSTART4 are starting X, Y, Z tank position in CIG Earth Axis System.
YAW, PTH, ROL are current tank yaw, pitch, roll angles in CIG Earth Axis System.
YWSTART4, PTSTART4, RLSTART4 are starting yaw, pitch, roll angles in CIG Earth Axis System.
Compute tank route on roadway:

IF(X.GE.XSTART4.AND.Y.GE.YSTART4.AND.Y.LE.Y1) GO TO 120       .. Smal Twn
IF(Y.GE.Y1.AND.X.LE.X1.AND.X.GE.X2)       GO TO 130       .. Lg Twn
IF(X.LE.X2.AND.Y.GT.Y3)       GO TO 140       .. NW Inters.
GO TO 150       .. SW Inters.

Drive tank North to Northnorth intersection (large town):

120 CONTINUE
             .. Entry Point to Drive Tank North
          Y=Y+AVWTNKV*AQDELT2
             .. Drive Tank North to Large Town
          X=XSTART4
             .. Put Tank on Center of Roadway
          YAW=YAWSTART4
             .. Head Tank North
          GO TO 160
             .. Output Tank Parameters to CIG

WHERE:       Y is present tank Y position, CIG Earth Axis.
AVWTNKV is tank velocity, Ft/33 milliseconds.
X is present tank X position, CIG Earth Axis System.
AQDELT2 is visual CPU integration constant.
XSTART is tank X starting position, CIG Earth Axis.
YAW is present tank yaw, CIG Earth Axis.
YAWSTART is tank yaw starting angle, CIG Earth Axis.

Drive tank South to Northwest intersection:

130 CONTINUE
             .. Entry Point to Drive Tank South
          X=X-AVWTNKV*AQDELT2
             .. Drive Tank South
          Y=Y1
             .. Put Tank on Center of Roadway
          YAW=YAW1
             .. Head Tank South
          GO TO 160
             .. Output Tank Parameters to CIG

WHERE:       X is current tank X position, CIG Earth Axis System.
AVWTNKV is tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
Y is current tank Y position, CIG Earth Axis System.
Y1 is center of roadway between small town and large town.
YAW is current tank yaw angle, CIG Earth Axis System.
YAW1 is -90 degrees (West).

Drive tank South to Southwest intersection:

140 CONTINUE
             .. Entry Point to Drive Tank South
          Y=Y-AVWTNKV*AQDELT2
             .. Drive Tank South
          X=X2
             .. Put Tank on Center of Roadway
          YAW=YAW2
             .. Head Tank South
          GO TO 160
             .. Output Tank Parameters to CIG

WHERE:       Y is current tank Y position, CIG Earth Axis.
AVWTNKV is tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
NAVTRAEEIPCEN IH-335

X is current tank X position, CIG Earth Axis.
X2 is center of roadway between NW and SW corners.
YAW is current tank yaw angle, CIG Earth Axis.
YAW2 is 180 degrees (South heading).

Drive tank North to small town (SE corner):

150 CONTINUE
X=X+AVWTNKV*AQDELT2
Y=Y3
YAW=YAW3
160 CONTINUE
GO TO 5000

WHERE: X is current tank X position, CIG Earth Axis.
AVWTNKV is tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
Y is current tank Y position, CIG Earth Axis.
Y3 is center of roadway between SW corner and small town.
YAW is current tank heading, CIG Earth Axis System.
YAW3 is 90 degrees (East heading).

C. The tank driver algorithm for the river valley database is illustrated in Figure H-4. The tank drives back and forth on the South bridge, beginning on the West side of the bridge. This bridge is 400 feet above sea level (and the river) and is shown in Figure H-4.

Figure H-4. Tank Route in River Valley.
The tank position and attitude are set up for the initializer per
the road coordinates indicated in Figure H-4:

190 CONTINUE
   XSTART5=71460.0
   YSTART5=128295.0
   ZSTART5=500.0

   YWSTART5=+90.0
   PTSTART5=0.0
   RLSTART5=0.0

   XSTOP5=74460.0
   YSTOP5=128295.0
   ZSTOP5=500.0

   YSTOP5=-90.0
   PTSTOP5=0.0
   RLSTOP5=0.0

Tank driver control flag is set up so tank is initialized at
starting position only once for a given initial condition:

   IF(LVTNKST3) GO TO 210
   LVTNKST3=.TRUE.

WHERE: LVTNKST is set true to keep driving tank.
       Is set false to restart tank at starting position.

Initialize tank starting point and attitude:

   X=XSTART5
   Y=YSTART5
   Z=ZSTART5

   YAW=YWSTART5
   PTH=PTSTART5
   ROL=RLSTART5
   GO TO 220

WHERE: X, Y, Z are current tank X, Y, Z position in CIG Earth Axis.

XSTART5, YSTART5, ZSTART5 X, Y, Z position on South side of bridge,
CIG Earth Axis System.

YAW, PTH, ROL are current tank yaw, pitch, roll angles in CIG Earth
Axis System.

YWSTART5, PTSTART5, RLSTART5 are tank yaw, pitch, roll angles, CIG
Earth Axis.
Determine if tank should be driven East or West:

IF((X.LT.XSTART5).OR.(X.LT.XSTOP5.AND.XD.GT.0.)) GO TO 220 (East)
GO TO 230 (West)

Drive tank East to East side of bridge:

220 CONTINUE
.. Entry Point to Drive Tank East
XP=X .. Save Previous Position
X=X+AVWTNKV*AQDELT2 .. Drive Tank East
YAW=YWSTART5 .. Head Tank East
XD=X-XP .. Compute VEL Sign
GO TO 5000 .. Output Tank Parameters to CIG

WHERE:
XP is previous tank X position, CIG Earth Axis.
X is current tank X position, CIG Earth Axis.
AVWTNKV is tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is current tank heading, CIG Earth Axis System.
YWSTART5 is 90 degrees (heading East).
XD is current position delta (velocity sign).

Drive Tank West to West Side of Bridge:

230 CONTINUE .. Entry Point to Drive Tank West
XP=X .. Save Previous Position
X=X-AVWTNKV*AQDELT2 .. Drive Tank West
YAW=YWSTOP5 .. Head Tank West
XD=X-XP .. Compute VEL Sign

WHERE:
XP is previous tank X position, CIG Earth Axis.
X is current tank X position, CIG Earth Axis.
AVWTNKV is current tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is current tank heading, CIG Earth Axis System.
YWSTOP5 is -90 degrees (West heading).
XD is current position delta (velocity sign).

5000 CONTINUE .. Entry point to CIG data output

D. Next, extrapolations are computed for tank position attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

First, pass current tank position to visual (CIG):

AVWMX(3)=X
AVWMY(3)=Y
AVWMZ(3)=Z
Compute the extrapolated value of tank position 16.7 milliseconds from now:

\[
\begin{align*}
AVWMXD(3) &= 1.5*AVWMX(3) - 0.5*AVWMX1 \\
AVWMYD(3) &= 1.5*AVWMY(3) - 0.5*AVWMY1 \\
AVWMZD(3) &= 1.5*AVWMZ(3) - 0.5*AVWMZ1 \\
\end{align*}
\]

WHERE:
- AVWMXD(3), AVWMYD(3), AVWMZD(3) are extrapolated X, Y, Z tank position, CIG Earth Axis System.
- AVWMX(3), AVWMY(3), AVWMZ(3) are current X, Y, Z tank position, CIG Earth Axis System.
- AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z tank position, CIG Earth Axis System.

Save the previous frame position

\[
\begin{align*}
AVWMX1 &= AVWMX(3) \\
AVWMY1 &= AVWMY(3) \\
AVWMZ1 &= AVWMZ(3) \\
\end{align*}
\]

WHERE:
- AVWMX1, AVWMY1, AVWMZ1 are last frame's tank X, Y, Z position, CIG Earth Axis System.
- AVWMX(3), AVWMY(3), AVWMZ(3) are current tank X, Y, Z position, CIG Earth Axis System.

First, pass current tank attitude to visual (CIG):

\[
\begin{align*}
AVWMYW(3) &= YAW \\
AVWMPT(3) &= PTH \\
AVWMRL(3) &= ROL \\
\end{align*}
\]

Compute the extrapolated value of tank attitude 16.7 milliseconds from now:

\[
\begin{align*}
AVWMYWD(3) &= 1.5*AVWMYW(3) - 0.5*AVWMYW1 \\
AVWMPTD(3) &= 1.5*AVWMPT(3) - 0.5*AVWMPT1 \\
AVWMRLD(3) &= 1.5*AVWMRL(3) - 0.5*AVWMRL1 \\
\end{align*}
\]

WHERE:
- AVWMYWD(3), AVWMPTD(3), AVWMRLD(3) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
- AVWMYW(3), AVWMPT(3), AVWMRL(3) are current yaw, pitch, roll angles, CIG Earth Axis.
- AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

\[
\begin{align*}
AVWMYW1 &= AVWMYW(3) \\
AVWMPT1 &= AVWMPT(3) \\
AVWMRL1 &= AVWMRL(3) \\
\end{align*}
\]
WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
AVWMYW(3), AVWMPT(3), AVWMRL(3) are current yaw, pitch, roll angles, CIG Earth Axis System.

E. The input module, output module, and input/output parameters are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request.
- Sets Gun Range, Twin Town, River Valley Flags
- Resets Tank Start Point Flag

Output Module: (3) ZM6SWCGI - Tank X, Y, Z Pos CIG Earth, Extrapolations
- Tank Pt, Yw, Rl Ang, CIG Earth, Extrapolations
TRUCK DRIVER

This subroutine computes the drive parameters for the truck convoy. The road coordinates are determined based on the CIG database loaded and then the appropriate drive parameters are computed to make the trucks move along that specified road. Figure I-1 shows the truck convoy in a typical air-to-ground database scene.

Figure I-1. Truck Convoy.

TRUCK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The truck driver algorithm for the gunnery range is illustrated in Figure I-2. The trucks drive back and forth on the roadway which is north of the gunnery range. This road runs east and west between the two points indicated in Figure I-2.
The truck position and attitude are set up for the initializer for the roadway shown in Figure I-2:

```
IF (IVWAES.NE.8) GO TO 90 .. Exit, if not Gunnery Range
XSTART3=139938. .. Start Truck on West End of Roadway
YSTART3=5969.
ZSTART3=0.0

XSTOP3=151919. .. Stop Truck on East End of Roadway
YSTOP3=5969.
ZSTOP3=0.0

YWSTART3=90.0 .. Head Truck East
PTSTART3=0.
RLSTART3=0.

YWSTOP3=-90. .. Head Truck West
PTSTOP3=0.
RLSTOP3=0.
```

WHERE: XSTART, YSTART, ZSTART are X, Y, Z truck starting coordinates, CIG Earth Axis System.
XSTOP, YSTOP, ZSTOP are X, Y, Z truck stopping coordinates, CIG Earth Axis System.
YWSTART, PTSTART, RLSTART are X, Y, Z truck yaw, pitch, roll starting angles, CIG Earth Axis.
YWSTOP, PTSTOP, RLSTOP are X, Y, Z truck yaw, pitch, roll return angles, CIG Earth Axis System.

Truck driver control flag is set up so truck is initialized at starting position only once for a given initial condition:

IF(LVTRKST) GO TO 10  .. Keep Truck Going if Already Moving
LVTRKST=.TRUE.  .. Reset Truck Control Flag For Next Pass

WHERE: LVTRKST is set true to keep driving truck.
Is set false to restart truck at starting position.

Initialize truck starting position and attitude:

X=XSTART3  .. Get Starting X, Y, Z Position For Truck
Y=YSTART3
Z=ZSTART3

ROL=RLSTART3  .. Get Starting Yaw, Pitch, Roll Angle For
PTH=PTSTART3
YAW=YWSTART3

10 CONTINUE  .. Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current X, Y, Z truck position in CIG Earth Axis.
XSTART, YSTART, ZSTART are starting X, Y, Z truck position in CIG Earth Axis System.
YAW, PTH, ROL are current truck yaw, pitch, roll angles in CIG Earth Axis System.
RLSTART3, PTSTART3, YWSTART3 are starting yaw, pitch, roll angles in CIG Earth Axis System.

Determine if truck should be driven to the East or West:

IF((X.LT.XSTART3).OR.(X.LT.XSTOP3.AND.XD.GT.0.)) GO TO 20  (East)
GO TO 30  (West)

Drive truck to the East Building.

20 CONTINUE  .. Entry Point to Drive Truck East
XP=X  .. Save Previous Position
X=X+AVWTRKV*AQDELT2  .. Drive Truck to East
YAW=YWSTART3  .. Head Truck East
XD=X-XP  .. Compute VEL Sign
GO TO 40  .. Output Truck Parameters to CIG
Drive truck to the West Building.

30 CONTINUE
XP=X
X=X-AVWTRKV*AQDELT2
YAW=YWSTOP3
XD=X-XP
40 CONTINUE
GO TO 240

WHERE: XP is previous X position, CIG Earth Axis.
X is current X position, CIG Earth Axis.
AVWTRKV is truck velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is truck heading, CIG Earth Axis.
YWSTOP3 is yaw value for truck to head West.
XD is current minus previous X position (velocity sign).

8. The truck driver algorithm for the twin town database is illustrated in Figure I-3. The truck drives around the rectangular roadway path shown in Figure I-3. The truck starts at the small town and makes a series of left hand turns at each intersection until this driver is terminated.
The truck position and attitude are set up for the initializer per the road coordinates indicated in Figure I-3:

90 CONTINUE  .. Entry point for Twin Towns
   IF (IVWAES.NE.16) GO TO 190  .. Exit, if not Twin Towns
   XSTART4=17950  .. Southeast Roadway Intersection (Small Town)
   YSTART4=128390  (Truck Always Starts Here)
   ZSTART4=0.
   YWSTART4=0.0  .. Head Truck Towards Large Town (North)
   PTSTART4=0.0
   RLSTART4=0.0

   X1=17950.  .. Northeast Roadway Intersection (Large Small)
   Y1=150390  (Truck Will Make a Left Hand Turn Here)
   Z1=0.
   YAW1=-90.0  .. Head Truck Left (West) to Northwest
   PTH1=0.
   ROL1=0.

   X2=-5980.  .. Northwest Roadway Intersection
   Y2=150390.
   Z2=0.
   YAW2=180.0  .. Head Truck South
   PTH2=0.
   ROL2=0.

   X3=-5980.  .. Southwest Roadway Intersection
   Y3=128390.
   Z3=0.0
   YAW3=+90.0  .. Head Truck East to Small Town
   PTH3=0.
   ROL3=0.

WHERE:  XSTART4, YSTART4, ZSTART4 are X, Y, Z starting coordinates, CIG Earth Axis System.
   YWSTART4, PTSTART4, RLSTART4 are X, Y, Z starting yaw, pitch, roll angles, CIG Earth Axis System.
   X1, Y1, Z1 are X, Y, Z coordinates of Northeast intersection, (Large Town), CIG Earth Axis System.
   YAW1, PTH1, ROL1 are yaw, pitch, roll angles to make truck turn left (to West), CIG Earth Axis System.
   X2, Y2, Z2 are X, Y, Z coordinates of Northwest intersection, CIG Earth Axis System.
   YAW2, PTH2, ROL2 are yaw, pitch, roll angles to make truck turn left (to South), CIG Earth Axis System.

127
X3, Y3, Z3 are X, Y, Z coordinates of Southwest intersection, CIG Earth Axis System.

YAW, PTH3, ROL3 are yaw, pitch, roll angles to make truck turn left (to East), CIG Earth Axis System.

Truck driver control flag is set up so truck is initialized at starting position only once for a given initial condition:

```
IF(LVTRKST2) GO TO 100 .. Keep Truck Going if Already in Motion
LVTRKST2=.TRUE. .. Reset Truck Control Flag For Next Pass
```

WHERE: LVTRKST is set to true to keep driving truck.
Is set to false to restart truck at starting position.

```
X=XSTART4 .. Get Starting X, Y, Z Position For Truck
Y=YSTART4
Z=ZSTART4

YAW=YWSTART4 .. Get Starting Yaw, Pitch, Roll Angle For Truck
PTH=PTSTART4
ROL=RLSTART4
```

100 CONTINUE Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current truck position in CIG Earth Axis System.
XSTART4, YSTART4, ZSTART4 are starting X, Y, Z truck position in CIG Earth Axis System.
YAW, PTH, ROL are current truck yaw, pitch, roll angles in CIG Earth Axis System.
YWSTART4, PTSTART4, RLSTART4 are starting yaw, pitch, roll angles in CIG Earth Axis System.

Compute truck route on roadway:

```
IF(X.GE.XSTART4.AND.Y.GE.YSTART4.AND.Y.LE.Y1) GO TO 120 .. Smal Twn
IF(Y.GE.Y1.AND.X.LE.X1.AND.X.GE.X2) GO TO 130 .. Lg Twn
IF(X.LE.X2.AND.Y.GT.Y3) GO TO 140 .. NW Inters
GO TO 150 .. SW Inters
```

Drive truck North to Northeast intersection (large town):

```
120 CONTINUE .. Entry Point to Drive Truck North
Y=Y+AVWTRKV*AQDELT2 .. Drive Truck North to Large Town
X=XSTART4
YAW=YWSTART4 .. Put Truck on Center of Roadway
GO TO 160 .. Head Truck North
```

WHERE: Y is present truck Y position, CIG Earth Axis.
AVWTRKV is truck velocity, Ft/33 milliseconds.
X is present truck X position, CIG Earth Axis System.
AQDELT2 is visual CPU integration constant.
XSTART is truck X starting position, CIG Earth Axis.
YAW is present truck yaw, CIG Earth Axis.
YRSTART is truck yaw starting angle, CIG Earth Axis.

Drive truck West to Northwest intersection:

130 CONTINUE .. Entry Point to Drive Truck West
   X=X-AVWTRKV*AQDELT2 .. Drive Truck West
   Y=Y1 .. Put Truck on Center of Roadway
   YAW=YAW1 .. Head Truck West
   GO TO 160 .. Output Truck Parameters to CIG

WHERE: X is current truck X position, CIG Earth Axis System.
   AVWTRKV is truck velocity, Ft/33 milliseconds.
   AQDELT2 is visual CPU integration constant.
   Y is current truck Y position, CIG Earth Axis System.
   Y1 is center of roadway between small town and large town.
   YAW is current truck yaw angle, CIG Earth Axis System.
   YAW1 is -90 degrees (West).

Drive truck South to Southwest intersection:

140 CONTINUE .. Entry Point to Drive Truck South
   Y=Y-AVWTRKV*AQDELT2 .. Drive Truck South
   X=X2 .. Put Truck on Center of Roadway
   YAW=YAW2 .. Head Truck South
   GO TO 160 .. Output Truck Parameters to CIG

WHERE: Y is current truck Y position, CIG Earth Axis System.
   AVWTRKV is truck velocity, Ft/33 milliseconds.
   AQDELT2 is visual CPU integration constant.
   X is current truck X position, CIG Earth Axis System.
   X2 is center of roadway between NW and SW corners.
   YAW is current truck yaw angle, CIG Earth Axis System.
   YAW2 is 180 degrees (South heading).

Drive truck East to small town (SE corner):

150 CONTINUE .. Entry Point to Drive Truck East
   X=X-AVWTRKV*AQDELT2 .. Drive Truck East
   Y=Y3 .. Keep Y on Road
   YAW=YAW3 .. Head Truck East
   160 CONTINUE .. Output Truck Parameters to CIG
   GO TO 240

WHERE: X is current truck X position, CIG Earth Axis System.
   AVWTRKV is truck velocity, Ft/33 milliseconds.
   AQDELT2 is visual CPU integration constant.
   Y is current truck Y position, CIG Earth Axis System.
Y3 is center of roadway between SW corner and small town. YAW is current truck heading, CIG Earth Axis System. YAW3 is 90 degrees (East heading).

WHERE: X is current truck X position, CIG Earth Axis.
AVWTRKV is truck velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
Y is current truck Y position, CIG Earth Axis.
Y3 is center of roadway between SW corner and small town. YAW is current truck heading, CIG Earth Axis System. YAW3 is 90 degrees (East heading).

C. The truck driver algorithm for the river valley database is illustrated in Figure 1-4. The truck drives back and forth on the North bridge, beginning on the West side of the bridge. This bridge is 400 feet above sea level (and the river) and is shown in Figure 1-4.

Figure 1-4. Truck Convoy in River Valley.

The truck position and attitude are set up for the initializer per the road coordinates indicated in Figure 1-4:
190 CONTINUE               .. Entry point for River Valley
   XSTART5=71460.          .. Start Truck on West Side of Bridge
   YSTART5=150390.         ZSTART5=400.
   YWSTART5=+90.0          .. Head Truck East to East Side of Bridge
   PTSTART5=0.            .. Stop Truck on East Side of Bridge
   RLSTART5=0.
   XSTOP5=74460.          .. Head Truck West to West Side of Bridge
   YSTOP5=150390.         ZSTOP5=400.
   YWSTOP5=-90.0
   PTSTOP5=0.
   RLSTOP5=0.

Truck driver control flag is set up so truck is initialized at starting position only once for a given initial condition:

   IF(LVTRKST3) GO TO 210  .. Keep Going if Truck Already Moving
   LVTRKST3=.TRUE.         .. Reset Truck Control Flag For Next Pass

WHERE: LVTRKST is set true to keep driving truck.
       Is set false to restart truck at starting position.

Initialize truck starting point and attitude:

   X=XSTART5               .. Get Starting X, Y, Z Position For Truck
   Y=YSTART5
   Z=ZSTART5
   YAW=YWSTART5            .. Get Starting Yaw, Pitch, Roll Angle For Truck
   PTH=PTSTART5
   ROL=RLSTART5
   GO TO 220               .. Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current truck X, Y, Z position in CIG Earth Axis.
       XSTART5, YSTART5, ZSTART5 X, Y, Z position on West side bridge, CIG Earth Axis System.
       YAW, PTH, ROL are current truck yaw, pitch, roll angles in CIG Earth Axis System.
       YWSTART5, PTSTART5, RLSTART5 are truck yaw, pitch, roll angles, CIG Earth Axis.

Determine if truck should be driven East or West:

   IF((X.LT.XSTART5).OR.(X.LT.XSTOP5.AND.XD.GT.0.)) GO TO 220 (East)
   GO TO 230 (West)
Drive truck East to East side of bridge:

220 CONTINUE  
.. Entry Point to Drive Truck East
XP=X  
.. Save Previous Position
X=X+AVWTRKV*AQDELT2  
.. Drive Truck East
YAW=YWSTART5  
.. Head Truck East
XD=X-XP  
.. Compute VEL Sign
GO TO 240  
.. Output Truck Parameters to CIG

WHERE:  
XP is previous truck X position, CIG Earth Axis.
X is current truck X position, CIG Earth Axis.
AVWTRKV is truck velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is current truck heading, CIG Earth Axis System.
YWSTART5 is 90 degrees (heading East).
XD is current position delta (velocity sign).

Drive truck West to West side of bridge:

230 CONTINUE  
.. Entry Point to Drive Truck West
XP=X  
.. Save Previous Position
X=X-AVWTRKV*AQDELT2  
.. Drive Truck West
YAW=YWSTOP5  
.. Head Truck West
XD=X-XP  
.. Compute VEL Sign
240 CONTINUE  
.. Entry point to CIG data output

WHERE:  
XP is previous truck X position, CIG Earth Axis.
X is current truck X position, CIG Earth Axis.
AVWTRKV is current truck velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is current truck heading, CIG Earth Axis System.
YWSTOP5 is -90 degrees (West heading).
XD is current position delta (velocity sign).

D. Next, extrapolations are computed for truck position attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of truck position 16.7 milliseconds from now:

First, pass truck position to visual (CIG):

AVWWMX(4)=X
AVWMMY(4)=Y
AVWWMZ(4)=Z
AVWMXD(4) = 1.5*AVWMX(4) - .5*AVWMX1
AVWMYD(4) = 1.5*AVWMY(4) - .5*AVWMY1
AVWMZD(4) = 1.5*AVWMZ(4) - .5*AVWMZ1

WHERE: AVWMXD(4), AVWMYD(4), AVWMZD(4) are extrapolated X, Y, Z truck position, CIG Earth Axis System.
AVWMX(4), AVWMY(4), AVWMZ(4) are current X, Y, Z truck position, CIG Earth Axis System.
AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z truck position, CIG Earth Axis System.

Save the previous frame position.

AVWMX1 = AVWMX(4)
AVWMY1 = AVWMY(4)
AVWMZ1 = AVWMZ(4)

WHERE: AVWMX1, AVWMY1, AVWMZ1 are last frame's truck X, Y, Z position, CIG Earth Axis System.
AVWMX(4), AVWMY(4), AVWMZ(4) are current truck X, Y, Z position, CIG Earth Axis System.

First, pass truck attitudes to visual (CIG):

AVWMYW(4) = YAW
AVWMPT(4) = PTH
AVWMRL(4) = ROL

Compute the extrapolated value of truck attitude 16.7 milliseconds from now:

AVWMYW(4) = 1.5*AVWMYW(4) - .5*AVWMYW1
AVWMPT(4) = 1.5*AVWMPT(4) - .5*AVWMPT1
AVWMRL(4) = 1.5*AVWMRL(4) - .5*AVWMRL1

WHERE: AVWMYW(4), AVWMPT(4), AVWMRL(4) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW(4), AVWMPT(4), AVWMRL(4) are current yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

AVWMYW1 = AVWMYW(4)
AVWMPT1 = AVWMPT(4)
AVWMRL1 = AVWMRL(4)
NAVTRAEOIPCEN IH-335

WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
AVWMYW(4), AVWMPT(4), AVWMRL(4) are current yaw, pitch, roll angles, CIG Earth Axis System.

E. The input module, output module, and input/output parameters are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request
- Sets Gun Range, Twin Town, River Valley Flags
- Resets Truck Start Pt Flag

Output Module: (4) ZM6SWCGI - Truck X, Y, Z Pos CIG Earth, Extrapolations
- Truck PT, YW, RL Ang, CIG Earth, Extrapolations
Distribution List

Naval Training Equipment Center
Orlando, FL 32813
Commander
HQ, TRADOC
Attn: ATTNG-PA
Ft. Monroe, VA 23651

Center Library
Naval Personnel Research and Development Center
San Diego, CA 92152

Defense Technical Information Center
Cameron Station
Alexandria, VA 22314

Library
Division of Public Documents
Government Printing Office
Washington, D.C. 20402

Director
Training Analysis & Evaluation Group
Department of the Navy
Orlando, FL 32813

U.S. Coast Guard (G-P-1/62)
400 Seventh Street, SW
Washington, D.C. 20590

National Aviation Facilities Experimental Center
Library
Atlantic City, NJ 08405

Dr. R.A. Cook
Flight Systems Department - Q153
RAE, Farnborough
Hants, ENGLAND

OUSDR&E (R&AT) (E&LS)
CDR Paul R. Chatelier
Room 3D129, The Pentagon
Washington, D.C. 20301

Dr. Jesse Orlansky
Science and Technology Division
Institute for Defense Analyses
1 400 Army-Navy Drive
Arlington, VA 22202

Chief of Naval Operations
OP-987H
2 Attn: Dr. R.G. Smith
Washington, D.C. 20350

Scientific Technical Information Office
12 NASA
Washington, D.C. 20546

Commanding Officer
Air Force Office of Scientific Research
Technical Library
Washington, D.C. 20301

Commander
2 Naval Air Systems Command - AIR 340F
Attn: CDR C. Hutchins
Washington, D.C. 20361

Chief
ARI Field Unit
1 P. O. Box 476
Ft. Rucker, AL 36362

Dr. J.M. Rolfe
1 Research Branch
Hq. RAF Support Command
RAF Upwood
Huntington, Cambs, ENGLAND

Chief of Naval Operations
OP-115
Attn: I. Shiff
Washington, D.C. 20350

Technical Library
1 Naval Training Equipment Center
Orlando, FL 32813

Chief of Naval Operations
OP-596
Washington, D.C. 20350

1 of 3
NAVTRAEEQPCEN IH-335

Distribution List (Cont'd)

Commander
Naval Air Test Center, CT 176
Patuxent River, MD 20670

Office of Deputy Chief of Naval Operations
Manpower, Personnel and Training (OP-01)
Washington, D.C. 20350

Mr. Ken Staples
Flight Research Division
Royal Aircraft Establishment
Bedform MK41 6AE, ENGLAND

Dr. Martin Tolcott
Office of Naval Research
800 N. Quincy Street
Department of the Navy
Arlington, VA 22217

Commander
Naval Air Development Center
Attn: Technical Library
Warminster, PA 18974

Naval Research Laboratory
Attn: Library
Washington, D.C. 20375

Chief of Naval Education and Training Liaison Office
AFHRL/OTLN
Williams AFB, AZ 85224

Office of Naval Material
MAT OBD2, CP5, Room 678
Attn: R. Fisher
Washington, D.C. 20360

Commander Officer
Naval Education Training Program and Development Center
Attn: Technical Library
Pensacola, FL 32509

Commander
Pacific Missile Test Center
Point Mugu, CA 93042

Commander
Naval Air System Command, AIR 413
Washington, D.C. 20361

1 Dr. Thomas Longridge
AFHRL/OTR
Williams AFB, AZ 85224

1 Commander
Naval Air Systems Command, AIR-950D
Technical Library
Washington, D.C. 20361

Dr. David C. Nagel

1 LM-239-3
NASA Ames Research Center
Moffett Field, CA 94035

Federal Aviation Administration

1 Technical Library
Bureau Research & Development
Washington, D.C. 20590

Commander
Naval Weapons Center

1 Human Factors Branch (Code 3194)
Attn: Mr. Ronald A. Erickson
China Lake, CA 93555

Lt Col Jefferson Koonce

1 USAFA/DFBL
USAFA Academy, CO 80840

CDR Joseph Funaro

6 Code 602
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18972

1 Mr. Don Gum
AFWAL/FIGD
Wright-Patterson AFB, OH 45433

Mr. James Basinger

1 ASD/YWE
Wright-Patterson AFB, OH 45433

Mr. Rowland Bowles
NASA Langley Research Center
MS 125-B
1 Hampton, VA 23365

Mr. Brian Goldiez
PM TRADE

1 DRCPM-PND-RE
Naval Training Center
Orlando, FL 32813

2 of 3
Mr. David L. Key
U.S. Army AVRADCOM, M.S. 215-1
Ames Research Center
Moffett Field, CA 94035

1 Capt. A. Ginsburg
Aviation Vision Laboratory
AFAMRL/HEA
Wright-Patterson AFB, OH 45433