DEVELOPMENT OF KU-BAND RENDEZVOUS
RADAR TRACKING AND ACQUISITION
SIMULATION PROGRAMS

FINAL REPORT ON:
CONTRACT NO. NAS 9-17501
DRL NO. T-2003
ITEM NO. 2

SUBMITTED TO:
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058
J. W. GRIFFIN - TECHNICAL MONITOR

BY:
HUGHES
AIRCRAFT COMPANY
DENVER ENGINEERING LABORATORIES
SPACE AND COMMUNICATIONS GROUP
8000 EAST MAPLEWOOD AVENUE
ENGLEWOOD, COLORADO 80911-4999

1 AUGUST 1986
## DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>NAME</th>
<th>MAIL CODE</th>
<th>NO. COPIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. W. GRIFFIN</td>
<td>EE62</td>
<td>5</td>
</tr>
<tr>
<td>CAROL COATS</td>
<td>BE61</td>
<td>1</td>
</tr>
<tr>
<td>NEW TECHNOLOGY</td>
<td>AL32</td>
<td>1</td>
</tr>
<tr>
<td>TECHNICAL LIBRARY</td>
<td>JM2</td>
<td>1</td>
</tr>
</tbody>
</table>

The delivery address for all copies identified above is:

NASA Lyndon B. Johnson Space Center  
Attn: Mail Code/Name  
Houston, Texas 77058  
Contract NAS 9-17501
# TABLE OF CONTENTS

## 1.0 INTRODUCTION
1.1 Contractual Data Summary ........................................... 1-1  
1.2 Technical Summary .................................................. 1-3  
1.3 Conclusions .......................................................... 1-8  

## 2.0 SIMULATION PROGRAM UPGRADES
2.1 Historical Background ................................................. 2-2  
2.1.1 Brief Description of Original Simulation Model ............... 2-2  
2.1.2 Developments Leading to Proposed Simulation Upgrades .... 2-4  
2.2 Angle Tracking Loop Upgrades ...................................... 2-7  
2.2.1 Problem Definition .............................................. 2-7  
2.2.2 Definition of Algorithm Modifications .......................... 2-8  
2.2.2.1 Pattern Model Changes .................................... 2-8  
2.2.2.2 Digital Processing Model Changes ........................... 2-9  
2.2.3 Software Design Documentation ................................ 2-20  
2.2.4 Integration and Test Data ...................................... 2-44  
2.2.4.1 Antenna Pattern Module Tests .............................. 2-44  
2.2.4.2 Loop Filter Module Tests .................................. 2-50  
2.3 AGC Upgrades .......................................................... 2-51  
2.3.1 Problem Definition .............................................. 2-51  
2.3.2 Definition of Algorithm Modifications .......................... 2-56  
2.3.2.1 AGC Model Improvements .................................. 2-56  
2.3.2.2 RSS Model Improvements .................................... 2-60  
2.3.2.3 A/D Saturation Noise Model Improvements ................ 2-60  
2.3.3 Software Design Documentation ................................ 2-61  
2.3.4 Integration and Test Data ...................................... 2-63  
2.3.4.1 Test Definition ............................................ 2-63  
2.3.4.2 Test Results ............................................... 2-79  
2.4 Radar Processing Parameter Changes ................................ 2-80  
2.4.1 Problem Definition .............................................. 2-80  
2.4.2 Algorithm Modifications ........................................ 2-83  
2.4.3 Software Design Documentation ................................ 2-83  
2.4.4 Integration and Test Data ...................................... 2-86  
2.5 Velocity Processor Changes ........................................ 2-94  
2.5.1 Problem Definition .............................................. 2-94  
2.5.2 Algorithm Modifications ........................................ 2-97  
2.5.3 Software Design Documentation ................................ 2-97  
2.5.4 Integration and Test Data ...................................... 2-106  
2.5.4.1 Test Definition ............................................ 2-106  
2.5.4.2 Test Results ............................................... 2-106  

## 3.0 SORTE DATA ANALYSIS ................................................. 3-1  
3.1 SORTE Program Summary .............................................. 3-1  
3.1.1 Flight Trajectory and Target Selection ....................... 3-1  
3.1.2 Test Setup ........................................................ 3-6  
3.1.3 Data Acquisition and Processing ............................... 3-10  
3.1.3.1 Ku-Band Radar Data Processing ............................ 3-10  
3.1.3.2 WSMR Sensor Data Processing ................................ 3-10  
3.1.3.3 Final Data Processing ....................................... 3-12
### TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.4</td>
<td>Summary of Flight Tests</td>
<td>3-14</td>
</tr>
<tr>
<td>3.2</td>
<td>Analysis Approach and Preliminary Findings</td>
<td>3-14</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Preliminary Findings</td>
<td>3-17</td>
</tr>
<tr>
<td>3.3</td>
<td>Range Data Analysis</td>
<td></td>
</tr>
<tr>
<td>3.3.1</td>
<td>Discussion of Range Difference Data Statistics Problems</td>
<td>3-18</td>
</tr>
<tr>
<td>3.3.1.1</td>
<td>Description of Potential Error Sources</td>
<td>3-19</td>
</tr>
<tr>
<td>3.3.1.2</td>
<td>Discussion of Individual Problem Cases</td>
<td>3-20</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Discussion of Discontinuous Jumps in Range</td>
<td>3-27</td>
</tr>
<tr>
<td>3.3.2.1</td>
<td>Discussion of Jump Mechanism</td>
<td>3-31</td>
</tr>
<tr>
<td>3.4</td>
<td>Range Rate Data Analysis</td>
<td></td>
</tr>
<tr>
<td>3.4.1</td>
<td>Range Acceleration Effects</td>
<td>3-36</td>
</tr>
<tr>
<td>3.4.1.1</td>
<td>Analysis of Acceleration Effects on the Velocity Processor</td>
<td>3-37</td>
</tr>
<tr>
<td>3.4.1.2</td>
<td>Range Acceleration Effects in the SORTE Data</td>
<td>3-46</td>
</tr>
<tr>
<td>3.4.2</td>
<td>GDOP Effects</td>
<td>3-54</td>
</tr>
<tr>
<td>3.4.2.1</td>
<td>A Qualitative Description of GDOP</td>
<td>3-54</td>
</tr>
<tr>
<td>3.4.2.2</td>
<td>GDOP Analysis of SORTE Range Rate Data</td>
<td>3-57</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Target Rotation Effects</td>
<td>3-68</td>
</tr>
<tr>
<td>3.4.3.1</td>
<td>Evidence Supporting the Target Spin Theory</td>
<td>3-68</td>
</tr>
<tr>
<td>3.4.3.2</td>
<td>Modified Analysis of the Difference Range Rate Data</td>
<td>3-71</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Low SNR Effects</td>
<td>3-73</td>
</tr>
<tr>
<td>3.5</td>
<td>Roll and Pitch Angle Data Analysis</td>
<td>3-76</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Description of Angle Error Sources</td>
<td>3-79</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Discussion of SORTE Angle Difference Data Problems</td>
<td>3-83</td>
</tr>
<tr>
<td>3.5.2.1</td>
<td>Explanation of GDOP-Induced Error in Angle at Long Range</td>
<td>3-97</td>
</tr>
<tr>
<td>3.6</td>
<td>ILOS Roll and Pitch Angle Data Analysis</td>
<td>3-101</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Preliminary Analysis Results</td>
<td>3-102</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Description of Angle Rate Error Sources</td>
<td>3-105</td>
</tr>
<tr>
<td>3.6.3</td>
<td>SORTE Angle Rate Data Analysis</td>
<td>3-110</td>
</tr>
<tr>
<td>3.6.3.1</td>
<td>Detailed Analysis of the H30SKAF Flight</td>
<td>3-115</td>
</tr>
<tr>
<td>3.6.4</td>
<td>A Discussion of the Servo Experiment</td>
<td>3-124</td>
</tr>
<tr>
<td>4.0</td>
<td>PALAPA MISSION DATA ANALYSIS</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1</td>
<td>Qualitative Discussion of the Data</td>
<td>4-6</td>
</tr>
<tr>
<td>4.2</td>
<td>Some Simple Quantitative Data Analysis</td>
<td>4-9</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Reference Data Generation</td>
<td>4-10</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Data Analysis Results</td>
<td>4-16</td>
</tr>
<tr>
<td>4.3</td>
<td>Simulation Results</td>
<td>4-16</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Reference Generation</td>
<td>4-16</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Simulation Performance Against Palapa Reference</td>
<td>4-17</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Comparison with Flight Data Performance</td>
<td>4-19</td>
</tr>
<tr>
<td>5.0</td>
<td>REFERENCES</td>
<td>5-1</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>Baseline Program Listing</td>
<td>A-1</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>Final Program Listing</td>
<td>B-1</td>
</tr>
</tbody>
</table>
**TABLE OF CONTENTS (Continued)**

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Line-by-Line Listing of Changes</td>
<td>C-1</td>
</tr>
<tr>
<td>D</td>
<td>GDOP Analysis</td>
<td>D-1</td>
</tr>
<tr>
<td>E</td>
<td>Angle Transformation Analysis</td>
<td>E-1</td>
</tr>
<tr>
<td>F</td>
<td>Acceleration Effects on Velocity Discriminant</td>
<td>F-1</td>
</tr>
<tr>
<td>G</td>
<td>Summary of WSMR Ku-Band Tests</td>
<td>G-1</td>
</tr>
<tr>
<td>H</td>
<td>Addendum to SORTE Angle Rate Data Analysis</td>
<td>H-1</td>
</tr>
</tbody>
</table>
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Number</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1-1</td>
<td>Simplified Diagram of Range and Range Rate Tracking Loop</td>
<td>2-3</td>
</tr>
<tr>
<td>2.1-2</td>
<td>Simplified Diagram of Ku-Band Angle Rate and Angle Tracker</td>
<td>2-5</td>
</tr>
<tr>
<td>2.2-1</td>
<td>Sum Channel Gain Pattern</td>
<td>2-10</td>
</tr>
<tr>
<td>2.2-2</td>
<td>Azimuth Difference Channel Gain Pattern</td>
<td>2-11</td>
</tr>
<tr>
<td>2.2-3</td>
<td>Elevation Difference Channel Gain Pattern</td>
<td>2-12</td>
</tr>
<tr>
<td>2.2-4</td>
<td>Sum-to-Azimuth Difference Channel Phase Pattern</td>
<td>2-13</td>
</tr>
<tr>
<td>2.2-5</td>
<td>Sum-to-Elevation Difference Channel Phase Pattern</td>
<td>2-14</td>
</tr>
<tr>
<td>2.2-6</td>
<td>Original Angle Tracking Loop Models</td>
<td>2-15</td>
</tr>
<tr>
<td>2.2-7</td>
<td>Updated Angle Tracking Loop Models</td>
<td>2-17</td>
</tr>
<tr>
<td>2.2-8</td>
<td>Baseline Version of Subroutine Signal</td>
<td>2-22</td>
</tr>
<tr>
<td>2.2-9</td>
<td>Deliverable Version of Subroutine Signal</td>
<td>2-27</td>
</tr>
<tr>
<td>2.2-10</td>
<td>Summary of Modifications to Subroutine Signal</td>
<td>2-32</td>
</tr>
<tr>
<td>2.2-11</td>
<td>Baseline Version of Subroutine ATRACK</td>
<td>2-35</td>
</tr>
<tr>
<td>2.2-12</td>
<td>Deliverable Version of Subroutine ATRACK</td>
<td>2-38</td>
</tr>
<tr>
<td>2.2-13</td>
<td>Summary of Modifications to Subroutine ATRACK</td>
<td>2-41</td>
</tr>
<tr>
<td>2.2-14</td>
<td>Deliverable Version of Subroutine KSAT</td>
<td>2-45</td>
</tr>
<tr>
<td>2.2-15</td>
<td>Deliverable Version of Subroutine READPAT</td>
<td>2-46</td>
</tr>
<tr>
<td>2.2-16</td>
<td>Deliverable Version of Subroutine INTERP</td>
<td>2-48</td>
</tr>
<tr>
<td>2.3-1</td>
<td>AGC Profile for the Range Profile Given in Figure 2.3-2</td>
<td>2-54</td>
</tr>
<tr>
<td>2.3-2</td>
<td>Range Profile Used to Generate Profile in Figure 2.3-1</td>
<td>2-55</td>
</tr>
<tr>
<td>2.3-3</td>
<td>Simplified Diagram of the AGC Tracking Loop</td>
<td>2-57</td>
</tr>
<tr>
<td>2.3-4</td>
<td>Baseline Version of Subroutine RSS</td>
<td>2-62</td>
</tr>
<tr>
<td>2.3-5</td>
<td>Deliverable Version of Subroutine RSS</td>
<td>2-64</td>
</tr>
<tr>
<td>2.3-6</td>
<td>Summary of Modifications to Subroutine RSS</td>
<td>2-65</td>
</tr>
<tr>
<td>2.3-7</td>
<td>Baseline Version of Subroutine SATNSE</td>
<td>2-67</td>
</tr>
<tr>
<td>2.3-8</td>
<td>Deliverable Version of Subroutine SATNSE</td>
<td>2-68</td>
</tr>
<tr>
<td>2.3-9</td>
<td>Summary of Modifications to Subroutine SATNSE</td>
<td>2-69</td>
</tr>
<tr>
<td>2.3-10</td>
<td>Baseline Version of Subroutine DISCRM</td>
<td>2-70</td>
</tr>
<tr>
<td>2.3-11</td>
<td>Deliverable Version of Subroutine DISCRM</td>
<td>2-74</td>
</tr>
<tr>
<td>2.3-12</td>
<td>Summary of Modifications to Subroutine DISCRM</td>
<td>2-78</td>
</tr>
<tr>
<td>2.3-13</td>
<td>Simulated Radar Signal Strength RCS = +10 dBsm</td>
<td>2-81</td>
</tr>
<tr>
<td>2.3-14</td>
<td>Simulated Radar Signal Strength RCS = 40 dBsm</td>
<td>2-82</td>
</tr>
<tr>
<td>2.4-1</td>
<td>Hysteresis Loop for PRF Transition</td>
<td>2-84</td>
</tr>
<tr>
<td>2.4-2</td>
<td>Hysteresis Loop for Sample Rate Transition</td>
<td>2-85</td>
</tr>
<tr>
<td>2.4-3</td>
<td>Baseline Version of Subroutine CNTRLs</td>
<td>2-87</td>
</tr>
<tr>
<td>2.4-4</td>
<td>Deliverable Version of Subroutine CNTRLs</td>
<td>2-89</td>
</tr>
<tr>
<td>2.4-5</td>
<td>Summary of Modifications to Subroutine CNTRLs</td>
<td>2-92</td>
</tr>
<tr>
<td>2.5-1</td>
<td>Baseline Version of Subroutine VELPRO</td>
<td>2-98</td>
</tr>
<tr>
<td>2.5-2</td>
<td>Deliverable Version of Subroutine VELPRO</td>
<td>2-101</td>
</tr>
<tr>
<td>2.5-3</td>
<td>Summary of Modifications to Subroutine VELPRO</td>
<td>2-104</td>
</tr>
<tr>
<td>2.5-4</td>
<td>Range Profile for Scenario Number 1</td>
<td>2-107</td>
</tr>
<tr>
<td>2.5-5</td>
<td>Range Rate Profile for Scenarios Number 1 and 2</td>
<td>2-108</td>
</tr>
<tr>
<td>2.5-6</td>
<td>Range Profile for Scenario Number 2</td>
<td>2-109</td>
</tr>
<tr>
<td>2.5-7</td>
<td>Velocity Error for Scenario Number 1 Defined By Figures 2.5-4 and 2.5-5</td>
<td>2-110</td>
</tr>
<tr>
<td>2.5-8</td>
<td>Velocity Error for Scenario Number 2 Defined By Figures 2.5-5 and 2.5-6</td>
<td>2-112</td>
</tr>
<tr>
<td>Number</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.1-1</td>
<td>Range History for Shuttle-SMMS Rendezvous During Mission 41C in April 1984</td>
<td>3-2</td>
</tr>
<tr>
<td>3.1-2</td>
<td>Range Versus Range Rate Profile for Shuttle SMMS Rendezvous during Mission 41C in April 1984</td>
<td>3-2</td>
</tr>
<tr>
<td>3.1-3</td>
<td>Dual Luneburg Lens Installation on the HU-1H Helicopter</td>
<td>3-4</td>
</tr>
<tr>
<td>3.1-4</td>
<td>Illustration of the PEARL Site in Relation to the White Sands Missile Range</td>
<td>3-7</td>
</tr>
<tr>
<td>3.1-5</td>
<td>Positions of the Ku-Band Radar, the CINES, and the WSMR TMR Radars for Some Example Target Trajectories</td>
<td>3-9</td>
</tr>
<tr>
<td>3.1-6</td>
<td>Illustration of the Ku-Band Radar Data Acquisition Process for the SORTE Program at WSMR</td>
<td>3-11</td>
</tr>
<tr>
<td>3.1-7</td>
<td>Simplified Diagram of Final Processing of WSMR Sensor and Ku-Band Radar Data</td>
<td>3-15</td>
</tr>
<tr>
<td>3.1-8</td>
<td>Example of a Difference Data Profile</td>
<td>3-16</td>
</tr>
<tr>
<td>3.3-1</td>
<td>Illustration of Correlation Between Target Return Signal Strength and Range Tracker Random Error</td>
<td>3-21</td>
</tr>
<tr>
<td>3.3-2</td>
<td>BEST Data for GEM2 and GEM3</td>
<td>3-23</td>
</tr>
<tr>
<td>3.3-3</td>
<td>BEST Range Difference Data Profiles for SAT2 and SAT3</td>
<td>3-25</td>
</tr>
<tr>
<td>3.3-4</td>
<td>Illustration of Jumps in BEST Range Data</td>
<td>3-26</td>
</tr>
<tr>
<td>3.3-5</td>
<td>SAT1 Range Difference Data Profile</td>
<td>3-26</td>
</tr>
<tr>
<td>3.3-6</td>
<td>Comparison of SAT1 and SAT2 X-Y Ground Track</td>
<td>3-28</td>
</tr>
<tr>
<td>3.3-7</td>
<td>Comparison of H30SKAH and H30SKAF BEST Range Difference Data Profiles</td>
<td>3-29</td>
</tr>
<tr>
<td>3.3-8</td>
<td>Comparison of H30SKAE's BEST Range Difference Profile and BEST Y Profile</td>
<td>3-30</td>
</tr>
<tr>
<td>3.4-1</td>
<td>Illustration of Range Acceleration Error Sources in the Range Rate Signal Processing</td>
<td>3-44</td>
</tr>
<tr>
<td>3.4-2</td>
<td>Illustration of Oscillation in Range Rate Data Due to Target Rotation</td>
<td>3-49</td>
</tr>
<tr>
<td>3.4-3</td>
<td>BAL2 TMR Range Rate Difference Data After Compensating for Target Rotation Effects</td>
<td>3-50</td>
</tr>
<tr>
<td>3.4-4</td>
<td>Expanded View of the Oscillation Induced in the Ku-Band Radar Rate Due to Target Rotation</td>
<td>3-51</td>
</tr>
<tr>
<td>3.4-5</td>
<td>HEL30AF BEST Range Rate Difference Profile to be Compared with Range Acceleration Profile of Figure 3.4-6</td>
<td>3-52</td>
</tr>
<tr>
<td>3.4-6</td>
<td>HEL30AF BEST Range Acceleration Profile to be Compared with the Range Rate Difference Profile of Figure 3.4-5</td>
<td>3-53</td>
</tr>
<tr>
<td>3.4-7</td>
<td>HEL30AF Range Rate Difference Profile to be Compared with Range Acceleration Profile of Figure 3.4-6</td>
<td>3-55</td>
</tr>
<tr>
<td>3.4-8</td>
<td>Simulation Generated HEL30AF Range Rate Data Referenced to HEL30AF BEST Range Rate Data</td>
<td>3-56</td>
</tr>
<tr>
<td>3.4-9</td>
<td>Illustration of a Severe GDOP Velocity Error Situation</td>
<td>3-58</td>
</tr>
<tr>
<td>3.4-10</td>
<td>GDOP-Induced Range Rate Error Standard Deviation Profile for GEM2</td>
<td>3-61</td>
</tr>
<tr>
<td>3.4-11</td>
<td>GDOP-Induced Range Rate Error Standard Deviation Profile for BAL7</td>
<td>3-62</td>
</tr>
<tr>
<td>3.4-12</td>
<td>BAL7 TMR Range Rate Difference Profile After Compensation for Target Rotation Effects</td>
<td>3-63</td>
</tr>
<tr>
<td>3.4-13</td>
<td>GDOP-Induced Range Rate Error Standard Deviation Profile for SAT2</td>
<td>3-65</td>
</tr>
<tr>
<td>Number</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>3.4-14</td>
<td>GDOP-Induced Range Rate Error Standard Deviation Profile for H30SKAG</td>
<td>3-66</td>
</tr>
<tr>
<td>3.4-15</td>
<td>GDOP-Induced Range Rate Error Standard Deviation Profile for HEL30AF</td>
<td>3-67</td>
</tr>
<tr>
<td>3.4-16</td>
<td>GDOP-Induced Range Rate Error Standard Deviation Profile for HJ146AD</td>
<td>3-69</td>
</tr>
<tr>
<td>3.4-17</td>
<td>GDOP-Induced Range Rate Error Standard Deviation Profile for HL246AE</td>
<td>3-70</td>
</tr>
<tr>
<td>3.4-18</td>
<td>Illustration of the Phase Difference Between the Ku-Band Radar Range Rate and the TMR Range Rate</td>
<td>3-72</td>
</tr>
<tr>
<td>3.4-19</td>
<td>Flight Geometries for HJ and HL Series of Experiments Showing Orientation of Luneberg Lenses with Respect to Ku-Band Radar</td>
<td>3-75</td>
</tr>
<tr>
<td>3.4-20</td>
<td>Comparison of Radar Cross Section Profile and the Range Rate Difference Profile for HL346AD</td>
<td>3-77</td>
</tr>
<tr>
<td>3.5-1</td>
<td>Illustration of the Effect of a Change in Lower Azimuth Angle on the Roll Angle Difference Data for HEL30AF</td>
<td>3-81</td>
</tr>
<tr>
<td>3.5-2</td>
<td>Illustration of the Pitch Angle Difference Data for the BAL6 and BAL7 Profiles</td>
<td>3-84</td>
</tr>
<tr>
<td>3.5-3</td>
<td>TMR Roll and Pitch Angle Difference Data for the GEM2 Profile</td>
<td>3-85</td>
</tr>
<tr>
<td>3.5-4</td>
<td>TMR Roll and Pitch Angle Difference Data for the GEM3 Profile</td>
<td>3-86</td>
</tr>
<tr>
<td>3.5-5</td>
<td>Illustration of High Correlation Between Y-Brass Cap Coordinate and the Angle Difference Data</td>
<td>3-88</td>
</tr>
<tr>
<td>3.5-6</td>
<td>CINE Roll and Pitch Angle Difference Data for H30SKAH to be Compared With the BEST Data of Figure 3.5-5</td>
<td>3-89</td>
</tr>
<tr>
<td>3.5-7</td>
<td>BEST Roll and Pitch Angle Difference Data for HEL30AJ</td>
<td>3-91</td>
</tr>
<tr>
<td>3.5-8</td>
<td>Illustration of X-Y Ground Track for HEL30AJ to be Compared With Figure 3.5-7</td>
<td>3-92</td>
</tr>
<tr>
<td>3.5-9</td>
<td>CINE Pitch Angle Difference Data to be Compared With Figure 3.5-7</td>
<td>3-92</td>
</tr>
<tr>
<td>3.5-10</td>
<td>Illustration of Correlation Between the Roll Angle Difference Data and the BEST Altitude Profile</td>
<td>3-93</td>
</tr>
<tr>
<td>3.5-11</td>
<td>BEST and CINE Roll Angle Difference Data. The Negative-Going Glitch is Due to Angle Acceleration</td>
<td>3-95</td>
</tr>
<tr>
<td>3.5-12</td>
<td>Illustration of Correlation Between Roll Angle Difference Data and the BEST Altitude Profile</td>
<td>3-96</td>
</tr>
<tr>
<td>3.5-13</td>
<td>Illustration of Correlation Between Roll Angle Difference Data and BEST Altitude Profile</td>
<td>3-98</td>
</tr>
<tr>
<td>3.5-14</td>
<td>A Comparison of the CINE Altitude and the BEST Altitude for the HJ146AC Profile</td>
<td>3-99</td>
</tr>
<tr>
<td>3.5-15</td>
<td>Illustration of Effect of Altitude Error on Roll Angle Estimate</td>
<td>3-100</td>
</tr>
<tr>
<td>3.6-1</td>
<td>A Comparison of the Ku MDM Pitch Angle and ILOS Pitch Angle Rate Profile for H30SKAF</td>
<td>3-103</td>
</tr>
<tr>
<td>3.6-2</td>
<td>A Comparison of the Ku MDM Roll Angle and ILOS Roll Rate Profiles for H30SKAF</td>
<td>3-104</td>
</tr>
<tr>
<td>3.6-3</td>
<td>Second Order Analog Model of the Angle and Angle Rate Tracking Loops</td>
<td>3-107</td>
</tr>
<tr>
<td>3.6-4</td>
<td>Angle and Angle Rate Error Due to an Acceleration of 0.04 Degrees/Sec</td>
<td>3-109</td>
</tr>
<tr>
<td>Number</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>3.6-5</td>
<td>A Comparison of the CINE Pitch Angle Acceleration Profile and the CINE Pitch Rate Difference Data Profile for H30SKAF.....</td>
<td>3-116</td>
</tr>
<tr>
<td>3.6-6</td>
<td>A Comparison of the CINE Roll Angle Acceleration Profile and the CINE Roll Rate Difference Data Profile for H30SKAF.....</td>
<td>3-117</td>
</tr>
<tr>
<td>3.6-7</td>
<td>CINE Roll and Pitch Angle Difference Data Profile for H30SKAF........</td>
<td>3-120</td>
</tr>
<tr>
<td>3.6-8</td>
<td>A Comparison of the CINE KU Pitch Rate Difference Data and the CINE Sim Pitch Rate Difference Data for H30SKAF...........</td>
<td>3-121</td>
</tr>
<tr>
<td>3.6-9</td>
<td>A Comparison of the CINE KU Roll Rate Difference Data and the CINE Sim Roll Rate Difference Data for H30SKAF..............</td>
<td>3-122</td>
</tr>
<tr>
<td>3.6-10</td>
<td>A Comparison of the CINE Ku-Band Pitch Angle Difference Data and the CINE Sim Pitch Angle Difference Data for H30SKAF.....</td>
<td>3-123</td>
</tr>
<tr>
<td>3.6-11</td>
<td>A Comparison of the CINE Ku-Band Roll Angle Difference Data and the CINE Sim Pitch Angle Difference Data for H30SKAF.....</td>
<td>3-125</td>
</tr>
<tr>
<td>3.6-12</td>
<td>Second Order Analog Model of the Modified Angle Rate Tracking Loop. ...........................................................................</td>
<td>3-124</td>
</tr>
<tr>
<td>4.1-1</td>
<td>Ku-Band Radar Range and Range Rate Profiles for the Rendezvous with the Palapa Satellite........................................</td>
<td>4-2</td>
</tr>
<tr>
<td>4.1-2</td>
<td>Ku-Band Radar Pitch and Roll Angle Profiles for the Rendezvous with the Palapa Satellite.........................................</td>
<td>4-3</td>
</tr>
<tr>
<td>4.1-3</td>
<td>Ku-Band Radar ILOS Pitch and Roll Rate Profiles for the Rendezvous with the Palapa Satellite...................................</td>
<td>4-4</td>
</tr>
<tr>
<td>4.1-4</td>
<td>Expanded View of Roll and Pitch Rate Profiles for the Palapa Rendezvous.................................................................</td>
<td>4-7</td>
</tr>
<tr>
<td>4.1-5</td>
<td>Expanded View of Roll Rate Data for the Palapa Rendezvous Illustrating the Fine Structure of the Data..........................</td>
<td>4-8</td>
</tr>
<tr>
<td>4.2-1</td>
<td>Smoothed and Unsmoothed Ku MDM Range Data. A 13-Sample Window was Used for Smoothing................................................</td>
<td>4-11</td>
</tr>
<tr>
<td>4.2-2</td>
<td>Smoothed and Unsmoothed Ku MDM Range Rate Data. A 13-Sample Window was Used for Smoothing...........................................</td>
<td>4-12</td>
</tr>
<tr>
<td>4.2-3</td>
<td>Smoothed and Unsmoothed Ku MDM Pitch Angle Data. A 13-Sample Window was Used for Smoothing...........................................</td>
<td>4-13</td>
</tr>
<tr>
<td>4.2-4</td>
<td>Smoothed and Unsmoothed Ku MDM ILOS Roll Rate Data. A 51 Sample Window was Used for Smoothing.....................................</td>
<td>4-14</td>
</tr>
<tr>
<td>4-3-1</td>
<td>A comparison of the Ku-Band Radar and the Simulation Range Rate Difference Data for the Palapa Satellite Rendezvous........</td>
<td>4-21</td>
</tr>
<tr>
<td>4.3-2</td>
<td>A Comparison of the Ku-Band Radar and the Simulation ILOS Roll Rate Difference Data for the Palapa Satellite Rendezvous.....</td>
<td>4-22</td>
</tr>
<tr>
<td>Number</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1.1-1</td>
<td>Final Report Sections Cross Referenced to Work Breakdown Structure Items</td>
<td>1-2</td>
</tr>
<tr>
<td>1.2-1</td>
<td>Radar Measurement Error Specifications</td>
<td>1-4</td>
</tr>
<tr>
<td>1.2-2</td>
<td>Summary of SORTE Difference Data Performance and Comparison with Ku-Band Radar Specifications</td>
<td>1-6</td>
</tr>
<tr>
<td>1.2-3</td>
<td>Summary of Analysis of the Random Components of the Ku-Band Radar Data from the PALAPA Satellite Rendezvous on Mission 51A</td>
<td>1-9</td>
</tr>
<tr>
<td>1.2-4</td>
<td>Performance of the Ku-Band Radar Simulation Model Using the Smoothed PALAPA Satellite Rendezvous Radar Data of Mission 51A as the Input Trajectory</td>
<td>1-9</td>
</tr>
<tr>
<td>2.2-1</td>
<td>Angle Tracker Constants</td>
<td>2-20</td>
</tr>
<tr>
<td>2.2-2</td>
<td>$k_2$ and $k_3$ Values</td>
<td>2-20</td>
</tr>
<tr>
<td>2.2-3</td>
<td>A Comparison of the Standard Deviation of the Angle Tracking Performance for the Old and New Simulation Models</td>
<td>2-52</td>
</tr>
<tr>
<td>2.2-4</td>
<td>A Comparison of the Means of the Angle Tracking Performance for the Old and New Simulation Models</td>
<td>2-53</td>
</tr>
<tr>
<td>2.3-1</td>
<td>AGC Calculation Constants</td>
<td>2-59</td>
</tr>
<tr>
<td>2.4-1</td>
<td>New Range Interval Boundaries</td>
<td>2-86</td>
</tr>
<tr>
<td>2.4-2</td>
<td>Sample Rate Transition Hysteresis Loop Test Results</td>
<td>2-95</td>
</tr>
<tr>
<td>2.4-3</td>
<td>Definition of MPRF</td>
<td>2-95</td>
</tr>
<tr>
<td>2.4-4</td>
<td>PRF Transition Hysteresis Loop Test Results</td>
<td>2-96</td>
</tr>
<tr>
<td>3.1-1</td>
<td>Range and Range Rate Coverage by Test Run</td>
<td>3-3</td>
</tr>
<tr>
<td>3.1-2</td>
<td>Range and Range Rate Coverage by Test Run for Tethered Ballon and GEMsphere Tests</td>
<td>3-6</td>
</tr>
<tr>
<td>3.1-3</td>
<td>WSMR Radar and CINE Data Acquisition Items</td>
<td>3-12</td>
</tr>
<tr>
<td>3.1-4</td>
<td>Available WSMR Sensors for Each Test Run</td>
<td>3-13</td>
</tr>
<tr>
<td>3.3-1</td>
<td>Summary of First Cut at Range Error Data Analysis</td>
<td>3-18</td>
</tr>
<tr>
<td>3.3-2</td>
<td>Summary of Range Jump Investigation</td>
<td>3-32</td>
</tr>
<tr>
<td>3.4-1</td>
<td>Summary of Range Acceleration Effects on Range Rate</td>
<td>3-38</td>
</tr>
<tr>
<td>3.4-2</td>
<td>Test Cases Where Range Acceleration was an Apparent Problem</td>
<td>3-47</td>
</tr>
<tr>
<td>3.4-3</td>
<td>Test Cases Where GDOP Produced Significant Range Rate Error</td>
<td>3-59</td>
</tr>
<tr>
<td>3.4-4</td>
<td>Summary of the HL Series with Problems in the BEST or TMR Range Rate Difference Data</td>
<td>3-74</td>
</tr>
<tr>
<td>3.5-1</td>
<td>Summary of Initial Roll and Angle Performance Assessment</td>
<td>3-78</td>
</tr>
<tr>
<td>3.5-2</td>
<td>Categorization of Roll and Pitch Angle Failures by Flight Series</td>
<td>3-78</td>
</tr>
<tr>
<td>3.5-3</td>
<td>Summary of Angle Difference Data Failures</td>
<td>3-94</td>
</tr>
<tr>
<td>3.6-1</td>
<td>Summary of Initial ILOS Roll and Pitch Rate Performance Assessment</td>
<td>3-101</td>
</tr>
<tr>
<td>3.6-2</td>
<td>Comparison of Scaled and Unscaled Pitch Rate Difference Data Statistics</td>
<td>3-111</td>
</tr>
<tr>
<td>3.6-3</td>
<td>Comparison of Scaled and Unscaled Roll Rate Difference Data Statistics</td>
<td>3-113</td>
</tr>
<tr>
<td>4.2-1</td>
<td>Summary of Random Component Analysis of the Ku-Band Radar Data from the Palapa Satellite Rendezvous of Mission 51A</td>
<td>4-15</td>
</tr>
<tr>
<td>4.3-1</td>
<td>Performance of the Ku-Band Radar Simulation Model Using the Palapa Satellite Rendezvous of Mission 51A as the Input Trajectory</td>
<td>4-18</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

This report summarizes and documents all work performed on the development of the Ku-Band Rendezvous Radar Tracking and Acquisition Simulation Program project, NASA Contract No. NAS9-17501. Its submittal fulfills the Data Requirements List (DRL) Number T-2003 Item Number 2, and item D in the Work Breakdown Structure (WBS).

The project had four major technical objectives:

1) Improve the fidelity of the Space Shuttle Radar tracking simulation model developed under NASA contract number NAS9-15840.

2) Review and analyze the data from the Shuttle Orbiter Radar Test and Evaluation (SORTIE) program experiments performed at the White Sands Missile Range (WSMR).

3) Evaluate selected flight rendezvous radar data.

4) Evaluate problems with the Inertial Line-of-Sight (ILOS) angle rate tracker using the improved fidelity angle rate tracker simulation model.

1.1 CONTRACTUAL DATA SUMMARY

All project work, including the submission of this report, was performed in accordance with the revised schedule described in Modification Number 25, 20 Jan 86.

All items in the original work statement were completed. Table 1.1-1 below shows the relationship of the sections in this report to the work breakdown structure.

The final review, as per item C in the WBS, was held at JSC from 27 May to 30 May 86.
<table>
<thead>
<tr>
<th>WORK BREAKDOWN STRUCTURE ITEM</th>
<th>FINAL REPORT SECTIONS/TITLES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A1 - MISSION DATA REVIEW</strong></td>
<td>4. PALAPA MISSION ANALYSIS</td>
</tr>
<tr>
<td><strong>A2 - SORTE DATA REVIEW</strong></td>
<td>3. SORTE DATA ANALYSIS</td>
</tr>
<tr>
<td></td>
<td>APPENDIX D - GDOP ANALYSIS</td>
</tr>
<tr>
<td></td>
<td>APPENDIX E - ANGLE TRANSFORMS</td>
</tr>
<tr>
<td></td>
<td>APPENDIX F - RANGE ACCELERATION</td>
</tr>
<tr>
<td></td>
<td>APPENDIX G - SUMMARY OF WSMR</td>
</tr>
<tr>
<td></td>
<td>KU BAND TESTS</td>
</tr>
<tr>
<td><strong>B1- SOFTWARE MODIFICATION DEFINITION</strong></td>
<td>2.2.1 ANGLE TRACKING LOOP</td>
</tr>
<tr>
<td></td>
<td>2.3.1 AGC UPGRADES</td>
</tr>
<tr>
<td></td>
<td>2.4.1 RADAR PROCESSING</td>
</tr>
<tr>
<td></td>
<td>2.5.1 VELOCITY PROCESSOR ENHANCEMENT</td>
</tr>
<tr>
<td><strong>B2a- SOFTWARE MODIFICATION AND PROGRAMMING</strong></td>
<td>2.2.2, 3, 4 ANGLE TRACKING LOOP</td>
</tr>
<tr>
<td></td>
<td>2.3.2, 3, 4 AGC UPGRADE</td>
</tr>
<tr>
<td></td>
<td>2.4.2, 3, 4 RADAR PROCESSING</td>
</tr>
<tr>
<td></td>
<td>2.5.2, 3, 4 VELOCITY PROCESSOR ENHANCEMENT</td>
</tr>
<tr>
<td><strong>B2b- SOFTWARE MODIFICATION DOCUMENTATION</strong></td>
<td>APPENDIX A - BASELINE PROGRAM</td>
</tr>
<tr>
<td></td>
<td>APPENDIX B - FINAL PROGRAM</td>
</tr>
<tr>
<td></td>
<td>APPENDIX C - LINE-BY-LINE LISTING OF CHANGES</td>
</tr>
</tbody>
</table>
An initial evaluation of the Ku-Band tracking simulation model developed for use in the Shuttle Engineering Simulator (SES) at the Johnson Space Center (JSC) revealed that the fidelity could be improved in several modules. These included the modules associated with the angle tracker, the Automatic Gain Control (AGC) and the Radar Signal Strength (RSS) module, the velocity processor module, and the radar signal processing parameter module. Fidelity improvements have been made in all of these modules within the constraints of the original simulation model development requirements.

Improvements in the angle tracking loop model primarily consisted of the addition of high fidelity models of the antenna sum and difference patterns. These new pattern models utilize measured data which became available in mid-1983.

Changes in the velocity processor and the radar signal processing parameter modules were precipitated by changes made in the radar since 1980, when the modules were first written and tested.

Improvements in the AGC and RSS modules resulted from a more thorough development of the theory of operation of the AGC and RSS. Details of the changes to each of these modules, including test results to verify their correctness, are provided in Section 2.3 of this report.

The majority of effort and resources of this project were expended on the analysis of the test data generated by the SORTE program at WSMR. (A description of the SORTE program is provided in Reference 1.) The purpose of these tests was to use the highly accurate WSMR system of sensors to analyze the accuracy of the Space Shuttle Radar parameter estimates. The method of analysis was a multi-step procedure developed to suit the limited resources of the project. First, the radar-generated data and the WSMR-generated data were differenced. Then, the mean and standard deviation of the difference data were calculated and compared with the requirements for each radar parameter specified in Reference 2 and shown in Table 1.2-1. Those cases exceeding the specifications were analyzed in further detail to
### TABLE 1.2-1 RADAR MEASUREMENT ERROR SPECIFICATIONS

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Range</th>
<th>Mean--Error(1)--Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range (ft):</td>
<td>100 to 8K</td>
<td>80(2)</td>
</tr>
<tr>
<td></td>
<td>100 to 60K</td>
<td>80</td>
</tr>
<tr>
<td>Range rate (ft/s):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decreasing range:</td>
<td>0 to 148</td>
<td>1</td>
</tr>
<tr>
<td>Increasing range:</td>
<td>0 to 75</td>
<td>1</td>
</tr>
<tr>
<td>Pitch (deg):</td>
<td>0 to 30(3)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(mr):</td>
<td></td>
</tr>
<tr>
<td>Roll (deg):</td>
<td>0 to 30(3)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(mr):</td>
<td></td>
</tr>
<tr>
<td>Pitch rate (mr/s):</td>
<td>0 to 20(4)</td>
<td>0.14</td>
</tr>
<tr>
<td></td>
<td>(deg/s):</td>
<td>0.008</td>
</tr>
<tr>
<td>Roll rate (mr/s):</td>
<td>0 to 20(4)</td>
<td>0.14</td>
</tr>
<tr>
<td></td>
<td>(deg/s):</td>
<td>0.008</td>
</tr>
</tbody>
</table>

**NOTES:**

1. Both mean and standard deviation specifications are given as three sigma values.

2. The range error specification increases by a factor of 0.0016 (range) at distances greater than 8.2 nautical miles.

3. Pitch and Roll coverage range specifications include the spans from -30 to + 30 degrees.

4. Pitch rate and roll rate coverage range specifications include the spans from -20 to + 20 milliradians per second.
determine whether the radar data was out of specification, whether experimental errors in the reference sensor data collection process were responsible or whether a combination of both problems applied. A brief summary of the findings of that data analysis is given below.

Table 1.2-2 summarizes the results of the first pass through the data. This data indicated four major problem areas: range rate standard deviation, roll and pitch angle standard deviation, and ILOS angle rate mean and standard deviation. Extensive analysis of the range rate in the second stages of the procedure showed that the error was due to several sources. In many cases where the TMR system was the reference, the error was in the reference data. It was induced by the positioning of the sensors - an error known as Geometric Dilution of Precision (GDOP). In some cases, range rate error was caused by target rotation effects. Range acceleration-induced bias obscured the true range rate random performance in the majority of cases. The range accelerations (or decelerations) experienced in the SORTE program flight were typically much higher than those experienced in space operations, especially for ranges less than 5 nautical miles.

Analysis of the problems in the SORTE angle data revealed the principal cause to be GDOP in the TMR sensor system. A weak target return signal was a problem in some of the flights where the target was at long range. In those cases where the CINE reference system was available, the angle data error performance was demonstrated to be excellent.

An examination of the ILOS angle rate data in conjunction with the corresponding angle data showed that the angle rate data was incorrectly scaled. Further investigation has shown that the scale factor is approximately 2.0. Rescaling the data by a factor of 1/2 and differencing it with the WSMR data showed a significant improvement in the mean and standard deviation in the majority of the cases. Although in many cases the means and standard deviations were still outside the specification limits, some additional analysis demonstrated that this residual error was caused by angle acceleration. A closed-formed analysis of the second order model representing the angle rate tracking loop has shown that an angle acceleration of 0.04 degrees per second per second per second produces an asymptotic angle rate bias of 0.106
### TABLE 1.2-2  SUMMARY OF SORTE DIFFERENCE DATA PERFORMANCE AND COMPARISON WITH THE KU BAND RADAR SPECIFICATION

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
<th>Best/TMR</th>
<th>Failing %</th>
<th>Number</th>
<th>Failing %</th>
<th>Cine</th>
<th>Total%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>26.7 ft</td>
<td>3</td>
<td>4.8%</td>
<td>0</td>
<td>0%</td>
<td>4.8%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td></td>
<td>4.8%</td>
<td>0</td>
<td>0%</td>
<td>4.8%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>st.dev. range</td>
<td>4</td>
<td>6.4%</td>
<td>0</td>
<td>0%</td>
<td>6.4%</td>
<td></td>
</tr>
<tr>
<td>Range Rate</td>
<td>.333 ft/s</td>
<td>2</td>
<td>3.2%</td>
<td>2</td>
<td>3.2%</td>
<td>6.4%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td></td>
<td>3.2%</td>
<td>2</td>
<td>3.2%</td>
<td>6.4%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>st.dev. rate</td>
<td>35</td>
<td>56.4%</td>
<td>24</td>
<td>38.7%</td>
<td>95.1%</td>
<td></td>
</tr>
<tr>
<td>Roll</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>5</td>
<td>8.0%</td>
<td>1</td>
<td>1.6%</td>
<td>9.6%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>st.dev.</td>
<td>23</td>
<td>37.0%</td>
<td>4</td>
<td>6.4%</td>
<td>43.4%</td>
<td></td>
</tr>
<tr>
<td>Pitch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>8</td>
<td>12.9%</td>
<td>1</td>
<td>1.6%</td>
<td>14.5%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>st.dev.</td>
<td>11</td>
<td>17.7%</td>
<td>1</td>
<td>1.6%</td>
<td>19.3%</td>
<td></td>
</tr>
<tr>
<td>Roll rate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>33</td>
<td>53.2%</td>
<td>25</td>
<td>40.3%</td>
<td>93.5%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>st.dev.</td>
<td>36</td>
<td>58.0%</td>
<td>26</td>
<td>42.0%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>Pitch rate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>36</td>
<td>58.0%</td>
<td>26</td>
<td>42.0%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>st.dev.</td>
<td>36</td>
<td>58.0%</td>
<td>26</td>
<td>42.0%</td>
<td>100.0%</td>
<td></td>
</tr>
</tbody>
</table>

(Data was compiled from a total of 62 difference data sets.)
degrees per second in the widest bandwidth case. Examination of the angle acceleration profiles in some of the test runs has shown that 0.04 degrees per second per second accelerations were not uncommon. Accelerations of this magnitude would naturally degrade the Ku-Band Radar ILOS angle rate tracker statistics in those cases. Complete details of the angle rate data analysis are provided in Section 3.6.

There are two possible sources of a scale factor error. One source could be the processing required to transfer the data from CA LSI4/90 disk to magnetic tape to the VAX 11/780. A second source of the scale factor error could be the scaling of the ILOS roll rate and pitch rate in the microprocessor of the Electronics Assembly No. 1 (EA-1) of the Ku-Band Radar. At the writing of this report both possibilities were being investigated, but a determination of the source and the exact magnitude of the scale factor had not been completed.

Complete details of these analyses, which are quite involved and vary from experiment to experiment, are provided in Section 3 of the report. In addition, many of the anomalies found in the data, such as jumps in range and pure sine wave oscillation in range rate, are addressed in Section 3.

A limited amount of effort was applied to the area of flight data reduction. JSC and Lockheed Engineering and Management Services Company (LEMSCO) personnel provided the radar data in VAX-11/780 compatible form for the entire rendezvous of the shuttle with the Palapa 1B Satellite during mission 51A. This flight profile was used to investigate the variance of the random error found in all radar measured data and to investigate the fidelity of the simulation against a typical satellite rendezvous profile. Details of the analysis technique used to extract the variances of the random errors in the radar data are provided in Sections 4.1 and 4.2 along with a discussion of the legitimacy of the technique. Results of the analysis showed that the range, range rate, roll angle, and pitch angle random errors were within specification over the entire profile. On the other hand, ILOS roll rate and ILOS pitch rate were within specification for ranges outside 3.8 nautical miles, but were out of specification for some intervals when the range was less than 3.8 nm. These results are of no surprise to the engineers who have
already reviewed flight data for many different rendezvous. The purpose of this exercise was to quantify the characteristics of the random error. Table 1.2-3 summarizes the standard deviation of the error for each of the six parameters over three different range intervals corresponding to the three different radar tracker bandwidths.

The flight data file was also used to investigate the fidelity of the radar tracking simulation model. Details of the method employed to make this determination can be found in Section 4.3. Table 1.2-4 summarizes the results. A comparison of the simulated data and flight data revealed an excellent match in range, range rate, roll angle and pitch angle. The simulation angle rate error data did not match the flight angle rate error data very well, especially inside 3.8 nautical miles range where the wider tracker bandwidths are instituted. Based on the excellent match of the simulation when compared to the SORTE data (see Section 3.6.3), it is conjectured that the reference trajectory injected into the simulation was in error. In particular, it is felt that the heavy smoothing of the angle rate data to form a reference, erroneously removed some true shuttle-target dynamics.

1.3 CONCLUSIONS

There are two general areas where conclusions can be drawn: (1) SORTE program results and (2) simulation fidelity.

SORTE Program Results. The SORTE program can be considered highly successful for one single reason: it demonstrated the sensitivity of the ILOS angle rate tracker to angle acceleration. The analysis of this data, combined with the Palapa rendezvous data analysis, has demonstrated that the fluctuation in the angle rate meters for target ranges less than 1.9 nautical miles is due to rendezvous dynamics and/or beam wander on the target, but not thermal noise problems. The angle acceleration data helped verify the angle and angle rate tracker design parameters through equations 3-12 and 3-19.
TABLE 1.2-3 SUMMARY OF ANALYSIS OF THE RANDOM COMPONENTS OF THE KU-BAND RADAR DATA FROM THE PALAPA SATELLITE RENDEZVOUS OF MISSION 51A

<table>
<thead>
<tr>
<th>TIME INTERVAL, SEC</th>
<th>RANGE INTERVAL, FT</th>
<th>STD. MEAN</th>
<th>STD. MEAN</th>
<th>STD. MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>4855 - 5890</td>
<td>43520 - 23040</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5890 - 6530</td>
<td>23040 - 11520</td>
<td>20.45</td>
<td>10.97</td>
<td>5.3</td>
</tr>
<tr>
<td>6530 - 6993</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME INTERVAL, SEC</th>
<th>RANGE INTERVAL, FT</th>
<th>STD. MEAN</th>
<th>STD. MEAN</th>
<th>STD. MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>4855 - 5890</td>
<td>43520 - 23040</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5890 - 6530</td>
<td>23040 - 11520</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6530 - 6993</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

TABLE 1.2-4 PERFORMANCE OF THE KU-BAND RADAR SIMULATION MODEL USING THE SMOOTHED PALAPA SATELLITE RENDEZVOUS RADAR DATA OF MISSION 51A AS THE INPUT TRAJECTORY

<table>
<thead>
<tr>
<th>TIME INTERVAL, SEC</th>
<th>RANGE INTERVAL, FT</th>
<th>STD. MEAN</th>
<th>STD. MEAN</th>
<th>STD. MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>4855 - 5890</td>
<td>43520 - 23040</td>
<td>99.2</td>
<td>8.57</td>
<td>3.1</td>
</tr>
<tr>
<td>5890 - 6530</td>
<td>23040 - 11520</td>
<td>99.2</td>
<td>5.37</td>
<td>1.0</td>
</tr>
<tr>
<td>6530 - 6993</td>
<td></td>
<td>99.6</td>
<td>3.1</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME INTERVAL, SEC</th>
<th>RANGE INTERVAL, FT</th>
<th>STD. MEAN</th>
<th>STD. MEAN</th>
<th>STD. MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>4855 - 5890</td>
<td>43520 - 23040</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5890 - 6530</td>
<td>23040 - 11520</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6530 - 6993</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

1-9
Conclusions about radar parameter estimation performance are as follows. The range and angle data error performance was demonstrated to be excellent. Range rate and angle rate error performance was obscured by acceleration effects, GDOP and other assorted problems. In both cases the specifications on the random component are quite severe which makes them susceptible to bias induced by acceleration. In the case of range rate, the acceleration encountered in space operations, especially for ranges less than 5 nautical miles, will be quite small and will not present a problem. On the other hand, it is not clear just what magnitude of angle acceleration to expect in space operations.

There is one final conclusion about the SORTE program results. If any additional data analysis is to be done, then the CINE reference data should be used wherever possible. This is because TMR system data is corrupted by GDOP in many cases. This phenomenon obscures the radar parameter estimation performance in these cases.

Simulation Fidelity. Prior to the study reported herein, the SES radar simulation results agreed well with the flight data in range, range rate and angle data at all ranges. However, the simulation angle rate data performance appeared to be much better than the flight data especially for ranges less than 1.9 nautical miles. Until this study, this problem was blamed on an inaccurate model of the angle rate tracker. However, based on the SORTE angle rate data analysis of Section 3.6.3 it is clear that the problem is in either the fidelity of the rendezvous flight dynamics generation or in the radar target effects model or both. Further work must be done in this area to make an exact determination.
The purpose of this section is to document all changes to the Ku-Band radar tracking performance simulation model developed for the SES at JSC under NASA contract number NASA-15840. There were two general types of changes: (1) corrections in various parameter settings of the radar, and (2) improvements in the fidelity of the mathematical models. Both types of modifications were aimed at bringing the simulation model operation into better alignment with the actual radar operation.

The general format for documenting the modifications is as follows. First, the problem with the original simulation model is defined. Second, the changes in the algorithm are given along with the evidence supporting the model fidelity improvement. Third, the exact changes in the software are documented by providing the original module listing, the present module listing, and a listing of the difference. Last, the tests to validate the changes are defined and the results of those tests are provided. At this point it should be noted that only a limited amount of validation testing was done for each modification due to limited resources for this portion of the project. However, the testing was extensive enough so that only a handful of unusual scenarios will yield bogus results.

This section is structured as follows: Section 2.1 gives a brief history of the simulation development and some discussion of the fidelity problem areas. Section 2.2 documents the angle tracking loop changes. Section 2.3 documents the upgrade of the AGC and RSS module. Section 2.4 provides details of the radar signal processing parameters module upgrade and Section 2.5 documents the velocity processor module enhancements. In support of Section 2, Appendix A provides complete listings of the original simulation program; Appendix B contains a listing of the upgraded simulation program; and Appendix C gives a listing of the file created by differencing the original and upgraded simulation programs.
2.1 HISTORICAL BACKGROUND

The Ku-Band Rendezvous Radar performance computer simulation model was developed under contract to NASA JSC in 1979. This model was installed in the Shuttle Engineering Simulator (SES) which is a man-in-the-loop, real-time simulator. The purpose of the model was to provide for target rendezvous training of astronauts and target rendezvous optimization analysis. Complete details of the simulation development are given in References 3 and 4. In what follows, a summary description of the model will be presented along with a discussion of the shortcomings in its performance.

2.1.1 Brief Description of Original Simulation Model

The general philosophy of the simulation development was to provide as much model accuracy as possible within the constraints of real-time operation. A summary of the accuracy of the simulation model under this real-time constraint can be broken into an assessment of the accuracy of the three major components that comprise the model. These components are: (1) the range tracking loop, (2) the angle tracking loop, and, (3) the velocity processor.

Figure 2.1-1 gives a simplified diagram of the Ku-Band Radar's range tracking loop and velocity processor. Except for the analog signal processing done in the receiver, the majority of the range tracking loop is implemented in digital hardware. All of the computer run time savings and shortcuts in these two models were realized in the target return signal generation and the signal processing through the range discriminant, \( D_R \), and the velocity discriminant, \( D_V \), formation. The target was treated as a collection of point scatterers, and the receiver and signal processor (through the doppler filter output) were treated as a linear device. Hence, a closed-form solution could be used to compute the target return from a single scatterer at the doppler filter output. Then, the filter output for the collection of points could be obtained by summing individual contributions. The target was assumed to have constant range rate and
FIGURE 2.1-1 SIMPLIFIED DIAGRAM OF RANGE AND RANGE RATE TRACKING LOOP
constant position in the antenna pattern over a complete data cycle. These assumptions have little effect on model accuracy under normal operating conditions: low target range and angle acceleration. The remainder of the range tracking loop and the velocity processor are implemented in digital hardware. Models of these processors are exact and do not degrade the performance of the range tracking loop or the velocity processor. In summary, accuracy of the range tracking loop and the velocity processor module were expected to be, and have been proven to be, excellent. The only real problem in fidelity was expected in the velocity processor in the presence of target range acceleration. The error in this case is a predictable quantity as discussed in Appendix F.

Figure 2.1-2 gives a simplified diagram of the Ku-Band radar's angle and angle rate tracking loop. Generation of the angle discriminants is done in a manner that is similar to the range and velocity discriminant generation. However, the angle discriminant generation accuracy is much more sensitive to the models of the antenna sum and difference patterns employed. In the original version of the simulation, conventional mathematical models of these patterns, rather than actual measured data, were used. The remainder of the angle and angle rate tracking loop that required modeling is the loop filter which is composed of two parts: a digital section and an analog section. The digital section was modeled with high accuracy, while the analog section was modeled as a simple analog integrator. A detailed discussion justifying this representation of the analog section can be found in Reference 5. There are two general areas in this angle tracking model with potential for improvement: (1) the sum and difference antenna pattern models, and (2) the analog (servo) electronics section in the loop filter.

2.1.2 Developments Leading To Proposed Simulation Upgrades

Several events led to the set of simulation modifications developed under the present contract. What follows is a chronology of these events and their implications. The simulation model code was delivered to JSC and installed in the SES in July 1981. At about this time, the Ku-Band radar
FIGURE 2.1-2 SIMPLIFIED DIAGRAM OF KU-BAND ANGLE RATE AND ANGLE TRACKER
was beginning comprehensive system testing. As a result of this testing, several parameters in the tracking mode were changed. These included pulsewidth and PRF switch point, the transmit power switch point, and the elimination of velocity ambiguity resolution in the 7kHz PRF mode. This led to the definition of the signal processing parameter module changes described in Section 2.4 and the velocity processor upgrades given in Section 2.5.

As system testing continued through 1982 and early 1983, a very comprehensive model of the AGC and RSS was developed to calibrate the system and to help interpret test results and anomalies. This model was further refined to help in planning and evaluating the early flight tests of the radar, e.g., STS-7, STS-11, and STS-13. The model was documented in Reference 4 and is the basis of the upgrades described in Section 2.3. The first flight test of the radar on June 22, 1983, when the shuttle released and recaptured the Shuttle Pallet Satellite (SPAS), showed that the simulation was in excellent agreement with the flight data for ranges out to 1000 feet. The first rendezvous with a target occurred in April of 1984 when the shuttle rescued and repaired the Solar Maximum Mission Satellite (SMMS). The radar was used to track SMMS from a range of 110,000 feet in to 100 feet. A comparison of the flight data with simulation data over this interval of operation showed the range, range rate, roll angle, and pitch angle to be in reasonably good agreement. However, the ILOS roll and pitch angle rate data from the simulation was far better than that experienced in flight, especially for the widest tracking loop bandwidth (ranges less than 1.9 nautical miles). This was the first confirmation that there was a problem in the angle rate tracking loop model fidelity. Analysis of an antenna model upgrade had already begun in late 1983. The intent of the upgrade was to replace the closed-formed math models with highly accurate measured data which became available in mid-1983. The results of the study, completed in mid-1984 and documented in References 6 and 7, demonstrated that part of the problem in the angle rate model performance was inaccurate models of the antenna patterns.

In fact, it was conjectured that the design of angle rate tracking loop should have incorporated a more comprehensive model of the antenna patterns and that this was one source of the tracking problems inside 1.9 miles.
All of the above events motivated the upgrade of the angle tracking loop in the SES Radar Simulation model which is documented in Section 2.2. The upgraded model was then used to a limited extent to troubleshoot the poor angle rate performance found in the flight data. Results of this analysis are found in Sections 4.2.

2.2 ANGLE TRACKING LOOP UPGRADES

2.2.1 Problem Definition

As discussed in Section 2.1, the original model of the angle tracking loop had two areas of potential fidelity problems: (1) the antenna pattern models and (2) the model of the analog (servo) electronics. In the original model, the antenna patterns were represented by closed-form equations because there was insufficient antenna pattern measurement data available. The original model of the servo electronics, while simple, represented a reasonable tradeoff in model complexity. At the time, the attitude of the model developers was to use these simple models of the antenna pattern and servo electronics and compare their performance against the flight data, when it became available. The first radar flight data that became available was the shuttle rendezvous with SMMS in April of 1984. It indicated the angle rate tracker performance was noisier than expected, while the simulation showed the angle rate tracker performance to be well within the maximum noise specification. As noted in Section 2.1, an investigation of the antenna pattern fidelity effects on the angle rate tracking performance, documented in References 6 and 7, showed that the simple antenna pattern model was a significant contributor to the errors in performance estimates. The angle rate tracker modifications developed during this investigation served as the basis for the SES model upgrade documented in this section.

Prior to the project reported upon herein, the effects of a more accurate servo model had not been investigated. However, some servo model enhancements were investigated on the present project as part of a larger analysis of the angle rate tracking loop performance problems. Results of the angle rate tracking loop analysis and the potential servo model enhancements are documented in Section 4.2.
2.2.2 Definition of Algorithm Modifications

The angle tracking loop algorithm was modified in two areas: (1) the antenna patterns module and (2) digital portion of the track loop filter. Changes in the antenna patterns module were major revisions, while the changes in the digital hardware section were relatively minor.

2.2.2.1 Pattern Model Changes

The original antenna patterns were modeled by analytic equations. The sum pattern was modeled as a surface of revolution about the antenna boresight with a shape given by the expression

\[(2-1) \quad \text{sumpat} = \frac{\sin(bx)}{bx}\]

where the constant \(b\) was chosen so that the pattern model beamwidth matched the beamwidth of the measured data. The difference pattern was modeled as the derivative of the sum pattern and was given by the equation

\[(2-2) \quad \text{difpat} = \frac{a(b\cos(bx)-\sin(bx))}{(bx)}\]

The constant \(a\) is chosen to place the 100 percent pattern modulation point at the proper angle off boresight. This model of the sum and difference patterns assumed (1) an infinite null depth on boresight, and (2) the phasing between the sum and difference channel was either 0 or 180 degrees with an instantaneous phase transition on boresight.

The updated antenna pattern models use an extensive set of measured data with interpolation between data points, rather than closed-form equations. Data measurements were taken for five parameters: sum channel gain, elevation difference channel gain, sum-to-elevation difference channel phase, azimuth difference channel gain, and sum-to-azimuth difference channel phase. Data was measured on an 8 degree by 8 degree grid in azimuth and elevation with a resolution of 0.2 degrees. Data sets exist for radar transmit frequencies: 1 (13.779 GHz), 3 (13.883 GHz) and 5 (13.987 GHz). However, to conserve memory, only the data for transmit frequency 1 is used.
for all five frequency slots in the simulation. This model of the antenna patterns was first developed for the angle tracking performance investigation reported in References 6 and 7. In that case, bicubic spline interpolation was used to generate the sum pattern gain values and both sum-to-difference channel phase values, while linear interpolation was used to generate the difference channel gain values. Three dimensional plots (from Reference 6) of the resulting patterns are shown in Figure 2.2-1 through 2.2-5. These new antenna pattern models are quite accurate and provide the following important features: finite null depth on boresight and non-instantaneous phase transition through boresight. Initially, the antenna model described above was installed in the SES simulation. However, it was found that the bicubic spline interpolation was causing the simulation to run far too slowly. This violated the real-time run constraint applied to original simulation development. To improve program speed, an investigation into the use of two dimensional linear interpolation of all parameters was undertaken. This investigation surfaced two significant results: (1) changes in the angle and angle rate tracking loop performance were minimal and (2) simulation run time was significantly reduced. The reduction in run time was about an order of magnitude, although no official timing tests were performed.

2.2.2.2 Digital Processing Model Changes

These changes specifically apply to the digital hardware section of the angle tracking loop filter (see Figure 2.1-2). This includes the section of the hardware from the angle discriminant output to the input of the digital-to-analog converter (DAC) in the Electronics Assembly 1 (EA-1). The philosophy here was to change this model from an approximate representation of the digital hardware to an exact representation. The changes include: (1) performing finite bit multiplication with the exact digital constants used in the radar, (2) performing finite bit addition, (3) the addition of saturation check models at appropriate points in the system model, and (4) the addition of a DAC model that converts input bits to a voltage which is input to the gimbal motor model. For comparison, Figure 2.2-6 shows the original loop configuration, while Figure 2.2-7 gives the upgraded version of the loop. Fidelity enhancements provided by these modifications is only second order at best. However, these changes do provide very accurate data at intermediate
FIGURE 2.2-1  SUM CHANNEL GAIN PATTERN

2-10
FIGURE 2.2-2 AZIMUTH DIFFERENCE CHANNEL GAIN PATTERN

2-11
FIGURE 2.2-3  ELEVATION DIFFERENCE CHANNEL GAIN PATTERN
FIGURE 2.2-4  SUM-TO-AZIMUTH DIFFERENCE CHANNEL PHASE PATTERN

2-13
FIGURE 2.2-5 SUM-TO-ELEVATION DIFFERENCE CHANNEL PHASE PATTERN
a. $\dot{\alpha}$ angle tracking loop filter.

b. $\dot{\beta}$ angle tracking filter

FIGURE 2.2-6 ORIGINAL ANGLE TRACKING LOOP FILTER MODELS
2-15
FIGURE 2.2-7  ALPHA TRACKING LOOP MODEL
Figure 2.2-7 Beta Tracking Loop Model (continued)
points throughout the digital hardware section. For example, the alpha error voltage and beta error voltage are easily accessed test points in the actual digital hardware. The upgraded loop model can now compute similar voltage traces for direct comparison with actual data.

Figure 2.2-6 gives a block diagram of the original alpha and beta tracking loop filters. The equations describing those filters are summarized below. The first step is to update the smoothed ILOS azimuth and elevation rates using the expressions

\[
\begin{align*}
\dot{\theta}_{AZ}(n) &= \dot{\theta}_{AZ}(n-1) + T_s K_{eq} D_{AZ}(n) \\
\dot{\theta}_{EL}(n) &= \dot{\theta}_{EL}(n-1) + T_s K_{eq} D_{EL}(n)
\end{align*}
\]

where

- \( \dot{\theta}_{EL} \) = smoothed target inertial LOS elevation rate,
- \( \dot{\theta}_{AZ} \) = smoothed target inertial LOS azimuth rate,
- \( T_s \) = update interval,
- \( K_{eq} \) = loop constant
- \( D_{EL} \) = elevation channel discriminant
- \( D_{AZ} \) = azimuth channel discriminant

Next, the alpha and beta gimbal rates are updated with the equations

\[
\begin{align*}
\dot{\alpha}(n) &= (\omega_{TX}(n) + \omega_{BZ} \sin(\beta)) / \cos(\beta) - \omega_{BX} n \\
\dot{\beta}(n) &= \omega_{TY}(n) - \omega_{BY}(n)
\end{align*}
\]

where

- \( \omega_{TX}(n) \) = \( \dot{\theta}_{AZ}(n) + K_{eq} T_s D_{AZ}(n) \)
- \( \omega_{TY}(n) \) = \( \dot{\theta}_{EL}(n) + K_{eq} T_s D_{EL}(n) \)
- \( \omega_{BX}(n) \) = X-component of body inertial angular velocity at time sample \( n \) expressed in L-coordinates.

Finally, the new alpha and beta gimbal positions are computed from the expressions

\[
\begin{align*}
\alpha(n) &= \alpha(n-1) + T_s \times \dot{\alpha}(n) \\
\beta(n) &= \beta(n-1) + T_s \times \dot{\beta}(n)
\end{align*}
\]
Figure 2.2-7 gives the block diagrams for the upgraded alpha and beta angle tracking loop filter models. The equations defining this upgraded algorithm are defined as follows. The smoothed ILOS azimuth and elevation rates are given by

\begin{align*}
\hat{\theta}_{\text{AZ}}(n) &= k_6E_1(n) \\
\hat{\theta}_{\text{AZ}}(n) &= k_6A_1(n)
\end{align*}

where

\begin{align*}
E_1'(n) &= E_1(n-1) + k_3D_{\text{EL}}(n) \\
E_1(n) &= \text{SAT}(E_1'(n), 2^{15}) \\
\text{SAT}(x, y) &= \begin{cases} 
  y, & x > y \\
  x, & x < y \\
  -y, & x < -y
\end{cases}
\end{align*}

Similar expressions hold for $A_1(n)$ and $A_1'(n)$. The so-called alpha rate error ($A_3$) and beta rate error ($B_3$) voltages (at the DAC output) are given by the expression

\begin{align*}
E_3(n) &= k_4\text{SAT}(E_1(n) + k_2D_{\text{EL}}(n), 2^{15})/32 \\
A_3(n) &= k_4\text{SAT}(A_1(n) + k_2D_{\text{AZ}}(n), 2^{15})/32
\end{align*}

Then, the predicted alpha beta gimbal rates are expressed as

\begin{align*}
\hat{\alpha}(n) &= (\omega_{\text{TX}}(n) + \omega_{\text{BZ}} \sin(\beta)/\cos(\beta)) - \omega_{\text{BX}} \\
\hat{\beta}(n) &= (\omega_{\text{TY}}(n) - \omega_{\text{BY}}(n))
\end{align*}

where

\begin{align*}
\omega_{\text{TX}}(n) &= k_5A_3(n)T_5 \\
\omega_{\text{TY}}(n) &= k_5E_3(n)T_5
\end{align*}

The final step in the modified algorithm is to update the position of the alpha and beta gimbals. This step is identical to the original algorithm and is given by equation (2-5).
The constants $k_4$, $k_5$ and $k_6$ in equations (2-6) through (2-8) do not change as a function of bandwidth. Values for these constants are summarized in Table 2.2-1 below. The constants $k_2$ and $k_3$ differ for the alpha and beta tracking loops and change with angle tracker bandwidth. Values for these constants are given in Table 2.2-2.

**TABLE 2.2-1 ANGLE TRACKER CONSTANTS**

<table>
<thead>
<tr>
<th>CONSTANTS</th>
<th>VALUE</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_4$</td>
<td>0.0048876</td>
<td>volts/bit</td>
</tr>
<tr>
<td>$k_5$</td>
<td>1.18/5</td>
<td>deg/sec-bit</td>
</tr>
<tr>
<td>$k_6$</td>
<td>0.000576/16</td>
<td>deg/sec-bit</td>
</tr>
</tbody>
</table>

**TABLE 2.2-2 $k_2$ AND $k_3$ VALUES**

<table>
<thead>
<tr>
<th>PRF, kHz</th>
<th>Range, nm</th>
<th>$32k_2$</th>
<th>$32k_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha$</td>
<td>$\beta$</td>
</tr>
<tr>
<td>7</td>
<td>1.9</td>
<td>662</td>
<td>866</td>
</tr>
<tr>
<td>7</td>
<td>1.9 to 3.8</td>
<td>407</td>
<td>532</td>
</tr>
<tr>
<td>7</td>
<td>3.8 to 8.2</td>
<td>149</td>
<td>195</td>
</tr>
<tr>
<td>8.2</td>
<td>8.2</td>
<td>149</td>
<td>195</td>
</tr>
</tbody>
</table>

2.2.3 **Software Design Documentation**

The changes described in the previous subsection affected the following existing subroutines: SIGNAL and ATRACK. Changes in the sum and difference channel signal amplitude computation were incorporated into SIGNAL. The changes in the digital hardware section of the loop filter, documented by equations (2-6) through (2-8), were incorporated into ATRACK.
Some remarks about the listings which will be presented below, and throughout this section, are appropriate at this time as an aid to their interpretation. The "original" or "baseline" versions of the subroutines are those which were present in the baseline simulation program HACSIM. The "final" or "modified" versions are those which appear in the deliverable program FINSIMI. The listings of the difference between the baseline and deliverable versions of the subroutines include both those lines which were deleted from the original program, and those which were added to form the final program. The line numbers identifying the deleted lines refer to lines in the original subroutine and the line numbers which appear next to the added lines refer to lines in the final version of the subroutine.

Figure 2.2-8 is a listing of the original version of SIGNAL as it existed in the baseline program HACSIM. Figure 2.2-9 is a listing of the modified version of SIGNAL which is in the deliverable program FINSIMI. Figure 2.2-10 is a summary of the differences between the subroutines.

Figure 2.2-11 is a listing of the original version of ATRACK. Figure 2.2-12 is a listing of the modified version of ATRACK. Figure 2.2-13 is a listing of the differences between the two subroutines.

Modifications to the angle rate tracking loop required the generation of three new routines: KSAT, READPAT, and INTERP. KSAT is a generalized routine that checks for saturation of a digital signal. Inputs include the untested signal of interest and the desired saturation level. The output is the tested (and possibly modified) signal. READPAT is the subroutine that is used to read the measured antenna pattern data into the appropriate common blocks. This subroutine is executed only one time, and this is upon the first call to the subroutine INTERP. Subroutine INTERP computes the sum pattern gain, the azimuth difference pattern gain, the elevation difference pattern gain, the sum-to-elevation difference channel phase, and the sum-to-azimuth difference channel phase for a given pair of azimuth and elevation angles. As mentioned in the previous subsection, the values are computed using two dimensional linear interpolation and the measured data. The inputs to the subroutine are the azimuth and elevation angle. The data computed by the subroutine is passed back to the calling program via a labeled common block.
SUBROUTINE SIGNAL
REAL IRDOT, IRNG
COMMON /CNTL/IPWR, IMODE, ITXP, IASM, IDUMC(5), DUMC(3)
COMMON /OUTPUT/IIDUM(3), SRNG, DUM1(8), IDUM2(4)
COMMON /CNTL/IIDUMS(13), MTINT, MRNG, MSAM, MPRF, MBEKTRK, MBTSUM,
      MBT(8)
COMMON /TGTDAT/NT, RAU(3,100), RANGE(100), RADVEL(100), RO(3).
COMMON /ROU(3). CRRANGE, CVEL
COMMON /SATDAT/RADAR(3), N20, RT(70,3), SIG(70)
COMMON /RTDAT/IRDOT, IRNG, DUM2(5), MDF(5)
COMMON /SIGDAT/SPAZ, SMAZ, SPEL, SMEL, EARLY, LATE, DF1, DF5,
      DF2, DF4, SIGBAR
COMMON /XFORMS/TLB(3,3), TLBD(3,3), TLT(3,3), TLTD(3,3)
COMPLEX CSUM, CDIFAZ, CDIFEL, CEARLY, CLATE, CDF1, CDF5, CDF2, CDF4,
      CDF3
DIMENSION CTP(10,2), DF'WTS(5,10), ALAM(5), ALAMD(3), NFREQ(2)
DATA CTP/9.03318, 9.799E-4, 4.03318, 1.9599E-3, 9.8E-4, 4.9E-4,
      22, 2.45E-4, 1.225E-4/
DATA NFREQ/1,5/
DATA ALAM/177.3733, 176.0447, 178.7149, 176.7089, 178.0393/
DATA ALAMD/1.272461E-2, 2.969089E-2, 3.309023E-1/
REAL CTP

STEP 1: PRELIMINARY COMPUTATIONS AND PARAMETER INITIALIZATION

STEP 1-1: INITIALIZE DISCRIMINANT COMPONENTS (NOTE: THESE ARE THE
COMPONENT SIGNALS AFTER SQUARE-LAW DETECTION).

SPAZ=0.0
SMAZ=0.0
SPEL=0.0
SMEL=0.0
EARLY=0.0
LATE=0.0
DF1=0.0
DF5=0.0
DF2=0.0
DF4=0.0
SIGBAR=0.0
NFMAX=NFREQ(IMODE)
DO 55 I=1,NFMAX
C STEP 1-2: INITIALIZE COMPLEX DISCRIMINANT COMPONENTS BEFORE EACH XMIT FREQUENCY (NOTE: THESE ARE THE COMPONENT SIGNALS BEFORE SQUARE-LAW DETECTION).

CSUM=(0.,0.)
CDIFFAZ=(0.,0.)
CDIFFEL=(0.,0.)
CEARLY=(0.,0.)
CLATEI=(0.,0.)
CDIFF= (0.,0.)
CDO 45 K=1,NT

IF(I.GT.1) GO TO 35

C STEP 2: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR FOR KTH SCATTERER.

C STEP 2-1: COMPUTE SUM PATTERN ANGLE.

PSI=ACOS(ABS(RAU(3,K)))

C STEP 2-2: COMPUTE ANTENNA SUM PATTERN MULTIPLICATION FACTOR.

X=SPAT(PSI)

C STEP 2-3: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR.

XX=SIG(K)*X

C NOTE: IF IN ACTIVE MODE SET XX=1.0.

IF(IMODE.EQ.1) XX=1.0

C STEP 2-4: CHECK ANTENNA STEERING MODE (IF IN GPC-DES OR MANUAL — SKIP STEP 4).

CSTEP 3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION FACTORS FOR KTH SCATTERER.

C STEP 3-1: COMPUTE AZ AND EL DIFFERENCE PATTERN ANGLES.

DELAZ=ASIN(RAU(2,K))
DELEL=ASIN(RAU(1,K))

C STEP 3-2: COMPUTE AZ AND EL DIFFERENCE PATTERN MULTIPLICATION FACTORS.

Y=DPAT(DELAZ)
Z=DPAT(DELEL)

C STEP 3-3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION FACTORS (INCLUDE RCS AND SUM PATTERN WEIGHTINGS).

DAZ=XX*Y
DEL=XX*Z

C STEP 4: COMPUTE RANGE GATE WEIGHTING FOR KTH SCATTERER

C DEFINITION: CTP=4./(C*PULSEWIDTH) WHERE C IS SPEED OF LIGHT.

C STEP 4-1: COMPUTE RANGE GATE LOCATION WRT RANGE GATE CENTER.

FIGURE 2.2-8 BASELINE VERSION OF SUBROUTINE SIGNAL

PAGE 2

2-23
SRNGX=10.*INT(0.03125*IRNG)
DELX=CTP(MRNG,IMODE)*(RANGE(K)-SRNGX)

C  STEP 4-2: COMPUTE EARLY AND LATE RANGE GATE WEIGHTINGS FOR KTH SCATTERER.
        II=INT((DELX+.7)/2.)
        IF(II.LE.1) II=1
        IF(II.GE.5) II=5
        GO TO (21,22,23,24,21).II
21 RGE=0.0
       RGL=0.0
       GO TO 25
22 RGE=3.+DELX
       RGL=0.0
       GO TO 25
23 RGE=1.-DELX
       RGL=1.+DELX
       GO TO 25
24 RGE=0.0
       RGL=3.-DELX

C  STEP 4-3: COMPUTE RANGE GATE WEIGHT FOR NON-RANGE DISCRIMINANT COMPONENTS.
25 RGWGT=5.*(RGL+RGE)

C  STEP 4-4: APPLY RANGE GATE WEIGHTING TO SUM AND DIFFERENCE CHANNEL MULTIPLICATION FACTORS.
       RGE'S*RGE
       RGL'S*RGL
       S=S*RGWGT
       DAZ'=DAZ*RGWGT
       DEU=DEU*RGWGT

C  DEFINITION: ALAMO(MPRF)=2. =PI/(PRF=LAMBDA)
C  DEFINITION: THE CONSTANT 0.196348=PI/16.
C
C  STEP 5: COMPUTE DOPPLER FILTER PHASE SHIFT AND WEIGHTING FOR KTH SCATTERER. NOTE: THIS CALCULATION IS INDEPENDENT OF XMIT FREQUENCY AND ASSUMES NO ACCELERATION OVER DATA CYCLE.
       DEFINITION: RANGE(K) IS RANGE OF KTH SCATTERER TO ANTENNA PHASE CENTER.
       DEFINITION: ALAM(K)=PI/LAMBDA WHERE LAMBDA IS XMIT FREQUENCY.

C  STEP 5-2: COMPUTE DOPPLER FREQUENCY CORRESPONDING TO RADIAL VELOCITY OF KTH SCATTERER.
       FDT=2.*ALAMO(MPRF)*RADVEL(K)

C  STEP 5-2: COMPUTE DOPPLER FREQUENCY CORRESPONDING TO RADIAL VELOCITY OF KTH SCATTERER.

C  STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER TRACKING FILTERS.
       DO 30 J=1,5
             ARG=0.196348+MDF(J)-FDT
             DFMTS(J,K)=DOPFIL(ARG)
       30

C  STEP 6: COMPUTE PHASE FACTOR ASSOCIATED WITH KTH SCATTERER RANGE OF TARGET C.G.
       (NOTE: PHASE IS REFERENCED TO PHASE ASSOCIATED WITH RANGE OF TARGET C.G.)

C  DEFINITION: RANGE(K) IS RANGE OF KTH SCATTERER TO ANTENNA PHASE CENTER.

C  STEP 6-1: COMPUTE PHASE REFERENCED TO TARGET C.G.
       DELPSI=ALAM(I)*(RANGE(K)-CGRNGE)

FIGURE 2.2-8 BASELINE VERSION OF SUBROUTINE SIGNAL

PAGE 3
C STEP 6-2: COMPUTE PHASE FACTOR, I.E. \( \text{EXP}((j \cdot \text{DELPHI})). \)
PHASE = \( \text{EXP}((j \cdot \text{DELPSI})) \)
PHASE1 = PHASE

C STEP 6-3: COMBINE RANGE PHASE FACTOR AND DOPPLER FILTER \( \times 3 \)
WEIGHT AND PHASE FACTOR.
PHASE = PHASE + DFWTS(3, K)

C STEP 7-1: ADD KTH SCATTERER CONTRIBUTION TO SUM CHANNEL SIGNAL.
CSUM = CSUM + S * PHASE

C STEP 7-2: ADD KTH SCATTERER CONTRIBUTION TO SUM CHANNEL SIGNAL.
CSUM = CSUM + S * PHASE

C STEP 7-3: ADD KTH SCATTERER CONTRIBUTION TO AZ AND EL DIFFERENCE SIGNALS.
CDIFAZ = CDIFAZ + DAZ * PHASE1
CDIFEL = CDIFEL + DEL * PHASE1

C STEP 7-4: ADD KTH SCATTERER CONTRIBUTION TO RANGE DISCRIMINANT COMPONENT SIGNALS.
CEARLY = CEARLY + RG1 * PHASE1
CLATE = CLATE + RG1 * PHASE1

C STEP 7-5: ADD KTH SCATTERER CONTRIBUTION TO VELOCITY DISCRIMINANT COMPONENT SIGNALS.
PHASE1 = PHASE1 * S
CDF2 = CDF2 + PHASE1 * DFWTS(2, K)
CDF4 = CDF4 + PHASE1 * DFWTS(4, K)

C STEP 7-6: ADD KTH SCATTERER CONTRIBUTION TO ON-TARGET DISCRIMINANT COMPONENT SIGNALS.
CDF1 = CDF1 + PHASE1 * DFWTS(1, K)
CDF5 = CDF5 + PHASE1 * DFWTS(5, K)

45 CONTINUE

C STEP 8-1: CHECK ANTENNA STEERING MODE — SKIP STEPS 8-2 AND 8-3 IF IN GPC-DES OR MANUAL MODE.
IF (IASM.EQ.2.0 .OR. IASM.EQ.4) GO TO 50

C STEP 8-2: COMPUTE AZ DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
SPA2 = SPA2 + CABS(CSUM + CDIFAZ)**2
SMA2 = SMA2 + CABS(CSUM - CDIFAZ)**2

C STEP 8-3: COMPUTE EL DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
SPEL = SPEL + CABS(CSUM + CDIFEL)**2
SMEL = SMEL + CABS(CSUM - CDIFEL)**2

C STEP 8-4: COMPUTE RANGE DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
EARLY = EARLY + CABS(CEARLY)**2
LATE = LATE + CABS(CLATE)**2

40 CEARLY = CEARLY + RG1 * PHASE1
CLATE = CLATE + RG1 * PHASE1

50 CONTINUE

C STEP 8-5: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET DISCRIMINANT COMPONENTS AT ITH FREQUENCY AND SQUARE-LAW DETECT THESE COMPONENTS.

FIGURE 2.2-8 BASELINE VERSION OF SUBROUTINE SIGNAL
PAGE 4
C STEP 8-5: COMPUTE VELOCITY DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
DF2=DF2+CABS(CDF2)**2
DF4=DF4+CABS(CDF4)**2
C STEP 8-6: COMPUTE ON-TARGET DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
DF1=DF1+CABS(CDF1)**2
DF5=DF5+CABS(CDF5)**2
C
C * STEP 9: COMPUTE EFFECTIVE CROSS-SECTION AVERAGED OVER PROPER NUMBER OF TRANSMIT FREQUENCIES.
C
C SIGBAR=SIGBAR+CABS(CSUM)**2
55 CONTINUE
SIGBAR=SIGBAR/FLOAT(NFREQ(IWODE))
C
C NOTE: DEBUGGING PRINT STATEMENTS
C WRITE(6,980) (I,SIG(I),I=1,NT)
980 FORMAT(' I,SIG =',18,F14.4)
C WRITE(6,982) NT,DAZ,DEL,RGE,RGL,RGWGT,MDF(3)
982 FORMAT(' NT,DAZ,DEL,RGE,RGL,RGWGT,F3 =',15,6F10.2,15)
C WRITE(6,981) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),
C 2 DFWTS(5,1)
981 FORMAT(' DFWTS =',10F12.4)
C 901 FORMAT(' DF WTS =',10F12.4)
C 902 FORMAT(' DFWTS =',10F12.4)
C 902 FORMAT(' DFWTS =',10F12.4)
C
C 901 FORMAT(' DF WTS =',10F12.4)
C 902 FORMAT(' DFWTS =',10F12.4)
C 902 FORMAT(' DFWTS =',10F12.4)
C
C 901 FORMAT(' DF WTS =',10F12.4)
C
SUBROUTINE SIGNAL
REAL IDOT, IRNG
COMMON /CNTL/IPWR, IMODE, ITYP, IASM, IDUMC(5), DUMC(3)
COMMON /OUTPUT/ IDUM(3), SRNG, DUM1(6), IDUM2(4)
COMMON /ICNTL/ IDUMS(13), MTKINT, MRNG, MPRF, MBKTRK, MBTSUM,
               MBT(8)
COMMON /TGTDAT/ NT, RUA(3, 100), RANGE(100), RADVEL(100), RO(3),
               ROU(3), CGRNGE, CGVEL
COMMON /SATDAT/ RADAR(3), N20, RT(70, 3), SIG(70)
COMMON /RTDAT/ IRDOT, IRNG, DUM2(5), MDF(5)
COMMON /SIGDAT/ SPAZ, SMAZ, SPEL, SMEL, EARLY, LATE, DF1, DF5,
               DF2, DF4, SSBAR
COMMON /XFORMS/ TLB(3, 3), TLBD(3, 3), TLT(3, 3), TLTD(3, 3)
COMMON /SUDIPH/ X, Y, Z, PAZ, PEL
COMPLEX CSUM, CDIFAZ, CDIFEL, CEARLY, CLATE, CDF1, CDF5, CDF2, CDF4,
               CDF3, CDF6
DIMENSION CTP(10, 2), DFWTS(5, 100), ALAM(5), ALAMD(3), NFREQ(2)
DATA CTP/ 9.799E-4, 2.819E-4, 2.199E-4, 2.9599E-4, 2.339E-4, 2.5799E-4,
          2.199E-4, 2.9599E-4, 2.339E-4, 2.5799E-4/
DATA NFREQ/1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100/
DATA ALAM/ 177.3733, 176.3447, 178.3474, 179.3097, 180.3097/
DATA ALAMD/ 1.299E-2, 2.999E-2, 3.999E-2, 4.999E-2, 5.999E-2/
REAL LATE
COMPLEX DAZ, DEL
DATA ILOOP/ 1/

---------------------------------------------------------------------

MODIFIED JAN 10 1986 BY M. MEYER
MODIFICATIONS TO SUBROUTINE SIGNAL INCLUDE
CALCULATION OF THE AZIMUTH AND ELEVATION ANGLES
USE OF MEASURED ANTENNA PATTERNS INSTEAD
OF FUNCTIONS SPAT AND DPAT AND A
FACTOR IN THE DIFFERENCE CHANNELS SIGNAL
WHICH ACCOUNTS FOR
THE FINITE WIDTH PHASE
TRANSITION IN THE REAL PHASE PATTERNS.
---------------------------------------------------------------------

• STEP 0: READ IN ANTENNA PATTERNS AND SET PHASE BALANCE •

IF (ILOOP .NE. 1) GO TO 11

---------------------------------------------------------------------

PRECEEDING PAGE BLANK NOT FILMED

FIGURE 2.2-9 DELIVERABLE VERSION OF SUBROUTINE SIGNAL
PAGE 1

2-27
CALL READPAT
PBAL=0.
ILOOP=0
CONTINUE

* STEP 1: PRELIMINARY COMPUTATIONS AND PARAMETER INITIALIZATION *

SPAR=0.0
SMR=0.0
SPR=0.0
SML=0.0
EARLY=0.0
LATE=0.0
DF1=0.0
DF2=0.0
DF3=0.0
DF4=0.0
SIGBAR=0.0
NFMAX=NFREQ(IMODE)
DO 55 I=1,NFMAX

* STEP 1-2: INITIALIZE COMPLEX DISCRIMINANT COMPONENTS BEFORE EACH XMIT FREQUENCY (NOTE: THESE ARE THE COMPONENT SIGNALS BEFORE SQUARE-LAW DETECTION).

CSUM=(0.,0.)
CIFAX=(0.,0.)
CIFEL=(0.,0.)
CEARLY=(0.,0.)
CLATE=(0.,0.)
CDIF1=(0.,0.)
CDIF2=(0.,0.)
CDIF4=(0.,0.)
DO 45 K=1,NT

IF(I.GT.1) GO TO 35

* STEP 2: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR FOR KTH SCATTERER *

AZ=ATAN2D(RAU(2,K),ABS(RAU(3,K)))
EL=ATAN2D(RAU(1,K),ABS(RAU(3,K)))

* STEP 2-2: COMPUTE ANTENNA SUM, DIFFERENCE AND PHASE FACTORS CALL INTERP(AZ,EL)

XX=SIG(K)*X

* STEP 2-3: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR.

NOTE: IF IN ACTIVE MODE SET XX=1.0.

IF(IMODE.EQ.1) XX=1.0
S=XX*X

* STEP 2-4: CHECK ANTENNA STEERING MODE (IF IN GPC-DES OR MANUAL SKIP STEP 4).

IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20

C

FIGURE 2.2-9 DELIVERABLE VERSION OF SUBROUTINE SIGNAL PAGE 2
STEP 3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION FACTORS FOR KTH SCATTERER.

STEP 3-3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION FACTORS (INCLUDE RCS AND SUM PATTERN WEIGHTINGS).

**AZ** = \(XX \times \text{COMPLEX} (\cos (\text{PAZ} + \text{PBAL}), \sin (\text{PAZ} + \text{PBAL}))\)

**EL** = \(XX \times \text{COMPLEX} (\cos (\text{PEL} + \text{PBAL}), \sin (\text{PEL} + \text{PBAL}))\)

STEP 4: COMPUTE RANGE GATE WEIGHTING FOR KTH SCATTERER

**DEFINITION**: \(\text{CTP} = \frac{4}{(C \times \text{PULSEWIDTH})}\) WHERE C IS SPEED OF LIGHT.

STEP 4-1: COMPUTE RANGE GATE LOCATION WRT RANGE GATE CENTER.

\(\text{SRNGX} = \text{MRNG} \times \frac{1}{\text{IMODE}} \times (\text{RANGE} - \text{SRNGX})\)

STEP 4-2: COMPUTE EARLY AND LATE RANGE GATE WEIGHTINGS FOR KTH SCATTERER.

**RGE** = 1.0E-4

**RGL** = 1.0E-4

GO TO 25

**RGE** = 3.0E3 + DELX

**RGL** = 0.0

GO TO 25

**RGE** = 1.0E3 - DELX

**RGL** = 1.0E3

GO TO 25

**RGE** = 0.0

**RGL** = 3.0E3 - DELX

STEP 4-3: COMPUTE RANGE GATE WEIGHT FOR NON-RANGE DISCRIMINANT COMPONENTS.

**RGWGT** = 0.5 \times (RGL + RGE)

STEP 4-4: APPLY RANGE GATE WEIGHTING TO SUM AND DIFFERENCE CHANNEL MULTIPLICATION FACTORS.

**RGWGT** \times \text{AZ} \times \text{RGWGT}

**RGWGT** \times \text{DEL} \times \text{RGWGT}

STEP 5: COMPUTE DOPPLER FILTER PHASE SHIFT AND WEIGHTING FOR KTH SCATTERER. NOTE: THIS CALCULATION IS INDEPENDENT OF XMIT FREQUENCY AND ASSUMES NO ACCELERATION OVER DATA CYCLE.

**DEFINITION**: \(\text{ALAMD} = 2 \times \pi / (\text{PRF} \times \text{LAMBD})\)

**DEFINITION**: THE CONSTANT \(0.196348 = \pi / 16\).

**DFT** = \(2 \times \text{ALAMD} \times \text{RADVEL(K)}\)

**FIGURE 2.2-9** DELIVERABLE VERSION OF SUBROUTINE SIGNAL

PAGE 3
STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER TRACKING FILTERS.

DO 30 J=1,5
ARG=0.196348*MDF(J)-FDT
DFWTS(J,K)=DOPFIL(ARG)
30

STEP 6: COMPUTE PHASE FACTOR ASSOCIATED WITH KTH SCATTERER RANGE

DEFINITION: RANGE(K) IS RANGE OF KTH SCATTERER TO ANTENNA PHASE CENTER.
DEFINITION: ALAM=4.*PI/LAMBDA WHERE LAMBDA IS XMIT FREQUENCY.

STEP 6-1: COMPUTE PHASE REFERENCED TO TARGET C.G.
DELPHI=ALAM(I)*(RANGE(K)-CGRNGE)
PHASE=EXP(CMPLX(0.,DELPHI))
PHASE1=PHASE

STEP 6-2: COMPUTE PHASE FACTOR, I.E., EXP(J*DELPHI).
PHASE=PHASE*DFWTS(3,K)

STEP 6-3: COMBINE RANGE PHASE FACTOR AND DOPPLER FILTER WEIGHT AND PHASE FACTOR.
PHASE1=PHASE1,DFWTS(3,K)

STEP 7: ADD (VECTORIALLY) KTH SCATTERER CONTRIBUTION TO EACH DISCRIMINANT'S COMPONENT SIGNALS.

STEP 7-1: ADD KTH SCATTERER CONTRIBUTION TO SUM CHANNEL SIGNAL.
CSUM=CSUM+S*PHASE

STEP 7-2: CHECK ANTENNA STEERING MODE — SKIP STEP 8-3 IF IN GPC-DES OR MANUAL MODE.
IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 40

STEP 7-3: ADD KTH SCATTERER CONTRIBUTION TO AZ AND EL DIFFERENCE CHANNELS SIGNALS.
CDIFAZ=CDIFAZ+DAZ*PHASE
CDIFEL=CDIFEL+DEL*PHASE

STEP 7-4: ADD KTH SCATTERER CONTRIBUTION TO RANGE DISCRIMINANT COMPONENT SIGNALS.
CEARLY=CEARLY+RGE*PHASE
CLATE=CLATE+RGL*PHASE

STEP 7-5: ADD KTH SCATTERER CONTRIBUTION TO VELOCITY DISCRIMINANT COMPONENT SIGNALS.
PHASE1=PHASE1+S
CDF2=CDF2+PHASE1*DFWTS(2,K)
CDF4=CDF4+PHASE1*DFWTS(4,K)

STEP 7-6: ADD KTH SCATTERER CONTRIBUTION TO ON-TARGET DISCRIMINANT COMPONENT SIGNALS.
CDF1=CDF1+PHASE1*DFWTS(1,K)
CDF5=CDF5+PHASE1*DFWTS(5,K)
45 CONTINUE

STEP 8: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET DISCRIMINANT COMPONENTS AT ITH FREQUENCY AND SQUARE.

FIGURE 2.2-9 DELIVERABLE VERSION OF SUBROUTINE SIGNAL
C LAW DETECT THESE COMPONENTS.

C

C STEP 8-1: CHECK ANTENNA STEERING MODE — SKIP STEPS 9-2 AND 9-3
C IF IN GPC-DES OR MANUAL.
C IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 50

C STEP 8-2: COMPUTE AZ DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
C SPAZ=SPAZ+CABS(CSUM+CDIFAZ)**2
C SMAZ=SMAZ+CABS(CSUM-CDIFAZ)**2

C STEP 8-3: COMPUTE EL DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
C SPEL=SPEL+CABS(CSUM+CDIFEL)**2
C SMEL=SMEL+CABS(CSUM-CDIFEL)**2

C STEP 8-4: COMPUTE RANGE DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
C 50 EARLY=EARLY+CABS(CEARLY)**2
C LATE=LATE+CABS(CLATE)**2

C STEP 8-5: COMPUTE VELOCITY DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
C DF2=DF2+CABS(CDF2)**2
C DF4=DF4+CABS(CDF4)**2

C STEP 8-6: COMPUTE ON-TARGET DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
C DF1=DF1+CABS(CDF1)**2
C DF5=DF5+CABS(CDF5)**2

C

C STEP 9: COMPUTE EFFECTIVE CROSS-SECTION AVERAGED OVER PROPER NUMBER OF TRANSMIT FREQUENCIES.
C
C SIGBAR=SIGBAR+CABS(CSUM)**2
C SIGBAR=SIGBAR/FLOAT(NFREQ(IMODE))

C NOTE: DEBUGGING PRINT STATEMENTS
C WRITE(6,900) (I,SIG(I),I=1,NT)
C WRITE(6,902) NT,S.DAZ,DEL,RGE,RGL,RGWGT,MDF(3)
C WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1),
C 2 DFWTS(5,1)
C 900 FORMAT(' I,SIG=',18,F14.4)
C 901 FORMAT( ' NT,S.DAZ,DEL,RGE,RGL,RGWGT,F3 = ',15,6F10.2,15)
C 902 FORMAT( ' NT,S.DAZ,DEL,RGE,RGL,RGWGT,F3 = ',15,6F10.2,15)

C END
C

FIGURE 2.2-9 DELIVERABLE VERSION OF SUBROUTINE SIGNAL

PAGE 5

2-31
LINES DELETED FROM BASELINE PROGRAM
35  COMPLEX CSUM, CDIFAZ, CDIFEL, CEARLY, CLATE, CDF1, CDF5, CDF2, CDF4, 00020230

LINES ADDED TO DELIVERABLE PROGRAM
35  COMMON /SUDIPH/ X.Y.Z, PAZ, PEL
36  COMPLEX CSUM, CDIFAZ, CDIFEL, CEARLY, CLATE, CDF1, CDF5, CDF2, CDF4, 00020230

LINES DELETED FROM BASELINE PROGRAM
43  C

LINES ADDED TO DELIVERABLE PROGRAM
44  COMMON DAZ, DEL
45  DATA ILOOP/1/
46  C
47  C
48  C
49  C MODIFIED JAN 10 1986 BY M. MEYER
50  C MODIFICATIONS TO SUBROUTINE SIGNAL INCLUDE
51  C CALCULATION OF THE AZIMUTH AND ELEVATION ANGLES
52  C USE OF MEASURED ANTENNA PATTERNS INSTEAD
53  C OF FUNCTIONS SPAT AND DPAT AND A
54  C FACTOR IN THE DIFFERENCE CHANNELS SIGNAL
55  C WHICH ACCOUNTS FOR THE FINITE WIDTH PHASE
56  C TRANSITION IN THE REAL PHASE PATTERNS.
57  C
58  C
59  C
60  C
61  C • STEP 0: READ IN ANTENNA PATTERNS AND SET PHASE BALANCE •
62  C
63  C
64  IF (ILOOP.NE.1) GO TO 11
65      CALL READPAT
66      PBAL=0.
67      ILOOP=0
68      11 CONTINUE
69  C 00020320

LINES DELETED FROM BASELINE PROGRAM
86  C STEP 2-1: COMPUTE SUM PATTERN ANGLE.
87      PSI=ACOS(ABS(RAU(3,K))) 00020750
88  C
89  C STEP 2-2: COMPUTE ANTENNA SUM PATTERN MULTIPLICATION FACTOR.
90      X=SPAT(PSI) 00020760
91  C

LINES ADDED TO DELIVERABLE PROGRAM

PREVIOUS PAGE BLANK NOT FILMED

FIGURE 2.2-10 SUMMARY OF MODIFICATION TO SUBROUTINE SIGNAL
PAGE 1

2-32
C STEP 2-1: COMPUTE AZIMUTH AND ELEVATION ANGLE.
AZ = ATAN2D(RAU(2,K), ABS(RAU(3,K)))
EL = ATAN2D(RAU(1,K), ABS(RAU(3,K)))
C STEP 2-2: COMPUTE ANTENNA SUM, DIFFERENCE AND PHASE FACTORS
CALL INTERP(AZ,EL)

C STEP 3-1: COMPUTE AZ AND EL DIFFERENCE PATTERN ANGLES.
DELAZ = -ASIN(RAU(2,K))
DELEL = -ASIN(RAU(1,K))
C STEP 3-2: COMPUTE AZ AND EL DIFFERENCE PATTERN MULTIPLICATION FACTORS.
Y = DPAT(DELAZ)
Z = DPAT(DELEL)

C STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER

LINES DELETED FROM BASELINE PROGRAM
107 C STEP 3-1: COMPUTE AZ AND EL DIFFERENCE PATTERN ANGLES.
108 DELAZ = -ASIN(RAU(2,K))
109 DELEL = -ASIN(RAU(1,K))
110 C STEP 3-2: COMPUTE AZ AND EL DIFFERENCE PATTERN MULTIPLICATION FACTORS.
111 Y = DPAT(DELAZ)
112 Z = DPAT(DELEL)

LINES ADDED TO DELIVERABLE PROGRAM
133 C

LINES DELETED FROM BASELINE PROGRAM
118 DAZ = XX*Y
119 DEL = XX*Z

LINES ADDED TO DELIVERABLE PROGRAM
136 C

LINES DELETED FROM BASELINE PROGRAM
138 21 RGE = 0.0
139 RGL = 0.0
140 GO TO 25

LINES ADDED TO DELIVERABLE PROGRAM
157 21 RGE = 1.0E-4
158 RGL = 1.0E-4
159 GO TO 25

LINES DELETED FROM BASELINE PROGRAM
174 C
175 C STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER

LINES ADDED TO DELIVERABLE PROGRAM
193 C
194 C STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER

LINES DELETED FROM BASELINE PROGRAM
274 C
275 C NOTE: DEBUGGING PRINT STATEMENTS

LINES ADDED TO DELIVERABLE PROGRAM
293 C
294 C NOTE: DEBUGGING PRINT STATEMENTS

FIGURE 2.2-10 SUMMARY OF MODIFICATION TO SUBROUTINE SIGNAL

PAGE 2

2-33
LINES DELETED FROM BASELINE PROGRAM
279 C WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1).
280 C 2 DFWTS(5,1)
281 902 FORMAT('NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 = ',I5,6F10.2,15)

LINES ADDED TO DELIVERABLE PROGRAM
298 C WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1).
299 C 2 DFWTS(5,1)
300 902 FORMAT('NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 = ',I5,6F10.2,15)

Number of difference sections found: 9
Number of difference records found: 48

DIFFERENCES /IGNORE=() /MERGED=1 /OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF1.FOR;1-
SYS$DISK3:[MCCOLLOUGH]SIGNALH.FOR;2-
SYS$DISK3:[MCCOLLOUGH]SIGNALF.FOR;2

FIGURE 2.2-10 SUMMARY OF MODIFICATION TO SUBROUTINE SIGNAL

PAGE 3

2-34
SUBROUTINE ATRACK
REAL INTT.IAZDSC.IELDSC
COMMON /CNTL/IIPWR.IMODE.IDUMC(7),DUMC(3)
COMMON /INPUT/DUM(6),EWB(3),DUM(18)
COMMON /OUTPUT/IIDUM(3),DIUM(2),SPANG,SRANG,SPRTE,SRRTE,SRSS.
COMMON /ICNTL/I2DUM(14),MRNG.MSAM.MPRF.IDUM2(11)
COMMON /SYSDAT/TSAM.DR(3).CP.SP.PSI.PSBIAS.ALBIAS.BTBIAS.
COMMON /ATDAT/CA,SA.CB.SB,AZRATE.ELRATE,ALRATE.BTRATE,AL.BT.
DIMENSION AT1(IO,2),AT2(Ie,2).TX1(3,3).TX2(3,3).TX3(3,3).TBL(3,3)
DIMENSION TDC(3)
DATA AT1/9*1.5529E-3,2.166E-4,6*3.9750E-3,1.5529E-3,
2 3*2.166E-4,AT2/6.5907E-3,2.3725E-3,
2 3*1.e546E-2,6.5907E-3,3.2.3725E-3/
DATA TDC/0.1195161,0.2561557/
C DEFINITION: AT1=KEQ=(WN=*2)/(4.*DIFFERENCE PATTERN SLOPE) WHERE
WN IS NATURAL FREQUENCY OF THE
C DEFINITION: AT2=KEQ*TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE
C CONVERGENCE TIME.
C
TCON=TSAM/TDC(MPRF).
C
C *****************************************
C * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *
C *****************************************
C
C *****************************************
C * STEP 1: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANS- *
C FORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW ANGLE ERROR WR1 BODY FRAME).
C *****************************************
C
CALL GAMMA(TX1,-(BT+BTHIAS))
CALL ETA(TX2,-(AL+ALBIA))
CALL MULT33(TX2,TX1,TX3)
CALL PHI(TX2,-PSI)
CALL MULT33(TX2,TX3,TBL)
C
C *****************************************
C * STEP 2: UPDATE ESTIMATED TARGET INERTIAL AZIMUTH AND ELEVATION *
C *****************************************

FIGURE 2.2-II BASELINE VERSION OF SUBROUTINE ATRACK
PAGE 1
QUANTIZE THE ANGLE DISCRIMINANTS TO 3/16 DB.
IAZDSC=INTT(5.333333,AZDISC,TCON+B.5)/TCON
IELDSC=INTT(5.333333,ELDISC,TCON+B.5)/TCON
IF(IAZDSC.GT.255)IAZDSC=255
IF(IELDSC.GT.255)IELDSC=255
IF(IAZDSC.LT.-256)IAZDSC=-256
IF(IELDSC.LT.-256)IELDSC=-256
ADSC=e.e431,IAZDSC
EDSC=B.B431,IELDSC
UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.
AZRATE=AZRATE+TSAM,ATI(MRNG.IMODE),ADSC
UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.
ELRATE=ELRATE+TSAM-ATI(MRNG.IMODE),EDSC
STEP 3: UPDATE INNER AND OUTER GIMBAL RATES.
C OUTER GIMBAL FRAME.
WXG=CP,EWB(1)+SP,EWB(2)
WYG=CA,(-SP,EWB(1)+CP,EWB(2))+SA*EWB(3)
WGZ=-SA,(-SP,EWB(1)+CP,EWB(2))+CA,EWB(3)
C OUTER GIMBAL RATE.
IF(ABS(CB).LT.I.BE-6) GO TO 2
ALRATE=(AZRATE+AT2(MRNG.IMODE),ADSC+WGZ,SB)/CB-WGX
GO TO 4
2 ALRATE=e.
4 CONTINUE
C INNER GIMBAL RATE.
BTRATE=(ELRATE+AT2(MRNG.IMODE),EDSC)-WGY
C ADD ALPHA AND BETA TO OUTPUT IN DEG
SSALP=AL=57.29576
SSBET=BT,57.29576
C STEP 4: UPDATE INNER AND OUTER GIMBAL POSITIONS.
AL=AL+TSAM,ALRATE
BT=BT+TSAM,BTRATE
C ADD ALPHA AND BETA TO OUTPUT IN DEG
C * STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO *
C BODY FRAME FOR USE IN DISPLAYS AND G AND N *
C NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW *
C ANGLE ERROR WRT BODY FRAME.
C UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
C FOR DISPLAY.
SPRTE=1000.*(TBL(2,1)*AZRATE+TBL(2,2)*ELRATE)
C UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
C FOR DISPLAY.
SRRTE=1000.*(TBL(1,1)*AZRATE+TBL(1,2)*ELRATE)
C UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
SPANG=ASIN(TBL(1,3))*57.29576
C UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.
IF(TBL(2,3).EQ.0.0.AND.TBL(3,3).EQ.0.0) GO TO 5
SRANG=ATAN2(-TBL(2,3),TBL(3,3))*57.29576
GO TO 7
5 IF(TBL(1,3).GT.0.0) SRANG=-90.0
0025730
0025740
0025790
0025800
0025820
0025830
0025840
0025850
0025860
0025900
0025910
0025920
0025930
0025940
0025950
0025960
0025970
0025980
0025990
0026000
0026010
0026020
0026030
0026040
0026050
0026060
0026070
0026130
0026140
0026150
0026160
0026170
0026180
0026190
0026200
0026210
0026220
0026230
0026240
0026250
0026260
0026270
0026300
0026310
0026320

FIGURE 2.2-11 BASELINE VERSION OF SUBROUTINE ATRACK
PAGE 2
IF(TBL(1,3).LT.0.0) SRANG=90.0
IF(TBL(1,3).EQ.0.0) STOP
C RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND
C -180.<SRANG<180.
7 IF(SPANG.LE.90.) GO TO 10
SPANG=-(180.-ABS(SPANG))*(SPANG/ABS(SPANG))
SRANG=(180.-ABS(SRANG))*(SRANG/ABS(SRANG))
10 CONTINUE
C NOTE: DEBUGGING PRINT STATEMENTS.
C WRITE(6,899)
899 FORMAT(/' ATRACK DEBUGGING DATA')
C WRITE(6,900) ALRATE,BTRATE,AZRATE,ELRATE,SRRTE,SPRTE
C WRITE(6,901) TBL(1,1),TBL(1,2),TBL(2,1),TBL(2,2)
C WRITE(6,902) AZDISC,ELDISC,ADSC,EDSC
900 FORMAT(' ALR,BTR,AZR,ELR,SRR,SPR=',6F10.2)
901 FORMAT(' TBL 2X2 ..',4F10.4)
902 FORMAT(' AZD,ELD,AD,ED ..',4F10.4)
RETURN
END

FIGURE 2.2-11 BASELINE VERSION OF SUBROUTINE ATRACK

PAGE 3 2-37
**THIS SUBROUTINE UPDATES AZ AND EL INERTIAL LOS RATES, THE ALPHA AND BETA GIMBAL RATES, THE ALPHAS AND BETA GIMBAL POSITIONS, AND THE TARGET PITCH AND ROLL ANGLES FOR THE DISPLAY.**

SUBROUTINE ATRACK
REAL INTT,K4,K5,K6
INTEGER AT1A(10,2),AT1E(10,2),AT2A(10,2),AT2E(10,2)
COMMON /CNTL/IPWR,IMODE,IMUC(7),DMUC(3)
COMMON /INPUT/DDUM(6),EWE(3),DDUM2(18)
COMMON /OUTPUT/1IDUM(3),DDUM2(2),DDUM3(18),DDUM4(3),SSALP,SSBET
COMMON /ICNTL/DDUM2(1),MRNG,MSAM,MPRF,DDUM2(11)
COMMON /SYSDAT/TSAM,DDUM4(5)
COMMON /ATDAT/DDUM(7)
COMMON /DSCRM/TSAM,DDUM4(11),DPAM,DPAM,DDUM(7)
COMMON /TDC/0.05122118,0.05122118,0.2561557/

DATA AT1A/9,5.1,5.1,5.1,3.1/ 
DATA AT1E/9,5.1,5.1,5.1,3.1/ 
DATA AT2A/9,487,149,5,3,149/ 
DATA AT2E/9,532,195,6,532,3,195/ 
DATA K6/3,6005,6,5,5,236/,DTR/.0174533/

DATA TDC/0.05122118,0.1195161,0.2561557/
DEFINITION: AT1=KEQ-(WN-2)/(4,DIFFERENCE PATTERN SLOPE) WHERE
WN IS NATURAL FREQUENCY OF THE LOOP.
DEFINITION: AT2=KEQ,TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE CONVERGENCE TIME.

TCON=TSAM/TDC(MPRF)

ATRACK MODIFIED JAN 28 1986 BY M. MEYER
MODIFICATIONS TO SUBROUTINE ATRACK WERE IMPLEMENTED TO UPDATE THE LOOP CONSTANTS AND MORE ACCURATELY SIMULATE THE ACTUAL SIGNAL PROCESSING PERFORMED BY THE RADAR.

NEW LOOP CONSTANTS JAN 28 1986

DATA AT1A/9,5.1,5.1,5.1,3.1/ 
DATA AT1E/9,5.1,5.1,5.1,3.1/ 
DATA AT2A/9,487,149,5,3,149/ 
DATA AT2E/9,532,195,6,532,3,195/ 
DATA K6/3,6005,6,5,5,236/,DTR/.0174533/

DATA TDC/0.05122118,0.1195161,0.2561557/
DEFINITION: AT1=KEQ-(WN-2)/(4,DIFFERENCE PATTERN SLOPE) WHERE
WN IS NATURAL FREQUENCY OF THE LOOP.
DEFINITION: AT2=KEQ,TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE CONVERGENCE TIME.

TCON=TSAM/TDC(MPRF)

FIGURE 2.2-12 DELIVERABLE VERSION OF SUBROUTINE ATRACK

PAGE 1
**STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE**

- Call Gamma(Tx1, -(BT+BTbias))
- Call Theta(Tx2, -(AL+ALbias))
- Call Mult33(Tx2, Tx1, Tx3)
- Call Phi(Tx2, -PSI)
- Call Mult33(Tx2, Tx3, TBL)

**STEP 2: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANSFORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW ANGLE ERROR WRT BODY FRAME).**

- Quantize the estimated target inertial azimuth and elevation rates in antenna LOS frame.

**STEP 3: UPDATE INNER AND OUTER GIMBAL RATES.**

- Compute required components of orbiter angular velocity vector in outer gimbal frame.

**FIGURE 2.2-12** DELIVERABLE VERSION OF SUBROUTINE ATRACK

PAGE 2

2-39
STEP 5: UPDATE INNER AND OUTER GIMBAL POSITIONS.

C INNER GIMBAL RATE
BTRATE = BTRATE - WGY

C OUTER GIMBAL POSITIONS (ALPHA ANGLE)
AL = AL + TSAM * ALRATE

C INNER GIMBAL POSITION (BETA ANGLE)
BT = BT + TSAM * BTRATE

C ADD ALPHA AND BETA TO OUTPUT IN DEG
SSALP = AL * .57.29576
SSBET = BT * .57.29576

C STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO BODY FRAME FOR USE IN DISPLAY.

C NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW ANGLE ERROR WRT BODY FRAME.

C UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
SPRTE = SRT(2,1) * AZRATE + SRT(2,2) * ELRATE

C UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
SRRTE = SRT(1,1) * AZRATE + SRT(1,2) * ELRATE

C UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
SPAN AS = SRT(1,3) * .57.29576

C UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.
IF (SRT(2,3).EQ..E.E) AND (SRT(3,3).EQ..E.E) GO TO 5
SRANG = ATAN2 (-SRT(2,3), SRT(3,3)) * .57.29576

IF (SRT(1,3).EQ..E.E) STOP

C RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND -180.<SRANG<180

C NOTE DEBUGGING PRINT STATEMENTS.
C WRITE(6, 899)
C WRITE(6, 900) ALRATE, BTRATE, AZRATE, ELRATE, SRT(2,1), SRT(2,2)
C WRITE(6, 901) SRT(1,1), SRT(1,2), SRT(1,3)
C WRITE(6, 902) ADISC, ELDISC, IADISC, IELDISC

FIGURE 2.2-12 DELIVERABLE VERSION OF SUBROUTINE ATRACK PAGE 3
LINES DELETED FROM BASELINE PROGRAM
25  REAL INTT, IAZDSC, IELDSC
    COMMON /CNTL/IPWR, IMODE, IDUMC(7), DUMC(3) 00025350

LINES ADDED TO DELIVERABLE PROGRAM
25  REAL INTT, K4, K5, K6 00025335
26  INTEGER ATIA(10,2), ATIE(10,2), AT2A(10,2), AT2E(10,2)
27  COMMON /CNTL/IPWR, IMODE, IDUMC(7), DUMC(3) 00025350

LINES DELETED FROM BASELINE PROGRAM
36  DIMENSION AT1(10,2), AT2(10,2), TX1(3,3), TX2(3,3), TX3(3,3), TBL(3,3)00025450
37  DIMENSION TDC(3)
38  DATA AT1/9*1.5529E-3, 2.0106E-4, 6*3.9750E-3, 1.5529E-3, 32.0166E-4/ 00025460
40  3 6*1.0546E-2, 2.6.5907E-3, 3*2.3725E-3, 32.0166E-4/ 00025480
41  DATA TDC/0.65122118, 0.1195161, 0.2561557/

LINES ADDED TO DELIVERABLE PROGRAM
37  DIMENSION TX1(3,3), TX2(3,3), TX3(3,3), TBL(3,3)
38  DIMENSION TDC(3)
39  C
40  C
41  C ATRACK MODIFIED JAN 28 1986 BY M. MEYER
42  C MODIFICATIONS TO SUBROUTINE ATRACK WERE IMPLEMENTED
43  C TO UPDATE THE LOOP CONSTANTS AND MORE ACCURATELY
44  C SIMULATE THE ACTUAL SIGNAL PROCESSING PERFORMED
45  C BY THE RADAR
46  C
47  C
48  C
49  C
50  C
51  DATA AT1A/9*5.1, 6*13.5, 3*1/ 00025360
52  DATA AT1E/9*6.1, 6*16, 6.2*1.2/ 00025370
53  DATA AT2A/9*407, 149, 6*662, 407, 3*149/ 00025380
54  DATA AT2E/9*532, 195, 6*866, 532, 3*195/ 00025390
56  C
57  DATA TDC/0.65122118, 0.1195161, 0.2561557/

LINES DELETED FROM BASELINE PROGRAM
75  ADSC=0.0431*IAZDSC 00025730
76  EDSC=0.0431*IELDSC 00025740
77  C UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.
78  AZRATE=AZRATE+TSAM*ATI(MRNG, IMODE)+ADSC 00025790
79  C UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE. 00025800
80  C
81  C
82  C
83  C
84  C
85  C
86  C
87  C
88  C
89  C
90  C
91  C
92  C
93  C
94  C
95  C
96  C
97  C
98  C
99  C

PRECEDING PAGE BLANK NOT FILMED

FIGURE 2.2-13 SUMMARY OF MODIFICATIONS TO SUBROUTINE ATRACK
PAGE 1

2-41
FIGURE 2.2-13 SUMMARY OF MODIFICATIONS TO SUBROUTINE ATRACK

PAGE 2
FIGURE 2.2-13 SUMMARY OF MODIFICATIONS TO SUBROUTINE ATRACK

PAGE 3

2-43
Figure 2.2-14 is a listing of the function KSAT which has been added to the final program. Figure 2.2-15 is a listing of the subroutine READPAT, and Figure 2.2-16 is a listing of the subroutine INTERP. (No original listings or summaries of changes exist because these are new subroutines.)

2.2.4 Integration and Test Data

There were two major sections of code that required testing: the antenna pattern module and the loop filter module. Methods for testing these modules and the test results are summarized in this subsection.

2.2.4.1 Antenna Pattern Module Tests

The subroutines that generate all of the antenna parameter data were written and validated during the study documented in Reference 6. As discussed in Section 2.2.2, these original subroutines were modified by replacing bicubic spline interpolation with two dimensional linear interpolation. After these subroutines were modified, two types of tests were performed to help validate their correctness: a static test and a dynamic test.

The static test of the pattern interpolation routines was to generate three-dimensional plots of all five antenna pattern parameters on an 8 degree by 8 degree grid with a resolution of at least 0.1 degrees and examine the data for any obvious flaws. This task was done using the DISSPLA package on the Building 44 VAX/780 at JSC. An examination of three-dimensional data showed no obvious errors. Unfortunately, we cannot present the data because no high quality hardcopy unit was available.

The second test was a dynamic test. Its purpose was to demonstrate the sign of the slope of the difference patterns was correct and that the general behavior of the pattern interpolation routines in a dynamic environment was satisfactory. This test is defined as follows. First, the subroutines were installed in the angle tracking loop simulation program...
* INTEGER FUNCTION KSAT JAN 28 1986

* THIS FUNCTION CHECKS ATRACK LOOP FOR SATURATION

INTEGER FUNCTION KSAT(K)

IF(K.GE.0) THEN
    KSAT=JMIN0(K,2**15)
ELSE
    KSAT=JMAX0(K,-2**15)
END IF
RETURN
END

FIGURE 2.2-14 DELIVERABLE VERSION OF SUBROUTINE KSAT
subroutine reodPAT

Read in the sum, phase, and difference patterns

real a1inear( 41,41 ), ellinear( 41,41 )
real sallinear( 41,41 ), sellinear( 41,41 )
real pallinear( 41,41 ), pellinear( 41,41 )
common / linear / allinear, ellinear
common / linear1 / sallinear, sellinear
common / linear2 / pallinear, pellinear

open( unit=3, file='[KUBAND.HOWARD.MARK]a11.dat',
     access='sequential', form='unformatted',
     status='old', readonly )
read( 3 ) ( allinear( i,j ), j = 1,41, i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]el1.dat',
     access='sequential', form='unformatted',
     status='old', readonly )
read( 3 ) ( ellinear( i,j ), j = 1,41, i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]sz1.dat',
     access='sequential', form='unformatted',
     status='old', readonly )
read( 3 ) ( sellinear( i,j ), j = 1,41, i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]szs.dat',
     access='sequential', form='unformatted',
     status='old', readonly )
read( 3 ) ( sellinear( i,j ), j = 1,41, i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]szl.dat',
     access='sequential', form='unformatted',
     status='old', readonly )
read( 3 ) ( sellinear( i,j ), j = 1,41, i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]oz1p.dat',
     access='sequential', form='unformatted',
     status='old', readonly )
read( 3 ) ( pallinear( i,j ), j = 1,41, i = 1,41 )
close( 3 )
open( unit=3, file='[KUBAND.HOWARD.MARK]ellp.dat',
    access='sequential', form='unformatted',
    status='old', readonly )
read( 3 ) ( ( pelinear( i,j ), j = 1,41 ), i = 1,41 )
close( 3 )
return
end
Subroutine: Antenna pattern interpolation.
Input: Azimuth and elevation angles in degrees.
Output: Interpolated difference, sum, and phase values
for all 18 antenna patterns.

subroutine interp( az, el)

Linearly interpolate the gain, phase and difference patterns

real allinear( 41,41 ), ellinear( 41,41 )
real sallinear(41,41), sellinear(41,41)
real pallinear(41,41), pellinear(41,41)
common / linear / allinear, ellinear
common / linear1 / sallinear, sellinear
common / linear2 / pallinear, pellinear
common / SUDIPH / X,Y,Z,PAZ,PEL

iax = jint( ( az + 4. ) * 5. )
ix = jint( ( el + 4. ) * 5. )
az0 = floatj( iax ) / 5. - 4.
el0 = floatj( ix ) / 5. - 4.

iaz = jint ( ( az + 4. ) * 5. ) + 1
jel = jint ( ( el + 4. ) * 5. ) + 1

find azd values

f0 = 10.** ( allinear( iaz, jel ) ) / 20. 
f1 = 10.** ( allinear( iaz+1, jel ) ) / 20. 
f2 = 10.** ( allinear( iaz, jel+1 ) ) / 20. 
f3 = 10.** ( allinear( iaz+1, jel+1 ) ) / 20. 
fa = f0 + (f1-f0)/.2 * ( az-az0 )
f = f2 + (f3-f2)/.2 * ( az-az0 )
fx = fa + (fb-fa)/.2 * ( el-e10 )

FIGURE 2.2-16 DELIVERABLE VERSION OF SUBROUTINE INTERP
PAGE 1
\[ Y = f_x \]

**find eld values**

\[ f_0 = 10.**((s_1)linear(iaz, jel) /20.) \]
\[ f_1 = 10.**((s_1)linear(iaz+1, jel) /20.) \]
\[ f_2 = 10.**((s_1)linear(iaz, jel+1) /20.) \]
\[ f_3 = 10.**((s_1)linear(iaz+1, jel+1) /20.) \]

\[ f_a = f_0 + (f_1-f_0)/.2 \cdot (az-az_0) \]
\[ f_b = f_2 + (f_3-f_2)/.2 \cdot (az-az_0) \]
\[ f_x = f_a + (f_b-f_a)/.2 \cdot (el-el_0) \]

**Z = f_x**

**find azs values**

\[ f_0 = 10.**((s_1)linear(iaz, jel) /20.) \]
\[ f_1 = 10.**((s_1)linear(iaz+1, jel) /20.) \]
\[ f_2 = 10.**((s_1)linear(iaz, jel+1) /20.) \]
\[ f_3 = 10.**((s_1)linear(iaz+1, jel+1) /20.) \]

\[ f_a = f_0 + (f_1-f_0)/.2 \cdot (az-az_0) \]
\[ f_b = f_2 + (f_3-f_2)/.2 \cdot (az-az_0) \]
\[ f_x = f_a + (f_b-f_a)/.2 \cdot (el-el_0) \]

**X = f_x**

**find azp values**

\[ f_0 = (p_1)linear(iaz, jel) \]
\[ f_1 = (p_1)linear(iaz+1, jel) \]
\[ f_2 = (p_1)linear(iaz, jel+1) \]
\[ f_3 = (p_1)linear(iaz+1, jel+1) \]

\[ f_a = f_0 + (f_1-f_0)/.2 \cdot (az-az_0) \]
\[ f_b = f_2 + (f_3-f_2)/.2 \cdot (az-az_0) \]
\[ f_x = f_a + (f_b-f_a)/.2 \cdot (el-el_0) \]

**PAZ=f_x**

! phase in degrees

**find elp values**

\[ f_0 = (p_1)linear(iaz, jel) \]
\[ f_1 = (p_1)linear(iaz+1, jel) \]
\[ f_2 = (p_1)linear(iaz, jel+1) \]
\[ f_3 = (p_1)linear(iaz+1, jel+1) \]

\[ f_a = f_0 + (f_1-f_0)/.2 \cdot (az-az_0) \]
\[ f_b = f_2 + (f_3-f_2)/.2 \cdot (az-az_0) \]
\[ f_x = f_a + (f_b-f_a)/.2 \cdot (el-el_0) \]

**PEL=f_x**

! phase in degrees

return

end

**FIGURE 2.2-16 DELIVERABLE VERSION OF SUBROUTINE INTERP**

PAGE 2

2-49
documented in Reference 6. Then the program was run with a 0 dBsm target, fixed at one nautical mile range for 100 seconds. Next, the original program with bicubic spline interpolation was run with the same scenario. Then, the statistical aspects of the output time histories of the four parameters, azimuth, elevation, azimuth rate, and elevation rate, were compared. Results of the comparison showed that the differences in performance was negligible for all four parameters. These results confirm that the two dimensional linear interpolation antenna pattern model has been implemented properly in a dynamic environment.

Once initial tests were completed and the subroutines changes were validated using the simulation program of Reference 6, these modified routines were then lifted as a unit from this program and installed in the SES simulation code. Additional tests were run on the modified SES code to further validate the pattern changes and other loop changes. These tests and their results are described in the following subsection.

2.2.4.2 Loop Filter Module Tests

The loop filter module changes were validated using various qualitative tests. The purpose of these tests were to verify (1) the proper slope sign in the difference patterns, (2) the noise properties of the tracking loop, and (3) the transient response of the tracking loop filter.

A very simple test was used to verify the slope sign in the difference patterns. A stationary, 0 dBsm target was placed at one nautical mile and was tracked by the simulation for a period of 100 seconds. Results of this test showed that the target was tracked in a stable, steady-state fashion in both angle and angle rate, ensuring that the sign of the difference pattern slope was correct.

The next set of tests established the statistical properties of the angle tracking loop. That is, these tests established the approximate noise bandwidth of the loop. Again a 0 dBsm, stationary point target was tracked at the following three ranges: 1 nautical mile, 3 nautical miles, and
5 nautical miles. These ranges were selected to exercise the three different loop bandwidths of the tracker. Tables 2.2-3 and 2.2-4 summarize the results of these tests for the original SES simulation code and the modified simulation code. A comparison of the means and the standard deviations of the old and the new version show a fairly reasonable matchup of the data. This is significant proof that the angle tracker upgrades have been installed properly. The fact that the modified version is noisier than the original version is encouraging since this brings the angle tracking loop simulation data into better agreement with the flight data.

This brings us naturally to our third set of angle tracking loop tests. These tests involved injecting trajectories from the SORTE experiment and the Palapa B rendezvous into the simulation and comparing the simulation predictions with the actual radar data in each case. Discussions of the results of these simulation experiments are delayed until sections 3.7 and 4.2, respectively. The purpose of these tests was to determine the overall fidelity of the angle and ILOS angle rate trackers in terms of random properties and transient response properties for typical rendezvous situations.

2.3 AGC UPGRADES

2.3.1 PROBLEM DEFINITION

The AGC module discussed in this section includes three components: (1) calculation of the AGC update, (2) calculation of the RSS, and (3) calculation of the A/D saturation effects (if any). High fidelity models for each of these components were defined and discussed in detail in the final report for NASA Contract No. NAS9-15840 (Reference 4). However, an examination of the simulation code published in the appendix of that report revealed that these components had not been completely upgraded in several areas. The most serious problem with this less accurate model is a 12 dB discontinuous jump in AGC, and therefore RSS, accompanying a transition in the sample rate under normal target conditions (greater than a 0 dBsm Radar Cross Section (RCS)). This effect is demonstrated in Figures 2.3-1 and 2.3-2. Now, under normal target conditions, theory and actual operational data show that there is no such discontinuity at this transition.
<table>
<thead>
<tr>
<th>TABLE 2.2-3</th>
<th>A COMPARISON OF THE STANDARD DEVIATIONS OF THE ANGLE TRACKING PERFORMANCE FOR THE OLD AND THE NEW SIMULATION MODELS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANGE</td>
<td>6000 FEET</td>
</tr>
<tr>
<td>VERSION</td>
<td>OLD</td>
</tr>
<tr>
<td>Roll Angle</td>
<td>5.27 E-3</td>
</tr>
<tr>
<td>Pitch Angle</td>
<td>5.74 E-3</td>
</tr>
<tr>
<td>ILOS Roll Rate</td>
<td>1.58 E-3</td>
</tr>
<tr>
<td>ILOS Pitch Rate</td>
<td>1.26 E-3</td>
</tr>
<tr>
<td>RANGE</td>
<td>6000 FEET</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>VERSION</td>
<td>OLD</td>
</tr>
<tr>
<td>Roll Angle</td>
<td>1.88 E-6</td>
</tr>
<tr>
<td>Pitch Angle</td>
<td>3.09 E-4</td>
</tr>
<tr>
<td>ILOS Roll Rate</td>
<td>1.26 E-5</td>
</tr>
<tr>
<td>ILOS Pitch Rate</td>
<td>2.46 E-5</td>
</tr>
</tbody>
</table>
FIGURE 2.3-1 AGC PROFILE FOR THE RANGE PROFILE GIVEN IN FIGURE 2.3-2. DISCONTINUITIES OCCUR AT THE SAMPLE RATE TRANSITION.
FIGURE 2.3-2  RANGE PROFILE USED TO GENERATE AGC PROFILE OF FIGURE 2.3-1
The purpose of this subsection is to point out the weaknesses of the current SES simulation code in these areas and define the corrections.

2.3.2  Definition of Algorithm Modifications

2.3.2.1  AGC Model Improvements

To facilitate a description of the weaknesses in the baseline version of the AGC model, a concise definition of the high fidelity model (from Reference 4) is provided below.

The upgraded AGC model includes the following features:

1. The AGC increment for the next data cycle is determined by subtracting the mean signal level at the log converter output (see Figure 2.3-3) from a prestored value which represents a signal power of $4q^2$ at the A/D input.

2. It includes the effects of quantization noise injected by the A/D converter.

3. It allows a maximum of 10 dB increment in AGC or a minimum of -10 dB decrement in AGC per data cycle.

4. The absolute AGC value cannot drop below 6 dB, the nominal search AGC value.

A crude A/D converter saturation model has been implemented in conjunction with this model to increase AGC response fidelity in anticipation of large, sudden increases in satellite RCS values.

The AGC algorithm can be summarized as follows:

Step 1: Compute the AGC change, $\Delta$AGC, based on the present mean signal level estimate at the log converter output.

Step 2: If $\Delta$AGC $\geq$ 10 dB, then $\Delta$AGC = 10 dB, or if $\Delta$AGC $\leq$ -10 dB, then $\Delta$AGC = -10 dB.

Step 3: Compute the new AGC.

Step 4: If new AGC $\leq$ 6 dB, then new AGC = 6 dB.
FIGURE 2.3-3 SIMPLIFIED DIAGRAM OF THE AGC TRACKING LOOP
Computation of the change in AGC, AGC, is done using the following expression.

\[
(2-9) \quad \text{AGCERR}(N) = k_1 G/(\text{AGC}(N)(\text{SNR}_{DT}(N)+1)+k_2)
\]

where \( G \) = Signal-to-noise power ratio (SNR) gain from the A/D output to the doppler filter output,

\[
\text{SNR}_{DT} = \text{Signal-to-thermal noise power ratio at the doppler filter output,}
\]

\[
k_1 = (2q)^2/N_t,
\]

\[
k_2 = (q)^2/(12N_t)
\]

\[
N_t = \text{unAGC'd thermal noise power at the A/D input.}
\]

The updated AGC value is computed with the expression

\[
(2-10) \quad \text{AGC}(N+1) = \text{AGC}(N)\text{AGCERR}(N)
\]

\( k_1 \) can be interpreted as the ratio of the desired AGC'd track signal power level at the A/D input to unAGC'd thermal noise power level at the A/D input, \( k_2 \) is interpreted as the ratio of the quantization noise power, \( q^2/12 \), to the unAGC'd thermal noise power at the A/D input. Finally, to be consistent with the baseline code, we will set \( G = 4 \ P_s \). The values for \( k_1, k_2, G \), and \( P_s \) for the various modes and range intervals are summarized in Table 2.3-1.

Some comments on the accuracy of this algorithm versus actual AGC operation are in order. We first note that the form for predicting the AGC change given in Equation (2-9) is quite accurate. It has the A/D quantization noise and the noise floor concept folded into the calculation. As noted earlier, the quantization noise includes only the contribution from the A/D converter and is assumed to have a power of \( q^2/12 \) where \( q \) represents the voltage of a single A/D step. All other quantization noise sources are dwarfed in comparison to this source, especially when comparing their relative
### TABLE 2.3-1 AGC CALCULATION CONSTANTS

<table>
<thead>
<tr>
<th>Range Interval, Ft.</th>
<th>( N_{T,q^2} )</th>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( G )</th>
<th>( P_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;2560</td>
<td>124.80</td>
<td>0.0321</td>
<td>0.00067</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>(2560, 5750)</td>
<td>7.84</td>
<td>0.51</td>
<td>0.011</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>(5760, 11510)</td>
<td>7.84</td>
<td>0.51</td>
<td>0.011</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>(11510, 23030)</td>
<td>7.84</td>
<td>0.51</td>
<td>0.011</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>(23040, 43510)</td>
<td>7.84</td>
<td>0.51</td>
<td>0.011</td>
<td>32</td>
<td>8</td>
</tr>
<tr>
<td>&gt;43510</td>
<td>7.84</td>
<td>0.51</td>
<td>0.011</td>
<td>64</td>
<td>16</td>
</tr>
<tr>
<td>Active</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;49910</td>
<td>124.8</td>
<td>0.0321</td>
<td>0.00067</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>&gt;49920</td>
<td>7.84</td>
<td>0.51</td>
<td>0.011</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

Effects at the doppler filter output. The search thermal noise AGC value or the "noise floor" in this expression is fixed at 6 dB. This floor represents the search AGC value at the time the target is detected. In reality, this number is a random process, fluctuating from acquisition to acquisition. However, we treat the noise floor as a deterministic value and assign it a value equal to the mean of the random process, i.e., 6 dB for all acquisitions.

The following errors were found in the baseline code and corrected. First, and most serious, \( k_1 \) and \( k_2 \) did not change at the sample rate transition as shown in Table 2.3-1, causing the 12 dB discontinuity in AGC as discussed earlier. In the baseline program \( k_1 \) and \( k_2 \) assumed the low sample rate values at all ranges. Secondly, the AGC was allowed to drop to 0 dB rather than limit at the nominal search AGC level of 6 dB. Both of these errors were corrected in the upgraded simulation code documented in Section 2.3.3.
2.3.2.2 RSS Model Improvements

The Ku-Band Radar computes the RSS using the following very simple relation.

\[(2-11) \quad \text{RSS}(N) = (10 \log \left(1/\text{AGC}(N)\right)-6)k_o\]

where AGC\( (N) \) is the latest estimate of AGC and the value 6 represents the nominal search AGC value (or search "noise floor"). The value of \( k_o \) is 5 volts/160 dB which converts the RSS from dB to voltage from the display meter. Full scale AGC is 160 dB which corresponds to a full scale meter voltage of 5 volts. Since the AGC is not allowed to drop below 6 dB, the RSS will not drop below 0 volts.

In the baseline code for the RSS module there were two errors: (1) the nominal search AGC value was set to 0 dB, and (2) the scale factor \( k_o \) was ignored. Both corrections have been made in the present version of the RSS as documented in Section 2.3.3.

2.3.2.3 A/D Saturation Noise Model Improvements

A simple model for injecting A/D saturation effects into the tracking signal response was developed in anticipation of encountering sudden, large increases in receive signal strength when rendezvousing with various satellite targets. The model is fairly crude and is based on the concept that the total signal-plus-noise power at the A/D output should be limited to \((7q)^2\). The basic idea of the model can be expressed as follows:

Step 1: Compute the signal-plus-noise power at the A/D input.

Step 2: If the total power is greater than \((7q)^2\), then limit this power to \((7q)\).

The total signal-plus-noise power at the A/D input is computed using the expression,
Total Power = AGC(N)N t(SNR t(N)/G+1)

where SNR t/G is equivalent to SNR vt, the signal-to-thermal noise power ratio at the A/D input. SNR vt is represented in this form because it is not easy to compute directly within the simulation, while SNR t and G are easily accessed. Hence, the indirect form of the calculation is used.

In the computer simulation code all powers are normalized to the unAGC'd thermal noise power at the ADC input. So the implementation of the saturation noise model is given by the inequality.

AGC(N)(SNR t(N)/G+1) ≤ (7q) 2/N t

where (7q) 2 is the maximum total power at the ADC output and N t represents the unAGC'd thermal noise power (2.8q) 2 in the low sample rate mode and (11.2q) 2 in the high sample rate mode).

There are two errors in the baseline version of the saturation noise model code:

(1) The value for N t in the low sample rate case is (1.4q) 2 rather than (2.8q) 2, and

(2) The value for N t in the high sample rate mode is (1.4q) 2 rather than (11.2q) 2. The errors have been corrected in the final version of the code and are documented in Section 2.3.3.

2.3.3 Software Design Documentation

The simulation changes documented in Section 2.3.2 affect these subroutines in the baseline code: (1) RSS, (2) SATNSE, and (3) DISCRM.

Three lines of code were changed in the baseline version of RSS shown in Figure 2.3-4. These included:

(1) Changing the AGCERR computation to properly reflect the sample rate transition
*THIS SUBROUTINE COMPUTES THE RADAR SIGNAL STRENGTH AND UPDATES* 

**SUBROUTINE RSS**

COMMON /CNTL/IPWR, IMODE, IDUM1(7), DUM1(3)
COMMON /ICNTL/IDUM2(14), MRNG, IDUM6(12)
COMMON /OUTPUT/IDUM7(3), DUM5(6), SRSS, IDUM4(4)
COMMON /AGCDAT/AGCO, AGCODB, SNRDT, SNRDTD
DIMENSION PS(16,2)
DATA PS/9, l., 2., 5*l., 2., 4., 8., 8., 16./, ONV/O.04166666/

**STEP 1: UPDATE SYSTEM AGC**

AGCERR = 4.*PS(MRNG, IMODE)/(AGCO, (SNRDT+I.Q)+QNV)
IF(AGCERR.GT.IO.) AGCERR = IO.e
IF(AGCERR.LT.Q.1) AGCERR = 0.1

**STEP 2: UPDATE RADAR SIGNAL STRENGTH VALUE**

IF(AGCO.LT. t .0E-15) AGC0 = 1.0E-15
SRSS = IO.*ALOG10(SRSS)
RETURN
END

**FIGURE 2.3-4 BASELINE VERSION OF SUBROUTINE RSS**
(2) Adding the 6 dB AGC floor level, and
(3) Subtracting 6 dB from the RSS computation.
Figures 2.3-5 gives the new version of RSS and Figure 2.3-6 provides a summary of the changes.

Two lines of code were changed in the baseline version of SATNSE shown in Figure 2.3-7. The ratio of \((7q)^2/N_t\) was changed from 12.25 to 6.25 for the low sample rate mode. Also, the array PS was updated to reflect the values given in Table 2.3-1 for the various range intervals. The new version of SATNSE is listed in Figure 2.3-8 and a summary of the changes is provided in Figure 2.3-9.

Three major changes were made in the baseline version of subroutine DISCRM shown in Figure 2.3-10. Two changes involved altering values of constants. The values of the array PS were updated to those of Table 2.3-1. The constant QNV was converted to function of the sample rate and its values were computed appropriately. The updated version of DISCRM is given in Figure 2.3-11 and the changes are summarized in Figure 2.3-12.

2.3.4 Integration and Test Data

Accurate Radar Signal Strength (RSS) simulation outputs was one of the major objectives of the AGC update. With this objective in mind, the tests performed were to validate the RSS output. Since the RSS is a function of the AGC, a proper RSS output would validate the AGC modifications.

2.3.4.1 Test Definition

A test trajectory was constructed where the target originated at a 2400 ft range, moved out to about 4000 ft and then closed to 1400 ft thus moving the simulated target through two sample rate changes. Two simulation runs of this trajectory were made with the radar cross section (RCS) set to 10d Bsm. A 10 dBsm target was chosen since that is a common actual target RCS. One simulation run was made with the RCS set to -40 dBsm. This RCS was chosen because one could predict a discontinuity in RSS at the sample rate changes from the equations and it was desirable to see if the simulation also produced this result. 2-63
SUBROUTINE RSS

COMMON /CNTL/IPWR,IMODE,IMTM1(7),IDUM1(3)
COMMON /ICNTL/IDL(14),IMOD,IMAM,IMTM6(11)
COMMON /OUTPUT/IDUM7(3),IDUM8(4),SRSS,IMTM4(4)
COMMON /AGCDAT/AGC,AGCDB,SNRDT,SNRTE

DIMENSION PS(IO,2),ONV(2),AI(2)

DATA PS/9.4.,2.5.,1.,5.,4.,2.,1.,4.,6.,8.,8.,16./
DATA ONV/1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12./
DATA SNRTE/.0007,0.1,0.32,1./

SUBROUTINE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986

C--------------------------------------------------------------------------
C STEP 1: UPDATE SYSTEM AGC
C--------------------------------------------------------------------------
C
C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
C--------------------------------------------------------------------------
C
SUBROUITE RRS HAS BEEN UPDATED TO CORRESPOND TO THE
DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986

C--------------------------------------------------------------------------
C STEP 2: UPDATE RADAR SIGNAL STRENGTH VALUE
C--------------------------------------------------------------------------
C
RETURN
END

FIGURE 2.3-5 DELIVERABLE VERSION OF SUBROUTINE RSS

2-64
LINES DELETED FROM BASELINE PROGRAM
24 COMMON /ICNTL/IDUM2(14),MRNG,IDUM6(12) 00029320
25 COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4) 00029330
26 COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD 00029340
27 DIMENSION PS(10,2) 00029350
28 DATA PS/9*1.,2..5*1.,2..4.,8.,8..16./,ONV/0.04166666/ 00029360
29 DATA ONV/e.e4166666/ 00029370

LINES ADDED TO DELIVERABLE PROGRAM
24 COMMON /ICNTL/IDUM2(14),MRNG,MSAM,IDUM6(11) 00029330
25 COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4) 00029340
26 COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD 00029350
27 DIMENSION PS(10,2),ONV(2),A1(2) 00029360
28 DATA PS/9*1..2..5*1..2..4..8..8..16./,ONV/0.00067, .01/,A1/.0321,.51/ 00029370

C SUBROUTINE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
C DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
C KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986

C ------------------------------

C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
34 AGCERR=4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+ONV)
35 IF(AGCERR.GT.10.) AGCERR=10.0 00029420

C ------------------------------

C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
48 IF(AGCO.GT.2.5) AGCO=2.5 00029440
49 AGCODB=10.*ALOG10(AGCO) 00029500

C ------------------------------

C SUBROUTINE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
C DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
C KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986

C ------------------------------

C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
41 AGCERR=4.*PS(MRNG,IMODE)/(AGCO*(SNRDT+1.0)+ONV(MSAM))
42 IF(AGCERR.GT.10.) AGCERR=10.0 00029440

C ------------------------------

C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
41 IF(AGCO.GT.1.0) AGCO=1.0 00029490
42 AGCODB=10.*ALOG10(AGCO) 00029500

C ------------------------------

C UPDATED FEB 17, 1986
49 IF(AGCO.GT.0.25) AGCO=0.25 00029500
50 AGCODB=10.*ALOG10(AGCO) 00029500

C ------------------------------

C SUBROUTINE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
C DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
C KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986

C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
48 SRSS=1./AGCO 00029560
49 SRSS=10.*ALOG10(SRSS) 00029570
50 RETURN 00029580

FIGURE 2.3-6 SUMMARY OF MODIFICATIONS TO SUBROUTINE RSS

PAGE 1

2-65
****
LINES ADDED TO DELIVERABLE PROGRAM
56       SRSS=1./ACCO
57       C UPDATED FEB 17, 1986
58       SRSS=10.* ALOG10(SRSS)-6.0
59       RETURN

Number of difference sections found: 4
Number of difference records found: 19

DIFFERENCES /IGNORE=() / MERGED=1 / OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF5.FOR;1-
      SYS$DISK3:[MCCOLLOUGH]RSSH.FOR;2-
      SYS$DISK3:[MCCOLLOUGH]RSSF.FOR;2

FIGURE 2.3-6 SUMMARY OF MODIFICATIONS TO SUBROUTINE RSS

PAGE 2
**C**

************************************************************************

* THIS SUBROUTINE DETERMINES WHETHER THE SIGNAL PLUS NOISE IS SATURATING THE A/D — IF SO, THEN THE SNR AT DOPPLER FILTER OUTPUT IS LIMITED TO THE VALUE THAT JUST SATURATES THE A/D.

************************************************************************

SUBROUTINE SATNSE(SNF)
COMMON /CNTL/IPWR,IMODE
COMMON /ICNTL/IDUM(14),MRNG
COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
DIMENSION PS(10,2)
DATA PS/9*10.8,2.,5*1.2.,4..8.,8.,16./
SNF=1.
X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)
X=12.25/X
IF(X.GT.1) RETURN
SNF=X
RETURN
END

**FIGURE 2.3-7 BASELINE VERSION OF SUBROUTINE SATNSE**
• THIS SUBROUTINE DETERMINES WHETHER THE SIGNAL PLUS NOISE IS SATURATING THE A/D — IF SO, THEN THE SNR AT DOPPLER FILTER OUTPUT IS LIMITED TO THE VALUE THAT JUST SATURATES THE A/D.

SUBROUTINE SATNSE(SNF)
COMMON /CNTL/IPWR,IMODE
COMMON /ICNTL/IDUM(14),MRNG
COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
DIMENSION PS(16,2)

DATA PS/9,4.e2.,5s4.,2.,4.,8.,8.,16./
SNF=1.
X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.e0)

X=12.25/X WAS REPLACED BY X=6.25/X TO MORE ACCURATELY REFLECT A/D SATURATION BY M. MEYER FEB 17, 1986

X-6.25/X
IF(X.GT.1) RETURN
SNF=X
RETURN
END

FIGURE 2.3-8 DELIVERABLE VERSION OF SUBROUTINE SATNSE
LINES DELETED FROM BASELINE PROGRAM
28 DIMENSION PS(10,2) 0035670
29 DATA PS/9=10,0,2.,5*1,2.,4.,8..8..16./ 0035680
30 SNF=1. 0035690
31 X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0) 0035700
32 X=12.25/X 0035710
33 IF(X.GT.1) RETURN 0035720

LINES ADDED TO DELIVERABLE PROGRAM
28 DIMENSION PS(10,2) 0035670
29 C 0035680
30 C PS VALUES WERE UPDATED FEB 17,1986 BY M. MEYER--------- 0035690
31 C DATA PS/9=4.e,2.,5=4.,2.,4.,8.,8.,16./ 0035700
32 C SNF=1. 0035710
33 C X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0) 0035720
34 C X=6.25/X WAS REPLACED BY X=6.25/X TO MORE ACCURATELY 0035730
35 C REFLECT A/D SATURATION BY M. MEYER FEB 17, 1986 0035740
36 C X=6.25/X 0035750
37 C IF(X.GT.1) RETURN 0035760

Number of difference sections found: 1 0035770
Number of difference records found: 12 0035780

DIFFERENCES /IGNORE=/MERGED=1/OUTPUT=SYS$DISK3: [MCOLLough]DIFF3.FOR;1-
SYS$DISK3: [MCOLLough]SATNSEH.FOR;2-
SYS$DISK3: [MCOLLough]SATNSEF.FOR;2

FIGURE 2.3-9 SUMMARY OF MODIFICATIONS OF SUBROUTINE SATNSE

2-69
THIS SUBROUTINE ADDS THE EQUIVALENT NOISE TO THE ANGLE, RANGE, VELOCITY AND ON-TARGET DISCRIMINANT COMPONENTS AND THEN COMPUTES THE ANGLE, RANGE, VELOCITY, AND ON-TARGET DISCRIMINANTS.

SUBROUTINE DISCRM
REAL LATE,MEAN
COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
2 SRTE,SRSS,MADVF,MARDVF,MDRDVF
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
COMMON /ICNTL/I3DUM(14),MRNG,MSAM,MPRF,IDUM1(10)
COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA,
2 DUMS(3)
COMMON /TGTDAT/NT,DUM5(506),CGRNGE,CGVEL
COMMON /DSCRM/AZDISC,ELDISC,DDISC,VRDISC,RRTE,ODISC,SIGBR1,SNRD,
2 SIGDB
COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
2 DF2,DF4,SIGBAR
COMMON /NOISE/NS1,NS2,NN(10),GAUSS(326)
COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
DIMENSION NFREQ(2),PDIA(2),PDIR(2),PDIV(2),PS(10,2),BN(2),PT(3)
DATA NFREQ/1.5,.5/
DATA TDC/9772.4,.1195161/.5/.2561557/
NOTE: DEBUGGING PRINT STATEMENTS.
WRITE(6,900) SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE
900 FORMAT(' SPZ.SMZ.SPL.SML.E.L -'5F16.2)
WRITE(6,901) DF1,DF5,DF2,DF4,SIGBAR
901 FORMAT(' DF1.DF5.DF2.DF4.SIG -'5F16.2)
NOTE: THIS IS THE CONSTANT USED IN ACTIVE MODE.
YY=GA*PS(MRNG,IMODE)/(CGRNGE**2+BN(MSAM))
S1=YY/FLOAT(NFREQ(1MODE))

FIGURE 2.3-10 BASELINE VERSION OF SUBROUTINE DISCRM
PAGE 1

2-70
GO TO 10
C NOTE: THIS IS THE CONSTANT USED IN PASSIVE MODE.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCC MODS 2-15-B3 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C CONTINUE
PTIX=PT(ITXP)
IF(SRNG.LT.64)PTIX=4.2
IST7=0
IF(IST7.EQ.1)PTIX=4.2
C YY=GP*PS(WRNG,IMODE)*PTIX  /CGRN*E**4*B(N(MSAM))
SI=YY/FLOAT(NFREQ(IMODE))
C 10 SNRDT=YY*SIGBAR
WRITE(6,221)YY,SIGBAR
FORMAT('YY,SIGBAR -'.F14.5)
SNRDTD=ALOG(SNRDT)
SIGBAR=ALOG(SIGBAR)
C C221 WRITE(6,990) SNRDTD,SIGBAR
990 FORMAT('SNRDTD,SIGBAR ='.2F14.2)
C 2-2: COMPUTE PEAK SIGNAL POWER TO AVERAGE THERMAL NOISE POWER 
AT DOPPLER FILTER OUTPUT.
C SNRDT=YY*SIGBAR
WRITE(6,221)YY,SIGBAR
FORMAT('YY,SIGBAR -'.F14.5)
SNRDTD=ALOG(SNRDT)
SIGBAR=ALOG(SIGBAR)
C STEP 1-3: COMPUTE PEAK SIGNAL POWER TO TOTAL (THERMAL PLUS 
QUANTIZATION) NOISE POWER AT THE DOPPLER FILTER OUTPUT.
C CALL SATNISE(SNF)
XX=SNF*AGCO
XX=XX/(XX+ONV)
S1=S1*XX
YY=YY*XX
SNRD=YY*SIGBAR
SNRD=ALOG(SNRD)
C STEP 1-4: UPDATE NOISE SEQUENCE.
NN(1)=MOD(NN(1)+I.320)+I
DO 15 I=2,18
15 NN(I)=MOD(NN(I-1)+29,320)+1
IDI=NN(1)
GAUSS(ID1)=ANORM(NS1,NS2)
C STEP 2-1: CHECK ANTENNA STEERING MODE — SKIP STEP 2 IF IN 
GPC-DES OR MANUAL.
CCCCCCCCCCCCCCCCCCCCCCCCC MOD FEB 16 1983 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC 
IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20
C STEP 2-2: COMPUTE ANGLE DISCRIMINANT COMPONENT SCALE FACTOR.
ASCALE=S1*PDIA(IMODE)
C STEP 2-3: COMPUTE STATISTICS OF ADDITIVE NOISE FOR ANGLE 
DISCRIMINANT COMPONENTS.
MEAN=PDIA(IMODE)
VARM=VARM(2.*S1*SPAZ+1.)
VARM=VARM(2.*S1*SPAZ+1.)
VARPEL=VARM(2.*S1*SPAZ+1.)
VARPEL=VARM(2.*S1*SPAZ+1.)
C STEP 2-4: ADD EQUIVALENT NOISE TO ANGLE DISCRIMINANT COMPONENT 
SIGNS.
IDS=NN(6)

FIGURE 2.3-10 BASELINE VERSION OF SUBROUTINE DISCRM
PAGE 2
SPAZ=ABS(ASCALE*SPAZ+MEAN+VARPAZ*GAUSS(ID1))  0023640
SMAZ=ABS(ASCALE*SMAZ+MEAN+VARMAZ*GAUSS(ID6))  0023650
ID2=NN(2)  0023660
ID7=NN(7)  0023670
SPEL=ABS(ASCALE*SPEL+MEAN+VARPEL*GAUSS(ID2))  0023680
SMEL=ABS(ASCALE*SMEL+MEAN+VARMEL*GAUSS(ID7))  0023690

STEP 2-5: COMPUTE AZ AND EL DISCRIMINANT COMPONENTS.
AZDISC=10.* ALOG10(SPAZ/SMAZ)  0023710
ELDISC=10.* ALOG10(SPEL/SMEL)  0023720
AZDISC=0.  0023730
ELDISC=0.  0023740

STEP 3: COMPUTE RANGE DISCRIMINANT (INCLUDES NOISE) •

STEP 3-1: COMPUTE RANGE DISCRIMINANT COMPONENT SCALE FACTOR.
20 RSSCALE=S1*PDIV(IMODE)  0023750

STEP 3-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR RANGE DISCRIMINANT.
MEAN=PDIV(IMODE)  0023760
VARELY=SQRT(2.*SI*EARLY+I.)  0023770
VARMEL=SQRT(2.*SI-LATE+I.)  0023780

STEP 3-3: ADD EQUIVALENT NOISE TO RANGE DISCRIMINANT COMPONENT SIGNALS.
ID3=NN(3)  0023790
ID8=NN(8)  0023800
EARLY=ABS(RSCALE*EARLY+MEAN+VARELY,GAUSS(ID3))  0023810
LATE=ABS(RSCALE,LATE+MEAN+VARMEL,GAUSS(ID8))  0023820

STEP 3-4: COMPUTE RANGE DISCRIMINANT.
RDISC=ALOG(LATE/EARLY)  0023830

STEP 4: COMPUTE VELOCITY DISCRIMINANT (INCLUDES NOISE) •

STEP 4-1: COMPUTE VELOCITY DISCRIMINANT COMPONENT SCALE FACTOR.
VSCALE=SI*PDIV(IMODE)  0023840

STEP 4-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR VELOCITY DISCRIMINANT COMPONENTS.
MEAN=PDIV(IMODE)  0023850
VARDF2=SQRT(2.*SI,DF2+I.)  0023860
VARDF4=SQRT(2.*SI,DF4+I.)  0023870

STEP 4-3: ADD EQUIVALENT NOISE TO VELOCITY DISCRIMINANT COMPONENT SIGNALS.
ID4=NN(4)  0023880
ID9=NN(9)  0023890
DF2=ABS(VSCALE*DF2+MEAN+VARDF2*GAUSS(ID4))  0023900
DF4=ABS(VSCALE*DF4+MEAN+VARDF4*GAUSS(ID9))  0023910

STEP 4-4: COMPUTE VELOCITY DISCRIMINANT.
VDISC=ALOG10(DF2/DF4)  0023920

STEP 5: COMPUTE ON-TARGET DISCRIMINANT — USED FOR BREAK-TRACK AND VELOCITY DATA INVALID DETERMINATION •

FIGURE 2.3-10 BASELINE VERSION OF SUBROUTINE DISCRM
PAGE 3

2-72
STEP 5-1: COMPUTE STATISTICS OF ADDITIVE NOISE FOR OUTER DOPPLER FILTER SIGNALS.

\[
\begin{align*}
\text{VARDF1} &= \text{SORT}(2. \cdot \text{SI} \cdot \text{DF1} + 1.) \\
\text{VARDF5} &= \text{SORT}(2. \cdot \text{SI} \cdot \text{DF5} + 1.)
\end{align*}
\]

STEP 5-2: ADD EQUIVALENT NOISE TO OUTER DOPPLER FILTER SIGNALS.

\[
\begin{align*}
\text{IDS} &= \text{NN}(5) \\
\text{ID10} &= \text{NN}(10) \\
\text{DF1} &= \text{ABS}(\text{VSCALE} \cdot \text{DF1} + \text{MEA} + \text{VARD1} \cdot \text{GAUSS}(\text{ID5})) \\
\text{DF5} &= \text{ABS}(\text{VSCALE} \cdot \text{DF5} + \text{MEAN} + \text{VARD5} \cdot \text{GAUSS}(\text{ID10}))
\end{align*}
\]

STEP 5-3: COMPUTE ON-TARGET DISCRIMINANT.

NOTE: THE FACTOR OF \(\text{SORT}(2.)\) IS DUE TO THE METHOD OF NORMALIZATION OF DISCRIMINANT COMPONENTS.

\[
\text{ODISC} = 10. \cdot \text{ALOG10}((\text{EARLY} + \text{LATE}) \cdot \text{SORT}(2.)/(\text{DF1} + \text{DF5}))
\]

NOTE: DEBUGGING PRINT STATEMENTS.

\[
\begin{align*}
\text{WRITE}(6,902) &= \text{AZDISC,ELDISC,RDISC,VDISC,ODISC} \\
\text{WRITE}(6,903) &= \text{SNRD,SIGDB,SIGBAR} \\
\text{WRITE}(6,904) &= \text{SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE} \\
\text{WRITE}(6,905) &= \text{DF1,DF5,DF2,DF4,SIGBAR}
\end{align*}
\]

\[
\begin{align*}
902 \text{ FORMAT}(/' & \text{AZD,ELD,RD,VD,OD} = ',5F14.6) \\
903 \text{ FORMAT}(/' & \text{SNRD,SIGDB,SIGBAR} = ',3F14.6) \\
904 \text{ FORMAT}(/' & \text{SPZ,SMZ,SPS,SMEL,E,L+NOISE} = ',6F10.2) \\
905 \text{ FORMAT}(/' & \text{DF1,DF5,DF2,DF4,SIG+NOISE} = ',5F10.2)
\end{align*}
\]

RETURN

END

FIGURE 2.3-10 BASELINE VERSION OF SUBROUTINE DISCRM

PAGE 4

2-73
SUBROUTINE DISCRM

REAL LATE, MEAN

COMMON /OUTPUT/ MSWF, MTF, MSF, SRNG, SRDOT, SPANG, SRANG, SPRTE,
        SRTE, SRSS, MADVF, MRDF, MARDVF, MRRDF

COMMON /CNTL/ IPWR, IMODE, ITXP, IASM, IDUMC(5), DUMC(3)

COMMON /ICNTRL / I3DUM(14), MRNG, MSAM, MPRF, IDUM4(10)

COMMON /SYSDAT / TSAM, OR(3), CP, SP, PS1, PSBIA, ALBIA, BTBIA, GP, GA,
        DUMC(3)

COMMON /TDAT/NT, DUM5(546), CRNGE, CGVEL

COMMON /DSCRM/ AZDISC, ELDISC, RDISC, VRRT, ODISC, SGBR1, SNRD,
        SIGDB

COMMON /SIGDAT/ SPAZ, SNAZ, SPEL, SML, EARLY, LATE, DF1, DF5

COMMON /NOISE/ NS1, NS2, NN(10), GAUSS(320)

COMMON /AGCDAT/ AGCCO, AGCCDB, SNRD, SNRDTD

DIMENSION NFREQ(2), PDIA(2), PDIR(2), PDIV(2), PS(10, 2), BN(2), PT(3)

DIMENSION QNV(2)

NOTE: DEBUGGING PRINT STATEMENTS.

WRITE(6,900) SPAZ, SNAZ, SPEL, SMEL, EARLY, LATE

WRITE(6,990) DF1, DF5, DF2, DF4, SIGBAR

900 FORMAT(' SPZ, SNAZ, SPEL, SMEL, EARLY, LATE ')

990 FORMAT(' DF1, DF5, DF2, DF4, SIGBAR')
C STEP 1-1: COMPUTE CONSTANT (NOTE: IT IS DIFFERENT FOR ACTIVE AND PASSIVE MODES).
IF(IMODE.EQ.2) GO TO 5
C NOTE: THIS IS THE CONSTANT USED IN ACTIVE MODE.
YY=GP+PS(MRNG,IMODE)/(CGRNGE+BN(MSAM))
S1=YY/FLOAT(NFREQ(IMODE))
GO TO 10
C NOTE: THIS IS THE CONSTANT USED IN PASSIVE MODE.

C CONTINUE
PTFIX=PT(ITXP)
IF(SRNG.LT.64.)PTFIX=4.2
IST=0
IF(IST.EQ.1)PTFIX=4.2

YY=GP+PS(MRNG,IMODE)*PTFIX /(CGRNGE+BN(MSAM))
S1=YY/FLOAT(NFREQ(IMODE))

C STEP 1-2: COMPUTE PEAK SIGNAL POWER TO AVERAGE THERMAL NOISE POWER AT DOPPLER FILTER OUTPUT.
SNRDT=YY+SIGBAR
WRITE(6,221)YY,SIGBAR
221 FORMAT('YY,SIGBAR -',2F14.5)
SNRD=YY+SIGBAR
ALOG(SNRD) =

C STEP 1-3: COMPUTE PEAK SIGNAL POWER TO TOTAL (THERMAL PLUS QUANTIZATION) NOISE POWER AT THE DOPPLER FILTER OUTPUT.
CALL SATNSE(SNF)
XX=SNF*AGCO
XX=XX/(XX+QNV(MSAM))
S1=YY+XX
YY=YY+XX
SNRD=YY+SIGBAR
SNRD=SNRD+ALOG(SNRD)

C STEP 1-4: UPDATE NOISE SEQUENCE.
NN(1)=MOD(NN(1)+3.28)+1
DO 15 I=2,10
15 NN(I)=MOD(NN(I-1)+29.320)+1
ID=NN(I)
GAUSS(ID) =ANORM(NS1,NS2)

C **************************************************************
C STEP 2: COMPUTE ANGLE DISCRIMINANT (INCLUDES NOISE)
C **************************************************************
C STEP 2-1: CHECK ANTENNA STEERING MODE — SKIP STEP 2 IF IN GPC-DES OR MANUAL.
C CCCMOD MFB 16 1983 CCCCCCCCCMOD IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 20
C STEP 2-2: COMPUTE ANGLE DISCRIMINANT COMPONENT SCALE FACTOR.
C AScale=S1*PDIA(IMODE)
C STEP 2-3: COMPUTE STATISTICS OF ADDITIVE NOISE FOR ANGLE DISCRIMINANT COMPONENTS.
C MEAN=PDIA(IMODE)
C VARP=SQRT(2.*S1+SPAZ+1.)
C VARMZ=SQRT(2.*S1+SMAZ+1.)

FIGURE 2.3-11 DELIVERABLE VERSION OF SUBROUTINE DISCRM PAGE 2
VARPEL = \sqrt{(2. \cdot S1 \cdot SPEL + 1.)}

VARMEL = \sqrt{(2. \cdot S1 \cdot SMEL + 1.)}

C

STEP 2-4: ADD EQUIVALENT NOISE TO ANGLE DISCRIMINANT COMPONENT SIGNALS.

ID6 = NN(6)

SPAZ = ABS(ASCALe \cdot SPAZ + MEAN + VARPAZ \cdot GAUSS(ID1))

SMAZ = ABS(ASCALe \cdot SMAZ + MEAN + VARMZ \cdot GAUSS(ID6))

ID2 = NN(2)

ID7 = NN(7)

SPEL = ABS(ASCALe \cdot SPEL + MEAN + VARPEL \cdot GAUSS(ID2))

SMEL = ABS(ASCALe \cdot SMEL + MEAN + VARMEL \cdot GAUSS(ID7))

C

STEP 2-5: COMPUTE AZ AND EL DISCRIMINANT COMPONENTS.

AZDISC_1 = \text{ALOG} (SPAZ / SMAZ)

ELDISC_1 = \text{ALOG} \left(\frac{SPEL}{SMEL}\right)

C

STEP 3: COMPUTE RANGE DISCRIMINANT (INCLUDES NOISE)

C

STEP 3-1: COMPUTE RANGE DISCRIMINANT COMPONENT SCALE FACTOR.

20 RSACLE = S1 \cdot PDIV(IMODE)

C

STEP 3-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR RANGE DISCRIMINANT.

MEAN = PDIV(IMODE)

VARELY = \sqrt{(2. \cdot S1 \cdot EARLY + 1.) \cdot TCON}

VARLTE = \sqrt{(2. \cdot S1 \cdot LATE + 1.) \cdot TCON}

C

STEP 3-3: ADD EQUIVALENT NOISE TO RANGE DISCRIMINANT COMPONENT SIGNALS.

ID3 = NN(3)

ID8 = NN(8)

EARLY = ABS(RSCALE \cdot EARLY + MEAN + VARELY \cdot GAUSS(ID3))

LATE = ABS(RSCALE \cdot LATE + MEAN + VARLTE \cdot GAUSS(ID8))

C

STEP 3-4: COMPUTE RANGE DISCRIMINANT.

RDISC = \text{ALOG} (LATE / EARLY)

C

STEP 4: COMPUTE VELOCITY DISCRIMINANT (INCLUDES NOISE)

C

STEP 4-1: COMPUTE VELOCITY DISCRIMINANT COMPONENT SCALE FACTOR.

VSCALE = S1 \cdot PDIV(IMODE)

C

STEP 4-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR VELOCITY DISCRIMINANT COMPONENTS.

MEAN = PDIV(IMODE)

VARDF2 = \sqrt{(2. \cdot S1 \cdot DF2 + 1.)}

VARDF4 = \sqrt{(2. \cdot S1 \cdot DF4 + 1.)}

C

STEP 4-3: ADD EQUIVALENT NOISE TO VELOCITY DISCRIMINANT COMPONENT SIGNALS.

ID4 = NN(4)

ID9 = NN(9)

DF2 = ABS(VSCALE \cdot DF2 + MEAN + VARDF2 \cdot GAUSS(ID4))

DF4 = ABS(VSCALE \cdot DF4 + MEAN + VARMF4 \cdot GAUSS(ID9))

C

STEP 4-4: COMPUTE VELOCITY DISCRIMINANT.

VDISC = \text{ALOG} (DF2 / DF4)

FIGURE 2.3-11 DELIVERABLE VERSION OF SUBROUTINE DISCRM
• STEP 5: COMPUTE ON-TARGET DISCRIMINANT — USED FOR BREAK-• TRACK AND VELOCITY DATA INVALID DETERMINATION •

STEP 5-1: COMPUTE STATISTICS OF ADDITIVE NOISE FOR OUTER DOPPLER FILTER SIGNALS.

VARDF1=SQRT(2.*S1*DF1+1.)
VARDF5=SQRT(2.*S1*DF5+1.)

STEP 5-2: ADD EQUIVALENT NOISE TO OUTER DOPPLER FILTER SIGNALS.

IDS=NN(5)
ID10=NN(10)
DF1=ABS(VSCALE*DF1+MEAN+VARDF1*GAUSS(ID5))
DF5=ABS(VSCALE*DF5+MEAN+VARDF5*GAUSS(ID10))

STEP 5-3: COMPUTE ON-TARGET DISCRIMINANT.

NOTE: THE FACTOR OF SORT(2.) IS DUE TO THE METHOD OF NORMALIZATION OF DISCRIMINANT COMPONENTS.

ODISC=10.*ALOG10(((EARLY+LATE)*SORT(2.))/(DF1+DF5))

WRITE(6,982) AZDISC,ELDISC,RDISC,VDISC,ODISC
WRITE(6,g83) SNRD.SIGDB,SIGBAR
WRITE(6,g84) SPAZ,SMZ,SPEL,SMEL,EARLY,LATE
WRITE(6,985) DF1,DF5,DF2,DF4.SIGBAR

RETURN
END

FIGURE 2.3-II DELIVERABLE VERSION OF SUBROUTINE DISCRM

PAGE 4

2-77
LINES DELETED FROM BASELINE PROGRAM
40    DATA NFREQ/1.5/,BN/972.4,616.6/,PS/9*1.2..5*1.2..4..8..8..16./00022930
41    2    PDIA.PDIR.PDIV/1.4142,3.1623,2.0.4,4721,2.8284,6.3246/,00022940
42    3    PT/42658.3125.,195.3/,QNV/.04166666/
43    DATA TDC/0.85122118,0.1195161,0.2561557/

LINES ADDED TO DELIVERABLE PROGRAM
40    DIMENSION QNV(2)
41    C
42    C
43    C
44    DATA NFREQ/1.5/,BN/972.4,616.6/
45    DATA PS/9*4..2..5*4..2..4..8..8..16./
46    2    PDIA.PDIR.PDIV/1.4142,3.1623,2.0.4,4721,2.8284,6.3246/,00022940
47    3    PT/42658.3125.,195.3/
48    DATA QNV/.00067.,011/
49    DATA TDC/0.85122118,0.1195161,0.2561557/

LINES DELETED FROM BASELINE PROGRAM
81    C    WRITE(6,221)YY,SIGBAR
82    C
83    C    WRITE(6,221)YY,SIGBAR
84    SNRDT=ALOG10(SNRDT)
85    221    FORMAT('YY,SIGBAR=',F14.5)
86    SNRDT=ALOG10(SNRDT)
87    221    FORMAT('YY,SIGBAR=',F14.5)
88    221    FORMAT('YY,SIGBAR=',F14.5)
89    SNRDT=ALOG10(SNRDT)
90    SNRDT=ALOG10(SNRDT)

LINES ADDED TO DELIVERABLE PROGRAM
87    C    WRITE(6,221)YY,SIGBAR
88    221    FORMAT('YY,SIGBAR=',F14.5)
89    SNRDT=ALOG10(SNRDT)
90    SNRDT=ALOG10(SNRDT)

LINES DELETED FROM BASELINE PROGRAM
93    XX/XX/(XX+QNV)
94    S1=S1+XX
95    00023296
96    00023300

LINES ADDED TO DELIVERABLE PROGRAM
99    XX=XX/(XX+QNV(MSAM))
100    S1=S1+XX
101    00023300

Number of difference sections found: 3
Number of difference records found: 12
DIFFERENCES /IGNORE=/MERGED=1/OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF4.FOR;1-
SYS$DISK3:[MCCOLLOUGH]DISCRMH.FOR;2-
SYS$DISK3:[MCCOLLOUGH]DISCRMF.FOR;2

FIGURE 2.3-12 SUMMARY OF MODIFICATIONS TO SUBROUTINE DISCRM

2-78
2.3.4.2 Test Results

The first test run with a RCS of 10 dBsm showed large (1 dB) discontinuities in the RSS at the sample rate changes. An examination of the RSS equation expressed in terms of \( SNR_{vt} \) and thermal noise power \( N_t \) showed that this result was not predicted. Consider the expression for RSS,

\[
(2-14) \quad RSS = 10 \log(SNR_{vt} + 1/G) + 10 \log(N_t/4q^2)
\]

Now, the sample rate change causes a 12 dB change in \( SNR_{vt} \) and a 12 dB change in thermal noise power \( N_t \) which offsets the \( SNR_{vt} \) change. Furthermore, at this range the \( SNR_{vt} \) is on the order of \( 10^4 \) and \( 1/G \) is \( 1/16 \) therefore the sample rate change shouldn't have introduced a discontinuity, but a discontinuity appeared in the data. The following AGC equations were then examined to determine an answer to this unexpected result:

\[
(2-15) \quad RSS = 10 \log(1/AGC)
\]

\[
(2-16) \quad AGC(N+1) = AGC(N)AGCERR(N)
\]

\[
(2-17) \quad AGCERR(N) = k_1 G/(AGC(N)(SNRDT(N)+1)+k_2)
\]

where \( G = \) signal to noise ratio gain from doppler filter.

\[ k_1 = (2q)^2/N_t \]
\[ k_2 = (q)^2/(12 N_t) \]

\( N_t \) un AGC'd thermal noise at the A/D input

It is seen that the variables \( k_1 \) and \( k_2 \) are functions of the thermal noise power \( N_t \). Therefore, since the thermal noise power changes by a factor of the ratios of the noise bandwidth of the high sample rate video filter to the noise bandwidth of the low sample rate video filter, the ratios between \( k_1 \) (high sample rate) and \( k_1 \) (low sample rate) and the ratio between \( k_2 \) (high sample rate) and \( k_2 \) (low sample rate) should be precisely the ratios of the noise bandwidth. In Table 2-1 (from page 2-3 of Reference 1), from which the values of \( k_1 \) and \( k_2 \) were taken, this was not true. The
appropriate modification of the variables $k_1$ and $k_2$ and subsequent simulation run showed that this solved the problem (see Figure 2.3-13) and that the RSS behaved as expected over the entire trajectory.

A third run of the simulation was then made with the RCS set to $-40$ dBsm. The output RSS plot (Figure 2.3-14) has discontinuities in the RSS at both high-to-low and low-to-high sample rate changes. Examination of equation 2-14 shows that this should be expected in both cases. Consider the high sample rate-to-low sample rate transition. The $\text{SNR}_v t$ in the high sample rate mode is less than 1. Therefore the $\text{SNR}_v t$ is on the same order of magnitude as $1/G$. Now, switching to the low sample rate mode increases $\text{SNR}_v t$ by 12 db and decrease the thermal noise power by 12 db. Although $\text{SNR}_v t$ changes by 12 db the change in the term $10 \log(\text{SNR}_v t + 1/G)$ is less than 12 db because $\text{SNR}_v t$ is on the same order as $1/G$ in the high sample rate mode. For the low sample rate-to-high sample rate case, the mechanism producing the discontinuity is the same except that the $\text{SNR}_v t$ decreases by 12 db and the noise power increases 12 db.

2.4 RADAR PROCESSING PARAMETER CHANGES

2.4.1 Problem Definition

Problems documented in this section were precipitated by several modifications in the radar design during the system test phase of the radar development. These modifications included changes in pulsewidth, PRF, and transmit power transition points. In addition, the original simulation model neglected to include the hysteresis loops governing the sample rate transition point and the PRF transition point. While ignoring the hysteresis loop produces only very minor performance error, the addition of this loop was a minor operation and was therefore included in the modifications package.
FIGURE 2.3-13 SIMULATED RADAR SIGNAL STRENGTH
RADAR CROSS SECTION = +10 dBsm
FIGURE 2.3-14  SIMULATED RADAR SIGNAL STRENGTH
RADAR CROSS SECTION = -40 dBsm

MEAN = 0.075       STANDARD DEVIATION = 0.105
2.4.2 Algorithm Modifications

Modifications to this algorithm include the following items:

- Moving the 7-kHz to 3-kHz PRF transition point from 9.8 nautical miles into 8.2 nautical miles.
- Adding hysteresis to the 7-kHz to 3-kHz PRF transition.
- Adding hysteresis to the high sample rate-to-low sample rate transition point.
- Updating the range interval boundary table.

Figure 2.4-1 provides an illustration of the hysteresis loop applied to the 7-kHz to 3-kHz PRF transition. Figures 2.4-2 defines the hysteresis loop applied to the sample rate transition. Also, the range interval boundaries were updated to accurately reflect those used in the radar processor. Table 2.4-1 summarizes the new boundaries and the track mode pulsewidth associated with those boundaries.

2.4.3 Software Design Documentation

The changes described in Sections 2.4.1 and 2.4.2 were implemented through modifications to subroutine CNTRLS. The modifications included:

- Modifying four lines of existing code, and adding code to simulate the hysteresis loop for sampling rate transition.
- Modifying four lines of existing code, and adding code to simulate the hysteresis loop for Pulse Repetition Frequency (PRF) transition.

2-83
FIGURE 2.4-1 HYSTERESIS LOOP FOR PRF TRANSITION
FIGURE 2.4-2 HYSTERESIS LOOP FOR SAMPLE RATE TRANSITION
TABLE 2.4-1 NEW RANGE INTERVAL BOUNDARIES

<table>
<thead>
<tr>
<th>MRNG*</th>
<th>RANGE INTERVAL, FEET</th>
<th>PULSEWIDTH, SEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>0.122</td>
</tr>
<tr>
<td>2</td>
<td>120 - 630</td>
<td>0.122</td>
</tr>
<tr>
<td>3</td>
<td>640 - 1510</td>
<td>0.122</td>
</tr>
<tr>
<td>4</td>
<td>1520 - 2550</td>
<td>0.122</td>
</tr>
<tr>
<td>5</td>
<td>2560 - 5750</td>
<td>2.07</td>
</tr>
<tr>
<td>6</td>
<td>5760 - 11510</td>
<td>4.15</td>
</tr>
<tr>
<td>7</td>
<td>11510 - 23030</td>
<td>8.3</td>
</tr>
<tr>
<td>8</td>
<td>23040 - 43510</td>
<td>16.6</td>
</tr>
<tr>
<td>9</td>
<td>43520 - 49910</td>
<td>33.2</td>
</tr>
<tr>
<td>10</td>
<td>49920 - 1.82E-6</td>
<td>33.2</td>
</tr>
</tbody>
</table>

It should be noted that minor changes to the values of constants were made in the main program and the subroutines DISCRM, RTRACK, SIGNAL, and RSS to accommodate the changes made to CNTRLS. These changes are minor, and are documented in Sections 2.2 and 2.3, so they will not be repeated here.

Figure 2.4-3 is a listing of the baseline version of CNTRLS. Figure 2.4-4 is a listing of the deliverable version of CNTRLS. The differences between the baseline and deliverable subroutines are listed in Figure 2.4-5.

2.4.4 Integration and Test Data

Testing of the high-sample to low-sample rate hysteresis loop defined in Figure 2.4-2 consisted of using the following scenario in the simulation. A 10 dBsm target was moved in range from 2400 feet to 4000 feet.
SUBROUTINE CNTRLS
REAL INTT,NFIL,IRNG,IRDOT
COMMON /CNTL/IPC_.IMODE.
IDUMC(7),DUMC(3)
COMMON /OUTPUT/IDT,L_(3),SRNG,SRDOT,DUM2(5),IDUM(4)
COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUMl(10),MPFOLD
COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
DIMENSION RI(4).FW(3)

C Ri(4) CHANGED TO 2560 FROM 2552
DATA RI/120.,240.,780.,2560.,5772.,11544.,23089.,43747.,
2 57722.,1.8228E+6/
DATA FW/7.7215,3.3890,0.2969/,NRI/1e/

C * STEP 1: SET RANGE INTERVAL PARAMETER *
XRNG=IRNG*0.3125
DO 60 I=1,NRI
60 CONTINUE
IF(XRNG.LE.RI(I)) GO TO 70
MRNG=I
IF(MRNG.GT.NRI) STOP

C * STEP 2: SET SAMPLE RATE PARAMETER *
IF(IMODE.GE.2) GO TO 74
IF(MRNG.GT.9) GO TO 72
MSAM=1
GO TO 80
72 MSAM=2
GO TO 80
74 MSAM=1
GO TO 80
76 MSAM=2

C * STEP 3: SET PRF PARAMETER *
STEP 3-1: DETERMINE IF IN ACTIVE OR PASSIVE MODE.
80 IF(IMODE.GE.2) GO TO 84
C
STEP 3-2: DETERMINE CORRECT PRF FOR GIVEN OPERATING MODE.

FIGURE 2.4-3 BASELINE VERSION OF SUBROUTINE CNTRLS
PAGE 1
2-87
IF(MRNG.GT.9) GO TO 82
  MPRF=1
  GO TO 90
82  MPRF=3
  GO TO 90
84  IF(MRNG.GT.9) GO TO 86
  MPRF=1
  GO TO 90
86  MPRF=2
  CONTINUE
C
C STEP 3-3: IF PRF HAS CHANGED FROM PREVIOUS DATA CYCLE, THEN
  RESET THE 5 DOPPLER TRACKING FILTERS ACCORDINGLY.
  IF(MPFOLD.EQ.MPRF) GO TO 96
    NFIL=INTT((-SRDOT/FW(MPRF))+0.5)+31998.
    XX=AMOD(NFIL,32.)
    MDF(1)=INT(XX)
    DO 95 I=1,4
      MDF(I+1)=MOD(MDF(1)+I,32)
95  MPFOLD=MPRF
C
C NOTE: DEBUGGING PRINT STATEMENTS.
C
WRITE(6,999) MPRF,MPFOLD,MDF(1)
999 FORMAT(' MPRF,MPFOLD,MDF1 =',318)
RETURN
END

FIGURE 2.4-3 BASELINE VERSION OF SUBROUTINE CNTRLS

PAGE 2

2-88
SUBROUTINE CNTRLS
REAL IRRNG,IRDOT
COMMON /CINTL/IPWR,IMODE,IDUMC(7),DUMC(3)
COMMON /OUTPUT/IDUMO(3),SRNG,SRDOT,DUM2(5),IDUM(4)
COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDUM1(10),M9FD
DIMENSION RI(10),FW(3)
DATA RI/120.,640.,1520.,2560.,5760.,11520.,23040.,43520.,
  249920.,1.8228E+6/
DATA FW/7.7215,3.3090,E.2969/,NRI/le/

C IMPLEMENTATION OF Hysterisis FOR THE SAMPLING RATE
CHANGE AND FOR THE PRF CHANGE ALONG WITH CHANGES IN
RI(RANGE INTERVAL) WAS COMPLETED FEB 6,1986 BY M. MEYER

C STEP 1: SET RANGE INTERVAL PARAMETER
XRNG=IRNG+.3125
DO 60 I=1,NRI
  IF(XRNG.LE.RI(I)) GO TO 70
60 CONTINUE
70 MRNG=I
  IF(MRNG.GT.NRI) STOP

C STEP 2: SET SAMPLE RATE PARAMETER
IF(IMODE.GE.2) GO TO 74
IF(MRNG.GT.9) GO TO 72
MSAM=1
GO TO 80
72 MSAM=2
GO TO 80
74 MSAM=1

C----- MODIFIED FEB 6 1986 BY M. MEYER

FIGURE 2.4-4 DELIVERABLE VERSION OF SUBROUTINE CNTRLS
**FIGURE 2.4-4** DELIVERABLE VERSION OF SUBROUTINE CNTRLs

PAGE 2

2-90
FIGURE 2.4-4  DELIVERABLE VERSION OF SUBROUTINE CNTRLS

PAGE 3

2-91
LINES DELETED FROM BASELINE PROGRAM
29 DATA RI/120..240.,780..2560.,5772..11544..23089..43747..
30 2 .7722..1.8228E+6/
31 DATA FW/7.7215.3.3096.0.2969/.NRI/10/
32 C

LINES ADDED TO DELIVERABLE PROGRAM
29 DATA RI/120..640.,1520..2560.,5760..11520..23040..43520..
30 2 4928..1.8228E+6/
31 DATA FW/7.7215.3.3096.0.2969/.NRI/10/
32 C

IMPLEMENTATION OF HYSTERESIS FOR THE SAMPLING RATE
CHANGE AND FOR THE PRF CHANGE ALONG WITH CHANGES IN
RI(RANGE INTERVAL) WAS COMPLETED FEB 6,1986 BY M. MEYER

LINES DELETED FROM BASELINE PROGRAM
52 74 IF(MRNG.GT.4) GO TO 76
53 MSAM=1
54 GO TO 80
55 76 MSAM=2
56 C

LINES ADDED TO DELIVERABLE PROGRAM
59 C***** MODIFIED FEB 6 1986 BY M. MEYER**************
60 74 IF(MSAM.EQ.1)THEN
61 IF(XRNG.GT.3200.)THEN
62 MSAM=2
63 ELSE
64 MSAM=1
65 C***** MODIFIED FEB 17,1986 BY M. MEYER **************
66 C***** GUARANTEES THE CORRECT LOOP BANDWIDTHS**********
67 C***** FOR THE HYSTERESIS LOOP***********************
68 C
69 IF(XRNG.GT.2560) MRNG=4
70 C
71 C*******************************************************************
72 END IF
73 ELSE
74 IF(XRNG.GT.2560.)THEN
75 MSAM=2
76 ELSE
77 MSAM=1
78 END IF
79 END IF

FIGURE 2.4-5 SUMMARY OF MODIFICATIONS TO SUBROUTINE CNTRLs
PAGE 1

2-92
LINES DELETED FROM BASELINE PROGRAM
70  84 IF(MRNG.GT.9) GO TO 86
71   MPRF=1
72   GO TO 90
73  86 MPRF=2
74  90 CONTINUE

LINES ADDED TO DELIVERABLE PROGRAM
94  84 IF(MPRF.EQ.1)THEN
95   IF(XRNG.GT.49920.)THEN
96     MPRF=2
97     ELSE
98     MPRF=1
99   ELSE
100    END IF
101   ELSE
102    IF(XRNG.GT.43520.)THEN
103     MPRF=2
104     ELSE
105     MODIFIED FEB 6 1986 BY M. MEYER
106     MODIFIED FEB 17, 1986 BY M. MEYER
107     GNUARANTEES THE CORRECT CONSTANTS
108     FOR THE LOW PRF
109
110     ELSE
111     ELSE
112     END IF
113     END IF
114    90 CONTINUE

Number of difference sections found: 3
Number of difference records found: 52

Differences /Ignore=1/Merged=1/Output=SYS$DISK3:[MCCOLLOUGH]DIFF6.FOR;1-
SYS$DISK3:[MCCOLLOUGH]CNTRLSH.FOR;2-
SYS$DISK3:[MCCOLLOUGH]CNTRLSF.FOR;2

FIGURE 2.4-5 SUMMARY OF MODIFICATIONS TO SUBROUTINE CNTRLs
PAGE 2
2-93
with a speed of 50 feet per second and then the target range was decreased from 4000 feet to 1400 feet at a speed of 80 feet per second. As this scenario was executed, the following parameters were output: time, range, and the sample rate control parameter, MSAM. MSAM=1 corresponds to the high sample rate, while MSAM=2 corresponds to the low sample rate. Table 2.4-2 provides a summary of the test results. A comparison with Figure 2.4-2 shows that the simulation code is performing to the design.

The test to validate the operation of hysteresis in the 7-kHz to 3-kHz PRF transition was similar to the sample rate hysteresis test. In this case, a 10 dBsm target was moved in range from 42,000 feet to 53,000 feet at a speed of 50 feet per second and then the range was decreased from 53,000 feet to 38,000 feet at a speed of 76 feet per second. In this case, the following data was output as the simulation progressed: time, range, and the PRF control parameter, MPRF. Table 2.4-3 defines MPRF. Results of the test are summarized in Table 2.4-4. A comparison of these results with Figure 2.4-1 shows that the new code is performing as required.

2.5 VELOCITY PROCESSOR CHANGES

2.5.1 Problem Definition

The changes in the velocity processor module consisted of removing the range rate ambiguity resolver in the 7 kHz PRF mode and correcting a bug that was traced to this module. Removal of the ambiguity resolver is a direct result of changes to the radar following system test. The bug in the velocity processor module software was uncovered when the trajectories from the SORTE experiments were used to drive the simulation. One of these trajectories produced an unexpected glitch in the range rate. A subsequent investigation pointed to a problem in addressing the model of the PROM used to convert the velocity discriminant value to a velocity estimate. The problem was fixed and is documented in the following subsections.
### TABLE 2.4-2 SAMPLE RATE TRANSITION HYSTERESIS LOOP TEST RESULTS

<table>
<thead>
<tr>
<th>TIME, SEC</th>
<th>RANGE, FT</th>
<th>MSAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.39999</td>
<td>3141.250</td>
<td>1</td>
</tr>
<tr>
<td>14.79999</td>
<td>3166.875</td>
<td>1</td>
</tr>
<tr>
<td>15.19999</td>
<td>3181.563</td>
<td>1</td>
</tr>
<tr>
<td>15.59999</td>
<td>3205.625</td>
<td>2</td>
</tr>
<tr>
<td>15.99999</td>
<td>3222.188</td>
<td>2</td>
</tr>
<tr>
<td>16.39999</td>
<td>3244.063</td>
<td>2</td>
</tr>
<tr>
<td>16.79999</td>
<td>3263.438</td>
<td>2</td>
</tr>
<tr>
<td>54.00005</td>
<td>2670.625</td>
<td>2</td>
</tr>
<tr>
<td>54.40005</td>
<td>2640.000</td>
<td>2</td>
</tr>
<tr>
<td>54.80006</td>
<td>2612.500</td>
<td>2</td>
</tr>
<tr>
<td>55.20006</td>
<td>2578.125</td>
<td>2</td>
</tr>
<tr>
<td>55.60006</td>
<td>2544.688</td>
<td>1</td>
</tr>
<tr>
<td>56.00006</td>
<td>2509.688</td>
<td>1</td>
</tr>
<tr>
<td>56.40006</td>
<td>2479.063</td>
<td>1</td>
</tr>
</tbody>
</table>

### TABLE 2.4-3 DEFINITION OF MPRF

<table>
<thead>
<tr>
<th>MPRF</th>
<th>PRF, HZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6969</td>
</tr>
<tr>
<td>2</td>
<td>2980</td>
</tr>
<tr>
<td>3</td>
<td>268</td>
</tr>
<tr>
<td>TIME, SEC</td>
<td>RANGE, FT</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>154.3999</td>
<td>49735.63</td>
</tr>
<tr>
<td>154.7999</td>
<td>49770.00</td>
</tr>
<tr>
<td>155.1999</td>
<td>49777.19</td>
</tr>
<tr>
<td>155.5999</td>
<td>49841.56</td>
</tr>
<tr>
<td>155.9999</td>
<td>49839.69</td>
</tr>
<tr>
<td>156.3999</td>
<td>49860.63</td>
</tr>
<tr>
<td>156.7999</td>
<td>49919.38</td>
</tr>
<tr>
<td>157.1999</td>
<td>49906.25</td>
</tr>
<tr>
<td>157.5999</td>
<td>49914.38</td>
</tr>
<tr>
<td>157.9999</td>
<td>49934.38</td>
</tr>
<tr>
<td>158.3999</td>
<td>49952.81</td>
</tr>
<tr>
<td>158.7999</td>
<td>49991.88</td>
</tr>
<tr>
<td>159.1999</td>
<td>50003.75</td>
</tr>
<tr>
<td>159.5999</td>
<td>49998.75</td>
</tr>
<tr>
<td>159.9998</td>
<td>50005.94</td>
</tr>
<tr>
<td>160.3998</td>
<td>50051.88</td>
</tr>
<tr>
<td>160.7998</td>
<td>50060.31</td>
</tr>
<tr>
<td>310.3985</td>
<td>43883.75</td>
</tr>
<tr>
<td>310.7985</td>
<td>43832.81</td>
</tr>
<tr>
<td>311.1985</td>
<td>43790.63</td>
</tr>
<tr>
<td>311.5984</td>
<td>43757.50</td>
</tr>
<tr>
<td>311.9984</td>
<td>43710.94</td>
</tr>
<tr>
<td>312.3984</td>
<td>43680.31</td>
</tr>
<tr>
<td>312.7984</td>
<td>43658.13</td>
</tr>
<tr>
<td>313.1984</td>
<td>43639.69</td>
</tr>
<tr>
<td>313.5984</td>
<td>43596.88</td>
</tr>
<tr>
<td>313.9984</td>
<td>43551.88</td>
</tr>
<tr>
<td>314.3984</td>
<td>43515.94</td>
</tr>
<tr>
<td>314.7984</td>
<td>43471.25</td>
</tr>
<tr>
<td>315.1984</td>
<td>43451.88</td>
</tr>
<tr>
<td>315.5984</td>
<td>43393.13</td>
</tr>
<tr>
<td>315.9984</td>
<td>43412.50</td>
</tr>
<tr>
<td>316.3984</td>
<td>43334.69</td>
</tr>
<tr>
<td>316.7984</td>
<td>43294.69</td>
</tr>
<tr>
<td>317.1984</td>
<td>43259.38</td>
</tr>
<tr>
<td>317.5984</td>
<td>43256.25</td>
</tr>
<tr>
<td>317.9984</td>
<td>43279.69</td>
</tr>
<tr>
<td>318.3983</td>
<td>43201.25</td>
</tr>
<tr>
<td>318.7983</td>
<td>43188.13</td>
</tr>
<tr>
<td>319.1983</td>
<td>43198.44</td>
</tr>
<tr>
<td>319.5983</td>
<td>43095.31</td>
</tr>
</tbody>
</table>
2.5.2 Algorithm Modifications

Removing the ambiguity resolver was straightforward. In the original algorithm, the range rate was determined by using (1) the filter number within the bank of 32 filters, (2) an estimate of the position within the given filter obtained from the velocity discriminant, and (3) the number of filter banks which is determined using an estimate of the range rate from the range tracking loop (see Figure 2.1-1). The ambiguity resolver is effectively disabled by holding the number of filter banks to zero, regardless of the range rate estimate from the tracker. In the actual implementation of the algorithm, holding the number of filter banks to zero translates to holding the variable IRVEL to a value of 4096 for opening velocities and to a value of 0 for closing velocities.

The problem with addressing the PROM which is used to convert velocity discriminant values to positions within a filter can be described as follows. There are only 128 addresses in the array representing the PROM. However, a mistake in the code that checks the discriminant (which effectively is the PROM address) allows a value of 129. If this condition is obtained, it can either cause the program to terminate or cause the velocity estimate to glitch. The latter condition was observed in one of the simulation runs. The problem was easily corrected by changing the bounds on the code that checks the velocity estimate for saturation.

2.5.3 Software Design Documentation

The changes described in Subsections 2.5.1 and 2.5.2 were implemented by making modifications to the subroutine VELPRO. In particular, code was added following STEP 1.4 in the subroutine to properly update the velocity estimate when the radar is in the 7 kHz PRF mode.

Figure 2.5-1 is a listing of the original version of VELPRO. Figure 2.5-2 is a listing of the updated, deliverable code. Finally, Figure 2.5-3 is a line-by-line summary of the differences between the two.
THIS SUBROUTINE COMPUTES AN ACCURATE, SMOOTHED VELOCITY USING...
THE KU-BAND RADAR VELOCITY PROCESSOR ALGORITHM.

SUBROUTINE VELPRO
REAL IRDOT,IRNG,INTT,IVEL,IVDISC,IFVEL,IRVEL,IR1,IR2,IR3,
2 IF3,IDEITA
COMMON /CNTL/IPt_,IMODE,IDUMC(7),DUMC(3)
COMMON /OUTPUT/IDUMe(3),SRNG,SRDOT,DUM2(5),IDUM(4)
COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IDL_1(10),MPFOLD
COMMON /SYSDAT/TS/u_,DUMS(14)
COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(S)
COMMON /DSCRM/DUM(2),RDISC,VOSC,RRTE,ODISC,DUM3(3)
DIMENSION IPROM(128),VTl(3),VT2(3),MW(4.3)
DATA IPROM/127,127,125,124,122,121,120,118,117,116,114,113,
3 111,110,109,107,106,105,103,102,101,99,98,97,95,94,93,92,90.
2 00027210
 89,88,87,85,84,83,82,81,79,78,77,76,75,73,72,71,70,69,68,67.
 4 66,65,64,63,62,61,60,59,58,57,56,55,54,53,52,51,50,49,49,48.
 5 47,46,45,44,44,43,42,41,41,40,39,38,38,37,36,36,35,34,34,33.
 7 22,21,21,20,20,19,19,19,18,18,17,17,16,16,15,15,15,15,14,14.
 8 0.5163982,0.4633489/ DATA MW/1.2.3.4.1.1.1.1.1.1.
 2 00027280
STEP 1-1: INTEGERIZE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION.
VDISC=5.333333*VDSC
IVDISC=INTT(VDISC+0.5)
IF(IVDISC.LT.-128.) IVDISC=-128.
IF(IVDISC.GT.127.) IVDISC=127.

STEP 1-2: COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS VELOCITY ESTIMATE.
INTE=MOD(2)
IF(IVDISC.LT.0.) INTE=MOD(INTE+1,32)

STEP 1-3: COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY ESTIMATE.
IF(IVVEL.LT.0.) IFRAC=IPROM(IV1)
IF(IVVEL.LT.0.) IFRAC=127-IFRAC

FIGURE 2.5-1 BASELINE VERSION OF SUBROUTINE VELPRO

PAGE 1

2-98
STEP 1-4: COMPUTE AMBIGUOUS VELOCITY ESTIMATE — COMBINE INTEGRAL AND FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF FILTER WIDTH.

\[ \text{IFVEL} = \text{FLOAT(IFRAC+128*INTEG)} \]

* STEP 2: SCALE ROUGH VELOCITY ESTIMATE *

STEP 2-1: SCALE LSB OF ROUGH RANGE RATE ESTIMATE TO 4 TIMES A DOPPLER WIDTH.

**Definition:**

\[ \text{VT1(MPRF)} = \frac{\text{RANGE LSB}}{\left(\frac{\text{MAX. UNAMBIGUOUS VELOCITY}}{8}\right)} \]

\[ \text{OR} \quad \text{VT1(MPRF)} = \frac{5}{\text{PRF*\text{LAM}}BAD(1)} \]

\[ R1 = \text{IRDOT} \cdot \text{VT1(MPRF)} / \text{TSAM} \]

\[ IRI = \text{AINT}(R1) \]

C

STEP 2-2: PERFORM SOME REQUIRED AUXILIARY CALCULATIONS.

\[ R2 = IRI / 8. \]

\[ IRI = \text{AINT}(R2) \]

\[ \text{IRVEL} = IRI + 4096. \]

* STEP 3: RESOLVE AMBIGUITY *

STEP 3-1: COMPUTE 3 MSB'S OF AMBIGUOUS VELOCITY ESTIMATE.

\[ I3 = \text{AINT}\left(\text{IFVEL} / 512.\right) \]

STEP 3-2: COMPUTE 3 LSB'S OF SCALED ROUGH RANGE RATE ESTIMATE.

\[ I3 = \text{ABS}(IR1-8. * I2) \]

\[ \text{IF}(R1 \leq 0.) \text{GO TO 10} \]

\[ \text{IRVEL} = \text{IRVEL} + 4096. \]

\[ I3 = 7. - I3 \]

10 CONTINUE

C

STEP 3-3: COMPARE 3 MSB'S AND 3 LSB'S AND INCREMENT NUMBER OF AMBIGUOUS FILTER BANK WIDTHS APPROPRIATELY.

\[ \text{IDELTA} = I3 - I3 \]

\[ \text{IF}(\text{IDELTA} \geq 4.) \text{IRVEL} = \text{IRVEL} - 4096. \]

\[ \text{IF}(\text{IDELTA} \leq -4.) \text{IRVEL} = \text{IRVEL} + 4096. \]

C

* STEP 4: COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE *

STEP 4-1: ADD NUMBER OF AMBIGUOUS FILTER BANK WIDTHS TO ESTIMATE OF FRACTIONAL FILTER BANK WIDTH. NOTE: LSB OF RESULTANT ESTIMATE REPRESENTS 1/4096 OF A FILTER BANK WIDTH.

\[ \text{IVEL} = \text{INTT}(\text{IRVEL} - \text{IFVEL}) \]

STEP 4-2: SCALE LSB OF RESULTANT ESTIMATE TO 0.05 FEET/SEC.

**Definition:**

\[ \text{VT2(MPRF)} = \frac{\text{FILTER SEPARATION}}{128.} / \left(\frac{\text{VELOCITY LSB}}{8196}\right) \]

\[ \text{OR} \quad \text{VT2(MPRF)} = \frac{\text{PRF*\text{LAM}}BAD}{\left(0.05 + 8196\right)} \]

\[ \text{IVEL} = \text{INTT}(\text{IVEL} + \text{VT2(MPRF)} + 0.5) \]

C

* STEP 5: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY *

C

STEP 5-1: UPDATE REGISTERS OF MOVING WINDOW AVERAGER.

\[ \text{DO 20 I=1,3} \]

\[ 20 \text{ VEST}(5-I) = \text{VEST}(4-I) \]

\[ \text{VEST}(1) = \text{IVEL} \]

C

FIGURE 2.5-1 BASELINE VERSION OF SUBROUTINE VELPRO

PAGE 2

2-99
STEP 5-2: COMPUTE MOVING WINDOW AVERAGE AND SCALE ANSWER INTO FEET/SEC FROM UNITS OF 0.05 FEET/SEC.

\[ M = MPRF \]

\[ M1 = M_{W1}(1.0) \]

\[ M2 = M_{W2}(2.0) \]

\[ M3 = M_{W3}(3.0) \]

\[ M4 = M_{W4}(4.0) \]

\[ SRDOT = 0.0125 \times (VEST(M1) + VEST(M2) + VEST(M3) + VEST(M4)) \]

STEP 6-1: USE ON-TARGET DISCRIMINANT AND VELOCITY DISCRIMINANT TO DETERMINE UPDATE OF FILTER BANK POSITION.

THE FOLLOWING RULES ARE USED:

CASE 1: ODISC>0 AND -51.<IVDISC<51. IMPLIES NO CHANGE.

CASE 2: ODISC>0 AND IVDISC>51. IMPLIES SHIFT -1.

CASE 3: ODISC<0 AND IVDISC<-51. IMPLIES SHIFT +1.

CASE 4: ODISC<0 AND IVDISC>0. IMPLIES SHIFT -2.

CASE 5: ODISC<0 AND IVDISC<0. IMPLIES SHIFT +2.

\[ \text{IF}(\text{ODISC} \geq 0) \text{ GO TO 30} \]

\[ \text{IF}(\text{IVDISC} \lt 0) \text{ MDF}(1) = \text{MOD}(\text{MDF}(1) + 2, 32) \]

\[ \text{IF}(\text{IVDISC} \geq 0) \text{ MDF}(1) = \text{MOD}(\text{MDF}(1) + 30, 32) \]

\[ \text{GO TO 40} \]

\[ \text{IF}(\text{IVDISC} \lt 0) \text{ MDF}(1) = \text{MOD}(\text{MDF}(1) + 31, 32) \]

\[ \text{IF}(\text{IVDISC} \geq 0) \text{ MDF}(1) = \text{MOD}(\text{MDF}(1) + 1, 32) \]

\[ \text{GO TO 40} \]

30

\[ \text{DO 50 I=1,4} \]

50

\[ \text{MDF}(I+1) = \text{MOD}(\text{MDF}(I) + 1, 32) \]

RETURN

END

FIGURE 2.5-1 BASELINE VERSION OF SUBROUTINE VELPRO

PAGE 3

2-100
THIS SUBROUTINE COMPUTES AN ACCURATE, SMOOTHED VELOCITY USING THE KU-BAND RADAR VELOCITY PROCESSOR ALGORITHM.

SUBROUTINE VELPRO
REAL IRDOT, IRNG, INTT, IVEL, IVDISC, IFVEL, IRVEL, IR1, IR2, IR3,
2 IFS, IDETA
COMMON /CNTL/IPWR, IMOOE, IDUMC(7), DUMC(3)
COMMON /OUTPUT/IDLME(3), SRNG, SRDOT, DUM(3)
COMMON /ICNTL/IIDUM(14), MRNG, MSAM, MPRF, IDUM1(10), MPFOLD
COMMON /SYSDAT/TSAM, DUMS(14)
COMMON /RTDAT/IRDOT, IRNG, RBJAS, VEST(4), MDF(5)
COMMON /DSCRM/DUM(2), RDISC, VDSC, RRTE, ODISC, DUM(3)
DIMENSION IPROM(128), VT1(3), VT2(3), MW(4, 3)
DATA IPROM/127, 127, 125, 124, 122, 121, 126, 118, 117, 116, 114, 113,
2 111, 109, 107, 106, 105, 103, 102, 101, 99, 98, 97, 95, 94, 93, 92, 90,
3 89, 88, 87, 85, 84, 83, 82, 81, 79, 78, 77, 76, 75, 73, 72, 71, 70, 69, 68, 67,
4 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47,
5 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33,
6 32, 31, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 20, 19, 18, 18, 17, 16, 16, 15, 14,
7 22, 22, 21, 21, 20, 20, 19, 19, 19, 18, 18, 17, 17, 16, 16, 15, 15, 15/0,
DATA VT1/1.012592E-2, 2.362726E-2, 2.633237E-2, 2.94633489/0,
DATA MW/1.1.2341, 1.221, 1.21, 1.2/0.

STEP 1: GENERATE AMBIGUOUS VELOCITY ESTIMATE

STEP 1-1: INTEGERIZE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION.
VDISC=INT(VDISC+6.5)
IF(IVDISC.LT.-128.) IVDISC=-128.
IF(IVDISC.GT.127.) IVDISC=127.

STEP 1-2: COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS VELOCITY ESTIMATE.
INTEG=MDF(2)

FIGURE 2.5-2 DELIVERABLE VERSION OF SUBROUTINE VELPRO
IF (IVDISC.LT.0.) INTEG=MOD(INTEG+1,32)

C STEP 1-3: COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY ESTIMATE.
C
IV1=INT(ABS(IVDISC))+1
IFRAC=IPROM(IV1)
IF (IVDISC.LT.0.) IFRAC=127-IFRAC

C CHANGED JAN 30 1986 BY H. MAGNUSSON
C
IF (IV1.GT.128) IV1,128
FRAC=1 PROM (IV1)
IF (IVDISC.LT.0.) IFRAC=127-IFRAC

C STEP 1-4: COMPUTE AMBIGUOUS VELOCITY ESTIMATE — COMBINE INTEGRAL AND FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF FILTER WIDTH.

C FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF A FILTER WIDTH.
IFVEL=FLOAT(IFRAC+128-INTEG)

C CHANGED FEB 6 1986 BY M. MEYER
C
IF (MPRF.EQ.1) THEN
IF (INTEG.GE.0.AND.INTEG.LE.21) THEN
IRVEL=0.
ELSE
IRVEL=4096.
END IF
GO TO 8
END IF

C STEP 2: SCALE ROUGH VELOCITY ESTIMATE
C
C STEP 2-1: SCALE LSB OF ROUGH RANGE RATE ESTIMATE TO 4 TIMES A DOPPLER WIDTH. DEFINITION:
VTI(MPRF)=(RANGE LSB)/((MAX. UNAMBIGUOUS VELOCITY)/8) OR VTI(MPRF)=S./(PRF*LAMBDA)
RI=IRDOT*VTI(MPRF)/TSAM
IR1=AINT(R1)

C STEP 2-2: PERFORM SOME REQUIRED AUXILIARY CALCULATIONS.
IR2=AINT(R2)
IRVEL=IR2+4096.

C STEP 3: RESOLVE AMBIGUITY
C
C STEP 3-1: COMPUTE 3 MSB’S OF AMBIGUOUS VELOCITY ESTIMATE.
IF3=AINT(IFVEL/512.)
C
C STEP 3-2: COMPUTE 3 LSB’S OF SCALED ROUGH RANGE RATE ESTIMATE.
IR3=ABS(IR1-B. +IR2)
IF (R1.LE.8.) GO TO 10
IRVEL=IRVEL+4096.
IR3=7.-IR3
CONTINUE
C
C STEP 3-3: COMPARE 3 MSB’S AND 3 LSB’S AND INCREMENT NUMBER OF AMBIGUOUS FILTER BANK WIDTHS APPROPRIately.
IDELTA=IR3-IF3
IF (IDELTA.GE.4.) IRVEL=IRVEL+4096.
IF (IDELTA.LE.-4.) IRVEL=IRVEL+4096.
C
C FIGURE 2.5-2 DELIVERABLE VERSION OF SUBROUTINE VELPRO

PAGE 2

2-102
8 CONTINUE

C*************************************************************************
C** STEP 4: COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE.                        
C*************************************************************************

C STEP 4-1: ADD NUMBER OF AMBIGUOUS FILTER BANK WIDTHS TO ESTIMATE
C           OF FRACTIONAL FILTER BANK WIDTH. NOTE: LSB OF RESULTANT
C           ESTIMATE REPRESENTS 1/4096 OF A FILTER BANK WIDTH.
C
IVEL=INTT(IRVEL-IFVEL)

C STEP 4-2: SCALE LSB OF RESULTANT ESTIMATE TO 0.05 FEET/SEC.
C DEFINITION: VT2(MPRF)=((FILTER SEPARATION)/128.)/(VELOCITY LSB)
C OR VT2(MPRF)=(PRF*LAMBDA)/(0.05*8192).
C
IVEL=INTT(IVEL*VT2(MPRF)+0.5)

C*************************************************************************
C** STEP 5: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY                      
C*************************************************************************

C STEP 5-1: UPDATE REGISTERS OF MOVING WINDOW AVERAGER.
DO 28 I=1,3
20 VEST(5-I)=VEST(4-I)
VEST(1)=IVEL

C STEP 5-2: COMPUTE MOVING WINDOW AVERAGE AND SCALE ANSWER INTO
C FEET/SEC FROM UNITS OF 0.05 FEET/SEC.

M=MPRF
M1=MW(1,M)
M2=MW(2,M)
M3=MW(3,M)
M4=MW(4,M)

SRDOT=0.0125*(VEST(M1)+VEST(M2)+VEST(M3)+VEST(M4))

C*************************************************************************
C** STEP 6: RESET DOPPLER FILTER BANK                                 
C*************************************************************************

C STEP 6-1: USE ON-TARGET DISCRIMINANT AND VELOCITY DISCRIMINANT TO
C DETERMINE UPDATE OF FILTER BANK POSITION.
C THE FOLLOWING RULES ARE USED:

CASE 1: ODISC>0. AND -51.<IVDISC<51. IMPLIES NO CHANGE.
CASE 2: ODISC>0. AND IVDISC>51. IMPLIES SHIFT -1.
CASE 3: ODISC>0. AND IVDISC<-51. IMPLIES SHIFT +1.
CASE 4: ODISC<0. AND IVDISC>51. IMPLIES SHIFT -2.
CASE 5: ODISC<0. AND IVDISC<0. IMPLIES SHIFT +2.

IF(ODISC.GE.0.) GO TO 30
IF(IVDISC.LT.0.) MDF(1)=MOD(MDF(1)+2,32)
IF(IVDISC.GE.0.) MDF(1)=MOD(MDF(1)+30,32)
GO TO 40
30 IF(IVDISC.GT.51.) MDF(1)=MOD(MDF(1)+31,32)
IF(IVDISC.LT.-51.) MDF(1)=MOD(MDF(1)+1,32)
C
C STEP 6-2: RESET REMAINING FILTERS IN THE BANK-OF-5.
40 DO 50 I=1,4
50 MDF(1+I)=MOD(MDF(1)+I,32)
RETURN
END

FIGURE 2.5-2 DELIVERABLE VERSION OF SUBROUTINE VELPRO
PAGE 3

2-103
LINES DELETED FROM BASELINE PROGRAM
42 C

LINES ADDED TO DELIVERABLE PROGRAM
42 C
43 C
44 C
45 C SUBROUTINE VELPRO WAS MODIFIED FEB 6 1986 BY M. MEYER
46 C MODIFICATIONS CONSISTED OF CHECKING THE VARIABLE MPRF
47 C FOR A VALUE OF ONE (IMPLIES 7 KC MODE) AND IF TRUE
48 C ASSUMING THE VELOCITY ESTIMATE GIVEN BY THE VELOCITY
49 C DISCRIMINANT IS UNAMBIGUOUS.
50 C
51 C

LINES DELETED FROM BASELINE PROGRAM
62 IFRAC=IPROM(IV1)

LINES ADDED TO DELIVERABLE PROGRAM
71 C CHANGED JAN 30 1986 BY H. MAGNUSSON
72 C
73 C
74 IF(IV1.GT.128)IV1=128
75 IFRAC=IPROM(IV1)

LINES DELETED FROM BASELINE PROGRAM
69 C
70 C

LINES ADDED TO DELIVERABLE PROGRAM
82 C
83 C CHANGED FEB 6 1986 BY M. MEYER
84 C
85 C
86 IF(MPRF.EQ.1) THEN
87 IF(INTEG.GE.8.AND.INTEG.LE.21) THEN
88 IRVELE=8.
89 ELSE
90 IRVEL=4896.
91 END IF
92 GO TO 8
93 END IF
94 C

LINES DELETED FROM BASELINE PROGRAM
105 C

FIGURE 2.5-3 SUMMARY OF MODIFICATIONS TO SUBROUTINE VELPRO
PAGE 1

2-104
*****
LINES ADDED TO DELIVERABLE PROGRAM
129 B CONTINUE
130 C

Number of difference sections found: 4
Number of difference records found: 26

DIFFERENCES /IGNORE=()/MERGED=1/OUTPUT=SYS$DISK3:[MCCOLLOUGH]DIFF7.FOR:1-
SYS$DISK3:[MCCOLLOUGH]VELPROH.FOR:2-
SYS$DISK3:[MCCOLLOUGH]VELPROF.FOR:2

FIGURE 2.5-3 SUMMARY OF MODIFICATIONS TO SUBROUTINE VELPRO
2.5.4 Integration and Test Data

2.5.4.1 Test Definition

The philosophy for validating the ambiguity resolver modification was to test two things: (1) the boundaries where the velocity goes ambiguous in the 7 kHz mode ( +75 feet per second or -175 feet per second) and (2) to insure that the velocity becomes unambiguous when the PRF is switched to 3 kHz. Two simulation scenarios were defined to test these two features.

To check the boundaries on the ambiguous velocity in the 7 kHz PRF mode, the following scenario was used. A 10 dBsm target was given an initial range of 30,000 feet and an initial opening velocity of +100 feet per second. This velocity was held for 100 seconds and then target was decelerated at a rate of 1 foot per second for the next 300 seconds. At this point, the scenario was terminated. Plots of the range and range rate time histories are provided in Figures 2.5-4 and 2.5-5, respectively.

A similar scenario was used to insure that the velocity becomes unambiguous when the PRF is switched to 3 kHz and vice-versa. In fact, the range rate profile is identical to that given in Figure 2.5-5. The initial range in this case is 42,000 feet, so the range profile is shifted upward by +12,000 feet as shown in Figure 2.5-6. The purpose of this shifted profile is to insure that the radar transitions to the 3 kHz PRF. The design of this scenario demonstrates that ambiguous opening targets become unambiguous when transitioning from 7 kHz to 3 kHz PRF. It also demonstrates that a closing target with velocity less than -175 feet per second will become ambiguous at the transition from 3 kHz PRF to 7 kHz PRF.

2.5.4.2 Test Results

Figure 2.5-7 gives a plot of the difference between the true target velocity and target velocity predicted by the radar as a function of time. A comparison of this plot against the range rate profile of Figure 2.5-5 shows that the velocity processor model has the proper boundaries on the unambiguous zone: (+75 fps, -175 fps).
FIGURE 2.5-4 RANGE PROFILE FOR SCENARIO NO. 1
FIGURE 2.5-5 RANGE RATE PROFILE FOR SCENARIOS NO. 1 AND NO. 2
FIGURE 2.5-6 RANGE PROFILE FOR SCENARIO NO. 2

2-109
FIGURE 2.5-7 VELOCITY ERROR FOR SCENARIO NO. 1 DEFINED BY FIGURES 2.5-4 AND 2.5-5
Figure 2.5-8 gives a plot similar to Figure 2.5-7 for the second scenario. In this case, the velocity difference time history should be compared to the range profile plot of Figure 2.5-6. Taken together, these data show that velocity becomes unambiguous at the transition from 7 kHz to 3 kHz PRF at a range of 49,920 feet and the velocity becomes ambiguous at the 3 kHz to 7 kHz PRF transition at a range of 43,510 feet. Notice that this second test validates the PRF hysteresis loop as well.
FIGURE 2.5-8 VELOCITY ERROR FOR SCENARIO NO.2 DEFINED BY FIGURES 2.5-5 AND 2.5-6
(THIS PAGE INTENTIONALLY LEFT BLANK)
3.0 SORTE DATA ANALYSIS

The purpose of this section is to describe the extent of the analysis performed on the SORTE data and provide the results of that analysis. Section 3.1 provides some background data on the SORTE program, describing the test setup and the test procedures. Section 3.2 defines the approach in the analysis and a summary of the findings. Section 3.3 describes the Ku-Band Radar's range measurement performance. Section 3.4 provides analysis of the range rate measurements. Sections 3.5 and 3.6 provide an analysis of the angle and angle rate measurements. Finally, Section 3.7 gives a comparison of simulation generated data and the SORTE data. The simulation data was generated by injecting the corresponding SORTE trajectory into the simulation.

3.1 SORTE PROGRAM SUMMARY

The purpose of the Shuttle Orbiter Radar Test and Evaluation (SORTE) program was to evaluate the accuracies of the following Ku-Band Radar measurements: range, range rate, roll angle, pitch angle, ILLOS roll rate, and ILLOS pitch rate. These accuracies were to be determined by using the precision measuring system at the White Sands Missile Range (WSMR) as a reference. In the following paragraphs a brief description of the test setup, test procedures, and post-test data processing will be provided for reference throughout Section 3.

3.1.1 Flight Trajectory and Target Selection

Selection of trajectories and targets was driven by the test objectives. The principal objective was to determine the Ku-Band Radar measurement accuracies using flight conditions that simulate an actual shuttle-satellite rendezvous as closely as possible. Since actual rendezvous data existed at the time the SORTE test trajectories were defined, these trajectories were patterned after the Solar Maximum Mission Satellite (SMMS) - Shuttle rendezvous obtained from Mission 41C in April 1984. Figure 3.1-1 gives a range history of the rendezvous and Figure 3.1-2 gives a range versus
FIGURE 3.1-1 RANGE HISTORY FOR SHUTTLE-SMMS RENDEZVOUS DURING MISSION 41C IN APRIL 1984

FIGURE 3.1-2 RANGE VERSUS RANGE RATE PROFILE FOR SHUTTLE-SMMS RENDEZVOUS DURING MISSION 41C IN APRIL 1984
range rate plot of the rendezvous. As shown on these two figures, the trajectory is divided up into several smaller trajectories which are labeled as shown in the figure. The principal reason for subdividing the trajectory was a 10 minute upper limit on the length of a given test run. This limit was established to avoid data tape changes, causing loss of data, during the tests. Table 3.1-1 (Reference 1) gives the range interval and range rate interval of operation for each of these tests.

**TABLE 3.1-1 RANGE AND RANGE RATE COVERAGE BY TEST RUN**

<table>
<thead>
<tr>
<th>Test Run</th>
<th>Range (thousands of feet)</th>
<th>Range Rate (feet per second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H30SKAE</td>
<td>3.6 to 2.6</td>
<td>0.0 to 33.5</td>
</tr>
<tr>
<td>H30SKAF</td>
<td>2.2 to 4.1</td>
<td>-22.0 to -3.0</td>
</tr>
<tr>
<td>H30SKAG</td>
<td>2.2 to 2.6</td>
<td>-12.0 to -3.0</td>
</tr>
<tr>
<td>H30SKAH</td>
<td>2.2 to 4.0</td>
<td>-31.0 to -4.0</td>
</tr>
<tr>
<td>H30SKAI</td>
<td>3.3 to 3.6</td>
<td>-18.4 to 4.8</td>
</tr>
<tr>
<td>HEL30AF</td>
<td>7.0 to 12.2</td>
<td>-60.0 to 7.0</td>
</tr>
<tr>
<td>HEL30AG</td>
<td>7.0 to 13.0</td>
<td>-56.0 to -8.0</td>
</tr>
<tr>
<td>HEL30AI</td>
<td>5.6 to 13.3</td>
<td>-38.0 to -10.0</td>
</tr>
<tr>
<td>HEL30AJ</td>
<td>6.0 to 13.6</td>
<td>-45.0 to 0.0</td>
</tr>
<tr>
<td>HJ146AC</td>
<td>45.0 to 63.0</td>
<td>-55.0 to -10.0</td>
</tr>
<tr>
<td>HJ146AD</td>
<td>47.0 to 64.0</td>
<td>-50.0 to 4.0</td>
</tr>
<tr>
<td>HJ146AE</td>
<td>46.0 to 65.0</td>
<td>-21.0 to 10.0</td>
</tr>
<tr>
<td>HL146AE</td>
<td>42.6 to 46.5</td>
<td>-21.0 to 10.0</td>
</tr>
<tr>
<td>HL246AD</td>
<td>43.4 to 47.3</td>
<td>4.0 to 17.5</td>
</tr>
<tr>
<td>HL246AE</td>
<td>41.5 to 47.2</td>
<td>-5.0 to 20.0</td>
</tr>
<tr>
<td>HL346AD</td>
<td>48.0 to 48.9</td>
<td>-6.0 to 14.0</td>
</tr>
<tr>
<td>HL346AE</td>
<td>47.0 to 49.0</td>
<td>-23.0 to 26.0</td>
</tr>
<tr>
<td>HL346AF</td>
<td>47.1 to 49.1</td>
<td>-5.0 to 13.0</td>
</tr>
<tr>
<td>HL446AC</td>
<td>47.8 to 48.8</td>
<td>-13.0 to 10.0</td>
</tr>
<tr>
<td>HL446AD</td>
<td>47.0 to 49.7</td>
<td>-20.0 to 37.0</td>
</tr>
<tr>
<td>HL446AE</td>
<td>46.8 to 49.3</td>
<td>-16.0 to 16.0</td>
</tr>
<tr>
<td>HL546AC</td>
<td>41.3 to 47.0</td>
<td>-22.0 to 0.0</td>
</tr>
<tr>
<td>HL546AE</td>
<td>41.2 to 47.7</td>
<td>-21.0 to 55.0</td>
</tr>
<tr>
<td>HL546AF</td>
<td>40.8 to 46.5</td>
<td>-21.0 to 7.5</td>
</tr>
<tr>
<td>HL546AG</td>
<td>40.7 to 46.7</td>
<td>-25.0 to 25.0</td>
</tr>
</tbody>
</table>

The target selected for use in these flight tests was a UH-1H helicopter. To enhance the Radar Cross Section (RCS) of the helicopter, a pair of Luneberg lenses were mounted on the underside of the helicopter as shown in Figure 3.1-3. The "main beams" of these lenses were angled off from the helicopter nose and were pointed downward slightly. As will be shown in
the analysis of the range rate performance in Section 3.3, these target enhancements were effective for those trajectories where the helicopter flew approximately down the Line-of-Sight (LOS) of the radar. However, this enhancement configuration was not effective when the helicopter flew a trajectory that was perpendicular, or Cross Line-of-Sight (XLOS) to the radar.

A second series of tests was based on the second major objective of the SORTE program: determining the effects of a conducting and non-conducting tether in the radar antenna beam. The purpose of these tests was to evaluate the usefulness of the Ku-Band Radar for tracking the Tethered Satellite System (TSS) on a future shuttle mission. The target for these tests consisted of a mockup of the TSS suspended from two, 10-foot inch diameter, Helium-filled balloons. This target was then tethered with a conducting or non-conducting tether. (As an aside, a red colored, Helium-filled balloon was tied to the tether at a point 50 feet below the main target to provide a secondary target for the cinetheodolites.) This balloon/target combination was then flown as closely as possible directly overhead relative to the radar. With the tether spool anchored within 20 feet of the radar, a significant portion of the tether would be in the beam when the target was directly overhead. Again the test duration was 10 minutes and altitudes from 300 to 3000 feet were planned to simulate reeling in and reeling out the TSS.

A third series of tests were performed. These tests consisted of filling a two meter in diameter Gemsphere\(^1\) with Helium, releasing it near the site of the radar, and tracking it for 10 minutes.

Table 3.1-2 summarizes the range and range rate intervals for the "tether tests" and the Gemsphere release tests. The tether tests are denoted by "SAT" and the Gemsphere tests are denoted by "BAL" or "GEM".

A Gemsphere is a metallic coated balloon with small protrusions (2-3 inches) distributed uniformly over the surface. These spheres are used by the National Weather Service to track upper atmosphere wind currents.
<table>
<thead>
<tr>
<th>Test Run</th>
<th>Range (thousands of feet)</th>
<th>Range Rate (feet per second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT1</td>
<td>2.5 to 2.6</td>
<td>- 5.0 to 5.0</td>
</tr>
<tr>
<td>SAT2</td>
<td>2.5 to 2.5</td>
<td>- 4.0 to 4.0</td>
</tr>
<tr>
<td>SAT3</td>
<td>1.2 to 2.5</td>
<td>- 6.0 to 4.0</td>
</tr>
<tr>
<td>SAT4 *</td>
<td>10.9 to 12.7</td>
<td>- 4.0 to 10.0</td>
</tr>
<tr>
<td>BAL1</td>
<td>0.8 to 10.4</td>
<td>2.5 to 29.0</td>
</tr>
<tr>
<td>BAL2</td>
<td>0.8 to 5.5</td>
<td>0.0 to 29.0</td>
</tr>
<tr>
<td>BAL5</td>
<td>8.0 to 10.3</td>
<td>8.0 to 17.0</td>
</tr>
<tr>
<td>BAL6</td>
<td>1.0 to 10.6</td>
<td>2.5 to 7.5</td>
</tr>
<tr>
<td>BAL7</td>
<td>1.0 to 10.6</td>
<td>7.5 to 31.0</td>
</tr>
<tr>
<td>GEM2</td>
<td>3.2 to 30.0</td>
<td>32.0 to 70.0</td>
</tr>
<tr>
<td>GEM3</td>
<td>2.1 to 26.0</td>
<td>30.0 to 68.0</td>
</tr>
</tbody>
</table>

* The tether broke between tests SAT3 and SAT4 so that the target was held only by a guidewire 12 kft in length.

Please note that the above summary is not meant to be an exhaustive summary of the SORTE tests, but a summary of the three principal series of tests for which data analysis has been performed and included in this report.

3.1.2 Test Setup

The radar was situated very near the brass cap at the PEARL site at WSMR. Figure 3.1-4 (taken from Reference 1) shows the PEARL site relative to the layout of the entire White Sands Range. The deployed assembly of the radar, including the transmitter, receiver, gimbal and antenna, were placed on a platform inside a radome near the brass cap. The radar was a few feet
FIGURE 3.1-4 ILLUSTRATION OF THE PEARL SITE IN RELATION TO THE WHITE SANDS MISSILE RANGE LAYOUT
south and east of the brass cap and about 20 feet higher. Its exact White Sands Coordinate System (WSCS) location (from Reference 1) was:

- East: 485,227.79 feet
- North: 265,161.98 feet
- Up: 2,618.43 feet.

The deployed assembly was oriented so that 0 degrees alpha and 0 degrees beta corresponded to the antenna boresight pointing 30 degrees east of north and elevated 30 degrees. This orientation was chosen to reduce the stress on the gimbals in a 1-g environment and to avoid ground clutter during radar operation.

Two types of sensor systems were used by WSMR to provide a target tracking reference. One system of sensors consisted of a set of cinetheodolites, designated as cines in the rest of this report. This set usually consisted of five cines for a given flight test. These five cines were chosen from a large number of cines which are widely distributed over the southern end of WSMR. Choice of the five cines for a given test was based upon the geometry of the flight profile for that particular test.

The second system of sensors consisted of a set of three AN/MPS-36 radars, denoted as R350, R393, and R394 by WSWR. Data from these radars is combined and processed to produce an estimate of target range and range rate. The combination of these radars and the post flight signal processing is called the Target Motion Resolution (TMR) system at White Sands. Details of TMR data processing are described in Reference 8.

Figure 3.1-5 (from Reference 1) gives a view of the Ku-Band Radar position, the cine positions (for a given set of trajectories), and the TMR radar positions in WSCS. It also provides the ground track for the HJ146, HL246, and HEL30 trajectories.
FIGURE 3.1-5 POSITIONS OF THE KUBAND RADAR, THE CINES, AND THE WSMR FOR SOME EXAMPLE TARGET TRAJECTORIES
3.1.3 Data Acquisition and Processing

The common element among the three data acquisition systems, the Ku-Band Radar, the cines and the TMR, was the time stamping of the data gathered by each system. WSMR provided universal timing which was networked to each radar and cine site and to the Ku-Band Radar so that the data could be time coded as it was gathered.

3.1.3.1 Ku-Band Radar Data Processing

Ku-Band Radar data acquisition for the SORTE program is best summarized via the illustration of Figure 3.1-6. Two types of data were gathered on the system test equipment (STE) computer (LSI 4/90): data from the MDM output and analog data which was digitized and recorded on disk. Each set of data included a range time stamp.

Once the Ku-Band Radar data for a particular test was recorded at the PEARL site, the disk was taken to Building 1646 at WSMR to be processed by a second LSI/490 computer. The purpose of this processing was to transfer the data in a VAX 11/780 compatible format to tape. Two tapes were made: one was for storage at the WSMR data processing facility and the other was to be used at Johnson Space Center (JSC) for further data analysis on the Building 44 VAX 11/780.

3.1.3.2 WSMR Sensor Data Processing

Data acquired by the individual WSMR radars and cines is summarized in Table 3.1-3. The data gathered by the various radar and cine sites is passed in real-time over the Precision Acquisition System (PAS) network to the central data processing facility at WSMR. This data is then post-processed to produce three sets of data. Each data set consists of the target position \( (X, Y, Z) \), the target velocity \( (\dot{X}, \dot{Y}, \dot{Z}) \), and the time code for the entire flight test. Target position and velocity values are given in the PEARL site brass cap coordinate system which is a North-East-Down (NED) system whose origin resides at the brass cap. The three post processing methods are described below.
FIGURE 3.1-6 ILLUSTRATION OF THE KUBAND RADAR DATA ACQUISITION PROCESs FOR THE SORTE PROGRAM AT WSMR
The first data set is obtained by processing only cinetheodolite data to produce target position and velocity as a function of range time. This data set is called cine data in the sequel. Data from the three WSMR Radars is processed using the TMR algorithms to produce target position and velocity as a function of range time. This second data set is denoted as the TMR data throughout the remainder of the report. The third data set combines the best features of the cine processing and the TMR processing. The cines produce highly accurate position data, while the TMR produces very accurate velocity data. Hence, the new system, called the "BEST" system, uses the cine data for the initial position estimate and propagates the position using velocity data from the TMR.

All three data sets were generated for those flight tests where both systems of sensors were operable. Table 3.1-4 taken from Reference 1 summarizes the available sensors for each of the flight tests.

### Final Data Processing

At this point in the data processing scheme, the Ku-Band Radar data resides on a VAX 11/780 compatible tape. These data are in standard shuttle orbiter body coordinates. The post-processed WSMR data has also been loaded onto tape in a VAX 11/780 format and delivered to JSC. These WSMR Sensor data have been converted to the PEARL site brass cap coordinate system described in the previous subsection.
### TABLE 3.1-4 AVAILABLE WSMR SENSORS FOR EACH TEST RUN

<table>
<thead>
<tr>
<th>Test Run</th>
<th>Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAL1, Nov 4, T0-60302</td>
<td>Radar 394, no optics.</td>
</tr>
<tr>
<td>BAL2, Nov 4, T0-61350</td>
<td>Radar 394, no optics.</td>
</tr>
<tr>
<td>BAL5, Nov 4, T0-62785</td>
<td>Radar 394, no optics.</td>
</tr>
<tr>
<td>BAL6, Nov 4, T0-63348</td>
<td>Radar 394, no optics.</td>
</tr>
<tr>
<td>BAL7, Nov 4, T0-63346</td>
<td>Radar 394, no optics.</td>
</tr>
<tr>
<td>GEM2, Oct 16, T0-76421</td>
<td>TMR, no optics.</td>
</tr>
<tr>
<td>GEM3, Oct 16, T0-77603</td>
<td>TMR, no optics.</td>
</tr>
<tr>
<td>H30SKAE, Oct 3, T0-56647</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>H30SKAF, Oct 3, T0-56987</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>H30SKAG, Oct 3, T0-60657</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>H30SKAH, Oct 3, T0-60821</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>H30SKAI, Oct 3, T0-61113</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HEL30AF, Oct 3, T0-56123</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HEL30AG, Oct 3, T0-57558</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HEL30AI, Oct 3, T0-61665</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HEL30AJ, Oct 3, T0-62488</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HJ146AC, Oct 1, T0-67031</td>
<td>TMR, no optics.</td>
</tr>
<tr>
<td>HJ146AD, Oct 5, T0-62415</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HJ146AE, Nov 4, T0-80843</td>
<td>Radar 394 and optics.</td>
</tr>
<tr>
<td>HJ146AE, Nov 4, T0-76124</td>
<td>Radar 394 and optics.</td>
</tr>
<tr>
<td>HL246AD, Oct 1, T0-60295</td>
<td>Radar, reduced optics.</td>
</tr>
<tr>
<td>HL246AE, Oct 5, T0-55880</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HL346AD, Oct 1, T0-65780</td>
<td>TMR, no optics.</td>
</tr>
<tr>
<td>HL346AE, Oct 5, T0-61367</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HL346AF, Nov 4, T0-79738</td>
<td>Radar 394 and optics.</td>
</tr>
<tr>
<td>HL446AC, Oct 1, T0-61463</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HL446AD, Oct 5, T0-57012</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HL446AE, Nov 4, T0-75072</td>
<td>Radar 394 and optics.</td>
</tr>
<tr>
<td>HL546AC, Oct 1, T0-59240</td>
<td>TMR, no optics.</td>
</tr>
<tr>
<td>HL546AE, Oct 5, T0-54805</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HL546AF, Oct 5, T0-63406</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>HL546AG, Nov 4, T0-72220</td>
<td>Radar 394 and optics.</td>
</tr>
<tr>
<td>SAT1, Oct 19, T0-50988</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>SAT2, Oct 19, T0-52227</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>SAT3, Oct 19, T0-53295</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>SAT4, Oct 19, T0-55207</td>
<td>TMR and optics.</td>
</tr>
<tr>
<td>SAT6, Oct 19, (Acquisition)</td>
<td>TMR, no optics.</td>
</tr>
<tr>
<td>SAT8, Oct 19, (Acquisition)</td>
<td>TMR, no optics.</td>
</tr>
</tbody>
</table>

The final component in the processing was performed on the computers at JSC by NASA and LEMSCO personnel and consisted of several steps. The first step involved transforming the WSMR sensor data from brass cap coordinates to the shuttle body coordinates. The mathematics of this transformation were developed by Bill Culpepper of LEMSCO and are documented in Reference 9. The next step was to compute difference profiles for each of
the radar parameters of interest. This means that for a given radar parameter, the Ku-Band Radar data profile was subtracted from the corresponding WSMR sensor data profile to produce the difference data profile. The final step was a statistical analysis of the resulting difference profile to produce a mean and a standard deviation for the interval and a diagram of this processing is shown in Figure 3.1-7. A sample result of this processing procedure is shown in Figure 3.1-8.

The procedure for analyzing this processed data is outlined in Section 3.2.

3.1.4 Summary of Flight Tests

There were 44 flight tests where data was gathered by both the WSMR sensors and the Ku-Band Radar. Careful notes were compiled by A. C. Lindberg of LEMSCO concerning the weather conditions and any anomalies that occurred during each of the tests. These notes, along with observations about the difference data profiles, are given in a summary form in Appendix C. Results of an extensive analysis of this data follows below.

3.2 ANALYSIS APPROACH AND PRELIMINARY FINDINGS

As anticipated (Reference 10) the SORTE data analysis activity was very limited due to available contract resources. Since this was expected, an analysis procedure was formulated to optimize the data reduction effort. The method developed was a two step procedure. The first step consisted of one complete pass through the data to identify any major problem areas. In the second step an extensive analysis of these problem areas was undertaken. The intent of this second step was to identify the dominant error source (or sources) and develop a quantitative estimate of its effect. The next level of priority in the data analysis was to resolve any significant anomalies found in the data.
FIGURE 3.1-7 SIMPLIFIED DIAGRAM OF FINAL PROCESSING OF WSMR SENSOR AND KU BAND RADAR DATA
FIGURE 3.1-8  EXAMPLE OF A DIFFERENCE DATA PLOT
3.2.1 Preliminary Findings

In the first step of the procedure, the means and standard deviations of the difference data was compared against the corresponding radar specification (listed in Table 1-2) to determine which parameters were within specification for each test run. This test surfaced major problems in the following parameters (also see Table 1-3 of Section 1):

(1) Range rate standard deviation (95\% failure)

(2) Roll rate mean and standard deviation (93\% failure)

(3) Pitch rate mean and standard deviation (100\% failure).

Problems of a smaller magnitude were also found in the angle data:

(4) Roll angle standard deviations (43\% failure)

(5) Pitch angle standard deviations (19\% failure).

Extensive analyses of the areas identified above were then undertaken. Results of these analyses are summarized in the following subsections. However, there are some general observations from these analyses that can be stated here. Almost all of the problems in the data can be attributed to the following categories:

(1) Large errors in the sensor data due to the sensor configuration and target position. This problem is commonly called Geometric Dilution of Precision or GDOP.

(2) Target acceleration in both range and angle.

(3) Low signal-to-noise power ratio (SNR) at the doppler filter output. This is principally due to a small radar cross section (RCS).
In addition, there were some general observations concerning the dominant error sources. These are that

(4) Different flight trajectories had different dominant error sources.

(5) The dominant error source could change within a given flight trajectory.

These observations on dominant error sources were found to be prevalent in the range rate analysis. Angle acceleration and transformation errors were found to be the major contributors to errors in the roll and pitch angle data. Angle acceleration and a scale factor error were the significant contributors to the problems in the ILOS roll and pitch angle rate data. All of these problems are discussed in detail in the following subsections.

3.3 RANGE DATA ANALYSIS

The first cut at analyzing the range error data was quite encouraging. The standard deviation of the range difference data was beyond the specification limit on four flights, and the mean was outside the specification limit on three flights. These cases are summarized in Table 3.3-1. In addition to these few problems with the range error data statistics, there were some anomalies in the range data. All of these anomalies took the form of discontinuous jumps in the Ku-Band Radar range estimate.

**TABLE 3.3-1 SUMMARY OF FIRST CUT AT RANGE ERROR DATA ANALYSIS**

<table>
<thead>
<tr>
<th>PROFILE</th>
<th>SENSOR</th>
<th>VALUE, FT</th>
<th>PROFILE</th>
<th>SENSOR</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEM3</td>
<td>TMR</td>
<td>27.3</td>
<td>GEM2</td>
<td>TMR</td>
<td>35.3</td>
</tr>
<tr>
<td>H3OSKAH</td>
<td>BEST</td>
<td>-41.5</td>
<td>GEM3</td>
<td>TMR</td>
<td>43.1</td>
</tr>
<tr>
<td>SAT3</td>
<td>BEST</td>
<td>30.2</td>
<td>SAT2</td>
<td>BEST</td>
<td>30.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SAT3</td>
<td>BEST</td>
<td>51.2</td>
</tr>
</tbody>
</table>
The purpose of this subsection is to describe the analysis of the range difference data statistics problems and provide some observations about the discontinuous jumps in the Ku-Band Radar range profile.

3.3.1 Discussion of Range Difference Data Statistics Problems

3.3.1.1 Description of Potential Error Sources

The potential sources of errors in the range difference data are the following:

- GDOP
- Low SNR (weak target return signal)
- Target Range Acceleration

We first demonstrate that target range acceleration is not a consideration in the present analysis because the value of the acceleration would have to be quite large to produce a range bias that would cause the radar range estimate to fail its specification. Consider an example: a \(-10\) feet/sec\(^2\) acceleration would generate a range bias of 5.87 feet in the narrowest bandwidth case of the range tracker. (This example was taken from Reference 3. The closed-formed expression for the asymptotic range bias in the presence of acceleration is provided there.) Furthermore, the bias is smaller for wider bandwidths of the tracker. Thus, in the following discussion, target range acceleration will not be considered as a source of error. The discussion will be limited to GDOP and weak target return signals.

Geometric Dilution of Precision or GDOP is the name given to the error induced in a multiple sensor measuring system due to the placement of sensors and the random fluctuations of the individual sensor measurements. Appendix D gives a quantitative, rigorous derivation of the GDOP-induced error in the TMR measuring system. (We didn't have the resources to do a similar computation for the CINE system.) The results of the calculations provide the following qualitative observation. GDOP-induced range error is the worst at very low altitude and directly over the PEARL site brass cap (and the Ku-Band

3-19
Radar). For in this case, range to the target from the brass cap origin is along the -Z axis. But, since all three TMR radars are roughly in the X-Y plane and they only measure R and ̅r, then they cannot determine the target Z-component very well. Any small error in the R measurement translates to a large error in the Z-component of target position.

Although we did not have time to work out the expressions for the GDOP-induced range error in the CINE system, we can comment on the CINE performance in the situation described above using some newly gained insight. In the case of CINE system, each sensor measures the target's azimuth and elevation. In the scenario at hand, azimuth and elevation will provide information about the target's Z-component of position. Hence, small error in azimuth and elevation should not translate to large errors in the Z-component.

Weak target return signals which, in turn, produce low SNR at the doppler filter output (<10dB) will generate large random fluctuations in the range data. However, this is only a problem for weak targets (<0 dBsm) at long range (>50,000 feet). A review of the range difference data and the corresponding range and RCS profile for all test runs, indicated that low SNR did not cause any of the failures listed in Table 3.3-1. Furthermore, it did not produce unusual problems in any of the other flight data examined. Figure 3.3-1 illustrates the high correlation between the target return signal strength (proportional to RCS) and the random fluctuations in the range difference data. The data shown in the figure is for flight HL146AE with an initial range of 46,500 feet and a final range 42,800 feet.

3.3.1.2 Discussion of Individual Problem Cases

Observe that all of the problem cases listed in Table 3.3-1 were out-of-specification (1) when compared to the BEST or TMR data only and (2) for flight trajectories where low attitudes and short ranges were involved. From the discussion of Section 3.3.1.1, these facts point to GDOP as the primary source or range error. There was one perplexing problem with assuming GDOP for all of these problem cases: why didn't all of the flight tests from
FIGURE 3.3-1 ILLUSTRATION OF CORRELATION BETWEEN TARGET RETURN SIGNAL STRENGTH AND RANGE TRACKER RANDOM ERROR
a given family, e.g., all H30SK's, suffer from the same problem? It turns out each general family has its own unique answer to this question. The answers for each family, GEM, BAL, SAT and H30SK, are provided below.

**GEM and BAL Series.** In this series, a helium filled GEMsphere was released from the brass cap and allowed to fly freely. Since all of these flights start at very low altitude over the brass cap, one would expect GDOP problems early in the flight for both GEM and BAL. However, a review of the flight log given in Appendix G shows that the only one radar (R-394) was available for the BAL series. Hence, there is no TMR or BEST solution available and consequently there is no problem with GDOP for the BAL tests.

Observe that the GEM3 failed both the mean and standard deviation specification while GEM2 failed only the standard deviation specification. Let's first examine the initial tracking altitude and range for both cases. For GEM2, the initial altitude and range are 2000' and 3000', respectively, and for GEM3 they are 1500' and 2000', respectively. At these altitudes, a delta of 500 feet makes a significant difference in the GDOP error. This difference can be seen in the BEST range difference profiles for GEM2 and GEM3 shown in Figure 3.3-2.

It has been observed in other test series (H30SK) that GDOP-induced range error is sensitive to the X-Y ground track, especially at low altitude. This problem is not as significant in this case. The predominant difference is the delta in initial altitudes.

There are some additional observations. First, to determine whether the range difference data mean and standard deviation were out-of-specification, they were both compared to 26.67 feet. This value is the limit for target ranges less than 8000 feet, while 1/3% of the range is used for ranges greater than 8000 feet. But, the target range interval was 3000 feet to 11,000 feet for GEM2 and 2000 feet to 26,000 feet for GEM3. Hence, a more correct determination of an out-of-specification condition would break the range difference data profile into intervals for ranges less than 8000 feet and greater than 8000 feet, compute means and standard deviations for each interval, and apply the correct specification to each interval.
FIGURE 3.3-2 BEST RANGE DIFFERENCE DATA PROFILES FOR GEM2 AND GEM3
Secondly, notice that the random component in the range difference data of Figures 3.3-2 is increasing with time. This correlates with the fact the target is moving away from the radar and further illustrates the effect of decreasing target return signal strength.

Thirdly, the jumps in range bias seen at the pulsewidth switch points adds significantly to the mean and standard deviation values.

**SAT Series.** The reason the SAT4 data was not a problem was because the altitude interval for the flight was 5100 feet to 68000 feet, and the range interval was 10,800 feet to 12,600 feet. As discussed previously, GDOP is not a problem at this altitude and range. Also, target return signal strength was not a factor at these ranges, even though the target RCS dropped to -10 dBsm at some points. Finally, since the balloon was tethered, range acceleration was not a consideration.

SAT2 and SAT3 were both susceptible to GDOP because their range of operation was less than 2600 feet. In fact, SAT3 started at 2600 feet range and finished at 1200 feet, while SAT2 remained fixed at approximately 2550 feet. The difference in range of these two cases would lead one to conclude that SAT3 would experience more severe GDOP effects than SAT2. That this conclusion is true is supported by the SAT2 and SAT3 BEST range difference profiles of Figure 3.3-3 and the problem summary of Table 3.3-1.

Discontinuous jumps of 60 feet were found in the SAT3 BEST range difference data at times 205 seconds and 280 seconds (see Figure 3.3-3). These jumps are not a problem with the Ku-Band Radar, but instead, are caused by the BEST range data as shown in Figure 3.3-4.

The SAT1 flight profile is very similar to the SAT2 but SAT1 range difference data statistics were within specification. A close examination of this data shows that GDOP has induced significant error in the SAT1 range difference data as shown in Figure 3.3-5. But why is the error less significant in this case? Analysis of the X-Y ground track and the altitude data for both cases shows that, while the SAT1 flight is at a slightly lower altitude, the SAT2 flight is more nearly over the radar where
FIGURE 3.3-3 BEST RANGE DIFFERENCE DATA PROFILES FOR SAT2 and SAT3
FIGURE 3.3-4 ILLUSTRATION OF JUMPS IN BEST RANGE DATA

FIGURE 3.3-5 SAT1 RANGE DIFFERENCE DATA PROFILE
the GDOP problem is most severe. Figure 3.3-6 compares the X-Y ground track of the SAT2 and SAT1 flights. Unfortunately, at the writing of the report, no qualitative GDOP computations were available to confirm these conjectures.

**H30SK Series.** In this series of tests, a helicopter flew toward the radar with a starting range of 4000 feet and a finishing range of 2000 feet. The altitude was maintained between 1500 feet to 1700 feet. H30SKAH was the only test of this series that indicated a problem with the range difference data statistics. It is reasonable to assume that the source of the error is GDOP. But, since all of the H30SK profiles are quite similar, why isn't there a problem with all of these runs? A review of the range difference profiles shows that there is significant GDOP error in all of the test runs. Figure 3.3-7 compares the BEST range difference data profiles of H30SKAE and H30SKAH. Both profiles vary significantly over the test duration with a trend toward negative range error. One major difference is that H30SKAH starts with a -20 foot offset, while H30SKAE starts with a +20 foot offset. The reason for this difference is not clear at the writing of this report.

While searching for a source of the difference in offsets described above, an interesting fact was uncovered. Figure 3.3-8 compares the BEST range difference profile and the Y-brass cap coordinate profile for H30SKAE. This comparison reveals a high correlation between these two parameters. It supports the contention that GDOP-induced range error is very sensitive to target position especially when the target is at low altitude and nearly overhead of the PEARL site brass cap. However, at this time, we have no closed-formed computation of GDOP-induced range error to support these conclusions.

### 3.3.2 Discussion of Discontinuous Jumps in Range

A review of the range difference data has surfaced some discontinuous jumps in range. These jumps are quite evident in the GEM and BAL series of data (see Figure 3.3-2). Examination of the corresponding range profile for these cases shows that the jumps occur at the Ku-Band Radar
FIGURE 3.3-6 COMPARISON OF SAT1 and SAT2 X-Y GROUND TRACK
FIGURE 3.3-7 COMPARISON OF H3OSKAH AND H3OSKAE BEST RANGE DIFFERENCE DATA PROFILES
FIGURE 3.3-8 COMPARISON OF H3OSKAE'S BEST RANGE DIFFERENCE PROFILE AND BEST-Y PROFILE

3-30
pulsewidth switch points. Some questions that come to mind immediately are as follows: Does a change in bias occur at each pulsewidth transition? Is the bias the same for a given pulsewidth transition or is it a random value? A comprehensive review of the range difference data was undertaken to answer these questions. The results of that data review are summarized in Table 3.3-2.

Some of the highlights of the review are as follows. First, and most important, there is some jump in bias at every pulsewidth transition. It is hard to discern a jump in the pulsewidth transitions at 23,030 feet and 49,920 feet because of random noise fluctuations due to a weak target return signal. Secondly, for a given pulsewidth transition, the value of the range jump was approximately the same. To confirm this statement, compare the 3200 foot and 5760 foot transition jumps for the GEM and BAL series. Thirdly, it was observed that the sign of the jump depended upon the direction of transition. This can be seen by comparing the 11520 transition point for the HEL30 series and the GEM series. A positive jump occurs in the HEL30 data where the target is closing and a negative jump is found in GEM data where the target is opening.

3.3.2.1 Discussion of Jump Mechanism

It is conjectured that these range jumps are caused by slight changes in timing for generation of the different pulsewidth values. Observe that the largest jumps found were 30 feet. This corresponds to a timing change of 60 nanoseconds using the 2 nanosecond/foot conversion for two way range. Considering the complexity of the pulse generation and range gate timing circuitry, it is not surprising to find timing bias on the order of 40-60 nanoseconds.

To confirm these conjectures requires a detailed evaluation of the pulsewidth generation and range gate timing circuitry, a study of this magnitude is far beyond the bounds of the present project resources. Anyone wishing to pursue this subject further should contact A.E. Miller, Jr., the
<table>
<thead>
<tr>
<th>PROFILE</th>
<th>RANGE COVERAGE, FT</th>
<th>NUMBER OF SWITCH POINTS</th>
<th>JUMP RANGE, FT</th>
<th>JUMP MAGNITUDE, FT</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAL1</td>
<td>800 to 10,500</td>
<td>2</td>
<td>3200</td>
<td>+5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5750</td>
<td>+20</td>
<td></td>
</tr>
<tr>
<td>BAL2</td>
<td>800 to 5,500</td>
<td>1</td>
<td>3200</td>
<td>+5</td>
<td></td>
</tr>
<tr>
<td>BAL5</td>
<td>8,000 to 10,300</td>
<td>0</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>BAL6</td>
<td>900 to 10,500</td>
<td>2</td>
<td>3200</td>
<td>+5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5750</td>
<td>+20</td>
<td></td>
</tr>
<tr>
<td>BAL7</td>
<td>1,000 to 10,700</td>
<td>2</td>
<td>3200</td>
<td>+5</td>
<td>'Needs Closer Examination'</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5750</td>
<td>+10(?)</td>
<td></td>
</tr>
<tr>
<td>GEM2</td>
<td>3,000 to 30,000</td>
<td>4</td>
<td>3200</td>
<td>-25(?)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5750</td>
<td>+20</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11520</td>
<td>+30</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>23030</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'Too Much Noise'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GEM3</td>
<td>2,200 to 26,000</td>
<td>4</td>
<td>3200</td>
<td>+5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5750</td>
<td>+15</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11520</td>
<td>+30</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>23030</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'Too Much Noise'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROFILE</td>
<td>RANGE COVERAGE, FT</td>
<td>NUMBER OF SWITCH POINTS</td>
<td>JUMP RANGE, FT</td>
<td>JUMP MAGNITUDE, FT</td>
<td>COMMENTS</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>----------</td>
</tr>
<tr>
<td>SAT1</td>
<td>2500 to 2570</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SAT2</td>
<td>2520 to 2540</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SAT3</td>
<td>2450 to 1200</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SAT4</td>
<td>10,900 to 12,700</td>
<td>1</td>
<td>11,520</td>
<td>-25</td>
<td>Sign Error?</td>
</tr>
<tr>
<td>H30SKAE</td>
<td>3650 to 2525</td>
<td>1</td>
<td>2,560</td>
<td>?</td>
<td>Requires Closer Evaluation</td>
</tr>
<tr>
<td>H30SKAF</td>
<td>4100 to 2200</td>
<td>1</td>
<td>2,560</td>
<td>?</td>
<td>Requires Closer Evaluation</td>
</tr>
<tr>
<td>H30SKAG</td>
<td>2625 to 2175</td>
<td>1</td>
<td>2,560</td>
<td>-6</td>
<td></td>
</tr>
<tr>
<td>H30SKAH</td>
<td>4000 to 2200</td>
<td>1</td>
<td>2,560</td>
<td>-6</td>
<td></td>
</tr>
<tr>
<td>H30SKAI</td>
<td>3675 to 3325</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>PROFILE</td>
<td>RANGE COVERAGE, FT</td>
<td>NUMBER OF</td>
<td>JUMP</td>
<td>JUMP</td>
<td>COMMENTS</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------</td>
<td>-----------</td>
<td>------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>HEL30AF</td>
<td>12,200 to 7,000</td>
<td>1</td>
<td>11,510</td>
<td>+30</td>
<td></td>
</tr>
<tr>
<td>HEL30AG</td>
<td>13,000 to 7,000</td>
<td>1</td>
<td>11,510</td>
<td>+30</td>
<td></td>
</tr>
<tr>
<td>HEL30AI</td>
<td>13,300 to 5,500</td>
<td>2</td>
<td>11,510</td>
<td>+40</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5,750</td>
<td>-25</td>
<td></td>
</tr>
<tr>
<td>HEL30AJ</td>
<td>13,700 to 6,000</td>
<td>1</td>
<td>11,510</td>
<td>+36</td>
<td></td>
</tr>
<tr>
<td>HJ146AC</td>
<td>62,500 to 44,800</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>HJ146AD</td>
<td>64,000 to 46,500</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>HL146AE</td>
<td>46,500 to 42,700</td>
<td>1</td>
<td>43,510</td>
<td>?</td>
<td>'Too Much Noise'</td>
</tr>
<tr>
<td>HL246AD</td>
<td>42,000 to 30,000</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>HL246AE</td>
<td>41,500 to 47,200</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>HL346AD</td>
<td>48,000 to 49,000</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>HL346AE</td>
<td>47,100 to 49,000</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>HL346AF</td>
<td>47,100 to 49,200</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>PROFILE</td>
<td>RANGE COVERAGE, FT</td>
<td>NUMBER OF SWITCH POINTS</td>
<td>JUMP RANGE, FT</td>
<td>JUMP MAGNITUDE, FT</td>
<td>COMMENTS</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------</td>
<td>-------------------------</td>
<td>----------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>HL446AC</td>
<td>48,700 to 47,500</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>HL446AD</td>
<td>47,750 to 49,750</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>to 47,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL446AE</td>
<td>48,800 to 49,300</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>to 46,800</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL546AC</td>
<td>47,000 to 41,500</td>
<td>1</td>
<td>43,510</td>
<td>+40</td>
<td></td>
</tr>
<tr>
<td>HL546AE</td>
<td>47,500 to 41,000</td>
<td>1</td>
<td>43,510</td>
<td>?</td>
<td>'Too Much Noise'</td>
</tr>
<tr>
<td>HL546AF</td>
<td>46,500 to 40,900</td>
<td>1</td>
<td>43,510</td>
<td>?</td>
<td>'Too Much Noise'</td>
</tr>
<tr>
<td>HL446AG</td>
<td>46,000 to 46,800</td>
<td>1</td>
<td>43,510</td>
<td>?</td>
<td>'Too Much Noise'</td>
</tr>
<tr>
<td></td>
<td>to 41,700</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Responsible Engineer (RE) for the signal processing unit, or R.S. Austin, the System Engineer who is familiar with this area. Both gentlemen are with HAC's Radar Systems Groups.

3.4 RANGE RATE DATA ANALYSIS

A first pass through the SORTE data revealed a high percentage (95%) of failures in the standard deviation or random component of the range rate data. This was very surprising because all previous data, including system test data and flight rendezvous data, had indicated that the range rate tracking performance was better than predicted and well within the specification. An intensive examination of the data revealed several diverse sources of errors. These error sources included

- Range acceleration,
- Geometric Dilution of Precision (GDOP),
- Small RCS (low SNR),
- Target rotation,
- Time skewing,

and combinations of the above error sources. Errors that affected the majority of the data were GDOP and range acceleration. Target effects, including small RCS and target rotation, caused significant problems in only a handful of cases.

Problems, such as GDOP and time skewing, are associated with the WSMR sensor system and data processing. Therefore, they do not impact the Ku-Band Radar performance. On the other hand, range acceleration, target rotation, and small target RCS will be encountered in a space flight operational environment. Hence, range rate tracker performance data due to these effects is quite realistic.
Table 3.4-1 provides a case-by-case summary of the range rate analysis. This summary gives the standard deviation of the target acceleration, the range rate standard deviation for the Cine and Best data, a measure of the GDOP effects, and comments noting the most significant contributors for each test run. Notice that in some cases one error source dominates at the beginning of a flight and transitions to a second dominant source. Take GEM3 as an example. Once target rotation effects were removed, it was found that GDOP predominated in the first 200 seconds of the flight, while target acceleration effects predominated for the remainder of the flight. This case is examined in depth in Section 3.4.3.

3.4.1 Range Acceleration Effects

3.4.1.1 Analysis of Acceleration Effects on the Velocity Processor

Target range acceleration induces error in the Ku-Band Radar's velocity estimate. This error is generated in two places in the signal processing: (1) the discriminant formation process and (2) the smoothing filter at the velocity processor output. These two effects are analyzed below.

The velocity discriminant was designed under the assumption that the velocity was constant over the period (called a data cycle) during which the data is taken. Now, if the target is accelerating in range, the velocity will not be constant over the data cycle and the velocity discriminant will be distorted, causing an error in the velocity estimate. To determine the amount of distortion in this estimate, the signal processing prior to velocity discriminant formation must be examined.

The duration of a data cycle is 51.2 miliseconds for the 7 kHz PRF mode and 119 miliseconds for the 3 kHz PRF mode. In both cases, the radar processes a total of 320 return pulses through each of 2 range gates to form the velocity discriminant. The 320 pulses in each range gate are processed 16 consecutive pulses at a time to form the approximate doppler filter outputs via a discrete fourier transform (DFT). Since there are 640 return pulses for the two range gates, then there are 40 outputs formed for each doppler filter. For a given filter, the magnitude of these 40 outputs
<table>
<thead>
<tr>
<th>PROFILE</th>
<th>ACCELERATION STD DEV</th>
<th>RANGE RATE STD DEV</th>
<th>AVERAGE PREDICTED GDOP STD DEV</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAL1</td>
<td>1.27</td>
<td>1.38</td>
<td>1.14</td>
<td>GDOP is present at beginning of profile. Acceleration is a component of the error. Primary source is phase skewing.</td>
</tr>
<tr>
<td>* BAL2</td>
<td>3.58</td>
<td>3.08</td>
<td>2.95</td>
<td>GDOP effect in first 200 seconds of profile is significant. Removal of phase skewing from KU and WSMR data leaves error which is approximately the error from acceleration.</td>
</tr>
<tr>
<td>BAL5</td>
<td>3.2</td>
<td>1.2</td>
<td>.49</td>
<td>Phase skewing is primary source of error. Some error also due to acceleration.</td>
</tr>
<tr>
<td>* BAL6</td>
<td>3.82</td>
<td>1.38</td>
<td>1.07</td>
<td>GDOP effect in first 100 seconds of profile significant. Phase skewing between WSMR and KUBAND data. Acceleration also significant factor.</td>
</tr>
<tr>
<td>BAL7</td>
<td>2.3</td>
<td>2.9</td>
<td>1.03</td>
<td>Phase skewing is major error. GDOP is present at beginning of profile.</td>
</tr>
<tr>
<td>GEM2</td>
<td>6.5</td>
<td>2.77</td>
<td>.355</td>
<td>One spike in range rate causes excessive standard deviation. Some error is due to acceleration but majority is due to phase skewing between KU and WSMR data.</td>
</tr>
<tr>
<td>GEM3</td>
<td>4.83</td>
<td>1.83</td>
<td></td>
<td>Acceleration effect is significant. Phase skewing is also large part of error.</td>
</tr>
<tr>
<td>SAT1</td>
<td>6.03</td>
<td>2.33</td>
<td></td>
<td>GDOP intensified by oscillating acceleration.</td>
</tr>
<tr>
<td>SAT2</td>
<td>9.56</td>
<td>3.06</td>
<td>1.41</td>
<td>GDOP compounds acceleration effect in BEST data. Acceleration data is approximately correct.</td>
</tr>
<tr>
<td>* SAT3</td>
<td>13.14</td>
<td>6.78</td>
<td>1.85</td>
<td>Invalid BEST data due to large GDOP effect.</td>
</tr>
<tr>
<td>SAT4</td>
<td>2.22</td>
<td>.73</td>
<td>.65</td>
<td>Combination of Acceleration, and possibly some skewing in time.</td>
</tr>
<tr>
<td>PROFILE</td>
<td>ACCELERATION STD DEV</td>
<td>RANGE RATE STD DEV</td>
<td>AVERAGE PREDICTED GDOF STD DEV</td>
<td>COMMENTS</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------</td>
<td>--------------------</td>
<td>-----------------------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BEST</td>
<td>CINE</td>
<td></td>
</tr>
<tr>
<td>H30SKAE</td>
<td>2.16</td>
<td>1.47</td>
<td>.32</td>
<td>1.43</td>
</tr>
<tr>
<td>** H30SKAF</td>
<td>.80</td>
<td>1.72</td>
<td>.33</td>
<td>1.37</td>
</tr>
<tr>
<td>** H30SKAG</td>
<td>1.22</td>
<td>.84</td>
<td>.83</td>
<td>1.9</td>
</tr>
<tr>
<td>H30SKAH</td>
<td>1.309</td>
<td>2.21</td>
<td>1.14</td>
<td>1.4</td>
</tr>
<tr>
<td>H30SKAI</td>
<td>2.61</td>
<td>1.06</td>
<td>.49</td>
<td>1.17</td>
</tr>
<tr>
<td>HEL30AF</td>
<td>1.71</td>
<td>.75</td>
<td>.50</td>
<td>.35</td>
</tr>
<tr>
<td>HEL30AG</td>
<td>1.01</td>
<td>.371</td>
<td>.41</td>
<td>.32</td>
</tr>
<tr>
<td>HEL30AI</td>
<td>.97</td>
<td>.67</td>
<td>.36</td>
<td>.38</td>
</tr>
<tr>
<td>HEL30AJ</td>
<td>1.2</td>
<td>.76</td>
<td>.38</td>
<td>.38</td>
</tr>
<tr>
<td>* HJ146AC</td>
<td>.62</td>
<td>.32</td>
<td>.1</td>
<td></td>
</tr>
<tr>
<td>* HJ146AD</td>
<td>.87</td>
<td>.36</td>
<td>.60</td>
<td>.1</td>
</tr>
<tr>
<td>HJ146AE</td>
<td>.65</td>
<td>.36</td>
<td>.52</td>
<td>.1</td>
</tr>
<tr>
<td>PROFILE</td>
<td>ACCELERATION STD DEV</td>
<td>RANGE RATE STD DEV</td>
<td>AVERAGE PREDICTED GDOP STD DEV</td>
<td>COMMENTS</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------</td>
<td>--------------------</td>
<td>--------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HL146AE</td>
<td>.576</td>
<td>.44</td>
<td>.94</td>
<td>Low SNR is believed to cause error. No acceleration effect.</td>
</tr>
<tr>
<td>* HL246AD</td>
<td>.63</td>
<td>.49</td>
<td>.70</td>
<td>Acceleration data is no good. Believe SNR is problem. Cine data affected by clouds.</td>
</tr>
<tr>
<td>* HL246AE</td>
<td>.63</td>
<td>.71</td>
<td>.1</td>
<td>Believe large portion of error is due to low SNR.</td>
</tr>
<tr>
<td>* HL346AD</td>
<td>.54</td>
<td>.66</td>
<td>.60</td>
<td>Low SNR is believed to be major cause of error. Acceleration data was bad.</td>
</tr>
<tr>
<td>* HL346AE</td>
<td>.86</td>
<td>.51</td>
<td>.69</td>
<td>Acceleration data is invalid. Error is due to low SNR.</td>
</tr>
<tr>
<td>HL346AF</td>
<td>.366</td>
<td>.55</td>
<td>.82</td>
<td>No acceleration problem. SNR is the major problem.</td>
</tr>
<tr>
<td>* HL446AC</td>
<td>.55</td>
<td>.42</td>
<td>.55</td>
<td>Acceleration data is invalid. Cine is inhibited by clouds. Estimated range acceleration shows acceleration data is invalid. Believe Low SNR is the main problem.</td>
</tr>
<tr>
<td>* HL446AD</td>
<td>.66</td>
<td>.54</td>
<td>.1</td>
<td>Believe significant portion of error is due to low SNR. Acceleration is small over most of profile while velocity error is large over all of the profile.</td>
</tr>
<tr>
<td>HL446AE</td>
<td>.4</td>
<td>.51</td>
<td>1.25</td>
<td>No acceleration effect. Spikes in cine data make it suspect. Errors could be due to low SNR.</td>
</tr>
<tr>
<td>* HL546AC</td>
<td>.57</td>
<td>1.3</td>
<td>.1</td>
<td>Range rate output for KU and TMR diverge after PRF change. There are 3 to 4 foot/sec errors. Unknown cause.</td>
</tr>
<tr>
<td>* HL546AE</td>
<td>1.5</td>
<td>.67</td>
<td>.75</td>
<td>Examination of range rate plots demonstrate a correlation between acceleration and range rate error. Acceleration is main error source here.</td>
</tr>
<tr>
<td>HL546AF</td>
<td>1.28</td>
<td>.54</td>
<td>.1</td>
<td>Low SNR could cause noisy range rate. Acceleration effect is small.</td>
</tr>
<tr>
<td>HL546AG</td>
<td>.44</td>
<td>.46</td>
<td>.67</td>
<td>Low SNR could cause problem.</td>
</tr>
</tbody>
</table>
TABLE 3.4-1 SUMMARY OF RANGE ACCELERATION EFFECTS ON RANGE RATES (Page 4 of 4)

Range rate statistics are in ft/sec. Acceleration statistics are in ft/sec/sec. GDOP is a function for the geometry, therefore the standard deviation of the error is changing over the profile. The average predicted GDOP standard deviation is obtained by calculating the standard deviation of the range rate error for WSMR at each time interval and averaging this over the whole profile. This is also expressed in ft/sec.

1. BEST Acceleration data was used to calculate standard deviation of acceleration data unless otherwise indicated. Approximations of acceleration were used when acceleration data and range rate data were uncorrelated.

* indicates acceleration was estimated from a BEST \((\text{delta range rate})/(\text{delta time})\) calculation.

** indicates acceleration was estimated from a CINE \((\text{delta range rate})/(\text{delta time})\) calculation.
are computed and summed together (a process called post detection integration or PDI) to form an integrated filter output. The velocity discriminant is then formed by comparing the values of the filter on each side of the current velocity tracking filter. This gives a measure of the position of the target velocity within the center tracking filter.

One concern is the effect of acceleration on each formation of a 16 point DFT. Consider a range acceleration of 10 feet/sec\(^2\), the change in velocity over the 16 point DFT is 0.023 feet/second in the 7 kHz case and 0.053 feet/second in the 3 kHz case. In both cases, this turns out to be 0.3% of a filter width. This produces insignificant degradation in individual 16 point DFT outputs. The second problem in the velocity discriminant formation caused by acceleration is the change in the filter output value over the 20 filter output formations for a given range gate. Again, assuming a range acceleration of 10 feet/second\(^2\), the velocity changes 0.46 feet/second over the 20 filter formations in the 7 kHz PRF mode and 1.075 feet/second in the 3 kHz PRF mode. However, due to the PDI process the total change predicted by the radar velocity discriminant is just 1/2 of this value. The PDI process can be viewed as an averaging process and the error can be obtained from the following equations:

\[
V = \frac{1}{20} \sum_{n=1}^{20} (V_0 + n\Delta v)
\]

or

\[
V = V_0 + \frac{\Delta V}{20} \sum_{n=1}^{20} n
\]

or

\[
V = V_0 + 10 \frac{\Delta V_{21}}{20} = V_0 + 10 \Delta V
\]

where

- \(V\) = Radar velocity estimate at the end of a data cycle,
- \(V_0\) = actual target velocity at beginning of a data cycle.
- \(\Delta V\) = true change in target velocity over a 16 point DFT formation.
Now, the actual velocity at the end of a data cycle is given by $V_0 + 20\Delta V$ and the error is therefore $100V$. Thus, for the example of a 10 feet/sec$^2$ range acceleration, the velocity error due to the PDI process would be 0.23 feet/second in the 7 kHz PRF mode and 0.54 feet/second in the 3 kHz PRF mode. This is a significant velocity error source. A complete, exact detailed analysis of the velocity error due to the velocity discriminant formation is given in Appendix F.

The second source of range acceleration error occurs in the moving window averager at the output of the velocity processor (see Figure 3.4-1). In the 7 kHz PRF mode the moving window filter averages the present data cycle velocity estimates with the previous 3 data cycle estimates. In the 3 kHz PRF mode the filter averages the present data cycle estimate with 1 previous data cycle estimate. For a given range acceleration value, this filtering produces the same error effect as the PDI processor. The estimated velocity in this case can be expressed as

$$\begin{align*}
(3-2) \quad V &= \frac{1}{N} \sum_{n=0}^{N-1} (V_0 + n\Delta V_D) \\
\text{or} \quad V &= V_0 + \Delta V_D \sum_{n=0}^{N-1} \frac{n}{N} \\
\text{or} \quad V &= V_0 + \Delta V_D \frac{N-1}{2} \quad \text{(Radar estimate)}
\end{align*}$$
FIGURE 3.4-1 ILLUSTRATION OF RANGE ACCELERATION ERROR SOURCES IN THE RANGE RATE SIGNAL PROCESSING
and the actual velocity is given by

\[ V = V_0 + (N-1) \Delta V_D \]  

where \( V_0 \) = true velocity at the beginning of the averaging period,

\( \Delta V_D \) = change in true velocity over one data cycle,

\( N \) = moving window filter width.

Clearly, the error induced by the moving window filter in the presence of range acceleration is \((N-1) \Delta V_D/2\). Using a range acceleration of 10 feet/sec\(^2\), \( \Delta V_D \) is 0.512 feet/sec and the induced error is 0.768 feet/second in the 7 kHz PRF mode. In the 3 kHz PRF mode, \( \Delta V_D \) is 1.19 feet/second and the induced error is 0.595 feet/second.

Combining the errors caused by the PDI processor and the moving window filter one obtains the following expressing for the radar velocity estimate,

\[ V = \frac{1}{N} \sum_{n=0}^{N-1} (V_0 - 10 \Delta V + n\Delta V_D) \]

or

\[ V = V_0 - 10 \Delta V + \frac{N-1}{2} \Delta V_D \]

or

\[ V = V_0 - \frac{\Delta V_D}{2} + \frac{N-1}{2} \Delta V_D \]

3-45
Subtracting equation 3-4 from 3-3 the velocity error estimate is

\[
(3-5) \quad \text{TOTAL VELOCITY ERROR} = \frac{N}{2} \Delta v_D = \frac{N}{2} T_D A_R
\]

where

- \( T_D \) = Data Cycle Length
- \( A_R \) = Range Acceleration

for the 7 kHz PRF mode and a 10 feet/sec\(^2\) range acceleration, the error 1.02 feet/sec and for the 3 kHz PRF mode and the same acceleration, the error is 1.19 feet/sec. This is a significant error in either PRF case.

In summary, equation 3-5 can be used as a tool to estimate the Ku-Band Radar velocity error in the presence of target range acceleration. This result was applied to the SORTE generated range rate difference data to determine when acceleration was a significant error source. Results of this exercise are discussed below.

3.4.1.2 Range Acceleration Effects in the SORTE Data

A crude measure used to determine those test cases that might be affected by range acceleration error was to compute the standard deviation of the Best range acceleration data. Then analyze those cases with acceleration standard deviations that were greater than 1 foot/sec\(^2\). Table 3.4-2 summarizes the results of this exercise. It gives the range acceleration standard deviation and the range rate difference standard deviation referenced to the Best data and the Cine data when available.

All of the SAT tests, except SAT4, appear to have the highest range acceleration standard deviation and correspondingly high delta range rate standard deviations. Since the target was a tethered GEM sphere that was reeled in and out very slowly, it is clear that, in fact, there was very little range acceleration. Further analysis revealed that GDOP contributed significant random error to the TMR (and Best) range rate data, producing a highly corrupted Best range acceleration data as well. GDOP was a significant factor due to the target's position (low altitude, directly over the brass.
TABLE 3.4-2 TEST CASES WHERE RANGE ACCELERATION WAS AN APPARENT PROBLEM

<table>
<thead>
<tr>
<th>PROFILE</th>
<th>BEST RANGE ACCELERATION</th>
<th>DELTA RANGE RATE</th>
<th>PROFILE</th>
<th>BEST RANGE ACCELERATION</th>
<th>DELTA RANGE RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STANDARD DEVIATION</td>
<td>STANDARD DEVIATION</td>
<td></td>
<td>STANDARD DEVIATION</td>
<td>STANDARD DEVIATION</td>
</tr>
<tr>
<td>BAL1</td>
<td>1.27</td>
<td>1.38</td>
<td>ND</td>
<td>1.38</td>
<td>1.41</td>
</tr>
<tr>
<td>BAL2</td>
<td>3.58</td>
<td>3.08</td>
<td>ND</td>
<td>3.08</td>
<td>3.10</td>
</tr>
<tr>
<td>BAL5</td>
<td>3.20</td>
<td>1.20</td>
<td>ND</td>
<td>1.20</td>
<td>1.20</td>
</tr>
<tr>
<td>BAL6</td>
<td>3.82</td>
<td>1.38</td>
<td>ND</td>
<td>1.38</td>
<td>1.38</td>
</tr>
<tr>
<td>BAL7</td>
<td>2.30</td>
<td>2.90</td>
<td>ND</td>
<td>2.90</td>
<td>2.90</td>
</tr>
<tr>
<td>GEM2</td>
<td>6.50</td>
<td>2.77</td>
<td>ND</td>
<td>2.77</td>
<td>2.77</td>
</tr>
<tr>
<td>GEM3</td>
<td>4.83</td>
<td>1.83</td>
<td>ND</td>
<td>1.83</td>
<td>1.83</td>
</tr>
<tr>
<td>SAT1</td>
<td>6.03</td>
<td>2.33</td>
<td>ND</td>
<td>2.33</td>
<td>2.33</td>
</tr>
<tr>
<td>SAT2</td>
<td>9.56</td>
<td>3.06</td>
<td>1.41</td>
<td>3.06</td>
<td>1.41</td>
</tr>
<tr>
<td>SAT3</td>
<td>13.14</td>
<td>6.78</td>
<td>1.85</td>
<td>6.78</td>
<td>1.85</td>
</tr>
<tr>
<td>SAT4</td>
<td>2.22</td>
<td>0.73</td>
<td>0.65</td>
<td>0.73</td>
<td>0.65</td>
</tr>
<tr>
<td>HEL30AF</td>
<td>1.71</td>
<td>0.75</td>
<td>0.50</td>
<td>0.75</td>
<td>0.50</td>
</tr>
<tr>
<td>HEL30AG</td>
<td>1.01</td>
<td>0.37</td>
<td>0.41</td>
<td>0.37</td>
<td>0.41</td>
</tr>
<tr>
<td>HL546AE</td>
<td>1.50</td>
<td>0.67</td>
<td>0.75</td>
<td>0.67</td>
<td>0.75</td>
</tr>
</tbody>
</table>

cap) relative to the 3 TMR radars. In these cases, the conclusion is that the TMR system is the principal contributing error source and that there is no problem with the Ku-Band Radar estimate.

In the SAT4 case, the target is still a tethered gem sphere, but at a much higher altitude nearly over the brass cap. Although not as severe as the first three SAT cases, GDOP again produces a significant apparent acceleration. Hence, GDOP is the primary contributor to the delta range rate behavior in this case. The effects of GDOP on the SAT cases is discussed in more detail in Section 3.4.2.

The group of test runs with the next highest apparent range acceleration standard deviations were the GEM and BAL tests. All of these tests consisted of releasing helium-filled GEM spheres at the brass cap and allowing them to fly freely. In this case, three factors contributed to the range acceleration standard deviation: (1) GDOP, especially at low altitude, (2) the spinning GEM sphere and (3) true target range acceleration. GDOP
effects are discussed in Section 3.4.2 and target rotation effects are discussed in Section 3.4.3.

Let's examine one of these cases in detail. Figures 3.4-2 and 3.4-3 show the BAL2 range rate difference data prior to and after compensating for target rotation effects, respectively. (Justification for the compensation is given in Section 3.4.3). This new data shows a significant reduction in the standard deviation. Also it will be shown in Section 3.4.2 that the major contributor in the first 125 seconds is GDOP. Now, let's analyze the remaining difference data (from 125 seconds to 300 seconds). The standard deviation of this data is approximately 0.67 feet/second, which is still beyond the specification limits. A significant contributor to this error is the target rotation effects. It turns out that the radar is tracking the spinning of the target as evidenced by the expanded plot of the Ku-Band Radar MDM range rate shown in Figure 3.4-4 during this period. The range rate is oscillatory in nature with peak-to-peak swings of 10 feet/second and a period of approximately 4 seconds. Close examination of Figure 3.4-4 reveals accelerations as high as 8 feet/sec$^2$. From the analysis of Section 3.4.1.1, this translates to a Ku-Band Radar range rate error of 0.82 feet/sec$^2$. This acceleration effects due to target rotation then becomes a significant contributor to the range rate error. In addition, there is some minor effect due to the target moving (accelerating) away from the Ku-Band Radar.

The range acceleration effects analysis for the remaining BAL tests and GEM tests follows in an identical manner to the BAL2 analysis provided above. The conclusions of those analyses are also identical.

The next group of tests that showed effects potentially caused by range acceleration was the HEL30 and the HL546 series. Consider the HEL30AF case. The range rate difference data of Figure 3.4-5 shows some definite trends rather than being purely random in nature. A comparison of the BEST range acceleration profile of Figure 3.4-6 indicates that the trends in the range rate difference profile are highly correlated with the range acceleration profile. Using the acceleration data of Figure 3.4-6 and equation 3.5, the expected Ku-Band Radar range rate error was computed and
TSS TEST DATA PROFILE BAL2
TEST DATE 11.4.85. REVISION 10

TO-61350. GMT=17.2.29.

DELTA RANGE RATE FT/SEC (KU - THR)

TIME SECONDS

MEAN- 0.00  STANDARD DEVIATION- 3.08

FIGURE 3.4-2 ILLUSTRATION OF OSCILLATION IN RANGE RATE DATA DUE TO TARGET ROTATION
FIGURE 3.4-3  BAL2 TMR RANGE RATE DIFFERENCE DATA AFTER COMPENSATING FOR TARGET ROTATION EFFECTS. NOTE: KUBAND DATA IS SHIFTED 1.6 DATA CYCLES RELATIVE TO THE TMR DATA.
FIGURE 3.4-6 EXPANDED VIEW OF THE OSCILLATION INDUCED IN THE KU BAND RADAR RATE DUE TO TARGET ROTATION
FIGURE 3.4-5 HEL30AF BEST RANGE RATE DIFFERENCE PROFILE TO BE COMPARED WITH RANGE ACCELERATION PROFILE OF FIGURE 3.4-6
FIGURE 3.4-6  HEL30AF BEST RANGE ACCELERATION PROFILE TO BE COMPARED WITH THE RANGE RATE DIFFERENCE PROFILE OF FIGURE 3.4-5

3-53
found to be too high by a factor of about 4. The range rate difference data referenced to the CINES (Figure 3.4-7) shows the scale factor to be reduced to 3, but this is still a significant discrepancy. To further probe this problem, the BEST profile for HEL30AF was used to drive the simulator. The simulation generated range rate was differenced with the BEST range rate data to produce the profile shown in Figure 3.4-8. This data gives the expected theoretical result. At the writing of this report, the source of the discrepancy in the data has not been resolved.

Analysis of the HEL30AG and HL546AE profiles gave similar results. Both profiles show high correlation between the BEST range rate difference data and the BEST range acceleration data. Also a scale factor error was found to be present in both cases. However the scale factor appeared to be closer to 2 rather than 3 or 4 as in the HEL30AF case.

3.4.2 GDOP Effects

3.4.2.1 A Qualitative Description of GDOP

Geometric Dilution of Precision (GDOP) is the name applied to the inaccuacies induced in a set of target measurements caused by the placement of the system sensors relative to the target and the random errors in the individual sensor measurements. A complete development of the theory of the TMR GDOP effects on range and range rate measurements is given in Appendix D.

One of the most significant facts that surfaced during the GDOP development can be described as follows. First, notice that the three TMR radars and the Ku-Band Radar lie approximately in a plane and the TMR radars surround the Ku-Band Radar (see Figure D-1). Also, observe that the TMR radars only supply target range and range rate measurements. Now, if the target is at low altitude over the brass cap or directly over the Ku-Band Radar, the TMR radars cannot measure the vertical component (or the Z-component) of the target velocity. Furthermore, any random errors in TMR measurements will translate into significant errors in the Z-component of velocity estimated by TMR. In this configuration, the Z-component translates to range rate as measured by the Ku-Band Radar. Hence, there is significant
FIGURE 3.4-7 HEL30AF CINE RANGE RATE DIFFERENCE PROFILE TO BE COMPARED WITH RANGE ACCELERATION PROFILE OF FIGURE 3.4-6
FIGURE 3.4-8  SIMULATION GENERATED HEL30AF RANGE RATE DATA REFERENCED TO THE HEL30AF BEST RANGE RATE DATA
GDOP error in the TMR range rate measurement. The situation described above is illustrated for a two dimensional case in Figure 3.4-9.

The general result of the above qualitative discussion is that any flight profile that puts the target at low altitude, directly over the Ku-Band Radar should have significant error in the TMR range rate measurements. Thus, we should expect TMR GDOP problems with the SAT1, SAT2, and SAT3 data. Also, GDOP problems should be found early in the flight for the BAL and GEM series of tests. To a lesser extent, one should expect GDOP problems with the H30SK series. Even though this flight profile is offset from the brass cap, it is still at a relatively low altitude.

Although there was not time to perform a GDOP analysis of the CINE sensor system, it is appropriate at this point to make some qualitative observations about the CINE GDOP performance in the situation described above. The CINE system develops the target position and velocity using azimuth, elevation, azimuth rate and elevation rate from each of 5 cinetheodolites. When the target is directly over the brass cap each individual cine should have reasonably good knowledge of the target vertical velocity component. Thus, contrary to the TMR system, the CINE system should experience very little problem with GDOP in the range rate measurement for the profiles cited above.

3.4.2.2 GDOP Analysis of SORTE Range Rate Data

The preliminary step in the analysis was to compute the standard deviation of the range rate error produced by GDOP at each point in the flight profile. Then the mean and standard deviation of this GDOP error profile was computed. In the preliminary analysis, the mean of the GDOP error profile was used to screen all of the test cases. If the mean of the GDOP error profile was greater than 0.25 feet/sec, then the test case was examined in further detail. Table 3.4-3 summarizes those cases with significant GDOP error problems that were analyzed in more detail. Results of those analyses are discussed below.
NOTE: NEITHER SENSOR'S MEASUREMENTS CONTAIN INFORMATION ABOUT Z

FIGURE 3.4-9 ILLUSTRATION OF A SEVERE GDOP VELOCITY SITUATION
<table>
<thead>
<tr>
<th>PROFILE</th>
<th>GDOP MEAN* RANGE RATE ERROR</th>
<th>DELTA RANGE RATE STANDARD DEVIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BEST</td>
</tr>
<tr>
<td>BAL1</td>
<td>1.14</td>
<td>1.38</td>
</tr>
<tr>
<td>BAL2</td>
<td>2.95</td>
<td>3.08</td>
</tr>
<tr>
<td>BAL6</td>
<td>1.07</td>
<td>1.38</td>
</tr>
<tr>
<td>BAL7</td>
<td>1.03</td>
<td>2.90</td>
</tr>
<tr>
<td>SAT1</td>
<td>--</td>
<td>2.33</td>
</tr>
<tr>
<td>SAT2</td>
<td>1.76</td>
<td>3.06</td>
</tr>
<tr>
<td>SAT3</td>
<td>2.97</td>
<td>6.78</td>
</tr>
<tr>
<td>H30SKAE</td>
<td>1.43</td>
<td>1.47</td>
</tr>
<tr>
<td>H30SKAF</td>
<td>1.37</td>
<td>1.72</td>
</tr>
<tr>
<td>H30SKAG</td>
<td>1.90</td>
<td>0.84</td>
</tr>
<tr>
<td>H30SKAH</td>
<td>1.40</td>
<td>2.21</td>
</tr>
<tr>
<td>H30SKAI</td>
<td>1.17</td>
<td>1.06</td>
</tr>
<tr>
<td>HEL30AF</td>
<td>0.35</td>
<td>0.75</td>
</tr>
<tr>
<td>HEL30AI</td>
<td>0.38</td>
<td>0.67</td>
</tr>
<tr>
<td>HEL30AJ</td>
<td>0.38</td>
<td>0.76</td>
</tr>
</tbody>
</table>

* This is the mean of the GDOP range rate error standard deviation profile.
GDOP induced range rate errors are very similar for the BAL and GEM series of tests. Figure 3.4-10 and 3.4-11 give typical examples of the GDOP range rate error for the GEM and the BAL tests, respectively. Both tests have similar shaped GDOP profiles; GDOP range error is large at the beginning of the flight and tapers off rapidly after 100 seconds or so. This behavior correlates perfectly with the qualitative description of GDOP given in Section 3.4.2.1. For these test cases, a helium-filled geosphere is released at the brass cap and allowed to free-fly. Hence, early in the flight, the target is at very low altitude, e.g. 1000 to 2000 feet. But the balloon rises rapidly to several thousand feet in altitude. Based on the qualitative discussion of Section 3.4.2.1, one would expect large GDOP range rate error at low altitude or early in the flight and small GDOP range rate error at high altitude or late in the flight. The behaviors of the GDOP computation shown in the two figures correlates perfectly with the intuitive explanation.

As further proof that the GDOP computation is correct, a range rate difference profile referenced to the TMR data was computed for the BAL7 profile and is plotted in Figure 3.4-12. A comparison of this profile with the BAL7 GDOP computation given in Figure 3.4-11, indicates good agreement in shape and magnitude between the two profiles. It shows that GDOP dominates over the first 200 seconds and that a different source range acceleration due to target rotation (as discussed in Section 3.4.1.2), dominates over the remainder of the flight.

One final observation about the GDOP calculations for BAL7 and GEM2 is warranted. The difference in magnitudes at the start of the profiles is due to the difference in altitude for initial tracking. As one would expect the initial altitude of BAL7 is much lower than for GEM2.

In the SAT series of tests, the helium-filled GEMsphere was tethered. For the SAT1 and SAT2 tests the range, and approximately the altitude, of the balloon was about 2500 feet above the Ku-Band Radar for the entire test. For the SAT3 test the initial range was 2500 feet and the balloon was reeled into 1200 feet final range. The range of the balloon in the SAT4 test was 10,000 to 12,000 feet. The SAT data of Table 3.4-3
FIGURE 3.4-10  GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION
PROFILE FOR GEM2

3-61
FIGURE 3.4-11  GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR BAL7

TEST DATE 11-4-85

MEAN = 1.03

FIGURE 3.4-11  GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR BAL7

3-62
FIGURE 3.4-12  BAL7 TMR RANGE RATE DIFFERENCE PROFILE AFTER COMPENSATION FOR TARGET ROTATION EFFECTS
correlates with these test descriptions, since we expect the test run with the lowest average altitude to have the worst average GDOP range rate error and worst difference range rate standard deviation. Figure 3.4-13 gives the GDOP range rate error standard deviation profile for the SAT2 test run.

Notice that GDOP range rate error does not appear to significantly affect the CINE data. That this is true can be seen by comparing the difference range rate data referenced to the BEST solution against the difference range rate data referenced to the CINE solution. In both the SAT2 and SAT3 runs, the standard deviation of the BEST data is 2-3 times greater than the CINE data. As was conjectured earlier, the CINE system is not susceptible to low altitude GDOP errors because the sensors in the system measure $\theta$ and $\dot{\theta}$, rather than $R$ and $\dot{R}$ which is measured by the TMR sensors.

In the H30SK series of runs, the target, a UH-1H helicopter, flew a 30 degree glide slope toward the radar, starting at a range of 4000 feet and altitude of 1700 feet and finishing at a range of 2000 feet and an altitude of 1500 feet. Figure 3.4-14 gives the GDOP range rate error standard deviation profile for a typical run (H30SKAC). This shows the anticipated behavior: GDOP increases with time because the altitude decreases with time.

A comparison of the CINE range rate difference data and the BEST range rate difference data for this series of test shows that GDOP range rate error is much less significant for the CINE system of sensors. This result is identical to the SAT series of tests and therefore similar comments apply.

The final tests shown in Table 3.4-3 is the HEL30 series. In these tests the helicopter flies toward the radar from a range of 12000 feet into 7000 feet. The starting altitude is 6000 feet and the final altitude is 5000 feet. Since these tests were at a higher altitude and further range than any of the previous sets of tests, one would expect the GDOP range rate error to be smaller than the other cases. This is verified by the data of Table 3.4-3. A GDOP range rate standard deviation plot for the HEL30AF profile is provided in Figure 3.4-15. This data confirms that the GDOP range rate error increases as the altitude decreases.
FIGURE 3.4-13  GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR SAT2
The HL- and HJ- series of tests consisted of tracking a helicopter at long range and high altitude for a duration of 10 minutes. All of the previous discussion on GDOP error would lead us to conclude that these tests, because of high altitude and long range, would have insignificant GDOP range rate error. The GDOP calculations given in Figures 3.4-16 and 3.4-17 support this conclusion for the HJ- and HL- series, respectively.

3.4.3 Target Rotation Effects

An examination of the range rate difference data referenced to the TMR for the GEM and BAL series of tests (see Figure 3.4-2) reveals peak-to-peak oscillations of 10 feet/second. Further investigations showed that the Ku-Band Radar range rate profile had peak-to-peak oscillations of 10 feet/second with a period of 4 seconds. This probing of the data also indicated that the TMR range rate profile also had oscillations with the same peak-to-peak value and period.

3.4.3.1 Evidence Supporting the Target Spin Theory

What was the source of these oscillations? It is conjectured that the source of these range rate oscillations was rotation of the GEMsphere with both the Ku-Band Radar and the TMR radars tracking slowly back and forth across the spinning balloon. There are two facts that lend support to this conjecture. First, examination of the SMMS rendezvous data from flight 41-C reveals a similar oscillation in range rate. In this case it was confirmed through visual observation that the SMMS was in fact rotating about its axis. Computation of the SMMS rotation speed from the peak-to-peak velocity value compared quite well with the rotation speed estimated from the visual observations. The second fact is that both the TMR radars and the Ku-Band Radar produced identical oscillatory patterns. Since these radars operate at widely different RF (2 GHz for the TMR and 14 GHz for the Ku-Band Radar) and the signal processing and waveforms are different, the observed effect must be generated by some mechanism that is independent of the radars. This leaves only the target and its dynamics. The only dynamics that would produce an oscillation in range rate is a spinning of the target.
FIGURE 3.4-15 GDOp-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR HEL30AF

TEST DATE 10-3-85

MEAN = .354
The HL- and HJ- series of tests consisted of tracking a helicopter at long range and high altitude for a duration of 10 minutes. All of the previous discussion on GDOP error would lead us to conclude that these tests, because of high altitude and long range, would have insignificant GDOP range rate error. The GDOP calculations given in Figures 3.4-16 and 3.4-17 support this conclusion for the HJ- and HL- series, respectively.

3.4.3 Target Rotation Effects

An examination of the range rate difference data referenced to the TMR for the GEM and BAL series of tests (see Figure 3.4-2) reveals peak-to-peak oscillations of 10 feet/second. Further investigations showed that the Ku-Band Radar range rate profile had peak-to-peak oscillations of 10 feet/second with a period of 4 seconds. This probing of the data also indicated that the TMR range rate profile also had oscillations with the same peak-to-peak value and period.

3.4.3.1 Evidence Supporting the Target Spin Theory

What was the source of these oscillations? It is conjectured that the source of these range rate oscillations was rotation of the GEMsphere with both the Ku-Band Radar and the TMR radars tracking slowly back and forth across the spinning balloon. There are two facts that lend support to this conjecture. First, examination of the SMMS rendezvous data from flight 41-C reveals a similar oscillation in range rate. In this case it was confirmed through visual observation that the SMMS was in fact rotating about its axis. Computation of the SMMS rotation speed from the peak-to-peak velocity value compared quite well with the rotation speed estimated from the visual observations. The second fact is that both the TMR radars and the Ku-Band Radar produced identical oscillatory patterns. Since these radars operate at widely different RF (2 GHz for the TMR and 14 GHz for the Ku-Band Radar) and the signal processing and waveforms are different, the observed effect must be generated by some mechanism that is independent of the radars. This leaves only the target and its dynamics. The only dynamics that would produce an oscillation in range rate is a spinning of the target.
FIGURE 3.4-16 GDOP-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR HJ146AD
TEST DATE 10-5-85

FIGURE 3.4-17 GDOp-INDUCED RANGE RATE ERROR STANDARD DEVIATION PROFILE FOR HL246AE
A natural question to ask is: does the spin rate computed from the oscillatory range rate data correspond to a reasonable value for a free-flying GEMsphere? This spin rate is computed from the following expression

\[ \theta = \frac{\Delta V}{R} \text{ (cycle/2\pi radians)} \]

where \( R \) = Radius of the GEMsphere
\( \Delta V \) = 1/2 the peak-to-peak range rate oscillation

Now, using \( R = 3 \) feet and \( \Delta V = 5 \) feet/second, it is found that the balloon is rotating at a speed 0.27 cycles/second or one revolution every 3.77 seconds. This rotation rate certainly seems reasonable, especially if there is any air turbulence to generate the tumbling or spinning effect.

3.4.3.2 Modified Analysis of the Difference Range Rate Data

If one accepts the conclusion that target spin produced the oscillation in the range rate data, then GEM and BAL range rate data must be re-evaluated using the following technique. First, observe that both TMR radar and Ku-Band Radar boresights oscillated back and forth over the target with a period of about 4 seconds. This period was identical for both the Ku-Band Radar and the TMR system as shown in Figure 3.4-18. However, a closer look at that data reveals that the oscillations are out of phase which is not surprising. This effect was denoted as “phase skewing” in Table 3.4-1 and this nomenclature will be retained in the sequel.

Now, to analyze the true radar performance, these target spinning effects must be removed. This is done by shifting the TMR range rate profile until the oscillations in this profile align with the Ku-Band Radar range rate profile oscillations. The aligned profiles are then differenced and the statistics of the resulting data are computed. The result of this process for the BAL7 profile is illustrated in Figure 3.4-3. The features found in that profile were then easily explained.
FIGURE 3.4-18 ILLUSTRATION OF THE PHASE DIFFERENCE BETWEEN THE KU BAND RADAR RANGE RATE AND THE TMR RANGE RATE

3-72
3.4.4 Low SNR Effects

Up to this point in the discussion, most of the range rate data problems have been reasonably well explained by the error sources discussed in the previous three subsections. However, there remain some test runs with apparent range rate problems that need to be addressed. These cases are part of the HL-series of tests. Table 3.4-4 summarizes those HL-runs that have significant problems in the BEST or TMR range rate difference data. These errors cannot be legitimately explained by target acceleration, rotation, or GDOP. What, then, can be the source? After some investigation, the following theory was developed. Examination of the HJ-series revealed the difference range rate statistics were much better than the HL. But both the HL- and HJ-series used a helicopter as a target and the flights were at about the same range and altitude. What was different between the flight trajectories of the two series? The HL series trajectory was a circular arc with Ku-Band Radar at the center, while the HJ series trajectory was more on a line directly toward the radar. These two trajectories are illustrated in Figure 3.4-19. This means that the helicopter was broadside to the radar in the HL series, but was nose-on to the radar in the HJ series. From the photograph of 3.1-3, one can see that the target enhancement devices (two Luneberg lenses mounted on the underside of the helicopter pointing forward) would help in the nose-on view, but would not provide much assistance in the broadside view.

It was learned during the System Design Verification Tests (SDVT) of the Ku-Band Radar that an UH-1H helicopter had a -5 to 5 dBSM RCS when viewed from broadside. It was also found during these tests that an SNR at the doppler filter output (denoted as SNR_D) of less than 10 dB caused visible degradation in the range rate performance and that the system breaks track for SNR_D less than 0 dB.

Using the information cited above one can compute the SNR_D for the HL series of runs from the expression

\[
(3-7) \quad \text{SNR}_D = 183.6 - 40 \log R \ (\text{FT}) + 10 \log \text{RCS} \ (\text{M}^2) + G
\]

where

- \( R \) = Range in feet
- RCS = Radar cross section in square meters
- \( G \) = Gain of the SNR through the digital processor
TABLE 3.4-4 SUMMARY OF THE HL-SERIES WITH PROBLEMS IN THE BEST OR TMR RANGE RATE DIFFERENCE DATA

<table>
<thead>
<tr>
<th>PROFILE</th>
<th>RANGE RATE DIFFERENCE STD. DEV., FT/SEC</th>
<th>REFERENCE</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL146AE</td>
<td>0.44</td>
<td>TMR</td>
<td>Strong RCS here</td>
</tr>
<tr>
<td>HL246AD</td>
<td>0.48</td>
<td>BEST</td>
<td>Lens fading in and out</td>
</tr>
<tr>
<td>HL246AE</td>
<td>0.52</td>
<td>BEST</td>
<td>Lens fading in and out</td>
</tr>
<tr>
<td>HL346AD</td>
<td>0.66</td>
<td>BEST</td>
<td>Large gaps where target fades</td>
</tr>
<tr>
<td>HL346AE</td>
<td>0.51</td>
<td>BEST</td>
<td>Lens fading in and out</td>
</tr>
<tr>
<td>HL346AF</td>
<td>0.56</td>
<td>TMR</td>
<td>No RSS here - RCS value?</td>
</tr>
<tr>
<td>HL446AC</td>
<td>0.41</td>
<td>BEST</td>
<td>RCS between 0 and 10dBSM</td>
</tr>
<tr>
<td>HL446AD</td>
<td>0.54</td>
<td>BEST</td>
<td>Large gaps where target fades</td>
</tr>
<tr>
<td>HL446AE</td>
<td>0.51</td>
<td>TMR</td>
<td>Target 0 dBSM</td>
</tr>
<tr>
<td>HL546AC</td>
<td>1.34</td>
<td>TMR</td>
<td>No RSS here - very small RCS</td>
</tr>
<tr>
<td>HL546AE</td>
<td>0.67</td>
<td>BEST</td>
<td>Large gaps where target fades</td>
</tr>
<tr>
<td>HL546AF</td>
<td>0.54</td>
<td>TMR</td>
<td>Lens fading in and out</td>
</tr>
<tr>
<td>HL546AG</td>
<td>0.46</td>
<td>TMR</td>
<td>Strong RCS here</td>
</tr>
</tbody>
</table>

3-74
FIGURE 3.4-19  FLIGHT GEOMETRIES FOR HJ AND HL SERIES OF EXPERIMENTS SHOWING ORIENTATION OF LUNEBERG LENS WITH RESPECT TO KUBAND RADAR
Assuming a target range of 47000 feet and a gain of 32, equation 3-7 reduces to

\[
\text{SNR}_D = 11.76 + \text{RCS (dBMS)}
\]

Now, if the broadside RCS of a UH-1H helicopter fluctuates between -5 and 5 dBMS, then the SNR \(_D\) fluctuates between 6.8 dB and 16.8 dB. From this calculation and the system test observations provided above, one can see that it is possible for the range rate estimate of the Ku-Band Radar to be corrupted by internal noise due to a weak return signal.

As further evidence to corroborate the effects of a weak target return signal on the range rate difference data performance, Figure 3.4-20 compares the target RCS profile against the range rate difference profile. One can see that there is a high correlation between the RCS strength and the range rate random error behavior. It is conjectured that the gaps in RCS are due to the lens moving out of view of the radar.

3.5 ROLL AND PITCH ANGLE DATA ANALYSIS

The first pass through the roll and pitch angle data in the analysis procedure is summarized in Table 3.5-1. The first thing that is apparent in this data is that the number of failing cases is lopsided toward the BEST/TME cases, and that there are virtually no CINE failures. Based on the analysis of the range and range rate data presented in the previous sections, one immediately suspects that GDOP-induced error plays a major role in most of these failures. To support this conjecture, most of the failures would have to be in those flights at low altitudes and very nearly over the PEARL site brass cap. This would principally include the following family of profiles: SAT, BAL, GEM, H30SK and, to a lesser extent, HEL30. Table 3.5-2 provides a breakdown of the failures by flight series. This data shows that the majority of the angle specification failures occur for the GEM through the H30SK series of flights. The failures listed for the HL- and HJ- series of flights will be shown to be caused by other error sources as well; namely, (1) angle acceleration and (2) weak target return signal strength.
FIGURE 3.4-20 COMPARISON OF THE RADAR CROSS SECTION PROFILE AND THE RANGE RATE DIFFERENCE PROFILE FOR HL346AD
TABLE 3.5-1 SUMMARY OF INITIAL ROLL AND PITCH ANGLE PERFORMANCE ASSESSMENT

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SPEC</th>
<th>NUMBER FAILING</th>
<th>PERCENT</th>
<th>NUMBER FAILING</th>
<th>PERCENT</th>
<th>TOTAL PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll Angle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.667 deg</td>
<td>5</td>
<td>8.0</td>
<td>1</td>
<td>1.6</td>
<td>9.6</td>
</tr>
<tr>
<td>STD DEV</td>
<td>0.153 deg</td>
<td>23</td>
<td>37.0</td>
<td>4</td>
<td>6.4</td>
<td>43.4</td>
</tr>
<tr>
<td>Pitch Angle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.667 deg</td>
<td>8</td>
<td>12.9</td>
<td>1</td>
<td>1.6</td>
<td>14.5</td>
</tr>
<tr>
<td>STD DEV</td>
<td>0.153 deg</td>
<td>11</td>
<td>17.7</td>
<td>1</td>
<td>1.6</td>
<td>19.3</td>
</tr>
</tbody>
</table>

*The data in this table is based on a combined total of 62 sets of data.

TABLE 3.5-2 CATEGORIZATION OF ROLL AND PITCH ANGLE FAILURES BY FLIGHT SERIES

<table>
<thead>
<tr>
<th>SERIES</th>
<th>NO. OF PITCH FAILURES</th>
<th>NO. OF ROLL FAILURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERIES</td>
<td>NUMBER IN SERIES</td>
<td>MEAN</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>GEM</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>BAL</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>SAT</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>H30SK</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>HEL30</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>HL146 TO 546'</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>HJ146</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
The second observation about the data of Tables 3.5-1 and 3.5-2 is that the roll angle standard deviation failures outnumber the pitch angle standard deviation failures two to one. This also is suspected to be related to GDOP. A quantitative analysis is being done concurrently with the writing of this report to confirm this conjecture.

3.5.1 Description of Angle Error Sources

Before launching into a description of the roll angle and pitch angle analysis, it is worthwhile to list some of the sources that induce error in the angle data. These sources are:

- GDOP
- Coordinate Transformation Inaccuracy
- Angle Acceleration
- Weak Target Return Signal (low SNR)

In the present set of tests, GDOP is the primary source causing failure. The other three sources are equally weighted and are a distant second. A short description of each of these errors sources follows.

GDOP. As discussed in the previous subsections, when the target is at low altitude over the PEARL site brass cap, the TMR sensor system develops a very poor estimate of the target's brass cap Z-coordinate and Z-velocity. This translates into poor range and range rate when the target is over the brass cap. Also observe that this same poor estimate of the target position is folded into the calculation of the target's roll and pitch angle. But, because this calculation is a nonlinear transformation, it is hard to guess the effects of the brass cap Z-component errors on roll and pitch. For reference, the following expression for roll and pitch are provided.

\[
\text{Roll Angle} = \arctan \left( \frac{Y_B}{Z_B} \right) \\
\text{Pitch Angle} = \arctan \left( \frac{X_B}{(Y_B^2 + Z_B^2)^{1/2}} \right)
\]

where \((X_B, Y_B, Z_B)\) is the target position in the shuttle body coordinate system. The position in body coordinates is obtained from the position in brass cap coordinates through the transformation:
(3-10) \[(X_B, Y_B, Z_B) = T_{BP} (X_p, Y_p, Z_p)\]

where \(T_{BP}\) is the transformation matrix and \((X_p, Y_p, Z_p)\) is the target position in PEARL site brass cap coordinates. The elements of \(T_{BP}\) are fixed by the orientation of the radar relative to the brass cap at WSMR. \(Z_p\) is the brass cap component of importance in this analysis. For the errors along this axis are large due to the geometry of the TMR radars.

**Coordinate Transformation Inaccuracies.** Prior to the discovery of GDOP as the principal error source in the angle data analysis, a significant amount of time was spent analyzing the effects of errors in the coordinate transformation, \(T_{BP}\), on the angle difference data. These errors take the form of inaccurate estimate of the four angles that compose this transformation: (1) the lower azimuth angle rotation about the brass cap Z-axis, (2) the elevation angle rotation about the new y-axis, (3) the upper azimuth angle rotation about the new Z-axis, and (4) another rotation about Z which transfers the data from the radar frame (Reference 9) to the shuttle body coordinate system. Nominal values for these angles are 30 degrees for the lower azimuth, 30 degrees for elevation, 0 degrees for upper azimuth, and 24.5 degrees for the final rotation. If any of these measured angles are in error, this produces a misalignment between the desired and the actual coordinate system. Appendix E provides a detailed analysis of the effect of misalignment on the computed roll angles.

The results of the analysis was that small errors in any of these angles can produce significant bias in the angle difference data. Consider an example. The value of the lower azimuth angle was changed from 30 to 30.5 degrees and the angle difference data for HEL30AF was recomputed. Figure 3.5-1 compares the original roll angle difference data against the modified difference data. Clearly, the bias has been reduced in the modified data case.

The detailed analysis of angle transformation error effects also showed that the odd-shaped trends found in the angle difference data could not be explained by this error source alone. Hence, other sources were pursued.
FIGURE 3.5-1 ILLUSTRATION OF THE EFFECT OF A CHANGE IN LOWER AZIMUTH ANGLE ON THE ROLL ANGLE DIFFERENCE DATA FOR HEL30AF

Mean = -0.15 deg.
Standard Deviation = 0.07 deg.
Lower Azimuth = 30.0 deg.

Mean = 0.02 deg.
Standard Deviation = 0.056 deg.
Lower Azimuth = 30.5 deg.
Angle Acceleration. The present configuration of the angle tracking loop can produce an asymptotic bias in angle in the presence of angle acceleration. This loop can be modelled as a second order loop with the following transfer function:

\[
\frac{\hat{\theta}(s)}{\theta(s)} = \frac{\omega_n^2 + \omega_n^2 T_s}{s^2 + \omega_n^2 T_s + \omega_n^2}
\]

where \( T = 2/\omega_n \) for the design of this particular system. If the target is accelerated at rate, \( a \), (which means \( \theta = (a/2)t^2 \)), then using the final value therein from control theory one can compute the asymptotic bias of the loop from the relation:

\[
\text{Angle Bias} = \frac{a}{\omega_n^2}
\]

where \( \omega_n \) is the natural radian frequency of the loop and \( a \) is the angle acceleration. Consider an example. Let \( \omega_n = 0.754 \text{ Hz} \) and \( a = 0.04 \text{ deg./sec}^2 \), then the asymptotic angle bias is 0.07 degrees.

Weak Target Return Signal. This error source is due to a low SNR at the doppler filter output caused by a weak target return signal. This just means that the thermal noise from the receiver is beginning to compete with the desired signal. This will corrupt the angle discriminant which, in turn, corrupts the performance of the angle tracking loop. Unfortunately, an \( \text{SNR}_D \) threshold where the angle tracking begins to degrade rapidly is not known. However, it is guessed that \( \text{SNR}_D \) less than 7-8 dB will induce significant degradation in angle tracking performance. What does this mean in terms of a target RCS? Using equation 3-7 and a range of 45000 feet, an \( \text{SNR}_D \) less than 8 dB implies an RCS of less than -4.5 dBSM. RCS and ranges of these values are found in some of the HL- and HJ- series. Hence, the cause of poor angle tracking performance in these cases is suspected to be weak target signal returns.
3.5.2 Discussion of SORTE Angle Difference Data Problems

**BAL Series.** According to the flight log given in Appendix G only one radar (R394) was operating during the BAL series of tests. Hence, there is no true TMR solution available and therefore TMR GDOP-induced angle errors cannot exist for this case. The reason for the errors in these cases are not understood at this time.

Figure 3.5-2 gives the difference pitch angle profiles for BAL6 and BAL7. Observe that the error is very large early in the profile (or low altitude) and tapers off rapidly as the gemspheres gain altitude. This shape profile looks suspiciously like a GDOP-induced error. Thus, it is not clear that only one TMR radar was working in this case. This problem probably can be resolved through the official WSMR test logs.

**GEM Series.** Figure 3.5-3 gives the TMR pitch and roll angle difference data for GEM2 and Figure 3.5-4 gives a similar plot for GEM3. These difference profiles have the same shape as the corresponding range and range rate profiles. It is conjectured that GDOP is the dominant error source early in the flight (through the first 150 seconds). In the latter portion of the flight, both roll and pitch level off to a constant bias term. The source of this bias error is probably due to error in the coordinate system transformation as discussed in Section 3.5.1. The angle accelerations involved are at least an order of magnitude too small to produce the bias indicated in the figures.

The initial pitch and roll error for GEM3 is slightly worse than the corresponding values for GEM2. In addition, the GEM3 profile starts at an altitude of approximately 1600 feet, while GEM2's initial altitude is about 2000 feet. These facts are consistent with the earlier description of GDOP-induced error as a function of altitude.

Concurrent with the writing of this report, an effort is under way to quantitatively verify the theory described above.
FIGURE 3.5-2 ILLUSTRATION OF THE PITCH ANGLE DIFFERENCE DATA FOR THE BAL6 AND BAL7 PROFILES
FIGURE 3-5-3  THR ROLL AND PITCH ANGLE DIFFERENCE DATA FOR THE GEM2 PROFILE

3-85
FIGURE 3.5-4  TMR ROLL AND PITCH ANGLE DIFFERENCE DATA
FOR THE GEM2 PROFILE
H30SK Series. Of this series of tests, H30SKAI gave the best results, i.e. the fewest specification failures. The major difference between this profile and the others is that the target flew to a final range of 3300 feet, rather than 2000 feet. Assumption of GDOP as the primary error source would fit this failure pattern quite well.

Of the remaining flights in this series, H30SKAH gave the worst results in roll and pitch angle difference data. Since the flight paths were quite similar, it is hard to decide just what the difference might be. Close examination reveals that the GDOP error, which is considered the principal source here, is quite sensitive to the ground track at these altitudes and ranges. In particular, Figure 3.5-5 shows that the pitch angle difference data and range difference data are both highly correlated with the Y- Brass cap coordinate. This lends support to the idea that errors are heavily position dependent.

Thus far, the H30SK series is the first series discussed where the CINE data is available as a reference. There are two observations we can make about this data and its relation to the TMR data. Firstly, the CINE roll and pitch angle differences for H30SKAH flight are well-behaved as shown in Figure 3.5-6. This is in direct contrast to the BEST data for the same flight. But remember the arguments from a previous section. While GDOP is a major problem for the TMR for targets at low altitude, directly over the brass cap, it is not a problem for the CINE sensor system. Hence, the data of Figure 3.5-6 does not conflict with the previously discussed data, but instead supports the conclusions of that discussion.

The second observation concerns the mean of the CINE data. The data of Figure 3.5-6 shows that there is a significant bias in the pitch angle (0.5 - 0.6 degrees) and a bias of approximately 0.2 - 0.3 degrees in range. These biases are consistent for all of the H30SK flights, including H30SKAI. The principal source of this error is believed to be errors in the angles of the brass cap-to-shuttle body coordinate transformation matrix. This data lends support to the GEM data analysis as well. The GEM angle difference data (see Figures 3.4-3 and 3.4-4) decayed to a fixed bias level. The H30SK CINE supports the argument that this bias is the result of transformation error and not a residue of GDOP.

3-87
Figure 3.5-5 Illustration of high correlation between Y-brass gap coordinate and the angle difference data.
FIGURE 3.5-6  CINE ROLL AND PITCH ANGLE DIFFERENCE DATA FOR H30SKAH TO BE COMPARED WITH THE BEST DATA OF FIGURE 3.5-5
**HEL30 Series.** Of the four flights in this series, HEL30AJ has the worst performance. The roll and pitch angle difference data provided in Figure 3.5-7 shows that while GDOP does affect the error in the first 300 seconds, the error rapidly increases in the last 100 seconds. As noted in previous discussions, GDOP not only increases with decreasing altitude but is very sensitive to the X-Y ground track. Figure 3.5-8 shows the X-Y ground track. The last 100 seconds of this profile correlates well with the roll and pitch data because it shows the target flying directly toward the brass cap.

Since all of the HEL30 flight profiles were quite similar, one wonders why the errors vary significantly from flight-to-flight. A closer examination of the data revealed that in both HEL30AJ and HEL30AI the final altitude was the lowest (3200' to 4000') and the errors in these two cases were the worst. On the other hand, HEL30AF and HEL30AG both had a final altitude of 5000 feet and both had significantly better angle difference data performance.

At this point it would be best to have quantitative calculations to support these conclusions. Unfortunately, this work is being done in parallel with the final report.

Finally, to add further support to the conclusion that the error shown in Figure 3.5-7 is a function of the TMR radar, Figure 3.5-9 gives the CINE pitch angle difference data for HEL30AJ. This data clearly shows there is not a problem with the Ku-Band roll and pitch angle estimates.

**HL- and HJ- Series.** Table 3.5-3 summarizes the cases that failed in the HL- and HJ- series. An analysis of the individual cases generated the following observations.

The roll angle difference data of HJ146AC showed a high correlation with the -Z (or altitude) profile as shown in Figure 3.5-10. Since the CINE result showed no problem in roll angle, GDOP is suspected. Although this is somewhat surprising at this range. Also observe that a weak target return signal was not a problem in this case as the random component had peak-to-peak fluctuations of 0.1 degrees. (The data set for HJ146AE was missing.)
TEST DATA PROFILE HEL30AJ
TEST DATE 10.3.85. REVISION 10
TO-62488. GMT-17. 21. 28.

DELTA ROLL ANGLE DES - 10 - BEST

MEAN= -0.01 STANDARD DEVIATION= 0.73

FIGURE 3.5-7 BEST ROLL AND PITCH ANGLE DIFFERENCE
DATA FOR HEL30AJ
FIGURE 3.5-8 ILLUSTRATION OF X-Y GROUND TRACK FOR HEL30AJ
TO BE COMPARED WITH FIGURE 3.5-7

FIGURE 3.5-9 CINE PITCH ANGLE DIFFERENCE DATA TO BE COMPARED WITH FIGURE 3.5-7
FIGURE 3.5-10 ILLUSTRATION OF CORRELATION BETWEEN THE ROLL ANGLE DIFFERENCE DATA AND THE BEST ALTITUDE PROFILE
<table>
<thead>
<tr>
<th>PROFILE</th>
<th>TMR/BEST MEAN</th>
<th>STD. DEV.</th>
<th>CINE MEAN</th>
<th>STD. DEV.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HJ146AC</td>
<td>--</td>
<td>0.1573R</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>HJ146AE</td>
<td>--</td>
<td>0.2364R</td>
<td>--</td>
<td>0.2359R</td>
</tr>
<tr>
<td>HL346AE</td>
<td>--</td>
<td>0.1870R</td>
<td>--</td>
<td>0.1622R</td>
</tr>
<tr>
<td>HL446AC</td>
<td>--</td>
<td>0.2439R</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>HL446AE</td>
<td>--</td>
<td>0.1972R</td>
<td>--</td>
<td>0.1968R</td>
</tr>
<tr>
<td>HL546AC</td>
<td>-0.6859R</td>
<td>2.8925R</td>
<td>ND</td>
<td>ND</td>
</tr>
<tr>
<td></td>
<td>0.9770P</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

P = Pitch  
R = Roll  
ND = No Data

The problem in the HL346AE data appears to be related to angle acceleration. The roll angle difference data of Figure 3.5-11 shows a significant increase in error around 200 seconds. The size of the change in bias would indicate an angle acceleration (or deceleration) with a magnitude of about 0.01 deg/sec². The CINE roll angle difference data shows the same major feature.

The trend in the HL446AC BEST roll angle difference data is apparently caused by GDOP, since it appears to be highly correlated with the altitude data as shown in Figure 3.5-12. If the problems were due to coordinate transformation, then the trends would have been found in the CINE data. Also a weak target return signal strength is not a problem as indicated by the peak-to-peak random fluctuations.

The problem with the HL446AE appears to be a glitch of about 2 degrees between 200 and 300 seconds into the flight. This glitch shows up in roll and pitch in both the CINE and the TMR data. The conjecture in this case is angle acceleration. The magnitude of the acceleration is on the order 0.04 deg/sec². Weak target return signal strength does not appear to be a problem in this case.
FIGURE 3.5-11  BEST AND CINE ROLL ANGLE DIFFERENCE DATA.  
THE NEGATIVE-GOING GLITCH IS DUE TO ANGLE 
ACCELERATION

3-95
FIGURE 3.5-12 ILLUSTRATION OF CORRELATION BETWEEN ROLL ANGLE DIFFERENCE DATA AND THE BEST ALTITUDE PROFILE
The HL546AC roll and pitch angle difference data are definitely corrupted by GDOP due to the target being at low altitude. Figure 3.5-13 compares the roll angle difference data with the altitude profile. High correlation between these two profiles is evident. Based on this data and the data from HJ146AC and HL446AC, GDOP appears to become a major factor for altitudes less than 5000 feet.

3.5.2.1 Explanation of GDOP-Induced Error in Angle at Long Range

When the analysis of the data was first started, it was thought that angle data failures in the long range cases, i.e. the HL- and HJ- series, would be for reasons other than GDOP just as in the range and range rate data analysis. However, the situation in this case is very different. An explanation of the difference follows.

In both the HJ- and HL- flight configurations, the roll and pitch angle calculations include the Z-component (or altitude component) of target position in the brass cap coordinate system. As explained in an earlier section, any time the target is very nearly in the plane containing the TMR radars, the error in the out-of-plane coordinate (or the Z-component) is extremely large. This is because the TMR radars measure range, so they can only achieve accurate X-Y target position components when the target is near the plane of radars.

Now, the CINEs do not have a problem measuring the Z-component in the HJ- and HL- case for two reasons. Firstly, the five CINEs were chosen to surround the target flight path as shown in Figure 3.1-5, so they will not have trouble with a long range target. Secondly, they do not have trouble with measuring a target's Z-component when the target is at low altitude near the plane of the CINEs.

From the argument of the previous paragraph, it can be concluded that the CINE Z-component (or altitude) data profile can be used as a reference to determine the error in the BEST Z-component (or altitude) profile. Figure 3.5-14 compares the CINE altitude profile to the BEST altitude profile for the HJ146AC flight. This comparison clearly shows the
FIGURE 3.5-13 ILLUSTRATION OF CORRELATION BETWEEN ROLL ANGLE DIFFERENCE DATA AND BEST ALTITUDE PROFILE
FIGURE 3.5-14 A COMPARISON OF THE CINE ALTITUDE AND THE BEST ALTITUDE FOR THE HJ146AC PROFILE
BEST altitude errors of 300 feet or more, especially at the lower altitudes. Now, how does this affect the roll and pitch angle accuracy? This is hard to answer for the present flight geometry. So let's simplify the situation. Assume that the pitch angle is zero and that the error is entirely in the roll angle. This situation is depicted in Figure 3.5-15.

![Figure 3.5-15 Illustration of Effect of Altitude Error on Roll Angle Estimate](image)

The error in roll angle can be calculated as follows:

Roll Angle Error = \( \frac{Z \cos(E)}{R} \)

where
- \( Z \) = Altitude Error = 500 feet
- \( E \) = Elevation Angle = 6.3 degrees
- \( R \) = Range = 45,000 feet

Using the above values for \( Z \), \( E \), and \( R \), the roll angle error is 0.63 degrees. This magnitude of the error fits with the data presented for the HL- and HJ-series tests.
3.6 ILOS ROLL AND PITCH ANGLE DATA ANALYSIS

Table 3.6-1 summarizes the results of the preliminary analysis of the ILOS roll and pitch range difference data. As in the range rate difference case, the number of failures in angle rate was quite alarming. Furthermore, it contradicts the flight rendezvous data. These data indicated a problem with the random component inside 1.9 nautical miles or in the widest tracker bandwidth case. For ranges greater than 1.9 nautical miles, the angle rate random component was well within specification.

**TABLE 3.6-1 SUMMARY OF INITIAL ILOS ROLL AND PITCH RATE PERFORMANCE ASSESSMENT**

<table>
<thead>
<tr>
<th></th>
<th>BEST/TMR</th>
<th>CINE</th>
<th>COMBINED</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARAMETER</td>
<td>SPEC</td>
<td>NUMBER</td>
<td>Failing</td>
</tr>
<tr>
<td>Roll Angle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>'0.0027 deg/sec'</td>
<td>33</td>
<td>53.2</td>
</tr>
<tr>
<td>STD DEV</td>
<td>'0.0027 deg/sec'</td>
<td>36</td>
<td>58.0</td>
</tr>
<tr>
<td>Pitch Angle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>'0.0027 deg/sec'</td>
<td>36</td>
<td>58.0</td>
</tr>
<tr>
<td>STD DEV</td>
<td>'0.0027 deg/sec'</td>
<td>36</td>
<td>58.0</td>
</tr>
</tbody>
</table>

*There are a total of 62 difference data sets.*

A second pass through the data showed that the only two cases passing the mean specification were SAT1 and SAT2. As it turns out, these two tests were the only two tests where the target remained stationary with respect to angular motion for the entire test. So, the preliminary indication was that there was something wrong in those cases with angular motion, which does not occur in the cases with no angular motion. An extensive analysis was undertaken to determine the nature of this problem. Results of that analysis are described below.
3.6.1 Preliminary Analysis Results

The analysis was started by looking at the angle rate difference data sets. These looked awful! In some cases, the difference data sets looked like scaled down copies of the Ku MDM angle rate profiles. It was clear that these difference data sets would be of little value in resolving the problem. The real break in this case came when it was decided to compare the Ku MDM angle rate profile with the corresponding Ku MDM angle profile. This comparison of the pitch and pitch rate for H30SKAF is provided in Figure 3.6-1. A similar comparison for roll and roll rate is given in Figure 3.6-2.

Now, since the earth's rotation rate is quite small, the ILOS angle rate integrated over a fixed interval should be equal to the total angle change over that same interval. Let's apply this rule to the data of Figures 3.6-1 and 3.6-2. Consider the time interval from 40 to 60 seconds. The average value of the roll rate is about -0.5 degrees/second and the average value of the pitch rate is about -0.35 degrees/seconds. Integrating over 20 seconds gives a total change of -10 degrees in roll and -7 degrees in pitch. Now, examining Figures 3.6-1 and 3.6-2 to determine the total angle change from the roll and pitch data, it is seen that the roll angle changes -5 degrees and the pitch angle changes about -3.5 degrees over the same 20 second interval. This tells us that either the angle or the angle rate is off by a factor of 2.

But, since the angle data analysis showed no such problem, it can be assumed that the scale factor problem is in the angle rate data.

At this point, several questions come to mind. What is the value of the scale factor? Is it a constant? What is the source of the error? The answer to the first two questions were easy. Additional analysis of the same data showed that scale factor was about 2 for the entire interval. Analysis of other data sets showed the same factor. The only exceptions to this rule were the tests conducted after the $k_5$ gain in the servo was increased by a factor of 4. In that case, the scale factor was 0.5. (This problem is addressed at the end of this section.)
FIGURE 3.6-1 A COMPARISON OF THE KU MDM PITCH ANGLE AND ILOS PITCH RATE PROFILES FOR H3OSKAF
FIGURE 3.6-2  A COMPARISON OF THE KU MDM ROLL ANGLE AND ILOS ROLL RATE PROFILES FOR H30SKAF
What is the source of this scale factor error? There are two places where this scale factor can corrupt the angle rate: (1) in the Ku-Band Radar itself, after the azimuth and elevation angle rates are converted to roll and pitch angle rates in the EA-1 microprocessor, and (2) in the data processing sequence developed for the SORTE program. In either case, a factor of 2 seems quite reasonable since that represents a slip of a single bit in the binary representation of the angle rate value. At the writing of this final report, both possibilities are being pursued.

Regardless of what the error source turns out to be, the Ku MDM angle rate data will be scaled down by a factor of 2. The scaled data will then be analyzed for other problems that were masked by the scale factor problem.

3.6.2 Description of Angle Rate Error Sources

There are several sources that can corrupt the angle rate data besides the scale factor problem. Among these are:

- GDOP
- Angle Acceleration
- Weak Target Return Signals

A discussion of each of these is provided below.

GDOP. This error source will have the same affect on the angle rate as on the angle. However, we are not interested in wrestling with GDOP problems in the present analysis. Therefore, only the CINE reference data will be used in this analysis, since this system as configured is immune to GDOP.

Angle Acceleration. As will be demonstrated in the next section, this was the primary error source in the data examined, once GDOP was removed. The effect of angle acceleration on the ILOS angle rate tracking loop is identical to the acceleration effects on the angle tracking loop described in equations 3-11 and 3-12. That is, prolonged angle acceleration
produces an asymptotic bias in the ILOS angle rate estimate. This can be ascertained from the following arguments.

Figure 3.6-3 illustrates the analog second order loop which is used to represent the ILOS angle rate tracking loop in the following analysis. The transfer function for this loop can be expressed as

\[
\dot{\theta}(s)/\theta(s) = s\frac{w_n^2}{s^2 + w_n^2Ts + w_n^2}
\]  

(3-13)

Since the loop is critically damped, then \( T = 2/w_n \) where \( w_n \) is the natural frequency of the loop, and \( T \) is the loop settling time. To determine the response to angle acceleration we set

(3-14) \[ \dot{\theta}(t) = \frac{At^2}{2} \] (angle position)

or \[ \ddot{\theta}(t) = At \] (angle rate)

or \[ \dddot{\theta}(t) = A \] (angle acceleration)

where \( A \) is the angle acceleration and the \( \dot{\theta} \) notation represents the derivative of the variable with respect to time. The Laplace transform of this quantity is

(3-15) \[ \theta(s) = A/s^3 \]

Using equations 3-13 and 3-15, the Laplace domain representation of the tracking loop response is

(3-16) \[ \dot{\theta}(s) = \frac{(A/s^2)(w_n^2)}{(s^2 + 2w_nTs + w_n^2)} \]

The inverse Laplace transform of 3-16 is

(3-17) \[ \dot{\theta}(t) = At(1+exp(-w_n t)) - \left(\frac{2A}{w_n}\right)(1-exp(-w_n t)) \]
FIGURE 3.6-3 SECOND ORDER ANALOG MODEL OF THE ANGLE AND ANGLE RATE TRACKING LOOPS

NOTE: FOR A CRITICAL DAMPED LOOP $\tau = \frac{2}{\omega_n}$
To obtain the error in the angle rate estimate, the true angle rate $-At$ is subtracted from equation 3-17. This gives

$$\Delta \dot{\theta}(t) = -At \exp(-w_n t) + (2A/w_n)(1-\exp(-w_n t))$$  

(3-18)

The asymptotic value is obtained by allowing $t$ to approach infinity, which gives

$$\Delta \dot{\theta} = 2A/w_n$$  

(3-19)

Now, what sort of angle rate error does this expression produce for the Ku-Band angle rate tracking loop parameters? In the widest bandwidth case, $w_n = 2\pi (0.12)$. If we consider an angle acceleration of 0.04 degrees per $\text{sec}^2$, this gives an angle rate bias of 0.11 degrees/sec. This is a significant amount of bias. For reference, Figure 3.6-4 shows the response of the angle and angle rate loops in the presence of a 0.04 deg/\text{sec}^2 constant acceleration.

**Weak Target Return Signal.** A weak target return signal produces a low SNR at the doppler filter output, which, in turn, produces noisy angle discriminants. These noisy angle discriminants get injected into the angle rate tracking loop filter which smooths the noise on the output angle rate estimate. The target return signal is usually the weakest at long range (greater than 40000 feet) where the angle rate tracker bandwidth is the narrowest. Now, the $\text{SNR}_D$ threshold required to produce out-of-spec performance is estimated to be 7-8 dB as in the angle tracking case. In the present set of tests, this condition will only be achieved in some of the long range tests, e.g. some of the HL- and HJ- series tests. In general, weak target return signals should not be a problem.
# Table 3.6-2 Comparison of Scaled and Unscaled Roll Rate Difference Data Statistics (Page 1 of 2)

<table>
<thead>
<tr>
<th>PROFILE</th>
<th>MEAN OLD</th>
<th>MEAN NEW</th>
<th>STANDARD DEVIATION OLD</th>
<th>STANDARD DEVIATION NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT2.CIN</td>
<td>1.40 E-3</td>
<td>-1.81 E-3</td>
<td>4.02 E-2</td>
<td>6.00 E-2</td>
</tr>
<tr>
<td>SAT2.BST</td>
<td>9.00 E-4</td>
<td>-7.55 E-5</td>
<td>4.95 E-2</td>
<td>7.68 E-2</td>
</tr>
<tr>
<td>SAT3.CIN</td>
<td>1.00 E-1</td>
<td>2.67 E-4</td>
<td>1.41 E-1</td>
<td>5.74 E-2</td>
</tr>
<tr>
<td>SAT3.BST</td>
<td>8.51 E-2</td>
<td>2.90 E-4</td>
<td>4.83 E-1</td>
<td>1.15 E-1</td>
</tr>
<tr>
<td>SAT4.CIN</td>
<td>-1.98 E-2</td>
<td>-3.38 E-3</td>
<td>3.71 E-1</td>
<td>2.52 E-2</td>
</tr>
<tr>
<td>SAT4.BST</td>
<td>-1.93 E-2</td>
<td>-4.00 E-3</td>
<td>3.82 E-2</td>
<td>2.89 E-2</td>
</tr>
<tr>
<td>H30SKAE.BST</td>
<td>2.33 E-1</td>
<td>1.38 E-2</td>
<td>1.10 E-1</td>
<td>4.92 E-2</td>
</tr>
<tr>
<td>H30SKAE.CIN</td>
<td>2.39 E-1</td>
<td>5.19 E-3</td>
<td>1.04 E-1</td>
<td>2.74 E-2</td>
</tr>
<tr>
<td>H30SKAF.BST</td>
<td>4.07 E-2</td>
<td>-1.16 E-2</td>
<td>1.91 E-1</td>
<td>8.45 E-2</td>
</tr>
<tr>
<td>H30SKAF.CIN</td>
<td>3.44 E-2</td>
<td>8.76 E-4</td>
<td>1.88 E-3</td>
<td>5.72 E-2</td>
</tr>
<tr>
<td>H30SKAG.BST</td>
<td>9.00 E-4</td>
<td>3.32 E-2</td>
<td>7.88 E-2</td>
<td>3.12 E-2</td>
</tr>
<tr>
<td>H30SKAG.CIN</td>
<td>5.33 E-2</td>
<td>9.13 E-3</td>
<td>1.05 E-1</td>
<td>6.47 E-2</td>
</tr>
<tr>
<td>H30SKAH.BST</td>
<td>6.41 E-2</td>
<td>6.49 E-3</td>
<td>1.44 E-1</td>
<td>7.48 E-2</td>
</tr>
<tr>
<td>H30SKAH.CIN</td>
<td>7.26 E-2</td>
<td>-1.25 E-3</td>
<td>1.41 E-1</td>
<td>6.31 E-2</td>
</tr>
<tr>
<td>H30SKAI.BST</td>
<td>1.02 E-1</td>
<td>2.35 E-3</td>
<td>2.16 E-2</td>
<td>4.33 E-2</td>
</tr>
<tr>
<td>H30SKAI.CIN</td>
<td>6.08 E-2</td>
<td>-4.16 E-3</td>
<td>8.88 E-2</td>
<td>4.27 E-2</td>
</tr>
<tr>
<td>HEL30AF.BST</td>
<td>1.04 E-1</td>
<td>5.39 E-3</td>
<td>9.13 E-2</td>
<td>1.99 E-2</td>
</tr>
<tr>
<td>HEL30AF.CIN</td>
<td>1.04 E-1</td>
<td>4.20 E-3</td>
<td>9.14 E-2</td>
<td>2.21 E-2</td>
</tr>
<tr>
<td>HEL30AG.BST</td>
<td>5.30 E-3</td>
<td>4.27 E-4</td>
<td>5.04 E-2</td>
<td>2.09 E-2</td>
</tr>
<tr>
<td>HEL30AG.CIN</td>
<td>6.00 E-3</td>
<td>-6.49 E-4</td>
<td>5.02 E-2</td>
<td>2.01 E-2</td>
</tr>
<tr>
<td>HEL30AI.BST</td>
<td>4.30 E-3</td>
<td>-6.15 E-3</td>
<td>4.07 E-2</td>
<td>2.13 E-2</td>
</tr>
<tr>
<td>HEL30AI.CIN</td>
<td>5.10 E-3</td>
<td>-7.81 E-3</td>
<td>4.09 E-2</td>
<td>1.41 E-2</td>
</tr>
<tr>
<td>HEL30AJ.BST</td>
<td>1.17 E-2</td>
<td>-1.21 E-3</td>
<td>5.91 E-2</td>
<td>3.59 E-2</td>
</tr>
<tr>
<td>HEL30AJ.CIN</td>
<td>1.31 E-2</td>
<td>-5.62 E-3</td>
<td>5.66 E-2</td>
<td>1.28 E-2</td>
</tr>
<tr>
<td>PROFILE</td>
<td>MEAN</td>
<td>STANDARD DEVIATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>--------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLD</td>
<td>NEW</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLD</td>
<td>NEW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL246AD.BST</td>
<td>7.64 E-2</td>
<td>-6.35 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.15 E-2</td>
<td>8.87 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL246AD.CIN</td>
<td>7.57 E-2</td>
<td>-6.83 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.60 E-3</td>
<td>4.59 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL246AE.BST</td>
<td>6.30 E-2</td>
<td>-4.06 E-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.35 E-2</td>
<td>8.69 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL246AE.CIN</td>
<td>6.32 E-2</td>
<td>-4.53 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.35 E-2</td>
<td>4.55 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL346AD.BST</td>
<td>6.20 E-2</td>
<td>-4.29 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.14 E-2</td>
<td>1.11 E-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL346AD.CIN</td>
<td>6.34 E-2</td>
<td>-4.39 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.70 E-3</td>
<td>7.04 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL346AE.BST</td>
<td>7.55 E-2</td>
<td>-4.26 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.54 E-2</td>
<td>1.04 E-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL346AE.CIN</td>
<td>7.60 E-2</td>
<td>-5.01 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.52 E-2</td>
<td>8.20 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL446AC.BST</td>
<td>7.06 E-2</td>
<td>-5.95 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.05 E-2</td>
<td>1.65 E-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL446AC.CIN</td>
<td>6.90 E-2</td>
<td>-5.18 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.70 E-3</td>
<td>5.17 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL446AD.BST</td>
<td>6.08 E-2</td>
<td>-4.93 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.88 E-2</td>
<td>8.18 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL446AD.CIN</td>
<td>6.09 E-2</td>
<td>-5.05 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.87 E-2</td>
<td>4.92 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL546AE.BST</td>
<td>9.06 E-2</td>
<td>-4.36 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.30 E-2</td>
<td>8.55 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL546AE.CIN</td>
<td>9.07 E-2</td>
<td>-4.58 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.26 E-2</td>
<td>5.36 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL546AC.TMR</td>
<td>7.52 E-2</td>
<td>1.17 E-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.40 E-2</td>
<td>1.21 E-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HL546AF.TMR</td>
<td>7.51 E-2</td>
<td>-5.05 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.07 E-2</td>
<td>6.87 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HJ146AC.BST</td>
<td>6.00 E-2</td>
<td>-3.41 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.67 E-2</td>
<td>1.21 E-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HJ146AC.CIN</td>
<td>6.34 E-2</td>
<td>-3.09 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.36 E-2</td>
<td>5.78 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HJ146AD.BST</td>
<td>5.88 E-2</td>
<td>-5.04 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.43 E-2</td>
<td>7.74 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HJ146AD.CIN</td>
<td>5.89 E-2</td>
<td>-5.19 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.41 E-2</td>
<td>6.61 E-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROFILE</td>
<td>MEAN</td>
<td>STANDARD DEVIATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>--------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLD</td>
<td>NEW</td>
<td>OLD</td>
<td>NEW</td>
</tr>
<tr>
<td>SAT2.CIN</td>
<td>2.49 E-2</td>
<td>-1.27 E-3</td>
<td>1.18 E-1</td>
<td>4.05 E-2</td>
</tr>
<tr>
<td>SAT2.BST</td>
<td>2.32 E-2</td>
<td>-5.67 E-4</td>
<td>1.04 E-1</td>
<td>2.91 E-2</td>
</tr>
<tr>
<td>SAT3.CIN</td>
<td>-7.40 E-3</td>
<td>-1.84 E-2</td>
<td>1.39 E-1</td>
<td>4.75 E-1</td>
</tr>
<tr>
<td>SAT3.BST</td>
<td>-8.20 E-3</td>
<td>-5.27 E-2</td>
<td>9.35 E-1</td>
<td>8.53 E-2</td>
</tr>
<tr>
<td>SAT4.CIN</td>
<td>3.13 E-2</td>
<td>-4.41 E-3</td>
<td>5.34 E-2</td>
<td>1.98 E-2</td>
</tr>
<tr>
<td>SAT4.BST</td>
<td>2.96 E-2</td>
<td>-4.46 E-3</td>
<td>5.15 E-2</td>
<td>1.91 E-2</td>
</tr>
<tr>
<td>H30SKAE.BST</td>
<td>-1.48 E-1</td>
<td>-2.07 E-2</td>
<td>6.11 E-2</td>
<td>3.65 E-2</td>
</tr>
<tr>
<td>H30SKAE.CIN</td>
<td>-1.65 E-1</td>
<td>-1.71 E-2</td>
<td>7.93 E-2</td>
<td>2.52 E-2</td>
</tr>
<tr>
<td>H30SKAF.BST</td>
<td>-1.67 E-1</td>
<td>-6.38 E-3</td>
<td>1.80 E-1</td>
<td>1.55 E-2</td>
</tr>
<tr>
<td>H30SKAF.CIN</td>
<td>-1.56 E-1</td>
<td>-1.10 E-3</td>
<td>1.69 E-1</td>
<td>5.58 E-2</td>
</tr>
<tr>
<td>H30SKAG.BST</td>
<td>-3.42 E-1</td>
<td>2.53 E-2</td>
<td>4.28 E-2</td>
<td>2.84 E-2</td>
</tr>
<tr>
<td>H30SKAH.BST</td>
<td>-1.54 E-1</td>
<td>-9.67 E-3</td>
<td>1.39 E-1</td>
<td>3.86 E-2</td>
</tr>
<tr>
<td>H30SKAH.CIN</td>
<td>-1.63 E-1</td>
<td>-5.25 E-3</td>
<td>1.28 E-1</td>
<td>3.17 E-2</td>
</tr>
<tr>
<td>H30SKAI.BST</td>
<td>1.84 E-2</td>
<td>-3.93 E-3</td>
<td>5.54 E-2</td>
<td>1.56 E-2</td>
</tr>
<tr>
<td>H30SKAI.CIN</td>
<td>-1.14 E-1</td>
<td>5.88 E-3</td>
<td>1.37 E-1</td>
<td>5.14 E-2</td>
</tr>
<tr>
<td>HEL30AF.BST</td>
<td>-6.73 E-3</td>
<td>-6.38 E-3</td>
<td>8.26 E-2</td>
<td>1.55 E-2</td>
</tr>
<tr>
<td>HEL30AF.CIN</td>
<td>-6.77 E-2</td>
<td>-5.95 E-3</td>
<td>8.47 E-2</td>
<td>1.61 E-2</td>
</tr>
<tr>
<td>HEL30AG.BST</td>
<td>-6.87 E-2</td>
<td>-2.77 E-3</td>
<td>9.36 E-2</td>
<td>1.31 E-2</td>
</tr>
<tr>
<td>HEL30AG.CIN</td>
<td>-7.14 E-2</td>
<td>-2.58 E-3</td>
<td>9.32 E-2</td>
<td>1.32 E-2</td>
</tr>
<tr>
<td>HEL30AI.BST</td>
<td>-3.16 E-2</td>
<td>-1.62 E-3</td>
<td>5.11 E-2</td>
<td>1.23 E-2</td>
</tr>
<tr>
<td>HEL30AI.CIN</td>
<td>-3.35 E-2</td>
<td>2.25 E-3</td>
<td>4.99 E-2</td>
<td>1.24 E-2</td>
</tr>
<tr>
<td>HEL30AJ.BST</td>
<td>-3.73 E-2</td>
<td>-5.70 E-4</td>
<td>7.69 E-2</td>
<td>1.52 E-2</td>
</tr>
<tr>
<td>HEL30AJ.CIN</td>
<td>-4.20 E-2</td>
<td>4.65 E-4</td>
<td>6.45 E-2</td>
<td>1.08 E-2</td>
</tr>
<tr>
<td>PROFILE</td>
<td>MEAN OLD</td>
<td>MEAN NEW</td>
<td>STANDARD DEVIATION OLD</td>
<td>STANDARD DEVIATION NEW</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>----------</td>
<td>------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>HL246AD.BST</td>
<td>2.66 E-2</td>
<td>-2.26 E-3</td>
<td>1.76 E-2</td>
<td>5.19 E-3</td>
</tr>
<tr>
<td>HL246AD.CIN</td>
<td>2.35 E-2</td>
<td>-1.49 E-3</td>
<td>1.09 E-2</td>
<td>3.28 E-3</td>
</tr>
<tr>
<td>HL246AE.BST</td>
<td>2.32 E-2</td>
<td>-3.59 E-4</td>
<td>1.40 E-2</td>
<td>6.45 E-3</td>
</tr>
<tr>
<td>HL246AE.CIN</td>
<td>2.27 E-2</td>
<td>-1.71 E-4</td>
<td>1.18 E-2</td>
<td>5.99 E-3</td>
</tr>
<tr>
<td>HL346AD.BST</td>
<td>2.43 E-2</td>
<td>-3.04 E-3</td>
<td>1.55 E-2</td>
<td>7.50 E-3</td>
</tr>
<tr>
<td>HL346AD.CIN</td>
<td>2.24 E-2</td>
<td>-2.63 E-3</td>
<td>1.42 E-2</td>
<td>4.36 E-3</td>
</tr>
<tr>
<td>HL346AE.BST</td>
<td>2.66 E-2</td>
<td>4.61 E-4</td>
<td>1.86 E-2</td>
<td>7.63 E-2</td>
</tr>
<tr>
<td>HL346AE.CIN</td>
<td>2.59 E-2</td>
<td>9.35 E-4</td>
<td>1.72 E-2</td>
<td>6.96 E-3</td>
</tr>
<tr>
<td>HL446AC.BST</td>
<td>2.43 E-2</td>
<td>2.93 E-3</td>
<td>1.94 E-2</td>
<td>6.66 E-3</td>
</tr>
<tr>
<td>HL446AC.CIN</td>
<td>2.59 E-2</td>
<td>-3.43 E-3</td>
<td>1.13 E-2</td>
<td>4.07 E-3</td>
</tr>
<tr>
<td>HL446AD.BST</td>
<td>1.86 E-2</td>
<td>-1.06 E-4</td>
<td>1.23 E-2</td>
<td>7.96 E-3</td>
</tr>
<tr>
<td>HL446AD.CIN</td>
<td>1.85 E-2</td>
<td>-4.82 E-5</td>
<td>1.04 E-2</td>
<td>7.55 E-3</td>
</tr>
<tr>
<td>HL546AE.BST</td>
<td>3.15 E-2</td>
<td>-1.88 E-3</td>
<td>1.42 E-2</td>
<td>6.81 E-3</td>
</tr>
<tr>
<td>HL546AE.CIN</td>
<td>3.14 E-2</td>
<td>-1.76 E-3</td>
<td>1.23 E-2</td>
<td>6.09 E-3</td>
</tr>
<tr>
<td>HL546AC.TMR</td>
<td>4.31 E-2</td>
<td>-7.54 E-3</td>
<td>1.22 E-2</td>
<td>4.29 E-2</td>
</tr>
<tr>
<td>HL546AF.TMR</td>
<td>2.28 E-2</td>
<td>-7.91 E-5</td>
<td>1.16 E-2</td>
<td>5.39 E-3</td>
</tr>
<tr>
<td>HJ146AC.BST</td>
<td>2.02 E-2</td>
<td>-4.08 E-3</td>
<td>1.46 E-2</td>
<td>7.87 E-3</td>
</tr>
<tr>
<td>HJ146AC.CIN</td>
<td>2.10 E-2</td>
<td>-2.21 E-3</td>
<td>9.80 E-3</td>
<td>8.73 E-3</td>
</tr>
<tr>
<td>HJ146AD.BST</td>
<td>1.37 E-2</td>
<td>3.74 E-3</td>
<td>1.31 E-2</td>
<td>7.11 E-3</td>
</tr>
<tr>
<td>HJ146AD.CIN</td>
<td>1.35 E-2</td>
<td>5.29 E-4</td>
<td>1.25 E-2</td>
<td>6.94 E-3</td>
</tr>
</tbody>
</table>
FIGURE 3.6-5 A COMPARISON OF THE CINE PITCH ANGLE ACCELERATION PROFILE AND THE CINE PITCH RATE DIFFERENCE DATA PROFILE FOR H30SKAF
FIGURE 3.6-6 A COMPARISON OF THE CINE ROLL ANGLE ACCELERATION PROFILE AND THE CINE ROLL RATE DIFFERENCE DATA PROFILE FOR H30SKAF
performed with the widest angle tracking noise bandwidth. However, the HEL30 and H30SK series were performed at ranges of 2000 to 12000 offset in X and Y from the radar while SAT tests were over a range interval of 2500 feet to 1200 feet directly over the radar. The difference here is that slight wind disturbances in the SAT test configuration translate into reasonably large angle accelerations that produce momentary biases. These biases, in turn, produce large standard deviations in the difference data. This phenomenon will be examined in detail in the next subsection.

3.6.3.1 Acceleration Effects. In this case it turns out that acceleration is the primary source of error in the angle rate difference data once the scale factor of two has been removed. Figure 3.6-5 compares the pitch angle acceleration profile against the pitch rate difference data profile. Figure 3.6-6 gives a similar comparison for roll angle. Observe that the angle acceleration profile shape and the angle rate difference profile shape are highly correlated for both roll and pitch. Next, it will be demonstrated that not only are the shapes highly correlated, but that they are related by the expression given in equation 3-18 or 3-19.

Consider the interval 20 to 40 seconds in the pitch data of Figure 3.6-5. The average pitch angle acceleration during this period is -0.02 degrees/sec/sec. Using equation 3-19, the corresponding average pitch angle rate bias error is computed as 0.053 degrees/sec. This value agrees quite well with the pitch rate difference data in the same 20 to 40 second time interval.

Consider a similar calculation for the roll rate for the time interval 100 to 110 seconds. The average acceleration in this case is about -0.04 degrees/sec/sec and the computed roll rate bias error is 0.11 degrees/second. The average roll rate error taken for the same interval from the roll rate difference profile is about 0.09 degrees/second. Hence, the calculated data and the measured data agree reasonable well.

The conclusion from the above discussion is that the primary error source in the H30SKAF case is angle acceleration. Although the other flights must be evaluated on a case-by-case basis to determine the dominant
error source, one can draw an additional conclusion from the above data analysis. It was observed that very small angle accelerations, i.e. acceleration less than 0.04 deg/sec/sec produced angle rate biases of 0.11 deg/sec which is 40 times the specification on the standard deviation. Based on these numbers it is reasonable to conclude that the primary error source in the other tests will be angle acceleration as well. At shorter ranges, the same wind turbulence will cause larger angle rate errors and at longer ranges the reverse will be true.

Another important conclusion that can be drawn from this comparison is that the model shown in Figure 3.6-4 is an accurate representation of the angle and angle rate tracking loop. Furthermore, it shows that the actual bandwidths of these tracking loops (which is related to the natural frequency $f_n$ of the loop) matches the intended design bandwidth values. This is verified by the matching of the angle acceleration and the angle rate difference data through the relation 3-19 and the matching of the angle acceleration and the angle difference data through the equation 3-12. Both of these expressions contain $w_n$ which is the natural radian frequency of the loop.

Reflecting upon the comments above, it may well be that the problems with the angle rate tracker at close range during a space flight rendezvous are related to very slight angle accelerations of the target. A target acceleration of 0.01 deg/sec/sec causes an angle rate bias that is 10 times greater than the standard deviation specification. It is not known whether 0.01 deg/sec/sec angle acceleration is typically encountered in the shuttle-satellite rendezvous. However, it is recommended that radar data from some typical rendezvous be analyzed for acceleration bias problems. If this turns out to be the problem, it casts a new light on potential solutions to the angle rate tracking loop.

Before leaving this subsection, there is some additional evidence that lends additional support to the angle acceleration theory. The intent here is to demonstrate that the bias found in the pitch and roll angle difference data is consistent with the magnitude of angle acceleration given in the plots of Figures 3.6-5 and 3.6-6. Pitch angle difference data and roll
FIGURE 3.6-4 ANGLE AND ANGLE RATE ERROR DUE TO AN ACCELERATION OF 0.04 DEGREES/SEC
FIGURE 3.6-7  CINE ROLL AND PITCH ANGLE DIFFERENCE
DATA PROFILE FOR H30SKAF

3-120
FIGURE 3.6-8 A COMPARISON OF THE CINE KU PITCH RATE DIFFERENCE DATA AND THE CINE SIM PITCH RATE DIFFERENCE DATA FOR H30SKAF
3.6.3 SORTE Angle Rate Data Analysis

Since the scale factor problem was discovered near the end of the contract performance period, only limited analysis of the angle rate data could be done. This analysis consists of (1) recomputing all of the means and standard deviations of the roll and pitch angle rate data, excluding the November 4, 1985 flights due to the servo gain change, and (2) performing an in-depth analysis of a single flight (H30SKAF).

Table 3.6-2 compares the mean and standard deviation of the roll rate difference data, generated from the rescaled roll rate data, to the mean and standard deviation of the original roll rate difference data. Table 3.6-3 gives a similar comparison for the pitch rate data. Observe that both the means and the standard deviations of these difference data improve by at least a factor of two in every case. A comparison with the specification reveals that many of the mean values are within specification and most of the rest of the mean values are very close to the spec limit.

There are some general observations that can be made concerning the recomputed standard deviation values. Firstly, every value of standard deviation is still outside the specification limit and in only a few the values are just slightly outside the limit. This fact is still alarming. However, analysis of a sample case will demonstrate the source of error for many of these cases. Secondly, a comparison of the standard deviations for the various flight series is illuminating and encouraging. The performance of these flight series can order from best to worst as follows:

- **Best Performance:** HL- and HJ- series
- **Intermediate Performance:** H30SK- and HEL30- series
- **Worst Performance:** SAT series

This ordering is quite reasonable. The HL- and HJ- series should give the best performance for two reasons: (1) they were performed at long range with the narrowest angle tracking noise bandwidth and (2) at long range the angle accelerations are reduced. The HEL30- H30SK-, and SAT- series all were
angle difference data for H30SKAF are provided in Figure 3.6-7 for reference. Consider the pitch angle acceleration data time interval of 115 seconds to 125 seconds. In this interval the average acceleration is about 0.05 deg/sec/sec. Using equation 3-12, the angle bias error is computed as 0.088 degrees. If this is added to the pitch angle difference mean shown in Figure 3.6-7, then the total predicted angle error is -0.728 degrees. The pitch angle difference data of Figure 3.6-7 shows an average error of about -0.75 degrees for the same time interval. Hence, the pitch angle acceleration profile agrees with the pitch angle difference profile as well as the pitch angle rate difference profile.

Let's also do a calculation for the roll angle. Consider the time interval 120 to 125 seconds. The average roll angle acceleration is 0.05 deg/sec/sec and the calculated roll angle bias error is 0.088 degrees. Adding this to the mean error of the roll angle difference profile, the total computed average roll angle error for the time interval is 0.208 degrees. A review of the measured roll angle difference data for the same time period shows an average roll angle bias error of -0.22 to -0.225 degrees. Again the roll angle acceleration profile agrees with both the roll rate difference profile and the roll angle difference profile. Hence, the data seems consistent among the three variables for both roll and pitch.

**Simulation Verification.** As further proof that the scale factor of 2 should be removed and that acceleration is the major contributor to the angle rate errors, the H30SKAF CINE profile was injected into the final version of the simulation and angle rate and angle difference data was generated. Figure 3.6-8 compares the Ku-Band pitch angle rate difference data to the simulation pitch angle rate difference data (both sets are referenced to the CINE data), for the H30SKAF profile. Figure 3.6-9 gives a similar set for the roll angle rate difference data. These comparisons show that the simulation accurately reflects the angle rate response of the Ku-Band (at least for the present flight profile). It shows that the acceleration errors appear in the simulation response and are of the same magnitude. It also shows that there is no scale factor problem between the simulation and the modified measured angle rate data.
FIGURE 3.6-9 A COMPARISON OF THE CINE KU ROLL RATE DIFFERENCE DATA AND THE CINE SIM ROLL RATE DIFFERENCE DATA FOR H3OSKAF
FIGURE 3.6-10  A COMPARISON OF THE CINE KU-BAND PITCH
ANGLE DIFFERENCE DATA AND THE CINE SIM PITCH
ANGLE DIFFERENCE DATA FOR H3OSKAF

3-123
Figure 3.6-10 compares the CINE Ku pitch angle difference data and the CINE sim pitch angle difference data for the H30SKAF flight profile. A similar comparison for the roll angle difference data is provided in Figure 3.6-11. A first impression is that the data does not match as well as the angle rate comparison. However, it should be noted that the sim data is not quantized to 0.1 degrees as it is in the Ku-Band Radar. A quantized version of the simulation data would probably show a better fit.

3.6.4 A Discussion of the Servo Experiment

Once the scale factor problem in the angle rate was discovered in one of the sets of data, all of the data sets were scrutinized to determine whether the scale factor was the same for all cases. LEMSCO personnel discovered that the data from the 4 November 1985 flights had a scale factor of 0.5, rather than a factor of 2. All of these flight tests were flown with an increase by a factor of 4 in the $k_5$ gain of the angle tracking loop. Thus, it became clear that the output angle rate is scaled inversely with the change in the $k_5$ gain. What follows is a derivation of this fact.

Figure 3.6-12 gives an equivalent second order analog model representation of the angle rate tracking loop modified to include the $k_5$ gain (compare this configuration with the model in Figure 3.6-4).
FIGURE 3.6-11 A COMPARISON OF THE CINE KU-BAND ROLL ANGLE DIFFERENCE DATA AND THE CINE SIM PITCH ANGLE DIFFERENCE DATA FOR H3OSKAF

3-125
The transfer function for this loop is given by the expression

\[ \frac{\hat{\theta}(s)}{\theta(s)} = \frac{sw_n^2}{(s^2 + 2Kw_n s + Kw_n^2)} \]

where \( w_n \) has been defined previously and \( K \) is associated with the \( k_5 \) constant and represents the gain inserted into the Ku-Band Radar servo electronics during the SORTE program at WSMR. For the normal operational design of the radar \( K = 1 \), but in the SORTE experiments the gain \( K \) was raised to a value of 4.

It is now demonstrated that the angle rate output is divided down by a factor of \( K \). This fact is most easily demonstrated by choosing a particular input. The input chosen is a ramp in angle or a step in angle rate.

\[ \theta(t) = At \quad \text{(angle position)} \]

\[ \dot{\theta}(t) = A \quad \text{(angle rate)} \]

Where \( A \) is the value of the angle rate and should be the value output by the tracking loop. The \( s \)-domain representation of the ramp is

\[ \theta(s) = A/s^2 \]

and the \( s \)-domain response to this input is

\[ \frac{\hat{\theta}(s)}{\theta(s)} = (A/s^2)(s^2 + 2kw_n s + kw_n^2) \]

To determine the steady-state response to this input, apply the final value theorem from control theory. The final value theorem is

\[ \lim_{s \to 0} s \hat{\theta} = \hat{\theta}(s) \]

and the result is

\[ \hat{\theta} = A/K. \]
This shows that the output of the loop is the true angle rate divided by K. So when K is 1 as in the operational design, angle rate meters give a true indication of the target's ILOS angle rate. However, when K is different from 1 then the angle rate meters give a scaled version of the true angle rate.

The above result is consistent with the data from the SORTE program servo experiments (disregarding the scale factor of 2). The implication is, that to determine the true angle rate noise performance with the increased gain, one must scale the angle rate data by the gain K prior to determining the noise properties. When this is done with the present data it is found that the noise performance does not improve, but degrades, when increasing the $k_5$ gain.

If the problem with the angle rate loop in flight is too much random noise, then the noise bandwidth of the loop should be decreased. Since the noise bandwidth is proportional to $k_5$, the value of this gain should be decreased to reduce the noise bandwidth. However, if the problem with the angle rate performance in flight is related to biases induced by fluctuating angle acceleration, then the solution to the problem is much more difficult. The bias in angle rate due to acceleration can be shown to be proportional to $\tau$ (see Figure 3.6-12); therefore, to decrease the bias due to acceleration, $\tau$ must somehow be reduced. A change in the gain $k_5$ will not change the bias in angle rate.

The real solution to the angle acceleration problem is to use a third order loop (commonly called an alpha-beta-gamma filter). This type of loop will not suffer from angle rate bias in the presence of a constant angle acceleration.
4.0 PALAPA MISSION DATA ANALYSIS

The purpose of this section is to provide analysis of some space flight rendezvous data. The particular set of data supplied for this exercise was the radar data from the space shuttle rendezvous with the Palapa B Satellite during mission 51A in November 1984. The summary of the analysis is done on three different levels. Firstly, a general qualitative discussion is presented to point out all significant features in the data. Secondly, some limited quantitative analysis of the data is given to provide the reader with a feel for the radar errors encountered in an operational environment. Thirdly, the results of injecting the smoothed Palapa profile into the simulation are compared to the actual data. In addition, the validity of this simulation technique is discussed.

4.1 QUALITATIVE DISCUSSION OF THE DATA

Excluding the Radar Signal Strength (RSS), there are six basic target parameters that the Ku-Band Radar tracks during a rendezvous: range, range rate, roll and pitch angles and inertial line of sight (ILOS) roll rate and pitch rate. Figure 4.1-1 shows the range and range rate data for the entire rendezvous which was approximately 9000 seconds in duration. Figure 4.1-2 gives a similar plot for the roll and pitch angle data, and Figure 4.1-3 gives the data for ILOS roll and pitch rate. Some general qualitative observations about these data follow.

The range and range rate data of Figure 4.1-1 looks very well behaved (at least on the scale shown in the figure). It will be shown in the next section that the random component of these data are well within specification for these time intervals, corresponding to three different range tracker bandwidths. Also, observe that the glitches in the data in the intervals 0 to 400 seconds and 2500 to 3000 seconds are not caused by the radar, but instead are missing data due to data link drop-out or some other communication link problem.
FIGURE 4.1-1    KU-BAND RADAR RANGE AND RANGE RATE PROFILES
FOR THE RENDEZVOUS WITH THE PALAPA SATELLITE

4-2
FIGURE 4.1-2  KU-BAND RADAR PITCH AND ROLL ANGLE PROFILES FOR THE RENDEZVOUS WITH THE PALAPA SATELLITE
FIGURE 4.1-3 KU-BAND RADAR ILOS PITCH AND ROLL RATE PROFILES FOR THE RENDEZVOUS WITH THE PALAPA SATELLITE
Except for a few time intervals, the roll and pitch angle data is very near zero for the entire rendezvous. This just means that the shuttle and the target are coming together along the -Z axis of the Shuttle Body Coordinate System. The most prominent features in these data is the large angular change in the data over the time interval 2200 seconds to 3200 seconds. This corresponds to an intentional change in the Orbiter's attitude and preparation for what is known as a TI burn. This injects the Orbiter into the final phase of the rendezvous. Also notice that there is some nonzero angular positions in the time after 5500 seconds. During this time, the Orbiter is performing several small "hops" to move toward the target. In summary, the data is well-behaved and, as shown in the next section, the random component is well within specification for both the roll and pitch angle in all three bandwidth intervals.

The ILOS roll and pitch rate data of Figure 4.1-3 has some interesting features. First, the glitch in the data over the interval 2500 to 3000 seconds is caused by data link drop-out as in the range and range rate case. The hump in the roll rate data from 2000 to 3500 seconds is associated with the TI burn maneuver, but the mechanism producing it cannot be stated for certain. It could be caused by true target inertial angle rate or, it could be that the body rate during the maneuver was not compensated perfectly. Similar comments apply to the pitch rate data over this time interval.

The next significant feature that can be picked up from these data are the bandwidth switch points, especially in the roll rate data. These switch points are marked by a noticeable step increase in the "random" component of the data. The first switch point (which is the hardest to see on the scale of the data) occurs at range 23,030 feet and approximately 6000 seconds. The second switch point is quite prominent and occurs at a 11,510 feet and approximately 6500 seconds. For a long time, this increase in the random component was solely due to the increase in tracker noise bandwidth when the bandwidth is switched. However, based on the analysis of the SORTE test data, it is now felt that a significant part is due to very slight inertial angle accelerations and the angle rate biases induced by these accelerations. Also observe that these angle rates can also be produced by beam wander on the target, especially for ranges less than 1000 feet.
Another feature of the data is that the envelope of the random component in roll and pitch rate appears to grow from time 6500 seconds to 9500 seconds as the range decreases into 100 feet. Figure 4.1-4 gives an expanded view of this envelope for roll and pitch rates. This observation supports the statements of the preceding paragraph. If the fluctuation in the data were caused by thermal noise, then the random component would certainly not grow with decreasing range and increasing target signal strength. On the other hand, problems with actual inertial cross line-of-sight movement (producing angle acceleration) would increase with decreasing range, and problems with beam wander on the target would also increase with decreasing range. Neither of these problems can be controlled with adjustments in the angle rate tracking loop parameters.

Figure 4.1-5 gives an even more expanded view of the roll rate data for the time interval 8000 to 8100 seconds. A qualitative observation about this data is that it appears to have a less random or more deterministic character to it. It is more oscillatory in nature. (Spectral analysis of the data would verify this statement.)

The final observation concerns the pitch rate data. The significant bias seen in the data is due to the orbital rate. That is, the shuttle orbits the earth approximately every 90 minutes. This produces a rate of 0.067 degrees per second and corresponds perfectly to the pitch rate bias. This is reasonable since pitch is the angular movement in the plane of the orbit due to the attitude of the shuttle during the rendezvous.

4.2 SOME SIMPLE QUANTITATIVE DATA ANALYSIS

To perform an accurate quantitative analysis of the Ku-Band Radar requires accurate reference data. That is, data generated by an independent sensor or set of sensors whose measurement accuracies are as good or better than the Ku-Band Radar. The purpose of the SORTE program was to provide such a reference and, from this data, develop some quantitative estimates of radar performance. However, the SORTE program experiments could not exactly duplicate space flight conditions. Hence, a quantitative analysis of the Palapa flight data was undertaken using a psuedo-reference. The psuedo
FIGURE 4.1-4 EXPANDED VIEW OF ROLL AND PITCH RATE PROFILES FOR THE PALAPA RENDEZVOUS
FIGURE 4.1-5  EXPANDED VIEW OF ROLL RATE DATA FOR THE PALAPA RENDEZVOUS ILLUSTRATING THE FINE STRUCTURE OF THE DATA
reference generation and the dangers associated with it are discussed in the next subsection. Results of the data analysis are provided in the subsection following the reference discussion.

4.2.1 Reference Data Generation

Assumptions. Generation of the reference data set was accomplished by making the following assumptions. First, it is assumed that the average of the radar parameter estimates over short intervals (10-50 seconds) are bias-free and represent the target's true parameter average in that interval. There is one significant drawback in this assumption: prolonged range and angle acceleration produce significant biases in range, range rate, angle and angle rate. To alleviate this problem to some extent, an analysis of the parameter bias error was ignored in the present exercise.

The second assumption is that the fluctuations in the data over small intervals is due to radar thermal or quantization noise. Hence, these features were eliminated when forming the reference. The danger in doing this was not discovered until after the fact, during SORTE data analysis. These so-called random fluctuations, especially in the angle rates, may be induced by the shuttle - Palapa rendezvous dynamics. In the case of the angle rate, for example, significant short-term angle rate bias could be induced by slight angle accelerations due to flight control adjustments by the shuttle pilot. These short-term biases on a larger time scale appear to have a random nature and were removed for the data analysis reported below. It is now believed that the discrepancy between the simulation angle rate and the flight angle rate data is due to removing true fluctuations in the angle rate data.

Method. The basic method for developing a data reference was to smooth the radar flight data using a short-term averaging technique. The technique was moving window averaging and can be represented by the following expression:
The value of $N$ used for range, range rate, and roll and pitch angle was 13 samples (at 1 second per sample), and a value of 51 samples was used for ILOS roll and pitch rate. The larger value for the angle rate was to suppress the more severe fluctuations in that data. Figures 4.2-1 and 4.2-2 compare the smoothed and unsmoothed range and range rate data for a window of length 13. Figure 4.2-3 compares the smoothed and unsmoothed pitch angle data. The "steps" in the unsmoothed pitch angle data is due to 0.1 degree quantization of the roll and pitch angle data prior to transmission over the MDM to the shuttle general purpose computer (GPC). These "steps" are eliminated in the smoothed data, as they should be.

Figure 4.2-4 gives the smoothed and unsmoothed ILOS roll rate. In this case, a window of length 51 was used to heavily smooth the "noisy" angle rate. As discussed above, this was probably a mistake, since these fluctuations may have been induced by actual shuttle motion and/or beam wander. However, the validity of this statement cannot be established without a true reference.

4.2.2 Data Analysis Results

Table 4.2-1 summarizes the results of the Palapa rendezvous radar data analysis. This analysis computes the standard deviation of the random component only. Furthermore, the analysis is done for three distinct time intervals corresponding to the three different tracking bandwidths. (It is a fact that the range and angle trackers both have three different bandwidth values and that these values are switched at the same points in range. Also, the bandwidth values of both trackers increase with decreasing range intervals.)
FIGURE 4.2-1 SMOOTHED AND UNSMOOTHED KU MDM RANGE DATA. A 13 SAMPLE WINDOW WAS USED FOR SMOOTHING.
FIGURE 4.2-2 SMOOTHED AND UNSMOOTHED KU MDM RANGE RATE DATA. A 13 SAMPLE WINDOW WAS USED FOR SMOOTHING.
FIGURE 4.2-3 SMOOTHED AND UNSMOOTHED KU MDM PITCH ANGLE DATA. A 13 SAMPLE WINDOW WAS USED FOR SMOOTHING.
FIGURE 4.2-4 SMOOTHED AND UNSMOOTHED KU MDM ILOS ROLL RATE DATA. A 51 SAMPLE WINDOW WAS USED FOR SMOOTHING.
<table>
<thead>
<tr>
<th>TIME INTERVAL, SEC</th>
<th>4855 - 5890</th>
<th>5890 - 6530</th>
<th>6530 - 6993</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANGE INTERVAL, FT</td>
<td>43520 - 23040</td>
<td>23040 - 5760</td>
<td>5760 - 6530</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPECIFICATION</th>
<th>STD DEV</th>
<th>STD DEV</th>
<th>STD DEV</th>
<th>STD DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range, Ft</td>
<td>26.7 Ft*</td>
<td>20.45</td>
<td>10.97</td>
<td>5.3</td>
</tr>
<tr>
<td>Range Rate, Ft</td>
<td>0.333 Ft/Sec</td>
<td>0.119</td>
<td>0.088</td>
<td>0.076</td>
</tr>
<tr>
<td>Roll Angle, Deg</td>
<td>0.153 deg</td>
<td>0.037</td>
<td>0.026</td>
<td>0.031</td>
</tr>
<tr>
<td>Pitch Angle, Deg</td>
<td>0.153 deg</td>
<td>0.034</td>
<td>0.056</td>
<td>0.052</td>
</tr>
<tr>
<td>ILOS Roll Rate, Deg/Sec</td>
<td>2.7 E-3 deg/sec</td>
<td>8.9 E-4</td>
<td>2.9 E-3</td>
<td>4.7 E-3</td>
</tr>
<tr>
<td>ILOS Pitch Rate, Deg/Sec</td>
<td>2.7 E-3 deg/sec</td>
<td>1.4 E-3</td>
<td>4.4 E-3</td>
<td>6.8 E-3</td>
</tr>
</tbody>
</table>

*The three sigma range specification is 1 percent of range for ranges greater than 8000 feet.

The data of Table 4.2-1 shows that range, range rate and angle are well within their respective specifications for all three range intervals. On the other hand, the angle rate data is within specification for the narrowest bandwidth, but is out of specification for the other bandwidths. Please observe that these data can neither be considered as best-case or worst-case random component analysis. Short-term range accelerations will induce short-term bias in the range and range rate that have been removed with the present smoothing technique. Now, these short-term biases can add to the standard deviation of the random component. However, a calculation of range bias generated by typical acceleration shows that this problem does not add significantly to the range data. Hence, the range data analysis of Table 4.2-1 is an accurate reflection of the radar range performance in flight.
On the other hand, if there is appreciable change in range acceleration, e.g., greater than 10 feet/sec/sec, the bias profile of the range rate data will be affected significantly. A changing range acceleration over a given bandwidth interval produces a changing range rate bias over the corresponding interval (as shown in the SORTE data). This changing bias could add significantly to the random component, putting it out of specification. However, in the range interval of most importance, e.g., ranges less than 5 nautical miles, the deceleration is of very small magnitude. Hence, the range rate data analysis of Table 4.2-1 is an accurate reflection of radar performance under space flight rendezvous conditions.

Angle acceleration will also induce bias in the angle and angle rate data. Hence, a varying bias due to a varying angle acceleration could induce addition error in the random component. Calculation of this error in Section 3.5 for the angle tracker shows that the bias, under heavy angle acceleration, does not influence the random component significantly. Thus, the angle data analysis of Table 4.2-1 gives representative performance in a space operations environment. Observe that the angle data standard deviation is better than specification by a factor of 5.

At close range, it is hard to decide whether the fluctuations seen in the radar angle rate data are caused by true target shuttle motion, or beam wander, or by radar noise. If the randomness is based on radar noise, then the data of Table 4.2-1 is representative of radar performance and is out of specification. If the fluctuations in the data are non-noise related, as the SORTE data indicates, then the data of Table 4.2-1 is a worst-case result, and it may be that the angle rate is really within specification once the proper reference is applied.

4.3 SIMULATION RESULTS

4.3.1 Reference Generation

To generate simulation data for the Palapa Satellite rendezvous, a reference flight trajectory had to be developed. This development can be described as follows.
The required inputs to the simulation are the target's position and velocity vectors in shuttle body coordinates and the shuttle angular velocity vector, $\mathbf{W}_B$, in shuttle body coordinates. The target's position and velocity vectors can be obtained from the smoothed range, range rate, and roll angle and pitch angle data described in Section 4.2.1. To obtain the shuttle angular velocity vector requires some additional thought.

The radar data can provide us with two of the three components of the shuttle body angular velocity vector components in body coordinates. These are the $X$-component and the $Y$-component. The $Z$-component representing vehicle cannot be obtained from the data and is assumed zero. The $X$-component is determined by computing the roll rate from first differences of the roll angle data and subtracting the smooth ILOS roll rate value. The $Y$-component is determined in a similar fashion using the smoothed pitch angle and smoothed ILOS pitch rate information. Mathematically, this can be expressed as

\[
\text{(4-2)} \quad \begin{align*}
\text{EWB}_1(N) & = \frac{(\text{SRANG}(N) - \text{SRANG}(N-1))}{\Delta T} - \text{SRRTE}(N) \\
\text{EWB}_2(N) & = \frac{(\text{SPANG}(N) - \text{SPANG}(N-1))}{\Delta T} - \text{SPRTE}(N)
\end{align*}
\]

where $\text{EWB}_1, \text{EWB}_2$ = $X$- and $Y$- component of the shuttle body angular velocity vector

$\text{SRANG}(N)$ = Nth value of smoothed roll angle

$\text{SPANG}(N)$ = Nth value of smoothed pitch angle

$\text{SRRTE}(N)$ = Nth value of smoothed roll rate

$\text{SPRTE}(N)$ = Nth value of smoothed pitch rate

$\Delta T$ = Sampling period (1 second)

4.3.2 Simulation Performance Against Palapa Reference

Table 4.3-1 summarizes the results of injecting the Palapa reference data into the Ku-Band Radar simulation program and computing statistics over the same range intervals as in the flight data analysis of Section 4.2. The simulation outputs were differenced with their corresponding reference data, and the mean and standard deviation were computed. Comparing this data against the specification yields the following observations. The
range standard deviations are within specification, while the mean is not. However, the reason the mean is not in specification is due to artificially setting a bias in the program code. This bias value should probably be changed. The range rate data is within the mean and standard deviation specification for all cases. The same is true for roll angle and pitch angle. However, the ILOS roll and pitch rate standard deviations are slightly out-of-specification in all three range intervals. The reason for this will be drawn into focus in the next section where the simulation and flight difference data statistics are compared.

### TABLE 4.3-1 PERFORMANCE OF THE KU BAND RADAR SIMULATION MODEL USING THE PALAPA SATELLITE RENDEZVOUS OF MISSION 51A AS THE INPUT TRAJECTORY

<table>
<thead>
<tr>
<th>TIME INTERVAL, SEC</th>
<th>4855 - 5890</th>
<th>5890 - 6530</th>
<th>6530 - 6993</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANGE INTERVAL, FT</td>
<td>43520 - 23040</td>
<td>23040 - 11520</td>
<td>11520 - 5760</td>
</tr>
<tr>
<td>MEAN</td>
<td>STD DEV</td>
<td>MEAN</td>
<td>STD DEV</td>
</tr>
<tr>
<td>Range, Ft</td>
<td>99.2</td>
<td>8.57</td>
<td>99.2</td>
</tr>
<tr>
<td>Range Rate, Ft/Sec</td>
<td>-0.04</td>
<td>0.06</td>
<td>0.0</td>
</tr>
<tr>
<td>Roll Angle, Deg</td>
<td>0.015</td>
<td>0.044</td>
<td>0.029</td>
</tr>
<tr>
<td>Pitch Angle, Deg</td>
<td>0.066</td>
<td>0.036</td>
<td>0.064</td>
</tr>
<tr>
<td>ILOS Roll Rate, Deg/Sec</td>
<td>3.59 E-4</td>
<td>1.02 E-4</td>
<td>3.11 E-4</td>
</tr>
<tr>
<td>ILOS Pitch Rate, Deg/Sec</td>
<td>-1.22 E-3</td>
<td>4.24 E-3</td>
<td>-1.01 E-3</td>
</tr>
</tbody>
</table>
4.3.3 **Comparison with Flight Data Performance**

The purpose of this subsection is to compare the flight difference data to the simulation difference data. First, the statistics of these two data sets are compared. The range standard deviations compare quite well. The simulation seems to give more optimistic estimates here. However, the simulation shows a decreasing trend in sigma as the absolute range decreases, just as the flight data does. The simulation range rate standard deviation compares well with the corresponding flight data. Again, the simulation shows more optimistic performance than the flight data and the comparison becomes closer at close range. The differences in the range rate performance are not serious enough to question the fidelity of the simulation in this area. The roll and pitch angle standard deviations for the flight and simulation data are both excellent for all three range intervals.

A comparison of the angle rate data statistics shows some inconsistency from range interval-to-range interval. In both roll and pitch rate, the flight data showed the random component progressively getting worse as the range decreases. The simulation data on the other hand seems to fluctuate as the range decreases. This seems confusing! Let's try to make some sense of it by considering the closest range interval. In this case, roll rate flight data is 2.5 times worse than the simulation, and the pitch rate flight data is 3 times worse than the simulation data. As discussed earlier, it is felt that the source of this error was use of the wrong reference for the flight data analysis. That is, the reference was wrong because the apparent randomness in the angle rate was removed with heavy smoothing to form the reference. Based on the analysis of the SORTE angle rate data, it is now felt this "randomness" is, in fact, part of the rendezvous dynamics or, at very close range (less than 2000 feet), beam wander on the target. Another fact that heavily supports this conclusion, is that a comparison of the SORTE flight data and corresponding simulation data showed excellent agreement (see Figures 3.6-8 and 3.6-9). In this case there was a very accurate reference to inject into the simulation. It is recommended that significantly less smoothing be used in the generation of the angle rate data reference.
Another method of analysis is to compare the difference data profiles of the flight and simulation data. Figures 4.3-1 and 4.3-2 make this comparison for range rate and roll rate in the time interval 6500 to 7000 feet (or range interval 11500 to 5700 feet). The reason for the discontinuous jump of 0.12 feet/sec in the simulation range rate data is not known at the present. Otherwise, the data confirms the discussion given above.
FIGURE 4.3-1 A COMPARISON OF THE KU-BAND RADAR AND THE SIMULATION RANGE RATE DIFFERENCE DATA FOR THE PALAPA SATELLITE RENDEZVOUS
FIGURE 4.3-2 A COMPARISON OF THE KU-BAND RADAR AND THE SIMULATION ILOS ROLL RATE DIFFERENCE DATA FOR THE PALAPA SATELLITE RENDEZVOUS
5.0 REFERENCES

1. Shuttle Orbiter Radar Test and Evaluation, Job Order No. 16-659, LEMSCO Inc., Houston, TX, April 1986


APPENDIX A

SOURCE CODE LISTING OF BASELINE PROGRAM

This appendix is a listing of the baseline program which was obtained from JSC at the beginning of the contract. The program is available on the Building 44 VAX and resides in the KUBAND.HOWARD directory. The name of the source program is HACSIM.
COMMON /TARGET/ITARG, SRCS
COMMON /ACTDAT/R, ARDOT, SPANG, SRPTE, SRRTE, AL, BT, SALF, SBTA,
1ER(3), ERT(3), AZRATE, ELRATE, AZRTE, ELRTE,
COMMON /TERM/TERM
COMMON /OUTPUT/MSWF, MTF, MSF, SSRNG, SSROD, SSSPANG, SSRRANG, SSSPRTE,
2 SSRRTE, SSRSS, MADVF, MRDVF, MARDVF, MRRDVF
3 , SSALP, SSBET
COMMON /SYSDAT/TS, DUM2(14)

TEST DATA FROM WS32TDATA1

CHARACTER*9 FPRO(18)
CHARACTER*32 IXT, IYT(22), LPRO(18)
DATA IXT/'TIME SECONDS$'/
DATA IYT(1)/'RANGE FEET$'/
DATA IYT(2)/'RANGE RATE FT/SEC$'/
DATA IYT(3)/'ROLL ANGLE DEG$'/
DATA IYT(4)/'PITCH ANGLE DEG$'/
DATA IYT(5)/'ROLL RATE DEG/SEC$'/
DATA IYT(6)/'PITCH RATE DEG/SEC$'/
DATA IYT(7)/' ALPHA DEG$'/
DATA IYT(8)/' BETA DEG$'/
DATA IYT(9)/'AZ RATE DEG/SEC$'/
DATA IYT(10)/'EL RATE DEG/SEC$'/
DATA IYT(11)/' X (NORTH) FEET$'/
DATA IYT(12)/' Y (EAST) FEET$'/
DATA IYT(13)/' Z (ALTITUDE) FEET$'/
DATA IYT(14)/' ELEVATION ANGLE DEG$'/
DATA IYT(15)/'DELTA RANGE FEET$'/
DATA IYT(16)/'DELTA RANGE RATE FT/SEC$'/
DATA IYT(17)/'DELTA ROLL ANGLE DEG$'/
DATA IYT(18)/'DELTA PITCH ANGLE DEG$'/
DATA IYT(19)/'DELTA ROLL RATE DEG/SEC$'/
DATA IYT(20)/'DELTA PITCH RATE DEG/SEC$'/
DATA IYT(21)/'DELTA ALPHA DEG$'/
DATA IYT(22)/'DELTA BETA DEG$'/

DATA LPRO(1)/' SIMULATION PROFILE HJ1465$'/
DATA LPRO(2)/' SIMULATION PROFILE HL1465$'/
DATA LPRO(3)/' SIMULATION PROFILE HL2465$'/
DATA LPRO(4)/' SIMULATION PROFILE HL3465$'/
DATA LPRO(5)/' SIMULATION PROFILE HL4465$'/
DATA LPRO(6)/' SIMULATION PROFILE HL5465$'/
DATA LPRO(7)/' SIMULATION PROFILE BJ1465$'/
DATA LPRO(8)/' SIMULATION PROFILE BL1465$'/
DATA LPRO(9)/' SIMULATION PROFILE BL2465$'/
DATA LPRO(10)/' SIMULATION PROFILE BL3465$'/
DATA LPRO(11)/' SIMULATION PROFILE BL4465$'/
DATA LPRO(12)/' SIMULATION PROFILE BL5465$'/
DATA LPRO(13)/' SIMULATION PROFILE CBP485$'/
DATA LPRO(14)/' SIMULATION PROFILE CBM485$'/
DATA LPRO(15)/' SIMULATION PROFILE CBP385$'/
DATA LPRO(16)/' SIMULATION PROFILE CBM385$'/

ORIGINAL PAGE IS OF POOR QUALITY
DATA LPRO(17)/' SIMULATION PROFILE CLP16$/'
DATA LPRO(18)/' SIMULATION PROFILE CLM16$/'
DIMENSION RID(12)
DATA FPRO(1)/'HJ146.JSC'/
DATA FPRO(2)/'HL146.BIN'/
DATA FPRO(3)/'HL246.BIN'/
DATA FPRO(4)/'HL346.BIN'/
DATA FPRO(5)/'HL446.BIN'/
DATA FPRO(6)/'BL446.BIN'/
DATA FPRO(7)/'BJ146.BIN'/
DATA FPRO(8)/'BL146.BIN'/
DATA FPRO(9)/'BL246.BIN'/
DATA FPRO(10)/'BL346.BIN'/
DATA FPRO(11)/'BL446.BIN'/
DATA FPRO(12)/'BL546.BIN'/
DATA FPRO(13)/'C6P48.BIN'/
DATA FPRO(14)/'C6M48.BIN'/
DATA FPRO(15)/'C6P30.BIN'/
DATA FPRO(16)/'C6M30.BIN'/
DATA FPRO(17)/'CLP16.BIN'/
DATA FPRO(18)/'CLM16.BIN'/
CHARACTER,9 UNIT7
BYTE IC(12)
COMMON /TMR/X,Y,Z,VX,VY,VZ,
1 DLP(3),DEL(3),DUE(3),
2 DSU(3),THAZL1,THEL1,THAZU1
COMMON /INPUT/RO(3),VO(3),ENB(3)
DIMENSION TP(2401),D(2401,22)
C
WRITE (6,*)'1 : TEK'
WRITE (6,*)'2 : VT125'
WRITE (6,*)'3 : VT240'
WRITE (6,*)'4 : PC'
READ (5,*)ITERM
WRITE(6,*)'PROFILE NUMBER PROFILE'
DO L=1,18
WRITE(6,200) L,LPRO(L)
WRITE(6,200) L,LPRO(L)
ENDDO
WRITE(6,*)'INPUT PROFILE NUMBER'
READ(5,*)ITAPE
WRITE(6,*)'ENTER NAME OF BINARY INPUT FILE'
READ(5,1001)UNIT7
COMMON /INPUT/RO(3),VO(3),ENB(3)
COMMON /INPUT/RO(3),VO(3),ENB(3)
FORMAT(7X,12,9X,A32)
FORMAT(7X,12,9X,A32)
WRITE(6,200) L,LPRO(L)
WRITE(6,200) L,LPRO(L)
WRITE(6,*)'INPUT PROFILE NUMBER'
WRITE(6,*)'INPUT PROFILE NUMBER'
READ(5,*)ITAPE
READ(5,*)ITAPE
WRITE(6,*)'ENTER NAME OF BINARY INPUT FILE'
WRITE(6,*)'ENTER NAME OF BINARY INPUT FILE'
READ(5,1001)UNIT7
READ(5,1001)UNIT7
C FORMAT(A24)
FORMAT(A24)
UNIT7=FPRO(ITAPE)
UNIT7=FPRO(ITAPE)
OPEN(UNIT=4,FORM='UNFORMATTED',STATUS='OLD',
OPEN(UNIT=4,FORM='UNFORMATTED',STATUS='OLD',
FILE=UNIT7)
FILE=UNIT7)
C
C FORMAT(5BA2)
WRITE(6,150)(IC(I),I=1,30)
WRITE(6,150)(IC(I),I=1,30)
IFTRK=0
IFTRK=0
WRITE(6,*)'INPUT 1 IF YOU WANT TO FILTER USING TRACK FLAG'
WRITE(6,*)'INPUT 1 IF YOU WANT TO FILTER USING TRACK FLAG'
READ(5,*)IFTRK
READ(5,*)IFTRK
WRITE(6,*)'INPUT RSC IN SQUARE METERS'
WRITE(6,*)'INPUT RSC IN SQUARE METERS'
READ (5,*)RCSM
READ (5,*)RCSM
SRCS=RCSM*3.28*3.28
SRCS=RCSM*3.28*3.28
SRCS=SORT(SRCS)
SRCS=SORT(SRCS)
WRITE(6,*)'SRCS=',SRCS
WRITE(6,*)'TOUT=',TOUT
WRITE(6,*)'TOUT=',TOUT
THAZL1=30.0;
THAZL1=30.0;
THEL1=30.0;
THEL1=30.0;
THAZU1=0.
THAZU1=0.
DLP(1)=0.2347
DLP(1)=0.2347
DLP(2)=0.85
DLP(3)=9.748
DEL(1)=0.192738
DEL(2)=0.855573
DEL(3)=3.299135
DUE(1)=0.88
DUE(2)=0.55
DUE(3)=0.39988
DDE(1)=5.46
DUE(2)=5.0
DUE(3)=3.0
DUE(4)=0.88
DUE(5)=0.55
DUE(6)=0.39988
DUE(7)=5.46
DUE(8)=5.0
DUE(9)=3.0

WRITE(6,'(A,E12.4)')' INPUT 1 FOR SCREEN OUTPUT'
READ(5,'(F8.4)')OUT
J=1
READ(4,'(F8.4)')T,X,Y,Z,VX,VY,VZ
T=I1-T
WRITE(6,'(A,E12.4)')' TS=',TS
CONTINUE
READ(4,'(F8.4)')T,X,Y,Z,VX,VY,VZ
C DATA IN METERS
CALL TMR2KU
IF(TOUT.EQ.1)THEN
WRITE(6,'(F8.4)')T,SSRNG,SSRRDOT,SSPANG,SRANG,SSPRTE,SRRT,E,SAF,BTA,
AZRATE,ELRATE,AZRTE,ELRTE
ENDIF
CALL EXEC
IF(IFTRK.EQ.1.AND.MTF.EQ.0)GO TO 1
J=J+1
IF(J.EQ.2001)GO TO 99
TP(J)=T
D(J,1)=SSRNG
D(J,2)=SSRRDOT
D(J,3)=SSPANG
D(J,4)=SRANG
D(J,5)=SSPRTE
D(J,6)=SRRT,E
D(J,7)=SSALP
D(J,8)=SSBET
D(J,9)=AzRATE
D(J,10)=ELRATE
D(J,11)=X
D(J,12)=Y
D(J,13)=Z
D(J,14)=ATAND(-Z/(X*X+Y*Y))
D(J,15)=SSRNG-R
D(J,16)=SSRRDOT-ARDOT
D(J,17)=SSPANG-IRANG
D(J,18)=SSPRTE-SPANG
D(J,19)=SSRT,E-SRTE
D(J,20)=SSPRTE-SPRTE
D(J,21)=SSALP-SALF
D(J,22)=SSBET-SBTA
GO TO 1

99 CONTINUE
IXD=0
94 CONTINUE
WRITE(6,'(A,E12.4)')' RCS IN METERS=',RCSM
WRITE(6,'(A,E12.4)')' PARA AXES TITLE'
DO I=1,22
WRITE(6,'(F8.4)')I,YT(I)
68 FORMAT(1X,14,10X,A32)
ENDDO
WRITE(6,*)'INPUT IXD, IYD IXD=0 FOR TIME'
READ(5,*)IXD, IYD
CALL SORT(TP,D,J, ITAPE, IXD, IYD)
GO TO 94
END

SUBROUTINE SORT(T,D,J, ITAPE, IXD, IYD)
CHARACTER*32 IXT,IYT(22),LPRO(18)
DIMENSION ITILT(8),IXL(22),IYL(8)
DATA IXT/'TIME SECONDS$'/
DATA IYT(1)/'RANGE FEET$/'
DATA IYT(2)/'RANGE RATE FT/SEC$/'
DATA IYT(3)/'ROLL ANGLE DEG$/'
DATA IYT(4)/'PITCH ANGLE DEG$/'
DATA IYT(5)/'ROLL RATE DEG/SEC$/'
DATA IYT(6)/'PITCH RATE DEG/SEC$/'
DATA IYT(7)/'ALPHA DEG$/'
DATA IYT(8)/'BETA DEG$/'
DATA IYT(9)/'AZ RATE DEG/SEC$/'
DATA IYT(10)/'EL RATE DEG/SEC$/'
DATA IYT(11)/'X (NORTH) FEET$/'
DATA IYT(12)/'Y (EAST) FEET$/'
DATA IYT(13)/'-Z (ALTITUDE) FEET$/'
DATA IYT(14)/'ELEVATION ANGLE DEG$/'
DATA IYT(15)/'DELTA RANGE FEET$/'
DATA IYT(16)/'DELTA RANGE RATE FT/SEC$/'
DATA IYT(17)/'DELTA ROLL ANGLE DEG$/'
DATA IYT(18)/'DELTA PITCH ANGLE DEG$/'
DATA IYT(19)/'DELTA ROLL RATE DEG/SEC$/'
DATA IYT(20)/'DELTA PITCH RATE DEG/SEC$/'
DATA IYT(21)/'DELTA ALPHA DEG$/'
DATA IYT(22)/'DELTA BETA DEG$/'
DATA LPRO(1)/'SIMULATION PROFILE HJ1465$'/
DATA LPRO(2)/'SIMULATION PROFILE HL1465$'/
DATA LPRO(3)/'SIMULATION PROFILE HL2465$'/
DATA LPRO(4)/'SIMULATION PROFILE HL3465$'/
DATA LPRO(5)/'SIMULATION PROFILE HL4465$'/
DATA LPRO(6)/'SIMULATION PROFILE HL5465$'/
DATA LPRO(7)/'SIMULATION PROFILE BJ1465$'/
DATA LPRO(8)/'SIMULATION PROFILE BL1465$'/
DATA LPRO(9)/'SIMULATION PROFILE BL2465$'/
DATA LPRO(10)/'SIMULATION PROFILE BL3465$'/
DATA LPRO(11)/'SIMULATION PROFILE BL4465$'/
DATA LPRO(12)/'SIMULATION PROFILE BL5465$'/
DATA LPRO(13)/'SIMULATION PROFILE CSP485$'/
DATA LPRO(14)/'SIMULATION PROFILE CSM485$'/
DATA LPRO(15)/'SIMULATION PROFILE CSP385$'/
DATA LPRO(16)/'SIMULATION PROFILE CSM385$'/
DATA LPRO(17)/'SIMULATION PROFILE CLP165$'/
DATA LPRO(18)/'SIMULATION PROFILE CLM165$'/

JPRO=ITAPE
CALL FIXIT(ITILT,LPRO(JPRO))
IF(IXD.EQ.0)THEN
DO I=1,J
X(I)=T(I)
Y(I)=0(I,IXD)
ENDDO
CALL FIXIT(IYL,IXT)
CALL FIXIT(IYL,ITY(D))
ELSE
DO I=1,J
X(I)=D(I,IXD)
Y(I)=0(I,IXD)
ENDDO
CALL FIXIT(IYL, IXT(IYD))
CALL FIXIT(IYL, IYT(IYD))
ENDIF
CALL PLOTIT(ITILT, IXL, IYL, X, Y, J)
RETURN
END
SUBROUTINE FIXIT(IOUT, IN)
DIMENSION IOUT(8)
CHARACTER*4 ITEM(8)
CHARACTER*32 ITEMP
ITEMP(I) = IN(I:14)
ITEMP(2) = IN(5:8)
ITEMP(3) = IN(9:12)
ITEMP(4) = IN(13:16)
ITEMP(5) = IN(17:20)
ITEMP(6) = IN(21:24)
ITEMP(7) = IN(25:28)
ITEMP(8) = IN(29:32)
ENCODE(32,999, IOUT)(ITEMP(I), I=1, 8)
FORMAT(8A4)
RETURN
END
SUBROUTINE PLOTIT(ITILT, IXL, IYL, X, Y, J)
COMMON /TERM/I
TERM
DIMENSION ITILT(B, IXL(B), IYL(8))
DIMENSION X(11), Y(11)
BYTE CR(2)
COMMON/TMR/A, B, C, D, E, F, G, (3), AH(3), AI(3), AJ(3), THAZL1, THEL1, THAZU1
CR(1) = 27
CR(2) = 12
XMAX = X(1)
XMIN = X(1)
YMAX = Y(1)
YMIN = Y(1)
DO I=1, J
IF(X(I).GT.XMAX) XMAX = X(I)
IF(X(I).LT.XMIN) XMIN = X(I)
IF(Y(I).GT.YMAX) YMAX = Y(I)
IF(Y(I).LT.YMIN) YMIN = Y(I)
END DO
IF(XMAX.EQ.XMIN) XMAX = XMIN + 1.1
IF(YMAX.EQ.YMIN) YMAX = YMIN + 1.1
IF (ITERM.EQ.1) CALL TEKALL(4114, 4Be, e, 1, e)
IF (ITERM.EQ.2) CALL REGIS (1, e)
IF (ITERM.EQ.31 CALL PVT24e
CALL BGNPL(-1)
CALL FLATBD
CALL PAGE(14., 18.)
CALL HEIGHT(.3)
CALL TITLE(ITILT, 100, IXL, 100, IYL, 100, 9.0, 13.5)
CALL MESSAG('LOWER AZIMUTH=', I100, 1.7, 13.)
CALL REALNO(THAZL1, 2, 'ABUT', 'ABUT')
CALL MESSAG('UPPER AZIMUTH=', I100, 1.7, 12.5)
CALL REALNO(THAU1, 2, 'ABUT', 'ABUT')
CALL MESSAG('ELEVATION=', I110, 1.7, 12.)
CALL REALNO(THEL1, 2, 'ABUT', 'ABUT')
CALL BLK1(1.5, 7.5, 11.9, 13.5, 4)
CALL HEADIN(ITILT, -100, -8.4)
CALL HEADIN('LOWER AZIMUTH=', 100, 4.4)
CALL HEADIN('UPPER AZIMUTH=', 100, 4.4)
CALL HEADIN('ELEVATION=', 100, 4.4)
C CALL REALNO(THEL1,2,'ABUT','ABUT')
CALL YAXANG(0.)
CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
CALL CURVE(X,Y,J.e)

C KK=J/3e
C K=d_
C DO I=1,KK
C K-3EH-K
C CALL RLINT(K,X(K),Y(K))
C ENDDO

CALL GRID(1,1)
CALL HEIGHT(.1)
CALL RESET('HEIGHT')
888 FORMAT( '+'. 2A1
227 FORMAT(192 ))
CALL DONEPL

C MICKEY MOUSE FIX
IF (IMM.EQ.0) THEN
REWIND (5)
READ(5,192) IC
192 FORMAT(192 )
WRITE(6,888) CR
ENDIF
RETURN
END

SUBROUTINE TMR2KU
C MODED JWG 2/8/85
C ,,, INPUT VIA COMMON VIA X,Y,Z,VX,VY,VZ
C -, OUTPUT VIA COMMON /ACTDAT/
C *** WHITE SANDS TO KU-BAND RADAR PARAMETER CONVERSION ***

****** COMMENTARY ******

** PURPOSE **
C THIS SOFTWARE TAKES THE POSITION AND VELOCITY OF A TARGET REFERENCED
C TO THE PEARL SITE SURVEY CAP AND CALCULATES THE VALUES OF THE KU-BAND
C RADAR PARAMETERS AS SEEN AT THE KU-BAND RADAR GIMBAL AXES INTERSECTION.
C THESE CALCULATIONS INVOLVE COORDINATE ROTATIONS THROUGH A THREE-AXIS
C POSITIONER AND FOUR TRANSLATIONS FROM THE PEARL CAP TO THE RADAR GIMBAL
C AXES INTERSECTION.
C THESE CALCULATIONS ARE TO BE DONE BY WSMR DATA REDUCTION USING THE WSMR
C RANGE REFERENCE ESTIMATIONS OF TARGET LOCATION WITH TIME. COMPARISON
C CAN BE MADE DIRECTLY WITH THE KU-BAND OUTPUTS FOR THE SAME TIME VALUES.

** INPUTS & CONSTANTS **
C WSMR PROVIDED INPUTS:
C WSMR WILL PROVIDE TARGET POSITION - X, Y, Z - AND VELOCITY - VX, VY,
C VZ AS INPUTS TO THIS PROGRAM.
C UNITS ARE FEET AND FEET/SECOND.
C THE COORDINATE SYSTEM IS:
C ORIGIN = PEARL SURVEY CAP
C X-AXIS IS POSITIVE TOWARD THE NORTH
C Y-AXIS IS POSITIVE TOWARD THE EAST
C NEGATIVE Z-AXIS IS UPWARD ALONG THE LOCAL VERTICAL.
C
C

A-7
FOR ANY GIVEN TEST THE FOLLOWING PARAMETERS WILL BE DEFINED ON THE
SIMULATION MAGNETIC DATA TAPE AND WILL REMAIN CONSTANT FOR THAT TEST:

DSU(I) = 1,3 IS THE LOCATION OF THE KU-BAND RADAR GIMBAL AXES IN
UPPER AZIMUTH COORDINATES.
THAZL1 IS THE LOWER AZIMUTH AXIS ROTATION ANGLE IN DEGREES.
THEL1 IS THE ELEVATION AXIS ROTATION ANGLE IN DEGREES.
THAZU1 IS THE UPPER AZIMUTH AXIS ROTATION ANGLE IN DEGREES.

ONE TIME INPUT CONSTANTS:

THE FOLLOWING PARAMETERS WILL BE MEASURED AFTER INSTALLATION OF THE
ANTENNA PEDESTAL AT THE PEARL SITE. THEIR VALUES SHOULD NOT CHANGE.
THEY ARE CURRENTLY DEFINED AS ZERO IN THIS SOFTWARE.

DLP(I) = 1,3 LOCATION OF THE LOWER AZIMUTH ORIGIN IN PEARL
COORDINATES.
DEL(I) = 1,3 LOCATION OF THE ELEVATION ORIGIN IN LOWER AZIMUTH
COORDINATES.
DUE(I) = 1,3 LOCATION OF THE UPPER AZIMUTH ORIGIN IN ELEVATION
COORDINATES.

** SOFTWARE OUTPUTS **

THIS SOFTWARE PRODUCES THE FOLLOWING OUTPUTS REFERENCED TO THE
RADAR GIMBAL AXES INTERSECTION.

R = RANGE (FT)
ARDOT = RANGE RATE (FT/SEC)
SRANG = ROLL ANGLE (DEG)
SPANG = PITCH ANGLE (DEG)
SRRTE = INERTIAL ROLL RATE (DEG/SEC)
SPRTE = INERTIAL PITCH RATE (DEG/SEC)
SALF = ALPHA ANGLE (DEG)
SBTA = BETA ANGLE (DEG)
AZRTE = AZIMUTH ANGLE RATE (DEG/SEC)
ELRTE = ELEVATION ANGLE RATE (DEG/SEC)

** EXAMPLE **

AN EXAMPLE CASE IS INCLUDED IN THE CODE. IF THIS SOURCE IS COMPILED,
LINKED, AND EXECUTED, OUTPUTS WILL GO TO UNIT 6. THEIR VALUES SHOULD
BE:

R = 43760.6816 ARDOT = -9.87364578
SRANG = 25.2644926 SPANG = 28.2407990
SRRTE = -0.926815558E-01 SPRTE = .688237743E-02
SALF = -36.1578255 SBTA = 9.27430439
AZRTE = .302744657E-01 ELRTE = -.105446591

COMMON /TMR/X,Y,Z,VX,VY,VZ,
1     DLP(3),DEL(3),DUE(3),
2     DSU(3),THAZL1,THEL1,THAZU1
COMMON /INPUT/RO(3),VO(3),EWB(3)
COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,SBTA,
1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
DIMENSION DLP(3),DEL(3),DUE(3),DSU(3)
DIMENSION AZL(3,3),ELV(3,3),AZU(3,3)
DIMENSION DPT(3),DUT(3),DET(3),DST(3)
DIMENSION DLAZ(3,3),DELY(3),DAZU(3)
DIMENSION VPT(3),VLAZ(3),VELV(3),VST(3)
DATA DEGRAD/57.275/,PI/3.14159/

THE EWB PARAMETERS ARE ALWAYS DEFINED AS 0.0
EWB(1)=0.0
EWB(2)=0.0
EWB(3)=0.0

EXAMPLE CASE VALUES:

A-8
** INPUTS **
WSMR will normally provide X,Y,Z,VX,VY,VZ. Ref is Pearl survey point.
This is provided via common TMR block

DPT(1)=X
DPT(2)=Y
DPT(3)=Z
VPT(1)=VX
VPT(2)=VY
VPT(3)=VZ

** CONSTANTS **
DLP(I); DEL(I); AND DUE(I) will be provided one time after installation
of the antenna pedestal
This is provided via common TMR block

DLP(1)=0.0
DLP(2)=0.0
DLP(3)=0.0
DEL(1)=0.0
DEL(2)=0.0
DEL(3)=0.0
DUE(1)=0.0
DUE(2)=0.0
DUE(3)=0.0

** CONSTANTS FROM SIMULATION DATA TAPE **
This is provided via common TMR block

DSU(1)=0.0
DSU(2)=0.0
DSU(3)=0.0
THAZL1=0.0
THEL1=0.0
THAZU1=0.0

Example angle values are equated here.
THAZL1=THAZL2
THEL1=THEL2
THAZU1=THAZU2

Convert to radians
THAZL=THAZL1/DEGRAD
THEL=THEL1/DEGRAD
THAZU=THAZU1/DEGRAD

Set up the rotational matrices
CALL AZGEN(AZL,THAZL)
CALL ELGEN(ELV,THEL)
CALL AZGEN(AZU,THAZU)

Convert target in Pearl to target at gimbals
DO 11 I=1,3
11 DLT(I)=DPT(I)-DLP(I)
CALL MULT31(AZL,DLT,DLAZ)
DO 21 I=1,3
21 DET(I)=DLAZ(I)-DEL(I)
CALL MULT31(ELV,DET,DELV)
DO 31 I=1,3
31 DUT(I)=DELV(I)-DUE(I)
CALL MULT31(AZU,DUT,DAZU)
DO 41 I=1,3
41 DST(I)=DAZU(I)-DSU(I)
C THESE ARE THE THREE TARGET COORDINATES IN RADAR GIMBAL REFERENCE.
RO(1)=DST(1)
RO(2)=DST(2)
RO(3)=DST(3)
C CONVERT TO VELOCITIES REFERENCED TO GIMBALS
CALL MULT31(AZL,VPT,VLAZ)
CALL MULT31(ELV,VLAZ,VELV)
CALL MULT31(AZU,VELV,VST)
C THESE ARE VELOCITIES IN GIMBAL REFERENCE.
VO(1)=VST(1)
VO(2)=VST(2)
VO(3)=VST(3)
C C RO(1) VO(1) I=1,3 SHUTTLE BODY POS AND VEL VECTOR
C C CALCULATE THE KU-BAND RADAR PARAMETERS BASED ON THE INPUTS.
C23=COSD(23.)
S23=SIND(23.)
XI=RO(2)*C23-RO(3)*S23
YI=RO(2)*S23-RO(3)*C23
ZI=RO(1)
RO(1)=XI
RO(2)=YI
RO(3)=ZI
VX=VO(2)*C23-VO(3)*S23
VY=VO(2)*S23-VO(3)*C23
VZ=VO(1)
VO(1)=VX
VO(2)=VY
VO(3)=VZ
CALL ACT
SRRTE=SRRTE*(DEGRAD/1000.)
SPRTE=SPRTE*(DEGRAD/1000.)
SALF=AL*DEGRAD
SBTA=BT*DEGRAD
AZRTE=AZRATE*DEGRAD
ELRTE=ELRATE*DEGRAD
C THE EXAMPLE CASE RESULTS ARE:
C WRITE(6,*)R,ARDOT
C WRITE(6,*)SRANG,SPANG
C WRITE(6,*)SRRTE,SPRTE
C WRITE(6,*)SALF,SBTA
C WRITE(6,*)AZRTE,ELRTE
RETURN
END
SUBROUTINE AZGEN(AZ,ANGAZ)
C THIS SUBROUTINE PRODUCES A 3X3 MATRIX, AZ, FOR
C AN AZIMUTH TABLE ROTATION OF ANGAZ RADIANS.
DIMENSION AZ(3,3)
DO 10 I=1,3
10 DO 10 J=1,3
10 AZ(I,J)=0.0
AZ(1,1)=COSD(ANGAZ)
AZ(1,2)=SIND(ANGAZ)
AZ(2,1)=SIND(ANGAZ)
AZ(2,2)=COSD(ANGAZ)
AZ(3,3)=1.0
RETURN
SUBROUTINE ELGEN(EL, ANGEL)
DIMENSION EL(3,3)
DO 10 I=1,3
10 EL(I,1)=0.0
EL(1,1)=COS(ANGEL)
EL(1,3)=SIN(ANGEL)
EL(2,2)=1.0
EL(3,1)=SIN(ANGEL)
EL(3,3)=COS(ANGEL)
RETURN
END

SUBROUTINE ACT

*****************************************************************************
* THIS SUBROUTINE INITIALIZES THE ANGLE TRACKING LOOPS, THE RANGE TRACKING LOOP, AND THE VELOCITY PROCESSOR —— STEADY STATE CONDITIONS ARE ASSUMED. 
*****************************************************************************

SUBROUTINE ACT
COMMON /ACIDAT/R,ARDOT,SPANG,SRANGE,SRRTE,SRF,SLF,SBTA
2,ER(3),EV(3),ERTO(3),DR(3),CP,SP,PSI,PSBIAS,DUM2(7),TRB(3,3)
COMMON /INPUT/ERT(3),EVT(3),EBW(3),DUM(18)
COMMON /SYSDAT/TSAM,CP,SP,PSI,PSBIAS,DUM2(7),TRB(3,3)

CSTEP 1-1: COMPUTE INITIAL INNER AND OUTER GIMBAL POSITIONS.
C (NOTE: TRANSFORM CONSISTS OF TRANSLATION PLUS ROTATION.)
PREFORMTRANSLATION —— SHIFT TO RADAR FRAME ORIGIN.
DO 1 I=1,3
1 ERTO(I)=ERT(I)-DR(I)
C TRANSFORM TARGET POSITION FROM BODY TO RADAR FRAME.
CALL MULT31(TRB,ERTO,ER)
C TRANSFORM TARGET VELOCITY FROM BODY TO RADAR FRAME.
CALL MULT31(TRB,EVT,EV)

C COMPUTE INNER(BETA) GIMBAL POSITION — BT.
IF(ER(1).EQ.0.0.AND.SQ.EQ.0.0) STOP
BT=ATAN2(ER(1),SQ)
ER2=ER(2)
ER3=ER(3)

C COMPUTE OUTER(DELTA) GIMBAL POSITION — AL.
IF(ER2.EQ.0.0.AND.ER3.EQ.0.0) GO TO 8
AL=ATAN2(ER2,ER3)
GO TO 9

CSTEP 1-2: COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH AND ELEVATION RATES.

PRELIMINARY TRIGONOMETRIC COMPUTATIONS.
9 CA=COS(AL)
SB=SIN(AL)
CB=COS(BT)
$\text{SD} = \sin(BT)$

C TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER

C GIMBAL(G) REFERENCE FRAME.

$WGX = CP \cdot WGB + SP \cdot WGB^2$  \hspace{1cm} 00015980

$WGY = CA \cdot (-SP \cdot WGB + CP \cdot WGB^2) + SA \cdot WGB^3$  \hspace{1cm} 00015990

$WGZ = SA \cdot (-SP \cdot WGB + CP \cdot WGB^2) + CA \cdot WGB^3$  \hspace{1cm} 00015990

C COMPUTE THE RANGE TO TARGET.

$R = \sqrt{(ER1 + ER2)^2 + (ER3 + ER2)^2 + (ER3)^2}$  \hspace{1cm} 00015940

$YZ = (ERI \cdot EV1 + ER2 \cdot EV2 + ER3 \cdot EV3) / R$  \hspace{1cm} 00015950

C COMPUTE RANGE RATE TO TARGET

$ARD = (ER1 \cdot EV1 + ER2 \cdot EV2 + ER3 \cdot EV3) / R$  \hspace{1cm} 00015960

C COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE (AZRATE).

$VGY = CA \cdot EV2 + SA \cdot EV3$  \hspace{1cm} 00015970

$AZRATE = VGY / R + (CB \cdot WGX - SB \cdot WGZ)$  \hspace{1cm} 00015980

C COMPUTE TARGET INERTIAL LOS ELEVATION RATE (ELRATE).

$ELRATE = (CB \cdot EV1 - SB \cdot (SA \cdot EV2 + CA \cdot EV3)) / R + WGY$  \hspace{1cm} 00015990

C -----------------------------------------------

C STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE

C CALL GAMMA(TX1, -(BT+BTBIAS))

C CALL THETA(TX2, -(AL+ALBIAS))

C CALL MULT33(TX2, TX1, TX3)

C CALL PHI(TX2, PSI)

C CALL MULT33(TX2, TX3, TBL)

C -----------------------------------------------

C STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO BODY FRAME FOR DISPLAY.

C SPHTE = $180^\circ + (TBL(2,1) \cdot AZRATE + TBL(2,2) \cdot ELRATE)$

C UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES

C FOR DISPLAY.

C SPHTE = $180^\circ + (TBL(2,1) \cdot AZRATE + TBL(2,2) \cdot ELRATE) - 180^\circ$  \hspace{1cm} 00026200

C SPHTE = $180^\circ + (TBL(2,1) \cdot AZRATE + TBL(2,2) \cdot ELRATE)$  \hspace{1cm} 00026220

C UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES

C FOR DISPLAY.

C SPHTE = $180^\circ + (TBL(1,1) \cdot AZRATE + TBL(1,2) \cdot ELRATE)$  \hspace{1cm} 00026240

C UPDATE ANTENA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.

C SPHTE = $ASIN(TBL(1,3)) + 57.29576$  \hspace{1cm} 00026260

C UPDATE ANTENA IN ORBITER BODY COORDINATES FOR DISPLAY.

C IF(TBL(1,3) = 0.0) AND TBL(3,3) = 0.0 THEN GO TO 5

C SRANG = $ATAN2(-TBL(2,3), TBL(3,3)) + 57.29576$  \hspace{1cm} 00026300

C GO TO 7

C IF(TBL(1,3) = 0.0) AND TBL(3,3) = 0.0 THEN GO TO 5

C SRANG = $ATAN2(-TBL(2,3), TBL(3,3)) + 57.29576$  \hspace{1cm} 00026310

C IF(TBL(1,3) = 0.0) AND TBL(3,3) = 0.0 THEN GO TO 5

C SRANG = $ATAN2(-TBL(2,3), TBL(3,3)) + 57.29576$  \hspace{1cm} 00026330

C RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., $-90^\circ < \text{SPHTE} < 90^\circ$.

C IF(SPHTE < 0) THEN GO TO 7

C SRANG = $ATAN2(-TBL(2,3), TBL(3,3)) + 57.29576$  \hspace{1cm} 00026350

C IF(SPHTE > 90) THEN GO TO 7

C SRANG = $ATAN2(-TBL(2,3), TBL(3,3)) + 57.29576$  \hspace{1cm} 00026370

C CONTINUE

C RETURN

C END

C ********** SUBROUTINE INITIALIZES ALL DATA REQUIRED BY THE SEARCH. **********

A-12
SUBROUTINE INITIALIZES ALL DATA REQUIRED BY THE SEARCH, ACQUISITION, AND TRACK SUBPROGRAMS.

SUBROUTINE DATA
REAL IDUM1
COMMON /RTDAT/IDUM1(2),RBIAS,DUM1(9)
COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA,
2 TGTSG,GPS,GAUSS(32)
DIMENSION A(3,3),B(3,3),C(3,3)
REAL LT,KTS

C SYSTEM PARAMETERS
PI=3.1415926
PII=PI/180.
C RADAR FRAME YAW ANGLE IN BODY COORDINATES (DEGREES).
PSI=PII*67.e
CP=COS(PSI)
SP=SIN(PSI)
C RADAR LOCATION OFFSET FROM ORBITER C.G. IN BODY COORD. (FEET)
C VALUES MODIFIED MAR 24 83 PER FM8 MEMO

DR(1)=45.738
DR(2)=11.138
DR(3)=5.79

C RANGE BIAS ERROR IS COMPUTED IN SUBROUTINE RTRACK AS FUNCTION OF RANGE
C ALPHA GIMBAL BIAS.
ALBIAS=0.0
C BETA GIMBAL BIAS.
BTBIAS=0.0
C RADAR PLATFORM ORIENTATION ERRORS WITH RESPECT TO BODY FRAME.
C YAW ANGLE ERROR.
PSBIAS=PII*0.1
C ROLL ANGLE ERROR.
RLBIAS=PII*0.25
C PITCH ANGLE ERROR.
PTBIAS=PII*0.25
C
C NBIA=0 FOR NO BIAS AND RADAR AT ORIGIN
C
C NBIA=1
IF(NBIAS.NE.0)GO TO 700
701 FORMAT(' ALL ANGLE BIAS SET TO ZERO RADAR AT ORGIN')
DO 4 I=1,3
  4 DR(I)=0.0
  PS=0.0
  PSBIAS=0.0
  RLBIAS=0.0
  PTBIAS=0.0
700 CONTINUE
C COMPUTE MATRIX OF TRANSFORMATION FROM BODY FRAME TO RADAR FRAME.
CALL PHI(B,PSI+PSBIAS)
CALL THETA(A,RLBIAS)
CALL MULT33(A,B,C)
CALL GAMMA(A,PTBIAS)
CALL MULT33(A,C,TRB)
C *****************************************
C * SYSTEM SAMPLE INTERVAL *
C *****************************************
C *****************************************************
C * COMPUTE SNR CONSTANT *
C *****************************************************
C EQUIVALENT ONE-SIDED NOISE POWER SPECTRAL DENSITY (MW/KHZ)
KTS=137.5
KTS=10.*(+0.1*KTS)
C SYSTEM LOSSES ON TRANSMIT (DB).
LT=2.5
LT=10.*(+0.1*LT)
C ONE-WAY ANTENNA GAIN (DB).
G=37.7
G=10.*(+0.1*G)
ALMBA=0.070845
C CONSTANT FOR PASSIVE TRACKING SNR COMPUTATION.
CPA=4.*(+0.1*ALMBA+2)/((4.*PI)**3*LT*KTS)
C BEACON PARAMETER (DBM)
BCN=44.0
BCN=10.*(+0.1*BCN)
C CONSTANT FOR ACTIVE TRACKING SNR COMPUTATION.
GA=4.*G+ALMBA**2+2*BCN/((4.*PI)**2*LT*KTS)
C CONSTANT FOR PASSIVE MODE VIDEO SNR COMPUTATION (DB).
GPS=183.9
C CONSTANT FOR ACTIVE MODE VIDEO SNR COMPUTATION (DB).
GAS=146.9
C *****************************************************
C * RANDOM NUMBER GENERATOR SEEDS *
C *****************************************************
C NS1=48
NS2=135
NN(I)=0
C INITIALIZE NOISE SEQUENCE.
DO 2 I=1,32
2 GAUSS(I)=ANORM(NS1,NS2)
IF(ITEST.EQ.2)GO TO 6341
ITEST=2
C WRITE(6,592)
592 FORMAT(1HI,' RANDOM NUMBER INITIALIZATION')
C WRITE(6,593)(GAUSS(I),I=1,320)
593 FORMAT(8F8.4)
C WRITE(6,592)
6341 CONTINUE
C *****************************************************
C * DEFINE TARGET PARAMETERS *
C *****************************************************
C TARGET SEARCH CROSS-SECTION (FIXED TEMPORARILY).
TGTSIG=10.0
RETURN
END
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,ISRCHE,ISRCHG,IAZS,IELS,ISLR,
EDRNG,EDPA,EDRA
COMMON /ICNTL/10LDPW,10LDMD,10LDSM,ISHOLD,KMSCLK,KMUP,KSNCLK,
KSNMAX,KACCLK,MTP,MZ1,MZ0,MSS,MTKINT,MRNG,MSAM,MPRF,
MBKTRK,MBTSUM,MBT(8)
COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
SRTE,SRSS,MADVF,MRDVF,MRDVF,MRDVF
COMMON /INPUT/ERTO(3),EVT0(3),EWB(3),TBT(3,3),TBTD(3,3)
COMMON /ATDAT/DUM1(10),PREF,REF
COMMON /SYSDAT/TS,DUM2(14)
COMMON /CGMAIN/R0(3),V0(3),A0(3)
COMMON /DSCRM/DUM3(6),SIGBAR,SNRD,SIGDB
COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDD
C ITARG = 0 POINT TARGET RCS OF POINT TARGET
C SRCIS IS VARIABLE NAME OF RCS VALUE
C SRC = 3.27 IS IMSQ TARGET.
C SRC = 3.27
DO I=1,3
DO J=1,3
EWT(I,J)=0.
TBT(I,J)=0.
IF(I.EQ.J)TBT(I,J)=1.
TBTD(I,J)=0.
ENDDO
ENDDO
KOLD=1
CALL SYSINT
IPWR=3
IMODE=2
IAXM=1
ITXP=1
ISRCHE=0
IAZS=0
IELS=0
ISLR=0
ISRCHE=0
EDRNG=500.0
EDPA=0.0
EDRA=0.0
PII=3.14159265/180.
EDPA=EDPA*PII
EDRA=EDRA*PII
MTF=0
MTP=1
MTP=1
RETURN
END
FUNCTION ANORM(K1,K2)
Y1=RNDFU(K1)
Y2=RNDFU(K2)
TPI=6.2831852
ANORM=SNRT(-2.*ALOG(Y1))*COS(TPI*Y2)
RETURN
END
* THIS SUBROUTINE UPDATES AZ AND EL INERTIAL LOS RATES, THE • 00025250
* ALPHA AND BETA GIMBAL RATES, THE ALPHA AND BETA GIMBAL • 00025270
* POSITIONS, AND THE TARGET PITCH AND ROLL ANGLES FOR THE • 00025280
* DISPLAY. • 00025290

SUBROUTINE ATRACK • 00025300
REAL INTT, IAZDSC, IELDSC • 00025310
COMMON /CNTL/ IPWR, IMODE, IDUMC(7), DUMC(3) • 00025320
COMMON /INPUT/ DUM(6), EMB(3), DUM2(18) • 00025330
COMMON /OUTPUT/ IDUM(3), DUM(3), SPANG, SPRTE, SRRTE, SRSS • 00025340
COMMON /ICNTL/ I2DUM(14), MRNG, MSAM, MPRF, IDUM2(11) • 00025350
COMMON /SYSDAT/ TSAM, DR(3), CP, SP, PSI, PSBIAS, ABLIAS, BTBIAS • 00025360
COMMON /ICNTL/ I2DUM(14), MRNG, MSAM, MPRF, IDUM2(11) • 00025370
COMMON /SYSDAT/ TSAM, DR(3), CP, SP, PSI, PSBIAS, ABLIAS, BTBIAS • 00025380
COMMON /ICNTL/ I2DUM(14), MRNG, MSAM, MPRF, IDUM2(11) • 00025390
COMMON /SYSDAT/ TSAM, DR(3), CP, SP, PSI, PSBIAS, ABLIAS, BTBIAS • 00025400
COMMON /DSCRM/ AZDISC, ELDISC, DUMI(7) • 00025410
COMMON /DSCRM/ AZDISC, ELDISC, DUMI(7) • 00025420
DIMENSION ATI(1e.2), AT2(1e.2), TX1(3,3), TX2(3,3), TX3(3,3), TBL(3,3) • 00025430
DIMENSION TDC(3) • 00025440
DATA AT1/9*1.5529E-3, 2.6E-4, 6.3975E-3, 1.5529E-3, 2.6E-4/ • 00025450
DATA AT2/9.6.5967E-3, 2.3725E-3, 3.23725E-3/ • 00025460
DATA TDC/0.65122118, 6.1195161, 0.2561557/ • 00025470
DEFINITION: ATI = KEO, = (WN) / (4xDIFFERENCE PATTERN SLOPE) WHERE • 00025480
WN IS NATURAL FREQUENCY OF THE LOOP. • 00025490
DEFINITION: AT2, KEO = TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE • 00025500
CONVERGENCE TIME. • 00025510
TCON = TSAM / TDC(MPRF) • 00025520
STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE • 00025530
STEP 1: UPDATE ANTAGNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANS- • 00025540
FORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW • 00025550
ANGLE ERROR WRT BODY FRAME). • 00025560
CALL GAMMA(TX1, -(BT+BTBIAS)) • 00025570
CALL THETA(TX2, -(AL+ALBIAS)) • 00025580
CALL MULT33(TX2, TX1, TX3) • 00025590
CALL PHI(TX2, -PSI) • 00025600
CALL MULT33(TX2, TX3, TBL) • 00025610
STEP 2: UPDATE ESTIMATED TARGET INERTIAL AZIMUTH AND ELEVATION • 00025620
RATES IN ANTENNA LOS FRAME. • 00025630
QUANTIZE THE ANGLE DISCRIMINANTS TO 3/16 DB. • 00025640
IAZDSC=INTT(5.333333+AZDSC*TCON+0.5)/TCON • 00025650
IELDSC=INTT(5.333333+ELDSC*TCON+0.5)/TCON • 00025660
IF(IELDSC.GT.255) IELDSC=255 • 00025670
IF(AZDSC.GT.255) AZDSC=255 • 00025680
IF(IELDSC.LT.-256) IELDSC=-256 • 00025690
IF(AZDSC.LT.-256) AZDSC=-256 • 00025700
ADSC = 0.8431*1AZDSC • 00025710
EDSC = 0.8431*1ELDSC • 00025720
C UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE. • 00025730
AZRATE=AZRATE+TSAM*ATI(MRNG, IMODE)*ADSC • 00025740
C UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE. • 00025750
ELRATE = ELRATE + TSAM * AT1(MRNG, IMODE) * EDSC

C **********************************************************************
C * STEP 3: UPDATE INNER AND OUTER GIMBAL RATES. *
C **********************************************************************
C COMPUTE REQUIRED COMPONENTS OF ORBITER ANGULAR VELOCITY VECTOR IN
C OUTER GIMBAL FRAME.
C
WGX = CP * EWB(1) * SP * EWB(2)
WGY = CA * (-SP * EWB(1) + CP * EWB(2)) + SA * EWB(3)
WGZ = SA * (-SP * EWB(1) + CP * EWB(2)) + CA * EWB(3)
C OUTER GIMBAL RATE.
IF (ABS (CB) .LT. 1.E-6) GO TO 2

ALRATE = (AZRATE + AT2(MRNG, IMODE) * ADSC + WGZ * SB) / CB - WGX
GO TO 4

2 ALRATE = 0.
4 CONTINUE
C INNER GIMBAL RATE.
BTRATE = (ELRATE + AT2(MRNG, IMODE) * EDSC) - V_Y
C
C **********************************************************************
C * STEP 4: UPDATE INNER AND OUTER GIMBAL POSITIONS. *
C **********************************************************************
C OUTER GIMBAL POSITION (ALPHA ANGLE)
AL = AL + TSAM * ALRATE
C INNER GIMBAL POSITION (BETA ANGLE)
BT = BT + TSAM * BTRATE
C
C ADD ALPHA AND BETA TO OUTPUT IN DEG
SSALP = AL * 57.29576
SSBET = BT * 57.29576
C
C **********************************************************************
C * STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO *
C * BODY FRAME FOR USE IN DISPLAYS AND G AND N. *
C **********************************************************************
C NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW
C ANGLE ERROR WRT BODY FRAME.
C UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
C FOR DISPLAY.
SPRTE = 1800. * (TBL(2,1) * AZRATE + TBL(2,2) * ELRATE)
C UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
C FOR DISPLAY.
SRRTE = 1800. * (TBL(1,1) * AZRATE + TBL(1,2) * ELRATE)
C UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
SPANG = ASIN(TBL(1,3)) * 57.29576
C UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.
IF (TBL(2,3) .EQ. 0.0 AND TBL(3,3) .EQ. 0.0) GO TO 5
SRANG = ATAN2(-TBL(2,3), TBL(3,3)) * 57.29576
GO TO 7

5 IF (TBL(1,3) .GT. 0.0) SRANG = 90.0
IF (TBL(1,3) .LT. 0.0) SRANG = -90.0
IF (TBL(1,3) .EQ. 0.0) STOP
GO TO 7

7 IF (SPRTE .LE. 90.) GO TO 10
SRANG = (180. - ABS(SPANG)) * (SPANG / ABS(SPANG))
SRANG = (180. - ABS(SRANG)) * (SRANG / ABS(SRANG))
10 CONTINUE
C RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND
-180.<SRANG<180.
C
180. < SPANG < 180.
7 IF (SPRTE .LE. 90.) GO TO 10

C NOTE: DEBUGGING PRINT STATEMENTS.
C WRITE(*,899)
899 FORMAT (/ 'ATRACK DEBBUGGING DATA'
C WRITE(6,900) ALRATE, BTRATE, AZRATE, ELRATE, SPRTE, SPRTF
C WRITE(6,901) TBL(1,1), TBL(1,2), TBL(2,1), TBL(2,2)
C WRITE(6,902) ADISC, ELDISC, ADSC, EDSC

A-17
SUBROUTINE BRKTRK
REAL IVMAX, THRSHC, THRSHO, IVDISC, INTT, IODISC
COMMON /ICNTL/IDUM2(17), MBKTRK, MBTSUM, MBT(8)
COMMON /DSCRM/DUM(3), VDISC, DUMI, 0DISC, DUM2(3)
DATA IVMAX, THRSHC, THRSHO/51., 14., -11./
SUBROUTINE BRKTRK
REAL IVMAX, THRSHC, THRSHO, IVDISC, INTT, IODISC
COMMON /ICNTL/IDUM2(17), MBKTRK, MBTSUM, MBT(8)
COMMON /DSCRM/DUM(3), VDISC, DUMI, 0DISC, DUM2(3)
DATA IVMAX, THRSHC, THRSHO/51., 14., -11./

• THIS SUBROUTINE IMPLEMENTS THE BREAK-TRACK ALGORITHM •

* STEP 1: DETERMINE STATUS OF L-H DISCRETE (FTH) *

C STEP 1-1: QUANTIZE THE VELOCITY DISCRIMINANT TO 3/16 DB STEPS.
IVDISC=INTT(VDISC*5.333333+0.5)
C STEP 1-2: DETERMINE STATUS OF L-H DISCRETE.
IFTH=0
IF(ABS(VDISC).GE.IVMAX) IFTH=1
C
• STEP 2: DETERMINE STATUS OF ON-TARGET DISCRETE (OT) *

C STEP 2-1: QUANTIZE THE O-DISCRIMINANT TO 3/16 DB STEPS.
IODISC=INTT(ODISC*5.333333+0.5)
C STEP 2-2: DETERMINE STATUS OF ON-TARGET DISCRIMINANT.
IOT=0
IF(IODISC.GE.THRSHC) IOT=1
C
• STEP 3: DETERMINE STATUS OF ADJACENT ON-TARGET DISCRETE (AOT) *

IAOT=0
IF(IODISC.LE.THRSHO) IAOT=1
C
• STEP 4: COMBINE ABOVE DISCRETES TO DETERMINE STATUS OF NO-TARGET DISCRETE (NOTARG).
C DEFINITION: THE NO-TARGET DISCRETE IS HIGH (OR 1) IF THE DISCRETES FTH, OT, AND AOT ARE ALL LOW (OR 0).
NOTARG=(1-IFTH)*(1-IOT)*(1-IAOT)
C
• STEP 5: DETERMINE STATUS OF BREAK-TRACK FLAG (MBKTRK) *
C DEFINITION: BREAK-TRACK SHALL BE DECLARED IF NOTARG=1 FOR AT LEAST 5 OF THE MOST RECENT 8 DATA CYCLES.
C
STEP 5-1: UPDATE MOVING WINDOW-OF-8 SUM (MBTSUM).
MBTSUM=MBTSUM+(NOTARG-MBT(1))
C
STEP 5-2: UPDATE STORAGE REGISTERS.
DO 10 I=1,7
STEP 5-3: DETERMINE STATUS OF BREAK-TRACK FLAG (1=BREAK-TRACK).

MBKTRK=MBTSUM/5

NOTE: DEBUGGING PRINT STATEMENTS.

WRITE(6,900) ID,THO,THC,VD,THV,THS

900 FORMAT(' ID,THO,THC,VD,THV,THS=',6F15.8)

RETURN

END

SUBROUTINE (FAR

COMMON /CNTL/INPRW,JMODE,ITXP,IASM,IDUMC(5),EDRNG,DUMC(2)

COMMON /OUTPUT/MSWF,MTF,MSF,DUMI(7),IDUMI(4)

COMMON /ICONTL/IDUM2(8),KACCLK,MTP,IDUM3(4),MRNG,MSAM,MPRF

COMMON /TGTDAT/NT,DUM3(SEQO),RO(3),ROU(3),CGRNGE,CGVEL

COMMON /DETDAT/SIGMA,CGANG

DIMENSION RI(6),PW(6),NP(6),FW(3),TPRI(3),TS(2),P(4)

DATA NRI,NSRCN/6.37,

C,ALMDA/983.5,0.076845/,RI/2552.,5772.,11544.,23089.,43747.,57722./,PW/0.122,4.15,8.3,

F,16.6,33.2,66.4/,NP/1,2,4,8,16,32/,FW/7.7215,3.369e-0.296S/,TS/e.122,2.e75/,TPRI/143.5,334.7,3731.1/

DATA P/6,e.e,.61,.63,2,.64,.668,.612,.615,.643,.653,.676,.167,6e68766

3.882,.918,.955,.966,.976,.986,.989,.991,.997,.996/

PI-3.14159265 .
C SET SAMPLE RATE.
   MSAM=2
C SET PRF.
   MPRF=1

C ***********************************************
C * STEP 2: COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT *
C ***********************************************
   SNR=SNRV(SIGMA,GCRNGE)

C ***********************************************
C * STEP 3: IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNR *
C ***********************************************
   SNR=SNR+BETA2

C ***********************************************
C * STEP 4: COMPUTE NET PROCESSOR GAIN AND COMBINE *
C WITH SNR TO FORM SNRD.  *
C ***********************************************
   SNRD=SNR+SNRD

C STEP 3-1: CHECK SCAN FLAG.
   IF(MSF.EQ.1) GO TO 25

C STEP 3-2: COMPUTE BEAMSHAPE LOSS — BASED UPON C.G. POSITION OFF BORESIGHT.
   BETA2=SPAT(CGANG)**2

C STEP 3-3: ADD BEAMSHAPE LOSS TO NOMINAL SNR, I.E. COMPUTE ACTUAL SNR.
   SNR=SNR+BETA2

C STEP 4-1: COMPUTE RANGE GATE LOSS (RGL) — DIFFERS FOR GPC AND AUTO/ MANUAL MODES.

C COMPUTE EQUIVALENT RANGE OF XMIT PULSEWIDTH.
   25 CTD2=C.PW(WRNG)/2.

C DETERMINE OPERATING MODE
   IF(IAWM.GE.3) GO TO 30

C COMPUTE RGL FOR GPC MODES.
   DEL=ABS(EDRNG-GRNGE)/CTD2
   IF(DEL.GE.1.5) RGL=0.0
   IF(DEL.GE.0.5.AND.DEL.LT.1.5) RGL=6666666*(1.5-DEL)**2
   GO TO 35

C COMPUTE RGL FOR AUTO/ MANUAL MODES
   DEL=ABS(EDRNGE)/CTD2
   IF(DEL.LE.1.0) RGL=DEL
   2 RGL=DEL*DEL

C STEP 4-2: COMPUTE NET PRESUM GAIN — SAME FOR ALL PASSIVE ANTENNA STEERING MODES.

C COMPUTE DOPPLER FREQUENCY ASSOCIATED WITH TARGET RADIAL VELOCITY
   FDOP=2.*CGVEL/ALMDA*1.0E-6

C COMPUTE ARGUMENT ASSOCIATED WITH TARGET VELOCITY
   ARG=PI*FDOP*TS(MSAM)
C COMPUTE NET PRESUM GAIN
    PSG=SUM(ARG.NP(MRNG))
C STEP 4-3: COMPUTE NET DOPPLER FILTER GAIN — SAME FOR ALL PASSIVE
    ANTENNA STEERING MODES.
C COMPUTE NUMBER OF DOPPLER FILTER NEAREST TARGET.
    MFIL=MOD(INT(CGVEL/FW(MPRF)+320.5),32)
C COMPUTE ARGUMENT ASSOCIATED WITH TARGET DOPPLER
    ARG=PI*(FLOAT(MFIL)/32.+FDOP+TPRI(MPRF))
C COMPUTE NET DOPPLER FILTER GAIN
    DFG=SUM(ARG,16)
C STEP 4-4: COMPUTE NET PROCESSOR GAIN.
    PPG=RCL+PSG+DFG
C STEP 4-5: COMPUTE SNR AT DOPPLER FILTER OUTPUT
    SNR=SNR+PPG
C
C • STEP 5: DETERMINE PROBABILITY OF DETECTION BASED UPON SNR •
C
C STEP 5-1: DETERMINE INDEX TO ACCESS APPROPRIATE CURVE
    IF(IASM.GE.3) GO TO 40
    NCRV=1
    GO TO 45
40 NCRV=3
C ADJUST INDEX FOR SCANNING
    45 NCRV=NCRV+MSF
C STEP 5-2: CONVERT SNR TO DB.
    IF(SNR.LE.1.0E-88) GO TO 50
    SNR=10.*ALOG10(SNR)
    GO TO 55
50 SNR=100.
C STEP 5-3: SNR OUTSIDE (O DB, +20 DB) INTERVAL" — IF SO, SET
    OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.
C IF SNRD < 0. DB — DECLARE A MISS.
    55 IF(SNR.LE.0.) GO TO 60
C IF SNRD > 20. DB — DECLARE A HIT.
    IF(SNR.GT.20.) GO TO 65
C STEP 5-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR
    INTERPOLATION.
    SCALE=(SNR+8.)*2.+1.00000001
    ISNR=INT(SCALE)
    REMAIN=SCALE-FLOAT(ISNR)
C STEP 5-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION.
    PROB=P(ISNR)+REMAIN*(P(ISNR+1)-P(ISNR))
C
C • STEP 6: DETERMINE OUTCOME OF DETECTION ATTEMPT •
C
C X=RNDU(NSRCH)
IF(X.LE.PROB) GO TO 65

C

C • STEP 7: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT •

STEP 7-1: IF NO DETECTION — SET TARGET PRESENT FLAG LOW.

60 MTP=0
RETURN

STEP 7-2: IF DETECTION SUCCESSFUL — SET TARGET PRESENT FLAG HIGH AND INITIALIZE ACQUISITION CLOCK.

65 MTP=1
KACCLK=0
RETURN
END

C

C • THIS SUBROUTINE UPDATES ALL RADAR INTERNAL CONTROLS. •

SUBROUTINE CNTRLS
REAL INTT,NFIL,IRNG,IRDOT
COMMON /CNTL/IPWR,IMODE,DUMC(7),DUMC(3)
COMMON /OUTPUT/IDUMB(3),SRNG,SRODOT,DUM2(5),IDUM(4)
COMMON /CNTLT/IDUM(14),MRNG,MSAM,MPRF,IDUM1(10),MPFOLD
COMMON /RTDAT/IRDOT,IRNG,RI,VEST(4),MDF(5)
DIMENSION RI(10),P_(3)
C RI(4) CHANGED TO 2560 FROM 2552
DATA RI/120.,240.,780.,2560.,5772.,11544.,23089.,43747.,
2.,57722.,1.8228E+6/
DATA P_/7.7215,3.3090,B.2969/,NFIL/10/
C

C • STEP 1: SET RANGE INTERVAL PARAMETER •

XRNG=IRNG=0.3125
DO 60 I=1,NRI
IF(XRNG.LE.RI(I)) GO TO 70
60 CONTINUE
70 MRNG=1
IF(MRNG.GT.NRI) STOP
C

C • STEP 2: SET SAMPLE RATE PARAMETER •

IF(IMODE.GE.2) GO TO 74
IF(MRNG.GT.9) GO TO 72
MSAM=1
GO TO 80
72 MSAM=2
GO TO 80
74 MSAM=2
GO TO 76
76 MSAM=2
C

C • STEP 3: SET PRF PARAMETER •

STEP 3-1: DETERMINE IF IN ACTIVE OR PASSIVE MODE.
```plaintext
80 IF(IMODE .GE. 2) GO TO 84
82 IF(MRNG .GT. 9) GO TO 82
84 IF(MRNG .GT. 9) GO TO 86
86 CONTINUE

STEP 3-2: DETERMINE CORRECT PRF FOR GIVEN OPERATING MODE.
  IF(MPRF .LT. 1) GO TO 90
  GO TO 90
84 IF(MRNG .GT. 9) GO TO 86
  MPRF = 2
  CONTINUE

STEP 3-3: IF PRF HAS CHANGED FROM PREVIOUS DATA CYCLE, THEN
  RESET THE 5 DOPPLER TRACKING FILTERS ACCORDINGLY.
  IF(MPFOLD .EQ. MPRF)
    GO TO 96
  NFI L = INT(((-SRDOT/F1N(MPRF)) .- 1.5) + 31998.0)
  XX = AMOD(NFI L, 32.)
  MDF(1) = INT(XX)
  DO 95 I = 1, 4
    MDF(I+1) = MOD(KDF(I+1), 32)
   95 MDF(I) = INT(MDF(I))
  96 MPFOLD = MPRF

NOTE: DEBUGGING PRINT STATEMENTS.
  WRITE(6, 999) MPRF, MPFOLD, MDF(1)
  RETURN
END

SUBROUTINE DETECT
COMMON /CNTL/IPWR, IMODE, ITXP, IASM, IDUMC(5), EDRNC, DUMC(2)
COMMON /ICNTL/IDUM2(9), MTP, IDUM3(17)
COMMON /SYSDAT/DUM2(12), TGTSIG, GPS, GAS
COMMON /TGTDAT/NT, DUM3(508), RO(3), ROU(3), CRNGE, CGVEL
COMMON /DETDAT/CGANC

STEP 1: COMPUTE TARGET PARAMETERS WRT RADAR
  CALL TRNSFM
  CALL PVTRAN

STEP 1-2: COMPUTE TARGET C.G. ANGLE OFF-BORESIGHT (NON-SCANNING).
  CGANG = ACOS((-ROU(3))

STEP 1-3: DETERMINE TARGET CROSS-SECTION.
  SIGMA = TGTSIG

STEP 2: PRELIMINARY DETECTION MODE DETERMINATION

STEP 2-1: DETERMINE WHETHER ACTIVE OR PASSIVE.
```

A-23
IF(IMODE.EQ.1) GO TO 5

C

* STEP 2-2: GPC MODES OR AUTO/MANUAL MODES *
IF(IASM.GE.3) GO TO 10
GO TO 15

C

* STEP 3: ACTIVE MODE DETECTION PROCESS *

5 CALL SINGLE
RETURN

C

* STEP 4: PASSIVE AUTO/MANUAL MODE DETECTION PROCESS *

STEP 4-1: CHECK SHORT RANGE FIRST — CALL SINGLE-HIT DETECTION MODEL.
10 CALL SINGLE

C

STEP 4-2: CHECK FOR SUCCESS IN SINGLE-HIT DETECTION — IF NOT SUCCESSFUL, THEN TRY LONG RANGE SEARCH.
IF(MTP.EQ.0) CALL CFAR
RETURN

C

* STEP 5: PASSIVE GPC MODES DETECTION PROCESS *

STEP 5-1: CHECK DESIGNATED RANGE.
15 IF(EDRNG.GT.2552.) GO TO 20

C

STEP 5-2: IF DESIGNATED RANGE < 0.42 NM — USE SINGLE-HIT DETECTION MODEL.
CALL SINGLE
RETURN

C

STEP 5-3: IF DESIGNATED RANGE > 0.42 NM — USE CFAR DETECTION MODEL.
20 CALL CFAR
RETURN

END

SUBROUTINE DISCRM
REAL LATE,MEAN
COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SRPTE,
2 SRRT,E,SSS,MADV,F,MADV,F,MADV,F
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
COMMON /CNTL/I3DUM(14),MRNG,WSAM,MPRF,IDUMA(18)
COMMON /SYSDAT/TG,DR(3),CP,PS,PSBIAS,PSBIAS,ABBIAS,BTBIAS,GP,GA,
2 DUMS(3)
COMMON /TGRD/NT,DUM(586),CGRNGE,CGVEL
COMMON /DISCRM/AZDISC,ELDISC,RDISC,VDISC,DRTE,ODISC,SIGBAR,SNRD,
2 SIGBAR
COMMON /SIGDAT/SB,SMAZ,SP,SMEL,EARLY,LATE,DF1,DF5,
2 DF2,DF4,SIGBAR

SUBROUTINE DISCRM
REAL LATE,MEAN
COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SRPTE,
2 SRRT,E,SSS,MADV,F,MADV,F,MADV,F
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
COMMON /CNTL/I3DUM(14),MRNG,WSAM,MPRF,IDUMA(18)
COMMON /SYSDAT/TG,DR(3),CP,PS,PSBIAS,PSBIAS,ABBIAS,BTBIAS,GP,GA,
2 DUMS(3)
COMMON /TGRD/NT,DUM(586),CGRNGE,CGVEL
COMMON /DISCRM/AZDISC,ELDISC,RDISC,VDISC,DRTE,ODISC,SIGBAR,SNRD,
2 SIGBAR
COMMON /SIGDAT/SB,SMAZ,SP,SMEL,EARLY,LATE,DF1,DF5,
2 DF2,DF4,SIGBAR
STEP 1: COMPUTE CONSTANT USED IN SIGNAL SCALING AND COMPUTATION

NOTE: DEBUGGING PRINT STATEMENTS.

WRITE(6,900) SPAZ,SMAZ,SPEL,SMEL, EARLY, LATE
WRITE(6,901) DF1,DF5,DF2,DF4,SIGBAR

900 FORMAT(' SPZ,SMZ,SPL,SML,E,L -' ,6F10.2)
901 FORMAT(' DF1,DF5,DF2,DF4,SIG -=',5F10.2)

STEP 1-1: COMPUTE CONSTANT (NOTE: IT IS DIFFERENT FOR ACTIVE AND PASSIVE MODES).

IF(IMODE.EQ.2) GO TO 5
YY = GA*PS(MRNG,IMODE)/(CRNGE,E2,BN(MSAM))
SI = YY/FLOAT(NFREQ(IMODE))
GO TO 10

NOTE: THIS IS THE CONSTANT USED IN PASSIVE MODE.

YY = GP*PS(MRNG,IMODE)*PFTIX /(CRNGE*E2+BN(MSAM))
SI = YY/FLOAT(NFREQ(IMODE))
GO TO 10

NOTE: THIS IS THE CONSTANT USED IN PASSIVE MODE.

STEP 1-2: COMPUTE PEAK SIGNAL POWER TO AVERAGE THERMAL NOISE POWER AT DOPPLER FILTER OUTPUT.

SNRDT = YY*SIGBAR
WRITE(6,221) YY,SIGBAR
221 FORMAT('YY,SIGBAR -',F14.5)
SNRDTD = 10.*ALOG10(SNRDT)
SIGDB = 10.*ALOG10(SIGBAR)
SIGBR = SIGBAR
WRITE(6,990) SNRDT, SIGDB
990 FORMAT(' SNRDT,SIGDB -',2F14.2)

STEP 1-3: COMPUTE PEAK SIGNAL POWER TO TOTAL (THERMAL PLUS QUANTIZATION) NOISE POWER AT THE DOPPLER FILTER OUTPUT.

CALL SATNSE(SNF)
XX = SNF*AGCO
XX = XX/(XX+ONV)
S1 = S1*XX
YY = YY*XX
SNRD = YY*SIGBAR
SNRD = 10.*ALOG10(SNRD)

STEP 1-4: UPDATE NOISE SEQUENCE.
NN(1)=MOD(NN(1)+1,328)+1

DO 15 I=2,18

15 NN(I)=MOD(NN(I-1)+29,328)+1

IDI=NN(1)

GAUS(ID1)=ANORM(NS1,NS2)

STEP 2-1: CHECK ANTENNA STEERING MODE — SKIP STEP 2 IF IN

GPC-DES OR MANUAL.

CCCCCCCCCCCCCCCCC MOD FEB 16 1983 CCCCCCCCCCCCCCCCCCCCCCCCCCCC

IF(IASM.EQ.2.0R.IASM.EQ.4)

GO TO 28

C

C STEP 2-2: COMPUTE ANGLE DISCRIMINANT COMPONENT SCALE FACTOR.

ASCALE=S1*PDIA(IMODE)

C

C STEP 2-3: COMPUTE STATISTICS OF ADDITIVE NOISE FOR ANGLE

DISCRIMINANT COMPONENTS.

MEAN=PDIA(IMODE)

VARPAD=SQR(2. *S1*SPPAZ+1.)

VARMAD=SQR(2. *S1*SMADZ+1.)

VAREL= SQR(2. *S1*SMADL+1.)

VARMEL=SQR(2. *S1*SMEDL+1.)

STEP 2-4: ADD EQUIVALENT NOISE TO ANGLE DISCRIMINANT COMPONENT

SIGNALS.

ID6=NN(6)

SPAZ=ABS(ASCALE*SPAZ+MEAN+VARPAZ*GAUSS(ID1))

SMADZ=ABS(ASCALE*SMADZ+MEAN+VARMADZ*GAUSS(ID6))

ID2=NN(2)

ID7=NN(7)

SPEL=ABS(ASCALE*SPEL+MEAN+VAREL*GAUSS(ID2))

SMEL=ABS(ASCALE*SMEL+MEAN+VARMEL*GAUSS(ID7))

STEP 2-5: COMPUTE AZ AND EL DISCRIMINANT COMPONENTS.

AZDISC=10. *ALOG10(SPAZ/SMAZ)

ELDISC=10. *ALOG10(SPEL/SMEL)

STEP 3: COMPUTE RANGE DISCRIMINANT (INCLUDES NOISE)

STEP 3-1: COMPUTE RANGE DISCRIMINANT COMPONENT SCALE FACTOR.

RSCALE=S1*PDIR(IMODE)

STEP 3-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR RANGE

DISCRIMINANT.

MEAN=PDIR(IMODE)

YREL=SQR(2. *S1*YEARLY+1.)*TCON

YRLTE=SQR(2. *S1*YRLATE+1.)*TCON

STEP 3-3: ADD EQUIVALENT NOISE TO RANGE DISCRIMINANT COMPONENT

SIGNALS.

ID3=NN(3)

ID8=NN(8)

YEARLY=ABS(RSCALE*YEARLY+MEAN+YREL*GAUSS(ID3))

YRLATE=ABS(RSCALE*YRLATE+MEAN+YRLTE*GAUSS(ID8))

STEP 3-4: COMPUTE RANGE DISCRIMINANT.

RDISC=10. *ALOG10(LATE/EARLY)
C
C **************************************************************************
C * STEP 4: COMPUTE VELOCITY DISCRIMINANT (INCLUDES NOISE) *
C **************************************************************************
C STEP 4-1: COMPUTE VELOCITY DISCRIMINANT COMPONENT SCALE FACTOR.
VSCALE=IMODE/PDIV(IMODE)
C STEP 4-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR VELOCITY
C DISCRIMINANT COMPONENTS.
MEAN=PDIV(IMODE)
VARDF2=SQRT(2.*S1+DF4+1.)
VARDF4=SQRT(2.*S1+DF4+1.)
C STEP 4-3: ADD EQUIVALENT NOISE TO VELOCITY DISCRIMINANT
C COMPONENT SIGNALS.
ID4=NN(4)
ID9=NN(9)
DF2=ABS(VSCALE+DF2+MEAN+VARDF2+GAUSS(ID4))
DF4=ABS(VSCALE+DF4+MEAN+VARDF4+GAUSS(ID9))
C STEP 4-4: COMPUTE VELOCITY DISCRIMINANT.
VDISC=10.*ALOG10(DF2/DF4)
C **************************************************************************
C **************************************************************************
C * STEP 5: COMPUTE ON-TARGET DISCRIMINANT — USED FOR BREAK- *
C * TRACK AND VELOCITY DATA INVALID DETERMINATION *
C **************************************************************************
C STEP 5-1: COMPUTE STATISTICS OF ADDITIVE NOISE FOR OUTER DOPPLER
C FILTER SIGNALS.
VARDF1=SQRT(2.*S1+DF1+1.)
VARDF5=SQRT(2.*S1+DF5+1.)
C STEP 5-2: ADD EQUIVALENT NOISE TO OUTER DOPPLER FILTER SIGNALS.
ID5=NN(5)
ID10=NN(10)
DF1=ABS(VSCALE+DF1+MEAN+VARDF1+GAUSS(ID5))
DF5=ABS(VSCALE+DF5+MEAN+VARDF5+GAUSS(ID10))
C STEP 5-3: COMPUTE ON-TARGET DISCRIMINANT.
C NOTE: THE FACTOR OF SQRT(2.) IS DUE TO THE METHOD OF
C NORMALIZATION OF DISCRIMINANT COMPONENTS.
ODISC=10.*ALOG10((EARLY+LATE)+SQRT(2.)/(DF1+DF5))
C NOTE: DEBUGGING PRINT STATEMENTS.
C WRITE(6,902) AZDISC,ELDISC,RDISC,VDISC,ODISC
C WRITE(6,903) SNRD,SIGDB,SIGBAR
C WRITE(6,904) SPAZ,SMZ,SEPZ,SMEL,EARLY,LATE
C 902 FORMAT( ' AZD,ELD,RD,VD,OD =',3F14.6)
C 903 FORMAT(' SNRD,SIGDB,SIGBAR =',3F14.6)
C 904 FORMAT(' SPZ,SMZ,SEPZ,SMEL,E,L+NOISE =',6F10.2)
C 905 FORMAT(' DF1,DF5,DF2,DF4,SIG+NOISE =',6F10.2)
C RETURN
C END
C **************************************************************************
C THIS FUNCTION COMPUTES THE DOPPLER FILTER OUTPUT AMPLITUDE 
C AND PHASE FOR AN INPUT SIGNAL OF FREQUENCY X.
C **************************************************************************
COMPLEX FUNCTION DOPFIL(X)
COMPLEX DENOM, NUMER
DENOM = 1.0 - CEXP(CMPLX(0., X))
DENOM = 16. * DENOM

C CHECK FOR DENOMINATOR EQUAL TO ZERO.
XX = CABS(DENOM)
IF (XX .GT. 1.0E-06) GO TO 10
DOPFIL = (1.0, 0.0)
RETURN
10 NUMER = 1.0 - CEXP(CMPLX(0.16., X))
DOPFIL = NUMER / DENOM
RETURN
END

THIS FUNCTION GIVES THE ANTENNA DIFFERENCE PATTERN WEIGHTING OF THE RADAR SIGNAL FOR THE GIVEN ANGLE (IN RADIANS) OFF BORESIGHT.

NOTE: THIS PATTERN IS THE DERIVATIVE OF THE SUM PATTERN

FUNCTION DPAT(X)
IF (ABS(X) .GT. 1.0E-4) GO TO 10
DPAT = 0.6228 * X
RETURN
Y = 93.80 * X
DPAT = 1.1465 * (Y * COS(Y) - SIN(Y)) / (Y * Y)
RETURN
END

EXECUTIVE PROGRAM: INTERFACE WITH PARENT SIMULATION

SUBROUTINE EXEC
COMMON /CNTL/IPR, IMODE, ITXPI, IASM, IDUMC(5), DUMC(3)
COMMON /OUTPUT/MSW, MTF, MSF, DUM(7), IDUM2(4)
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KMUP, IDUM3(1), MTP, IDUM5(17)
DATA DATINT/I.e/
2 0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0
0 0 0 3 1 2 0

PREPARE TO RECEIVE THE PRODUCT YOU'VE BEEN WAITING FOR...
THE ACTUAL SES SPACE SHUTTLE RADAR SIMULATION...
**Step 1: Check System Power Switch**

IF(IPWR.GT.1) GO TO 5

**Step 2: Check System Mode Switch**

IF(IMODE.LT.3) GO TO 7

**Step 3: Determine Whether System in Standby**

IF(IPWR.GT.2) GO TO 15

**Step 4: Determine Whether Warmup Period Exceeded**

15 IF(KMSCLK.GT.KMWUP) GO TO 20

**Step 5: Determine If There Has Been an Antenna Steering Mode Change**

20 IF(IASM.EQ.IOLDSM) GO TO 25

**Step 6: Initialize All Target and System Data**

IF(DATINT.NE.1.0) GO TO 1

CALL SETIT
CALL DATA
CALL SYSINT
IOLDPW= IF(R
DATINT,,=e.B
II=1
IF(II.EQ.1) GO TO 30

**Step 1: Check System Power Switch**

IF(IPWR.GT.1) GO TO 5

**Step 2: Check System Mode Switch**

IF(IMODE.EQ.IOLDMD) GO TO le

**Step 3: Determine Whether System in Standby**

IF(IPWR.GT.2) GO TO 15

**Step 4: Determine Whether Warmup Period Exceeded**

15 IF(KMSCLK.GT.KMWUP) GO TO 20

**Step 5: Determine If There Has Been an Antenna Steering Mode Change**

20 IF(IASM.EQ.IOLDSM) GO TO 25

**Step 6: Initialize All Target and System Data**

IF(DATINT.NE.1.0) GO TO 1

CALL SETIT
CALL DATA
CALL SYSINT
IOLDPW= IF(R
DATINT,,=e.B
II=1
IF(II.EQ.1) GO TO 30
**STEP 5: DETERMINE WHETHER SYSTEM IS IN SEARCH AND ACQUISITION OR TRACK MODE.**

IF(MTF.EQ.0 .OR. MTP.EQ.1) GO TO 30

C IF TRACK FLAG DOWN — GO TO SEARCH MODE.
CALL SEARCH
RETURN
C IF TRACK FLAG IS UP — GO TO TRACK MODE.
30 CALL TRACK
RETURN
END

* THIS SUBROUTINE Generates A (3X3) MATRIX TGA THAT Produces A ROTATION OF GA Radians ABOUT THE Y-AXIS. *

SUBROUTINE GAMMA(TGA,GA)
DIMENSION TGA(3,3)
DO 10 I=1,3
DO 10 J=1,3
TGA(1,J)=0.0
TGA(2,2)=1.0
TGA(1,1)=COS(GA)
TGA(1,3)=-SIN(GA)
TGA(3,3)=TGA(1,1)
TGA(3,1)=TGA(1,3)
RETURN
END

* THIS FUNCTION CHECKS FOR NEGATIVE ARGUMENT FOR INT FUNCTION AND CORRECTS THE QUANTIZATION PROCEDURE. *

REAL FUNCTION INTT(Y)
X=Y
IF(X.LT.0.0) X=X-1.0
INTT=INT(X)
RETURN
END

* THIS SUBROUTINE Multiplies THE (3X3) MATRIX A AND THE (3X1) VECTOR B TO OBTAIN THE (3X1) VECTOR C. *

SUBROUTINE MULT31(A,B,C)
DIMENSION A(3,3).B(3).C(3)
DO 10 I=1,3
C(I)=0.0
DO 10 J=1,3
C(I) = C(I)+A(I,J)*B(J)
RETURN
END
THIS SUBROUTINE MULTIPLIES THE (3X3) MATRIX A AND THE (3X3) MATRIX B TO OBTAIN THE (3X3) MATRIX C.

```fortran
SUBROUTINE MULT33(A,B,C)
DIMENSION A(3,3), B(3,3), C(3,3)
DO 10 I=1,3
    DO 20 J=1,3
        C(I,J)=0.0
    20 CONTINUE
    DO 30 K=1,3
        C(I,J)=C(I,J)+A(I,K)*B(K,J)
    30 CONTINUE
10 CONTINUE
RETURN
END
```

THIS SUBROUTINE GENERATES A (3X3) MATRIX TPH THAT PRODUCES A ROTATION OF PH RADIANS ABOUT THE Z-AXIS.

```fortran
SUBROUTINE PHI(TPH,PH)
DIMENSION TPH(3,3)
DO 10 I=1,3
    DO 20 J=1,3
        TPH(I,J)=0.0
    20 CONTINUE
    TPH(1,1)=COS(PH)
    TPH(2,2)=TPH(1,1)
    TPH(1,2)=SIN(PH)
    TPH(2,1)=-TPH(1,2)
10 CONTINUE
RETURN
END
```

THIS SUBROUTINE GENERATES A (3X3) MATRIX TPHD THAT REPRESENTS THE DERIVATIVE OF A MATRIX THAT REPRESENTS UNIFORM ROTATION ABOUT THE Z-AXIS. THE ROTATION SPEED IS W AND THE ANGLE AT WHICH THE DERIV. IS TAKEN IS PH.

```fortran
SUBROUTINE PHID(TPHD,PH,W)
DIMENSION TPHD(3,3)
DO 10 I=1,3
    TPHD(3,I)=0.0
    TPHD(1,1)=W*SIN(PH)
    TPHD(2,2)=TPHD(1,1)
    TPHD(1,2)=W*COS(PH)
    TPHD(2,1)=-TPHD(1,2)
10 CONTINUE
RETURN
END
```

THIS SUBROUTINE UPDATES THE POSITION OF THE ANTENNA GIMBALS.

```fortran
```

[Page A-31]
SUBROUTINE POINT
COMMON /OUTPUT/DUM1(3),DUM4(2),SPANG,SRANG,DUM5(3),IDUM2(4)
COMMON /SYSDAT/TSUM(3),CG,SG,DUM2(9)
COMMON /ATDAT/DUM1(4),SALRTE,SBTRTE,DUM3(2),AL,BT,PREF,RREF,
2 AREF,BREF
DATA AK/2.e/,TAU/1.414/,PI/3.141592653/
C
C * STEP 1: PRELIMINARY COMPUTATIONS *
C
C CR=\cos(-RREF)
SR=\sin(-RREF)
CP=\cos(-PREF)
SP=\sin(-PREF)
C
C * STEP 2: COMPUTE ANTENNA REFERENCE ROLL/PITCH ANGLES IN THE 
C RADAR FRAME.
C
C XX=CG*SP-SG*SR=CP
YY=SG*SP'-CG=SR
ZZ=CR*CP
IF(YY.EQ.0.0.AND.ZZ.EQ.0.0) GO TO 1
AREF=ATAN2(YY,ZZ)
GO TO 2
1 IF(XX.GT.0.0) AREF=PI/2.
IF(XX.LT.0.0) AREF=PI/2.
2 BREF=ASIN(XX)
C
C *STEP 3: UPDATE OUTER (ALPHA) GIMBAL RATE AND POSITION
C COMPUTE ALPHA LOOP POSITION ERROR.
ERRA=AREF-AL
C UPDATE SMOOTHED ALPHA GIMBAL RATE ESTIMATE.
SBTRTE=SBTRTE+TS*AK*ERRA
C UPDATE ALPHA GIMBAL RATE.
ALRATE=AK=TAU*ERRA+SBTRTE
C CHECK FOR ALPHA GIMBAL RATE LIMITING.
IF(ABS(ALRATE).GT.56.) AREF=56.*ALRATE/ABS(ALRATE)
C UPDATE ALPHA GIMBAL POSITION.
AL=AL+TS*ALRATE
C
C *STEP 4: UPDATE INNER (BETA) GIMBAL RATE AND POSITION
C COMPUTE BETA LOOP POSITION ERROR.
ERRB=BREF-BT
C UPDATE SMOOTHED BETA GIMBAL RATE ESTIMATE.
SBTRTE=SBTRTE+TS*AK*ERRB
C UPDATE BETA GIMBAL RATE.
BTRATE=AK=TAU*ERRB+SBTRTE
C CHECK FOR BETA GIMBAL RATE LIMITING.
IF(ABS(BTRATE).GT.56.) BTRATE=56.*BTRATE/ABS(BTRATE)
C UPDATE BETA GIMBAL POSITION.
BT=BT+TS*BTRATE
C
C *STEP 5: ANTENNA IN OBSCURATION REGION *
C CALL SCNWRN
C
C *STEP 6: COMPUTE ANTENNA ROLL/PITCH ANGLES IN THE BODY FRAME *
C
A-32
THIS SUBROUTINE COMPUTES TARGET C.G. POSITION AND VELOCITY WRT ANTENNA LOS COORDINATES.

**Subroutine PVTRAN**

```fortran
C
COMMON /TEST1/RA(3)
COMMON /CNTL/IP#R,IK#R,DE
COMMON /INPUT/ERT(3),EVT(3),DUM(21)
COMMON /OUTPUT/MSWF,MTF,MSF,DUMO(7),IDUMO(4)
COMMON /ICNTL/IDUM6(9),MTP,IDUM7(3),MTKINT
COMMON /SYSDAT/TSAM,DR(3),DUM2(11)
COMMON /TGTDAT/NT,RAU(3,1e6),RANGE(Ie7),RADVEL(1e6),RO(3),2ROU(3),CGRNGE,CGVEL
COMMON /SATDAT/RADAR(3),N2e,RT(7e,3),SIG(Te),ROLD,ICLOSE,ICLOLD
COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
COMMON /TARGET/ITARG,SRCS
DIMENSION ROR(3),ROD(3),VI(3),RL(3),RAD(3),RLD(3),XRT(3)

**STEP 1:** COMPUTE TARGET C.G. POSITION IN ORBITER BODY FRAME.

**STEP 1-1:** ADD RADAR OFFSET IN ORBITER BODY FRAME.

```
C • STEP 2: COMPUTE TARGET C.G. RADIAL VELOCITY WRT ANTENNA LOS FRAME (OR RADAR).
C *********************************************************
C C STEP 2-1: COMPUTE TARGET C.G. VELOCITY COMPONENTS WRT ANTENNA LOS FRAME.
C CALL MUTLT31(TLBG,ROR,V1)
C CALL MUTLT31(TLBG,EVT,ROD)
DO 15 I=1,3
15 ROD(I)=ROD(I)+V1(I)
C C STEP 2-2: COMPUTE TARGET C.G. RADIAL VELOCITY WRT ANTENNA LOS.
CGVEL=0.0
DO 20 I=1,3
20 CGVEL=CGVEL+ROD(I)*ROU(I)
C *********************************************************
C C STEP 3: COMPUTE TARGET SCATTERING CHARACTERISTICS: ILLUMINATED POINTS, THE POINT LOCATIONS, AND THE RCS FOR EACH POINT.
C *********************************************************
C C STEP 3-1: IF IN ACTIVE MODE, SEARCH MODE, OR TRACKER INITIALIZATION ASSUME SINGLE SCATTERER LOCATED AT TARGET FRAME ORIGIN.
C ITARG=0 POINT TARGET
C ITARG=1 SPAS
C ITARG=2 SMM
C IF(ITARG.EQ.0) GO TO 24
C CHECK CONDITION.
C IF(MODE.NE.1.AND.WTKINT.NE.0.AND.MTP.NE.0) GO TO 30
C IF ABOVE CONDITION TRUE — THEN SET PARAMETERS AS FOLLOWS AND DO NOT CALL TARGET MODEL.
C 24 NT=I
C SIG(I)=SRCS
DO 25 I=1,3
25 RT(I)=0.0
DO 25 I=1,3
25 RT(I)=RT(I)+TT(J,I)=RT(K,I)
C C STEP 3-2: COMPUTE LOCATION OF RADAR IN TARGET FRAME.
30 DO 35 I=1,3
35 RADAR(I)=RADAR(I)-TLT(J,I)*ROU(J)
C IF(ITARG.EQ.0)GO TO 40
C C STEP 3-3: COMPUTE TARGET SCATTERING CHARACTERISTICS.
C IF(ITARG.EQ.2)CALL SMM
C IF(ITARG.EQ.1)CALL SPAS
C NT=N20
C 40 DO 70 K=1,NT
C *********************************************************
C C • STEP 4: COMPUTE KTH SCATTERER POSITION, RANGE, AND DIRECTION VECTOR WRT ANTENNA LOS FRAME (OR RADAR).
C *********************************************************
C C STEP 4-1: COMPUTE KTH SCATTERER POSITION WRT ANTENNA LOS FRAME.
DO 45 J=1,3
45 RL(I)=0.0
DO 45 I=1,3
45 RL(I)=RL(I)+TLT(J,I)*RT(K,I)
DO 50 I=1,3
50 RA(I)=ROU(I)+RL(I)
STEP 4-2: COMPUTE RANGE OF KTH SCATTERER WRT RADAR.

\[
\text{RANGE}(K) = \sqrt{\text{RA}(1)^2 + \text{RA}(2)^2 + \text{RA}(3)^2 + \text{RA}(3)}
\]

STEP 4-3: COMPUTE UNIT VECTOR IN DIRECTION OF KTH SCATTERER WRT ANTENNA LOS FRAME.

\[
\text{RAU}(I,K) = \frac{\text{RA}(I)}{\text{RANGE}(K)}
\]

STEP 5-1: COMPUTE KTH SCATTERER VELOCITY COMPONENTS WRT ANTENNA LOS FRAME.

\[
\text{XRT}(I) = \text{RT}(K,I)
\]

CALL MULT31(TLTD, XRT, RLD)

DO 58 I=1,3

58 XRT(I) = XRT(I) + RLD(I)

DO 60 I=1,3

60 RAD(I) = RAD(I) + RLD(I)

STEP 5-2: COMPUTE KTH SCATTERER RADIAL VELOCITY WRT TO RADAR.

\[
\text{RADVEL}(K) = \text{RADVEL}(K) + \text{RAD}(I) \cdot \text{RAU}(I,K)
\]

70 CONTINUE

NOTE: DEBUGGING PRINT STATEMENTS.

WRITE(6, 900) RO(1), RO(2), RO(3), CGRANGE, CGVEL
WRITE(6, 901) RAU(1,1), RAU(2,1), RAU(3,1), RANGE(1), RADVEL(1)
WRITE(6, 902) RO(1), RO(2), RO(3), CGRANGE, CGVEL
WRITE(6, 903) RO(1), RO(2), RO(3), CGRANGE, CGVEL

900 FORMAT(//' RO1, RO2, RO3, CGRANGE, CGVEL ')
901 FORMAT(//' RAU1, RAU2, RAU3, RADVEL ')
902 FORMAT(//' SPAS RCS DATA: ')
903 FORMAT(//' SPAS RCS DATA: ')
RETURN
END

FUNCTION: THIS FUNCTION GENERATES A RANDOM NUMBER FROM A UNIFORM DISTRIBUTION.

FUNCTION RNDU(IRAN)
DATA MU/524287/, ETA/997/
IF(IRAN.EQ.0) GO TO 10
IRAN = ETA + IRAN
IKEEP = IRAN + MU
IRAN = IRAN + IKEEP + MU
XRAN = IRAN
XRAN = XRAN + MU
RNDU = XRAN
10 RETURN
END

FUNCTION: THIS SUBROUTINE COMPUTES THE RADAR SIGNAL STRENGTH AND UPDATES THE AGC SETTING.
SUBROUTINE RSS
COMMON /CNTL/IPWR,IMODE,DM1(7),DM1(3)
COMMON /CNTL/IPWR,IMODE,DM1(7),DM1(3)
COMMON /OUTPUT/IDUM7(3),DUM3(S),SRSS,DUM4(4)
COMMON /AGCDAT/AGCC,AGCD(14),MRNG,SNRDT
DIMENSION PS(Ie,2)
DATA PS/9*1.,2.,5*1.,2.,4.,B.,8.,IB./
C
C     STP 1: UPDATE SYSTEM AGC
C
C     STP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
AGCERR=4.*PS(MRNG,IMODE)/(AGCO*SNRDT+1.0)+QNV
IF(AGCERR.GT.10.) AGCERR=10.0
IF(AGCERR.LT.0.1) AGCERR=0.1
C
C     STP 1-2: COMPUTE NEW AGC VALUE AND CHECK LIMITS.
AGCO=AGCC+AGCD
IF(AGCO.GT.1.0E15) AGCO=1.0E15
AGCD=10.*ALOG10(AGCO)
C
********* 2: UPDATE RADAR SIGNAL STRENGTH VALUE *********
C
IF(AGCO.LT.1.0E-15) AGCO=1.0E-15
SRSS=1./AGCO
SRSS=10.*ALOG10(SRSS)
RETURN
END
C
********** THIS SUBROUTINE UPDATES RANGE AND RANGE RATE ESTIMATES. **********
C
SUBROUTINE RTRACK
REAL INTT,IRDISC,IRNG,IRDOT
COMMON /CNTL/IPWR,IMODE,DM1(7),DM1(3)
COMMON /OUTPUT/IDUM7(3),SRNG,SNRDT,DM2(5),DM1(4)
COMMON /OUTPUT/IDUM1(14),MRNG,MSAM,MPRF,DM1(10),MPFOLD
COMMON /SYSDAT/TSAM,DUM(S)
COMMON /RTDAT/IRDOT,IRNG,RGBIAS,VEST(4),MDF(5)
COMMON /DSCRM/DM2(2),DSC,RISC,RKTE,ODISC,DM3(3)
DIMENSION RTI(10,2),RT2(10,2),TDC(3),RGBIAS(2)
DATA RT1/9.0,125.0,25.6+0.125,2.1,0.25,0.5,8.25,0.5,0.5.
DATA RT2/9.0,125.0,25.6+0.125,2.1,0.25,0.5,8.25,0.5,0.5.
DATA TDC/0.05122118,0.1195161,0.2561557/
DATA RGBIAS/32.3,94.7
C
********** 1: UPDATE ROUGH RANGE RATE ESTIMATE **********
C
C     STP 1-1: INTEGERIZE RANGE DISCRIMINANT AND CHECK FOR SATURATION.
RDISC=INT(IRDISC+TCON+0.5)/TCON
C
C     STP 1-2: COMPUTE ROUGH RANGE RATE PREDICTION FROM ALPHA-BETA TRACKING EQUATIONS.
C
A-36
DEFINITION: RTI(MRNG,IMODE) CORRESPONDS TO BETA IN ALPHA-BETA TRACK.

STEP 2: UPDATE RANGE ESTIMATE *

STEP 2-1: UPDATE RANGE ESTIMATE USING ALPHA-BETA TRACKER EQUATIONS.

STEP 2-2: CONVERT RANGE ESTIMATE (IRNG) TO FEET USING THE FACT THAT THE LSB OF IRNG REPRESENTS 5/16 FEET.

STEP 2-3: ADD FIXED BIAS TO FINAL RANGE ESTIMATE.

FORCE BREAK TRACK IF RANGE LESS THAN 100 FT

RETURN

END

SUBROUTINE SATNSE(SNF)

SUBROUTINE SCAN

A-37
DIMENSION TIMINT(31), ANGINT(31), RSW(10), TSW(10)

DATA TIMINT/7.1, 4.2, 6.3, 4.4, 3.5, 4.6, 7.8, 9.1, 10.4, 11.8, 0.0/0012820
1 13.4, 13.0, 14.2, 15.5, 15.6, 16.8, 18.9, 21.1, 22.2, 23.4, 24.5, 0012840
2 34.2, 33.5, 28.6, 21.1, 14.9, 11.8, 9.9, 4.3, 6.8, 0.0/
DATA ANGINT/0.0, 1.2, 2.7, 3.2, 4.5, 5.2, 6.1, 7.9, 8.8, 9.1, 10.9, 0.0/0012860
1 11.9, 13.0, 14.2, 15.5, 15.6, 16.8, 18.9, 21.1, 22.2, 23.4, 24.5, 0012870
2 25.6, 27.2, 28.9, 21.1, 14.9, 11.8, 9.9, 4.3, 6.8, 0.0/
DATA TSW/60.0, 54.3, 43.2, 33.5, 28.6, 21.1, 14.9, 11.8, 9.9, 4.3, 6.8, 0/.0012890
2 RSW/48609.2, 55982.6, 62584.3, 71698.6, 91142.5, 151938.6, 0012900
3 243464.0, 394949.8, 581941.8, 1822845.6/0012910

PI = 180.0 / 3.141592653

C C C C C C C

C C C C C

C * STEP 1: DETERMINE WHETHER TO PERFORM SCAN INITIALIZATION (MSF=0) OR SCAN UPDATE (MSF=1).
C
C **********************************************************************************************
C
C IF(MSF.EQ.1) GO TO 15
C
C **********************************************************************************************
C
C * STEP 2: PERFORM SCAN INITIALIZATION
C
C INITIALIZE ALL FLAGS.
MSF=1
C INITIALIZE RING MONITORS.
IAROLD=0
ITROLD=10
C INITIALIZE SCAN CLOCK.
KSNCLK=0
C INITIALIZE SCAN TIME PARAMETER.
KSN=0
C
C DETERMINE SWITCH POINT PARAMETER.
DO 5 I=1,10
IF(EDRNG.LT.RSW(I)) GO TO 10
5 CONTINUE
10 MSWITCH=1
C
C **********************************************************************************************
C
C * STEP 3: UPDATE SCAN CLOCKS
C
C STEP 3-1: UPDATE SCAN CLOCK (TRACKS TOTAL ELAPSED TIME FROM SCAN INITIATION).
15 KSNCLK=KSNCLK+1
T=FLOAT(KSNCLK)*TSAM
C
C STEP 3-2: UPDATE SCAN TIME PARAMETER (USED TO DETERMINE BORESIGHT POSITION IN SCAN PATTERN).
IF(T LE.TSW(MSWITCH)) KSN=KSN+1
IF(T GT.TSW(MSWITCH)) KSN=KSN-1
TSN=FLOAT(KSN)*TSAM
C
C **********************************************************************************************
C
C * STEP 4: DETERMINE ANTENNA POSITION TO NEAREST SCAN RING
C
C **********************************************************************************************
C
C DO 20 I=1,31
IF(TSN.LT.TIMINT(I)) GO TO 25
20 CONTINUE
25 IARNG=1
C
C **********************************************************************************************
C
C * STEP 5: DETERMINE TARGET POSITION IN SCAN PATTERN (SCAN RING NUMBER FOR TARGET)
C
C **********************************************************************************************
C
C A-38
C

C STEP 5-1: DETERMINE TARGET POSITION EXACTLY.
ALOLD=AL
BTOLD=BT
AL=AREF
BT=BREF
CALL TRNSFM
CALL PVTRAN
AL=ALOLD
BT=BTOLD

C STEP 5-2: DETERMINE TARGET SCAN RING NUMBER.

C DETERMINE TARGET ANGLE OFF SCAN DESIGNATES (DEGREES).

CGANG=ACOS(-ROU(3))/PI

C

C STEP 6-1: CHECK CONDITION.
IF(IARNG.EQ.ITRNG.AND.IAROLD.NE.ITROLD) CALL DETECT

C STEP 6-2: UPDATE RING NUMBER MONITOR.
IAROLD=IARNG
ITROLD=ITRNG

C

C STEP 7-1: CHECK ALL POSSIBLE TERMINATION CONDITIONS.

C CONDITION 1: T > 60. SECONDS"
IF(T.GE.60.) GO TO 40

C CONDITION 2: NEXT SCAN TIME PARAMETER < 0."
ITEMP=KN-1
IF(ITEMP.LT.0) GO TO 40

C CONDITION 3: DETECT A TARGET"
IF(MTP.EQ.0) RETURN

C STEP 7-2: PERFORM SCAN TERMINATION STEPS --- IF TERMINATION COND
ITION OBTAINED.

MSF=0
KSNCLK=0
KSN=0
ISRCHG=0
ISRCHC=0
RETURN
END

C

C THIS SUBROUTINE DETERMINES WHETHER THE ANTENNA IS IN THE OB-
• SCURATION ZONE AND SETS THE SCAN WARNING FLAG APPROPRIATELY.

SUBROUTINE SCNWRN
COMMON /OUTPUT/MSWF, IDUMO(2), DUMO(7), IDUMO1(4)
COMMON /ATDAT/DUM(8), A, B, DUMA(4)
DIMENSION ICLEAR(36.72)
DATA ICLEAR/17=1.13=e.6=1.1B=1.12=e.6=1.1Bel.12*e.6=1.
1 18.1.12=e.6=1.19,1.11=0.6=1.19=1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.19,1.11=0.6=1.

* THIS SUBROUTINE COMPUTES THE RESPONSE TO ALL DISPLAYS AND CONTROLS WHEN THE RADAR IS IN ANY OF THE SEARCH MODES.

SUBROUTINE SEARCH
COMMON /CNTL/IDUM(3),IASM,ISRCHC,ISRCHG,IAZS,IELS,ISLR,EDRNG, EDPA,EDRA
COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE, SRRTE,SRSS, DUM(8)
COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,K_P,KSNCLK, KSNMAX,KACCLK,MTP,MZ1,MZ2,MSS,MTKINT,MRNG,MSAM,MPRF, IDUMI(16)
COMMON /SYSDAT/TS,DUMS(14)
COMMON /ATDAT/DUM(16),PREF,REF,DUMA(2)
DIMENSION SLWRTE(2)
DATA SLWRTE/6.9814E-3,3.49e7E-1/

* DETERMINE ANTENNA STEERING MODE.

GO TO (10,20,30,40), IASM
**STEP 4: PERFORM SCAN SEQUENCE**

10 IF(MSF.EQ.1) GO TO 14
   IF(MZ1.EQ.1.AND.ISRCHG.EQ.1) GO TO 14

**STEP 2: PERFORM GIMBAL POINTING SEQUENCE**

**STEP 2-1: UPDATE ROLL/PITCH REFERENCES**

IF(ISHOLD.EQ.1.AND.ISRCHG.EQ.1) GO TO 12
   RREF=EDRA
   PREF=EDPA
12 ISHOLD=ISRCHG

**STEP 2-2: UPDATE POSITION OF GIMBALS**

CALL POINT

**STEP 2-3: DETERMINE WHETHER BORESIGHT IN ZONE 1 AND/OR ZONE 0 AND TAKE APPROPRIATE ACTION.**

CALL ZONECK

IF NOT IN ZONE 0, THEN DETECTION IS NOT ALLOWED.

**STEP 3: CHECK FOR TARGET DETECTION — IF IN ZONE 0**

CALL DETECT
RETURN

**STEP 4: PERFORM SCAN SEQUENCE**

14 CALL SCAN
RETURN

**STEP 1: PERFORM GIMBAL POINTING SEQUENCE**

**STEP 1-1: UPDATE ROLL/PITCH REFERENCE ANGLES.**

20 PREF=EDPA
   RREF=EDRA

**STEP 1-2: UPDATE POSITION OF GIMBALS.**

CALL POINT

**STEP 1-3: DETERMINE WHETHER BORESIGHT IN ZONE 1 AND/OR ZONE 0 AND TAKE APPROPRIATE ACTION.**

CALL ZONECK

IF BORESIGHT NOT IN ZONE 0, THEN TARGET DETECTION NOT ALLOWED.
IF(W20.EQ.0) RETURN

C
C • STEP 2: CHECK FOR TARGET DETECTION — IF IN ZONE 0 •
C
C CALL DETECT
C RETURN
C
C • STEP 2: SCAN SEQUENCE •
C
C CALL SCAN
C RETURN

C
C • STEP 1: DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN •
C
C 30 IF(ISRCHC.EQ.1) GO TO 32
C
C • STEP 2: PERFORM GIMBAL POINTING SEQUENCE •
C
C • STEP 2: UPDATE ROLL/PITCH REFERENCE ANGLES.
PREF=PREF+FLOAT(ELS)+SLWRTE(ISLR+1)*TS
RREF=RREF+FLOAT(IAZS)+SLWRTE(ISLR+1)*TS
C
C • STEP 2-2: UPDATE POSITION OF GIMBALS.
CALL POINT
C
C • STEP 2-3: DETERMINE SLEW RATE AND TAKE APPROPRIATE ACTION.
C IF SLEW RATE IS GREATER THAN 0.4 DEG/SEC, THEN TARGET DET-00006230
C IF(ISLR.GT.0) RETURN
C
C • STEP 3: CHECK FOR TARGET DETECTION — IF SLEW RATE < 0.4 DEG PER SECOND •
C
C CALL DETECT
C RETURN
C
C • STEP 4: PERFORM SCAN SEQUENCE •
C
C 32 CALL SCAN
C RETURN
C
C • MANUAL SEARCH AND ACQUISITION Mode •
C
C • STEP 1: UPDATE ANTENNA POSITION •
C
C • STEP 1: UPDATE ROLL/PITCH REFERENCE ANGLES.
40 PREF=PREF+FLOAT(ELS)+SLWRTE(ISLR+1)*TS
RREF=RREF+FLOAT(IAZS)+SLWRTE(ISLR+1)*TS
C
C • STEP 1: UPDATE POSITION OF GIMBALS.
CALL POINT
STEP 1-3: DETERMINE SLEW RATE AND TAKE APPROPRIATE ACTION.

IF SLEW RATE IS GREATER THAN 0.4 DEG/SEC, THEN TARGET DETECTION IS NOT ALLOWED.

IF(ISLR.GT.0) RETURN

* STEP 2: CHECK FOR TARGET DETECTION — IF SLEW RATE <0.4 DEG PER SECOND. *

CALL DETECT
RETURN
END

* THIS SUBROUTINE GENERATES THE NOISE-FREE ANGLE, RANGE, VELOCITY AND ON-TARGET DISCRIMINANT COMPONENTS. *

SUBROUTINE SIGNAL
REAL IRDOT, IRNG
COMMON /CNTL/IPWR,IMOOI,ITXP,IASM,IDUMC(5),DUMC(3)
COMMON /OUTPUT/IIDLIM(3),SRNG,DUMI(6),IDUM2(4)
COMMON /INT/MTKINT,MRNG,MSAM,MPRF,MBKTRK,MBTSUM,
2 MBT(8)
COMMON /TGTDAT/NT,RAU(3,100),RANGE(100),RADVEL(100),RO(3)
COMMON /RTDAT/IRDOT,IRNG,DUM2(5),MDF(5)
COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
2 SIGBAR
COMMON /XFORMS/TI.B(3,3),TLBD(3,3),TLTD(3,3),XFORM(3,3)
COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,
2 CDF2,CDF4,CDiff
DIMENSION CTP(10,2),DF'WTS(5,100),ALAM(3),ALAMD(3),NFREQ(2)
DATA CTP/9=.O3318,9.799E--4,4_.O3318,1.9599E-3,9.8E-4,
2 2.45E-4,.225E-4/
DATA NFREQ/1,5/,ALAM/177.3733,176.0447,178.7149,176.7089,
2 178.0393/,ALAMD/1.272461E-2,2.969888E-2,3.309023E-1/
REAL LAT

* STEP 1: PRELIMINARY COMPUTATIONS AND PARAMETER INITIALIZATION *

* STEP 1-1: INITIALIZE DISCRIMINANT COMPONENTS (NOTE: THESE ARE THE COMPONENT SIGNALS AFTER SQUARE-LAW DETECTION).

SPAZ=0.0
SMAZ=0.0
SPEL=0.0
SMEL=0.0
EARLY=0.0
LATE=0.0
DF1=0.0
DF2=0.0
DF3=0.0
SIGBAR=0.0

NFMAX=NFREQ(1MODE)
DO 55 i=1,NFMAX

A-43
C STEP 1-2: INITIALIZE COMPLEX DISCRIMINANT COMPONENTS BEFORE EACH XMIT FREQUENCY (NOTE: THESE ARE THE COMPONENT SIGNALS BEFORE SQUARE-LAW DETECTION).

CSUM=(0.0. )
CDIFAZ=(0.0. )
CDIFEL=(0.0. )
CEARLY=(0.0. )
CLATE=(0.0. )
CDF1=(0.0. )
CDF2=(0.0. )
CDr_=(0.0. )

DO 45 K=1,NT

C IF (I.GT.1) GO TO 35

C C****************************
C * STEP 2: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR FOR KTH SCATTERER
C *
C C****************************
C
C STEP 2-1: COMPUTE SUM PATTERN ANGLE.
PSI=ACOS(ABS(RAU(3,K)))

C STEP 2-2: COMPUTE ANTENNA SUM PATTERN MULTIPLICATION FACTOR.
X=SPAT(PSI)

C STEP 2-3: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR.
XX=SIG(K)*X

C NOTE: IF IN ACTIVE MODE SET XX=1.0.
IF(IMODE.EQ.1) XX=1.0
S=XX*X

C STEP 2-4: CHECK ANTENNA STEERING MODE (IF IN GPC-DES OR MANUAL MOD
--- SKIP STEP 4).
IF(IASM.EQ.2 OR IASM.EQ.4) GO TO 2e

C C****************************
C * STEP 3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION FACTORS FOR KTH SCATTERER
C *
C C****************************
C
C STEP 3-1: COMPUTE AZ AND EL DIFFERENCE PATTERN ANGLES.
DELAZ=ASIN(RAU(2,K))
DELEL=ASIN(RAU(1,K))

C STEP 3-2: COMPUTE AZ AND EL DIFFERENCE PATTERN MULTIPLICATION FACTORS.
Y=DPAT(DELAZ)
Z=DPAT(DELEL)

C STEP 3-3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION FACTORS (INCLUDE RCS AND SUM PATTERN WEIGHTINGS).
DAZ=XX*Y
DEL=XX*Z

C C****************************
C * STEP 4: COMPUTE RANGE GATE WEIGHTING FOR KTH SCATTERER
C C DEFINITION: CTP=4./(C*PULSEWIDTH) WHERE C IS SPEED OF LIGHT.
C
C STEP 4-1: COMPUTE RANGE GATE LOCATION WRT RANGE GATE CENTER.
SRNGX=10.*AINT(0.03125*IRNG)
DELX=CTP(MRNG,IMODE)*(RANGE(K)-SRNGX)

C

STEP 4-2: COMPUTE EARLY AND LATE RANGE GATE WEIGHTINGS FOR KTH SCATTERER.
II=INT((DELX+7.)/2.)
IF(II.LE.1) II=1
IF(II.GE.5) II=5
GO TO (21,22,23,24,21),II

21 RGE=0.0
RGL=0.0
GO TO 25

22 RGE=3.+DELX
RGL=0.0
GO TO 25

23 RGE=1.-DELX
RGL=1.+DELX
GO TO 25

24 RGE=0.0
RGL=3.-DELX

C

STEP 4-3: COMPUTE RANGE GATE WEIGHT FOR NON-RANGE DISCRIMINANT COMPONENTS.
25 RGWGT=0.5*(RGL+RGE)

C

STEP 4-4: APPLY RANGE GATE WEIGHTING TO SUM AND DIFFERENCE CHANNEL MULTIPLICATION FACTORS.
RGE=S*RGE
RGL=S*RGL
S=S*RGWGT
DAZ=DAZ+RGWGT
DEL=DEL*RGWGT

C

STEP 5: COMPUTE DOPPLER FILTER PHASE SHIFT AND WEIGHTING FOR KTH SCATTERER. NOTE: THIS
CALCULATION IS INDEPENDENT OF XMIT FREQUENCY AND ASSUMES NO ACCELERATION OVER DATA CYCLE.

DEFINITION: ALAMD(MPRF)=2.*PI/(PRF*LAMBDA)
DEFINITION: THE CONSTANT 0.196348=PI/16.

STEP 5-2: COMPUTE DOPPLER FREQUENCY CORRESPONDING TO RADIAL VELOCITY OF KTH SCATTERER.
FDT=2.*ALAMD(MPRF)-RADVEL(K)

STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER TRACKING FILTERS.
DO 30 J=1,5.
ARG=0.196348+MDF(J)-FDT
30 DFWTs(J,K)=OOPFL(ARG)

C

STEP 6: COMPUTE PHASE FACTOR ASSOCIATED WITH KTH SCATTERER RANGE (NOTE: PHASE IS REFERENCED TO PHASE ASSOCIATED WITH RANGE OF TARGET C.G.)
DELPsi=ALAMD(I)*(RANGE(K)-CGRNGE)

C

DEFINITION: RANGE(K) IS RANGE OF KTH SCATTERER TO ANTENNA PHASE CENTER
DEFINITION: ALAMD=2.*PI/LAMBDA WHERE LAMBDA IS XMIT FREQUENCY.

C

STEP 6-1: COMPUTE PHASE REFERENCED TO TARGET C.G.
35 DELPSI=ALAMD(I)*(RANGE(K)-CGRNGE)
C STEP 6-2: COMPUTE PHASE FACTOR, I.E. EXP(J•DELPHI).
   PHASE=CEXP(COMPLEX(@,DELPSI))
   PHASE1=PHASE
C
C STEP 6-3: COMBINE RANGE PHASE FACTOR AND DOPPLER FILTER =3
   WEIGHT AND PHASE FACTOR.
   PHASE=PHASE+DFWTS(3,K)
C
C STEP 7: ADD (VECTORIALLY) KTH SCATTERER CONTRIBUTION TO EACH
   DISCRIMINANT'S COMPONENT SIGNALS.
C
C STEP 7-1: ADD KTH SCATTERER CONTRIBUTION TO SUM CHANNEL SIGNAL.
   CSUM=CSUM+PHASE
C
C STEP 7-2: CHECK ANTENNA STEERING MODE —— SKIP STEP 8-3 IF IN
   GPC-DES OR MANUAL MODE.
   IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 48
C
C STEP 7-3: ADD KTH SCATTERER CONTRIBUTION TO AZ AND EL DIFFERENCE
   CHANNELS SIGNALS.
   CDFAZ=CDFAZ+DAZ•PHASE
   CDFE=CDFE+DEL•PHASE
C
C STEP 7-4: ADD KTH SCATTERER CONTRIBUTION TO RANGE DISCRIMINANT
   COMPONENT SIGNALS.
   40 EARLY=EARLY+RGE•PHASE
   CLATE=CLATE+RGL•PHASE
C
C STEP 7-5: ADD KTH SCATTERER CONTRIBUTION TO VELOCITY DISCRIMINANT
   COMPONENT SIGNALS.
   PHASE1=PHASE1+S
   CDF2=CDF2+PHASE1+DFWTS(2,K)
   CDF4=CDF4+PHASE1+DFWTS(4,K)
C
C STEP 7-6: ADD KTH SCATTERER CONTRIBUTION TO ON-TARGET DISCRIMINANT
   COMPONENT SIGNALS.
   CDF1=CDF1+PHASE1+DFWTS(1,K)
   CDF5=CDF5+PHASE1+DFWTS(5,K)
45 CONTINUE
C
C STEP 8: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET
   DISCRIMINANT COMPONENTS AT ITH FREQUENCY AND SQUARE
   LAW DETECT THESE COMPONENTS.
C
C STEP 8-1: CHECK ANTENNA STEERING MODE —— SKIP STEPS 9-2 AND 9-3
   IF IN GPC-DES OR MANUAL.
   IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 58
C
C STEP 8-2: COMPUTE AZ DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
   SPAZ=SPAZ+RABS(CSUM+CDFAZ)**2
   SMAZ=SMAZ+RABS(CSUM+CDFAZ)**2
C
C STEP 8-3: COMPUTE EL DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
   SPEL=SPEL+RABS(CSUM+CDFEL)**2
   SMEL=SMEL+RABS(CSUM+CDFEL)**2
C
C STEP 8-4: COMPUTE RANGE DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT
   EARLY=EARLY+RABS(CEARLY)**2
   LATE=LATE+RABS(CLATE)**2
50 CONTINUE
C STEP 8-5: COMPUTE VELOCITY DISCRIMINANT COMPONENTS AND SQUARE-LAW
DETECT.
DF2=DF2+CABS(CDF2)**2
DF4=DF4+CABS(CDF4)**2
C STEP 8-6: COMPUTE ON-TARGET DISCRIMINANT COMPONENTS AND SQUARE-LAW
DETECT.
DF1=DF1+CABS(CDF1)**2
DF5=DF5+CABS(CDF5)**2
C ...
C STEP 9: COMPUTE EFFECTIVE CROSS-SECTION AVERAGED OVER PROPER
NUMBER OF TRANSMIT FREQUENCIES.
SIGBAR=SIGBAR+CABS(CSUM)**2
55 CONTINUE
SIGBAR=SIGBAR/FLOAT(NFREQ(IMOOE))
RETURN
END
C ...
C SUBROUTINE SINGLE
DIMENSION P(_I)
COMMON /CNTL/IPTR,IMOOE,ITXP,IASM,IDUM(5),DUMC(3)
COMMON /OUTPUT/MSWF,MTF,_F,D_(7),ID_I(4)
COMMON /ICNTL/IDUM2(8),KACCLK,MTP,IDUM3(5),MSAM,IDUM4(11)
COMMON /TGTDAT/NT,DUMI(S),RO(3),ROU(3),CGRNGE,CGVEL
COMMON /DETDAT/SIGMA,CGANG
DATA NSRCH/18S/
C ...
C STEP 1: COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT
C ...
C STEP 1-1: SET SAMPLE RATE TO OBTAIN CORRECT NOISE BW IN SNRV COMP.
MSAM=1
IF (IMODE.EQ.1) MSAM=2
C ...
C STEP 1-2: COMPUTE NOMINAL SNRV.
SNR=SNRV(SIGMA,CGANG)
C ...
C STEP 2: IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNRV
C ...
C STEP 2-1: CHECK SCAN FLAG.
IF(MSF.EQ.1) GO TO 1
STEP 2-2: COMPUTE BEAMSHAPE LOSS — BASED UPON C.G. POSITION OFF BORESIGHT.
\[
\text{BETA}_2 = \text{SPAT(CGANG)} \times 2
\]
STEP 2-3: ADD BEAMSHAPE LOSS TO NOMINAL SNR, I.E. COMPUTE ACTUAL SNR
\[
\text{SNR} = \text{SNR} \times \text{BETA}_2
\]
STEP 3-1: DETERMINE INDEX TO ACCESS APPROPRIATE PD VERSUS SNR CURVE.
1 IF(IWDUE.EQ.2) GO TO 5
   NCRV = 1
   GO TO 15
5 IF(IA.M. LT.3) GO TO 10
   NCRV = 3
   GO TO 15
10 NCRV = 5
ADJUST INDEX FOR SCANNING.
15 NCRV = NCRV + MSF
STEP 3-2: CONVERT SNR TO DB.
   IF (SNR.LT.1.E-8) GO TO 20
   SNR = 10**ALOG10(SNR)
   GO TO 25
20 SNR = 100.
STEP 3-3: SNR OUTSIDE (-30 DB, 0 DB) INTERVAL” — IF SO, SET OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.
   IF SNR < -25 DB THEN SET PD=0.0 (DECLARE A MISS).
   IF(SNR.GT.-5.0) GO TO 35
   IF(SNR.LT.-25.) GO TO 30
   IF(SNR.GT.-5.0) GO TO 35
STEP 3-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR INTERPOLATION.
   SCALE = (SNR+25.)**2 + 1.000001
   ISNR = INT(SCALE)
   REMAIN = SCALE - FLOAT(ISNR)
STEP 3-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION.
   PROB = P(ISNR) + REMAIN*(P(ISNR+1) - P(ISNR))
   IF X.RNDU(HSRC) GO TO 35
STEP 4: DETERMINE OUTCOME OF DETECTION ATTEMPT.
   X = RNDU(HSRC)
   IF (X.LE.PROB) GO TO 35
STEP 5: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT.
   IF NO DETECTION — SET TARGET PRESENT FLAG LOW.
   MTP = 0
RETURN

STEP 5-2: IF DETECTION SUCCESSFUL — SET TARGET PRESENT FLAG HIGH AND INITIALIZE ACQUISITION CLOCK.

35 MTP=1
KACCLK=0
RETURN
END

FUNCTION SNRV(SIGMA,RANGE)
COMMON/CNTL/IF_IR,IMOOE,ITXP,IDUMC(6),DUMC(3)
COMleON
/ICNTL/IDUM(12),MSS,MTKINT,WRNG,MSAM,MPRF,IDUM2(10)
COMMON/SYSDAT/DUM(12),TGTSIG,GPS,GAS

DIMENSION PT(4),BN(2)

CCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCC
DATA PT/46.3,54.9,23.,6.2/,
BN/69.9,57.9/

C DETERMINE WHETHER ACTIVE OR PASSIVE MODE

IF(IMODE.EQ.1) GO TO 10

C PASSIVE MODE VIDEO SNR CALCULATION

SNRV=GPS+PT(ITXP)+10.*ALOG10(SIGMA)-BN(MSAM)-40.*ALOG10(RANGE)
RETURN

C ACTIVE MODE VIDEO SNR CALCULATION

SNRV=GAS-20.*ALOG10(RANGE)
SNRV=10.**(0.1*SNRV)
RETURN
END

MODIFIED FOR LENS

THIS SUBROUTINE MODELS THE SPAS SPACECRAFT SCATTERING PROPERTIES.

SES SPAS MODEL AS OF JULY 7, 1981.
DATA DEFINITION: INCLUDES SCATTERER LOCATION IN TARGET FRAME, MAXIMUM SCATTERER RCS VALUE, ANGULAR EXTENT OF NONZERO RCS, AND OTHER MISCELLANEOUS DATA REQUIRED BY THE ROUTINE.

SEED FOR RANDOM NUMBER GENERATOR
DATA KSEED/45,678,908,587,5678,897,345,7777,67,4.
1 568,899,444,886,999,555,222,70,80,8000,
2 5,15,25,35,45,55,65,75,85,95,
3 7,17,27,37,47,57,67,77,87,97,
4 987,984,666,2398,76,412,759,409,899,561,
5 265,3895,9457,9643,937,656,453,980,567,2154,
6 801,86,99,31,86,106,4,9,3,987,
7 888,999/

DATA DESCRIPTION DIMENSIONS OF WIDE-ANGLE SCATTERERS
C DEFINITION: DIM = 2D/LAMBDA (UNITLESS)
DATA DIM/72,64,8/
C DEFINITION: WSCALE = SQRT(DIM^2/(4*PI)) (UNITS=FEET, NF = OF FREQ)
DATA WSCALE/72,0,2965/

FOR EACH DIFFUSE SCATTERER, SPECIFY NORMAL COMPONENT
DATA NORMAL/10=1,2,12=3/

COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
DATA TARG/4,12,6=-7.8=-35,37,4=-35,37,3=24,2=37.
2 66.3=35.3=12.3=3.5=-35.4=37.6=24.6=7=0,8,
3 1.75=1.05=-1.75=35.1=75=1.05=35.-1.05=-1.75=2.15,
4 -2.15,1.75=1.05=35.-35.-1.05=-1.75=35.1.05=35.-35
5 -1.75=35.-83=-1.05=-1.27=1.05=-35.3=1.05=1.9=-1.05
6 1.05=3=-1.05=3.5=1.75=1.05=35.-35=-1.75=2.15=2=-35,
7 2=34.2=1.05=2=1.27=1.75=1.05=35.-1.05=-1.75=0,0,
8 12=0.7=48.5=48.3=15.3=0.3=8.3=0.3=67=-86,
9 4=-48.4=255=425=425=-425=-425=-425=-3=-82=-82=3.,
A 6=0.0=2.38/

MINIMUM SUBTENDED ANGLE
DATA PHIMIN/4=0.0=90.14=0.0=16=0.0=48.5.4=88.0=6=0.0,
2 6=177.9,0,
3 11=0.98=12=0.58=35.30=0.45=0.3=0.10=0.4=0.177.4,
4 89.7=0.4=88.5=4=88.0=12=-8.48,
5 19=0.5=90.3=85.9=3=88.5=156..90..87.3=88.5.2=87.4=-0,
6 90.-4=178.5=0.0=178.0=178.0=90.0=0.0=0.0=0.0=0.0=6=88.5,
7 48.0/

MAXIMUM SUBTENDED ANGLE
DATA PHIMAX/4=90..2=90..5=90..2=1.3=180..3=2.1.4=180.,
2 4=91.5,4=92.6=90.6=180..48.,
3 10=180..90.13=180.4=156.155..135..2=180..145..3=180.,
4 2.6=180..90.3=180..4=91.5,4=92.6=180..6=180..138.,
5 12=180..7=96..5=180..3=94.1.3=91.5.180..156..92.3=91.5,2=92.6,
6 125..6=180..2=180..2=2=180..90..180..90..90..90..6=91.5,138./

RADII OF THE SCATTERERS (FEET)
00323580 00323590

A-50
DATA OFFSET /24*.3*4.1.2.29.0.2.35.3155.0.24.35.8*0.0.24*1.6*0.8.0.0/
C MISCELLANEOUS DATA.
DATA NTAR/61/,KWIDE/24/,PI/3.141592653/
DATA TTRAN/3*0.0/,INIT1/1/
C
C***********************************************************************
C* STEP 0: TRANSLATE POINT TARGETS BY TARGET FRAME OFFSET (TTRAN) *
C***********************************************************************
C IF(INIT1.NE.1) GO TO 2
C
C RANDOMIZE DIFFUSE SCATTERER RCS VALUES.
C ISEED=100
DO 107 I=1,1000
 X=RNDU(ISEED)
 DO 108 I=1,KWIDE
 X=RNDU(ISEED)
 CHANCE MADE 9-11-81
108 SIGMA(I)=SIGMA(I)+(X*.005)-0.0025
C CONVERT TARGET DATA APPROPRIATELY.
C FTM=0.3048
DO 101 I=1,NTAR
 SIGMA(I)=SORT(SIGMA(I))/FTM
 DO 102 J=1,NTAR
 TARG(J,I)=TARG(J,I)/FTM
 DO 103 J=1,NTAR
 PHIMIN(J,I)=COS(PHIMIN(J,I)*PI/180.)
103 PHIMAX(J,I)=COS(PHIMAX(J,I)*PI/180.)
 DO 105 I=1,NTAR
 OFFSET(I)=OFFSET(I)/FTM
C DO 1 K=1,NTAR
 DO 1 I=1,3
1 TARG(K,I)=TARG(K,I)+TTRAN(I)
INIT1=0
C
C***********************************************************************
C* STEP 1: DETERMINE WHICH SCATTERER ARE ILLUMINATED AND HAVE A *
C* NONZERO RCS IN THE DIRECTION OF THE RADAR. *
C***********************************************************************
C STEP 1-1: PERFORM REQUIRED INITIALIZATIONS.
2 CONTINUE
 NWIDTH=0
 KTARGET=0
C STEP 1-2: COMPUTE UNIT VECTOR IN DIRECTION OF RADAR FOR
I TH SCATTERING CENTER.
 DO 15 I=1,NTAR
 DO 5 J=1,3
 VECT(J)=RADAR(J)-TARG(I,J)
5 CONTINUE
 V NORM=SORT(VECT(1)**2+VECT(2)**2+VECT(3)**2)
 DO 10 J=1,3
 IF(ABS(VECT(J)).GT.0.5) WRITE(6,*) 'VECT GREATER THAN VNORM'
 COSPHI(J)=VECT(J)/VNORM
C STEP 1-3: DETERMINE WHETHER ITH SCATTERER HAS A NONZERO RCS IN THE

C DIRECTION OF THE RADAR.
 2 GO TO 15
10 CONTINUE
C
STEP 1-4: IF ITH SCATTERER RCS IS NONZERO THEN ADD TO VECTOR OF
ILLUMINATED SCATTERERS.
   KTAR=KTAR+1
   JHOT(KTAR)=I
   SIG(KTAR)=SIGMA(I)
   IF(I.LE.KWIDE) NWIDE=NWIDE+1
15 CONTINUE
C
• STEP 2: COMPUTE LOCATION OF SPECULAR POINTS THAT ARE ILLUMINATED
•
  DO 20 K=I,KTAR
     I=JHOT(K)
   DO 20 J=1,3
   R(K,J)=TARG(I,J)+OFFSET(I)*COSPHI(I,J)
20 CONTINUE
C
• STEP 3: COMPUTE SQUARE ROOT OF RCS FOR ALL ILLUMINATED WIDE
• ANGLE SCATTERERS (REPRESENTING DIFFUSE SCATTERING AREAS).
•
  DO 22 K=1,NWIDE
     I=JHOT(K)
   IO=NOR_L(I)
22 SIG(K)=SQRT(ABS(COSPHI(I,IO )))*SIGMA(I)
C
• STEP 4: CHECK FOR SHORT RANGE CONDITION
•
  STEP 4-1: DETERMINE RANGE TO RADAR IN TARGET FRAME.
   24 RANGE=SQRT(RADAR(1),,2+RADAR(2),,2+RADAR(3),,2)
   STEP 4-2: SET HYSTERESIS LOOP MONITORING VARIABLE.
   IF((ROLD.LT.81.OR.RANGE-ROLD.LE.0.).AND.RANGE.LE.270.) ICLOSE=1
   IF(RANGE-ROLD.GT.81.) CLOSE=-ICLOSE
  STEP 4-3: CHECK MONITORING VARIABLE TO DETERMINE IF SHORT RANGE
CONDITION EXISTS.
   IF(ICLOSE.EQ.0.OR.NWIDE.EQ.0) GO TO 55
C
• STEP 5: PROCEDURE FOR UPDATING OF DIFFUSE SCATTERING CENTER LOCATION — SHORT RANGE CONDITION ONLY.
•
  STEP 5-1: IF FIRST TIME THRU — PERFORM INITIALIZATION OF
DIFFERENCE EQUATIONS FOR ALL DIFFUSE SCATTERERS.
   IF(ICLOLD.EQ.1) GO TO 35
   DO 30 J=1,KWIDE
      IO=NOR_L(I)
      PHIOLD(I)=ACOS(COSPHI(I,IO ))
   DO 25 J=1,3
   V(I,J)=VOLD(I,J)+V(I,J)
25 CONTINUE
30 CONTINUE
GO TO 55

C STEP 5-2: UPDATE ANGULAR INCREMENT FOR EACH DIFFUSE SCATTERER
—— CHANGE IN ANGLE FROM SAMPLE-TO-SAMPLE.
35 DO 40 I=1,KWIDE
   IQ=正常(I)
   PHI(I,IQ)=acos(cospHI(I,IQ))
   DPHI(I)=(PHI(I,IQ)-PHIOLD(I))
   PHIOLD(I)=PHI(I,IQ)
40 CONTINUE

C STEP 5-3: UPDATE SCATTERER LOCATION FOR ALL ILLUMINATED DIFFUSE
SCATTERER — UPDATE DIFFERENCE EQUATIONS.
DO 50 K=1,NWIDE
   I=JHOT(K)
   DO 45 J=1,3
      IQ=NORMAL(I)
      IF(J.EQ.IQ) GO TO 45
      ALPH(I,J)=exp(-DIM(I,J)*abs(DPHI(I,COSPHI(I,IQ)))
      WRAN(I,J)=sqrt(1.-ALPH(I,J)**2)*Wscale(I,J)*(RNDU(KSEED(I,J)).-5)
      V(I,J)=ALPH(I,J)*VOLD(I,J)+WRAN(I,J)
      VOLD(I,J)=V(I,J)
      R(K,J)=R(K,J)+V(I,J)
45 CONTINUE
50 CONTINUE
55 CONTINUE

C **************************************************************************
C STEP 6: UPDATE PARAMETERS USED TO MONITOR TARGET POSITION
C • ON SHORT RANGE HYSTERESIS CURVE.
C **************************************************************************
C ROLD=Range
ICOLD=ICLOSE

WRITE(6,908) KTAR,NWIDE,ICLOSE,ROLD
908 FORMAT(11'TT',WT,IC,R =',3IB,F12.4)

C **************************************************************************
C NOTE: THE FOLLOWING STATEMENTS ARE PRINT STATEMENTS USED IN THE
C DEBUGGING PROCESS.
C **************************************************************************
C NOTE: DEBUGGING PRINT STATEMENTS.
C PRINT LOCATION OF RADAR IN TARGET FRAME.
WRITE(6,900) RADAR

C PRINT TABULAR LISTING OF ALL DATA ASSOCIATED WITH SPAS SCATTERERS.
WRITE(6,901) I.SIGMA(I),TARG(I,1),TARG(I,2),TARG(I,3),OFFSET(I)
8 ,PHIMIN(I,1),PHIMAX(I,1),PHIMIN(I,2),PHIMAX(I,2),PHIMIN(I,3),PHIMAX(I,3),
2 I=1,NTAR)

C PRINT TOTAL = OF SCATTERERS AND = OF DIFFUSE SCATTERERS.
WRITE(6,902) KTAR,NWIDE

C PRINT INFORMATION ASSOCIATED WITH ILLUMINATED SCATTERERS.
WRITE(6,903)
C WRITE(6,904) (I,JHOT(I),SIG(I),(R(I,J),J=1,3),
1 I=1,KTAR)
C
C PRINT DATA ASSOCIATED WITH DIFFUSE SCATTERER DIFFERENCE EQUATION.
WRITE(6,905)I,PHIOLD(I),
1 (V(I,L),L=1,3),(R(I,L),L=1,3)
I=1
WRITE(6,906)I,PHI(I),PHIOLD(I),DPHI(I)
WRITE(6,907)K,I,(VOLD(I,J),J=1,3),(ALPH(I,J),J=1,3),
1 (WRAN(I,J),J=1,3),(V(I,J),J=1,3),(R(I,J),J=1,3)
ALL
PRINT FORMAT STATEMENTS.
FORMAT(' IN FEET, RADAR = (',FS.1,',',F8.1,',',FS.1,')')
FORMAT(I12,F15.3,2(SX,3F7.3))
RETURN
END

FUNCTION SPAT(X)
NOTE: THE FOLLOWING VALUE OF B GIVES THE SUM PATTERN A SINGLE-SIDED
3 DB BEAMWIDTH OF 0.85 DEGREES.
Y=93.80*X
TEMP=ABS(Y)
IF(TEMP.GT.1.0E-06) GO TO 10
SPAT=1.0
RETURN
10 SPAT=SIN(Y)/Y
RETURN
END

FUNCTION SUM(X,N)
Y=SIN(X)**2
IF(Y.GT.1.0E-08) GO TO 10
SUM=N
RETURN
10 SUM=SIN(N*X)**2/(N*Y)
RETURN
END

SUBROUTINE SYSINT

** THIS FUNCTION GIVES THE ANTENNA SUM PATTERN WEIGHTING OF THE **
** RADAR SIGNAL FOR THE GIVEN ANGLE (IN RADIANS) OFF BORESIGHT **

FUNCTION SPAT(X)

NOTE: THE FOLLOWING VALUE OF B GIVES THE SUM PATTERN A SINGLE-SIDED
3 DB BEAMWIDTH OF 0.85 DEGREES.
Y=93.80*X
TEMP=ABS(Y)
IF(TEMP.GT.1.0E-06) GO TO 10
SPAT=1.0
RETURN
10 SPAT=SIN(Y)/Y
RETURN
END

FUNCTION SUM(X,N)
Y=SIN(X)**2
IF(Y.GT.1.0E-08) GO TO 10
SUM=N
RETURN
10 SUM=SIN(N*X)**2/(N*Y)
RETURN
END

** THIS SUBROUTINE RESETS THE SYSTEM UNDER THE FOLLOWING CONDITIONS **
** (1) BREAK-TRACK (TO SEARCH), (2) PASSIVE/ACTIVE MODE CHANGE (TO **
** SEARCH), AND (3) SYSTEM IN STANDBY (TO IDLE).**

SUBROUTINE SYSINT
COMMON /CNTL/IPWR, IMODE, ITYP, IASM, IDUMC(5), DUMC(3)
COMMON /OUTPUT/MSWF, MTF, MSF, SRNG, SRDOT, SPANG, SRANG, SRRTRE, SSRS, MADVF, MRDVDF, MARDVF, MRRDVF
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KWNUP, KSNCLK
COMMON /OUTPUT/MSWF, MTF, MSF, SRNG, SRDOT, SPANG, SRANG, SRRTRE
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KWNUP, KSNCLK, KSNMAX, KACCLK, MTP, MZ1, MZ0, MSS, MTKINT, MRNG, MSAM, MPRF
COMMON /ATDAT/ DUM1(4), ALRATE, STRATE, DUM2(2), AL, BT, PREF, RREF

C ••• STEP 1: INITIALIZE ALL INTERNAL FLAGS AND CONTROLS •••

IOLDMD= IMODE
IOLDSM= IASM
ISHOLD=0
MTP=1
MZ1=0
MZ0=0
MSS=0
MTKINT=0

C ••• STEP 2: INITIALIZE ALL INTERNAL CLOCKS •••

KACCLK=0
KSNCLK=0

C ••• STEP 3: INITIALIZE ALL DISPLAY FLAGS •••

MSWF=0
MSF=0
MTF=0
MADVF=0
MRDVDF=0
MARDVF=0
MRRDVF=0

C ••• STEP 4: INITIALIZE ALL DISPLAY METERS •••

SRNG=0.0
SRDOT=0.0
SPRTE=0.0
SRRTRE=0.0
SRSS=0.0

C ••• STEP 5: INITIALIZE GIMBAL POINTING LOOP •••

PII=3.14159265/180.
ALRATE=0.0
BTRATE=0.0
IF(IPWR.NE.1.AND.KMSCLK.NE.1) GO TO 5

C STEP 5-1: IF SYSTEM POWER OFF THEN ALIGN BORESIGHT WITH ZENITH.

PREF=0.0
RREF=0.0
AL=0.0
BT=0.0
SPANG=0.0
SRANG=0.0
IOLDPW=IPWR
RETURN

5 IF(IPWR.GT.2) GO TO 15

A-55
C STEP 5-2: IF SYSTEM IN STANDBY THEN HOLD GIMBALS AT POSITION WHEN
STANDBY ENTERED AND ZERO DISPLAYS.
IF(1OLDPW.EQ.IPW) GO TO 10
PREF=PII+SPANG
RREF=PII+SRANG
10 SPANG=0.0
SRANG=0.0
IOLDPW=IPWR
RETURN
C
C STEP 5-3: PREPARE GIMBAL LOOP FOR ENTRY INTO ANY OF SEARCH MODES.
15 PREF=PII+SPANG
RREF=PII+SRANG
IOLDPW=IPWR
RETURN
END

SUBROUTINE TGTACQ
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
COMMON /OUTPUT/MSWF,MTF,MSF,DUMI(7),MADVF,MRDVF,MARDVF,MRRDVF
COMMON /ICNTL/IDUM3(B),KACCLK,MTP,MZI,MZe,MSS,MTKINT,
COMMON /SYSDAT/TS,DUMS(14)
DIMENSION ADV(le,2),RDV(le,2),ARDV(le,2)
DATA ADV/9=l.e2,5.12,8=l.92,2,2.33/
DATA RDV/9*6.15,28.69,8*6.97,2=29.76/
DATA ARDV/9*8.2,28.69,7*8.2,26.23,2,29.76/
C
C • STEP 1: UPDATE ACQUISITION CLOCK *
KACCLK=KACCLK+1
ACCLK=ACCLK+TS
C
C • STEP 2: PERFORM ANGLE DATA VALID TEST — GPC-ACQ + AUTO ONLY *
IF((IASM.EQ.2.OR.IASM.EQ.4)) GO TO 10
IF((ACCLK.LT.ADV(MRNG,IMODE))) GO TO 10
MADVF=1
C
C • STEP 3: PERFORM RANGE AND RANGE RATE DATA VALID TEST *
10 IF((ACCLK.LT.RDV(MRNG,IMODE))) GO TO 15
MRDVF=1
MRRDVF=1
C
C IF GPC-DES OR MANUAL INITIALIZE RADAR TRACKING PARAMETERS.
C CCCCCCCCCCCCCCCCCCCCCC MOD MAR 24 1983 CCCCCCCCCCCCCCCCCCCCCC
15 IF((IASM.EQ.2.OR.IASM.EQ.4).AND.MRDVF.EQ.1) GO TO 20
C
C • STEP 4: PERFORM ANGLE RATE DATA VALID TEST — GPC-ACQ + AUTO *
C • MODES ONLY.
C •
C IF((ACCLK.LT.ARDV(MRNG,IMODE))) RETURN
A-56
MARDVF=1

* STEP 5: PERFORM STEADY STATE RADAR TRACKING INITIALIZATION *

20 KACCLK=0
MTF=1
RETURN

END

* THIS SUBROUTINE GENERATES A (3X3) MATRIX TTH THAT PRODUCES *
* A ROTATION OF TH RADIANS ABOUT THE X-AXIS. *

SUBROUTINE THETA(TTH,TH)
DIMENSION TTH(3,3)
DO 10 I=1,3
DO 10 J=1,3
TTH(I,J)=0.0
TTH(I,I)=1.0
TTH(2,2)=COS(TH)
TTH(3,3)=TTH(2,2)
TTH(2,3)=SIN(TH)
TTH(3,2)=-TTH(2,3)
RETURN
END

* THIS SUBROUTINE Initializes THE ANGLE TRACKING LOOPS, THE *
* RANGE TRACKING LOOP, AND THE VELOCITY PROCESSOR —- STEADY *
* STATE CONDITIONS ARE ASSUMED. *

SUBROUTINE TINIT
REAL INTT,IRNG,IRDOT,IVR
COMMON /CNTL/IPWR,IMODE,ITYP,IASM,IDUMC(5),DUMC(3)
COMMON /INPUT/ERT(3),EVT(3),EWB(3),DUM(18)
COMMON /OUTPUT/I3DUM(3),SRNG,DUMI(6),IDUMI(4)
COMMON /ICNTL/I1DUM(13),MTKINT,MPNG,MSAM,MPSF,MBKTRK,MBTSUM,
2 MBT(8),MPFOLD
COMMON /SYSDAT/TSAM,DR(3),CP,SP,PS1,PSBIAS,DUM2(7),TRB(3,3)
COMMON /TGDAT/NT,DUM5(50),RO(3),ROU(3),GGRNGE,GVEL
COMMON /SADAT/RADAR(3),KAR,RT(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,ALRAT,AL.BT.
2 DUM3(2)
COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
COMMON /XFORMS/TLB(3,3),TLB(3,3),TLT(3,3),TLT(3,3)
COMMOM /AGDAT/AGCO,AOCOD,SNRD,SNRTD
DIMENSION ER(3),EV(3),ERTO(3),FLTW(3,3),RI(10)
DATA FLTW/7.7215,3.3090,-0.2969/
DATA RI/120,240,780,2552,5772,11544,23089,43747,57722,1.8228E6,NRI/18,P1/3.141592653/

* STEP 0: INITIALIZE BREAK-TRACK ALGORITHM *

STEP 0--: INITIALIZE MOVING WINDOW-OF-8 REGISTERS.
DO 3 I=1,8
MBT(I) = 0

STEP 0-2: INITIALIZE SUM REGISTER.

MBSUM = 0

STEP 0-3: SET BREAK-TRACK FLAG TO LOW (OR 0) STATE.

MBTRK = 0

* STEP 1: INITIALIZE ANGLE TRACKING LOOP *

---------------------------------------------

IF (IASM.EQ.2 .OR. IASM.EQ.4) GO TO 5

STEP 1-1: COMPUTE INITIAL INNER AND OUTER GIMBAL POSITIONS.

(TRANSFORM CONSISTS OF TRANSLATION PLUS ROTATION.)

PERFORM TRANSLATION — SHIFT TO RADAR FRAME ORIGIN.

DO 1 I = 1, 3
1 ERT0(I) = ERT(I) - OR(I)

TRANSFORM TARGET POSITION FROM BODY TO RADAR FRAME.

CALL MULT31 (TRB, ERT0, ER)

TRANSFORM TARGET VELOCITY FROM BODY TO RADAR FRAME.

CALL MULT31 (TRB, EVT, EV)

S = SQRT (ER(2)**2 + ER(3)**2)

IF (ER(I).EQ.0.0 .AND. S = 0.0) STOP

BT = ATAN2 (ER(I), S)

ER2 = ER(2)

ER3 = ER(3)

AL = ATAN2 (ER2, ER3)

GO TO 9

8 IF (ER(I).GT.0.0) AL = PI/2.

IF (ER(I).LT.0.0) AL = -PI/2.

IF (ER(I).EQ.0.0) STOP

STEP 1-2: COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH AND ELEVATION RATES.

PRELIMINARY TRIGONOMETRIC COMPUTATIONS.

CA = COS(AL)

SA = SIN(AL)

CB = COS(BT)

SB = SIN(BT)

TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER GIMBAL(G) REFERENCE FRAME.

WGX = CA*EWB(1) + SP*EWB(2) + CP*EWB(3)

WGX = CA*(-SP*EWB(1) + CP*EWB(2)) + SA*EWB(3)

WGY = CA*(SP*EWB(1) + CP*EWB(2)) + SA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + SA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

COMPUTE THE RANGE TO TARGET.

R = SQRT (ER(I)**2 + ER2**2 + ER3**2)

COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).

AZRATE = VGY/RH + (CB*WGZ - SB*WGY)

AZRATE = VGY/RH + (CB*WGZ - SB*WGY)

R = SQRT (ER(I)**2 + ER2**2 + ER3**2)

COMPUTE INITIAL TARGET INERTIAL LOS ELEVATION RATE(ERATE).

ERATE = CB*EV(2) + CA*EV(3)

ERATE = CB*EV(2) + CA*EV(3)

STEP 1-3: COMPUTE INITIAL INNER AND OUTER GIMBAL RATES.

PRELIMINARY TRIGONOMETRIC COMPUTATIONS.

CA = COS(AL)

SA = SIN(AL)

CB = COS(BT)

SB = SIN(BT)

TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER GIMBAL(G) REFERENCE FRAME.

WGX = CA*EWB(1) + SP*EWB(2)

WGX = CA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + SA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + SA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

COMPUTE THE RANGE TO TARGET.

R = SQRT (ER(I)**2 + ER2**2 + ER3**2)

COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).

AZRATE = VGY/RH + (CB*WGZ - SB*WGY)

AZRATE = VGY/RH + (CB*WGZ - SB*WGY)

R = SQRT (ER(I)**2 + ER2**2 + ER3**2)

COMPUTE INITIAL TARGET INERTIAL LOS ELEVATION RATE(ERATE).

ERATE = CB*EV(2) + CA*EV(3)

ERATE = CB*EV(2) + CA*EV(3)

STEP 1-3: COMPUTE INITIAL INNER AND OUTER GIMBAL RATES.

PRELIMINARY TRIGONOMETRIC COMPUTATIONS.

CA = COS(AL)

SA = SIN(AL)

CB = COS(BT)

SB = SIN(BT)

TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER GIMBAL(G) REFERENCE FRAME.

WGX = CA*EWB(1) + SP*EWB(2)

WGX = CA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + SA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + SA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + SA*EWB(3)

WGY = CA*(-SP*EWB(1) + CP*EWB(2)) + SA*EWB(3)

WGB = SA*(SP*EWB(1) + CP*EWB(2)) + CA*EWB(3)
C COMPUTE INITIAL INNER GIMBAL RATE(BTRATE).
BTRATE=ELRATE-WGY

C *********************************************
C * STEP 2: INITIALIZE RANGE TRACKING LOOP *
C *********************************************

C STEP 2-1: TRANSFORM TARGET C.G. POSITION AND C.G. VELOCITY FROM
BODY TO ANTENNA LOS FRAME.
5 CALL TRNSFM
CALL PVTRAN

C STEP 2-2: INITIALIZE THE RANGE ESTIMATE REGISTER.
SRNG=CGRNGE
IRNG=INTT(SRNG+3.2+0.5)

C STEP 2-3: INITIALIZE THE RANGE RATE ESTIMATE REGISTER.
IRDRT=INTT(CGVEL+TSAM+3.2+0.5)

C *********************************************
C * STEP 3: SET OPERATING PARAMETERS BASED UPON INITIAL RANGE *
C AND SYSTEM MODE. *
C *********************************************

C STEP 3-1: DETERMINE CORRECT RANGE INTERVAL.
DO 30 I=1,NRI
30 CONTINUE

C STEP 3-2: DETERMINE CORRECT SAMPLE RATE.
40 IF(IMODE.GE.2) GO TO 44
   IF(MRNG.GT.9) GO TO 42
   MSAM=1
   GO TO 50
   42 MSAM=2
   GO TO 50
44 IF(MRNG.GT.4) GO TO 46
   MSAM=1
   GO TO 50
46 MSAM=2

C STEP 3-3: DETERMINE CORRECT PRF.
50 IF(IMODE.GE.2) GO TO 54
   IF(MRNG.GT.9) GO TO 52
   MPRF=1
   GO TO 60
   52 MPRF=3
   GO TO 60
54 IF(MRNG.GT.9) GO TO 56
   MPRF=1
   GO TO 60
56 MPRF=2
60 CONTINUE

C STEP 3-4: SET PRF TRANSITION FLAG.
   MPRF.Mesh=MPRF

C *********************************************
C * STEP 4: INITIALIZE VELOCITY PROCESSOR *
C *********************************************

C STEP 4-1: INITIALIZE MOVING WINDOW VELOCITY AVERAGING.
   DO 10 I=1,4
10 CONTINUE
10 VEST(I) = CGVEL = 20.

C STEP 4-2: SET INITIAL POSITION OF 5 DOPPLER FILTERS.
VR = CGVEL / FLTWID(MPRF)
IVR = INT((VR + 0.5) + 16000).
XX = MOD((IVR, 32.))
MDF(3) = INT((XX))
DO 20 I = 1, 5
MDF(I) = MOD(MDF(I), 32)
20
C **********************
C *STEP 5: INITIALIZE AGC LOOP*
C **********************
AGCO = 1.0
ITXP = 1
C **********************
C *STEP 6: SET TRACK INDICATOR TO ALLOW OPERATION OF TRACK LOOP*
C **********************
MTKINT = 1
ROLD = 0.
ICLOSE = 0
ICLOAD = 0
C NOTE: DEBUGGING PRINT STATEMENTS.
WRITE(6, 899)
WRITE(6, 900) AZRATE, ELRATE, ALRATE, BTRATE, AL, BT
WRITE(6, 901)
WRITE(6, 902) IRNG, IRDOT, SRNG
WRITE(6, 903)
WRITE(6, 904) (VEST(I), I = 1, 4), (MDF(J), J = 1, 5)
WRITE(6, 905)
WRITE(6, 906) IMODE, MRNG, MSAM, MPRF
899 FORMAT('* TRACKER INITIALIZATION: ' / ' ATTRACK: AZRATE',
  ' , ELRATE, ALRATE, BTRATE, AL, BT')
900 FORMAT(6F14.6)
901 FORMAT(' RTRACK: IRNG, IRDOT, SRNG')
902 FORMAT(218, F14.6)
903 FORMAT(' VTRACK: VEST, MDF')
904 FORMAT(4F14.6, 5I8)
905 FORMAT(4I8)
906 FORMAT(4I8)
RETURN
END

• THIS SUBROUTINE SIMULATES THE TRACKING MODES OF THE KU-BAND •
• RADAR.

******************************************************************************

SUBROUTINE TRACK
COMMON /CNTL/ IDUM(3), IASM, ISRCHC, ISRCHG, IAZS, IELS, ISLR, EDNRG,
  EDPA, EDRA
2 COMMON /OUTPUT/ MSWF, MTF, MSF, DUMO(7), IDUMO(4)
COMMON /ICNTL/IIDUM(13), MTKINT, MRNG, MSAM, MPRF, MBKTRK, IDUM2(9)
COMMON /SXSAT/ TSAM, DUM2(14)
COMMON /ATDAT/ DUM1(10), PEF, RREF, DUMA(2)
DIMENSION SLWRT(2)
DATA SLWRT/6.9B14E-3, 3.4907E-1/

A-60
**STEP 1**: INITIALIZE TRACK MODE — INITIALIZE ALL TRACK LOOPS

AND UPDATE STATUS OF DATA VALID FLAGS.

---

**STEP 1-1**: IF TRACK LOOPS INITIALIZED(MTKINT=1) SKIP STEP 1-2 AND IF ALL DATA VALID FLAGS ARE UP(MTF=1) SKIP STEP 1-2 AND 1-3.

IF(MTF.EQ.1) GO TO 6
IF(MTKINT.NE.1) GO TO 5

**STEP 1-1**: INITIALIZE RANGE, ANGLE, AND VELOCITY TRACK LOOPS — ASSUMES

STEADY STATE TRACKING OF TARGET C.G.

CALL TKINIT

**STEP 2**: PERFORM TRACKING LOOP UPDATE PROCEDURE

---

**STEP 2-1**: UPDATE TRANSFORMATION MATRICES AND MATRICE RATES.

CALL TRNSFM

**STEP 2-2**: TRANSFORM TARGET POSITION AND VELOCITY COMPONENTS FROM ORBITER BODY FRAME-TO-ANTENNA LOS FRAME.

CALL PVTRAN

**STEP 2-3**: GENERATE NOISE-FREE TARGET RETURN SIGNAL AND PROCESS SIGNAL TO PRODUCE NOISE-FREE DISCRIMINANT COMPONENTS.

CALL SIGNAL

**STEP 2-4**: ADD EQUIVALENT NOISE TO DISCRIMINANT COMPONENTS AND FORM ALL REQUIRED DISCRIMINANTS.

CALL DISCRM

**STEP 2-5**: UPDATE STATUS OF BREAK-TRACK FLAG.

CALL BRKTRK

**STEP 2-6**: CHECK STATUS OF BREAK-TRACK FLAG — IF BREAK-TRACK FLAG UP (MBKTRK=1) RESET SYSTEM AND RETURN TO SEARCH.

IF(MBKTRK.NE.1) GO TO 7
CALL SYSINT
RETURN

**STEP 2-7**: DETERMINE RADAR SIGNAL STRENGTH (FOR DISPLAY METER) AND UPDATE AGC VALUE.

CALL RSS

**STEP 2-8**: UPDATE ANTENNA GIMBAL POSITIONS AND RATES AND TARGET ANGLES AND ANGLE RATES FOR DISPLAY (GPC-ACQ AND AUTO MODES ONLY.)

IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 10
CALL ATRACK
GO TO 15

**STEP 2-8A**: IF IN GPC-ACQ OR AUTO MODE USE RADAR ESTIMATED TARGET ANGLES AS GIMBAL TRACK SERVO INPUT.

**STEP 2-8B**: IF IN GPC-DES MODE USE GPC-SUPPLIED ANGLE DESIGNATES AS GIMBAL TRACK SERVO INPUT.

PREF=EDPA
RREF=EDRA
CALL POINT
GO TO 15

C STEP 2-8C: IF IN MANUAL MODE USE CREW-SUPPLIED SLEW RATES TO DETER
MINE GIMBAL TRACK SERVO INPUT.
12
PREFF=PREFF+FLOAT(IES)+SLWRTF(ISLR+1)*TSAM
RREF=RREF+FLOAT(IAZS)+SLWRTF(ISLR+1)*TSAM
CALL POINT

C STEP 2-9: UPDATE THE RANGE AND RANGE RATE ESTIMATES.
15 CALL RTRACK

C STEP 2-10: UPDATE ACCURATE VELOCITY ESTIMATE USING VELOCITY
PROCESSOR.
CALL VELPRO

C STEP 2-11: UPDATE ALL RADAR INTERNAL CONTROLS.
CALL CNTRLS
20 RETURN
END

**********************************************************************
* THIS SUBROUTINE UPDATES ALL REQUIRED TRANSFORMATION            *
* MATRICES AND TRANSFORMATION MATRIX RATES.                      *
**********************************************************************

SUBROUTINE TRNSFM

COMMON /INPUT/DUM(9),TBT(3,3),TBTD(3,3)
COMMON /SYSDAT/DUM2(4),CP,SP,DUM4(9),TRB(3,3)
COMMON /ATDAT/CA,SA,CB,SB,DUMI(2),ALRATE,BTRATE,AL,BT,DUM3(4)
COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
DIMENSION TLR(3,3)

C STEP 1-1: PRELIMINARY COMPUTATIONS.

CB=cos(BT)
SB=sin(BT)
CA=cos(AL)
SA=sin(AL)

C STEP 1-2: COMPUTE TRANSFORMATION MATRIX TLB (BODY-TO-LOS FRAME).

TLR(1,1)=CB
TLR(1,2)=SB*SA
TLR(1,3)=SB*CA
TLR(2,1)=CA
TLR(2,2)=SA
TLR(2,3)=CA
TLR(3,1)=SB
TLR(3,2)=CB*SA
TLR(3,3)=CB*CA
CALL MULT33(TLR,TRB,TLB)

C STEP 1-3: COMPUTE TRANSFORMATION MATRIX TLT (TARGET-TO-LOS FRAME).

CALL MULT33(TLB,TBT,TLT)

**********************************************************************
* STEP 2: UPDATE TRANSFORMATION MATRIX RATES                         *
**********************************************************************
**STEP 2-1: COMPUTE TLB-DOT.**

\[
\begin{align*}
TLB_d(1.1) &= -BTRATE \times TLB(3.1) + ALRATE \times SB \times TLB(2.1) \\
TLB_d(1.2) &= -BTRATE \times TLB(3.2) + ALRATE \times SB \times TLB(2.2) \\
TLB_d(1.3) &= -BTRATE \times TLB(3.3) + ALRATE \times SB \times TLB(2.3) \\
TLB_{d,1} &= ALRATE \times SB \times TLB(2.1) \\
TLB_{d,2} &= ALRATE \times CP \times TLB(2.2) \\
TLB_{d,3} &= ALRATE \times CA \\
TLB_d(2.1) &= ALRATE \times SB \times TLB(2.3) \\
TLB_{d,2} &= ALRATE \times CP \times TLB(2.3) \\
TLB_{d,3} &= ALRATE \times CA
\end{align*}
\]

**STEP 2-2: COMPUTE TLT-DOT.**

\[
\begin{align*}
DO & \ 20 \ I = 1, 3 \\
& \ DO \ 20 \ J = 1, 3 \\
& \ DO \ 20 \ K = I, 3 \\
& \ 20 \ TLTD(I, J) = 0.0 \\
& \ DO \ 20 \ K = I, 3 \\
& \ 20 \ TLTD(I, J) = TLTD(I, J) + TLBD(I, K) \times TBT(K, J) + TLB(I, K) \times TBTD(K, J)
\end{align*}
\]

RETURN
END

---

**THIS SUBROUTINE COMPUTES AN ACCURATE, SMOOTHED VELOCITY USING THE KU-BAND RADAR VELOCITY PROCESSOR ALGORITHM.**

**SUBROUTINE VELPRO**

**REAL** IRDOT, IRNG, INTT, IVEL, IVDISC, IFVEL, IRVEL, IRI, IR2, IR3, IF3, IDelta

**COMMON** /CNTL/IPWR, IMOOE, IDUMC(7), DUMC(3)

**COMMON** /OUTPUT/IDLWB(3), SRNG, SRDOT, DUM2(5), DUM4(3)

**COMMON** /ICNTL/I1DUM(14), MRNG, MSAM, MPRT, IDUM1(10), MPFOLD

**COMMON** /SYSDAT/TSAM, DLIMS(14)

**COMMON** /RTDAT/IRDOT, IRNG, RBIA, VEST(4), MDF(5)

**COMMON** /DSCRM/DUM(2), RDISC, VDSC, RRTE, ODISC, DUM3(3)

**DIMENSION** IPROM(128), VT1(3), VT2(3), MW(4, 3)

**DATA** IPROM/127, 127, 125, 124, 122, 121, 120, 118, 117, 116, 114, 113, 111, 109, 107, 106, 105, 104, 102, 101, 99, 98, 97, 95, 94, 93, 92, 90

2 89, 88, 87, 85, 84, 83, 82, 81, 79, 78, 77, 76, 75, 73, 72, 71, 70, 69, 68, 67

3 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48

4 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33

5 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22

6 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

7 0, 0.5163982, 0.04633489

**DATA** MW/1.2.3.4.1.1.2.2.1.1.1.1/

**STEP 1: GENERATE AMBIGUOUS VELOCITY ESTIMATE**

**STEP 1-1: INTEGERIZE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION.**

\[
\begin{align*}
IVDISC &= 5.333333 \times VDISC \\
IF(IDISC.LT.-128.) \text{ IFDISC=}-128. \\
IF(IDISC.GT.127.) \text{ IFDISC=}127.
\end{align*}
\]

**STEP 1-2: COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS VELOCITY ESTIMATE.**

\[
\begin{align*}
\text{INTEG=MDF(2)} \\
IF(IDISC.LT.0.) \text{ INTEG=MOU}(\text{INTEG+1,32})
\end{align*}
\]
C STEP 1-3: COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY
C ESTIMATE.
C
IV=INT(ABS(IVDISC))+1
IFRAC=IPROM(I)
IF(IVDISC.LT.0.) IFRAC=127-IFRAC

C STEP 1-4: COMPUTE AMBIGUOUS VELOCITY ESTIMATE — COMBINE INTEGRAL
C AND FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF FILTER WIDTH.
C FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF A FILTER WIDTH.
IFVEL=FLOAT(IFRAC+128*INTERG)

C
C
C STEP 2: SCALE ROUGH VELOCITY ESTIMATE
C
C STEP 2-1: SCALE LSB OF ROUGH RANGE RATE ESTIMATE TO 4 TIMES A DOPPLER
C WIDTH. 00027620
C DEFINITION: VTI(MPRF)=(RANGE LSB)/((MAX. UNAMBIGUOUS VELOCITY)/8) 00027630
C OR VTI(MPRF)=S./(PRF,UU_DA)
R1=IRDOT*VTI(MPRF)/TSAM
IR1=AINT(R1)
C
C STEP 2-2: PERFORM SOME REQUIRED AUXILIARY CALCULATIONS.
C R2=IR1/8.
IR2=AINT(R2)
IRVEL=IR2+4096.
C
C
C
C STEP 3: RESOLVE AMBIGUITY
C
C STEP 3-1: COMPUTE 3 MSB'S OF AMBIGUOUS VELOCITY ESTIMATE.
IF3=AINT(IFVEL/512.)
C
C STEP 3-2: COMPUTE 3 LSB'S OF SCALED ROUGH RANGE RATE ESTIMATE.
IR3=ABS(IR1-8.+IR2)
IF(R1.LE.0.)GO TO 10
IRVEL=IRVEL+4096.
IR3=-7.-IR5
10 CONTINUE
C
C STEP 3-3: COMPARE 3 MSB'S AND 3 LSB'S AND INCREMENT NUMBER OF
C AMBIGUOUS FILTER BANK WIDTHS APPROPRIATELY.
IDELTA=IR3-IF3
IF(IDELTA.GE.4.) IRVEL=IRVEL+4096.
IF(IDELTA.LE.-4.) IRVEL=IRVEL+4096.
C
C
C
C STEP 4: COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE.
C
C STEP 4-1: ADD NUMBER OF AMBIGUOUS FILTER BANK WIDTHS TO ESTIMATE
C OF FRACTIONAL FILTER BANK WIDTH. NOTE: LSB OF RESULTANT
C ESTIMATE REPRESENTS 1/4096 OF A FILTER BANK WIDTH.
IVEL=INTT(IFVEL-IFVEL)
C
C STEP 4-2: SCALE LSB OF RESULTANT ESTIMATE TO 0.05 FEET/SEC.
C DEFINITION: VT2(MPRF)=((FILTER SEPARATION)/128.)/(VELOCITY LSB)
C OR VT2(MPRF)=(PRF,LAMBDA)/(0.05,8196).
IVEL=INTT(IVEL+VT2(MPRF)+0.5)

C
C
C
C STEP 5: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY

A-64
STEP 5-1: UPDATE REGISTERS OF MOVING WINDOW AVERAGER.
DO 20 I=1,3
  VEST(5-I)=VEST(4-I)
VEST(1)=IVEL
END

STEP 5-2: COMPUTE MOVING WINDOW AVERAGE AND SCALE ANSWER INTO
FEET/SEC FROM UNITS OF 0.05 FEET/SEC.

M=MRF
M1=MW(1,M)
M2=MW(2,M)
M3=MW(3,M)
M4=MW(4,M)
SRDOT=0.8125*(VEST(M1 )+VEST(M2 )+VEST(M3 )+VEST(M4 ))

STEP 6-1: USE ON-TARGET DISCRIMINANT AND VELOCITY DISCRIMINANT TO
DETERMINE UPDATE OF FILTER BANK POSITION.
THE FOLLOWING RULES ARE USED:
CASE 1: ODISC>0. AND -51.<IVDISC<51. IMPLIES NO CHANGE.
CASE 2: ODISC>0. AND IVDISC>51. IMPLIES SHIFT -1.
CASE 3: ODISC>0. AND IVDISC<-51. IMPLIES SHIFT +1.
CASE 4: ODISC<0. AND IVDISC>0. IMPLIES SHIFT -2.
CASE 5: ODISC<0. AND IVDISC<0. IMPLIES SHIFT +2.

IF(ODISC.GE.0.) GO TO 30
IF(IVDISC.LT.0.) MDF(1)=MOD(MDF(1)+2,32)
IF(IVDISC.GE.0.) MDF(1)=MOD(MDF(1)+30,32)
GO TO 40
30 IF(IVDISC.GT.51.) MDF(1)=MOD(MDF(1)+31,32)
IF(IVDISC.LT.-51.) MDF(1)=MOD(MDF(1)+1,32)

40 DO 50 I=1,4
  MDF(I+1)=MOD(MDF(I)+1,32)
RETURN
END

STEP 6-2: RESET REMAINING FILTERS IN THE BANK-OF-5.
DO 50 I=1,4
  MDF(I+1)=MOD(MDF(I)+1,32)
RETURN
END

THIS SUBROUTINE DETERMINES WHETHER ANTENNA IS IN ZONE 1 AND/OR
ZONE 0 (FOR GPC-AQO AND GPC-DES POINTING MODES ONLY).

SUBROUTINE ZONECK
COMMON /CNTL/IDUMC(9),EDRNG,EDPA,EDRA
COMMON /OUTPUT/IDUM1(3),DUM1(2),SPANG,SRANG,DUM3(3),IDUM3(4)
COMMON /ICNTL/IDUM4(15),M1,MZ0,IDUM4(15)
M0=0
M1=1
PI=3.141592653/180.
RB=PI1*SRANG
PB=PI1*SPANG
P=EDPA

00285600
00285610
00285620
00285630
00285640
00285650
00285660
00285670
00285680
00285690
00285700
00285710
00285720
00285730
00285740
00285750
00285760
00285770
00285780
00285790
00285800
00285810
00285820
00285830
00285840
00285850
00285860
00285870
00285880
00285890
00285900
00285910
00285920
00285930
00285940
00285950
00285960
00285970
00285980
00285990
00286000
00286010
00286020
00286030
00286040
00286050
00286060
00286070
00286080
00286090
00286100
00286110
00286120
00286130
00286140
00286150
00286160
00286170
00286180
00286190
R=E0RA
CPB=C0SP(PIB)
SPB=00IN(PIB)
C0RB=C0SP(PIB)
SRB=SIN(PIB)
CP=C0SP(P)
SP=SIN(P)
C0R=SIN(R)
SR=00IN(R)
ANG0IF=AC0S(SP+CRB*SP+SRB+SR+CPB+CRB+CP+CR)/PI
ANG0IF=ABS(ANG0IF)
IF(ANG0IF.GT.3.E) RETURN
MZ0=1
IF(ANG0IF.GT.0.E) RETURN
M21=1
RETURN
END

SMM MODEL AS OF JANUARY 13, 1982

SUBROUTINE SMM
DIMENSION ARRAYS

A) DIMENSION STATEMENTS
REAL KSEED
COMMON /SATDAT/RADAR(3), KTAR, R(70,3), SIG(70), ROLD, ICLOSE, ICLOLD
DIMENSION SIGMA(49), TARG(49,3), PHIMIN(49,3), PHIMAX(49,3)
DIMENSION OFFSET(49), JHOT(49,3), JHOT2E(49,3), PHI(49), FG(3)
DIMENSION VECT(3), COSPHI(49,3), COSPHN(49), COSPHN(49,3), ORIENT(49,3)
DIMENSION ALPH(19,3), V(19,3), DIM(19,3), WRAN(19,3), SDMAX(19,3)
DIMENSION WSCALE(19,3), DP11(19,3), PHIOLD(19,3), VOLD(19,3), KSEED(19,3)
DIMENSION TTRAN(3), ABG(19,3), TMAX(49), PL(49), SDMIN(19,3)

B) DATA STATEMENTS
1. KSEED- SEEDS FOR RANDOM NUMBER GENERATOR "ZUDU".
   DATA KSEED/45,678,988,670,36578,897,345,7777,67,4.
   1 5686,899,444,888,999,555,222,70,80,8000,
   2 5,15,25,35,45,55,65,75,85,95,
   3 7,17,27,37,47,57,67,77,87,97,
   4 9876,984,6666,2398,76,412,7589,4e9,899,561,
   5 205,3895,9457,9643,937,656,453/

2. DIM- THE GENERAL SIZE OF EACH DIFFUSE SCATTERER.
   DATA DIM/57+64.8/

3. WSCALE- WEIGHTING ASSIGNED TO EACH SIDE OF A DIFFUSE
   SCATTERER.
   DATA WSCALE/8+10.84,5.9386,2+5,6804,5.9386,5.6804,4+11.1026,
   1 6.7958,
   2 6.9686,2+2,7111,2+3,6148,2+2,5174,4.3894,2+5.8895,4.3894,
   3 5.8805,4+17.8803,2+6.7958,19.06/

4. ORIENT- THE I, J, K COMPONENTS OF THE NORMAL VECTOR OF EACH
   TARGET.
   a) I COMPONENT
   DATA ORIENT/13=0....9976....9976....9976.1....1,
   1 23=0....9976....9976....9976,1....2=1,
   2 1=1....2=6.422,2=8,....6.49,....6.361,1....4.924,8704,6428,1=637,
   2 2=6.9377....6337,2=0,1=1....2=6.422,9272,5158,2924,3=0,....6494,
   4=6.361,2=2,3=1....4924,8704,4.924,....866,....866,1=0,....6428,
   5=6.361,2=6.9377....6337,3=0,
   b) J COMPONENT
   2 1=1....2=6.422,2=8,....6.49,....6.361,1....4.924,8704,6428,1=637,
   3 2=6.9377....6337,2=0,1=1....2=6.422,9272,5158,2924,3=0,....6494,
   4=6.361,2=2,3=1....4924,8704,4.924,....866,....866,1=0,....6428,
   5=6.361,2=6.9377....6337,3=0,
   c) K COMPONENT
   6 2=0,....76,76,1,....1=7604,7716,0,....6.8704,4.924,76,0

A-66
5. ABG- ARRAY OF TRANSFORMATION ANGLES (RAD), ALPHA, BETA, GAMMA, FOR DIFFUSE SCATTERERS.

\[
\begin{align*}
\text{DATA ABG} & = \{3.141593, 2\pi \} \times 1.570796, 2\pi, 2.031232, 1.576796, 1.542392, 1.375023, 1.479522, 1.293842, 1.118418, 1.033960\}.
\end{align*}
\]

6. SIGMA- THE CALCULATED RCS FOR EACH TARGET IN M^2.

\[
\begin{align*}
\text{DATA SIGMA} & = \{2.031232, 2.031232, 1.576796, 1.542392, 1.375023, 1.479522, 1.293842, 1.118418, 1.033960\}.
\end{align*}
\]

7. TARG- TARGET POSITION (IN X,Y,Z COORDINATES) RELATIVE TO THE COORDINATE AXIS OF SSMM.

\[
\begin{align*}
\text{DATA TARG} & = \{1.2, 2.1, 3.0, 4.1, 5.2, 6.3, 7.4, 8.5, 9.6\}.
\end{align*}
\]

8. PHIMIN- MINIMUM ANGLE OF DEVIATION FROM SSMM COORDINATES RELATIVE TO TARGET NORMAL.

\[
\begin{align*}
\text{DATA PHIMIN} & = \{1.2, 2.1, 3.0, 4.1, 5.2, 6.3, 7.4, 8.5, 9.6\}.
\end{align*}
\]

9. PHIMAX- MAXIMUM ANGLE OF DEVIATION FROM SSMM COORDINATES RELATIVE TO TARGET NORMAL.

\[
\begin{align*}
\text{DATA PHIMAX} & = \{1.2, 2.1, 3.0, 4.1, 5.2, 6.3, 7.4, 8.5, 9.6\}.
\end{align*}
\]
III. RANDOMIZE DIFFUSE SCATTERER RCS VALUES.

I$\text{SEED}_1=100$
I$\text{SEED}_2=83$
DO 107 I=1,1000
X=RNDU(I$\text{SEED}_1$,I$\text{SEED}_2$)
DO 108 I=1,KWIDE
X=RNDU(I$\text{SEED}_1$,I$\text{SEED}_2$)

SIGMA(I)=SIGMA(I)*2.*X

CONVERT TARGET DATA APPROPRIATELY.

FTM=0.3848
DO 101 J=1,NTAR
SIGMA(I)=SORT(SIGMA(I))/FTM
DO 102 J=1,NTAR
DO 102 J=1,3
TARG(J,I)=TARG(J,I)/FTM
DO 103 J=1,NTAR
TMAX(J)=COS(TMAX(J)*PI/180.)
DO 103 J=1,3
PHIMIN(J,I)=COS(PHIMIN(J,I)*PI/180.)
103 .PHIMAX(J,I)=COS(PHIMAX(J,I)*PI/180.)
DO 105 I=1,NTAR
OFFSET(I)=OFFSET(I)/FTM

V. INITIALIZATION OF TARGET POSITION & COUNTING PARAMETERS
VI. DETERMINE WHICH TARGETS ARE ILLUMINATED.

WRITE(2,500)
500 FORMAT(1X,'TARGET ',2X,'COSPHN')

DO 15 I=1,NTAR
A) DETERMINE THE POSITION OF THE RADAR RELATIVE TO
TARGET SPECULAR POINT.
1. "VECT"—POSITION VECTOR
DO 5 J=1,3
VECT(J)=RADAR(J)-TARG(I,J)
CONTINUE
2. VNORM—MAGNITUDE OF "VECT".
VNORM=SQRT(VECT(1)**2+VECT(2)**2+VECT(3)**2)
B) DETERMINE THE COSINE OF THE ANGLE BETWEEN THE
RADAR POSITION RELATIVE TO THE TARGET SPECULAR PT. &
TARGET NORMAL.
1. CALCULATE THE ANGLE BY EMPLOYING THE DOT PRODUCT
OF THE TWO VECTORS: "COSPHI" & "ORIENT".
DP=0.
DO 7 J=1,3
COSPHI(J)=VEN(J)/VNORM
CONTINUE
2. COSPHN—COSINE OF THE ANGLE; RESULT OF THE DOT PRODUCT.
COSPHN(DP+COSPHI(I,J)*ORI(I,J))
C) TEST OF ILLUMINATION—TWO METHODS: COMPARE COSPHN W/TMAX
OR COMPARE COMPONENTS OF COSPHI W/PHIMIN & PHIMAX.
IF(PL(I).EQ.0.)GO TO 9
2. METHOD 1
IF(COSPHN(I).LT.TMAX(I))GO TO 15
GO TO 11
3. METHOD 2
DO 18 J=1,3
IF(COSPHI(I,J).LT.PHIMAX(I,J) .OR.COSPHI(I,J) .GT.PHIMIN(I,J))GO TO 15
CONTINUE
D) TARGET SHADOWING
1. TEST FIRST 19 TARGETS ONLY.
IF(I.GT.19)GO TO 13
2. FIND SHADOWING VECTOR BY TRANSFORMATION OF COSPHI
FROM SMS TO TARGET COORDINATES.
F1=COSPHI(1,1)*COS(ABG(I,1))+COSPHI(1,2)*SIN(ABG(I,1))
F2=COSPHI(1,2)*COS(ABG(I,1))-COSPHI(1,1)*SIN(ABG(I,1))
F3=COSPHI(1,3)
FB2=F2*COS(ABG(1,2))+F3*SIN(ABG(1,2))
FB3=F3*COS(ABG(1,2))-F2*SIN(ABG(1,2))
FG(1)=F1*COS(ABG(1,3))+FB2*SIN(ABG(1,3))
FG(2)=FB2*COS(ABG(1,3))-F1*SIN(ABG(1,3))
FG(3)=FB3

3. TEST FOR TARGET SHADOWING.

DO 12 J=1,3
  IF(FG(J).GT.SDMAX(I,J).OR.FG(J).LT.SDMIN(I,J))GO TO 15
12 CONTINUE

E) COUNT NUMBER OF ILLUMINATED TARGETS.

1. KTAR- # OF TARGETS ILLUMINATED

KTAR=KTAR+1

2. JHOT- TARGET IDENTIFICATION NUMBER

JHOT(KTAR)=I
SIG(KTAR)=SIGMA(I)

3. NWIDE- # OF DIFFUSE SCATTERERS

IF(I.LE.KWIDE) NWIDE=NWIDE+1
WRITE(2,100))(J),COSPHN(I)

100 FORMAT(1X,13,7X,F6.3)

CONTINUE

VII. UPDATE RANGE OF RADAR RELATIVE TO EACH TARGETS SPECULAR PT.

A) RANGE UPDATE

DO 20 K=1,KTAR
  I=JHOT(K)
  DO 20 J=1,3
  R(K,J)=TARG(I,J)+OFFSET(I)*COSPH(I,J)
20 CONTINUE

IF (JEE.EQ.0)GO TO 24

B) RE-EVALUATE RCS FOR DIFFUSE SCATTERERS

DO 22 K=1,NWIDE
  I=JHOT(K)
  SIG(K)=SORT(ABS(COSPHN(I)))*SIGMA(I)
22 CONTINUE

24 RANGE=SORT(RADAR(1)**2+RADAR(2)**2+RADAR(3)**2)

C) TEST FOR CLOSE RANGE

IF((RANGE-LT.01..OR.RANGE-ROLD.LE.0.).AND.RANGE.LE.270.) ICLOSE=1
IF(RANGE-ROLD.GT.0..AND.RANGE.GT.300.) ICLOSE=0

IF(ICLOSE.EQ.1) GO TO 55
IF(ICL0LD.EQ.1) GO TO 35

D) RANGE UPDATE FOR DIFFUSE SCATTERERS

1. PERFORMS INITIALIZATION OF DIFFERENCE EQUATIONS
   FOR ALL DIFFUSE SCATTERERS.

DO 30 I=1,KWIDE
  IF(COSPHN(I).GT.1.)COSPHN(I)=1.
  PH10LD(I)=ACOS(COSPHN(I))
  DO 25 J=1,3
    V(I,J)=WSCALE(I,J)+(ZUDU(KSEED(I,J))-5)
    VOLD(I,J)=V(I,J)
25 CONTINUE
b) TRANSFORMATION OF "V" FROM TARGET COORDINATES TO SMMS COORDINATES.

\[\begin{align*}
TGAM1 &= v(1,1) \cdot \cos(\alpha(1,3)) - v(1,2) \cdot \sin(\alpha(1,3)) \\
TGAM2 &= v(1,1) \cdot \sin(\alpha(1,3)) + v(1,2) \cdot \cos(\alpha(1,3)) \\
TBETA2 &= \cos(\alpha(1,2)) \cdot TGAM2 + \sin(\alpha(1,2)) \cdot v(1,3) \\
TBETA3 &= \sin(\alpha(1,2)) \cdot TGAM2 - \cos(\alpha(1,2)) \cdot v(1,3) \\
v(1,1) &= \cos(\alpha(1,1)) \cdot TGAM1 - \sin(\alpha(1,1)) \cdot TBETA2 \\
v(1,2) &= \sin(\alpha(1,1)) \cdot TGAM1 + \cos(\alpha(1,1)) \cdot TBETA2 \\
v(1,3) &= TBETA3 \\
\end{align*}\]

DO 26 J = 1, 3 
R(I, J) = R(I, J) + V(I, J) 
CONTINUE

26 CONTINUE
GO TO 55

2. UPDATES THE ANGLE BETWEEN THE RADAR VECTOR & THE TARGET NORMAL.

DO 40 I = 1, NWIDE 
PHI(I) = ACOS(COSPHN(I)) 
DPHI(I) = PHI(I) - PHIOLD(I) 
PHIOLD(I) = PHI(I) 
CONTINUE

3. UPDATES THE RANGE COMPONENTS DUE TO RADAR BEAM DEFLECTION OVER THE SURFACE OF THE DIFFUSE SCATTERER. THE TRANSFORMATION PERFORMS THE SAME FUNCTION DESCRIBED PREVIOUSLY.

DO 50 K = 1, NWIDE 
I = JHT(K) 
DO 45 J = 1, 3 
ALPH(I, J) = EXP(-DIM(I, J) * ABS(DPHI(I) * COSPHN(I))) 
WRAN(I, J) = SQRT(1 - ALPH(I, J)) * WSCALE(I, J) * ZUDU(KSEED(I, J)) - 0.5 
V(I, J) = ALPH(I, J) * VOLD(I, J) + WRAN(I, J) 
VOLD(I, J) = V(I, J) 
CONTINUE

45 CONTINUE
TGAM1 = v(1,1) \cdot \cos(\alpha(1,3)) - v(1,2) \cdot \sin(\alpha(1,3)) 
TGAM2 = v(1,1) \cdot \sin(\alpha(1,3)) + v(1,2) \cdot \cos(\alpha(1,3)) 
TBETA2 = \cos(\alpha(1,2)) \cdot TGAM2 + \sin(\alpha(1,2)) \cdot v(1,3) 
TBETA3 = \sin(\alpha(1,2)) \cdot TGAM2 - \cos(\alpha(1,2)) \cdot v(1,3) 
v(1,1) = \cos(\alpha(1,1)) \cdot TGAM1 - \sin(\alpha(1,1)) \cdot TBETA2 
v(1,2) = \sin(\alpha(1,1)) \cdot TGAM1 + \cos(\alpha(1,1)) \cdot TBETA2 
v(1,3) = TBETA3 
DO 46 J = 1, 3 
R(K, J) = R(K, J) + V(I, J) 
CONTINUE

46 CONTINUE
CONTINUE
CONTINUE
CONTINUE
55 CONTINUE
ROLD = RANGE 
ICLOD = ICLOSE 
RETURN 
END

FUNCTION ZUDU(KSEED) 
THIS SUBROUTINE GENERATES RANDOM NUMBERS.
DATA MU/524287/, XMU/524287/, IETA/997/ 
IF(KSEED) 20.10.20 
20 CONTINUE
KSEED = IETA + KSEED 
IKEEP = KSEED + MU 
KSEED = KSEED - IKEEP + MU 
XRAN = KSEED

A-71
XRAN=XRAN/MU
ZDU=XRAN
10 RETURN
END
APPENDIX B

SOURCE CODE LISTING OF FINAL DELIVERABLE PROGRAM

This appendix is a listing of the final simulation program delivered at the end of the contract. The program has been installed on the Building 44 VAX system at JSC under the Ku-Band account in the KUBAND.HOWARD.MARK directory. The name of the source program is FINSIM1.
MODIFIED 01/27/86 TO COMPUTE AND PLOT REF. RANGE ACCELERATION.

CMDMIN - KUBAND DATA : SSNG, SSDDOT, SSANG, SSPTTE, SSPPRTE, SSALF, SSBT.

WHITE SANDS - REF DATA : X, Y, Z, VX, VY, VZ

REF -> TMR2KU -> ACT : R, ARDOT, SPANG, SRANG, SPRTE, SPRTE, SALF, SBTA, SAZRTE, SELRTE

REF -> TMR2KU -> SIM : HRNG, HRDOT, HRANG, HPANG, HRRT, HPRT, HALP, HBET, HELRT, HELRT

COMMON /TARGET/ITARG, SRC
COMMON /ACTDAT/R, ARDOT, SPANG, SRANG, SPRTE, SSRTTE, AL, BT, SALF, SBTA
1 .ER(3), EV(3), ERTO(3), AZRATE, ELRATE, SAZRTE, SELRTE
2 .AX, AY, AZ, AA, AA, AY, AAZ, RACCEL
COMMON /TERM/ITERM, XMO, XDAY, XYR, TBIAS, XJMO, XJDAY, XJYR
COMMON /OUTPUT/MSWF, MSF, HRANG, HRRT, HPANG, HPRT, HALP, HBET
2 .HRRT, HRSS, MADVF, MRSDVF, MARDVF, MRRDVF
3 .HALP, HBET
COMMON /SYSDAT/TS, DUM2(14)
COMMON /TMR/X, Y, Z, VX, VY, VZ
1 .DLU(3), DEL(3), DUE(3)
2 .DSU(3), TLAZT, THEL1, TAZU1, A23
COMMON /INPUT/RO(3), VO(3), EMB(3)
COMMON /ICNTL/IDUM(16), MPRF
CHARACTER ANS, REPLY
CHARACTER*11 FPRO(57)
CHARACTER*40 IXT, LPRO(57)
CHARACTER*80 COMMENT
CHARACTER*11 UNIT77
INTEGER IREF
INTEGER*2 IS1, IS2
DIMENSION TP(2001), O(2001, 43)
DIMENSION ITILT(10)
DIMENSION RNEL(3), ROLD(3), VNEW(3), VOLD(3)
BYTE IC(120)

TEST DATA FROM WS32TDATA1

DATA LPRO(1)'/ SIM DATA PROFILE HL146AB$'/
DATA LPRO(2)'/ SIM DATA PROFILE HL246AB$'/

B-2
| DATA LPRO(3) | SIM DATA PROFILE HJ146AB$ |
| DATA LPRO(4) | SIM DATA PROFILE HEL30AB$ |
| DATA LPRO(5) | SIM DATA PROFILE H30SKAB$ |
| DATA LPRO(6) | SIM DATA PROFILE H30SKAC$ |
| DATA LPRO(7) | SIM DATA PROFILE HEL30AC$ |
| DATA LPRO(8) | SIM DATA PROFILE HEL30AD$ |
| DATA LPRO(9) | SIM DATA PROFILE HL246AC$ |
| DATA LPRO(10) | SIM DATA PROFILE HL346AB$ |
| DATA LPRO(11) | SIM DATA PROFILE HL446AB$ |
| DATA LPRO(12) | SIM DATA PROFILE H546AB$ |
| DATA LPRO(13) | SIM DATA PROFILE H546AC$ |
| DATA LPRO(14) | SIM DATA PROFILE HL246AD$ |
| DATA LPRO(15) | SIM DATA PROFILE HL446AC$ |
| DATA LPRO(16) | SIM DATA PROFILE HL146AD$ |
| DATA LPRO(17) | SIM DATA PROFILE HL346AS$ |
| DATA LPRO(18) | SIM DATA PROFILE HJ146AC$ |
| DATA LPRO(19) | SIM DATA PROFILE HEL30AS$ |
| DATA LPRO(20) | SIM DATA PROFILE HEL30AF$ |
| DATA LPRO(21) | SIM DATA PROFILE H30SKAD$ |
| DATA LPRO(22) | SIM DATA PROFILE H30SKAE$ |
| DATA LPRO(23) | SIM DATA PROFILE H30SKAS$ |
| DATA LPRO(24) | SIM DATA PROFILE HEL30AS$ |
| DATA LPRO(25) | SIM DATA PROFILE HEL30AC$ |
| DATA LPRO(26) | SIM DATA PROFILE H30SKAC$ |
| DATA LPRO(27) | SIM DATA PROFILE H30SKAC$ |
| DATA LPRO(28) | SIM DATA PROFILE H30SKAE$ |
| DATA LPRO(29) | SIM DATA PROFILE H30SKAG$ |
| DATA LPRO(30) | SIM DATA PROFILE HEL30AG$ |
| DATA LPRO(31) | SIM DATA PROFILE HL546AE$ |
| DATA LPRO(32) | SIM DATA PROFILE HL246AE$ |
| DATA LPRO(33) | SIM DATA PROFILE HL446AE$ |
| DATA LPRO(34) | SIM DATA PROFILE HL146AE$ |
| DATA LPRO(35) | SIM DATA PROFILE HL346AE$ |
| DATA LPRO(36) | SIM DATA PROFILE HJ146AE$ |
| DATA LPRO(37) | SIM DATA PROFILE HL546AE$ |
| DATA LPRO(38) | TSS SIM DATA PROFILE GEM1$ |
| DATA LPRO(39) | TSS SIM DATA PROFILE GEM2$ |
| DATA LPRO(40) | TSS SIM DATA PROFILE GEM3$ |
| DATA LPRO(41) | TSS SIM DATA PROFILE SAT1$ |
| DATA LPRO(42) | TSS SIM DATA PROFILE SAT2$ |
| DATA LPRO(43) | TSS SIM DATA PROFILE SAT3$ |
| DATA LPRO(44) | TSS SIM DATA PROFILE SAT4$ |
| DATA LPRO(45) | TSS SIM DATA PROFILE SAT5$ |
| DATA LPRO(46) | TSS SIM DATA PROFILE SAT6$ |
| DATA LPRO(47) | TSS SIM DATA PROFILE BAL1$ |
| DATA LPRO(48) | TSS SIM DATA PROFILE BAL2$ |
| DATA LPRO(49) | TSS SIM DATA PROFILE BAL3$ |
| DATA LPRO(50) | TSS SIM DATA PROFILE BAL4$ |
| DATA LPRO(51) | TSS SIM DATA PROFILE BAL5$ |
| DATA LPRO(52) | TSS SIM DATA PROFILE HL546AG$ |
| DATA LPRO(53) | SIM DATA PROFILE HL246AF$ |
| DATA LPRO(54) | SIM DATA PROFILE HL446AE$ |
| DATA LPRO(55) | SIM DATA PROFILE HL146AE$ |
| DATA LPRO(56) | SIM DATA PROFILE HL346AF$ |
| DATA LPRO(57) | SIM DATA PROFILE HJ146AE$ |
| DATA FPPO(1) | 'HL146AB.XXX' |
| DATA FPPO(2) | 'HL246AB.XXX' |
| DATA FPPO(3) | 'HL30AB.XXX' |
| DATA FPPO(4) | 'HEL30AB.XXX' |
| DATA FPPO(5) | 'H30SKAB.XXX' |
| DATA FPPO(6) | 'H30SKAC.XXX' |
| DATA FPPO(7) | 'HEL30AC.XXX' |
| DATA FPPO(8) | 'HEL30AD.XXX' |
| DATA FPPO(9) | 'HL246AC.XXX' |
DATA FPRO(10)/'HL346AB.XXX'/
DATA FPRO(11)/'HL446AB.XXX'/
DATA FPRO(12)/'HL546AB.XXX'/
DATA FPRO(13)/'HL546AC.XXX'/
DATA FPRO(14)/'HL246AD.XXX'/
DATA FPRO(15)/'HL446AC.XXX'/
DATA FPRO(16)/'HL146AC.XXX'/
DATA FPRO(17)/'HL346AD.XXX'/
DATA FPRO(18)/'HL146AC.XXX'/
DATA FPRO(19)/'HEL30AE.XXX'/
DATA FPRO(20)/'HEL30AF.XXX'/
DATA FPRO(21)/'HL30SKAD.XXX'/
DATA FPRO(22)/'HL30SKAE.XXX'/
DATA FPRO(23)/'HL30SKAF.XXX'/
DATA FPRO(24)/'HEL30AG.XXX'/
DATA FPRO(25)/'HEL30AH.XXX'/
DATA FPRO(26)/'HL30SKAG.XXX'/
DATA FPRO(27)/'HL30SKAH.XXX'/
DATA FPRO(28)/'HL30SKAI.XXX'/
DATA FPRO(29)/'HEL30AI.XXX'/
DATA FPRO(30)/'HEL30AJ.XXX'/
DATA FPRO(31)/'HL546AE.XXX'/
DATA FPRO(32)/'HL246AE.XXX'/
DATA FPRO(33)/'HL446AD.XXX'/
DATA FPRO(34)/'HL146AD.XXX'/
DATA FPRO(35)/'HL346AE.XXX'/
DATA FPRO(36)/'HL146AD.XXX'/
DATA FPRO(37)/'HL546AF.XXX'/
DATA FPRO(38)/'GEM1.XXX'/
DATA FPRO(39)/'GEM2.XXX'/
DATA FPRO(40)/'GEM3.XXX'/
DATA FPRO(41)/'SAT1.XXX'/
DATA FPRO(42)/'SAT2.XXX'/
DATA FPRO(43)/'SAT3.XXX'/
DATA FPRO(44)/'SAT4.XXX'/
DATA FPRO(45)/'SAT6.XXX'/
DATA FPRO(46)/'SAT8.XXX'/
DATA FPRO(47)/'BAL1.XXX'/
DATA FPRO(48)/'BAL2.XXX'/
DATA FPRO(49)/'BAL5.XXX'/
DATA FPRO(50)/'BAL6.XXX'/
DATA FPRO(51)/'BAL7.XXX'/
DATA FPRO(52)/'HL546AG.XXX'/
DATA FPRO(53)/'HL246AF.XXX'/
DATA FPRO(54)/'HL446AE.XXX'/
DATA FPRO(55)/'HL146AE.XXX'/
DATA FPRO(56)/'HL346AF.XXX'/
DATA FPRO(57)/'HL146AE.XXX'/

SIMULATION FILE MODIFICATION

A23=24.5
TS=0.051
WRITE (6,*) 'INPUT RCS IN SQUARE METERS'
READ (5,*) RCSM
SRCS=RCSM*3.28*3.28
SRCS=SQRT(SRCS)
ITARG=0

WRITE (6,*)'1 : TEK'
WRITE (6,*)'2 : VT125'
WRITE (6,*)'3 : VT1240'
WRITE (6,*)'4 : PC'
READ (5,*) ITERM

WRITE (6,*)'ENTER : 1 IF YOU ARE PROCESSING TMR DATA,'
WRITE (6,*)' 2 IF YOU ARE PROCESSING CINE DATA,'
WRITE (6,*)' 3 IF YOU ARE PROCESSING BEST DATA.'
READ (5,*) IREF

WRITE (6,*)'ENTER TIME INTERVAL ( 0.0 FOR THE WHOLE INTERVAL )'
READ (5,*) STIME, STTIME
IF (STTIME.EQ.0.0) STTIME=999

WRITE (6,*)'DO YOU WANT TO FILTER THE DATA ? (Y/N)' READ (5,2322) ANS
WRITE (6,*)'PROFILE NUMBER PROFILE'
DO L=1,19
WRITE (6,200) L, LPRO(L)
ENDDO
WRITE (6,*)'ENTER C TO CONTINUE, Q TO QUIT :
READ (5,101) REPLY
IF (REPLY.EQ.'C') THEN
DO L=20,38
WRITE (6,200) L, LPRO(L)
ENDDO
WRITE (6,*)'ENTER C TO CONTINUE, Q TO QUIT :
READ (5,101) REPLY
IF (REPLY.EQ.'C') THEN
DO L=39,57
WRITE (6,200) L, LPRO(L)
ENDDO
ENDIF
ENDIF
WRITE (6,*)'INPUT PROFILE NUMBER'
READ (5,*) ITAPE
UNIT7=FPRO(ITAPE)
CALL FIXIT(ITILT, LPRO(ITAPE))
IF (ITAPE.LT.38.AND. ITAPE.GT.51) GO TO 39
IF (ITAPE.GE.38.AND. ITAPE.LE.51) GO TO 49

IF (IREF.EQ.1) THEN
UNIT7(9:11)='JST'
ELSE IF (IREF.EQ.2) THEN
UNIT7(9:11)='JSC'
ELSE
UNIT7(9:11)='BST'
ENDIF
GO TO 59

IF (IREF.EQ.1) THEN
UNIT7(6:8)='JST'
ELSE IF (IREF.EQ.2) THEN
UNIT7(6:8)='JSC'
ELSE
UNIT7(6:8)='BST'
ENDIF

OPEN (UNIT=4, FORM='UNFORMATTED', STATUS='OLD', FILE=UNIT7)

TOUT=0.
THAZL1=30.
THEL1=30.
THAZU1=0.
C WRITE(6,*),' INPUT 1 FOR SCREEN OUTPUT'
C READ(5,*),TOUT
J=0
C READ START TIME
READ(4)TBias,GMTIME,XMO,XDAY,XYR
ILOOP=1
CONTINUE
READ(4,END=99)T,SSRNG,SSRDOT,SSRANG,SSRRT,E,SSPRTE
1,X,Y,Z,VX,VY,VZ,AX,AY,AZ,IS1,IS2,RSS,RFPWR,AERR,BERR,ALFX,
BETY,SCRR,SCPRT
IF (T.LT.STIME) GOTO 1
IFJ=2=13
ITF=1.0D0
J=1.0D0
IF (ITF.NE.IF1.AND.ANS.EQ.'Y') GOTO 1
CALL RPAB(SSRANG,SSPANG,SSALP,SSBET)
CALL TMR2KU
DO I=1,3
RNEW(I)=RO(I)
VNEW(I)=V0(I)
END DO
IF(ILOOP.NE.1) GOTO 7
6 CALL EXEC
IF(MPRF.EQ.1) THEN
TS=1.0D51
ELSE
TS=1.19
END IF
IF(ILOOP.EQ.1) THEN
T=1
ILOOP=0
GO TO 196
END IF
7 CONTINUE
T1=T+TS
IF(T1.GT.T) THEN
T1=T-TS
GO TO 196
END IF
DO I=1,3
RO(I)=(RNEW(I)-ROLD(I))*(T1-T2)/(T-T2)+ROLD(I)
V0(I)=(VNEW(I)-VOLD(I))*(T1-T2)/(T-T2)+VOLD(I)
END DO
GO TO 6
196 CONTINUE
T2=T
DO I=1,3
ROLD(I)=RNEW(I)
VOLD(I)=VNEW(I)
END DO
HRRTE=HRRTE*180.0/(3.14159*1000.)
HPRT=HPRT*180.0/(3.14159*1000.)
J=J+1

B-6
IF(J.EQ.2001)GO TO 99
IF(T.GE.STTIME)GO TO 99
TP(J)=T
D(J,1)=SSRNG
D(J,2)=SSRDO
D(J,3)=SSRANG
D(J,4)=SSPANG
D(J,5)=SSRTE
D(J,6)=SSRTE
D(J,7)=SSALP
D(J,8)=SSBET
D(J,9)=HRNG
D(J,10)=HRDOT
D(J,11)=RO(1)
D(J,12)=RO(2)
D(J,13)=RO(3)
D(J,14)=ATAND(-RO(3)/SQRT(RO(1)+RO(2)+RO(2))
D(J,15)=SSRNG-R
D(J,16)=SSRDOT-ARDOT
D(J,17)=SSRANG-SRANG
D(J,18)=SSPANG-SPANG
D(J,19)=SSRTE-SRTE
D(J,20)=SSRTE-SRTE
D(J,21)=SSALP-SALF
D(J,22)=SSBET-SBTA
D(J,23)=SAZT
D(J,24)=SELRT
D(J,25)=SSRTE
D(J,26)=SSRTE
D(J,27)=SSRTE
D(J,28)=SSRTE
D(J,29)=SSRTE
D(J,30)=SSRTE
D(J,31)=SSRTE
D(J,32)=SSRTE
IF(HRSS.LE.0) THEN
   D(J,33)=0
ELSE
   D(J,33)=(32+HRSS)-181.+(48+ALOG10(HRNG))
ENDIF
D(J,34)=RACCEL
D(J,35)=HRNG-R
D(J,36)=HRDOT-ARDOT
D(J,37)=HRANG-SRANG
D(J,38)=HRANG-SPANG
D(J,39)=HRANG-SRANG
D(J,40)=HRANG-SRANG
D(J,41)=HALP-SALF
D(J,42)=HBET-SBTA
D(J,43)=HRSS/32
IF(J.GT.2000)THEN
   WRITE(6,'*') MORE THAN 2000 POINTS'
   STOP
ENDIF
GO TO 1
99 CONTINUE
J=J-1
IXD=0
94 CONTINUE
CALL SORT(TP,D,J,ITILT,IXD,IXD,GMTIME,IREF)
GO TO 94
END
C ************************************************************************************************************
SUBROUTINE SORT(T,D,J,ITILT,IXD,IXD,GMTIME,IREF)
CHARACTER*40 IXT,IYT(43),PRONAME
CHARACTER*40 REFF
DIMENSION ITILT(10),IXL(10),IYL(10)
DATA IXT/'TIME SECONDS$'/
DATA IYT(1);/*KU MDM RANGE FEET$*/
DATA IYT(2);/*KU MDM RANGE RATE FT/SEC$*/
DATA IYT(3);/*KU MDM ROLL ANGLE DEGS*/
DATA IYT(4);/*KU MDM PITCH ANGLE DEGS*/
DATA IYT(5);/*KU MDM ROLL RATE DEG/SEC$*/
DATA IYT(6);/*KU MDM PITCH RATE DEG/SEC$*/
DATA IYT(7);/*KU MDM ALPHA DEGS*/
DATA IYT(8);/*KU MDM BETA DEGS*/
DATA IYT(9);/*SIM RANGE FEET$*/
DATA IYT(10);/*SIM RANGE RATE FT/SEC$*/
DATA IYT(11);/*WSMR X (NORTH) FEET$*/
DATA IYT(12);/*WSMR Y (EAST) FEET$*/
DATA IYT(13);/*WSMR -Z ALTITUDE FEET$*/
DATA IYT(14);/*WSMR ELEVATION ANGLE DEGS*/
DATA IYT(15);/*DELTA RANGE FEET (KU - WSMR )$*/
DATA IYT(16);/*DELTA RANGE RATE FT/SEC (KU - WSMR )$*/
DATA IYT(17);/*DELTA ROLL ANGLE DEG (KU - WSMR)$*/
DATA IYT(18);/*DELTA PITCH ANGLE DEG (KU - WSMR)$*/
DATA IYT(19);/*DELTA ROLL RATE DEG/SEC (KU - WSMR)$*/
DATA IYT(20);/*DELTA PITCH RATE DEG/SEC (KU - WSMR)$*/
DATA IYT(21);/*DELTA ALPHA DEG (KU - WSMR)$*/
DATA IYT(22);/*DELTA BETA DEG (KU - WSMR)$*/
DATA IYT(23);/*WSMR AZ RATE DEG/SEC$*/
DATA IYT(24);/*WSMR EL RATE DEG/SEC$*/
DATA IYT(25);/*KU SCANNER RSS ( VOLTS)$*/
DATA IYT(26);/*KU SCANNER RF POWER ( VOLTS)$*/
DATA IYT(27);/*KU SCANNER ALPHA ERROR ( VOLTS)$*/
DATA IYT(28);/*KU SCANNER BETA ERROR ( VOLTS)$*/
DATA IYT(29);/*KU SCANNER ALPHA X (VOLTS)$*/
DATA IYT(30);/*KU SCANNER BETA Y (VOLTS)$*/
DATA IYT(31);/*KU SCANNER ROLL RATE (VOLTS)$*/
DATA IYT(32);/*KU SCANNER PITCH RATE (VOLTS)$*/
DATA IYT(33);/*SIM RADAR CROSS SECTION (DBSM)$*/
DATA IYT(34);/*WSMR RANGE ACCELERATION FT/SEC/SEC$*/
DATA IYT(35);/*DELTA RANGE FEET (SIM-WSMR)$*/
DATA IYT(36);/*DELTA RANGE RATE FT/SEC (SIM-WSMR)$*/
DATA IYT(37);/*DELTA ROLL ANGLE DEG (SIM-WSMR)$*/
DATA IYT(38);/*DELTA PITCH ANGLE DEG (SIM-WSMR)$*/
DATA IYT(39);/*DELTA ROLL RATE DEG/SEC (SIM-WSMR)$*/
DATA IYT(40);/*DELTA PITCH RATE DEG/SEC (SIM-WSMR)$*/
DATA IYT(41);/*DELTA ALPHA DEG (SIM-WSMR)$*/
DATA IYT(42);/*DELTA BETA DEG (SIM-WSMR)$*/
DATA IYT(43);/*SIM RADAR SIGNAL STRENGTH$*/
IFLAG=1
IF (IREF.EQ.1) THEN
REFF='TMR'
ELSE IF (IREF.EQ.2) THEN
REFF='CINE'
ELSE
REFF='BEST'
ENDIF
DO I=1,43
L=INDEX(IYT(I),'WSMR')
IF (L.GT.0) THEN
IYT(I)(L:L+3) = REFF
ENDIF
ENDDO
CONTINUE
DO I=1,43
WRITE(6,68)1,1YT(I)

FORMAT(1X,14,10X,A40)

ENDDO

WRITE(6,*)'INPUT ID, YD ID=0 FOR TIME'

IF (IFLAG.EQ.0) THEN
  IFLAG=1
  IXD=6
  IYD=1
  GO TO 731
ENDIF

READ(5,,)IXD,IYD

731 IF(IXD.EQ.0) THEN
  DO I=1, J
    X(I)=T(I)
    Y(I)=D(I, IYD)
  ENDDO
  CALL FIXIT(IXL,IXT)
  CALL FIXIT(IYL,IYT(IYD))
ELSE
  DO I=1, J
    X(I)=D(I, IXD)
    Y(I)=D(I, IYD)
  ENDDO
  CALL FIXIT(IXL,IYT(IXD))
  CALL FIXIT(IYL,IYT(IYD))
ENDIF

CALL PLOTIT(ITILT,IXL,IYL,X,Y,J,GMTIME,IYD,IXD)

GOTO 1

CONTINUE

RETURN

END

C*************************************************************************
SUBROUTINE FIXIT(IOUT,IN)

DIMENSION IOUT(10)

CHARACTER*4 ITEMP(10)

CHARACTER*4 IN

ITEMP(1)=(IN(1:4))
ITEMP(2)=(IN(5:8))
ITEMP(3)=(IN(9:12))
ITEMP(4)=(IN(13:16))
ITEMP(5)=(IN(17:20))
ITEMP(6)=(IN(21:24))
ITEMP(7)=(IN(25:28))
ITEMP(8)=(IN(29:32))
ITEMP(9)=(IN(33:36))
ITEMP(10)=(IN(37:40))

ENCODE(40,999,IOUT)(ITEMP(I),I=1,10)

999 FORMAT(10A4)
 RETURN

END

C*************************************************************************
SUBROUTINE PLOTIT(ITILT,IXL,IYL,X,Y,J,GMTIME,IYD,IXD)

COMMON /TERM/ITERM,XMO,XDAY,XYR,TBIAS,XMO,XJDAY,XJYR

COMMON/TMR/A,B,C,D,E,F,G(3),AH(3),AI(3),AJ(3),THAZL1,THEL1,THAZU1

DOUBLE PRECISION SIG.AVG

BYTE CR(2)

DIMENSION ITILT(8),IXL(8),IYL(8)

DIMENSION X(1),Y(1),TINL(30)

WRITE(6,*)' 1 FOR MEAN AND STANDARD DEVIATION OF Y'

READ(5,*)ISTA

NSC=8

XMAX=X(1)

XMIN=X(1)

YMAX=Y(1)
YM1 = Y(1)
GMTIME = GMTIME/60. / 60.
GMTIME = INT(GMTIME)
GMHOUR = (GMTIME - GMHOUR)*60.
GMHOUR = INT(GMHOUR)
GMMIN = INT(GMHOUR)
GMSEC = INT((GMMIN - GMMIN)*60.)
DO I = 1, J
IF(X(I).GT.YMAX) ymax = X(I)
IF(X(I).LT.YMIN) ymin = X(I)
IF(Y(I).GT.YMAX) ymax = Y(I)
IF(Y(I).LT.YMIN) ymin = Y(I)
END DO
IF(YMAX.EQ.YMIN) ymax = YMIN + 1.1
IF(YMAX.EQ.YMIN) ymax = YMIN + 1.1
IF(YMAX.EQ.YMIN) ymax = 0.1
CONTINUE
YMAX1 = YMAX
YMIN1 = YMIN
IF (ITERM.EQ.1) CALL TEKALL(4114, 480, 0, 1, 0)
IF (ITERM.EQ.2) CALL REGIS (1.0)
IF (ITERM.EQ.3) CALL PVT240
IF (ITYD.EQ.1) CALL RINTL(X, Y, J, TINL, NTINL)
CALL BGNPL(-1)
CALL FLATBD
CALL PAGE(14., 20.)
CALL AREA2D (9.0, 14.0)
CALL HEIGHT(.45)
CALL TITLE(1.0, 100. 1X, 100. 1YL, 9.0, 13.5)
CALL MESSAG(ITILT, 100, -0.6, 16.5)
CALL RESET ('HEIGHT')
CALL HEIGHT (.3)
CALL TITLE('TEST DATES', 1, 9.0, 15.5)
IF (XMO.GE.10) THEN
CALL REALNO(XMO, 0, 3.0, 15.5)
ELSE
CALL REALNO(XMO, 0, 3.3, 15.5)
ENDIF
CALL REALNO(XDAY, 0, 3.9, 15.5)
IF (XDAY.GE.10) THEN
CALL REALNO(XYR, 0, 4.8, 15.5)
ELSE
CALL REALNO(XYR, 0, 4.5, 15.5)
ENDIF
CALL MESSAG(' REVISION 12$', 1, 9.0, 15.5)
POSITION CHANGED FROM 13.7 TO 14.2
CALL MESSAG('X-POSITION MOVED FORWARD BY 1.2')
CALL MESSAG(' To = GMT$ ', 1, 100. 1.2, 14.2)
IF (ISTA.EQ.1) THEN
AVG = 0
SIG = 0
DO I = 1, J
AVG = AVG + Y(I)
SIG = SIG + Y(I)**2
END DO
AVG = AVG/J
SIG = SIG/J - AVG*AVG
CALL MESSAG('MEAN = $', 1, 100. -8.9, -1.0)
CALL REALNO(AVG, 3, 'ABUT', 'ABUT')
CALL MESSAGE('STANDARD DEVIATION=',100,3,3,-2.0)
CALL REALNO(SIG,3,'ABUT','ABUT')
ENDIF
CALL XNAME(IXL,100)
CALL YNAME(IYL,100)
CALL INTAXS
CALL YAXANG(0.)
ENDIF
IF(NSC.EQ.0)THEN
CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
ENDIF
IF(NSC.EQ.1)THEN
CALL GRAF(XMIN,'SCALE',XMAX,YMIN,'SCALE',YMAX)
ENDIF
IF(NTINL.NE.0.AND.IXD.EQ.0)THEN
DO K=1,NTINL
   CALL RLVEC(TINL(K),YMIN1,TINL(K),YMAX1)
ENDDO
ENDIF
CALL CURVE(X,Y,J,.)
CALL GRID(I,1)
CALL HEIGHT(.1)
CALL RESET('HEIGHT')
CALL DONEP
C
SUBROUTINE RPAB(ROLLQ,PITCHQ,ALPHA,BETA)
DEGRAD-57.29576
PSI=67./DEGRAD
PIT=PITCHQ/DEGRAD
ROLL=ROLLQ/DEGRAD
XB=SIN(PIT)
YB=-((SIN(ROL))/SQRT(1.0-XB*XB))
Z=SQRT(1.0-XB*XB-YB*YB)
IF(ROLL.LT.90.0.AND.ROLL.GE.-90.0)Z=-Z
XR=XB*COS(PSI)+YB*SIN(PSI)
YR=YB*COS(PSI)-XB*SIN(PSI)
YRZ=SQRT(YR*YR+Z)*Z)
ALF=ASIN(YR/YRZ)
BTA=ASIN(-XR/SQRT((XR+XR+YR+YR+Z+Z))
ALPHA=ALF+DEGRAD
BETA=BTA+DEGRAD
IF(Z.GE.0.0.AND.YR.LE.0.0)ALPHA=(180.0+ALPHA)
IF(Z.GE.0.0.AND.YR.GT.0.0)ALPHA=(180.0-ALPHA)
RETURN
END
C
SUBROUTINE RINTL(T,R,N,TI,J)
DIMENSION RI(5),R(1),DS(5),TI(30),T(1)
DATA R 1/2550., 5750., 11510., 23030., 43510.1
RMAX=R(1)
RMIN=R(1)
DO 1 I=1,N
RMAX=AMAX1(RMAX,R(I))
RMIN=AMIN1(RMIN,R(I))
1 CONTINUE
MRMAX=1
MRMIN=1
DO 2 I=1,5
IF(RMAX.GT.R(I)) MRMAX=I
IF(RMIN.GT.R(I)) MRMIN=I
2 CONTINUE
J=0
IF(MRMAX.EQ.MRMIN) RETURN
J=0
DO 3 L=1,5
DS(L)=R(1)-RI(L)
3 CONTINUE
DO 4 I=1,N
DO 5 L=1,5
IF((R(I)-RI(L))*DS(L) .LT. 0 )THEN
J(J)=I
T(I)=T(I)
DS(L)=R(I)-RI(L)
ENDIF
5 CONTINUE
4 CONTINUE
RETURN
END

C **********************************************************************
C ** MODMED JWG 2/8/85
C **
C ** INPUT VIA COMMON VIA X,Y,Z,VX,VY,VZ,AX,AY,AZ
C ** OUTPUT VIA COMMON /ACTDAT/
C **
C *** WHITE SANDS TO KU-BAND RADAR PARAMETER CONVERSION ***
C **
C
C ** COMMENTARY **
C **
C ** PURPOSE **
C THIS SOFTWARE TAKES THE POSITION AND VELOCITY OF A TARGET REFERENCED
C TO THE PEARL SITE SURVEY CAP AND CALCULATES THE VALUES OF THE KU-BAND
C RADAR PARAMETERS AS SEEN AT THE KU-BAND RADAR GIMBAL AXES INTERSECTION.
C THESE CALCULATIONS INVOLVE COORDINATE ROTATIONS THROUGH A THREE-AXIS
C POSITIONER AND FOUR TRANSLATIONS FROM THE PEARL CAP TO THE RADAR GIMBAL
C AXES INTERSECTION.
C THESE CALCULATIONS ARE TO BE DONE BY WSMR DATA REDUCTION USING THE WSMR
C RANGE REFERENCE ESTIMATIONS OF TARGET LOCATION WITH TIME. COMPARISON
C CAN BE MADE DIRECTLY WITH THE KU-BAND OUTPUTS FOR THE SAME TIME VALUES.
C **
C ** INPUTS & CONSTANTS **
C WSMR PROVIDED INPUTS:
C WSMR WILL PROVIDE TARGET POSITION = X, Y, Z AND VELOCITY = VX, VY,
C VZ AS INPUTS TO THIS PROGRAM.
C UNITS ARE FEET AND FEET/SECOND.
C THE COORDINATE SYSTEM IS:
C ORIGIN = PEARL SURVEY CAP
C X-AXIS IS POSITIVE TOWARD THE NORTH
C Y-AXIS IS POSITIVE TOWARD THE EAST
C
B-12
NEGATIVE Z-AXIS IS UPWARD ALONG THE LOCAL VERTICAL.

CONSTANTS PROVIDED BY SIMULATION TEST TAPE:
FOR ANY GIVEN TEST THE FOLLOWING PARAMETERS WILL BE DEFINED ON THE SIMULATION MAGNETIC DATA TAPE AND WILL REMAIN CONSTANT FOR THAT TEST:
DSU(1) = 1.3 IS THE LOCATION OF THE KU-BAND RADAR GIMBAL AXES IN UPPER AZIMUTH COORDINATES.
THAZ1 = IS THE LOWER AZIMUTH AXIS ROTATION ANGLE IN DEGREES.
THEL1 = IS THE ELEVATION AXIS ROTATION ANGLE IN DEGREES.
THAZU1 = IS THE UPPER AZIMUTH AXIS ROTATION ANGLE IN DEGREES.

ONE TIME INPUT CONSTANTS:
THE FOLLOWING PARAMETERS WILL BE MEASURED AFTER INSTALLATION OF THE ANTENNA PEDESTAL AT THE PEARL SITE. THEIR VALUES SHOULD NOT CHANGE. THEY ARE CURRENTLY DEFINED AS ZERO IN THIS SOFTWARE.

DLP(1) = 1.3 LOCATION OF THE LOWER AZIMUTH ORIGIN IN PEARL COORDINATES.
DEL(1) = 1.3 LOCATION OF THE ELEVATION ORIGIN IN LOWER AZIMUTH COORDINATES.
DUE(1) = 1.3 LOCATION OF THE UPPER AZIMUTH ORIGIN IN ELEVATION COORDINATES.

** SOFTWARE OUTPUTS **
THIS SOFTWARE PRODUCES THE FOLLOWING OUTPUTS REFERENCED TO THE RADAR GIMBAL AXES INTERSECTION.

R = RANGE (FT)
ARDOT = RANGE RATE (FT/SEC)
SRANG = ROLL ANGLE (DEG)
SPANG = PITCH ANGLE (DEG)
SRRTE = INERTIAL ROLL RATE (DEG/SEC)
SALP = ALPHA ANGLE (DEG)
SBTA = BETA ANGLE (DEG)
AZRTE = AZIMUTH ANGLE RATE (DEG/SEC)
ELRTE = ELEVATION ANGLE RATE (DEG/SEC)

** EXAMPLE **
AN EXAMPLE CASE IS INCLUDED IN THE CODE. IF THIS SOURCE IS COMPILED, LINKED, AND EXECUTED, OUTPUTS WILL GO TO UNIT 6. THEIR VALUES SHOULD BE:

R = 43760.6016
ARDOT = -9.87364578
SRANG = 25.2446920
SPANG = 28.24687998
SRRTE = -.926818556E-1
SRRTE = - .688237743E-2
SALF = -36.1576255
SBTA = 9.27436439
AZRTE = -30.2744657E-81
ELRTE = -1.05446391

SUBROUTINE TM2KU
COMMON /TMR/X,Y,Z,VX,VY,VZ.
1 DLP(3),DEL(3),DUE(3),
2 DSU(3),THAZ1,THE1,THAZU1,A23
COMMON /INPUT/RO(3),VA(3),EMB(3)
COMMON /ACDAT/R,ARDOT,SPANG,SRANG,SRRTE,SRTE,AL,BT,SALF,SBTA,
1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
2,AX,AY,AZ,AAX,AAY,AAZ,RACCEL
DIMENSION DLP(3),DEL(3),DUE(3),DSU(3)
DIMENSION AZL(3,3),ELV(3,3),AZU(3,3)
DIMENSION OPT(3),DLT(3),DET(3),DUT(3),DST(3)
DIMENSION DLAZ(3),DLAV(3),DAZU(3)
DIMENSION VPT(3),VLAZ(3),VELV(3),VST(3)
DIMENSION APT(3),ALAZ(3),AEV(3),AST(3)
DATA DEGRAD/57.275/ P1/3.14159/
C THE EMB PARAMETERS ARE ALWAYS DEFINED AS 0.0
EMB(1)=0.0
EMB(2)=0.0
EMB(3)=0.0
C EXAMPLE CASE VALUES:
C X=39417.2812
C Y=16164.6678
C Z=-9999.65826
C VX=41.1736259
C VY=73.6755753
C VZ=1.6666667E-02
C THAZL2=45.0
C THEL2=45.0
C THAZU2=0.0
C
C ** INPUTS **
C WSMR WILL NORMALLY PROVIDE X,Y,Z,VX,VY,VZ. REF IS PEARL SURVEY POINT.
C THIS IS PROVIDED VIA COMMON TMR BLOCK
DPT(1)=X
DPT(2)=Y
DPT(3)=Z
VPT(1)=VX
VPT(2)=VY
VPT(3)=VZ
APT(1)=AX
APT(2)=AY
APT(3)=AZ
C
C ** CONSTANTS **
C DLP(I); DEL(I); AND DUE(I) WILL BE PROVIDED ONE TIME AFTER INSTALLATION
C OF THE ANTENNA PEDESTAL
C THIS IS PROVIDED VIA COMMON TMR BLOCK
C DLP(1)=.0
C DLP(2)=.0
C DLP(3)=.0
C DEL(1)=.0
C DEL(2)=.0
C DEL(3)=.0
C DUE(1)=.0
C DUE(2)=.0
C DUE(3)=.0
C
C ** CONSTANTS FROM SIMULATION DATA TAPE **
C THIS IS PROVIDED VIA COMMON TMR BLOCK
C DSU(1)=.0
C DSU(2)=.0
C DSU(3)=.0
C THAZL1=0.0
C THEL1=0.0
C THAZU1=0.0
C EXAMPLE ANGLE VALUES ARE EQUATED HERE.
C THAZL1=THAZL2
C THEL1=THEL2
C THAZU1=THAZU2
C CONVERT TO RADIANS
C THAZL=THAZL1/DEGRAD
C THEL=THEL1/DEGRAD
C THAZU=THAZU1/DEGRAD
C SET UP THE ROTATIONAL MATRICES
CALL AZGEN(AZL, THAZL)
CALL ELGEN(ELV, THEL)
CALL AZGEN(AZU, THAZU)

C CONVERT TARGET IN PEARL TO TARGET AT GIMBALS
DO 11 I=1,3
11 DLT(I)=OPT(I)-DLP(I)
CALL MULT31(AZL, DLT, DLAZ)
DO 21 I=1,3
21 DET(I)=DLAZ(I)-DEL(I)
CALL MULT31(ELV, DET, DELV)
DO 31 I=1,3
31 DUT(I)=DELV(I)-DUE(I)
CALL MULT31(AZU, DUT, DAZU)
DO 41 I=1,3
41 DST(I)=DAZU(I)-DSU(I)

C THESE ARE THE THREE TARGET COORDINATES IN RADAR GIMBAL REFERENCE:
RO(1)=OST(1)
RO(2)=OST(2)
RO(3)=OST(3)

C CONVERT TO VELOCITIES REFERENCED TO GIMBALS
CALL MULT31(AZL, VPT, VLAZ)
CALL MULT31(ELV, VLAZ, VELV)
CALL MULT31(AZU, VELV, VST)

C CONVERT TO ACCELERATIONS REFERENCED TO GIMBALS
CALL MULT31(AZL, VST, VST)
CALL MULT31(ELV, VST, VST)
CALL MULT31(AZU, VST, VST)

C THESE ARE VELOCITIES IN GIMBAL REFERENCE.
VO(1)=VST(1)
VO(2)=VST(2)
VO(3)=VST(3)

C RO(I) VO(I) I=1,3 SHUTTLE BODY POS AND VEL VECTOR
C
C CALCULATE THE KU-BAND RADAR PARAMETERS BASED ON THE INPUTS.
C23=COSE(A23)
S23=SINE(A23)
X1=RO(2)*C23-RO(3)*S23
Y1=RO(2)*S23-RO(3)*C23
Z1=RO(1)
RO(1)=X1
RO(2)=Y1
RO(3)=Z1
VX=VO(2)*C23-VO(3)*S23
VY=VO(2)*S23-VO(3)*C23
VZ=VO(1)
VO(1)=VX
VO(2)=VY
VO(3)=VZ
AAZ=AST(2)*C23-AST(3)*S23
AAY=AST(2)*S23-AST(3)*C23
AXX=AST(1)
CALL ACT
SRTR=SRTR*(DEGRAD/1000.)
SPRT=SPRT*(DEGRAD/1000.)
SALF=AL*DEGRAD
SBTA=BT*DEGRAD
AZRTE=AZRATE*DEGRAD
ELRTE=ELRATE*DEGRAD
RETURN
END

C ******************************************************************************
SUBROUTINE AZGEN(AZ, ANGAZ)
C THIS SUBROUTINE PRODUCES A 3X3 MATRIX, AZ, FOR
C AN AZIMUTH TABLE ROTATION OF ANGAZ RADIANS.

DIMENSION AZ(5,3)

DO 10 I=1,5
    DO 10 J=1,3
    AZ(I,J) = 0.0
10    AZ(1,1) = COS(ANGAZ)
         AZ(1,2) = SIN(ANGAZ)
         AZ(2,1) = -SIN(ANGAZ)
         AZ(2,2) = COS(ANGAZ)
         AZ(3,3) = 1.0
RETURN
END

C SUBROUTINE ELGEN(EL, ANGEL)

DIMENSION EL(3,3)

DO 10 I=1,5
    DO 10 J=1,3
    EL(I,J) = 0.0
10    EL(1,1) = COS(ANGEL)
         EL(1,3) = SIN(ANGEL)
         EL(2,2) = 1.0
         EL(3,1) = SIN(ANGEL)
         EL(3,3) = COS(ANGEL)
RETURN
END

C SUBROUTINE ACT

SUBROUTINE ACT
COMMON /ACTDAT/R, ARDOT, SPANG, SPRT, SRRTE, AL, BT, SALF, SBTA
2, ER(3), EV(3), ERTO(3), AZRATE, ELRATE, AZRTE, ELRTE
3, AX, AY, AAZ, AAY, AAZ, RACCEL
COMMON /INPUT/ ERT(3), EVT(3), DUM(18)
COMMON /SYSDAT/ TSAM, CP, SP, PSI, PSBIAS, DUM2(7), TRB(3,3)
DIMENSION FTLWID(3), RI(10)
DIMENSION TX1(3,3), TX2(3,3), TX3(3,3), TBL(3,3)
DATA PI/3.141592655/
DATA ONE/e/
IF (IONE.EQ.0) CALL DATA
IONE=1
C STEP 1-1: COMPUTE INITIAL INNER AND OUTER GIMBAL POSITIONS.
( NOTE: TRANSFORM CONSISTS OF TRANSLATION PLUS ROTATION.)
C PERFORM TRANSLATION — SHIFT TO RADAR FRAME ORIGIN.
    DO 1 J=1,3
1    ERTO(J) = ERT(1)+DR(J)
C TRANSFORM TARGET POSITION FROM BODY TO RADAR FRAME.
    CALL MULT31(TRB, ERTO, ER)
C TRANSFORM TARGET VELOCITY FROM BODY TO RADAR FRAME.
    CALL MULT31(TRB, EVT, EV)
    SQ=SORT(ER(2)*ER(2)+ER(3)*ER(3))
C COMPUTE INNER(BETA) GIMBAL POSITION — BT.
    IF (ER(1).EQ.0.0 AND SQ.EQ.0.0) STOP
    BT=-ATAN2(ER(1), SQ)
    ER2=ER(2)
RETURN
END
ER3 = ER(3)

C COMPUTE OUTER(ALPHA) GIMBAL POSITION — AL.
IF(ER2.EQ.0.0 .AND. ER3.EQ.0.0) GO TO 8
AL = ATAN2(ER2, ER3)
GO TO 9
8 IF(ER(1).GT.0.0) AL = PI/2.
IF(ER(1).LT.0.0) AL = -PI/2.
IF(ER(1).EQ.0.0) STOP

C STEP 1-2: COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH AND
ELEVATION RATES.
C PRELIMINARY TRIGONOMETRIC COMPUTATIONS.
9 CA = COS(AL)
SA = SIN(AL)
CB = COS(BT)
SB = SIN(BT)

C TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER
GIMBAL(G) REFERENCE FRAME.
WGX = CP* EWB(1) + SP* EWB(2)
WGY = CA* (-SP* EWB(1) + CP* EWB(2)) + SA* EWB(3)
WGZ = CB* EWB(1) - SB* EWB(2) + CA* EWB(3)

C COMPUTE RANGE RATE TO TARGET.
ARDOT = (ER(1)* EV(1) + ER(2)* EV(2) + ER(3)* EV(3))/R

C COMPUTE RANGE TO TARGET.
YZR = SQRT(ER(2)*ER(2) + ER(3)*ER(3))

C COMPUTE RANGE ACCELERATION TO TARGET.
VSQ = EV(1)**2 + EV(2)**2 + EV(3)**2
RACCEL = (VSQ + ER(1)*AAX + ER(2)*AAY + ER(3)*AAZ - ARDOT**2)/R

C COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).
VGY = CA* EV(2) + SA* EV(3)
AZRATE = VGY/R + (CB* WGX - SB* WGZ)

C COMPUTE INITIAL TARGET INERTIAL LOS ELEVATION RATE(ELRATE).
ELRATE = (CB* EV(1) - SB* (-SA* EV(2) + CA* EV(3)))/R + WGY

C COMPREHENSIVE

C TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO
BODY FRAME FOR USE IN DISPLAYS AND G AND N.

CALL GAMMA(TX1, -(BT+BTHIAS))
CALL THETA(TX2, -(AL+ALBIAS))
CALL MULT33(TX2, TX1, TX3)
CALL PHI(TX2, -PSI)
CALL MULT33(TX2, TX3, TBL)

C COMPREHENSIVE

C STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO
BODY FRAME FOR USE IN DISPLAYS AND G AND N.

NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW
ANGLE ERROR WRT BODY FRAME.

C UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
FOR DISPLAY.
SRPTE = 1000.* (TBL(2,1)*AZRATE + TBL(2,2)*ELRATE)

C UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
FOR DISPLAY.
SRPTE = 1000.* (TBL(1,1)*AZRATE + TBL(1,2)*ELRATE)

C UPDATE ANTIENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
SPANG = ASIN(TBL(1,3)) = 57.29576

C UPDATE ANTIENNA IN ORBITER BODY COORDINATES FOR DISPLAY.

B-17
IF(TBL(2,3).EQ.0.0.AND.TBL(3,3).EQ.0.0) GO TO 5
SRANG=ATAN2(-TBL(2,3),TBL(3,3))*.57.29576
GO TO 7
5 IF(TBL(1,3).GT.0.0) SRANG=-90.0
IF(TBL(1,3).LT.0.0) SRANG=90.0
IF(TBL(1,3).EQ.0.0) STOP
C RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90<SRANG<90. AND
C -180<SRANG<180.
7 IF(SRANG.LE.90.) GO TO 8
SPANG=(-180.-ABS(SRANG))+(SRANG/ABS(SRANG))
SRANG=(180.-ABS(SRANG))+(SRANG/ABS(SRANG))
10 CONTINUE
RETURN
END

**********************************************************************
C * THIS SUBROUTINE INITIALIZES ALL DATA REQUIRED BY THE SEARCH. *
C * ACQUISITION, AND TRACK SUBPROGRAMS. *
**********************************************************************

SUBROUTINE DATA
REAL IDUM1
COMMON /RTDAT/IDUM1(2),RBIAS,DUM1(9)
COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA,
2 TGTSIG,GPS,GAS,TRB(3,3)
COMMON /NOISE/NS1,NS2,NN(10),GAUSS(320)
DIMENSION A(3,3),B(3,3),C(3,3)
REAL LT,KTS

C * SYSTEM PARAMETERS *
C
PI=3.1415926
PI=PI/180.

C RADAR FRAME YAW ANGLE IN BODY COORDINATES (DEGREES).
PSI=PI/67.0
CP=COS(PSI)
SP=SIN(PSI)

C RADAR LOCATION OFFSET FROM ORBITER C.G IN BODY COORD. (FEET)
C **** VALUES MODIFIED MAR 24 83 PER FM8 MEMO **************
DR(1)=0.0
DR(2)=1.130
DR(3)=5.79

C RANGE BIAS ERROR IS COMPUTED IN SUBROUTINE RTRACK AS
C FUNCTION OF RANGE
C ALPHA GIMBAL BIAS.
ALBIAS=0.0
C BETA GIMBAL BIAS.
BTBIAS=0.0
C RADAR PLATFORM ORIENTATION ERRORS WITH RESPECT TO BODY FRAME.
C YAW ANGLE ERROR.
PSBIAS=PI/180.
C ROLL ANGLE ERROR.
RLBIAS = PI*0.0
C PITCH ANGLE ERROR.
PTBIAS = PI*0.0
C
C NBIAS = 0 FOR NO BIAS AND RADAR AT ORIGIN
C
C NBIAS = 0
IF( NBIAS .NE. 0 ) GO TO 700
701 FORMAT( ' ALL ANGLE BIAS SET TO ZERO RADAR AT ORIGIN' )
DO 4 I = 1, 3
4 DR(I) = 0.0
C PSI = 0.0
PSBIAS = 0.0
RLBIAS = 0.0
PTBIAS = 0.0
700 CONTINUE
C
C COMPUTE MATRIX OF TRANSFORMATION FROM BODY FRAME TO RADAR FRAME.
C
CALL PHI( B, PSI + PSBIAS )
CALL THETA( A, RLBIAS )
CALL MULT33( A, B, C )
CALL GAMMA( A, PTBIAS )
CALL MULT33( A, C, TRB )
C
C SYSTEM SAMPLE INTERVAL
C
C
C
C
C EQUIVALENT ONE-SIDED NOISE POWER SPECTRAL DENSITY (MW/KHZ)
KTS = 137.5
KTS = 10.0 * ( 0.1 * KTS )
C SYSTEM LOSSES ON TRANSMIT (DB).
LT = 2.5
LT = 10.0 * ( 0.1 * LT )
C ONE-WAY ANTENNA GAIN (DB).
G = 37.7
G = 10.0 * ( 0.1 * G )
C ALMBDA = 0.078845
C CONSTANT FOR PASSIVE TRACKING SNR COMPUTATION.
GP = 4.0 * ( G / 2.0 ) * ( ALMBDA / 2.0 ) * ( ( 4.0 * PI ) / 3.0 ) * LT / KTS
C BEACON PARAMETER (DBM)
BCN = 44.0
BCN = 10.0 * ( 0.1 * BCN )
C CONSTANT FOR ACTIVE TRACKING SNR COMPUTATION.
GA = 4.0 * ALMBDA * 2.0 * BCN / ( ( 4.0 * PI ) / 3.0 ) * KTS
C CONSTANT FOR PASSIVE MODE VIDEO SNR COMPUTATION (DB).
GPS = 183.9
C CONSTANT FOR ACTIVE MODE VIDEO SNR COMPUTATION (DB).
GAS = 146.9
C
C
C RANDOM NUMBER GENERATOR SEEDS
C
C
NS1 = 48
NS2 = 135
NN(1) = 0
C
C INITIALIZE NOISE SEQUENCE.
DO 2 I = 1, 320
2 GAUSS( I ) = ANORM( NS1, NS2 )
IF( ITEST .EQ. 2 ) GO TO 341
ITEST = 2
C
C WRITE( 6, 592 )
**FORMAT(1HI,' RANDOM NUMBER INITIALIZATION')**

**WRITE(6,593)(GAUSS(1.),i=1,320)**

**FORMAT(BF8.4)**

**WRITE(6,592)**

**CONTINUE**

---

***********************************

**DEFINE TARGET PARAMETERS**

---

**TARGET SEARCH CROSS-SECTION ( FIXED TEMPORARILY).**

**TGT SIG=10.0**

**RETURN**

**END**

**SUBROUTINE SETIT**

**COMMON /TARGET/ITARG,SRCS**

**COMMON /LEN1/ANGOFF**

**COMMON /SATDAT/RADAR(3),KTAR,R(70,3),SIG(70),ROLD.**

**I close,ICOLD,JHOT(60)**

**COMMON /CNTL/IPWR,IMODE,ITXP,IASM,ISRCG,ISRCHG,IAZS,IELS,ISLR,**

**2 EDRA,EDPA,EDRNC**

**COMMON /ATDAT/DUM1(10),PREF,RREF**

**COMMON /SYSDAT/TSDUM2(14)**

**COMMON /CGMAIN/RO(3),VO(3),AO(3)**

**COMMON /DSCRM/DL3(6),SIGBAR,SNRD,SIGDB**

**COMMON /AGCDAT/AGCO,AGCODB,SNRDT**

**C ITARG = 0**

**SRCS IS VARIABLE NAME OF RCS VALUE**

**C SRCS = 3.27 IS IMSO TARGET.**

**SRCS=3.27**

**DO I=1,3**

**DO J=1,3**

**TBT(I,J)=0.**

**IF(I.EQ.J)TBT(I,J)=1.**

**TBTD(I,J)=0.**

**ENDO**

**KOLD=1**

**CALL SYSINT**

**IPWR=3**

**IMODE=2**

**IASM=1**

**ITXP=1**

**ISRCHG=0**

**IAZS=0**

**IELS=0**

**ISLR=0**

**ISRCHG=8**

**EDRNG=500.0**

**EDRA=0.0**

**EDPA=0.0**

**PI1=3.14159265/180.**

**EDP=EDPA*PI1**

**EDRA=EDRA*PI1**

**MTP=0**

**MTP=1**

**RETURN**
FUNCTION ANORM(K1,K2)
Y1=RANDU(K1)
Y2=RANDU(K2)
TPI=6.2831852
ANORM=SORT(-2.*ALOG(Y1))*COS(TPI*Y2)
RETURN
END

**THIS FUNCTION GENERATES A RANDOM NUMBER FROM A GAUSSIAN PDF**
WITH ZERO MEAN AND UNIT VARIANCE.

**FUNCTION ANORM(K1,K2)**

**THIS SUBROUTINE UPDATES AZ AND EL INERTIAL LOS RATES, THE**
**ALPHA AND BETA GIMBAL RATES, THE ALPHA AND BETA GIMBAL**
**POSITIONS, AND THE TARGET PITCH AND ROLL ANGLES FOR THE**
**DISPLAY.**

SUBROUTINE ATRACK

REAL INTT,K4,K5,K6
INTEGER AT1A(10,2),AT1E(10,2),AT2A(10,2),AT2E(10,2)
COMMON /CNTL/PWR,IMODE,DUMMY(7),DUMM(3)
COMMON /INPUT/DUMM(6),EMB(3),DUMM(18)
COMMON /OUTPUT/DIDUMM(3),DIDUMM(2),SPANG,SRANG,SRTE,SRSS,
          2 IDUMM(4),SSALP,SSBET
COMMON /CNTL/IDUMM(14),MRNG,MSAM,GPRF,IMDUMM(11)
COMMON /SYSDAT/TSAM,DR(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,
          2 DUMM(5)
COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,BTRATE,AL,BT,
          2 DUMM(4)
COMMON /DSCRM/AZDISC,ELDISC,DUMM(7)
DIMENSION TX(3,3),TX2(3,3),TX3(3,3),TBL(3,3)
DIMENSION TDC(3)

ATRACK MODIFIED JAN 28 1986 BY M. MEYER
MODIFICATIONS TO SUBROUTINE ATRACK WERE IMPLEMENTED TO UPDATE THE LOOP CONSTANTS AND MORE ACCURATELY SIMULATE THE ACTUAL SIGNAL PROCESSING PERFORMED BY THE RADAR.

NEW LOOP CONSTANTS JAN 28 1986

DATA AT1A/9*5.1,6*13,5,3*1/
DATA AT1E/9*6.1,6*16,6,2*1,2/
DATA AT2A/9*407,149,6*562,407,3*149/
DATA AT2E/9*532,195,6*866,532,3*195/
DATA K6/3.60E-5/,K4/.00848876/,K5/.236/,DTOR/.0174533/

DATA TDC/0.0512118,0.1195161,0.2561557/

DEFINITION: AT1=KED*(WN**2)/(4.*DIFFERENCE PATTERN SLOPE) WHERE WN IS NATURAL FREQUENCY OF THE LOOP.
DEFINITION: AT2=KED+TAU WHERE TAU IS PROPORTIONAL TO STEP RESPONSE CONVERGENCE TIME.
C TCON=TSAM/TDC(MPRF)

C * STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *


C * STEP 1: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANSFORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW ANGLE ERROR WRT BODY FRAME).

CALL GAMMA(TX1,-(BT+BTBIAS))
CALL THETA(TX2,-(AL+ALBIAS))
CALL MULTI33(TX2,TX1,TX3)
CALL PHI(TX2,-PSI)
CALL MULTI33(TX2,TX3,TBL)

C * STEP 2: UPDATE ESTIMATED TARGET INERTIAL AZIMUTH AND ELEVATION RATES IN ANTENNA LOS FRAME.

C QUANTIZE THE ANGLE DISCRIMINANTS TO 3/16 DB.
IAZDSC=INTT(5.333333*AZDSC+TCON+0.5)/TCON
IELDSC=INTT(5.333333*ELDSC+TCON+0.5)/TCON
IF((IAZDSC.GT.255)IAZDSC=255
IF((IHANDSC.GT.255)IHANDSC=255
IF((IELDSC.LT.-256)IELDSC=-256
IF((IELDSC.LT.-256)IELDSC=-256

C NEW CODE AS OF JAN 28 1986

C UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.
IAZRATE=KSAT(IAZRATE+AT1A(MRNG,IMODE)*IAZDSC)

C UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.
IELRATE=KSAT(IELRATE+AT2E(MRNG,IMODE)*IELDSC)

AZRATE=K6*DTOR*FLOAT(IAZRATE)
ELRATE=K6*DTOR*FLOAT(IELDRATE)

IALRATE=KSAT(IALRATE+AT1A(MRNG,IMODE)*IAZDSC)
IBTRATE=KSAT(IBTRATE+AT2E(MRNG,IMODE)*IELDSC)

IF((IALRATE.GT.0) THEN
IALRATE=IALRATE/32
ELSE
IALRATE=IALRATE/32
END IF

IF((IBTRATE.GT.0) THEN
IBTRATE=IBTRATE/32
ELSE
IBTRATE=IBTRATE/32
END IF

C COMPUTE REQUIRED COMPONENTS OF ORBITER ANGULAR VELOCITY VECTOR IN OUTER GIMBAL FRAME.
WGX=CP*EWH(1)+SP*EWH(2)
WGY=CA*-SP*EWH(1)+CP*EWH(2)+SA*EWH(3)
WGZ=SA*-SP*EWH(1)+CP*EWH(2)+CA*EWH(3)

C OUTER GIMBAL RATE.
IF(ABS(CB).LT.1.0E-6) GO TO 2
ALRATE=(ALRATE+WGZ*SB)/CB-WGX
GO TO 4
2 ALRATE=0
4 CONTINUE
C INNER GIMBAL RATE.
BTRATE=BTRATE-WGY
C--- END OF JAN 28 1986 MODIFICATIONS---
C******************************************************************************
C******************************************************************************
C • STEP 4: UPDATE INNER AND OUTER GIMBAL POSITIONS. •
C******************************************************************************
C OUTER GIMBAL POSITION (ALPHA ANGLE)
AL=AL+TSAM*ALRATE
C INNER GIMBAL POSITION (BETA ANGLE)
BT=BT+TSAM*BTRATE
C ADD ALPHA AND BETA TO OUTPUT IN DEG
SSALP=AL+57.29576
SSBET=BT+57.29576
C******************************************************************************
C • STEP 5: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO •
C • BODY FRAME FOR USE IN DISPLAYS AND G AND N. •
C******************************************************************************
C NOTE: TRANSFORMATION TBL INCLUDES GIMBAL BIAS ERRORS AND RADAR YAW
C ANGLE ERROR WRT BODY FRAME.
C UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
FOR DISPLAY.
SRRTE=-1000.*(TBL(2,1)*AZRATE+TBL(2,2)*ELRATE)
C UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
FOR DISPLAY.
SRRTE=-1000.*(TBL(1,1)*AZRATE+TBL(1,2)*ELRATE)
C UPDATE ANTENNA PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
SPANG=ASIN(TBL(1,3))*57.29576
C UPDATE ANTENNA IN ORBITER BODY COORDINATES FOR DISPLAY.
IF(TBL(2,3).EQ.0.0.AND.TBL(3,3).EQ.0.0) GO TO 5
SRANG=ATAN2(-TBL(2,3),TBL(3,3))*57.29576
GO TO 7
5 IF(TBL(1,3).GT.0.0) SRANG=-90.0
IF(TBL(1,3).LT.0.0) SRANG=90.0
IF(TBL(1,3).EQ.0.0) STOP
C RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND
C -180.<SRANG<180.
7 IF(SRANG.LE.90.) GO TO 10
SPANG=(180.-ABS(SRANG))*ABS(SRANG)
SRANG=(180.-ABS(SRANG))*SRANG
10 CONTINUE
C******************************************************************************
C NOTE: DEBUGGING PRINT STATEMENTS.
C******************************************************************************
C WRITE(6,899)
899 FORMAT(/' ATRACK DEBUGGING DATA'/)
C WRITE(6,900) ALRATE,BRATE,ARATE,ELRATE,SRRTE,SPRTE
900 FORMAT(3F10.9)
C WRITE(6,901) TBL(1,1),TBL(1,2),TBL(2,1),TBL(2,2)
901 FORMAT(4F10.4)
C WRITE(6,902) AZDISC,ELDISC,IAZDISC,IELDISC
902 FORMAT(4F10.4)
C******************************************************************************
C • INTEGER FUNCTION KSAT JAN 28 1986 •
C******************************************************************************
C THIS FUNCTION CHECKS TRACK LOOP FOR SATURATION

INTEGER FUNCTION KSAT(K)

IF(K.GE.0) THEN
    KSAT=JMIN0(K,2**15)
ELSE
    KSAT=JMAX0(K,-2**15)
END IF
RETURN
END

C THIS SUBROUTINE IMPLEMENTS THE BREAK-TRACK ALGORITHM

SUBROUTINE BRKTRK
REAL IVMAX,THRSHC,THRSHO,IVDISC,INTT,IODISC
COMMON /ICNTL/IDT_2(17),MBKTRK,MBTSUM,MBT(8)
COMMON /DSCRM/DUM(3),VDISC,DUM1,ODISC,DUM2(3)
DATA IVMAX,THRSHC,THRSHO/51.,14.,-11./

C STEP 1: DETERMINE STATUS OF L-H DISCRETE (FTH)
C STEP 1-1: QUANTIZE THE VELOCITY DISCRIMINANT TO 3/16 DB STEPS.
    IVDISC=INTT(IVDISC=5.333333+0.5)
     IFTH=0
    IF(ABS(IVDISC).GE. IVMAX) IFTH=1

C STEP 2: DETERMINE STATUS OF ON-TARGET DISCRETE (OT)
C STEP 2-1: QUANTIZE THE O-DISCRIMINANT TO 3/16 DB STEPS.
    IODISC=INTT(IODISC=5.333333+0.5)
C STEP 2-2: DETERMINE STATUS OF ON-TARGET DISCRIMINANT.
    IOT=0
    IF(IODISC.GE.THRSHC) IOT=1

C STEP 3: DETERMINE STATUS OF ADJACENT ON-TARGET DISCRETE (AOT)
    IAOT=0
    IF(IODISC.LE.THRSHO) IAOT=1

C STEP 4: COMBINE ABOVE DISCRETES TO DETERMINE STATUS OF NO-TARGET DISCRETE (NOTARG).
DEFINITION: THE NO-TARGET DISCRETE IS HIGH (OR 1) IF THE DISCRETES FTH, OT, AND AOT ARE ALL LOW (OR 0).
    NOTARG=(1-IFTH)*(1-IOT)*(1-IAOT)

C STEP 5: DETERMINE STATUS OF BREAK-TRACK FLAG (MBKTRK)
DEFINITION: BREAK-TRACK SHALL BE DECLARED IF NOTARG=1 FOR AT
SUBROUTINE CFAR
COMMON /CNTL/IPWR,IMODE.ITXP,IASM,IDUMC(5),EDRNG,DUMC(2)
COMMON /OUTPUT/MSWF,MTF,MWF,DUMI(7),IDUMI(4)
COMMON /ICNTL/IDUM2(8),KACCLK,MTP,IDUM3(4),MRNG,MSAM,MPRF
COMMON /TGTDAT/SIGMA,CGANG
DIMENSION RI(6),PW(6),NP(6),FW(3),TPRI(3),TS(2),P(41)
DATA NRZ,NSRCH/6,37 \
BILLMOA/g83.5,e.e70845/,RI/2552.,5772.,11544.,23089.,43747.,57722./,PW/0.122,4.15,8.3,16.6,33.2,66.4/ 
NP/1,2,4,8,16,32/,FW/7.7215,3.3090,e.2969/,TS/e.122,2.075/, 
TPRI/143.5,334.7,3731.1/ 
PI=3.14159265 

STEP 1: SET INTERNAL CONTROLS BASED UPON SYSTEM OPERATING MODE

STEP 1-1: GPC MODES OR AUTO/MANUAL MODES"
IF(IASM.GE.3) GO TO 15

STEP 1-2: SET INTERNAL CONTROLS FOR APPROPRIATE MODE.
CONTROL SETTINGS FOR GPC MODES.

STEP 2: DETERMINE RANGE INTERVAL.
DO 5 1=1,NRI
MRNG=1
IF(R(I).GT.EDRNG) GO TO 10
5 CONTINUE

STEP 3: SET SAMPLE RATE
10 MSAM=2

STEP 4: DETERMINE PRF
MPRF=1
IF(EDRNG.GE.RI(6)) MPRF=2

******************************************************************************
THIS SUBROUTINE CONTAINS THE CFAR DETECTION MODEL -
******************************************************************************
GO TO 20
C CONTROL SETTINGS FOR AUTO/MANUAL MODES.
C SET RANGE INTERVAL.
15 \text{MRNG}=6
C SET SAMPLE RATE.
\text{MSAM}=2
C SET PRF.
\text{MPRF}=1
C *************** STEP 2: COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT ***************
20 \text{SNR=SNRV(SIGMA,CGRNGE)}
C *************** STEP 3: IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNRv ***************
C STEP 3-1: CHECK SCAN FLAG.
IF(MSF.EQ.1) GO TO 25
C STEP 3-2: COMPUTE BEAMSHAPE LOSS — BASED UPON C.G. POSITION OFF
BORESIGHT.
\text{BETA2=SPAT(CGANG)**2}
C STEP 3-3: ADD BEAMSHAPE LOSS TO NOMINAL SNRV, I.E. COMPUTE ACTUAL
\text{SNRV.}
\text{SNR=SNR-BETA2}
C ******************* STEP 4: COMPUTE NET PROCESSOR GAIN AND COMBINE WITH SNRV TO FORM ********************
C SNRD.  \text{STEP 4-1: COMPUTE RANGE GATE LOSS (RGL) — DIFFERS FOR GPC AND
\text{AUTO/MANUAL MODES.}}
25 \text{CTD2=PW(MRNG)/2.}
C DETERMINE OPERATING MODE
IF(IASM.GE.3) GO TO 30
C COMPUTE RGL FOR GPC MODES.
\text{DEL=ABS(EDRG--CGRNGE)/CTD2}
\text{IF(DEL.GE.1.5) RGL=0.0}
\text{IF(DEL.GE.0.5.AND.DEL.LT.1.5) RGL=.6666666*(1.5-DEL)**2}
\text{IF(DEL.LT.0.5) RGL=.6666666}
\text{GO TO 35}
C COMPUTE RGL FOR AUTO/MANUAL MODES
\text{DEL=DEL-INT(DEL)}
\text{IF(DEL.LE.1.0) RGL=DEL+DEL}
\text{IF(DEL.GT.1.0.AND.DEL.LT.4.5.AND.DEL1.LT.0.5)}
\text{2 RGL=(1.0-DEL1)**2}
\text{IF(DEL.GT.1.0.AND.DEL.LT.4.5.AND.DEL1.GE.0.5)}
\text{2 RGL=DEL1+DEL1}
C STEP 4-2: COMPUTE NET PRF GAIN — SAME FOR ALL PASSIVE ANTENNA
STEERING MODES.

COMPUTE DOPPLER FREQUENCY ASSOCIATED WITH TARGET RADIAL VELOCITY

\[ FDOP = -2. \cdot CGVEL / ALMDA + 1.0 \cdot 10^{-6} \]

COMPUTE ARGUMENT ASSOCIATED WITH TARGET VELOCITY

\[ ARG = \pi \cdot FDOP \cdot TS(MSAM) \]

COMPUTE NET PRESUM GAIN

\[ PS\bar{G} = \text{SUM}(\text{ARG}, NP(MRNG)) \]

STEP 4-3: COMPUTE NET DOPPLER FILTER GAIN — SAME FOR ALL PASSIVE ANTENNA STEERING MODES.

STEP 4-4: COMPUTE NET PROCESSOR GAIN.

\[ NP\bar{C} = RGL \cdot PSG \cdot DFG \]

STEP 4-5: COMPUTE SNR AT DOPPLER FILTER OUTPUT

\[ \text{SNR} = \text{SNR} \cdot > \text{G} \]

STEP 5-1: DETERMINE INDEX TO ACCESS APPROPRIATE CURVE

IF (IASM.GE.3) GO TO 40

NCRV = 1

GO TO 45

40 NCRV = 3

ADJUST INDEX FOR SCANNING

45 NCRV = NCRV + MSF

STEP 5-2: CONVERT SNR TO DB.

IF (SNR.LE.1.0E-06) GO TO 50

SNR = 10. \cdot \text{ALOG10}(SNR)

GO TO 55

50 SNR = 100.

STEP 5-3: SNR OUTSIDE (0 DB, +20 DB) INTERVAL — IF SO, SET OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.

IF SNRD < 0. DB — DECLARE A MISS.

55 IF (SNR.LE.0.) GO TO 60

IF SNRD > 20. DB — DECLARE A HIT.

60 IF (SNR.GT.20.) GO TO 65

STEP 5-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR INTERPOLATION.

\[ \text{SCALE} = \text{(SNR} + 0.) \cdot 2. + 1. \cdot 0000001 \]

\[ \text{ISNR} = \text{INT} (\text{SCALE}) \]

\[ \text{REMAIN} = \text{SCALE} - \text{FLOAT} (\text{ISNR}) \]

STEP 5-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION.

0000620

0000630

0000640

0000650

0000660

0000670

0000680

0000690

0000700

0000710

0000720

0000730

0000740

0000750

0000760

0000770

0000780

0000790

0000800

0000810

0000820

0000830

0000840

0000850

0000860

0000870

0000880

0000890

0000900

0000910

0000920

0000930

0000940

0000950

0000960

0000970

0000980

0000990

0001000

0001010

0001020

0001030

0001040

0001050

0001060

0001070

0001080

0001090

0001100

0001110

0001120

0001130

0001140

0001150

0001160

0001170

0001180

0001190

0001200

0001210

0001220

0001230

0001240

0001250
**PROB=P(ISNR)+REMAIN*(P(ISNR+1)-P(ISNR))**

**STEP 6: DETERMINE OUTCOME OF DETECTION ATTEMPT**

\[ X=RNDU(MSRCH) \]
\[ IF(X.LT.PROB) \text{GO TO 65} \]

**STEP 7: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT**

**STEP 7-1: IF NO DETECTION — SET TARGET PRESENT FLAG LOW.**

60 \text{MTP=0}
\text{RETURN}

**STEP 7-2: IF DETECTION SUCCESSFUL — SET TARGET PRESENT FLAG HIGH AND INITIALIZE ACQUISITION CLOCK.**

65 \text{MTP=1}
\text{KACCLK=0}
\text{RETURN}
\text{END}

**THIS SUBROUTINE UPDATES ALL RADAR INTERNAL CONTROLS.**

**SUBROUTINE CNTRLRS**

REAL \text{INTT,NFIL,IRNG,IRDOT}

COMMON /CNTL/I_1M,I_MODE,\text{IDUMC(7),DUMC(3)}
COMMON /OUTPUT/I_RNG,SRNG,SRDOT,\text{DUM2(5),IDUM(4)}
COMMON /ICNTL/I_RNG,MRNG,MSAM,MPRF,\text{IDUM1(10),MPFOLD}
COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)

**DIMENSION RI(14),FW(3)**

DATA RI/126.,646.,1526.,2566.,5766.,11526.,23646.,43526,
49926.,1.8228E+6/
DATA FW/7.7215,3.3E6,E.2969/,NRi/10/

**IMPARIMENT OF HYSTERESIS FOR THE SAMPLING RATE CHANGE AND FOR THE PRF CHANGE ALONG WITH CHANGES IN RI(RANGE INTERVAL) WAS COMPLETED FEB 6, 1986 BY M. MEYER**

**STEP 1: SET RANGE INTERVAL PARAMETER**

XRNG=IRNG+0.3125
DO 60 I=1,NRI
\text{IF(XRNG.LE.RI(I)) GO TO 70}
60 \text{CONTINUE}
70 MRNG=:
\text{IF(MRNG.GT.NRI) STOP}

**STEP 2: SET SAMPLE RATE PARAMETER**

\text{IF(I_MODE.GE.2) GO TO 74}
IF(MRNG.GT.9) GO TO 72
MSAM=1
GO TO 80

72 MSAM=2
GO TO 80

C***** MODIFIED FEB 6 1986 BY M. MEYER**************
74 IF(MSAM.EQ.1)THEN
     IF(XRNG.GT.3200.)THEN
         MSAM=2
     ELSE
         MSAM=1
     END IF
C***** MODIFIED FEB 17, 1986 BY M. MEYER *************
C***** GUARANTEES THE CORRECT LOOP BANDWIDTHS**********
C
     IF(XRNG.GT.2560) MRNG=4
     ELSE
         IF(XRNG.GT.2560.)THEN
             MSAM=2
         ELSE
             MSAM=1
         END IF
     END IF
C
     END IF
C
     STEP 3: SET PRF PARAMETER
*
C
     STEP 3-1: DETERMINE IF IN ACTIVE OR PASSIVE MODE.
80 IF(IMODE.GE.2) GO TO 84
C
     STEP 3-2: DETERMINE CORRECT PRF FOR GIVEN OPERATING MODE.
     IF(WRNG.GT.9) GO TO 82
     MPRF,=I
     GO TO 90
     ELSE
         MPRF=3
     GO TO 90
C
     C****** MODIFIED FEB 6 1986 BY M. MEYER **************
84 IF(MPRF.EQ.1)THEN
     IF(XRNG.GT.49920.)THEN
         MPRF=2
     ELSE
         MPRF=1
     END IF
     ELSE
         IF(XRNG.GT.43520.)THEN
             MPRF=2
         END IF
C****** MODIFIED FEB 17, 1986 BY M. MEYER*************
C****** GUARANTEES THE CORRECT CONSTANTS ***************
C
     MRNG=10
C
     C***********************************************
     ELSE
         MPRF=1
     END IF
     END IF
C
     STEP 3-3: IF PRF HAS CHANGED FROM PREVIOUS DATA CYCLE, THEN
C     RESET THE 5 DOPPLER TRACKING FILTERS ACCORDINGLY.
C
     CONTINUE
C
B-29
IF(MPFOLD.EQ.MPRF) GO TO 96
NFIL=INT((-SRDOT/FW(MPRF))+0.5)+31998.
XX=AMOD(NFIL,32.)
MDF(1)=INT(XX)
DO 95 I=1,4
   MDF(I+1)=MOD(MDF(1)+I,32)
95 MDF=MPRF
!
NOTE: DEBUGGING
WRITE(6,999) MPRF,MPFOLD,MDF(1)
999 FORMAT(' MPRF,MPFOLD,MDF1 =',318)
RETURN
END

* THIS SUBROUTINE PERFORMS THE TARGET DETECTION FUNCTION FOR ACTIVE AND PASSIVE MODES AND ALL ANTENNA STEERING MODES. *

SUBROUTINE DETECT
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),EDRNG,DUMC(2)
COMMON /ICNTL/IDUM2(9),MTP,IDUM3(17)
COMMON /SYSDAT/DUM2(12),TGTSIG,GPS,GAS
COMMON /TGTDAT/NT,DUM3(5),RO(3),ROU(3),CGRNGE,CGVEL
COMMON /DETDAT/SIGMA,CGANG

CALL TRNSFM
CALL PVTRAN
!
STEP 1-1: COMPUTE TARGET PARAMETERS WRT RADAR *

CALL TRNSFM
CALL PVTRAN
C
STEP 1-2: COMPUTE TARGET C.G. ANGLE OFF-BORESIGHT (NON-SCANNING).

CGANG=ACOS(-OU(3))
!
STEP 1-3: DETERMINE TARGET CROSS-SECTION.
SIGMA=TGTSIG

STEP 2: PRELIMINARY DETECTION MODE DETERMINATION *

IF(IMODE.EQ.1) GO TO 5
!
STEP 2-2: GPC MODES OR AUTO/MANUAL MODES"
IF(IASM.GE.3) GO TO 10
GO TO 15
!
STEP 3: ACTIVE MODE DETECTION PROCESS *
C
!
STEP 4: PASSIVE AUTO/MANUAL MODE DETECTION PROCESS -
C
C STEP 4-1: CHECK SHORT RANGE FIRST — CALL SINGLE-HIT DETECTION MODEL.
10 CALL SINGLE
C STEP 4-2: CHECK FOR SUCCESS IN SINGLE-HIT DETECTION — IF NOT SUCCESSFUL, THEN TRY LONG RANGE SEARCH.
IF(MTP.EQ.0) CALL CFAR
RETURN
C STEP 5: PASSIVE GPC MODES DETECTION PROCESS
C STEP 5-1: CHECK DESIGNATED RANGE.
15 IF(EDRNG.GT.2552.) GO TO 20
C STEP 5-2: IF DESIGNATED RANGE < 0.42 NM — USE SINGLE-HIT DETECTION MODEL.
CALL SINGLE
RETURN
C STEP 5-3: IF DESIGNATED RANGE > 0.42 NM — USE CFAR DETECTION MODEL.
20 CALL CFAR
RETURN
END

C THIS SUBROUTINE ADDS THE EQUIVALENT NOISE TO THE ANGLE, RANGE, VELOCITY AND ON-TARGET DISCRIMINANT COMPONENTS AND THEN COMPUTES THE ANGLE, RANGE, VELOCITY, AND ON-TARGET DISCRIMINANTS.

C SUBROUTINE DISCRM
REAL LATE, MEAN
COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SRDOT,SPANG,SRANG,SPRTE,
2 SRTE,SRSS,MDVF,MRDVF,MRDDVF
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
COMMON /ICNTL/I3DUM(14),MRNG,MSAM,MPRF,IDUM4(16)
COMMON /SYSDAT/TSAM,DR(3),CP,PS,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GA,
2 DUMS(3)
COMMON /TGTDAT/NT,DUMS(506),CORNQ,CGVEL
COMMON /DISCRM/AZDISC,ELDISC,RDISC,VRDISC,RRTE,ODISC,SIGBRI,SNRD,
2 SIGDB
COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
2 DF2,DF4,SIGBAR
COMMON /NOISE/NS1,NS2,NM(10),GAUS(320)
COMMON /AGCDAT/AGCO,AGCD,DB,SRDT,SNRDT
DIMENSION NFREQ(2),PDIA(2),PDIR(2),PDIV(2),PS(10,2),BN(2),PT(3)
2 TDC(3)

DIMENSION QNV(2)

C PS AND QNV CONSTANT CHANGES FEB 17, 1986 BY M. MEYER

DATA NFREQ/1.5/,BN/9772.4,616.6/
DATA PS/0.94.2,5,4.2,2,4,8,8,16/
2 ,PDIA,PDIR,PDIV/1.4142,3.1623,2,0,4.4721,2,8284.6,3246/
3 PT/42658.,3125.,195.3/
DATA QNV/0.0057,0.001/
DATA TDC/0.0512218,0.1195161,0.2561557/
C NOTE: DEBUGGING PRINT STATEMENTS.

B-31
**STEP 1**: Compute constant used in signal scaling and computation of noise statistics.

\[
\text{TC}_{\text{MO}} = \left(\frac{\text{TSAM}}{\text{TDC(MPRF)}}\right)^{0.5}
\]

**STEP 1-1**: Compute constant (note: it is different for active and passive modes).

```FORTRAN
IF(IMODE.EQ.2) GO TO 5
YY = GA/PS(MRNG,IMODE)/(CGRNGE*2*BN(MSAM))
S1 = YY/FLOAT(NFREQ(IMODE))
GOTO 10
```

**NOTE**: This is the constant used in passive mode.

**STEP 1-2**: Compute peak signal power to average thermal noise power at doppler filter output.

```FORTRAN
SNRDT = YY*SIGBAR
S1 = YY/FLOAT(NFREQ(IMODE))
```

**STEP 1-3**: Compute peak signal power to total (thermal plus quantization) noise power at the doppler filter output.

```FORTRAN
CALL SATNSE(SNF)
XX = SNF*AGCO
XX = XXI(XX+ONV(_))
SI = SI*XX
YY = YY*SIGBAR
SNRD = YY*SIGBAR
```

**STEP 1-4**: Update noise sequence.

```FORTRAN
NN(1) = MOD(NN(1)+1,320)+1
DO 15 I=2,10
15 NN(I) = MOD(NN(I-1)+29,320)+1
ID1 = NN(1)
GAUSS(ID1) = ANORM(NS1,NS2)
```

**STEP 2**: Compute angle discriminant (includes noise)
STEP 2-1: CHECK ANTENNA STEERING MODE — SKIP STEP 2 IF IN GPC-DES OR MANUAL.

STEP 2-2: COMPUTE ANGLE DISCRIMINANT COMPONENT SCALE FACTOR. ASCALE=S1+PDIA(IMODE)

STEP 2-3: COMPUTE STATISTICS OF ADDITIVE NOISE FOR ANGLE DISCRIMINANT COMPONENTS.

STEP 2-4: ADD EQUIVALENT NOISE TO ANGLE DISCRIMINANT COMPONENT SIGNALS.

STEP 2-5: COMPUTE AZ AND EL DISCRIMINANT COMPONENTS.

STEP 3-1: COMPUTE RANGE DISCRIMINANT (INCLUDES NOISE)

STEP 3-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR RANGE DISCRIMINANT.

STEP 3-3: ADD EQUIVALENT NOISE TO RANGE DISCRIMINANT COMPONENT SIGNALS.

STEP 3-4: COMPUTE RANGE DISCRIMINANT.

STEP 4-1: COMPUTE VELOCITY DISCRIMINANT (INCLUDES NOISE)

STEP 4-2: COMPUTE STATISTICS OF ADDITIVE NOISE FOR VELOCITY DISCRIMINANT COMPONENTS.
MEAN=PDIV(IMODE)

STEP 4-3: ADD EQUIVALENT NOISE TO VELOCITY DISCRIMINANT COMPONENT SIGNALS.

STEP 4-4: COMPUTE VELOCITY DISCRIMINANT.

STEP 5-1: COMPUTE STATISTICS OF ADDITIVE NOISE FOR OUTER DOPPLER FILTER SIGNALS.

STEP 5-2: ADD EQUIVALENT NOISE TO OUTER DOPPLER FILTER SIGNALS.

STEP 5-3: COMPUTE ON-TARGET DISCRIMINANT.

NOTE: THE FACTOR OF SORT(2.) IS DUE TO THE METHOD OF NORMALIZATION OF DISCRIMINANT COMPONENTS.

NOTE: DEBUGGING PRINT STATEMENTS.

THIS FUNCTION COMPUTES THE DOPPLER FILTER OUTPUT AMPLITUDE AND PHASE FOR AN INPUT SIGNAL OF FREQUENCY X.

COMPLEX FUNCTION DOPFIL(X)

INITIALIZE DENOM, NUMER.

CHECK FOR DENOMINATOR EQUAL TO ZERO.

RETURN

B-34
FUNCTION DPAT(X)
IF(ABS(X).GT.1.E-4) GO TO 10
DPAT=-0.6228*X
RETURN
10 Y=93.80*X
DPAT=1.1465*(Y*COS(Y)-SIN(Y))/(Y*Y)
RETURN
END

THE FUNCTION GIVES THE ANTENNA DIFFERENCE PATTERN WEIGHTING OF THE RADAR SIGNAL FOR THE GIVEN ANGLE (IN RADIANS) OFF BORESIGHT.

NOTE: THIS PATTERN IS THE DERIVATIVE OF THE SUM PATTERN

EXECUTIVE PROGRAM: INTERFACE WITH PARENT SIMULATION

SUBROUTINE EXEC
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
COMMON /OUTPUT/MSWF,MTF,MSF,DUM(7),I 
COMMON /ICNTL/IOLDPW,IOLDMD,IOLDSM,ISHOLD,KMSCLK,KWMUP,IDUM1(3),MTP,IDUMS(17)
2 DATA DATINT/1.0/
   KWMUP=1

STEP 0: INITIALIZE ALL TARGET AND SYSTEM DATA

IF(DATINT.NE.1.0) GO TO 1
   CALL SETIT
   CALL DATA
   CALL SYSINT
   IOLDPW=IPWR
***STEP 1: CHECK SYSTEM POWER SWITCH***

IF(IPWR.GT.1) GO TO 5

***STEP 2: CHECK SYSTEM MODE SWITCH***

IF(IMODE.LT.3) GO TO 7

***STEP 3: DETERMINE WHETHER SYSTEM IN STANDBY***

IF(IPWR.GT.2) GO TO 15

***STEP 4: DETERMINE WHETHER WARMUP PERIOD EXCEEDED***

IF(KMSCLK.GT.K_IUP) GO TO 20

***STEP 5: DETERMINE IF THERE HAS BEEN AN ANTENNA STEERING MODE CHANGE***

IF(IASM.EQ.OLDSM) GO TO 25

***STEP 5: DETERMINE WHETHER SYSTEM IS IN SEARCH AND ACQUISITION OR TRACK MODE***

IF(MTF.EQ.1.OR.MTP.EQ.1) GO TO 30

***STEP 5: DETERMINE WHETHER SYSTEM IS IN STANDBY AND ACQUISITION OR TRACK MODE***

IF(IPWR.GT.1) GO TO 30

***STEP 2: CHECK SYSTEM MODE SWITCH***

IF(IMODE.LT.3) GO TO 7

***STEP 3: DETERMINE WHETHER SYSTEM IN STANDBY***

IF(IPWR.GT.2) GO TO 15

***STEP 5: DETERMINE WHETHER SYSTEM IS IN SEARCH AND ACQUISITION OR TRACK MODE***

IF(MTF.EQ.1.OR.MTP.EQ.1) GO TO 30

***STEP 5: DETERMINE WHETHER SYSTEM IS IN STANDBY AND ACQUISITION OR TRACK MODE***

IF(IPWR.GT.1) GO TO 30

***STEP 2: CHECK SYSTEM MODE SWITCH***

IF(IMODE.LT.3) GO TO 7

***STEP 3: DETERMINE WHETHER SYSTEM IN STANDBY***

IF(IPWR.GT.2) GO TO 15

***STEP 4: DETERMINE WHETHER WARMUP PERIOD EXCEEDED***

IF(KMSCLK.GT.K_IUP) GO TO 20

***STEP 5: DETERMINE IF THERE HAS BEEN AN ANTENNA STEERING MODE CHANGE***

IF(IASM.EQ.OLDSM) GO TO 25

***STEP 5: DETERMINE WHETHER SYSTEM IS IN SEARCH AND ACQUISITION OR TRACK MODE***

IF(MTF.EQ.1.OR.MTP.EQ.1) GO TO 30

***STEP 5: DETERMINE WHETHER SYSTEM IS IN STANDBY AND ACQUISITION OR TRACK MODE***

IF(IPWR.GT.1) GO TO 30

***STEP 2: CHECK SYSTEM MODE SWITCH***

IF(IMODE.LT.3) GO TO 7

***STEP 3: DETERMINE WHETHER SYSTEM IN STANDBY***

IF(IPWR.GT.2) GO TO 15

***STEP 4: DETERMINE WHETHER WARMUP PERIOD EXCEEDED***

IF(KMSCLK.GT.K_IUP) GO TO 20

***STEP 5: DETERMINE IF THERE HAS BEEN AN ANTENNA STEERING MODE CHANGE***

IF(IASM.EQ.OLDSM) GO TO 25

***STEP 5: DETERMINE WHETHER SYSTEM IS IN SEARCH AND ACQUISITION OR TRACK MODE***

IF(MTF.EQ.1.OR.MTP.EQ.1) GO TO 30

***STEP 5: DETERMINE WHETHER SYSTEM IS IN STANDBY AND ACQUISITION OR TRACK MODE***

IF(IPWR.GT.1) GO TO 30
30 CALL TRACK
RETURN
END

* THIS SUBROUTINE GENERATES A (3X3) MATRIX TGA THAT PRODUCES
* A ROTATION OF GA RADIANS ABOUT THE Y-AXIS.

SUBROUTINE GAMMA(TGA,GA)
DIMENSION TGA(3,3)
DO 10 I=1,3
    DO 10 J=1,3
        TGA(I,J)=0.0
10    TGA(2,2)=1.0
    TGA(1,1)=COS(GA)
    TGA(1,3)=SIN(GA)
    TGA(3,1)=TGA(1,3)
    TGA(3,3)=TGA(I,I)
RETURN
END

* THIS FUNCTION CHECKS FOR NEGATIVE ARGUMENT FOR INT FUNCTION
* AND CORRECTS THE QUANTIZATION PROCEDURE.

REAL FUNCTION INTT(Y)
X=Y
IF(X.LT.0.0) X=X-1.0
INTT=AINT(X)
RETURN
END

* THIS SUBROUTINE MULTIPLIES THE (3X3) MATRIX A AND THE (3X1)
* VECTOR B TO OBTAIN THE (3X1) VECTOR C.

SUBROUTINE MULT31(A,B,C)
DIMENSION A(3,3),B(3),C(3)
DO 10 I=1,3
    C(I)=0.0
10    DO 10 J=1,3
    C(I)=C(I)+A(I,J)*B(J)
RETURN
END

* THIS SUBROUTINE MULTIPLIES THE (3X3) MATRIX A AND THE (3X3)
* MATRIX B TO OBTAIN THE (3X3) MATRIX C.

SUBROUTINE MULT33(A,B,C)
DIMENSION A(3,3),B(3,3),C(3,3)
DO 10 I=1,3
RETURN
END
DO 10 J=1,3
  C(I,J)=0.0
DO 10 K=1,3
  C(I,J) = C(I,J)+A(I,K)*B(K,J)
RETURN
END

C * THIS SUBROUTINE GENERATES A (3X3) MATRIX TPH THAT PRODUCES * A ROTATION OF PH RADIANS ABOUT THE Z-AXIS.

SUBROUTINE PHI(TPH,PH)
  DIMENSION TPH(3,3)
  DO 10 I=1,3
    TPH(3,I)=0.0
  TPH(3,3)=1.0
  TPH(1,1)=COS(PH)
  TPH(2,2)=TPH(1,1)
  TPH(1,2)=SIN(PH)
  TPH(2,1)=-TPH(1,2)
RETURN
END

C * THIS SUBROUTINE GENERATES A (3X3) MATRIX TPHD THAT REPRESENTS THE DERIVATIVE OF A MATRIX THAT REPRESENTS UNIFORM ROTATION ABOUT THE Z-AXIS. THE ROTATION SPEED IS W AND THE ANGLE AT WHICH THE DERIV. IS TAKEN IS PH.

SUBROUTINE PHID(TPHD,PH,W)
  DIMENSION TPHD(3,3)
  DO 10 I=1,3
    TPHD(3,I)=0.0
  TPHD(3,3)=1.0
  TPHD(1,1)=W*SIN(PH)
  TPHD(2,2)=TPHD(1,1)
  TPHD(1,2)=W*COS(PH)
  TPHD(2,1)=-TPHD(1,2)
RETURN
END

C * THIS SUBROUTINE UPDATES THE POSITION OF THE ANTENNA GIMBALS

SUBROUTINE POINT
  COMMON /OUTPUT/IDUM1(3),DUM4(2),SPAN,SRANG,DUM5(3),DUM2(4)
  COMMON /SYSDAT/TS,DUM(3),CG,SG,DUM2(9)
  COMMON /ATDAT/DUM1(4),SALRTE,SBTRTE,DUM3(3),AL,BT,PREF,RREF.
  AREF,BREF
  DATA AK/2.0/,TAU/1.414/,PI/3.141592653/

C * STEP 1: PRELIMINARY COMPUTATIONS

B-38
CR = COS(-RREF)
SR = SIN(-RREF)
CP = COS(-PREF)
SP = SIN(-PREF)

**STEP 2: COMPUTE ANTENNA REFERENCE ROLL/PITCH ANGLES IN THE RADAR FRAME.**

XX = CG*SP + SG*CR + CP
YY = SG*SP + CG*SR + CP
ZZ = CR*CP

IF (YY.EQ.0.0 .AND. ZZ.EQ.0.0) GO TO 1
AREF = ATAN2(YY, ZZ)
GO TO 2
1 IF (XX.GT.0.0) AREF = -PI/2.
IF (XX.LT.0.0) AREF = PI/2.
2 BREF = ASIN(XX)

**STEP 3: UPDATE OUTER (ALPHA) GIMBAL RATE AND POSITION.**

ERRA = AREF - AL
ERRB = BREF - BT

**STEP 4: UPDATE INNER (BETA) GIMBAL RATE AND POSITION.**

ERRA = AREF - AL
ERRB = BREF - BT

**STEP 5: ANTENNA IN OBSCURATION REGION.**

CALL SCNWN

**STEP 6: COMPUTE ANTENNA ROLL/PITCH ANGLES IN THE BODY FRAME.**

CA = COS(AL)
SA = SIN(AL)
CB = COS(BT)
SB = SIN(BT)

XX = CA*SB + SG*SA + CB
YY = SG*SB + CG*SA + CB
ZZ = CA*CB

IF (YY.EQ.0.0 .AND. ZZ.EQ.0.0) GO TO 3
SRANG = 57.29576*ATAN2(YY, ZZ)

B-39
GO TO 4
3 IF (XX GT 0.0) SRANG = SRANG + 90.0
   IF (XX LT 0.0) SRANG = SRANG - 90.0
4 SPANG = 57.29576 * ASIN(XX)
   IF (SRANG LE 0.0) GO TO 10
   SRANG = (180.0 - ABS(SRANG)) * (SRANG / ABS(SRANG))
10 RETURN
END

**********************************************************************
* THIS SUBROUTINE COMPUTES TARGET C.G. POSITION AND VELOCITY WRT * 
* ANTENNA LOS COORDINATES AND INDIVIDUAL SCATTERER POSITIONS AND VELOCITIES WRT ANTENNA LOS COORDINATES. *
**********************************************************************

SUBROUTINE PVTRAN

COMMON /TEST1/RA(3)
COMMON /CNTL/IPWR,IMODE
COMMON /INPUT/ERT(3),EVT(3),DUM(21)
COMMON /OUTPUT/MSWF,MTF,MSF,DUMO(7),IDUMO(4)
COMMON /CNTL/IDUM6(9),MTP,IDUM7(3),MTKINT
COMMON /SYSDAT/TSAN,DR(3),DUM2(11)
COMMON /TGTDAT/NT,RAU(3,1e6),RANGE(1e6),RADVEL(1e6),RO(3),
   CGRGNE,CGVEL
2 COMMON /SATDAT/RADAR(3),N2e,RT(7e3),SIG(70),ROLD,ICLOSE,ICLOLD
COMMON /TARGET/ITARG,SRCS
DIMENSION ROR(3),ROD(3),VI(3),RL(3),RAD(3),RLD(3),XRT(3)

*************************************************************************
* STEP 1: COMPUTE TARGET C.G. POSITION IN ANTENNA LOS FRAME *
*************************************************************************

C STEP 1-1: ADD RADAR OFFSET IN ORBITER BODY FRAME.
   DO 5 I = 1, 3
   5 ROR(I) = ERT(I) - DR(I)

C STEP 1-2: TRANSFORM TARGET C.G. POSITION FROM BODY FRAME TO 
    ANTENNA LOS FRAME.
   CALL MLJT31(TLB,ROR,RO)

C STEP 1-3: COMPUTE RANGE OF TARGET C.G. WRT RADAR.
   CGRNGE = SQRT(RO(1)**2 + RO(2)**2 + RO(3)**2)

C STEP 1-4: COMPUTE UNIT VECTOR IN DIRECTION OF TARGET C.G. WRT 
    ANTENNA LOS FRAME.
   DO 10 I = 1, 3
   10 ROD(I) = RO(I) / CGRNGE

*************************************************************************
* STEP 2: COMPUTE TARGET C.G. RADIAL VELOCITY WRT ANTENNA LOS FRAME (OR RADAR).
*************************************************************************

C STEP 2-1: COMPUTE TARGET C.G. VELOCITY COMPONENTS WRT ANTENNA LOS FRAME.
   CALL MLJT31(TLBD,ROR,RO)
   CALL MLJT31(TLB,EVT,ROD)
   DO 15 I = 1, 3
   15 ROD(I) = ROD(I) + VI(I)

B-40
STEP 2-2: COMPUTE TARGET C.G. RADIAL VELOCITY WRT ANTENNA LOS.
CGVEL=0.0
DO 20 I=1,3
20 CGVEL=CGVEL+ROD(I)+ROU(I)

STEP 3: COMPUTE TARGET SCATTERING CHARACTERISTICS — OF — ILLUMINATED POINTS, THE POINT LOCATIONS, AND THE RCS FOR EACH POINT.

STEP 3-1: IF IN ACTIVE MODE, SEARCH MODE, OR TRACKER INITIALIZATION
ASSUME SINGLE SCATTERER LOCATED AT TARGET FRAME ORIGIN.

ITARG=0 POINT TARGET
ITARG=1 SPAS
ITARG=2 SMM
IF(ITARG.EQ.0) GO TO 24
CHECK CONDITION.
IF(IMODE.NE.1.AND.MTKINT.NE.0.AND.MTP.NE.0) GO TO 30
IF ABOVE CONDITION TRUE — THEN SET PARAMETERS AS FOLLOWS AND DO NOT CALL TARGET MODEL.
NT=1
SIG(1)=SRC
DO 25 I=1,3
25 SIG(I)=SRC(I)
25 RT(1,I)=0.0
STEP 3-2: COMPUTE LOCATION OF RADAR IN TARGET FRAME.
30 DO 35 I=1,3
35 RADAR(I)=0.0
DO 35 J=1,3
35 RADAR(I)=RADAR(I)-TILT(J,I)*RT(K,I)
IF(ITARG.EQ.0)GO TO 40
STEP 3-3: COMPUTE TARGET SCATTERING CHARACTERISTICS.
IF(ITARG.EQ.2)CALL SMM
IF(ITARG.EQ.1)CALL SPAS
NT=N20
STEP 4-1: COMPUTE KTH SCATTERER POSITION WRT ANTENNA LOS FRAME.
DO 45 J=1,3
45 RL(J)=0.0
DO 45 I=1,3
45 RL(J)=RL(J)+TILT(J,I)*RT(K,I)
DO 50 I=1,3
50 RA(I)=RO(I)+RL(I)
STEP 4-2: COMPUTE RANGE OF KTH SCATTERER WRT RADAR.
RANGE(K)=SQRT(RA(1)+RA(2)+RA(3))
STEP 4-3: COMPUTE UNIT VECTOR IN DIRECTION OF KTH SCATTERER WRT ANTENNA LOS FRAME.
DO 55 I=1,3
55 RAU(I,K)=RA(I)/RANGE(K)
C  * STEP 5: COMPUTE KTH SCATTERER RADIAL VELOCITY WRT RADAR *
C  ******************************************
C 
C STEP 5-1: COMPUTE KTH SCATTERER VELOCITY COMPONENTS WRT ANTENNA LOS FRAME.
C 
DO 58 I=1,3
58 XRT(I)=RT(K,I)
CALL MULTI31(TLTD,XRT,RLD)
DO 60 I=1,3
60 RAD(I)=ROD(I)+RLD(I)
C 
C STEP 5-2: COMPUTE KTH SCATTERER RADIAL VELOCITY WRT TO RADAR.
C  RADVEL(K)=RADVEL(K)+RAD(I)*RAU(I,K)
DO 65 I=1,3
65 RADVEL(K)=RADVEL(K)+RAD(I)=RAU(I,K)
70 CONTINUE
C 
C NOTE: DEBUGGING PRINT STATEMENTS.
C 
C WRITE(6,991) RO(1),RO(2),RO(3),CGRNGE,CGVEL
C WRITE(6,992) RAU(1,1),RAU(2,1),RAU(3,1),RANGE(1),RADVEL(1)
C WRITE(6,993)I,(RT(I,J),J=1,3),SIG(I),I=1,N2E
C 
900 FORMAT(// 'RO1,RO2,RO3,CGR,CGV=',9(2F10.2))
901 FORMAT( 'RAU1,RAU2,RAU3,R,V=',5F10.2)
902 FORMAT( 'SPAS RCS DATA:/',1/,'9X','I','4X','R(I,1)',4X,'R(I,2)',4X,'R(I,3)',9X,'SIG(I)',/)
903 FORMAT(10,3F10.2,F15.1)
RETURN
END
C
C  FUNCTION RNDU(IRAN)
C  DATA MU/524287/,IETA/997/
C IF(IRAN.EQ.0) GO TO 10
C IRAN=IETA*IRAN
C IKEEP=IRAN/MU .
C IRAN=IRAN-IKEEP*MU
C XRAN=IRAN
C XRAN=XRAN/MU
C RNDU=XRAN
C 10 RETURN
C END
C
C  THIS SUBROUTINE COMPUTES THE RADAR SIGNAL STRENGTH AND UPDATES THE AGC SETTING.
C  
SUBROUTINE RSS
COMMON /CNTL/IPWR,IMODE,IDUM1(7),DUM1(3)
COMMON /ICNTL/IDUM2(14),HRNG,WSAM,IDUM6(11)
COMMON /OUTPUT/IDUM7(3),DUM3(6),SRSS,IDUM4(4)
COMMON /AGCODAT/AGCO,AGCODB,SNRDT,SNRDTD
DIMENSION PS(10,2),QNV(2),AI(2)
DATA PS/94.2,54.2,4.8,8.816./
DATA QNV/00007.,011./A1./0321,.51/
C 
C SUBROUTINE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
C DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
C KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986

C ****************************************
C * STEP 1: UPDATE SYSTEM AGC *
C ****************************************
C
C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.

AGCERR=AI(MSAM)*4.*PS(MRNG,IMODE)/(AGCO=(SNRDT+I.e)+QNV(MSAM))
IF(AGCERR.GT.18.) AGCERR=18.e
IF(AGCERR.LT.-e.1) AGCERR.,e.1

C STEP 1-2: COMPUTE NEW AGC VALUE AND CHECK LIMITS.

AGCO=AGCERR+AGCO

C ****************************************
C * STEP 2: UPDATE RADAR SIGNAL STRENGTH VALUE *
C ****************************************
C
C SRSS=1./AGCO
RETURN
END

C SUBROUTINE RTRACK
REAL INTT,IRDISC,IRNG,IRDOT
COMMON /CNTL/ZPWR,IMODE, IDUMC(7),DUMC(3)
COMMON /OUTPUT/IDLMB(3),SRNG,SRDOT,DUM2(5),IDUM(4)
COMMON /ICNTL/I1DUM(14),MRNG,MSAM,MPRF,IMODE(10),MRFOLD
COMMON /SYSDAT/TSAM,DUMS(14)
COMMON /RTDAT/IRDOT,IRNG,RBIAS,VEST(4),MDF(5)
COMMON /DSCRM/DUM3(3),RDSC,VDISC,RTTE,ODISC,DUM(14)
DIMENSION RTI(le,2),RT2(le,2),TDC(3),RGBIAS(2)
DATA RT1/9.e.125,e.25,_.e.125,2.,1.,2.e.5,e.25/,RT2/9=e.5,_.e,4.e.5,8.,8.,_.16./

C STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *

STEP 1-1: INTEGERIZE RANGE DISCRIMINANT AND CHECK FOR SATURATION.

IRDISC=INTT(RDISC+TCON+e.5)/TCON
IF(IRDISC.GT.127.) IRDISC=127.
IF(IRDISC.LT.-128.) IRDISC=-128.

C STEP 1-2: COMPUTE ROUGH RANGE RATE PREDICTION FROM ALPHA-BETA TRACKING EQUATIONS.
DEFINITION: RT1(MRNG,IMODE) CORRESPONDS TO BETA IN ALPHA-BETA TRACK.
RR1=IRDISC+RT1(MRNG,IMODE)
IRDOT=IRDOT+INTT(RR1+0.5)

STEP 2: UPDATE RANGE ESTIMATE

STEP 2-1: UPDATE RANGE ESTIMATE USING ALPHA-BETA TRACKER EQUATIONS.
DEFINITION: RT2 CORRESPONDS TO ALPHA IN ALPHA-BETA TRACKER.
R1=IRDISC+RT2(MRNG,IMODE)
IRNG=IRNG+IRDOT+INTT(R1+0.5)

STEP 2-2: CONVERT RANGE ESTIMATE (IRNG) TO FEET USING THE FACT THAT
THE LSB OF IRNG REPRESENTS 5/16 FEET.
RNG=0.3125*IRNG

STEP 2-3: ADD FIXED BIAS TO FINAL RANGE ESTIMATE.
SRNG=RNG+RGBIAS(MSAM)

FORCE BREAK TRACK IF RANGE LESS THAN 100 FT
IF(SRNG.LT.100.)CALL SYSINT
RETURN
END

THIS SUBROUTINE DETERMINES WHETHER THE SIGNAL PLUS NOISE
IS SATURATING THE A/D --- IF SO, THEN THE SNR AT DOPPLER
FILTER OUTPUT IS LIMITED TO THE VALUE THAT JUST SATUR-

SUBROUTINE SATNSE(SNF)
COMMON /CNTL/IPWR,,IMODE
COMMON /ICNTL/IDUM(14),MRNG
COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
DIMENSION PS(1B,2)

PS VALUES WERE UPDATED FEB 17, 1986 BY M. MEYER———
DATA PS/9.4.0.2..5=4.2..4..8.,8.,16. /
SNF=1.
X=AGCO*(SNRDT/(4.*PS(MRNG,IMODE))+1.0)

X=12.25/X WAS REPLACED BY X=6.25/X TO MORE ACCURATELY
REFLECT A/D SATURATION BY M. MEYER FEB 17, 1986

X=6.25/X
IF(X.GT.1) RETURN
SNF=X
RETURN
END

THIS SUBROUTINE CYCLES THRU THE LOGIC FOR ANY SCAN GENERATION.

SUBROUTINE SCAN
COMMON /CNTL/IDUM(4),ISRCHC,ISRCHG,IDUMC(3),EDRNG,DUMC(2)
**B-45**

```
COMMON /OUTPUT/MSWF,MTF,MSF,DUM1(7),IDUM2(4)
COMMON /CNTL,IDUM3(5),KSNCLK,IDUM4(2),MTP,IDUM5(17),MSWTCH,
2 KSN,IAROLD,ITROLD
COMMON /SYSDAT/TSAM,DUMS(14)
COMMON /TGTDAT/NT,DUM5(563),ROU(3),DUM3(2)
COMMON /ATDAT/DUM4(8),AL.BT,DUM5(2),AREF,BREF
DIMENSION TIMINT(31),ANGINT(31),RSW(le),TSW(le)
DATA TIMINT/.7,1.4,1.9,2.6,3.4,4.3,5.1,6.,7.,8.,9.1,10.4,11.8,
1 13.3,14.9,16.9,18.9,21.1,23.4,25.9,28.6,31.5,33.6,36.9,39.8,
2 43.2,46.8,50.3,54.3,58.4,60.8/.
DATA ANGINT/0.,7.,1.5,2.,2.3,6.4,4.5,2.6,1.7,7.9,8.9,9.8,18.9,
1 11.9,13.0,14.2,15.3,16.5,17.6,18.8,19.9,21.1,22.2,23.4,24.5/,.
2 25.6,26.7,27.8,28.9,30.8/.
DATA TSW/60.0,54.3,43.2,33.5,28.6,21.1,14.9,11.8,8.8,6.8/.
2 RSW/48699.2,55908.6,62584.3,71698.6,91142.5,151903.8,
3 243846.8,394949.8,881e41.8,1822845.8/
PI=180.0/3.141592653

C

C * STEP 1: DETERMINE WHETHER TO PERFORM SCAN INITIALIZATION(MSF=0) OR SCAN UPDATE(MSF=1).

C

C * STEP 2: PERFORM SCAN INITIALIZATION.

C INITIALIZE ALL FLAGS.
MSF=1
C INITIALIZE RING MONITORS.
IAROLD=0
ITROLD=10
C INITIALIZE SCAN CLOCK.
KSNCLK=0
C INITIALIZE SCAN TIME PARAMETER.
KSN=0
C

C DETERMINE SWITCH POINT PARAMETER.
DO 5 I=1,31
IF(EDRNG.LT.RSW(I)) GO TO 10
5 CONTINUE
10 MSWTCH=I

C

C * STEP 3: UPDATE SCAN CLOCKS.

C * STEP 3-1: UPDATE SCAN CLOCK (TRACKS TOTAL ELAPSED TIME FROM SCAN INITIATION).
15 KSNCLK=KSNCLK+1
T=FLOAT(KSNCLK)*TSAM
C

C STEP 3-2: UPDATE SCAN TIME PARAMETER (USED TO DETERMINE BORESIGHT POSITION IN SCAN PATTERN).
IF(T.LE.TSW(MSWTCH)) KSN=KSN+1
IF(T.GT.TSW(MSWTCH)) KSN=KSN-1
TSN=FLOAT(KSN)*TSAM
C

C * STEP 4: DETERMINE ANTENNA POSITION TO NEAREST SCAN RING.
DO 25 I=1,31
IF(TSN.LT.TIMINT(I)) GO TO 25
25 CONTINUE
```
25 IARNG=1

* STEP 5: DETERMINE TARGET POSITION IN SCAN PATTERN (SCAN RING NUMBER FOR TARGET)

* STEP 5-1: DETERMINE TARGET POSITION EXACTLY.
ALOLD=AL
BTOLD=BT
AL=AREF
BT=DEF
CALL TRNSFM
CALL PVTRAN
AL=ALOLD
BT=BTOLD

* STEP 5-2: DETERMINE TARGET SCAN RING NUMBER.

C DETERMINE TARGET ANGLE OFF SCAN DESIGNATES (DEGREES). 
CCCDDDDDDDDDDDDDDDDDDDDD MOD MAR 24 1983 CCCDDDDDDDDDDDDDDDDDDDDD
CGANG=ACOS(-ROU(3))=PII

C DETERMINE TARGET SCAN RING NUMBER.
DO 30 I=1,31
IF(CGANG.LT.ANGINT(I)) GO TO 35
30 CONTINUE
35 ITRNG=I
IF(CGANG.GT.3e.) ITRNG=32

* STEP 6: DETERMINE IF A DETECTION SHOULD BE ATTEMPTED *

C STEP 6-1: CHECK CONDITION.
IF(IARNG.EQ.ITRNG.AND.IAROLD.NE.ITROLD) CALL DETECT
C STEP 6-2: UPDATE RING NUMBER MONITOR.
IAROLD=IARNG
ITROLD=ITRNG

C STEP 7: CHECK FOR SCAN TERMINATION CONDITIONS *

C STEP 7-1: CHECK ALL POSSIBLE TERMINATION CONDITIONS.
C CONDITION = 1: T > 60. SECONDS"
IF(T.GE.60.) GO TO 40
C CONDITION = 2: NEXT SCAN TIME PARAMETER < 0. " 
ITEMP=KSN-1
IF(ITEMP.LT.0.) GO TO 40
C CONDITION = 3: DETECT A TARGET"
IF(MTP.EQ.0) RETURN
C STEP 7-2: PERFORM SCAN TERMINATION STEPS — IF TERMINATION COND 
ATION OBTAINED.
40 MSF=0
KSNCLK=0
KSN=0
ISRCHG=0
ISRCHC=0
RETURN
END

* THIS SUBROUTINE DETERMINES WHETHER THE ANTENNA IS IN THE OBSCURATION ZONE AND SETS THE SCAN WARNING FLAG APPROPRIATELY.

SUBROUTINE SCNWRN
COMMON /OUTPUT/MSWF,IDUM0(2),DUMO(7),IDUM01(4)
COMMON /ATDAT/DUM(8),A,B,DUMA(4)
DIMENSION ICLEAR(36,72)
DATA ICLEAR/17,1,13,0,6,1,18,1,12,0,6,1,19,1,11,0,6,1,20,1,10,0,6,1,
20,1,10,0,6,1,20+1,10,0,6,1,20,1,10,0,6,1,20,1,10+0,
6+1,20+1,10+0,6,1,19+1,11+0,6,1,18+1,12+0,6,1,17+1,13+0,
6+1,16+1,14+0,6,1,15+1,15+0,6,1,14+1,16+0,6,1,14+1,16+0,
6+1,13+1,17+0,6,1,12+1,18+0,6,1,11+1,19+0,6,1,10+1,29+0,6+1,
7+1,21+0,6,1,2,22+0,6,1,22+0,6,1,22+0,6,1,22+0,6,1,22+0,6,1,
8+4,1,26+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,
4+1,25+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,4+1,26+0,6,1,
A+4,1,25+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,4+1,26+0,6,1,
B+4,1,25+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,4+1,26+0,6,1,
C+4,1,25+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,4+1,26+0,6,1,
D+4,1,25+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,4+1,26+0,6,1,
E+4,1,25+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,4+1,26+0,6,1,
F+4,1,25+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,4+1,26+0,6,1,
G+4,1,25+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,4+1,26+0,6,1,
H+4,1,25+0,6,1,4+1,26+0,6,1,4+1,25+0,6,1,4+1,26+0,6,1,4+1,26+0,6,1/

ALPHA=A*57.3
BETA=B*57.3
IF(ABS(BETA).LE.90.) GO TO 1
BETA=-(180.-ABS(B))*B/ABS(B))
ALPHA=(180.-ABS(A))*A/ABS(A)
1 CONTINUE
IA=INT((ALPHA+180.)/5.+1.)
IB=INT((90.-BETA)/5.+1.)
MSWF=ICLEAR(IB,IA)
RETURN
END

* THIS SUBROUTINE COMPUTES THE RESPONSE TO ALL DISPLAYS AND CONTROLS WHEN THE RADAR IS IN ANY OF THE SEARCH MODES.

SUBROUTINE SEARCH
COMMON /CNTL/IDUM(3),IASM,ISRCHC,ISRCHG,IAZS,IELS,ISLR,EDRNG.
COMMON //EDPA,EDRA
2 COMMON /OUTPUT/MSWF,MTF,MSF,SRNG,SROOT,SPANG,SRANG,SPRTE.
2 COMMON /SRTE,SRSS,IDUM(4)
COMMON /CNTL/IOLDPW,IOLDMD,IOLDSM,ISSHOLD,KMSCLK,KWMP,KSNCLK.
2 COMMON /KSNMAX,KACCLK,MTP,MZI,MZ0,MSS,MTKINT,MRNG,MSAM,MPRF.
3 COMMON /IDUM(16)
COMMON /SYSDAT/TS,DUMS(14)
COMMON /ATDAT/DUM(18),PREF,REF,DUMA(2)
DIMENSION SLWRTE(2)
DATA SLWRTE/*6.9814E-3,3.4907E-1*/
**DETERMINE ANTENNA STEERING MODE.**

GO TO (10,20,30,40). IASM

**GPC-ACQ SEARCH AND ACQUISITION MODE.**

**GPC-DES SEARCH AND ACQUISITION MODE.**

**DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN**

IF(MSF.EQ.1) GO TO 14
IF(MZ1.EQ.1.AND.ISRCHG.EQ.1) GO TO 14

**STEP 2: PERFORM GIMBAL POINTING SEQUENCE**

**STEP 2-1: UPDATE ROLL/PITCH REFERENCES**

IF(ISHOLD.EQ.1.AND.ISRCHG.EQ.1) GO TO 12
RREF=EDRA
PREF=EDPA

**STEP 2-2: UPDATE POSITION OF GIMBALS.**

**STEP 2-3: DETERMINE WHETHER BORESIGHT IN ZONE I AND/OR ZONE 0 AND TAKE APPROPRIATE ACTION.**

**STEP 3: CHECK FOR TARGET DETECTION — IF IN ZONE 0**

**STEP 4: PERFORM SCAN SEQUENCE**

**DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN**

IF(MSF.EQ.1) GO TO 14
IF(MZ1.EQ.1.AND.ISRCHG.EQ.1) GO TO 14

**STEP 1: DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN**

**STEP 1-1: UPDATE ROLL/PITCH REFERENCES**

IF(ISHOLD.EQ.1.AND.ISRCHG.EQ.1) GO TO 12
RREF=EDRA
PREF=EDPA

**STEP 1-2: UPDATE POSITION OF GIMBALS.**
CALL POINT

STEP I-3: DETERMINE WHETHER BORESIGHT IN ZONE 1 AND/OR ZONE 0 AND TAKE APPROPRIATE ACTION.

CALL ZONECK

IF BORESIGHT NOT IN ZONE 0, THEN TARGET DETECTION NOT ALLOWED.

IF(Z0.EQ.0) RETURN

************************************************************************************

* STEP 1: CHECK FOR TARGET DETECTION — IF IN ZONE 0. *

************************************************************************************

CALL DETECT

RETURN

************************************************************************************

----------------- AUTO SEARCH AND ACQUISITION MODE -------------------------------

************************************************************************************

* STEP 1: DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN *

30 IF(ISRCHE.EQ.1) GO TO 32

************************************************************************************

* STEP 2: PERFORM GIMBAL POINTING SEQUENCE *

************************************************************************************

STEP 2-1: UPDATE ROLL/PITCH REFERENCE ANGLES.

CALL POINT

STEP 2-3: DETERMINE SLEW RATE AND TAKE APPROPRIATE ACTION.

IF SLEW RATE IS GREATER THAN 0.4 DEG/SEC, THEN TARGET DET-

IF(ISLR.GT.0) RETURN

************************************************************************************

* STEP 3: CHECK FOR TARGET DETECTION — IF SLEW RATE <0.4 DEG • PER SECOND. *

CALL DETECT

RETURN

************************************************************************************

STEP 4: PERFORM SCAN SEQUENCE *

CALL SCAN

RETURN

************************************************************************************

----------------- MANUAL SEARCH AND ACQUISITION MODE -------------------------------
STEP 1-1: UPDATE ROLL/PITCH REFERENCE ANGLES.
40 PREF=PREF+FLOAT(ELS)*SLWRTE(ISLR+1)*TS
     RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TS

STEP 1-2: UPDATE POSITION OF GIMBALS.
CALL POINT

STEP 1-3: DETERMINE SLEW RATE AND TAKE APPROPRIATE ACTION.
IF(ISLR.GT.0) RETURN

CALL DETECT
RETURN
END

* THIS SUBROUTINE GENERATES THE NOISE-FREE ANGLE, RANGE, VELOCITY •
* AND ON-TARGET DISCRIMINANT COMPONENTS.

SUBROUTINE SIGNAL
REAL IRDOT,IRNO
COMMON /CNTL/II_,IMODE,ITXP,IASM,
       IDUMC(5),DU_(3)
COMMON /OUTPUT/I1DUM(3),SRNG,DUMI(6),IDUM2(4)
COMMON /ICNTL/IDUMS(13).MTKINT.MRNG,MSAM,MPRF.MBKTRK,MBTSM,
       MBT(8)
COMMON /TGDAT/NT,RAU(3,100),RANGE(100),RADVEL(100),RO(3),
       ROU(3),CRNGE,CGVEL
COMMON /SATDAT/RADAR(3),N20,RT(Te,3),SIG(70)
COMMON /RTDAT/IRDOT,IRNG,DUM2(5),MDF(5)
COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,DF1,DF5,
       DF2,DF4,SIGBAR
COMMON /XFORMS/TLB(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)
COMMON /SUDIPH/ X,Y,Z,PAZ,PEL
REAL CTP(10,2),DFWTS(5,100),ALAM(5),ALAMD(3),NFREQ(2)
COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF5,CDF2,CDF4,
       CDF3,DFWTS,PHASE,PHASE1,DOPFIL
DIMENSION CTP(10,2),DFWTS(5,100),ALAM(5),ALAMD(3),NFREQ(2)
DATA CTP/9,.O3318,9.799E-4,_,.O331B,1.9599E-3,9.8E-4,4.9E-4,
       2,2.45E-4,1.225E-4/
DATA NFREQ/1,5/,ALAM/177.3733,176.0447,178.7149,176.7089,
       2 178.0393/,ALAMD/1.27261E-2,2.96989E-2,3.99623E-1/
DATA LOOP/1/

MODIFIED JAN 10 1986 BY M. MEYER
MODIFICATIONS TO SUBROUTINE SIGNAL INCLUDE
CALCULATION OF THE AZIMUTH AND ELEVATION ANGLES
USE OF MEASURED ANTENNA PATTERNS INSTEAD
OF FUNCTIONS SPAT AND DPAT AND A
FACTOR IN THE DIFFERENCE CHANNELS SIGNAL
WHICH ACCOUNTS FOR THE FINITE WIDTH PHASE
TRANSITION IN THE REAL PHASE PATTERNS.
IF (ILOOP.NE.1) GO TO 11
CALL READPAT
PBAL=0.
ILOOP=0
CONTINUE

**STEP 1: PRELIMINARY COMPUTATIONS AND PARAMETER INITIALIZATION**

**STEP 1-1: INITIALIZE DISCRIMINANT COMPONENTS (NOTE: THESE ARE THE COMPONENT SIGNALS AFTER SQUARE-LAW DETECTION).**

SPAZ=0.0
SMAZ=0.0
SPEL=0.0
SMEL=0.0
EARLY=0.0
LATE=0.0
DF1=0.0
DF5=0.0
DF2=0.0
DF4=0.0
SIGBAR=0.0

DO 55 I=1,NFMAX

**STEP 1-2: INITIALIZE COMPLEX DISCRIMINANT COMPONENTS BEFORE EACH XMIT FREQUENCY (NOTE: THESE ARE THE COMPONENT SIGNALS BEFORE SQUARE-LAW DETECTION).**

CSUM=(0.,0.)
CDIFAZ=(0.,0.)
CDIFEL=(0.,0.)
CEARLY=(0.,0.)
CLATE=(0.,0.)
CDF1=(0.,0.)
CDF5=(0.,0.)
CDF2=(0.,0.)
CDF4=(0.,0.)
DO 45 K=1,NT
IF(I.GT.1) GO TO 35

**STEP 2: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR FOR KTH SCATTERER.**

**STEP 2-1: COMPUTE AZIMUTH AND ELEVATION ANGLE.**

AZ=ATAN2D(RAU(2,K),ABS(RAU(3,K))
EL=ATAN2D(RAU(1,K),ABS(RAU(3,K))))

**STEP 2-2: COMPUTE ANTENNA SUM, DIFFERENCE AND PHASE FACTORS.**

CALL INTERP(AZ,EL)

**STEP 2-3: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR.**

NOTE: IF IN ACTIVE MODE SET XX=1.0.
XX=SIG(K)*X
IF(IMODE.EQ.1) XX=1.0
S=XX*X
STEP 2-4: CHECK ANTENNA STEERING MODE (IF IN GPC-DES OR MANUAL SKIP STEP 4).

IF(IASM.EQ.2.0.IASM.EQ.4) GO TO 2e

STEP 3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION FACTORS FOR KTH SCATTERER.

STEP 3-3: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION FACTORS (INCLUDE RCS AND SUM PATTERN WEIGHTINGS) AND PHASE DIFFERENCE AND BALANCE WEIGHTINGS.

DAZ=XX+Y*CMPLX(COSD(PAZ+PBAL),SIND(PAZ+PBAL))
DEL=XX+Z*CMPLX(COSD(PEL+PBAL),SIND(PEL+PBAL))

STEP 4: COMPUTE RANGE GATE WEIGHTING FOR KTH SCATTERER.

DEFINITION: CTP=4./(C,PULSEWIDTH) WHERE C IS SPEED OF LIGHT.

STEP 4-1: COMPUTE RANGE GATE LOCATION WRT RANGE GATE CENTER.

SRNGX=le.,AINT(e.e3125=IRNG)
DELX=CTP(MRNG,IMODE).(RANGE(K)-SRNGX)

STEP 4-2: COMPUTE EARLY AND LATE RANGE GATE WEIGHTINGS FOR KTH SCATTERER.

II-INT((DELX+7.)/2.)
IF(II.LE.1) II=1
IF(II.GE.5) II=5
GO TO (21,22,23,24,21),II

STEP 4-3: COMPUTE RANGE GATE WEIGHT FOR NON-RANGE DISCRIMINANT COMPONENTS.

RGWGT=.5*(RGL+RGE)

STEP 4-4: APPLY RANGE GATE WEIGHTING TO SUM AND DIFFERENCE CHANNEL MULTIPLICATION FACTORS.

RGE=RG+RGWGT
RGL=RG+RGWGT
S=R+RGWGT
DAZ=DAZ+RGWGT
DEL=DEL+RGWGT

STEP 5: COMPUTE DOPPLER FILTER PHASE SHIFT AND WEIGHTING FOR KTH SCATTERER. NOTE: THIS CALCULATION IS INDEPENDENT OF XMIT FREQUENCY AND ASSUMES NO ACCELERATION OVER DATA CYCLE.
C DEFINITION: ALAMD(MPRF)=2.*PI/(PRF*LAMBDA) 00021540
C DEFINITION: THE CONSTANT 0.196348=PI/16. 00021550
C STEP 5-2: COMPUTE DOPPLER FREQUENCY CORRESPONDING TO RADIAL VELOCITY 00021570
C OF KTH SCATTERER.
FDT=-2.*ALAMD(MPRF)+RADVEL(K) 00021590
C STEP 5-3: COMPUTE DOPPLER FILTER WEIGHTING FOR EACH OF FIVE DOPPLER 00021610
C TRACKING FILTERS.
DO 30 J=1,5 00021620
ARG=0.196348+MDF(J)-FDT 00021630
30 DFWTS(J,K)=DOPFIL(ARG) 00021650
C STEP 6: COMPUTE PHASE FACTOR ASSOCIATED WITH KTH SCATTERER RANGE 00021670
C (NOTE: PHASE IS REFERENCED TO PHASE ASSOCIATED WITH RANGE 00021680
C OF TARGET C.G.)
35 DELPSI=ALAM(I)*(RANGE(K)-CGRANGE) 00021770
C STEP 6-1: COMPUTE PHASE REFERENCED TO TARGET C.G.
PHASE=CEXP(CMPLX(0.,DELPSI)) 00021800
C STEP 6-2: COMPUTE PHASE FACTOR, I.E. EXP(J*DELPHI).
PHASE=CEXP(CMPLX(0.,DELPSI)) 00021800
PHASE1=PHASE 00021800
C STEP 6-3: COMBINE RANGE PHASE FACTOR AND DOPPLER FILTER WEIGHT AND PHASE FACTOR.
PHASE=PHASE*DFWTS(3,K) 00021850
C STEP 7: ADD (VECTORIALLY) KTH SCATTERER CONTRIBUTION TO EACH 00021870
C DISCRIMINANT'S COMPONENT SIGNALS.
CSUM=CSUM+PHASE 00021900
C STEP 7-1: ADD KTH SCATTERER CONTRIBUTION TO SUM CHANNEL SIGNAL.
CSUM=CSUM+PHASE 00021930
C STEP 7-2: CHECK ANTENNA STEERING MODE — SKIP STEP 8-3 IF IN 00021950
C GPC-DES OR MANUAL MODE.
IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 40 00021970
C STEP 7-3: ADD KTH SCATTERER CONTRIBUTION TO AZ AND EL DIFFERENCE 00021990
C CHANNELS SIGNALS.
CDIFAZ=CDIFAZ+DAZ*PHASE 00022000
CDIFEL=CDIFEL+DEL*PHASE 00022020
C STEP 7-4: ADD KTH SCATTERER CONTRIBUTION TO RANGE DISCRIMINANT 00022040
C COMPONENT SIGNALS.
40 CEARLY=CEARLY+RGE*PHASE 00022060
C STEP 7-5: ADD KTH SCATTERER CONTRIBUTION TO VELOCITY DISCRIMINANT 00022080
C COMPONENT SIGNALS.
PHASE1=PHASE1*S 00022090
CDF2=CDF2+PHASE1*DFWTS(2,K) 00022110
CDF4=CDF4+PHASE1*DFWTS(4,K) 00022130
C STEP 7-6: ADD KTH SCATTERER CONTRIBUTION TO ON-TARGET DISCRIMINANT 00022140
C COMPONENT SIGNALS.
CDF1=CDF1+PHASE1*DFWTS(1,K) 00022150
CDF2=CDF2+PHASE2*DFWTS(2,K) 00022170
CDF5 = CDF5 + PHASE1 * DFWTS(5,K)
45 CONTINUE

--------------------------------------------------------------------------------------------------

C ** STEP 8: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET **
C ** DISCRIMINANT COMPONENTS AT ITH FREQUENCY AND SQUARE **
C ** LAW DETECT THESE COMPONENTS. **
--------------------------------------------------------------------------------------------------

C STEP 8-1: CHECK ANTENNA STEERING MODE — SKIP STEPS 9-2 AND 9-3
C IF IN GPC-DES OR MANUAL.
IF (IASM.EQ.2.0 OR IASM.EQ.4) GO TO 50

C STEP 8-2: COMPUTE AZ DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
SPAZ = SPAZ + CABS(CSUM + CDIFAZ)**2
SMAZ = SMAZ + CABS(CSUM - CDIFAZ)**2

C STEP 8-3: COMPUTE EL DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
SPEL = SPEL + CABS(CSUM + CDIFEL)**2
SMEL = SMEL + CABS(CSUM - CDIFEL)**2

C STEP 8-4: COMPUTE RANGE DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
EARLY = EARLY + CABS(CEARLY)**2
LATE = LATE + CABS(CLATE)**2

C STEP 8-5: COMPUTE VELOCITY DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
DF2 = DF2 + CABS(CDF2)**2
DF4 = DF4 + CABS(CDF4)**2

C STEP 8-6: COMPUTE ON-TARGET DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.
DF1 = DF1 + CABS(CDF1)**2
DFS = DFS + CABS(CDFS)**2

--------------------------------------------------------------------------------------------------

** STEP 9: COMPUTE EFFECTIVE CROSS-SECTION AVERAGED OVER PROPER **
** NUMBER OF TRANSMIT FREQUENCIES. **
--------------------------------------------------------------------------------------------------

SIGBAR = SIGBAR + CABS(CSUM)**2

--------------------------------------------------------------------------------------------------

NOTE: DEBUGGING PRINT STATEMENTS
WRITE(6,900) (1,SIG(I), I=1,NFREQ
900 FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',15,6F10.2,I5)

SUBROUTINE SINGLE
DIMENSION P(41)
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUM(5),DLIMC(3)
COMMON /OUTPUT/MSWF,MTF,MSF,DUM(7),IDLIMI(4)

B-54
**COMCON**

```
COMMON /ICNTL/IDUM2(8),KACCLK,MTP,IDUM3(5),MSAM,IDUM4(11)
COMMON /TGDAT/NT,DUM1(500),RO(3),ROU(3),CGRNGE,CGVLEL
COMMON /DETDAT/NSRCH,105
COMMON /TGTDAT/NTG,RO(3),ROU(3),CGRNGE,CGVLEL
DATA NSRCH/105/
DATA P/6=0,.001,.003,.008,.012,.015,.043,.053,.076,.107
     2 .147,.193,.244,.312,.363,.447,.514,.590,.644,.706,.765,.815,.861,.000
     3 .882,.918,.937,.955,.966,.976,.980,.989,.991,.997,.996/

C STEP 1: COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT

C STEP 1-1: SET SAMPLE RATE TO OBTAIN CORRECT NOISE BW IN SNRV COMP.
    MSAM=1
     IF (IMODE.EQ.1) MSAM=2
C STEP 1-2: COMPUTE NOMINAL SNR.
    SNR=SNRV(SIGMA,CGRNGE)
C
C STEP 2: IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNR -

C STEP 2-1: CHECK SCAN FLAG.
     IF(MSF.EQ.1) GO TO 1
C STEP 2-2: COMPUTE BEAMSHAPE LOSS — BASED UPON C.G. POSITION OFF BORESIGHT.
     BETAD=SPAT(CGANG)**2
C STEP 2-3: ADD BEAMSHAPE LOSS TO NOMINALV, I.E. COMPUTE ACTUAL SNR
    SNR=SNR+BETAD
C
C STEP 3: DETERMINE PROBABILITY OF DETECTION, PD, BASED UPON SNR

C STEP 3-1: DETERMINE INDEX TO ACCESS APPROPRIATE PD VERSUS SNR CURVE.
    1 IF(IMODE.EQ.2) GO TO 5
    NCRV=1
     GO TO 15
    5 IF(IASM.LT.3) GO TO 10
    NCRV=3
     GO TO 15
    10 NCRV=5

C ADJUST INDEX FOR SCANNING.
   15 NCRV=NCRV+MSF
C
C STEP 3-2: CONVERT SNRV TO DB.
    IF(SNR.LT.1.E-88) GO TO 20
    SNR=10 ** ALOG10(SNR)
     GO TO 25
   20 SNR=100
C
C STEP 3-3: SNR OUTSIDE (-30 DB, 0 DB) INTERVAL — IF SO, SET OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.
    IF SNR < -25 DB THEN SET PD=0.0 (DECLARE A MISS).
   25 IF(SNR.LT.-25.) GO TO 30
C
C
```
C IF SNR > -5 DB THEN SET PD=1.0 (DECLARE A HIT).
   IF(SNR.GT.-5.) GO TO 35
C
C STEP 3-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR
   INTERPOLATION.
   SCALE=(SNR+25.)*2.+1.000001
   ISNR=INT(SCALE)
   REMAIN=SCALE-FLOAT(ISNR)
C
C STEP 3-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION.
   PROB=P(ISNR)+REMAIN*(P(ISNR+1)-P(ISNR))
C
C * STEP 4: DETERMINE OUTCOME OF DETECTION ATTEMPT *
C
   X=RNDU(NSRCH)
   IF(X.LE.PROB) GO TO 35
C
C * STEP 5: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT *
C
C STEP 5-1: IF NO DETECTION — SET TARGET PRESENT FLAG LOW.
   30 WTP=0
   RETURN
C
C STEP 5-2: IF DETECTION SUCCESSFUL — SET TARGET PRESENT FLAG
   HIGH AND INITIALIZE ACQUISITION CLOCK.
   35 WTP=1
   KACCLK=0
   RETURN
END
C
C * THIS FUNCTION COMPUTES THE NOMINAL SNR AT THE VIDEO OUTPUT *
C * — IT ASSUMES NO BEAMSHAPE OR SCAN LOSS. *
C
FUNCTION SNRV(SIGMA,RANGE)
COMON /CNTL/IPWR.IMODE.ITXP.IDUMC(6).DUMC(3)
COMON /ICNTL/IDUM(12),MSS,MTKINT,MRNG,MSAM,MPRF,IDUM2(10)
COMON /SYSDAT/DUM(12),TGSTSIG,GPS,GAS
DIMENSION PT(4),BN(2)

FUNCTION SNRV(SIGMA,RANGE)
COMMON /CNTL/IPWR.IMODE.ITXP.IDUMC(6).DUMC(3)
COMMON /ICNTL/IDUM(12),MSS,MTKINT,MRNG,MSAM,MPRF,IDUM2(10)
COMMON /SYSDAT/DUM(12),TGSTSIG,GPS,GAS
DIMENSION PT(4),BN(2)

CC

FUNCTION SNRV(SIGMA,RANGE)
COMMON /CNTL/IPWR.IMODE.ITXP.IDUMC(6).DUMC(3)
COMMON /ICNTL/IDUM(12),MSS,MTKINT,MRNG,MSAM,MPRF,IDUM2(10)
COMMON /SYSDAT/DUM(12),TGSTSIG,GPS,GAS
DIMENSION PT(4),BN(2)

CC

FUNCTION SNRV(SIGMA,RANGE)
COMMON /CNTL/IPWR.IMODE.ITXP.IDUMC(6).DUMC(3)
COMMON /ICNTL/IDUM(12),MSS,MTKINT,MRNG,MSAM,MPRF,IDUM2(10)
COMMON /SYSDAT/DUM(12),TGSTSIG,GPS,GAS
DIMENSION PT(4),BN(2)

C

C * DETERMINE WHETHER ACTIVE OR PASSIVE MODE *
C
   IF(IMODE.EQ.1) GO TO 10
C
C * PASSIVE MODE VIDEO SNR CALCULATION *
C
   SNRV=GPS+PT(ITXP)+10.*ALOG10(SIGMA)-BN(WSAM)-40.*ALOG10(RANGE)
   RETURN

C * ACTIVE MODE VIDEO SNR CALCULATION *


SUBROUTINE SPAS
COMMON /SATDAT/RADAR(3).KTAR.R(7e.3).SIG(7e).ROLD.ICLOSE.ICLOLD
DIMENSION SIGMA(61).TARG(61.3).PHIMIN(61.3).PHIMAX(61.3)
DIMENSION OFFSET(61)
DIMENSION PHI(61.3)
DIMENSION VECT(3).COSPHI(61.3)
DIMENSION TTRAN(3)

Bisssas_.ssosswessssssset.lwss_tssIii_i_1_.i_os_s_s_._Is.

• DATA DEFINITION: INCLUDES SCATTERER LOCATION IN TARGET FRAME, MAXIMUM SCATTERER RCS VALUE, ANGULAR EXTENT OF NONZERO RCS, AND OTHER MISCELLANEOUS DATA REQUIRED BY THE ROUTINE.

• SEED FOR RANDOM NUMBER GENERATOR
DATA KSEED/45,678,986,687,587,897,345,7777,67,4,
1 568,889,444,888,999,555,222,70,80,8000,
2 5,15,25,35,45,55,65,75,85,95,
3 7,17,27,37,47,57,67,77,87,97,
4 9876,984,6666,2398,7589,499,899,561,
5 205,3995,9643,9376,6556,4533,9805,5672,2154,
6 881,8899,31,85,166,4,9,3,987,
7 888,999/

DATA DESCRIBING DIMENSIONS OF WIDE-ANGLE SCATTERERS
DEFINITION: DIM==2*D/LAMDBA (UNITLESS)
DATA DIM
/72,64.8/
DATA WSCALE
/SQRT(D**2/(12,NF)) (UNITS=FEET, NF= OF FREQ)
DATA DIM /72=64.8/
DATA WSCALE /72=0.2965/

FOR EACH DIFFUSE SCATTERER, SPECIFY NORMAL COMPONENT
DATA NORMAL /10,1.2,2.12,3/

SQUARE ROOT OF RCS VALUES ( FEET).
DATA SIGMA/24, 85,3,2.6,2,61,1200,1.25,0,17,25,7,110,000,
2 100,850,1200,1117,0,4,80,100,900,850,750,850,920,
3 736,6,0,03,1250,1130,1400,900,1000,1150,32.39/

COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
DATA TARG /4,12,6,7,8,35,37,3,24,2,37,
1 .66,3,35,3,12,3,3,35,4,37,6,24,6,7,8,0,
2 1,75,1,85,1,75,1,75,1,95,35,35,1,85,35,35,35,35,35.
C MINIMUM SUBTENDED ANGLE
   DATA PHIMIN /4*0.0,4*90.0,14*0.0,16*0.0,4*188.5,4*188.0,6*0.0,
   2 6*177.9,0.0,
   3 11*0.0,12*0.0,50.0,35.0,30.0,0.0,45.0,0.0,30.0,0.0,187.4,
   4 89.7,0.0,4*98.5,4*188.0,12*0.0,48.0,
   5 19*0.0,5*90.0,3*85.9,3*88.5,156.0,90.0,87.7,3*88.5,2*87.4,0.0,
   6 90.0,4*188.5,0.0,178.0,0.0,178.0,90.0,0.0,90.0,0.0,6*88.5,
   7 48.0/
C MAXIMUM SUBTENDED ANGLE
   DATA PHIMAX /4*90.0,20*180.0,5*90.0,2.1*180.0,3*2.1*180.0,
   2 4*91.5,4*92.6*90.0,6*180.0,48.0,
   3 18*180.0,90.0,13*180.0,4*150.0,155.0,135.0,2*180.0,145.0,3*180.0,
   4 2.6*180.0,90.0,3*180.0,4*91.5,4*92.6*180.0,6*180.0,138.0,
   5 12*180.0,7*90.0,5*180.0,3*94.1,3*91.5,180.0,156.0,92.3,3*91.5,2*92.6,
   6 125.5*180.0,2.1*180.0,2.1*180.0,90.0,180.0,90.0,90.0,6*91.5,138.0/
C RADII OF THE SCATTERERS (FEET)
   DATA OFFSET /24*0.0,3.1,2.0*29.0,0.2*35.0,315.5*0.0,24.35,8*0.0,
   2 6.16*0.0,0.0/
C MISCELLANEOUS DATA.
   DATA NTAR/61/,KWIDE/24/,PI/3.141592653/
   DATA TTRAN/3*0.0/,INIT1/1/
C ******************************************
C ** STEP 0: TRANSLATE POINT TARGETS BY TARGET FRAME OFFSET (TRTRAN) **
C ******************************************
C IF(INIT1.NE.1) GO TO 2
C RANDOMIZE DIFFUSE SCATTERER RCS VALUES.
C ISEED=100
   DO 107 I=1,1000
   X=RNDU(ISEED)
   DO 108 I=1,KWIDE
   X=RNDU(ISEED)
   CHANCE MADE 9-11-81
108   SIG_A(I)=SIGMA(I)+(X*0.005)-0.0025
C CONVERT TARGET DATA APPROPRIATELY.
C FTM=0.3048
   DO 101 I=1,NTAR
   SIGMA(I)=SORT(SIGMA(I))/FTM
   DO 102 I=1,NTAR
   TARG(J,I)=TARG(J,I)/FTM
   DO 103 I=1,3
   TARG(J,I)=TARG(J,I)/FTM
   DO 103 I=1,NTAR
   DO 103 I=1,3
   PHIMIN(J,I)=COS(PHIMIN(J,I)+PI/180.)
   DO 103 I=1,NTAR
   DO 103 I=1,3
   PHIMAX(J,I)=COS(PHIMAX(J,I)+PI/180.)
   DO 105 I=1,NTAR
   OFFSET(I)=OFFSET(I)/FTM
   DO 101 K=1,NTAR
   DO 101 I=1,3
   C
1 TARG(K,I)=TARG(K,I)+TTRAN(I)
INITI=0

• STEP 1: DETERMINE WHICH SCATTERER ARE ILLUMINATED AND HAVE A NONZERO RCS IN THE DIRECTION OF THE RADAR.

• STEP 1-1: PERFORM REQUIRED INITIALIZATIONS.
2 CONTINUE
NWIDE=0
KTAI=0

• STEP 1-2: COMPUTE UNIT VECTOR IN DIRECTION OF RADAR FOR ITH SCATTERING CENTER.
DO 15 I=1,NTAR
DO 5 J=1,3
VECT(J)=RADAR(J)-TARG(I,J)
5 CONTINUE
VNORM=SQRT(VECT(1)**2+VECT(2)**2+VECT(3)**2)
DO 10 J=1,3
IF(ABS(VECT(J)).GT.ABS(VNORM))WRITE(6,*)'VECT GREATER THAN VNORM'
COSPHI(I,J)=VECT(J)/VNORM
10 CONTINUE

• STEP 1-3: DETERMINE WHETHER ITH SCATTERER HAS A NONZERO RCS IN THE DIRECTION OF THE RADAR.
2 GO TO 15
10 CONTINUE

• STEP 1-4: IF ITH SCATTERER RCS IS NONZERO THEN ADD TO VECTOR OF ILLUMINATED SCATTERERS.
KTAI=KTAI+1
JHOT(KTAI)=I
SIG(KTAI)=SIGMA(I)
IF(I.LE.KWIDE) NWIDE=NWIDE+1
15 CONTINUE

• STEP 2: COMPUTE LOCATION OF SPECULAR POINTS THAT ARE ILLUMINATED.

• STEP 3: COMPUTE SQUARE ROOT OF RCS FOR ALL ILLUMINATED WIDE ANGLE SCATTERERS (REPRESENTING DIFFUSE SCATTERING AREAS).

• STEP 4: CHECK FOR SHORT RANGE CONDITION.

• STEP 4-1: DETERMINE RANGE TO RADAR IN TARGET FRAME.
24 RANGE=SQRT(RADAR(1)**2+RADAR(2)**2+RADAR(3)**2)
C STEP 4-2: SET HYSTERESIS LOOP MONITORING VARIABLE.
IF((ROLD.LT..01. OR. RANGE-ROLD. LE. 0.) AND. RANGE. LE. 270.) ICLOSE=1
IF(RANGE-ROLD. GT. 0. AND. RANGE. GT. 50.) ICLOSE=0

C STEP 4-3: CHECK MONITORING VARIABLE TO DETERMINE IF SHORT RANGE
CONDITION EXISTS.
IF(ICLOSE.EQ.0 OR. NWIDE.EQ.0) GO TO 55

C *****************************************
C • STEP 5: PROCEDURE FOR UPDATING OF DIFFUSE SCATTERING
C • CENTER LOCATION — SHORT RANGE CONDITION ONLY.
C *****************************************
C STEP 5-1: IF FIRST TIME THRU — PERFORM INITIALIZATION OF
DIFFERENCE EQUATIONS FOR ALL DIFFUSE SCATTERERS.
IF(ICLOLD.EQ.1) GO TO 35
DO 30 I=1,KWIDE
I=NORMAL(I)
PHIOLD(I)=ACOS(COSPHI(I,1Q ))
DO 25 J=1,3
IF(J.EQ.1Q) GO TO 25
V(I,J)=WSCALE(I,J)*((RNDU(KSEED(I,J))-.5)
VOLD(I,J)=V(I,J)
R(I,J)=R(I,J)+V(I,J)
25 CONTINUE
30 CONTINUE
GO TO 55

C STEP 5-2: UPDATE ANGULAR INCREMENT FOR EACH DIFFUSE SCATTERER
CHANGE IN ANGLE FROM SAMPLE-TO-SAMPLE.
35 DO 40 I=1,KWIDE
I=JHOT(K)
I=NORMAL(I)
PHI(I,1Q )=ACOS(COSPHI(I,1Q ))
DPI(I)=(PHI(I,1Q )-PHIOLD(I))
PHIOLD(I)=PHI(I,1Q )
40 CONTINUE

C STEP 5-3: UPDATE SCATTERER LOCATION FOR ALL ILLUMINATED DIFFUSE
SCATTERER — UPDATE DIFFERENCE EQUATIONS.
DO 50 K=1,NWIDE
I=JHOT(K)
DO 45 J=1,3
I=NORMAL(I)
IF(J.EQ.1Q) GO TO 45
ALPH(I,J)=EXP(-DIM(I,J)*ABS(DPHI(I,J)+COSPHI(I,1Q ))))
WRAN(I,J)=SORT(1.-ALPH(I,J)*2.)*WSCALE(I,J)*((RNDU(KSEED(I,J))-.5)
V(I,J)=ALPH(I,J)*VOLD(I,J)+WRAN(I,J)
VOLD(I,J)=V(I,J)
R(K,J)=R(K,J)+V(I,J)
45 CONTINUE
50 CONTINUE
55 CONTINUE

C *****************************************
C • STEP 6: UPDATE PARAMETERS USED TO MONITOR TARGET POSITION
C • ON SHORT RANGE HYSTERESIS CURVE.
C *****************************************
C ROLD= RANGE
ICLOLD=ICLOSE

C WRITE(6,908) KTAR,NWIDE,ICLOSE,ROLD
908 FORMAT(/" TT,WT,IC,R =",318,F12.4)

B-60
NOTE: THE FOLLOWING STATEMENTS ARE PRINT STATEMENTS USED IN THE DEBUGGING PROCESS.

NOTE: DEBUGGING PRINT STATMENTS.
PRINT LOCATION OF RADAR IN TARGET FRAME.
WRITE(6,900) RADAR

PRINT TABULAR LISTING OF ALL DATA ASSOCIATED WITH SPAS SCATTERERS.
WRITE(6,901)(1,SIGMA(I),TARG(I,1),TARG(I,2),TARG(I,3),OFFSET(I)
  8 ,PHIMIN(I,1),PHIMAX(I,1),PHIMIN(I,2),PHIMAX(I,2),PHIMIN(I,3),PHIMAX(I,3),
  2 I=1,NTAR)

PRINT TOTAL OF SCATTERERS AND OF DIFFUSE SCATTERERS.
WRITE(6,902) KTAR,NWIDE

PRINT INFORMATION ASSOCIATED WITH ILLUMINATED SCATTERERS.
WRITE(6,903)
WRITE(6,904) (I,JHOT(I),SIG(I),(R(I,J),J=I,3),
  1 I=1,NTAR)

PRINT DATA ASSOCIATED WITH DIFFUSE SCATTERER DIFFERENCE EQUATION.
WRITE(6,905)I,PHIOLD(I),
  1 (V(I,L),L=1,3),(R(I,L),L=1,3)
  IG=NORMAL(I)
  WRITE(6,906) I,PHI(I,1Q ),PHIOLD(I),DPHI(I)
  WRITE(6,907)K,I,(VOLD(I,J),J=I,3),(ALPH(I,J),J=I,3),
  1 (WRAN(I,J),J=I,3),(V(I,J),J=I,3),(R(I,J),J=I,3)

ALL PRINT FORMAT STATEMENTS.

FUNCTION SPAT(X)
NOTE: THE FOLLOWING VALUE OF B GIVES THE SUM PATTERN A SINGLE-SIDED 3 DB BEAMWIDTH OF 0.85 DEGREES.
Y=93.86*X
TEMP=ABS(Y)
IF(TEMP.GT.1.0E-06) GO TO 10
SPAT=1.0
RETURN
10 SPAT=SIN(Y)/Y
RETURN
END
FUNCTION SUM(X, N)
  Y = SIN(X)**2
  IF(Y .GT. 1.0E-08) GO TO 10
  SUM = N
  RETURN
10
  SUM = SIN(N*X)**2/(N*Y)
  RETURN
END

THIS SUBROUTINE RESETS THE SYSTEM UNDER THE FOLLOWING CONDITIONS:
1) BREAK-TRACK (TO SEARCH),
2) PASSIVE/ACTIVE MODE CHANGE (TO SEARCH),
3) SYSTEM IN STANDBY (TO IDLE).

---

SUBROUTINE SYSINT

COMMON /CNTL/IPWR, IMODE, ITP, IASM, IDUMC(5), DUMC(3)
COMMON /OUTPUT/MSWF, MTF, MSF, SRNG, SRTDOT, SPANG, SRANG, SPRT, SRTE, SSRS, MADVF, MRRDVF, MARDVF
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KMSCLK, KSNCLK, KSWDF
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK
COMMON /ICNTL/IOLDPW, IOLDMD, IOLDSM, ISHOLD, KMSCLK, KSMCLK, KSNCLK, KSWDF, MSAM, MPRF, MRRDVF, KSNMAX, KSNMAX, KSNCLK, KSNCLK, KSNCLK, KSNCLK

**STEP 1: INITIALIZE ALL INTERNAL FLAGS AND CONTROLS**

**STEP 2: INITIALIZE ALL INTERNAL CLOCKS**

**STEP 3: INITIALIZE ALL DISPLAY FLAGS**

**STEP 4: INITIALIZE ALL DISPLAY METERS**
C ***********************
SRNG=0.0
SRDOT=0.0
SRPTE=0.0
SRRT=0.0
SRSS=0.0
C
********** STEP 5: INITIALIZE GIMBAL POINTING LOOP **********
PII=3.14159265/180.
ALRATE=0.0
BTRATE=0.0
IF(IPWR.NE.1.AND.KMSCLK.NE.1) GO TO 5
C
C STEP 5-1: IF SYSTEM POWER OFF THEN ALIGN BORESIGHT WITH ZENITH.
PREF=0.0
RREF=0.0
AL=0.0
BT=0.0
SPANG=0.0
SRANG=0.0
IOLDPW=IPWR
RETURN
5 IF(IPWR.GT.2) GO TO 15
C
C STEP 5-2: IF SYSTEM IN STANDBY THEN HOLD GIMBALS AT POSITION WHEN STANDBY ENTERED AND ZERO DISPLAYS.
IF(IOLDPW.EQ.IPWR) GO TO 10
10 SPANG=0.0
SRANG=0.0
IOLDPW=IPWR
RETURN
C
C STEP 5-3: PREPARE GIMBAL LOOP FOR ENTRY INTO ANY OF SEARCH MODES.
15 SPANG=0.0
SRANG=0.0
IOLDPW=IPWR
RETURN
END
C
C *********************************************************
• THIS SUBROUTINE UPDATES THE DATA VALID FLAG STATUS •
C *********************************************************

SUBROUTINE TGTACG
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
COMMON /OUTPUT/MSWF,MSF,DUM(7),MADVF,MARDVF,MRRDVF,MRDVF
COMMON /ICNTL/IDUM(8),KACC,KTP,MZ,MZB,MSS,MTKINT,
2 MRNG,IDUM(12)
COMMON /SYSDAT/TS,DUM(14)
DIMENSION ADV(10,2),RDV(10,2),ARDV(10,2)
DATA ADV/9*1.02,5.12,8*1.02,2*2.33/
DATA RDV/9*6.15,28.69,8*6.97,2*29.76/
DATA ARDV/9*8.2,28.69,7*8.2,26.23,2*29.76/
C
C *********************************************************
• STEP 1: UPDATE ACQUISITION CLOCK •
C
KACC=KACC+1
C
B-63
ACCLK=KACCLK+TS

* STEP 2: PERFORM ANGLE DATA VALID TEST — GPC-ACQ + AUTO ONLY *

IF(IASM.EQ.2.0 OR IASM.EQ.4) GO TO 10
IF(ACCLK.LT.ADV(MRNG,IMODE)) GO TO 10
MADVF=1

* STEP 3: PERFORM RANGE AND RANGE RATE DATA VALID TEST *

10 IF(ACCLK.LT.RDV(MRNG,IMODE)) GO TO 15
MRDVF=1
MRRDVF=1

IF(IASM.EQ.2.0 OR IASM.EQ.4) GO TO 10
IF(ACCLK.LT.RDV(MRNG,IMODE)) GO TO 15
MRDVF=1
MRRDVF=1

* MODES ONLY. *

=================================================================

11 IF((IASM.EQ.2.0 OR IASM.EQ.4) .AND. blRDVF.EQ.1) GO TO 2e

* STEP 4: PERFORM ANGLE RATE DATA VALID TEST — GPC-ACQ + AUTO *

* MODES ONLY. *

IF(ACCLK.LT.ARDV(MRNG,IMODE)) RETURN
MARDVF=1

* STEP 5: PERFORM STEADY STATE RADAR TRACKING INITIALIZATION *

20 KACCLK=0
MTF=1
RETURN
END

* THIS SUBROUTINE GENERATES A (3X3) MATRIX TTH THAT PRODUCES *
* A ROTATION OF TH RADIANS ABOUT THE X-AXIS. *

SUBROUTINE THETA(TTH,TH)
DIMENSION TTH(3,3)
DO 10 I=1,3
DO 10 J=1,3
       10 TTH(I,J)=0.0
TTH(1,1)=1.0
TTH(2,2)=COS(TH)
TTH(3,3)=TTH(2,2)
TTH(2,3)=SIN(TH)
TTH(3,2)=-TTH(2,3)
RETURN
END

* THIS SUBROUTINE INITIALIZES THE ANGLE TRACKING LOOPS, THE *
* RANGE TRACKING LOOP, AND THE VELOCITY PROCESSOR — STEADY *
* STATE CONDITIONS ARE ASSUMED. *

SUBROUTINE TKINIT
REAL INTT, IRNG, IRDOT, IVR
COMMON /CNTL/ IPWR, IMODE, ITXP, IASM, IDUMC(5), DUMC(3)
COMMON /INPUT/ ERT(3), EVT(3), EWB(3), DUM(18)
COMMON /OUTPUT/ IDUM(6), SRNG, DUM1(3), DUM2(6), IDUM1(4)
COMMON /ICNTL/ I1DUM(3), MTKINT, MRNG, MSAM, MPRF, MBKTRK, MBTSUM,
2 MFT(8), MPFOLD
COMMON /SYSDAT/ TSAM, DR(3), CP, SP, PSI, PSBIAS, DUM2(7), TRB(3,3)
COMMON /TGTDAT/ NT, DUMS(500), RO(3), ROU(3), CCRNGE, CGVEL
COMMON /SATDAT/ RADAR(3), KTAR, RT(70,3), SIG(70), ROLD, ICLOSE, ICLOD
COMMON /ATDAT/ CA, SA, CB, SB, AZRATE, ELRATE, ALRATE, BTRATE,
2 DUM3(2)
COMMON /RTDAT/ IRNG, IRDOT, ITXP(4), MDF(5)
DIMENSION ER(3), EV(3), ERTO(3), FLTWID(3)
DATA FLTWID/7.7215, 3.3e9e, 4.2969/
DATA RI/126., 646., 1526., 23846., 43526.,
2 249926., 8228e+6/, NRI/10/, PI/3.141592653/

C **RI DATA STATEMENT UPDATED FEB 6, 1986 BY M. MEYER ************
DATA RI/120., 640., 1520., 2560., 5760., 11520., 23040., 43520.,
2 49926., 1.8228E+6/, NRI/10/, PI/3.141592653/

C *STEP 0: INITIALIZE BREAK-TRACK ALGORITHM*
C *****************************************
C STEP 0-1: INITIALIZE MOVING WINDOW-OF-8 REGISTERS.
DO 3 I=1, 8
3 MBT(I)=0
C STEP 0-2: INITIALIZE SUM REGISTER.
MBTSUM=0
C STEP 0-3: SET BREAK-TRACK FLAG TO LOW (OR 0) STATE.
MBKTRK=0
C *****************************************
C *STEP 1: INITIALIZE ANGLE TRACKING LOOP*
C *****************************************
IF(IASM.EQ.2.0R.IASM.EQ._) GO TO 5
C STEP 1-1: COMPUTE INITIAL INNER AND OUTER GIMBAL POSITIONS.

C (NOTE: TRANSFORM CONSISTS OF TRANSLATION PLUS ROTATION.)
C PERFORM TRANSLATION —- SHIFT TO RADAR FRAME ORIGIN.
DO 1 I=1, 3
1 ERTO(I)=ERTO(I)-DR(I)
C TRANSFORM TARGET POSITION FROM BODY TO RADAR FRAME.
CALL MULT31(TRB, ERTO, ER)
C TRANSFORM TARGET VELOCITY FROM BODY TO RADAR FRAME.
CALL MULT31(TRB, EVT, EV)
C COMPUTE INNER(BETA) GIMBAL POSITION —- BT.
IF(ER(I).EQ.0.0 AND SQ.EQ.0.0) STOP
BT=ATAN2(ER(I), SQ)
ER2=ER(I)
ER3=ER(I)
C COMPUTE OUTER(ALPHA) GIMBAL POSITION —- AL.
IF(ER2.EQ.0.0 AND ER3.EQ.0.0) GO TO 8
AL=ATAN2(ER2, ER3)
GO TO 9
8 IF(ER(I).GT.0.0) AL=PI/2.
IF(ER(I).LT.0.0) AL=-PI/2.
IF(ER(I).EQ.0.0) STOP
CSTEP 1-2: COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH AND ELEVATION RATES.

B-65
PRELIMINARY TRIGONOMETRIC COMPUTATIONS.

9 CA=COS(AL)
SA=SIN(AL)
CB=COS(BT)
SB=SIN(BT)

TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER GIMBAL(G) REFERENCE FRAME.

WGX=CP*EWB(1)+CP*EWB(2)
WGZ=SA*(-SP*EWB(1)+CP*EWB(2))+CA*EWB(3)

COMPUTE THE RANGE TO TARGET.
R=SORT(ER(1)+ER(2)+ER(3)+ER(3))

COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).

AZRATE=VGY/R+(CB*WGX-SB*WGZ)

COMPUTE INITIAL TARGET INERTIAL LOS ELEVATION RATE(ELRATE).

ELRATE=(CB*EV(1)-SA*EV(2)+CA*EV(3))/R+VGY

COMPUTE THE RANGE TO TARGET.
R=SORT(ER(1)+ER(2)+ER(3))

COMPUTE INITIAL OUTER GIMBAL RATE(ALRATE).

RCB=R+CB

IF(ABS(RCB).LT.1.6E-6) GO TO 2

ALRATE=VGY/RCB GO TO 4

CONTINUE

COMPUTE INITIAL INNER GIMBAL RATE(BTRATE).

BTRATE=ELRATE-VGY

STEP 1-3: COMPUTE INITIAL INNER AND OUTER GIMBAL RATES.

STEP 2-1: TRANSFORM TARGET C.G. POSITION AND C.G. VELOCITY FROM BODY TO ANTENNA LOS FRAME.

CALL TRNSFM
CALL PVTRAN

STEP 2-2: INITIALIZE THE RANGE ESTIMATE REGISTER.
SRNG=CGRNG
IRNG=INTT(SRNG+3.2+0.5)

STEP 2-3: INITIALIZE THE RANGE RATE ESTIMATE REGISTER.
SRNG=INTT(CGVEL+TSAM*3.2+0.5)

STEP 3-1: DETERMINE CORRECT RANGE INTERVAL.

DO 36 I=1,NRI
MRNG=I
IF(RI(I) .GT. SRNG) GO TO 40
CONTINUE

STEP 3-2: DETERMINE CORRECT SAMPLE RATE.

IF(MODE.GE.2) GO TO 44
IF(MRNG.GT.9) GO TO 42
MSAM=1
GO TO 50

END

0015840
0015850
0015860
0015870
0015880
0015890
0015900
0015910
0015920
0015930
0015940
0015950
0015960
0015970
0015980
0015990
0016000
0016020
0016030
0016040
0016050
0016060
0016070
0016080
0016090
0016100
0016110
0016120
0016130
0016140
0016150
0016160
0016170
0016180
0016190
0016200
0016210
0016220
0016230
0016240
0016250
0016260
0016270
0016280
0016290
0016300
0016310
0016320
0016330
0016340
0016350
0016360
0016370
0016380
0016390
0016400
0016410
0016420
0016430
0016440
0016450
0016460
0016470
MSAM=1
GO TO 50
46 MSAM=2
C
C STEP 3-3: DETERMINE CORRECT PRF.
50 IF(IMODE.GE.2) GO TO 54
IF(MRNG.GT.9) GO TO 52
MPRF=1
GO TO 60
52 MPRF=3
GO TO 60
54 IF(MRNG.GT.9) GO TO 56
MPRF=1
GO TO 60
56 MPRF=2
60 CONTINUE
C
C STEP 3-4: SET PRF TRANSITION flag.
MPFOLD=MPRF
C
C*****************************************************
C* STEP 4: INITIALIZE VELOCITY PROCESSOR *
C*****************************************************
C
C STEP 4-1: INITIALIZE MOVING WINDOW VELOCITY AVERAGING.
DO 10 I=1,4
VEST(I)=CGVEL*20.
10
C
STEP 4-2: SET INITIAL POSITION OF 5 DOPPLER FILTERS.
VR=CGVEL/FLTWF(MPRF)
IVR=INT(VR+0.5)+16000.
XX=AMOD(IVR,32.)
MDF(3)=INT(XX)
DO 20 I=1,5
MD=MDF(3)+I-3+160
MDF(I)=MOD(MD,32)
20
C
C*****************************************************
C* STEP 5: INITIALIZE AGC LOOP *
C*****************************************************
AGCO=1.0
ITXP=1
C
C*****************************************************
C* STEP 6: SET TRACK INDICATOR TO ALLOW OPERATION OF TRACK LOOP *
C*****************************************************
MTKINT=1
C
C ROLD=0
ICLOSE=0
ICLOLD=0
C
NOTE: DEBUGGING PRINT STATEMENTS.
WRITE(6,889)
WRITE(6,900) AZRATE,ELRATE,ALRATE,BTRATE,AL,BT
WRITE(6,901)
WRITE(6,902) IRNG,IRDOT,SRNG
WRITE(6,903)
WRITE(6,904) (VEST(I),I=1,4),(MDF(J),J=1,5)
WRITE(6,905)
WRITE(6,906) IMODE,MNMG,MSAM,MPRF
889 FORMAT(//' TRACKER INITIALIZATION:/' ATACK: AZRATE',
2 ' ELRATE,ALRATE,BTRATE,AL,BT'
900 FORMAT(6F14.6)
**SUBROUTINE TRACK**

**COMMON** /CNTL/IDUM(3),IASM,ISRCHC,ISRCHG,IAZS,IELS,ISLR,EDRNG,
2 EDPA,EDRA
COMMON /OUTPUT/MSWF,MTF,MSF,DUMO(7),IDUMO(4)
COMMON /ICNTL/IIIDUM(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,IDUM2(9)
COMMON /SYSDAT/TSAM,DUM2(14)
COMMON /ATDAT/DUMI(IO),PREF,RREF,DUMA(2)
DIMENSION SLWRTE(2)
DATA SLWRTE/6.9814E-3,3.4907E-1/

**STEP 1:** INITIALIZE TRACK MODE —— INITIALIZE ALL TRACK LOOPS *
**STEP 1-1:** INITIALIZE RANGE, ANGLE, AND VELOCITY TRACK LOOPS —— ASSUMES STEADY STATE TRACKING OF TARGET C.G.
**STEP 1-2: UPDATE DATA VALID FLAG STATUS —— ONLY WHEN ENTERING TRACK FROM SEARCH.

**STEP 2:** PERFORM TRACKING LOOP UPDATE PROCEDURE *
**STEP 2-1:** UPDATE TRANSFORMATION MATRICES AND MATRICE RATES.
**STEP 2-2:** TRANSFORM TARGET POSITION AND VELOCITY COMPONENTS FROM ORBITER BODY FRAME-TO-ANTENNA LOS FRAME.
**STEP 2-3:** GENERATE NOISE-FREE TARGET RETURN SIGNAL AND PROCESS SIGNAL TO PRODUCE NOISE-FREE DISCRIMINANT COMPONENTS.
**STEP 2-4:** ADD EQUIVALENT NOISE TO DISCRIMINANT COMPONENTS AND FORM ALL REQUIRED DISCRIMINANTS.
**STEP 2-5: UPDATE STATUS OF BREAK-TRACK FLAG.**
CALL BRKTRK

STEP 2-6: CHECK STATUS OF BREAK-TRACK FLAG — IF BREAK-TRACK FLAG UP (MBKTRK=1) RESET SYSTEM AND RETURN TO SEARCH.

IF(MBKTRK.NE.1) GO TO 7

CALL SYSINT

RETURN

STEP 2-7: DETERMINE RADAR SIGNAL STRENGTH (FOR DISPLAY METER) AND UPDATE AGC VALUE.

7 CALL RSS

STEP 2-8: UPDATE ANTENNA GIMBAL POSITIONS AND RATES AND TARGET ANGLES AND ANGLE RATES FOR DISPLAY (GPC-ACQ AND AUTO MODES ONLY.)

IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 10

STEP 2-8A: IF IN GPC-ACQ OR AUTO MODE USE RADAR ESTIMATED TARGET ANGLES AS GIMBAL TRACK SERVO INPUT.

CALL ATRACK

GO TO 15

10 IF(IASM.EQ.4) GO TO 12

STEP 2-8B: IF IN GPC-DES MODE USE GPC-SUPPLIED ANGLE DESIGNATES AS GIMBAL TRACK SERVO INPUT.

PREF=EDPA

RREF=EDRA

CALL POINT

GO TO 15

STEP 2-BC: IF IN MANUAL MODE USE CREW-SUPPLIED SLEW RATES TO DETERMINE GIMBAL TRACK SERVO INPUT.

12 PREF=PREF+FLOAT(IELS)*SLWRTE(ISLR+1)*TSAM

RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TSAM

CALL POINT

STEP 2-9: UPDATE THE RANGE AND RANGE RATE ESTIMATES.

15 CALL RTRACK

STEP 2-10: UPDATE ACCURATE VELOCITY ESTIMATE USING VELOCITY PROCESSOR.

CALL VELPRO

STEP 2-11: UPDATE ALL RADAR INTERNAL CONTROLS.

CALL CNTRLS

20 RETURN

END

******************************************************************************
* THIS SUBROUTINE UPDATES ALL REQUIRED TRANSFORMATION MATRICES *
* MATRICES AND TRANSFORMATION MATRIX RATES. *
******************************************************************************

SUBROUTINE TRNSFM

COMMON /INPUT/DUM(9),TBT(3,3),TBTD(3,3)

COMMON /SYSDAT/DUM2(4),CP,SP,DUM4(9),TRB(3,3)

COMMON /ATDAT/CA,SA,CB,SB,DUM1(2),ALRATE,BTRATE,AL,BT,DUM3(4)

COMMON /XFORMS/TBL(3,3),TLBD(3,3),TLT(3,3),TLTD(3,3)

DIMENSION TLR(3,3)

******************************************************************************
* STEP 1: UPDATE TRANSFORMATION MATRICES *
******************************************************************************
**STEP 1-1: PRELIMINARY COMPUTATIONS.**

\[
\begin{align*}
C B &= \cos(BT) \\
S B &= \sin(BT) \\
C A &= \cos(AL) \\
S A &= \sin(AL)
\end{align*}
\]

**STEP 1-2: COMPUTE TRANSFORMATION MATRIX \( TLB \) (BODY-TO-LOS FRAME).**

\[
\begin{align*}
TLR(1,1) &= C B \\
TLR(1,2) &= S B \cdot S A \\
TLR(1,3) &= -S B \cdot C A \\
TLR(2,1) &= 0.0 \\
TLR(2,2) &= C A \\
TLR(2,3) &= S A \\
TLR(3,1) &= S B \\
TLR(3,2) &= -C B \cdot S A \\
TLR(3,3) &= C B \cdot C A
\end{align*}
\]

CALL MULT33(TLR, TRB, TLB)

**STEP 1-3: COMPUTE TRANSFORMATION MATRIX \( TLT \) (TARGET-TO-LOS FRAME).**

CALL MULT33(TLB, TBT, TLT)

**STEP 2-1: COMPUTE TLB-DOT.**

\[
\begin{align*}
TLBD(1,1) &= BTRATE \cdot TLB(3,1) + ALRATE \cdot S B \cdot TLB(2,1) \\
TLBD(1,2) &= BTRATE \cdot TLB(3,2) + ALRATE \cdot S B \cdot TLB(2,2) \\
TLBD(1,3) &= BTRATE \cdot TLB(3,3) + ALRATE \cdot S B \cdot TLB(2,3) \\
TLBD(2,1) &= ALRATE \cdot S P \cdot TLB(2,3) \\
TLBD(2,2) &= ALRATE \cdot C P \cdot TLB(2,3) \\
TLBD(2,3) &= ALRATE \cdot C A \\
TLBD(3,1) &= BTRATE \cdot TLB(1,1) - ALRATE \cdot C B \cdot TLB(2,1) \\
TLBD(3,2) &= BTRATE \cdot TLB(1,2) - ALRATE \cdot C B \cdot TLB(2,2) \\
TLBD(3,3) &= BTRATE \cdot TLB(1,3) - ALRATE \cdot C B \cdot TLB(2,3)
\end{align*}
\]

**STEP 2-2: COMPUTE TLT-DOT.**

DO 20 I=1,3
DO 20 J=1,3
TLD(I,J) = 0.0
DO 20 K=1,3
20 TLD(I,J) = TLD(I,J) + TLBD(I,K) \cdot TBT(K,J) + TLB(I,K) \cdot TBTD(K,J)
RETURN
END

**THIS SUBROUTINE COMPUTES AN ACCURATE, SMOOTHED VELOCITY USING**

**THE KU-BAND RADAR VELOCITY PROCESSOR ALGORITHM.**

SUBROUTINE VELPRO
REAL IRDOT, IRNG, INTT, IVEL, IVDISC, IFVEL, IRVEL, IR1, IR2, IR3,
IF3, IDELTA,
COMMON /CNTL/TPWR, IMODE, IDUMC(7), DUMC(3)
COMMON /OUTPUT/IDUM(3), SRNG, SRDOT, DUM2(5), IDUM(4)
COMMON /CNTL/IDUM(14), MRNG, MSAM, MPRF, IDUM1(10), MPFOLD
COMMON /SYSDAT/TSAM, DUMS(14)
COMMON /RTDAT/IRDOT, IRNG, RBIAS, VEST(4), MDF(5)
COMMON /DSCRA/DUM(2), RDISC, VDSC, RRT, ODISC, DUM3(3)
DIMENSION IPROM(128), VT1(3), VT2(3), MW(4,3)

B-70
**SUBROUTINE VELPRO WAS MODIFIED FEB 6 1986 BY M. MEYER**

**MODIFICATIONS CONSISTED OF CHECKING THE VARIABLE MPRF FOR A VALUE OF ONE (IMPLIES 7 KC MODE) AND IF TRUE ASSUMING THE VELOCITY ESTIMATE GIVEN BY THE VELOCITY DISCRIMINANT IS UNAMBIGUOUS.**

**STEP 1: GENERATE AMBIGUOUS VELOCITY ESTIMATE**

**STEP 1-1: INTEGRIZE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION.**

```fortran
  VDISC=5.333333+VDISC
  IVDISC=INT(VDISC+0.5)
  IF(IVDISC.LT.-128.) IVDISC=-128.
  IF(IVDISC.GT.127.) IVDISC=127.
```

**STEP 1-2: COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS VELOCITY ESTIMATE.**

```fortran
  INTEG=MOD(2)
  IF(IVDISC.LT.e.) INTEG=KX(INTEG+1,32)
```

**STEP 1-3: COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY ESTIMATE.**

```fortran
  IV1=INT(ABS(IVDISC))
  IF1=IPROM(IV1)
  IFRAC=IPROM(IV1)
  IF(IVDISC.LT.e.) IFRAC=127-IFRAC
```

**STEP 1-4: COMPUTE AMBIGUOUS VELOCITY ESTIMATE — COMBINE INTEGRAL AND FRACTIONAL PARTS. NOTE: LSB IS 1/128 OF FILTER WIDTH.**

```fortran
  IF(MPRF.EQ.1) THEN
    IF(INTEG.GE.0.AND.INTEG.LE.21) THEN
      IRVEL=0.
    ELSE
      IRVEL=4096.
    END IF
    GO TO 8
  END IF
```

**STEP 2: SCALE ROUGH VELOCITY ESTIMATE**

```fortran
  DATA MM/1.2.3.4.1.1.2.1.1.1/
```

**DATA IPROM/*127.127.125.124.122.121.120.118.117.116.114.113.**

**DATA VT1/1.e12592E-2,2.362726E-2,2.633237E-1/,VT2/1.294935 e.5163982,e.94633489/**

**DATA */1,2,3,4,1,1,2,2,1,1,1,1/**
C STEP 2-1: SCALE LSB OF ROUGH RANGE RATE ESTIMATE TO 4 TIMES A DOPPLER.
C WIDTH.
C DEFINITION: \( V_{T1}(MPRF) = \frac{(RANGE \text{ LSB})}{(MAX. UNAMBIGUOUS VELOCITY) / 8} \)
C OR \( V_{T1}(MPRF) = 5 \cdot (PRF \cdot \text{LAMBDA}) \)
C \( R1 = \text{INT}(V_{T1}(MPRF) / \text{TSAM}) \)
C \( IR1 = \text{AINT}(R1) \)
C
C STEP 2-2: PERFORM SOME REQUIRED AUXILIARY CALCULATIONS.
C \( R2 = IR1 / 8 \).
C \( IR2 = \text{AINT}(R2) \)
C \( IRVEL = IR2 + 4096. \)
C
C *************
C STEP 3: RESOLVE AMBIGUITY
C *************
C
C STEP 3-1: COMPUTE 3 MSB'S OF AMBIGUOUS VELOCITY ESTIMATE.
C \( IF3 = \text{AINT}(IFVEL / 512.) \)
C
C STEP 3-2: COMPUTE 3 LSB'S OF SCALED ROUGH RANGE RATE ESTIMATE.
C \( IR3 = \text{ABS}(IR1 - IR2) \)
C \( IF(R1 \leq \text{E.}) \text{GO TO 10} \)
C \( IRVEL = IRVEL + 4096. \)
C \( IR3 = 7 - IR3 \)
C 10 CONTINUE
C
C STEP 3-3: COMPARE 3 MSB'S AND 3 LSB'S AND INCREMENT NUMBER OF
C AMBIGUOUS FILTER BANK WIDTHS APPROPRIATELY.
C \( IDELTA = IR3 - IF3 \)
C \( IF(IDELTA \geq 4.) \text{IRVEL = IRVEL - 4096.} \)
C \( IF(IDELTA \leq -4.) \text{IRVEL = IRVEL + 4096.} \)
C 8 CONTINUE
C
C *************
C STEP 4: COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE
C *************
C
C STEP 4-1: ADD NUMBER OF AMBIGUOUS FILTER BANK WIDTHS TO ESTIMATE
C OF FRACTIONAL FILTER BANK WIDTH. NOTE: LSB OF RESULTANT
C ESTIMATE REPRESENTS 1/4096 OF A FILTER BANK WIDTH.
C \( IVEL = \text{INTT}(IRVEL - IFVEL) \)
C
C STEP 4-2: SCALE LSB OF RESULTANT ESTIMATE TO 0.05 FEET/SEC.
C DEFINITION: \( V_{T2}(MPRF) = \frac{(FILTER SEPARATION) / 128.}{(VELOCITY LSB)} \)
C OR \( V_{T2}(MPRF) = \frac{(PRF \cdot \text{LAMBDA})}{(0.05 \cdot 8196)} \)
C \( IVEL = \text{INTT}(IVEL \cdot V_{T2}(MPRF) + 0.5) \)
C
C *************
C STEP 5: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY
C *************
C
C STEP 5-1: UPDATE REGISTERS OF MOVING WINDOW AVERAGER.
C DO 20 I=1,3
C 20 VEST(5-I) = VEST(4-I)
C VEST(1) = IVEL
C
C STEP 5-2: COMPUTE MOVING WINDOW AVERAGE AND SCALE ANSWER INTO
C FEET/SEC FROM UNITS OF 0.05 FEET/SEC.
C \( M = \text{MPRF} \)
C \( M1 = M \cdot (1, M) \)
C \( M2 = M \cdot (2, M) \)
C \( M3 = M \cdot (3, M) \)
M4=MM(4,M)
SRDOT=0.8125*(VEST(M1)+VEST(M2)+VEST(M3)+VEST(M4))

STEP 6-1: USE ON-TARGET DISCRIMINANT AND VELOCITY DISCRIMINANT TO DETERMINE UPDATE OF FILTER BANK POSITION.
THE FOLLOWING RULES ARE USED:
CASE 1: ODISC>O. AND -51.<IVDISC<51. IMPLIES NO CHANGE.
CASE 2: ODISC>O. AND IVDISC>51. IMPLIES SHIFT -1.
CASE 3: ODISC<0. AND IVDISC<-51. IMPLIES SHIFT +1.
CASE 4: ODISC<0. AND IVDISC>O. IMPLIES SHIFT -2.
CASE 5: ODISC<0. AND IVDISC<0. IMPLIES SHIFT +2.

IF(ODISC.GE.e.) GO TO 30
IF(IVDISC.LT.O.) MDF(1)-=MOD(MDF(1)+2,32)
IF(IVDISC.GE.e.) MDF(1)=-MOD(MDF(1)+32,32)
GO TO 40
30 IF(IVDISC.GT.51.) MDF(1)=MOD(MDF(1)+31,32)
IF(IVDISC.LT.-51.) MDF(1)=MOD(MDF(1)+31,32)
GO TO 40

STEP 6-2: RESET REMAINING FILTERS IN THE BANK-OF-5.
40 DO 50 I=1,4
50 MDF(I+1)=MOD(MDF(I)+1,32)
RETURN
END

THIS SUBROUTINE DETERMINES WHETHER ANTENNA IS IN ZONE 1 AND/OR ZONE 0 (FOR GPC-ACQ AND GPC-DES POINTING MODES ONLY).

SUBROUTINE ZONECK
COMMON /CNTL/I,EDRM,EDPA,EDRA
COMMON /IOUT/IDUM1(3),DUM1(2),SPANG,SRANG,DUM3(3),IDUM3(4)
COMMON /ICNTL/IDUM2(10),MZ1,MZ0,IDUM4(15)
MZ0=0
MZ1=1
PI=3.141592653/180.
RB=PI*SRANG
PB=PI*SPANG
P=EDPA
R=EDRA
CPB=COS(PB)
SPB=SIN(PB)
CRB=COS(RB)
SRB=SIN(RB)
CP=COS(P)
SP=SIN(P)
CR=COS(R)
SR=SIN(R)
ANGDIFF=ACOS(SPB*CRB+SP*CR+SRB*SR+CPB*CR+CP*CR)/PI
ANGDIFF=ABS(ANGDIFF)
IF(ANGDIFF.GT.3.0) RETURN
MZ0=1

B-73
IF(ANGDIF.GT.0.3) RETURN
MZ1=1
RETURN
END

SMM MODEL AS OF JANUARY 13, 1982

II. DIMENSION ARRAYS & DATA STATEMENTS

A) DIMENSION STATEMENTS

REAL KSEED
COMMON /SATDAT/RADAR(3),KTAR,R(70,3),SIG(70),ROLD,ICLOSE,ICLOLD
DIMENSION SIGMA(49),TARG(49,3),PHIMIN(49,3),PHIMAX(49,3)
DIMENSION OFFSET(49),JHOT(49),JHOT20(49),PHI(49),FG(3)
DIMENSION VECT(3),COSPHI(49,3),COSPHN(49),ORIENT(49,3)
DIMENSION ALPH(19,3),DPM(19,3),DIM(19,3),WRAN(19,3),SDMAX(19,3)
DIMENSION WS(19,3),DPHI(19),PHOLD(19),VOLD(19,3),KSEED(19,3)
DIMENSION TTRAN(3),ABG(19,3),TMAX(49),PL(49),SDMIN(19,3)

B) DATA STATEMENTS

1. KSEED - SEEDS FOR RANDOM NUMBER GENERATOR "ZUDU".
   DATA KSEED/45,878,937,467,8797,345,7777,67,4156,78,44498,
   888999555,22278,58800,
2 5.15,25.35,45.55,65.75,85.95,
3 7.17,27.37,47.57,67.77,87.97,
4 9876,984,6662,3968,76,412,7589,409,889,561,
5 265.3895,9457,9643,937,656,453/

2. DIM - THE GENERAL SIZE OF EACH DIFFUSE SCATTERER.
   DATA DIM/57*64.8/

3. WS - WEIGHTING ASSIGNED TO EACH SIDE OF A DIFFUSE SCATTERER.
   DATA WS/8.1,8.84,5.9386,2.5.6804,5.9386,5.6804,4.11.1026,
   1 2.6.7958,
2 2.9.8682*2.7.1111,2*3.6148,2*2.5174,4.3894,2*5.8905,4.3894,
3 5.8905,4*17.8803,2*6.7958,19*0.6/

4. ORIENT - THE i,j,k COMPONENTS OF THE NORMAL VECTOR OF EACH TARGET.
   a) i COMPONENT
   DATA ORIENT/13*0.,-9976,-.9976,-.9976,1..-1..,
   1 23*0.,9976,-.9976,-.9976,.9976,1..-2.-1.,
   b) j COMPONENT
   2 1.,-1.,2*6.428,2*6.428,1.,-1.,2*6.428,6.428,1.,-1.,2*6.428,
   c) k COMPONENT
   6 2*0.,-7.669,7.669,1..-1.,-7.669,-.7.669,-.7604,7.716,0.,-8.704,
   7 3.824,-.8284,.8284,.8284,4*0.,-.766,.766,.766,37.466,8572,9563,1..-1.,
   8 -7.604,7.716,2*0.,2*0.,-8.704,-.4924,.8704,-.8704,-.5,.5,0.,1..766,
   9 .8242,-.8284,.8284,3*0.6/

5. ABG - ARRAY OF TRANSFORMATION ANGLES(RAD), ALPHA, BETA, GAMMA, FOR DIFFUSE SCATTERERS.
   a) ALPHA
   DATA ABG/4*3.141593,2*1.570796,2*0.,4*3.141593.0.,1.634563,
   1 -1.570796,1.570796,4.648623,1.570796,4.712389,
   b) BETA
   2 2*1.570796,2.443392,.6982,0.,3*1.41593,2.434725,.689444,
   3 1.570796,2.443392,.6982,1.634563,.689444,1.570796,1.542392,
   4 2*1.570796,1.542392,2*1.570796,
6. SIGMA— THE CALCULATED RCS FOR EACH TARGET IN M**2.
DATA SIGMA/2*. 1, 2*. 0154, 2*. 8274, 2*. 8133, 2*. 8121, 2*. 8194, 8121,
2*. 8194, 4*. 7026, 2*. 8066, 2*. 2419, 373, 7251, 21.84, 11.14, 18.83,
3*. 663, .2*. 321, 2*. 3. 63, .92, .97. 470, .82, 13. 470, .2*. 63, .470, .83,
4*. 634, .4*. 16995, .2*. 146615, .3322/

7. TARG— TARGET POSITION (IN X, Y, Z COORDINATES) RELATIVE TO
THE COORDINATE AXIS OF SMM.
   a) X COORDINATE
DATA TARG /9*. 1.394, 4-. 7741, .270, .231, .278, .231, 2. 491, -1. 497,
2*. 31, .231, .270, .231, 2. 491, -1. 497, 3. 862, -. 862, 2*. 555, 2*. 0, .2*. 555, 748, .439, 1. 897, -. 361, -1. 497, -1. 855, 2. 233, 2-. 233, 2*. 0, .2*. 826, -. 826, 2*. 555, .658, .568, .439, 2. 0, .6
5*. 555, 2*. 0, .2*. 748, .439, .865, 1. 897, .865, -. 207, -. 955, -. 884,
-. 3614, 2. 233, 2-. 233, 3*. 0, /
   c) Z COORDINATE
8. PHIMIN— MINIMUM ANGLE OF DEVIATION FROM SMM COORDINATES
   a) MINIMUM ANGLE SUBTENDED IN X-DIRECTION
   b) MINIMUM ANGLE SUBTENDED IN Y-DIRECTION
   c) MINIMUM ANGLE SUBTENDED IN Z-DIRECTION

9. PHIMAX— MAXIMUM ANGLE OF DEVIATION FROM SMM COORDINATES
   a) MAXIMUM ANGLE SUBTENDED IN X-DIRECTION
   b) MAXIMUM ANGLE SUBTENDED IN Y-DIRECTION
   c) MAXIMUM ANGLE SUBTENDED IN Z-DIRECTION

10. OFFSET— POSITION OF TARGET SPECULAR PT. RELATIVE TO TARGET
COORDINATES.
   DATA OFFSET /17*. 2*. 0, .2*. 0, .11*. 0, .74868, .8, 14, .2*. 0, .6518/

11. MISCELLANOUS
DATA PL/ 30*1.,2*0.,16*1.,0./
DATA TMAX/19*90.,11*5.2*0.,16*1.5,0./
DATA NTAR/49.,KWIDE/19.,/PI/3.141592653/
DATA TTRAN/3*0.0.,/INIT1/1/
IF(INIT1.NE.1) GO TO 2

12. SDMIN- MINIMUM ANGLE OF VIEW; TARGET SHADOWING.
   a) X-COORDINATE
   DATA SDMIN/2*0.6828,-1.,-0.7467,2*1.,-0.7467,12*1.,
   1 19*1.,
   c) Z-COORDINATE
   2 19*1.,

13. SDMAX- MAXIMUM ANGLE OF VIEW; TARGET SHADOWING.
   a) X-COORDINATE
   DATA SDMAX/8*1.,.4218,3*1.,.4218,.5637,.6046,.5637,.6046,
   1 2*1.,
   b) Y-COORDINATE
   2 19*1.,
   c) Z-COORDINATE
   3 19*1.,

III. RANDOMIZE DIFFUSE SCATTERER RCS VALUES.
ISEED1=100
ISEED2=83
DO 107 I=1,1000
X=RNDU(ISEED1,ISEED2)
DO 108 I=1,KWIDE
X=RNDU(ISEED1,ISEED2)
108 SIGMA(I)=SIGMA(I)*2.*X

IV. CONVERT TARGET DATA APPROPRIATELY.
FTM=0.3848
DO 101 J=1,NTAR
SIGMA(J)=SORT(SIGMA(J))/FTM
DO 102 J=1,NTAR
TARG(J,1)=TARG(J,1)/FTM
DO 103 J=1,3
TMAX(J)=COS(TMAX(J)*PI/180.)
DO 104 J=1,3
PHIMIN(J,1)=COS(PHIMIN(J,1)*PI/180.)
103 PHIMAX(J,1)=COS(PHIMAX(J,1)*PI/180.)
DO 105 J=1,NTAR
OFFSET(J)=OFFSET(J)/FTM
105 CONTINUE

V. INITIALIZATION OF TARGET POSITION & COUNTING PARAMETERS
NWIDE & KTA.R.
DO 1 K=1,NTAR
DO 1 I=1,3
TARG(K,I)=TARG(K,I)+TTRAN(I)
1 INITI=0
2 CONTINUE
NWIDE=0
KTA.R=0

VI. DETERMINE WHICH TARGETS ARE ILLUMINATED.
WRITE(2,500)

B-76
A) DETERMINE THE POSITION OF THE RADAR RELATIVE TO TARGET SPECULAR POINT.

1. "VECT"- POSITION VECTOR
   DO 5 J=1,3
   VECT(J)=RADAR(J)-TARG(I,J)
   5 CONTINUE

2. VNORM- MAGNITUDE OF "VECT".
   VNORM=SQRT(VECT(1)**2+VECT(2)**2+VECT(3)**2)

B) DETERMINE THE COSINE OF THE ANGLE BETWEEN THE RADAR POSITION RELATIVE TO THE TARGET SPECULAR PT. & TARGET NORMAL.

1. CALCUATE THE ANGLE BY EMPLOYING THE DOT PRODUCT OF THE TWO VECTORS: "COSPHI" & "ORIENT".
   DP=0.
   DO 7 J=1,3
   2. COSPHI- UNIT VECTOR OF "VECT": REPRESENTATIVE OF THE COSINE OF THE ANGLE BETWEEN "VECT" & SMMS COORDINATE AXIS.
      COSPHI(I,J)=VECT(J)/VNORM
      7 DP=DP+COSPHI(I,J)*ORIENT(I,J)
   3. COSPHN- COSINE OF THE ANGLE; RESULT OF THE DOT PRODUCT.
      COSPHN(I)=DP

C) TEST OF ILLUMINATION- TWO METHODS: COMPARE COSPHN W/TMAX OR COMPARE COMPONENTS OF COSPHI W/PHIMIN & PHIMAX.

1. PL- A FLAG: 0 INDICATES METHOD 1 & 1 INDICATES METHOD 2.
   IF(PL(I).EQ.0.)GO TO 9
2. METHOD 1
   IF(COSPHN(I).LT.TMAX(I))GO TO 15
   GO TO 11
3. METHOD 2
   9 DO 10 J=1,3
   2 GO TO 15
   10 CONTINUE

D) TARGET SHADOWING

1. TEST FIRST 19 TARGETS ONLY.
   IF(I.GT.19)GO TO 13
2. FIND SHADOWING VECTOR BY TRANSFORMATION OF COSPHI FROM SMMS TO TARGET COORDINATES.
   F1=COSPHI(I,1)*COS(ABG(I,1))+COSPHI(I,2)*SIN(ABG(I,1))
   F2=COSPHI(I,2)*COS(ABG(I,1))-COSPHI(I,1)*SIN(ABG(I,1))
   F3=COSPHI(I,3)
   FB2=F2+COS(ABG(I,2))+F3+SIN(ABG(I,2))
   FB3=F3+COS(ABG(I,2))-F2+SIN(ABG(I,2))
   FG(1)=F1*COS(ABG(I,3))+FB2*SIN(ABG(I,3))
   FG(2)=FB2*COS(ABG(I,3))-F1*SIN(ABG(I,3))
   FG(3)=FB3
3. TEST FOR TARGET SHADOWING.
   DO 12 J=1,3
   IF(FG(J).GT.SDMAX(I,J).OR.FG(J).LT.SDMIN(I,J))GO TO 15
   12 CONTINUE

E) COUNT NUMBER OF ILLUMINATED TARGETS.

1. KTAR- # OF TARGETS ILLUMINATED
VII. UPDATE RANGE OF RADAR RELATIVE TO EACH TARGETS SPECULAR PT.

A) RANGE UPDATE

DO 20 K=1,KTAR
   I=JHOT(K)
   DO 20 J=1,3
      R(K,J)=TARG(I,J)+OFFSET(I)+COSPHI(I,J)
20 CONTINUE

IEE=1
IF (IEE.EQ.0) GO TO 24

B) RE-EVALUATE RCS FOR DIFFUSE SCATTERERS

DO 22 K=1,NWIDE
   I=JHOT(K)
   SIG(K)=SORT(ABS(COSPHN(I)))*SIGMA(I)
22 CONTINUE

24 RANGE=SQRT(RADAR(1)**2+RADAR(2)**2+RADAR(3)**2)

C) TEST FOR CLOSE RANGE

IF((ROLD.LT.0.1.OR.RANGE-ROLD.LE.0.) .AND. RANGE.LE.270.) ICLOSE=1
IF((RANGE-ROLD.GT.0.). .AND. RANGE.GT.360.) ICLOSE=0
IF(ICLOSE.EQ.0.OR.NWIDE.EQ.0) GO TO 55
IF(ICL0LD.EQ.1) GO TO 35

D) RANGE UPDATE FOR DIFFUSE SCATTERERS

1. PERFORMS INITIALIZATION OF DIFFERENCE EQUATIONS FOR ALL DIFFUSE SCATTERERS.

DO 30 I=1,KWIDE
   IF(COSPHN(I).GT.1.) COSPHN(I)=1.
   PHIOLD(I)=ACOS(COSPHN(I))
   V(1,J)=VSAM(I,J)*ZUDU(KSEED(I,J))+.5
   VOLD(I,J)=V(1,J)
30 CONTINUE

b) TRANSFORMATION OF "V" FROM TARGET COORDINATES TO SMS Coordinates.

TGAM1=V(1,1)*COS(ABG(I,1))+V(1,2)*SIN(ABG(I,1))
TGAM2=V(1,1)*SIN(ABG(I,1))+V(1,2)*COS(ABG(I,1))
TBETA2=V(1,2)*COS(ABG(I,1))+V(1,3)*SIN(ABG(I,1))
TBETA3=V(1,2)*SIN(ABG(I,1))+TBETA2+COS(ABG(I,1))*V(1,3)
V(1,1)=V(1,2)*COS(ABG(I,1))+V(1,3)*SIN(ABG(I,1))
V(1,2)=V(1,3)*TBETA3
DO 26 J=1,3
   R(I,J)=R(I,J)+V(I,J)
26 CONTINUE
CONTINUE
GO TO 55

2. UPDATES THE ANGLE BETWEEN THE RADAR VECTOR & THE TARGET NORMAL.

DO 40 I=1,NWIDE
PHI(I)=ACOS(COSPHN(I))
DPHI(I)=(PHI(I)-PHIOLD(I))
PHIOLD(I)=PHI(I)
40 CONTINUE

3. UPDATES THE RANGE COMPONENTS DUE TO RADAR BEAM DEFLECTION OVER THE SURFACE OF THE DIFFUSE SCATTERER. THE TRANSFORMATION PERFORMS THE SAME FUNCTION DESCRIBED PREVIOUSLY.

DO 58 K=1,NWIDE
I=JHOT(K)
DO 45 J=1,3
ALPH(I,J)=EXP(-DIM(I,J)*ABS(DPHI(I)*COSPHN(I)))
WRAN(I,J)=SORT(1.-ALPH(I,J)**2)*WSCALE(I,J)*(ZUDU(KSEED(I,J))-.5)
V(I,J)=ALPH(I,J)*VOLD(I,J)+WRAN(I,J)
VOLD(I,J)=V(I,J)
45 CONTINUE

TGAMI=V(I,1)*COS(ABG(I,3))-V(I,2)*SIN(ABG(I,3))
TGAM2=V(I,1)*SIN(ABG(I,3))+V(I,2)*COS(ABG(I,3))
TBETA2=COS(ABG(I,2))*TGAM2-SIN(ABG(I,2)),V(I,3)
TBETA3=SIN(ABG(I,2)),TGAM2+COS(ABG(I,2)),V(I,3)
V(I,1)=COS(ABG(I,1)),TBETA3-SIN(ABG(I,1)),TBETA2
V(I,2)=SIN(ABG(I,1)),TBETA3+COS(ABG(I,1)),TBETA2
V(I,3)=TBETA3
DO 46 J=1,3
R(K,J)=R(K,J)+V(I,J)
46 CONTINUE

CONTINUE
ICLOD=ICLOSE
RETURN
END

FUNCTION ZUDU(KSEED)
THIS SUBROUTINE GENERATES RANDOM NUMBERS.
DATA MU/524287/,XMU/524287./,IETA/997/
IF(KSEED)20,10,20
20 CONTINUE
KSEED=IETA+KSEED
IKEEP=KSEED/MU
KSEED=KSEED-IKEEP*MU
XRAN=KSEED
XRAN=XRAN/MU
ZUDU=XRAN
10 RETURN
END

subroutine readPAT

real allinear(41,41), eallinear(41,41)

Read in the sum, phase, and difference patterns
real sallinear( 41,41 ), sellinear( 41,41 )
real pallinear( 41,41 ), pellinear( 41,41 )
common / linear / allinear, ellinear
common / linear1 / sallinear, sellinear
common / linear2 / pallinear, pellinear

open( unit=3, file='[KUBAND.HOWARD.MARK]ozld.dat',
  access='sequential', form='unformatted',
  status='old', readonly )
read( 3 ) ( ( allinear( i,j ), j = 1,41 ), i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]elld.dat',
  access='sequential', form='unformatted',
  status='old', readonly )
read( 3 ) ( ( ellinear( i,j ), j = 1,41 ), i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]ozls.dat',
  access='sequential', form='unformatted',
  status='old', readonly )
read( 3 ) ( ( sallinear( i,j ), j = 1,41 ), i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]ells.dat',
  access='sequential', form='unformatted',
  status='old', readonly )
read( 3 ) ( ( sellinear( i,j ), j = 1,41 ), i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]ozlp.dat',
  access='sequential', form='unformatted',
  status='old', readonly )
read( 3 ) ( ( pallinear( i,j ), j = 1,41 ), i = 1,41 )
close( 3 )

open( unit=3, file='[KUBAND.HOWARD.MARK]ellp.dat',
  access='sequential', form='unformatted',
  status='old', readonly )
read( 3 ) ( ( pellinear( i,j ), j = 1,41 ), i = 1,41 )
close( 3 )
return
end
Subroutine: Antenna pattern interpolation.
Input: Azimuth and elevation angles in degrees.
Output: Interpolated difference, sum, and phase values for all 18 antenna patterns.

```
subroutine interp( az, el)

  Linearly interpolate the gain, phase and difference patterns

  real allinear( 41,41 ), ellinear( 41,41 )
  real sallinear(41,41), sellinear(41,41)
  real pallinear(41,41), pelinear(41,41)
  common / linear / allinear, ellinear
  common / linear1 / sallinear, sellinear
  common / linear2 / pallinear, pelinear
  common / SUDPH / X,Y,Z,PAZ,PEL

  iax = jint( ( az + 4. ) * 5. )
  iex = jint( ( el + 4. ) * 5. )
  az0 = floatj( iax ) / 5. - 4.
  el0 = floatj( iex ) / 5. - 4.
  iaz = jint( ( az + 4. ) * 5. ) + 1
  jel = jint( ( el + 4. ) * 5. ) + 1

  find azd values
  f0 = 10.** ( allinear( iaz, jel ) ) /20.
  f1 = 10.** ( allinear( iaz+1, jel ) ) /20.
  f2 = 10.** ( allinear( iaz,jel+1 ) ) /20.
  f3 = 10.** ( allinear( iaz+1,jel+1 ) ) /20.
  fa = f0 + (f1-f0)/2. * ( az-az0 )
  fb = f2 + (f3-f2)/2. * ( az-az0 )
  fx = fa + (fb-fa)/2. * ( el-el0 )

  Y = fx

  find eld values
  f0 = 10.** ( eallinear( iaz,jel ) ) /20.
  f1 = 10.** ( eallinear( iaz+1,jel ) ) /20.
  f2 = 10.** ( eallinear( iaz,jel+1 ) ) /20.
  f3 = 10.** ( eallinear( iaz+1,jel+1 ) ) /20.
  fa = f0 + (f1-f0)/2. * ( az-az0 )
  fb = f2 + (f3-f2)/2. * ( az-az0 )
  fx = fa + (fb-fa)/2. * ( el-el0 )
```

\[ Z = f_x \]

---

**find azs values**

\[
f_0 = 10.0 \times \text{soleinear}(i_{az}, j_{el}) / 20.0 \]
\[
f_1 = 10.0 \times \text{soleinear}(i_{az+1}, j_{el}) / 20.0 \]
\[
f_2 = 10.0 \times \text{soleinear}(i_{az}, j_{el+1}) / 20.0 \]
\[
f_3 = 10.0 \times \text{soleinear}(i_{az+1}, j_{el+1}) / 20.0 \]
\[
f_a = f_0 + (f_1 - f_0) / 2 \times (a_{z} - a_{z0}) \]
\[
f_b = f_2 + (f_3 - f_2) / 2 \times (a_{z} - a_{z0}) \]
\[
f_x = f_a + (f_b - f_a) / 2 \times (e_{el} - e_{el0}) \]

---

**X = f_x**

---

**find azp values**

\[
f_0 = \text{pallinear}(i_{az}, j_{el}) \]
\[
f_1 = \text{pallinear}(i_{az+1}, j_{el}) \]
\[
f_2 = \text{pallinear}(i_{az}, j_{el+1}) \]
\[
f_3 = \text{pallinear}(i_{az+1}, j_{el+1}) \]
\[
f_a = f_0 + (f_1 - f_0) / 2 \times (a_{z} - a_{z0}) \]
\[
f_b = f_2 + (f_3 - f_2) / 2 \times (a_{z} - a_{z0}) \]
\[
f_x = f_a + (f_b - f_a) / 2 \times (e_{el} - e_{el0}) \]

\[ \text{PAZ} = f_x \quad \text{! phase in degrees} \]

---

**c**

---

**findelp values**

\[
f_0 = \text{pellinear}(i_{az}, j_{el}) \]
\[
f_1 = \text{pellinear}(i_{az+1}, j_{el}) \]
\[
f_2 = \text{pellinear}(i_{az}, j_{el+1}) \]
\[
f_3 = \text{pellinear}(i_{az+1}, j_{el+1}) \]
\[
f_a = f_0 + (f_1 - f_0) / 2 \times (a_{z} - a_{z0}) \]
\[
f_b = f_2 + (f_3 - f_2) / 2 \times (a_{z} - a_{z0}) \]
\[
f_x = f_a + (f_b - f_a) / 2 \times (e_{el} - e_{el0}) \]

\[ \text{PEL} = f_x \quad \text{! phase in degrees} \]

---

**return**

**end**
APPENDIX C

LINE BY LINE LISTING OF DIFFERENCES BETWEEN

BASELINE PROGRAM AND DELIVERABLE PROGRAM

This appendix lists the lines which have been deleted from the baseline program and those which were added to form the deliverable program.

The deleted and added lines are grouped by program module, and identified by line number and the labels "LINES DELETED FROM BASELINE PROGRAM" or "LINES ADDED TO DELIVERABLE PROGRAM" immediately preceding the lines deleted or added. The line numbers for the deleted lines refer to lines in the original baseline program. The line numbers identifying the added lines are the line numbers in the final, deliverable program.
LINES ADDED TO DELIVERABLE PROGRAM

MODIFIED 01/27/86 TO COMPUTE AND PLOT REF. RANGE ACCELERATION.

MDMIN - KUBAND DATA : SSRNG, SSRDOT, SSRANG, SSPANG, SSRRTE, SSPRTE, SSALP, SSBE
WHITE SANDS - REF DATA : X, Y, Z, VX, VY, VZ

REF -> TMR2KU -> ACT : R, ARDOT, SPANG, SRANG, SPRTE, SALF, SBTA, SAZRTE, SELRTE

REF -> TMR2KU -> SIM : HRNG, HRDOT, HRANG, HPANG, HRRTE, HPRTE, HALP, HBET, HELRT, HALRT

COMMON /TARGET/ITARG.SRCS
COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTRE,AL,BT,SALF,SBTA
1 ,ER(3),EV(3),ERTO(3),AZRATE,ElRATE,SAZRTE,SELRTE
2 ,AX,AY,AZ,AX,AAY,AAZ,RCCEL
COMMON /TERM/TERM,XMO,XDAY,XYR,TBIAS,XJMO,XJDAY,XJYR
COMMON /OUTPUT/MWIF,MTF,MSF,HRNG,HRDOT,HPANG,HRANG,HPRT
2 ,HRTRE,HRSS,MADSF,MARDF,MARDVF,MRDDVF
3 ,HALP,HBET
COMMON /SYSDAT/TSA,DM2(14)
COMMON /TMR/X,Y,Z,VX,VY,VZ
1 1 ,DLP(3),DEL(3),DUE(3)
2 ,DSU(3),THAZL1,THEI1,THAZU1,A23
COMMON /INPUT/RO(3),VO(3),EME(3)
COMMON /ICNTL/IDUM(16),MPRF
CHARACTER ANS.REPLY
CHARACTER*11 FPRO(57)
CHARACTER*48 IXTP,LPRD(57)
CHARACTER*100 COMMENT
CHARACTER*11 UNIT7
INTEGER IREF
INTEGER+2 IS1,IS2
DIMENSION TP(2001),D(2001,43)
DIMENSION ITLIT(10)
DIMENSION RNEN(3),ROLD(3),VNEW(3),VOLD(3)
BYTE IC(128)

TEST DATA FROM WS32TDATA1
<table>
<thead>
<tr>
<th>Page</th>
<th>DATA LPRO</th>
<th>SIM DATA PROFILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>LPRO(1)</td>
<td>HL146AB$</td>
</tr>
<tr>
<td>50</td>
<td>LPRO(2)</td>
<td>HL246ABS$</td>
</tr>
<tr>
<td>51</td>
<td>LPRO(3)</td>
<td>HJ146ABS$</td>
</tr>
<tr>
<td>52</td>
<td>LPRO(4)</td>
<td>HEL3eAB$</td>
</tr>
<tr>
<td>53</td>
<td>LPRO(5)</td>
<td>H3eSKAB$</td>
</tr>
<tr>
<td>54</td>
<td>LPRO(6)</td>
<td>H3eSKAC$</td>
</tr>
<tr>
<td>55</td>
<td>LPRO(7)</td>
<td>HEL3eAC$</td>
</tr>
<tr>
<td>56</td>
<td>LPRO(8)</td>
<td>HEL3eAD$</td>
</tr>
<tr>
<td>57</td>
<td>LPRO(9)</td>
<td>HL246AC$</td>
</tr>
<tr>
<td>58</td>
<td>LPRO(10)</td>
<td>HL446ABS$</td>
</tr>
<tr>
<td>59</td>
<td>LPRO(11)</td>
<td>HL546ABS$</td>
</tr>
<tr>
<td>60</td>
<td>LPRO(12)</td>
<td>HL546AC$</td>
</tr>
<tr>
<td>61</td>
<td>LPRO(13)</td>
<td>HL246AD$</td>
</tr>
<tr>
<td>62</td>
<td>LPRO(14)</td>
<td>HL446AC$</td>
</tr>
<tr>
<td>63</td>
<td>LPRO(15)</td>
<td>HL346AD$</td>
</tr>
<tr>
<td>64</td>
<td>LPRO(16)</td>
<td>HL446AC$</td>
</tr>
<tr>
<td>65</td>
<td>LPRO(17)</td>
<td>HL346AD$</td>
</tr>
<tr>
<td>66</td>
<td>LPRO(18)</td>
<td>HL346AD$</td>
</tr>
<tr>
<td>67</td>
<td>LPRO(19)</td>
<td>HEL30AES$</td>
</tr>
<tr>
<td>68</td>
<td>LPRO(20)</td>
<td>HEL30AFS$</td>
</tr>
<tr>
<td>69</td>
<td>LPRO(21)</td>
<td>H3eSKADS$</td>
</tr>
<tr>
<td>70</td>
<td>LPRO(22)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>71</td>
<td>LPRO(23)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>72</td>
<td>LPRO(24)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>73</td>
<td>LPRO(25)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>74</td>
<td>LPRO(26)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>75</td>
<td>LPRO(27)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>76</td>
<td>LPRO(28)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>77</td>
<td>LPRO(29)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>78</td>
<td>LPRO(30)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>79</td>
<td>LPRO(31)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>80</td>
<td>LPRO(32)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>81</td>
<td>LPRO(33)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>82</td>
<td>LPRO(34)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>83</td>
<td>LPRO(35)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>84</td>
<td>LPRO(36)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>85</td>
<td>LPRO(37)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>86</td>
<td>LPRO(38)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>87</td>
<td>LPRO(39)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>88</td>
<td>LPRO(40)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>89</td>
<td>LPRO(41)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>90</td>
<td>LPRO(42)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>91</td>
<td>LPRO(43)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>92</td>
<td>LPRO(44)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>93</td>
<td>LPRO(45)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>94</td>
<td>LPRO(46)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>95</td>
<td>LPRO(47)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>96</td>
<td>LPRO(48)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>97</td>
<td>LPRO(49)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>98</td>
<td>LPRO(50)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>99</td>
<td>LPRO(51)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>100</td>
<td>LPRO(52)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>101</td>
<td>LPRO(53)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>102</td>
<td>LPRO(54)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>103</td>
<td>LPRO(55)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>104</td>
<td>LPRO(56)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>105</td>
<td>LPRO(57)</td>
<td>H3eSKAES$</td>
</tr>
<tr>
<td>106</td>
<td>DATA FPFO</td>
<td>HL146AB.XXX</td>
</tr>
<tr>
<td>107</td>
<td>DATA FPFO</td>
<td>HL246AB.XXX</td>
</tr>
<tr>
<td>108</td>
<td>DATA FPFO</td>
<td>HJ146AB.XXX</td>
</tr>
<tr>
<td>109</td>
<td>DATA FPFO</td>
<td>HEL30AB.XXX</td>
</tr>
<tr>
<td>110</td>
<td>DATA FPFO</td>
<td>H3eSKAB.XXX</td>
</tr>
<tr>
<td>111</td>
<td>DATA FPFO</td>
<td>H3eSKAC.XXX</td>
</tr>
<tr>
<td>112</td>
<td>DATA FPFO</td>
<td>HEL30AC.XXX</td>
</tr>
</tbody>
</table>
DATA FPRO(8)/'HEL30AD.XXX'/
DATA FPRO(9)/'HL246AC.XXX'/
DATA FPRO(10)/'HL346AB.XXX'/
DATA FPRO(11)/'HL446AB.XXX'/
DATA FPRO(12)/'HL546AB.XXX'/
DATA FPRO(13)/'HL546AC.XXX'/
DATA FPRO(14)/'HL246AD.XXX'/
DATA FPRO(15)/'HL446AC.XXX'/
DATA FPRO(16)/'HL146AC.XXX'/
DATA FPRO(17)/'HEL30AE.XXX'/
DATA FPRO(18)/'HEL30AF.XXX'/
DATA FPRO(21)/'H305KAD.XXX'/
DATA FPRO(22)/'H305KAEIF.XXX'/
DATA FPRO(23)/'H305KAF.XXX'/
DATA FPRO(24)/'HEL30AG.XXX'/
DATA FPRO(25)/'HEL30AH.XXX'/
DATA FPRO(26)/'H305KAG.XXX'/
DATA FPRO(27)/'H305KAG.XXX'/
DATA FPRO(28)/'H305KAI.XXX'/
DATA FPRO(29)/'HEL30AI.XXX'/
DATA FPRO(30)/'HEL30AJ.XXX'/
DATA FPRO(31)/'HL546AE.XXX'/
DATA FPRO(32)/'HL246AE.XXX'/
DATA FPRO(33)/'HL446AD.XXX'/
DATA FPRO(34)/'HL146AD.XXX'/
DATA FPRO(35)/'HL346AE.XXX'/
DATA FPRO(36)/'HL146AE.XXX'/
DATA FPRO(37)/'HL546AF.XXX'/
DATA FPRO(38)/'GEM1.XXX'/
DATA FPRO(39)/'GEM2.XXX'/
DATA FPRO(40)/'GEM3.XXX'/
DATA FPRO(41)/'SAT1.XXX'/
DATA FPRO(42)/'SAT2.XXX'/
DATA FPRO(43)/'SAT3.XXX'/
DATA FPRO(44)/'SAT4.XXX'/
DATA FPRO(45)/'SAT6.XXX'/
DATA FPRO(46)/'SAT8.XXX'/
DATA FPRO(47)/'BAL1.XXX'/
DATA FPRO(48)/'BAL2.XXX'/
DATA FPRO(49)/'BAL5.XXX'/
DATA FPRO(50)/'BAL6.XXX'/
DATA FPRO(51)/'BAL7.XXX'/
DATA FPRO(52)/'HL546AG.XXX'/
DATA FPRO(53)/'HL246AF.XXX'/
DATA FPRO(54)/'HL446AE.XXX'/
DATA FPRO(55)/'HL146AE.XXX'/
DATA FPRO(56)/'HL346AF.XXX'/
DATA FPRO(57)/'HL146AE.XXX'/

C-4
COMMON /TERM/ITERM
COMMON /OUTPUT/MSWF,TMF,MSF,SSRNG,SSRDOT,SSPANG,SSRANG,SSPRTE.
COMMON SSRRT,SSRSS,MADVF,MRDVF,MARDVF,MRDVF
COMMON /SYSDAT/TS,DUM2(14)
TEST DATA FROM WS32TDATA1
CHARACTER*9 FPRO(18)
CHARACTER*32 IXT,IYT(22),LPRO(18)
DATA IXT/'TIME SECONDS$'/
DATA IYT(1)://'RANGE FEET$'/
DATA IYT(2)://'RANGE RATE FT/SEC$'/
DATA IYT(3)://'ROLL ANGLE DEGS$'/
DATA IYT(4)://'PITCH ANGLE DEGS$'/
DATA IYT(5)://'ROLL RATE DEG/SEC$'/
DATA IYT(6)://'PITCH RATE DEG/SEC$'/
DATA IYT(7)://'ALPHA DEGS$'/
DATA IYT(8)://'BETA DEGS$'/
DATA IYT(9)://'AZ RATE DEG/SEC$'/
DATA IYT(10)://'EL RATE DEG/SEC$'/
DATA IYT(11)://'X (NORTH) FEET$'/
DATA IYT(12)://'Y (EAST) FEET$'/
DATA IYT(13)://'Z (ALTITUDE) FEET$'/
DATA IYT(14)://'ELEVATION ANGLE DEGS$'/
DATA IYT(15)://'DELTA RANGE FEET$'/
DATA IYT(16)://'DELTA RANGE RATE FT/SEC$'/
DATA IYT(17)://'DELTA ROLL ANGLE DEGS$'/
DATA IYT(18)://'DELTA PITCH ANGLE DEGS$'/
DATA IYT(19)://'DELTA ROLL RATE DEG/SEC$'/
DATA IYT(20)://'DELTA PITCH RATE DEG/SEC$'/
DATA IYT(21)://'DELTA ALPHA DEGS$'/
DATA IYT(22)://'DELTA BETA DEGS$'/
DATA LPRO(1)://'SIMULATION PROFILE HJ1465$'/
DATA LPRO(2)://'SIMULATION PROFILE HL1465$'/
DATA LPRO(3)://'SIMULATION PROFILE HL2465$'/
DATA LPRO(4)://'SIMULATION PROFILE HL3465$'/
DATA LPRO(5)://'SIMULATION PROFILE HL4465$'/
DATA LPRO(6)://'SIMULATION PROFILE HL5465$'/
DATA LPRO(7)://'SIMULATION PROFILE BJ1465$'/
DATA LPRO(8)://'SIMULATION PROFILE BL1465$'/
DATA LPRO(9)://'SIMULATION PROFILE BL2465$'/
DATA LPRO(10)://'SIMULATION PROFILE BL3465$'/
DATA LPRO(11)://'SIMULATION PROFILE BL4465$'/
DATA LPRO(12)://'SIMULATION PROFILE BL5465$'/
DATA LPRO(13)://'SIMULATION PROFILE C6P485$'/
DATA LPRO(14)://'SIMULATION PROFILE C6M485$'/
DATA LPRO(15)://'SIMULATION PROFILE C6P38$'/
DATA LPRO(16)://'SIMULATION PROFILE C6M38$'/
DATA LPRO(17)://'SIMULATION PROFILE CLP165$'/
DATA LPRO(18)://'SIMULATION PROFILE CLM165$'/
DIMENSION RID(128)
DATA FPRO(1)://'HJ146.JSC'/
DATA FPRO(2)://'HL146.BIN'/
DATA FPRO(3)://'HL246.BIN'/
DATA FPRO(4)://'HL346.BIN'/
DATA FPRO(5)://'HL446.BIN'/
DATA FPRO(6)://'HL546.BIN'/
DATA FPRO(7)://'BJ146.BIN'/
DATA FPRO(8)://'BL146.BIN'/
DATA FPRO(9)://'BL246.BIN'/
DATA FPRO(10)://'BL346.BIN'/
DATA FPRO(11)://'BL446.BIN'/
DATA FPRO(12)://'BL546.BIN'/
DATA FPRO(13)://'C6P48.BIN'/
DATA FPRO(14)://'C6M48.BIN'/
DATA FPRO( 15)'/C6P30.BIN'/
DATA FPRO( 16)'/C6M30.BIN'/
DATA FPRO( 17)'/CPL16.BIN'/
DATA FPRO( 18)'/CLM16.BIN'/
CHARACTER*9 UNIT7
BYTE IC(128)
COMMON /TMXR,Y,Z,VX,VY,VZ,
1 DLP(3),DEL(3),DUE(3),
2 DSU(3),THA2L1,THEL1,THAZU1
COMMON /INPUT/RO(3),VO(3),EWB(3)
DIMENSION TP(2001),D(2001,22)
WRITE (6,,)'1 : TEK'
WRITE (6,*)'2 : VT125'
WRITE (6,*)'3 : VT240'
WRITE (6,*)'4 : PC'
READ (5,*)ITERM
WRITE(6,*)'PROFILE NUMBER PROFILE'
DO L=1,18
WRITE(6,200)L,LPRO(L)
ENDDO
WRITE(6,*)'INPUT PROFILE NUMBER'
READ(5,*)ITAPE
WRITE(6,*)'ENTER NAME OF BINARY INPUT FILE'
READ(5,150)UNIT7
COMMON /FPRO(ITAPE)
OPEN (UNIT=UNIT7,FORM='UNFORMATTED',STATUS='OLD',FILE=UNIT7)
READ(4)IC
WRITE(6,15B)(IC(I),I=1,30)
FORMAT (68A2)
IFTRK=0
WRITE(6,*)' INPUT 1 IF YOU WANT TO FILTER USING TRACK FLAG'
READ(5,*)IFTRK
WRITE(6,*)'INPUT RSC IN SQUARE METERS'
READ(5,*)RCSM
************
************
LINES ADDED TO DELIVERABLE PROGRAM
ITARG=0
WRITE (6,*)'1 : TEK'
WRITE (6,*)'2 : VT125'
WRITE (6,*)'3 : VT240'
WRITE (6,*)'4 : PC'
READ (5,*)ITERM
WRITE (6,*)'ENTER : 1 IF YOU ARE PROCESSING TMR DATA'
WRITE (6,*)'2 IF YOU ARE PROCESSING CINE DATA'
WRITE (6,*)'3 IF YOU ARE PROCESSING BEST DATA'
READ (5,*)IREF
WRITE(6,*)'ENTER TIME INTERVAL ( 0,0 FOR THE WHOLE INTERVAL )'
READ(5,STIME,STTIME)
IF (STTIME.EQ.0)STTIME=999
WRITE (6,*)'DO YOU WANT TO FILTER THE DATA ? (Y/N)'
READ (5,2322)ANS
WRITE(6,*)'PROFILE NUMBER PROFILE'
DO L=1,19
WRITE(6,200)L,LPRO(L)
FORMAT(7X,I2,9X,A32)
ENDDO
WRITE (6,*)'ENTER C TO CONTINUE. Q TO QUIT :'
READ (5,101) REPLY
IF (REPLY.EQ.'C') THEN
  DO L=20,38
    WRITE(6,200)L,LPRO(L)
  ENDDO
  WRITE (6,*)'ENTER C TO CONTINUE, Q TO QUIT :'
  READ (5,101) REPLY
  IF (REPLY.EQ.'C') THEN
    DO L=39,57
      WRITE(6,200)L,LPRO(L)
    ENDDO
  ENDIF
ENDIF
WRITE(6.*)'INPUT PROFILE NUMBER'
READ(5.*)ITAPE
UNIT7=FPRO(ITAPE)
CALL FIXIT(ITILT,LPRO(ITAPE))
IF (ITAPE.LT.39.AND.ITAPE.GT.51)GO TO 39
IF (ITAPE.GE.38.AND.ITAPE.LE.51)GO TO 49
C
IF (IREF.EQ.1) THEN
  UNIT7(9:11)=JST'
ELSE IF (IREF.EQ.2) THEN
  UNIT7(9:11)=JSC'
ELSE
  UNIT7(9:11)=BST'
ENDIF
GO TO 59
IF (IREF.EQ.1) THEN
  UNIT7(6:B)=JST'
ELSE
  IF (IREF.EQ.2) THEN
    UNIT7(6:B)=JSC'
  ELSE
    UNIT7(6:B)=BST'
  ENDIF
ENDIF
OPEN(UNIT=4,FORM='UNFORMATTED',STATUS='OLD', FILE=UNIT7)
TOUT=0.
******
LINES DELETED FROM BASELINE PROGRAM
196 WRITE(6,*)'SRC_=',SRCS
198 TQUT=,_.
200 LINES ADDED TO DELIVERABLE PROGRAM
201 WRITE(6,'1 FOR SCREEN OUTPUT')
203 READ(5,*)TOUT
205 J=0
207 C READ START TIME
209 READ(4)TBIAS,GMTIME,XMO,XDAY,YR
210 ILOOP=1
212 CONTINUE
214 READ(4,END=99)T,SSRNG,SSRDOT,SSRANG,SSRTE,SSPRTE
216 1,X,Y,Z,VX,VY,VZ,AX,AY,AZ,IS1,IS2,RSS,RFPR,AEPR,BERR,ALFX,
218 1 BETY,SCRR,SCPR
220 IF (T.LT.TTIME) GOTO 1
222 IJJ=2*13
ITF=IAND(IS2, IJJ)
IF (ITF.NE.1 .AND. ANS.EQ. 'Y') GO TO 1
CALL RPAB(SSRANG,SSPANG,SSALP,SSBET)
CALL TMR2KU
DO I=1,3
RNEW(I)=RO(I)
VNEW(I)=VO(I)
END DO
IF(UPLOAD.EQ.1) GO TO 7
CALL EXEC
IF(MPRF.EQ.1) THEN
T1=.151
ELSE
T1=.119
END IF
IF(UPLOAD.EQ.1) THEN
T1=T1+TS
GO TO 196
END IF
CONTINUE
T1=T1+TS
IF(T1.GT.T) THEN
T1=T1-TS
GO TO 196
END IF
DO I=1,3
ROL(I)=RNEW(I)-ROLD(I)
VO(I)=VNEW(I)-VOLD(I)
END DO
HRRTE=HRRTE+180./3.14159*1000.
HPRT=HPRT+180./3.14159*1000.
J=J+1
IF(J.EQ.2) THEN
GO TO 99
END IF
TP(J)=T
CONTINUE
READ (4, END=99) T, X, Y, Z, VX, VY, VZ
TS=T1-T
WRITE(6,1) T, SSRANG, SSSRRDOT, SSPANG, SRANG, SSRTE, SRRT, SALF, SBTA,
AZRATE, ELRATE, AZRTE, ELRTE
CONTINUE
READ (4, END=99) T, X, Y, Z, VX, VY, VZ
C DATA IN METERS
CALL TMR2KU
IF(TOUT.EQ.1) THEN
WRITE(6,100) T, SSRNG, SSSRRDOD, SSPANG, SRANG, SSRTE, SRRT, SALF, SBTA,
AZRATE, ELRATE, AZRTE, ELRTE
FORMAT(' ',2F9.1,9F9.3)
END IF
CALL EXEC
IF(IFTRK.EQ.1 .AND. MTF.EQ.0) GO TO 1

LINES DELETED FROM BASELINE PROGRAM
DSU(3)=5.46
WRITE(6,'(A)') ' INPUT 1 FOR SCREEN OUTPUT'
READ(5,'(A)') TOUT
J=0
READ (4, END=99) T, X, Y, Z, VX, VY, VZ
READ (4, END=99) T, X, Y, Z, VX, VY, VZ
TS=T1-T
WRITE(6,'(A)') ' TS=', TS
CONTINUE
READ (4, END=99) T, X, Y, Z, VX, VY, VZ
C DATA IN METERS
CALL TMR2KU
IF(TOUT.EQ.1) THEN
WRITE(6,100) T, SSRNG, SSSRRDOD, SSPANG, SRANG, SSRTE, SRRT, SALF, SBTA,
AZRATE, ELRATE, AZRTE, ELRTE
FORMAT(' ',2F9.1,9F9.3)
END IF
CALL EXEC
IF(IFTRK.EQ.1 .AND. MTF.EQ.0) GO TO 1

C-8
J=J+1
IF(J.EQ.2001)GO TO 99
TP(J)=T

************
LINES ADDED TO DELIVERABLE PROGRAM
D(J,3)=SSRANG
D(J,4)=SSPANG
D(J,5)=SSRTE

************
LINES DELETED FROM BASELINE PROGRAM
D(J,4)=SSPANG
D(J,3)=SSRANG
D(J,5)=SSRTE

************
LINES ADDED TO DELIVERABLE PROGRAM
D(J,9)=HRNG
D(J,10)=HRDOT
D(J,11)=RO(1)
D(J,12)=RO(2)
D(J,13)=RO(3)
D(J,14)=ATAND(-RO(3)/SQRT(RO(1)+RO(1)+RO(2)+RO(2)))
D(J,15)=SSRNG-R

************
LINES DELETED FROM BASELINE PROGRAM
D(J,9)=AZRTE
D(J,10)=ELRTE
D(J,11)=Y
D(J,12)=Z
D(J,13)=Z
D(J,14)=ATAND(-Z/(X*X+Y*Y))
D(J,15)=SSRNG-R

************
LINES ADDED TO DELIVERABLE PROGRAM
D(J,19)=SSRTE-SRTE
D(J,20)=SSPRTE-SPRTE

************
LINES DELETED FROM BASELINE PROGRAM
D(J,19)=SSRTE-SRTE
D(J,20)=SSPRTE-SPRTE

************
LINES ADDED TO DELIVERABLE PROGRAM
D(J,23)=AZRTE
D(J,24)=ELRTE
D(J,25)=RSS
D(J,26)=RFWR
D(J,27)=AEWR
D(J,28)=BERR
D(J,29)=ALFX
D(J,30)=BETY
D(J,31)=SCR
D(J,32)=SCR
D(J,33)=0
D(J,34)=0
D(J,35)=HRNG-R
D(J,36)=HRDOT-ARDOT
D(J,37)=HRANG-SRANG
IF(J.GT.2e+0)THEN
    WRITE(6,*), 'MORE THAN 2000 POINTS'
    STOP
ENDIF
GO TO 1
CONTINUE
J=J-1
IXD,J=1
CONTINUE
CALL SORT(TP,D,J,ITILT,IXD,IYD,GMTIME,IREF)
GO TO 94
END

SUBROUTINE SORT(T,D,J,ITILT,IXD,IYD,GMTIME,IREF)
DIMENSION D(2e+1,43),X(2e+1),Y(2e+1),T(2e+1)
CHARACTER*48
IXT,IYT(43),PRONAME
CHARACTER*4 REFF
DIMENSION ITILT(1e),IXL(1e),IYL(18)
DATA IXT/'TIME SEC/SECONDS$'/
DATA IYT(1)/'KU MDM RANGE FEET$'/
DATA IYT(2)/'KU MDM RANGE RATE FT/SEC$'/
DATA IYT(3)/'KU MDM ROLL ANGLE DEG$'/
DATA IYT(4)/'KU MDM PITCH ANGLE DEG$'/
DATA IYT(5)/'KU MDM ROLL RATE DEG/SEC$'/
DATA IYT(6)/'KU MDM PITCH RATE DEG/SEC$'/
DATA IYT(7)/'KU MDM ALPHA DEG$'/
DATA IYT(8)/'KU MDM BETA DEG$'/
DATA IYT(9)/'SIM RANGE FEET$'/
DATA IYT(10)/'SIM RANGE RATE FT/SEC$'/
DATA IYT(11)/'WSMR X (NORTH) FEET$'/
DATA IYT(12)/'WSMR Y (EAST) FEET$'/
DATA IYT(13)/'WSMR Z (ALITUDE) FT$'/
DATA IYT(14)/'WSMR ELEVATION ANGLE DEG$'/
DATA IYT(15)/'DELTA RANGE FEET ( KU - WSMR )$'/
DATA IYT(16)/'DELTA RANGE RATE FT/SEC ( KU - WSMR )$'/
DATA IYT(17)/'DELTA ROLL ANGLE DEG ( KU - WSMR )$'/
DATA IYT(18)/'DELTA PITCH ANGLE DEG ( KU - WSMR )$'/
DATA IYT(19)/'DELTA ROLL RATE DEG/SEC ( KU - WSMR )$'/
DATA IYT(20)/'DELTA PITCH RATE DEG/SEC ( KU - WSMR )$'/
DATA IYT(21)/'DELTA ALPHA DEG ( KU - WSMR )$'/
DATA IYT(22)/'DELTA BETA DEG ( KU - WSMR )$'/
DATA IYT(23)/'DELTA AZ ANGLE DEG/SEC$'/
DATA IYT(24)/'DELTA ELE ANGLE DEG/SEC$'/
DATA IYT(25)/'KU SCANNER RSS ( VOLTS )$'/
DATA IYT(26)/'KU SCANNER RF POWER ( VOLTS )$'/
DATA IYT(27)/'KU SCANNER ALPHA ERROR ( VOLTS )$'/
DATA IYT(28)/'KU SCANNER BETA ERROR ( VOLTS )$'/
DATA IYT(29)/'KU SCANNER ALPHA X ( VOLTS )$'/
DATA IYT(30)/'KU SCANNER BETA Y ( VOLTS )$'/
DATA IYT(31)/'KU SCANNER ROLL RATE ( VOLTS )$'/
DATA IYT(32)/'KU SCANNER PITCH RATE ( VOLTS )$'/
DATA IYT(33)/'SIM RADAR CROSS SECTION ( DBSM )$'/
DATA IYT(34)/'WSMR RANGE ACCELERATION FT/SEC/SEC$'/
DATA IYT(35)/'DELTA RANGE FEET (SIM-WSMR)$'/
DATA IYT(36)/'DELTA RANGE RATE FT/SEC (SIM-WSMR)$'/
DATA IYT(37)/'DELTA ROLL ANGLE DEG (SIM-WSMR)$'/
DATA IYT(38)/'DELTA PITCH ANGLE DEG (SIM-WSMR)$'/
DATA IYT(39)/'DELTA ROLL RATE DEG/SEC (SIM-WSMR)$'/
DATA IYT(40)/'DELTA PITCH RATE DEG/SEC (SIM--WSMR)$'/
DATA IYT(41)/'DELTA ALPHA DEG (SIM--WSMR)$'/
DATA IYT(42)/'DELTA BETA DEG (SIM--WSMR)$'/
DATA IYT(43)/'SIM RADAR SIGNAL STRENGTH$'/
IFLAG=1
IF (IREF.EQ.1)THEN
  REFF='TMR'
ELSE IF (IREF.EQ.2) THEN
  REFF='CINE'
ELSE
  REFF='BEST'
ENDIF
DO I=1,43
  L=INDEX(IYT(I),'WSMR °
  IF (L .GT. 1) THEN
    IYT(I)(L:L+3) = REFF
  ENDIF
ENDDO
CONTINUE
DO I=1,43
  WRITE(6,68)I,IYT(I)
FORMAT(1X,10X,A4e)
ENDDO
WRITE(6,*)'INPUT IXD,IYD IXD=0 FOR TIME'
IF (IFLAG.EQ.0) THEN
  IFLAG=1
  IXD=0
  IYD=1
  GO TO 731
ENDIF
READ(5,*)IXD,IYD
IF(IXD.EQ.0) THEN
  DO L=1,J
    WRITE(6,68)I, IYT(L)
  FORMAT (1X,14.1ex,A32)
  ENDDO
WRITE(6,*)'INPUT IXD,IYD IXD=0 FOR TIME'
READ(5,*)IXD,IYD
CALL SORT(TP,D,J, ITAPE, IXD, IYD)
GO TO 731
END
SUBROUTINE SORT(T,D,J, ITAPE, IXD, IYD)
DIMENSION D(2e11,22), x(2e11), T(2e11)
CHARACTER-32 IXT, IYT(22), LPRO(18)
DIMENSION TILT(8),XL(8), YL(8)
DATA IXT/'TIME SECONDS$'/
DATA IYT(1)/'RANGE FEET$'/
DATA IYT(2)/'RANGE RATE (FT/SEC$'/
DATA IYT(3)/'ROLL ANGLE DEG$'/
DATA IYT(4)/'PITCH ANGLE DEG$'/
DATA IYT(5)/'ROLL RATE DEG/SEC$'/
DATA IYT(6)/'PITCH RATE DEG/SEC$'/
DATA IYT(7)/'ALPHA DEG$'/
DATA IYT(8)/'BETA DEG$'/
DATA IYT(9)/'AZ RATE DEG/SEC$'/

198 DATA ITY(18)'/EL RATE DEG/SEC$' /
199 DATA ITY(11)'/ X (NORTH) FEET$' /
200 DATA ITY(12)'/ Y (EAST) FEET$' /
201 DATA ITY(13)'/-Z (ALTITUDE) FEET$' /
202 DATA ITY(14)'/ ELEVATION ANGLE DEG$' /
203 DATA ITY(15)'/ DELTA RANGE FEET$' /
204 DATA ITY(16)'/ DELTA RANGE RATE FT/SECS$' /
205 DATA ITY(17)'/ DELTA ROLL ANGLE DEG$' /
206 DATA ITY(18)'/ DELTA PITCH ANGLE DEG$' /
207 DATA ITY(19)'/ DELTA ROLL RATE DEG/SEC$' /
208 DATA ITY(20)'/ DELTA PITCH RATE DEG/SEC$' /
209 DATA ITY(21)'/ DELTA ALPHA DEG$' /
210 DATA ITY(22)'/ DELTA BETA DEG$' /
211 DATA LPRO(1)'/ SIMULATION PROFILE HJ1465$' /
212 DATA LPRO(2)'/ SIMULATION PROFILE HL1465$' /
213 DATA LPRO(3)'/ SIMULATION PROFILE HL2465$' /
214 DATA LPRO(4)'/ SIMULATION PROFILE HL3465$' /
215 DATA LPRO(5)'/ SIMULATION PROFILE HL4465$' /
216 DATA LPRO(6)'/ SIMULATION PROFILE HL5465$' /
217 DATA LPRO(7)'/ SIMULATION PROFILE BJ1465$' /
218 DATA LPRO(8)'/ SIMULATION PROFILE BL1465$' /
219 DATA LPRO(9)'/ SIMULATION PROFILE BL2465$' /
220 DATA LPRO(10)'/ SIMULATION PROFILE BL3465$' /
221 DATA LPRO(11)'/ SIMULATION PROFILE BL4465$' /
222 DATA LPRO(12)'/ SIMULATION PROFILE BL5465$' /
223 DATA LPRO(13)'/ SIMULATION PROFILE CBP4B$' /
224 DATA LPRO(14)'/ SIMULATION PROFILE CBM4B$' /
225 DATA LPRO(15)'/ SIMULATION PROFILE CBP3B$' /
226 DATA LPRO(16)'/ SIMULATION PROFILE CBM3B$' /
227 DATA LPRO(17)'/ SIMULATION PROFILE CLP165$' /
228 DATA LPRO(18)'/ SIMULATION PROFILE CLM165$' /
229 JPRO=ITAPE
230 CALL FIXIT(ITILT,LPRO(JPRO))
231 IF(IXD.EQ.0)THEN
232 DO 1=1,J
233 END
***********
LINES ADDED TO DELIVERABLE PROGRAM
461 CALL PLOTIT(ITILT,IYL,X,Y,J,GMTIME,IYD,IXD)
462 GO TO 1
463 CONTINUE
464 RETURN
465 END
466 C ***********************************************************************
467 SUBROUTINE FIXIT(IOUT,IN)
468 DIMENSION IOUT(18)
469 CHARACTER 4 ITEMP(10)
470 CHARACTER=40 IN
471 ITEMP(1)=(IN(1:4))
******
LINES DELETED FROM BASELINE PROGRAM
246 CALL PLOTIT(ITILT,IYL,X,Y,J)
247 RETURN
248 END
249 SUBROUTINE FIXIT(IOUT,IN)
250 DIMENSION IOUT(8)
251 CHARACTER=4 ITEMP(8)
252 CHARACTER=32 IN
253 ITEMP(1)=(IN(1:4))
******
LINES ADDED TO DELIVERABLE PROGRAM
479 ITEMP(9)=(IN(33:36))
480 ITEMP(10)=(IN(37:40))
481  ENCODE(48,999,1OUT)(ITEMP(1),I=1,10)
482  999  FORMAT(10A4)
483  RETURN
484  END
485  C ******************************************************
486  SUBROUTINE PLOTIT(ITILT,IXL.IYL.X,Y,J,GMTIME.IYD,IXD)
487  COMMON /TERM/ITERM,X,Y,J,GMTIME,IXD
488  COMMON/TMR/A,B,C,D,E,F,G(3),AH(3),AI(3),AJ(3),THAZL1,THELI,THAZU1
489  DOUBLE PRECISION SIG,AVG
490  BYTE CR(2)
491  DIMENSION ITILT(8),IXL(8),IYL(8)
492  DIMENSION X(1),Y(1),TINL(3)
493  WRITE(6,*('1 FOR MEAN AND STANDARD DEVIATION OF Y'))
494  READ(S,.)ISTA
495  NSC=0
496  XMAX=X(1)

LINES DELETED FROM BASELINE PROGRAM
261  ENCODE(32,999,1OUT)(ITEMP(1),I=1,8)
262  999  FORMAT(8A4)
263  RETURN
264  END
265  SUBROUTINE PLOTIT(ITILT,IXL,IYL,X,Y,J)
266  COMMON /TERM/ITERM
267  DIMENSION ITILT(8),IXL(8),IYL(8)
268  DIMENSION X(1),Y(1)
269  BYTE CR(2)
270  COMMON/TMR/A,B,C,D,E,F,G(3),AH(3),AI(3),AJ(3),THAZL1,THELI,THAZU1
271  CR(1)=27
272  CR(2)=12
273  XMAX=X(1)

LINES ADDED TO DELIVERABLE PROGRAM
500  GMHOUR=GMTIME/60./60.
501  GMHOUR=INT(GMHOUR)
502  GMMIN=(GMHOUR1-GMHOUR)*60.
503  GMMIN=INT(GMMIN)
504  GMSEC=INT((GMMIN-GMIN)*60.)
505  DO I=1,J

LINES DELETED FROM BASELINE PROGRAM
285  DO I=1,J

LINES ADDED TO DELIVERABLE PROGRAM
513  IF(YMAX.EQ.YMIN)YMAX=0.1
514  2 CONTINUE
515  YMAX=YMAX
516  YMIN=YMIN
517  IF (ITERM.EQ.1) CALL TEKALL(41480010)

LINES DELETED FROM BASELINE PROGRAM
285  IF (ITERM.EQ.1) CALL TEKALL(41480010)

LINES ADDED TO DELIVERABLE PROGRAM
520  IF (IYD.EQ.1)CALL RINTL(X,Y,J,TINL,NTINL)
521  CALL BGNPL(-1)
522  CALL FLATBD
523  CALL AREA2D (9.0,14.0)
524  CALL HEIGHT(.45)
525  CALL TITLE(ITILT100,IXL100,IYL100,9013.5)

C-13
CALL MESSAG('ITILT, 100, -0.6, 16.5)
CALL RESET ('HEIGHT')
CALL HEIGHT (.3)
1100=100
C 0.6 WAS SUBTRACTED TO CENTER AND 1 INCH WAS ADDED IN HEIGHT
CALL MESSAG('TEST DATE$', '-0.7, 15.5')
IF (XMO.GE.10) THEN
CALL REALNO(XMO, 0, 3.0, 15.5)
ELSE
CALL REALNO(XMO, 0, 3.3, 15.5)
ENDIF
CALL REALNO(XDAY, 0, 3.9, 15.5)
IF (XDAY.GE.10) THEN
CALL REALNO(XYR, 0, 4.8, 15.5)
ELSE
CALL REALNO(XYR, 0, 4.5, 15.5)
ENDIF
CALL MESSAG('TEST DATES', 100, 6.0, 15.5)
X-POSITION MOVED FORWARD BY 1.2
CALL MESSAG('TO= GMT=', 100, 1.2, 14.2)
CALL REALNO(GMTIME, 0, 1.1, 14.2)
CALL REALNO(GMHR, 0, 5.1, 14.2)
CALL REALNO(GMMIN, 0, 6.0, 14.2)
CALL REALNO(GMSEC, 0, 6.9, 14.2)
IF (ISTA.EQ.1) THEN
AVG=0
SIG=0
DO I=1,J
AVG=AVG+Y(I)
SIG=SIG+Y(I)**2
END DO
AVG=AVG/J
SIG=SQR((SIG/J-AVG)**2)
CALL MESSAG('MEAN=', 1100, 0.9, -2.0)
CALL REALNO(AVG, 3, 'ABUT', 'ABUT')
CALL MESSAG('STANDARD DEVIATION=', 1100, 3.3, -2.0)
CALL REALNO(SIG, 3, 'ABUT', 'ABUT')
ENDIF
CALL XNAME(IYL, 100)
CALL YNAME(IYL, 100)
CALL INTAX5
CALL YAXANG(0.)
IF (NSC.EQ.0) THEN
CALL GRAF(XMIN, 'SCALE', XMAX, YMIN, 'SCALE', YMAX)
ENDIF
IF (NSC.EQ.1) THEN
CALL GRAF(XMIN, 'SCALE', XMAX, YMIN, 'SCALE', YMAX)
ENDIF
IF (NTNL.NE.0 .AND. IXD.EQ.0) THEN
DO K=1,NTNL
IVEC=1302
CALL RLVEC (TINL(K), YMIN1, TINL(K), YMAX1, IVEC)
END DO
CALL CURVE(X, Y, J, 0)
CALL GRID(1,1)

LINES DELETED FROM BASELINE PROGRAM
CALL BGNP1L(-1)
CALL FLATBD
CALL PAGE(14., 18.)
CALL HEIGHT(.3)
CALL TITLE('ITILT, 100, IXL, 100, IYL, 100, 9.0, 13.5')
CALL MESSAG('LOWER AZIMUTH=$', I100,1.7,13.)
CALL REALNO(THAZL1,2, 'ABUT', 'ABUT')
CALL MESSAG('ELEVATION=$', I100,1.7,12.)
CALL REALNO(THEL1,2, 'ABUT', 'ABUT')
CALL BLNK1(1.5,7.5,11.9,13.5,4)
CALL HEADIN(ITILT,-100, -4.4)
CALL HEADIN('LOWER AZIMUTH=$',100,4,4)
CALL REALNO(THAZL1,2, 'ABUT', 'ABUT')
CALL HEADIN('UPPER AZIMUTH=$',100,4,4)
CALL REALNO(THAZU1,2, 'ABUT', 'ABUT')
CALL HEADIN('ELEVATION=$',100,4,4)
CALL REALNO(THEL1,2, 'ABUT', 'ABUT')
CALL YAXANG(0.)
CALL GRAF(XMIN, 'SCALE', XMAX, YMIN, 'SCALE', YMAX)
CALL CURVE(X,Y, J, 0)
KK=J/30
K=0
DO I=1, KK
  CALL RLINT(K,X(K),Y(K))
ENDDO
CALL GRID(I,1)

************
LINES ADDED TO DELIVERABLE PROGRAM
CALL DONEPL
CR(1)=27
CR(2)=12
WRITE(6,888)CR
888 FORMAT('INPUT 1 TO CHANGE SCALE OF Y AXIS')
WRITE(5,*)NSC
IF(NSC.EQ.1)THEN
  WRITE(6,*)'YMAX--',YMAX,' YMIN--', YMIN
  READ(5, *)YMAX
  READ(6,*)YMIN
  GO TO 2
ENDIF

************
LINES DELETED FROM BASELINE PROGRAM
CALL DONEPL
C MICKEY MOUSE FIX
IMM=1
IF(IMM.EQ.0)THEN
  REMIND (5)
  READ(5,192)IC
192 FORMAT(A1)
WRITE(6,888)CR
ENDIF

************
LINES ADDED TO DELIVERABLE PROGRAM
SUBROUTINE RPAB(ROLL0,PITCHQ,ALPHA,BETA)
DEGRAD=57.29576
PSI=67./DEGRAD
PIT=PITCHQ/DEGRAD
ROL = ROLLO/DEGRAD
XB = SIN(PIT)
YB = (SIN(ROL)) * SORT(1.0 - XB * XB)
Z = SORT(1.0 - XB * XB - YB * YB)
IF(ROLLQ.LE.90.0.AND.ROLLO.GE.-90.0) Z = -Z
XR = XB * COS(PSI) + YB * SIN(PSI)
YR = YB * COS(PSI) - XB * SIN(PSI)
YRZ = SORT(YR + YR + Z * Z)
ALF = ASIN(YR/YRZ)
BTA = ASIN(-XR/SORT(XR + XR + YR + YR + Z * Z))
ALPHA = ALF * DEGRAD
BETA = BTA * DEGRAD
IF(Z.GE.0.0.AND.YR.LE.0.0) ALPHA = (180.0 + ALPHA)
IF(Z.GE.0.0.AND.YR.GT.0.0) ALPHA = (180.0 - ALPHA)
RETURN
END

C ***************************************************
SUBROUTINE RINTL(T,R,N,TI,J)
DIMENSION RI(5), R(1), DS(5), TI(30), T(1)
DATA RI / 2558., 5758., 11518., 23030., 43518. /
RMAX = R(1)
RMIN = R(1)
DO 1 I = 1, N
RMAX = AMAX1(RMAX, R(I))
RMIN = AMIN1(RMIN, R(I))
1 CONTINUE
MRMAX = 1
MRMIN = 1
DO 2 I = 1, 5
IF(RMAX.GT.RI(1)) MRMAX = I
IF(RMIN.GT.RI(I)) MRMIN = I
2 CONTINUE
J = 0
IF(MRMAX.EQ.J .AND. MRMIN) RETURN
J = 0
DO 3 L = 1, 5
DS(L) = R(1) - RI(L)
3 CONTINUE
DO 4 I = 1, N
DO 5 L = 1, 5
IF((R(I) - RI(L)) * DS(L) .LT. 0) THEN
J = J + 1
T(I) = T(1)
DS(L) = R(I) - RI(L)
5 CONTINUE
END IF
4 CONTINUE
RETURN
END

C ***************************************************
C
C
INPUT VIA COMMON VIA X,Y,Z,VX,VY,VZ,AX,AY,AZ
OUTPUT VIA COMMON /ACTDAT/

LINES DELETED FROM BASELINE PROGRAM
SUBROUTINE TMR2KU
C
C
C
INPUT VIA COMMON VIA X,Y,Z,VX,VY,VZ
OUTPUT VIA COMMON /ACTDAT/

C-16
LINES ADDED TO DELIVERABLE PROGRAM
741 SUBROUTINE TMR2KU
742 COMMON /TMR/X,Y,Z,VX,VY,VZ,
743 1 DLP(3),DEL(3),DUE(3),
744 2 DSU(3),THAZL1,THEL1,THAZU1,A23
745 COMMON /INPUT/RO(3),VO(3),EMB(3)
746 COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,DBTA,
747 1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
748 2,AX,AY,AZ,AAX,AAY,AAZ,RACCEL
749 C DIMENSION DLP(3),DEL(3),DUE(3),DSU(3)

LINES DELETED FROM BASELINE PROGRAM
418 COMMON /TMR/X,Y,Z,VX,VY,VZ,
419 1 DLP(3),DEL(3),DUE(3),
420 2 DSU(3),THAZL1,THEL1,THAZU1
421 COMMON /INPUT/RO(3),VO(3),EMB(3)
422 COMMON /ACTDAT/R,ARDOT,SPANG,SRANG,SPRTE,SRRTE,AL,BT,SALF,DBTA,
423 1ER(3),EV(3),ERTO(3),AZRATE,ELRATE,AZRTE,ELRTE
424 C DIMENSION DLP(3),DEL(3),DUE(3),DSU(3)

LINES ADDED TO DELIVERABLE PROGRAM
754 DIMENSION APT(3),ALAZ(3),AELV(3),AST(3)
755 DATA DEGRAD/57.275/,PI/3.14159/

LINES DELETED FROM BASELINE PROGRAM
429 DATA DEGRAD/57.275/,PI/3.14159/

LINES ADDED TO DELIVERABLE PROGRAM
780 VPT(3) = VZ
781 APT(1) = AX
782 APT(2) = AY
783 APT(3) = AZ
784 C

LINES DELETED FROM BASELINE PROGRAM
454 VPT(3) = VZ
455 C

LINES ADDED TO DELIVERABLE PROGRAM
838 C CONVERT TO VELOCITIES REFERENCED TO GIMBALS
839 CALL MULT31(AZL,VPT,VLAZ)
840 CALL MULT31(ELV,VLAZ,VELV)
841 CALL MULT31(AZU,VELV,VST)
842 C CONVERT TO ACCELERATIONS REFERENCED TO GIMBALS
843 CALL MULT31(AZL,APT,ALAZ)
844 CALL MULT31(ELV,ALAZ,AELV)
845 CALL MULT31(AZU,AELV,AST)
846 C THESE ARE VELOCITIES IN GIMBAL REFERENCE.

LINES DELETED FROM BASELINE PROGRAM
509 C CONVERT TO VELOCITIES REFERENCED TO GIMBALS
510 CALL MULT31(AZL,VPT,VLAZ)
511 CALL MULT31(ELV,VLAZ,VELV)
512 CALL MULT31(AZU,VELV,VST)
513 C THESE ARE VELOCITIES IN GIMBAL REFERENCE.

LINES ADDED TO DELIVERABLE PROGRAM
854 C23=COSD(A23)
855 S23=SIND(A23)
856 X1=RO(2)*C23-RO(3)*S23
LINES DELETED FROM BASELINE PROGRAM
521  C23=COSD(23.)
522  S23=SIND(23.)
523  X1=RO(2)*C23-RO(3)*S23

LINES ADDED TO DELIVERABLE PROGRAM
868  AAX=AST(2)*C23-AST(3)*S23
869  AAY=AST(2)*S23-AST(3)*C23
870  AAZ=AST(1)
871  CALL ACT

LINES DELETED FROM BASELINE PROGRAM
535  CALL ACT

LINES ADDED TO DELIVERABLE PROGRAM
878  RETURN
879  END
880  C SUBROUTINE AZGEN(AZ,ANGAZ)
881  SUBROUTINE AZGEN(AZ,ANGAZ)

LINES DELETED FROM BASELINE PROGRAM
542  C THE EXAMPLE CASE RESULTS ARE:
543  C WRITE(6,*)R,ARDOT
544  C WRITE(6,*)SRANG,SPANG
545  C WRITE(6,*)SRTE,SPRTE
546  C WRITE(6,*)SAF,SBTA
547  C WRITE(6,*)AZRTE,ELRTE
548  RETURN
549  END
550  SUBROUTINE AZGEN(AZ,ANGAZ)

LINES ADDED TO DELIVERABLE PROGRAM
895  C SUBROUTINE ELGEN(EL,ANGEL)
896  SUBROUTINE ELGEN(EL,ANGEL)

LINES DELETED FROM BASELINE PROGRAM
564  SUBROUTINE ELGEN(EL,ANGEL)

LINES ADDED TO DELIVERABLE PROGRAM
908  C SUBROUTINE ACT
909  C SUBROUTINE ACT

LINES DELETED FROM BASELINE PROGRAM
576  C SUBROUTINE ACT

LINES ADDED TO DELIVERABLE PROGRAM
919  C SUBROUTINE ACT
920  SUBROUTINE ACT

LINES DELETED FROM BASELINE PROGRAM
586  SUBROUTINE ACT

LINES ADDED TO DELIVERABLE PROGRAM
923  COMMON /INPUT/ ERT(3),EVT(3),EWB(3),DUM(18)
924

LINES DELETED FROM BASELINE PROGRAM

C-18
LINES ADDED TO DELIVERABLE PROGRAM
973  C  COMPUTE RANGE ACCELERATION TO TARGET.
974  VSQ=EV(1)**2+EV(2)**2+EV(2)**2
975  RACCEL=(VSQ+ER(1)*AX+ER(2)*AAY+ER(3)*AZ-ARDOT**2)/R
976  C  COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).

******

LINES DELETED FROM BASELINE PROGRAM
658  C  COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).

******

LINES ADDED TO DELIVERABLE PROGRAM
1059  DR(1)=0.0
1060  DR(2)=11.130
1061  DR(3)=5.79
1062  C  RANGE BIAS ERROR IS COMPUTED IN SUBROUTINE RTRACK AS

******

LINES DELETED FROM BASELINE PROGRAM
721  DR(1)=45.738
722  DR(2)=11.130
723  DR(3)=5.79
724  C  RANGE BIAS ERROR IS COMPUTED IN SUBROUTINE RTRACK AS

******

LINES ADDED TO DELIVERABLE PROGRAM
1072  PSBIAS=PII=0.0
1073  C  ROLL ANGLE ERROR.
1074  C  ROLL ANGLE ERROR.
1075  RLBIAS=PII=0.0
1076  C  PITCH ANGLE ERROR.
1077  PTBIAS=PII=0.0
1078  C

******

LINES DELETED FROM BASELINE PROGRAM
734  PSBIAS=PII=0.1
735  C  ROLL ANGLE ERROR.
736  C  ROLL ANGLE ERROR.
737  RLBIAS=PII=0.25
738  C  PITCH ANGLE ERROR.
739  PTBIAS=PII=0.25
740  C

******

LINES ADDED TO DELIVERABLE PROGRAM
1081  NBIAS=0
1082  IF(NBIAS.NE.0)GO TO 700

******

LINES DELETED FROM BASELINE PROGRAM
743  NBIAS=1
744  IF(NBIAS.NE.0)GO TO 700

******

LINES ADDED TO DELIVERABLE PROGRAM
1230  REAL INTT,K4,K5,K6
1231  INTEGER ATIA(10,2),AT1E(10,2),AT2A(10,2),AT2E(10,2)
1232  COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)

******

LINES DELETED FROM BASELINE PROGRAM
892  REAL INTT,IAZDSC,IELDSC
893  COMMON /CNTL/IPWR,IMODE,IDUMC(7),DUMC(3)

******
LINES ADDED TO DELIVERABLE PROGRAM
1242  DIMENSION TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3) 00025450
1243  DIMENSION TDC(3)
1244  C******************************************************************************
1245  C
1246  C ATRACK MODIFIED JAN 28 1986 BY W. MEYER
1247  C MODIFICATIONS TO SUBROUTINE ATRACK WERE IMPLEMENTED
1248  C TO UPDATE THE LOOP CONSTANTS AND MORE ACCURATELY
1249  C SIMULATE THE ACTUAL SIGNAL PROCESSING PERFORMED
1250  C BY THE RADAR
1251  C
1252  C******************************************************************************
1253  C  NEW LOOP CONSTANTS JAN 28 1986
1254  C
1255  C
1256  DATA AT1/9*5.1,6*13.5,3*1/
1257  DATA AT1E/9*6.1,6*16.6,2*1.2/
1258  DATA AT2A/9*407.4,9*662.4,9*3.149/
1259  DATA AT2E/9*532,195,6*866.532,3*195/
1260  DATA K6/3.6E-5/,K4/.8E48876/,K5/.236/,DTOR/.174533/
1261  C
1262  DATA TDC/0.85122118,0.1195161,0.2561557/

************
LINES DELETED FROM BASELINE PROGRAM
903  DIMENSION AT1(16,2),AT2(16,2),TX1(3,3),TX2(3,3),TX3(3,3),TBL(3,3)00025450
904  DIMENSION TDC(3)
905  DATA AT1/9*1.55296-3,2.8186E-4,6*3.97586-3,1.55296-3, 00025460
906   3*2.8186E-4/,AT2/9*6.5967E-3,2.3725E-3, 00025470
907   3 6*1.85466-2,6.59676-3,3.2.37256-3/ 00025480
908  DATA TDC/0.85122118,0.1195161,0.2561557/

************
LINES ADDED TO DELIVERABLE PROGRAM
1296  C  NEW CODE AS OF JAN 28 1986
1297  C
1298  C
1299  C UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.
1300  C  IAZRATE=KSAT(IAZRATE+AT1A(MRNG,IMODE)*IAZDSC)
1301  C UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.
1302  C  IELRATE=KSAT(IELRATE+AT1E(MRNG,IMODE)*IELDSC)
1303  C
1304  C  AZRATE=K6*DTOR*FLOAT(IAZRATE)
1305  C  IELRATE=K5*DTOR*FLOAT(IELRATE)
1306  C
1307  C  IARATE=KSAT(IAZRATE+AT2A(MRNG,IMODE)*IAZDSC)
1308  C  IELRATE=KSAT(IELRATE+AT2E(MRNG,IMODE)*IELDSC)
1309  C
1310  C  IF(IALRATE.GT.0) THEN
1311    ALRATE=K4*K5*DTOR*FLOAT((IALRATE/32)
1312  ELSE
1313    ALRATE=K4*K5*DTOR*FLOAT((IALRATE-31)/32)
1314  END IF
1315  C
1316  C  IF(IBTRATE.GT.0) THEN
1317    BTRATE=K4*K5*DTOR*FLOAT((IBTRATE/32)
1318  ELSE
1319    BTRATE=K4*K5*DTOR*FLOAT((IBTRATE-31)/32)
1320  END IF
1321  C
1322  C******************************************************************************

************
LINES DELETED FROM BASELINE PROGRAM
942  ADSC=0.0431*IAZDSC 00025730
943  EDSC=0.0431*IELDSC 00025740
944 C UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE.
945 AZRATE=AZRATE+TSAM*AT1(MRNG,IMODE)+ADSC
946 C UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.
947 ELRATE=ELRATE+TSAM*AT1(MRNG,IMODE)+EDSC
948 C
949 C *******************************************************

**********
LINES ADDED TO DELIVERABLE PROGRAM
1332 ALRATE=(ALRATE+WGZ+SB)/CB-WGX
1333 GO TO 4

**********
LINES DELETED FROM BASELINE PROGRAM
959 ALRATE=(AZRATE+AT2(MRNG,IMODE)+ADSC+WGZ+SB)/CB-WGX
960 GO TO 4

**********
LINES ADDED TO DELIVERABLE PROGRAM
1337 BTRATE=BTRATE-WGY
1338 C
1339 C END OF JAN 28 1986 MODIFICATIONS
1340 C

**********
LINES DELETED FROM BASELINE PROGRAM
964 BTRATE=(ELRATE+AT2(MRNG,IMODE)+EDSC)-WGY
965 C

**********
LINES ADDED TO DELIVERABLE PROGRAM
1385 C WRITE(6,982) AZDISC,ELDISC,IAZDSC,IEDSC
1386 982 FORMAT(' ALR, BTR, AZR, ELR, SRR, SPR,=',6F18.9)
1387 981 FORMAT(' TBL 2X2 =',4F18.4)
1388 982 FORMAT(' AZD, ELD, AD, ED =',4F18.4)
1389 RETURN
1390 END
1391 C
1392 C *******************************************************
1393 C * INTEGER FUNCTION KSAT JAN 28 1986
1394 C *******************************************************
1395 C
1396 C THIS FUNCTION CHECKS ATRACK LOOP FOR SATURATION
1397 C
1398 INTEGER FUNCTION KSAT(K)
1399 C
1400 IF(K.GE.8) THEN
1401 KSAT=JMIN(K,2**15)
1402 ELSE
1403 KSAT=JMAX(K,-2**15)
1404 END IF
1405 RETURN
1406 END
1407 C

**********
LINES DELETED FROM BASELINE PROGRAM
1010 C WRITE(6,982) AZDISC,ELDISC,ADSC,EDSC
1011 982 FORMAT(' ALR, BTR, AZR, ELR, SRR, SPR,=',6F18.2)
1012 981 FORMAT(' TBL 2X2 =',4F18.4)
1013 982 FORMAT(' AZD, ELD, AD, ED =',4F18.4)
1014 RETURN
1015 END
1016 C

**********
LINES ADDED TO DELIVERABLE PROGRAM

IMPLEMENTATION OF HYSTERESIS FOR THE SAMPLING RATE CHANGE AND FOR THE PRF CHANGE ALONG WITH CHANGES IN RI (RANGE INTERVAL) WAS COMPLETED FEB 6, 1986 BY M. MEYER

LINES DELETED FROM BASELINE PROGRAM

LINES ADDED TO DELIVERABLE PROGRAM

LINES DELETED FROM BASELINE PROGRAM

LINES ADDED TO DELIVERABLE PROGRAM

C...... MODIFIED FEB 17, 1986 BY M. MEYER

C...... GUARANTEES THE CORRECT LOOP BANDWIDTHS

C...... FOR THE HYSTERESIS LOOP

C...... MODIFIED FEB 17, 1986 BY M. MEYER

C...... GUARANTEES THE CORRECT CONSTANTS
LINES DELETED FROM BASELINE PROGRAM
1339 IF(MRNG.GT.9) GO TO 86
1340 MPRF=1
1341 GO TO 90
1342 86 MPRF=2
1343 90 CONTINUE

LINES ADDED TO DELIVERABLE PROGRAM
1895 DIMENSION QNV(2)
1896

DATA NFREQ/1,5/,BN/9772.4,616.6/
DATA PS/9=4.,2.,5*4.,2.,4.,8.,8.,16./
DATA PDIA.PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/
DATA ONV/.8e67,.e11/
DATA TDC/e.05122118,e.1695161,0.2561557/

LINES DELETED FROM BASELINE PROGRAM
1463 DATA NFREQ/1,5/,BN/9772.4,616.6/,PS/9=1.,2.,5*1.,2.,4.,8.,8.,16./
1464 2 ,POIA,PDIR,PDIV/1.4142,3.1623,2.0,4.4721,2.8284,6.3246/
1465 3 PT/42658.,3125.,195.3/,QNV/.04166666/ 00022950
1466 DATA TDC/0.05122118,0.1195161,0.2561557/

LINES ADDED TO DELIVERABLE PROGRAM
1942 C WRITE(6,221)YY,SIGBAR
1943 221 FORMAT('YY,SIGBAR -',2F14.5)
1944 SNRDTD-1e..ALOG10(SNRDT) 00023240

LINES DELETED FROM BASELINE PROGRAM
1504 C WRITE(6,221)YY,SIGBAR
1505 C 221 FORMAT('YY,SIGBAR -',F14.5)
1506 SNRDTD-1e..ALOG10(SNRDT) 00023296

LINES ADDED TO DELIVERABLE PROGRAM
2603 COMMON /ICNTL/IDUM2(14),MRNG,MSAM,IDUM6(11)
2604 COMMON /OUTPUT/IDUM7(3),DUM5(6),SRSS,IDUM(4)
2605 COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
2606 COMMON /SNRSTD/AM1(21)
2607 DATA PS/9=4.,2.,5*4.,2.,4.,8.,8.,16./
DATA QNV/.00067,.011/.A1/.0321,.51/
C*************************************************************
C SUBROUTINE RSS HAS BEEN UPDATED TO CORRESPOND TO THE
C DERIVATION OF AGCERR PRESENTED IN THE FINAL REPORT ON
C KUBAND COMPUTER SIMULATION. M. MEYER FEB 17, 1986
C*************************************************************
C*************************************************************

LINES DELETED FROM BASELINE PROGRAM
2165 COMMON /ICNTL/IDUM2(14),MRNG,IDUM6(12)
2166 COMMON /OUTPUT/IDUM7(3),DM3(6),SRSS,DM4(4)
2167 COMMON /AGCDAT/AGCO,AGCODB,SNRDT,SNRDTD
2168 DIMENSION PS(10,2)
2169 DATA PS/9.1.,2.,5_1.,2.,4.,8.,8.,16./,QNV/8.04166666/
C*************************************************************

LINES ADDED TO DELIVERABLE PROGRAM
2169 C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
2170 C UPDATED FEB 17, 1986
2171 AGCERR=AI(MSAM)*4..PS(MRNG,IMODE)/(AGCO,(SNRDT+10)+QNV(MSAM))
2172 IF(AGCERR.GT.18.) AGCERR=18.8

C*************************************************************

LINES DELETED FROM BASELINE PROGRAM
2175 C STEP 1-1: COMPUTE AGC ERROR AND CHECK LIMITS.
2176 AGCERR=4..PS(MRNG,IMODE)/(AGCO+(SNRDT+10)+QNV)
2177 IF(AGCERR.GT.18.) AGCERR=18.8

C*************************************************************

LINES ADDED TO DELIVERABLE PROGRAM
2182 IF(AGCO.GT.8.25) AGCO=8.25
2183 AGCODB=10.*ALOG18(AGCO)

C*************************************************************

LINES DELETED FROM BASELINE PROGRAM
2189 SRSS=1./AGCO
2190 SRSS=10.*ALOG10(SRSS)-6.0
2191 RETURN

C*************************************************************

LINES ADDED TO DELIVERABLE PROGRAM
2194 DIMENSION PS(10,2)
2195 C PS VALUES WERE UPDATED FEB 17, 1986 BY M. MEYER
2196 DATA PS/9.1.,2.,5_1.,2.,4.,8.,8.,16./,QNV/8.04166666/
2197 SNF=1
2198 X=AGCO=(SNRDT/4.*PS(MRNG,IMODE)+1.0)
2199 X=12.25/X WAS REPLACED BY X=6.25/X TO MORE ACCURATELY
2200 C REFLECT A/D SATURATION BY M. MEYER FEB 17, 1986
C*************************************************************

C-24
2276 IF(X.GT.1) RETURN 00035720

LINES DELETED FROM BASELINE PROGRAM
2267 DIMENSION PS(10,2) 00035670
2268 DATA PS/9*10.2,5*1.2,4..8..8..16,/ 00035680
2269 SNF=1. 00035690
2270 X=ACOS(SNRDT/(4.*PS(MRNG,1MODE))+1.0) 00035700
2271 X=12.25/X 00035710
2272 IF(X.GT.1) RETURN 00035720

LINES ADDED TO DELIVERABLE PROGRAM
3108 COMMON /SUDIPH/ X,Y,Z,PAZ,PEL
3109 COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF2,CDF4, 00020230

LINES DELETED FROM BASELINE PROGRAM
2645 COMPLEX CSUM,CDIFAZ,CDIFEL,CEARLY,CLATE,CDF1,CDF2,CDF4, 00020230

LINES ADDED TO DELIVERABLE PROGRAM
3180 COMPLEX DAZ,DEL
3181 DATA ILOOP/1/

C MODIFIED JAN 10 1986 BY M. MEYER
C MODIFICATIONS TO SUBROUTINE SIGNAL INCLUDE
C CALCULATION OF THE AZIMUTH AND ELEVATION ANGLES
C USE OF MEASURED ANTENNA PATTERNS INSTEAD
C OF FUNCTIONS SPAT AND DPAT AND A
C FACTOR IN THE DIFFERENCE CHANNELS SIGNAL
C WHICH ACCOUNTS FOR THE FINITE WIDTH PHASE
C TRANSITION IN THE REAL PHASE PATTERNS.

STEP 0: READ IN ANTENNA PATTERN TERMS AND SET PHASE BALANCE

IF (ILOOP.NE.1) GO TO 11
CALL READPAT
PBAL=0.
ILOOP=0
CONTINUE

LINES DELETED FROM BASELINE PROGRAM
2653 C 00020320

LINES ADDED TO DELIVERABLE PROGRAM
3176 C STEP 2-1: COMPUTE AZIMUTH AND ELEVATION ANGLE.
3177 AZ=ATAN2D(RAU(2,K),ABS(RAU(3,K))) 00020770
3178 EL=ATAN2D(RAU(1,K),ABS(RAU(3,K))) 00020780
3179 C STEP 2-2: COMPUTE ANTENNA SUM, DIFFERENCE AND PHASE FACTORS
3180 CALL INTERP(AZ,EL) 00020790
3181 C 00020800

LINES DELETED FROM BASELINE PROGRAM
2696 C STEP 2-1: COMPUTE SUM PATTERN ANGLE.
2697 PSI=ACOS(ABS(RAU(3,K))) 00020750
2698 C 00020760
2699 C STEP 2-2: COMPUTE ANTENNA SUM PATTERN MULTIPLICATION FACTOR. 00020780
**LINES ADDED TO DELIVERABLE PROGRAM**

**LINES DELETED FROM BASELINE PROGRAM**

**STEP 3-1:** Compute AZ and EL difference pattern angles.

```plaintext
2717 C STEP 3-1: COMPUTE AZ AND EL DIFFERENCE PATTERN ANGLES.
2718 DELAZ=ASIN(RAU(2,K))
2719 DELEL=ASIN(RAU(1,K))
```

**STEP 3-2:** Compute AZ and EL difference pattern multiplication factors.

```plaintext
2723 Y=DPAT(DELAZ)
2724 Z=DPAT(DELEL)
```

**LINES ADDED TO DELIVERABLE PROGRAM**

**AND PHASE DIFFERENCE AND BALANCE WEIGHTINGS**

```plaintext
3261 DAZ=XX*Y*CMPLX(COSD(PAZ+PBAL),SIND(PAZ+PBAL))
3262 DEL=XX*Z*CMPLX(COSD(PEL+PBAL),SIND(PEL+PBAL))
```

**LINES DELETED FROM BASELINE PROGRAM**

```plaintext
2728 DAZ=XX*Y
2729 DEL=XX*Z
```

**STEP 5-3:** Compute Doppler filter weighting for each of five Doppler.

```plaintext
3357 C 3358 C NOTE: DEBUGGING PRINT STATEMENTS
```

**LINES DELETED FROM BASELINE PROGRAM**

```plaintext
2884 C 2885 C NOTE: DEBUGGING PRINT STATEMENTS
```

**LINES ADDED TO DELIVERABLE PROGRAM**

```plaintext
3362 C WRITE(6,981) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1), .
3363 C 2 DFWTS(5,1)
3364 982 FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 = ',15,6F10.2,15)
```
LINES DELETED FROM BASELINE PROGRAM
2889 C WRITE(6,901) DFWTS(1,K),DFWTS(2,K),DFWTS(3,1),DFWTS(4,1), 00022650
2890 C 2 DFWTS(5,1) 00022660
2891 902 FORMAT(' NT,S,DAZ,DEL,RGE,RGL,RGWGT,F3 =',15,6F10.2,15) 00022670

LINES ADDED TO DELIVERABLE PROGRAM
4035 C ======RI DATA STATEMENT UPDATED FEB 6, 1986 BY M. MEYER ========= 00015350
4036 DATA RI/120., .640., .1520., .2560., .5760., 11520., 23040., 43520., 00015360
4037 2 49920., 1.8228E+6/, NRI/1B/, PI/3.141592653/ 00015370
4038 C =roses=

LINES DELETED FROM BASELINE PROGRAM
3562 DATA RI/120., 240., 780., 2552., 5772., 11544., 23089., 43747., 00015350
3563 2 5772., 1.8228E+6/, NRI/1B/, PI/3.141592653/ 00015360
3564 C =============

LINES ADDED TO DELIVERABLE PROGRAM
4101 C STEP 1-3: COMPUTE INITIAL INNER AND OUTER GIMBAL RATES. 00016020
4182 C COMPUTE INITIAL OUTER GIMBAL RATE(ALRATE). 00016030

LINES DELETED FROM BASELINE PROGRAM
3627 C STEP 1-3: COMPUTE INITIAL INNER AND OUTER GIMBAL RATES. 00016020
3628 C COMPUTE INITIAL OUTER GIMBAL RATE(ALRATE). 00016030

LINES ADDED TO DELIVERABLE PROGRAM
4413 C 4414 C 4415 C 4416 C SUBROUTINE VELPRO WAS MODIFIED FEB 6 1986 BY M. MEYER 4417 C MODIFICATIONS CONSISTED OF CHECKING THE VARIABLE MPRF 4418 C FOR A VALUE OF ONE (IMPLIES 7 KC MODE) AND IF TRUE 4419 C ASSUMING THE VELOCITY ESTIMATE GIVEN BY THE VELOCITY 4420 C DISCRIMINANT IS UNAMBIGUOUS. 4421 C 4422 C 00027290

LINES DELETED FROM BASELINE PROGRAM 3939 C 00027290

LINES ADDED TO DELIVERABLE PROGRAM
4442 C 4443 C CHANGED JAN 30 1986 BY H. MAGNUSSON 4444 C 4445 C IF(IV1.GT.128)IV1=128 4446 C IFRAC=IPROM(IV1) 00027490

LINES DELETED FROM BASELINE PROGRAM 3959 C 00027490

LINES ADDED TO DELIVERABLE PROGRAM
4453 C 4454 C CHANGED FEB 6 1986 BY M. MEYER 4455 C 4456 C 4457 C IF(MPRF.EQ.1) THEN 4458 C IF(INTEG.GE.0.AND.INTEG.LE.21) THEN 4459 C IRVEL=0. 4460 C ELSE

C-27
subroutine readPAT

Read in the sum, phase, and difference patterns

real allinear( 41,41 ), elinear( 41,41 )
real salinear( 41,41 ), sellinear( 41,41 )
real pallinear( 41,41 ), pellinear( 41,41 )
common / linear / allinear, elinear
common / linear1 / salinear, sellinear
common / linear2 / pallinear, pellinear

open( unit=3, file='[KUBAND.HOWARD.MARK]azld.dat',
      access='sequential', form='unformatted',
      status='old', readonly )
read( 3 ) ( ( allinear( i,j ), j = 1,41 ), i = 1,41 )
close( 3 )
open( unit=3, file='[KUBAND.HOWARD.MARK]elld.dat',
      access='sequential', form='unformatted',
      status='old', readonly )
Subroutine: Antenna pattern interpolation.

Input: Azimuth and elevation angles in degrees.

Output: Interpolated difference, sum, and phase values for all 18 antenna patterns.

---

subroutine interp( az, el)

    ! Linearly interpolate the gain, phase and difference patterns

    real allinear( 41,41 ), ellinear( 41,41 )
    real sallinear(41,41), eellinear(41,41)
    real pollinear(41,41), pellinear(41,41)
    common / linear / allinear, ellinear
    common / linear1 / sallinear, eellinear
    common / linear2 / pollinear, pellinear
    common / SUDIPH / X,Y,Z,PAZ,PEL

    iax = jint( ( az + 4. ) * 5. )
    iex = jint( ( el + 4. ) * 5. )
az0 = float(j(ian) / 5. - 4.
e10 = float(j(ien) / 5. - 4.

iaz = jint ((az + 4.) * 5. + 1
je1 = jint ((el + 4.) * 5. + 1

--- find az0 values ---

f0 = 10.**aolinear(iaz, je1) / 20.
f1 = 10.**aolinear(iaz+1, je1) / 20.
f2 = 10.**aolinear(iaz, je1+1) / 20.
f3 = 10.**aolinear(iaz+1, je1+1) / 20.

fa = f0 + (f1-f0)/2 * (az-az0)
f1 = f1 + (f2-f1)/2 * (az-az0)
f2 = f2 + (f3-f2)/2 * (az-az0)
f3 = f3 + (f1-f3)/2 * (az-az0)

fx = fa + (fb-fa)/2 * (el-e10)

--- find eld values ---

--- find az0 values ---

f0 = 10.**(salinear(iaz, je1))/20.
f1 = 10.**(salinear(iaz+1, je1))/20.
f2 = 10.**(salinear(iaz, je1+1))/20.
f3 = 10.**(salinear(iaz+1, je1+1))/20.

fo = f0 + (f1-f0)/2 * (az-az0)
f1 = f1 + (f2-f1)/2 * (az-az0)
f2 = f2 + (f3-f2)/2 * (az-az0)
f3 = f3 + (f1-f3)/2 * (az-az0)

fx = fo + (fb-fa)/2 * (el-e10)

--- find azp values ---

f0 = palinear(iaz, je1)
f1 = palinear(iaz+1, je1)
f2 = palinear(iaz, je1+1)
f3 = palinear(iaz+1, je1+1)

fo = f0 + (f1-f0)/2 * (az-az0)
f1 = f1 + (f2-f1)/2 * (az-az0)
f2 = f2 + (f1-f0)/2 * (az-az0)
f3 = f3 + (f1-f3)/2 * (az-az0)

fx = fo + (fb-fa)/2 * (el-e10)

PAZ=fx ! phase in degrees

--- find elp values ---

f0 = pelinear(iaz, je1)
f1 = pelinear(iaz+1, je1)
f2 = pelinear(iaz, je1+1)

--- find azp values ---

--- find elp values ---

--- find az0 values ---
5128  \[ f3 = \text{pellinear}(i\text{az}+1,j\text{el}+1) \]
5129
5130  \[ f0 = f0 + (f1-f0)/2*(az-az0) \]
5131  \[ fb = f2 + (f3-f2)/2*(az-az0) \]
5132  \[ fx = fo + (fb-f0)/2*(el-e10) \]
5133
5134  PEL=fx  \quad \text{phase in degrees}
5135
5136  return
5137
5138  end

********
LINES DELETED FROM BASELINE PROGRAM
********

Number of difference sections found: 62
Number of difference records found: 1052

DIFFERENCES /IGNORE=/MERGED=1/OUTPUT=USER1:[KUBAND.HOWARD.MARK]FINHAC.DIF;1-
 USER1:[KUBAND.HOWARD.MARK]FINSIM1.FOR;8-
 USER1:[KUBAND.HOWARD]HACS1M.FOR;1
INTRODUCTION

This appendix presents the details of the analysis of GDOP. GDOP is the term used to describe the effects of range and range rate measurement errors from sensors at various geometries relative to the target on subsequent calculations of target position and velocity. The problem is best understood by referring to Figure D1, which shows the WSMR range geometry, with the Brass Cap location at the origin. (This is done to simplify the math which follows.)

Each of the three TMR radars measures the range from itself to the target along with line of sight velocity (Range Rate) of the target relative to each radar. These measurements of range, denoted as R1, R2, and R3, are used to compute the X, Y, and Z coordinates of the target, relative to Brass Cap. These will be denoted as X, Y, and Z. Given X, Y, and Z, the range from Brass Cap to the target, R, can be found. Note that the locations of the three radars are denoted by the coordinate sets (X1,Y1,Z1), (X2,Y2,Z2), and (X3,Y3,Z3). Using the above data, range rate (change in R with respect to time) can also be computed.

GDOP occurs when R1, R2, and R3 contain errors. The errors may be bias errors (constant) or randomly varying (stochastic). The overall effect of errors in R1, R2, and R3 is that they cause the computed values of X, Y, and Z, and thus R, to be in error. The detailed analysis of this phenomenon will be developed in the rest of this section. Examples of its effect on the WSMR experimental data will also be presented.
FIGURE D1 GEOMETRY OF TMR RADARS AT WSMR RELATIVE TO BRASS CAP
D-2

RANGE ERRORS

D-2.1 Example

As a intuitive introduction to the range error problem, we will first consider a two dimensional problem shown in Figure D2. In Figure D2, the target is approximately midway between the radars, and is nearly above the "Brass Cap" reference point. The true range from Brass Cap to target is R.

This is the range we are trying to measure with the radars. The true ranges from the radars to the target are R1 and R2. If the exact values of R1 and R2 were measured by the radars then X and Y could be found by solving the pair of equations below:

\[
\begin{align*}
R_1^2 &= (X - X1)^2 + (Y - Y1)^2 \\
R_2^2 &= (X - X2)^2 + (Y - Y2)^2
\end{align*}
\]

And R could be found by substituting X and Y into the equation:

\[
R^2 = X^2 + Y^2
\]

This solution is graphically shown in the figure as point P, which is the intersection of the two circles of radii R1 and R2.

If each of the radar range measurements was in error by an amount DR, then the apparent ranges measured would be R1+DR and R2+DR. These ranges are shown as circular arcs in Figure D2. Note that their intersection is at point Q, which has coordinates XQ and YQ. As can be seen in the figure, the range from Brass Cap to point Q is significantly different from the true range. Note that the values XQ and YQ would be obtained by substituting R1+DR and R2+DR in the set of equations above.

The situation portrayed in Figure D2 is "worst case" in the sense that small errors in R1 and R2 produce large errors in R. This is because of the geometry of the situation. Although it will not be described here in detail, the reader should have little trouble convincing himself that other
FIGURE D2  TWO DIMENSIONAL DIAGRAM SHOWING EFFECTS OF RADAR RANGE ERRORS ON ESTIMATE OF TARGET RANGE
geometries, for example where the target is far removed from the radars, produce smaller errors.

**Mathematical Analysis**

We now consider the general three dimensional case. The notation which is used below is consistent with Figure D1.

The true range from Brass Cap to the target is given by:

\[ R^2 = x^2 + y^2 + z^2 \]

The range as computed using data from the three radars is given by:

\[ RR^2 = xR^2 + yR^2 + zR^2 \]

where XR, YR, and ZR are computed from the radar range data using the set of equations below:

\[ R_1^2 = (XR - X_1^2) + (YR - Y_1)^2 + (ZR - Z_1)^2 \]
\[ R_2^2 = (XR - X_2^2) + (YR - Y_2)^2 + (ZR - Z_2)^2 \]
\[ R_3^2 = (XR - X_3^2) + (YR - Y_3)^2 + (ZR - Z_3)^2 \]

In general XR, YR, and ZR will contain errors, because of errors in R1, R2, and R3. To analyze the effects of errors in R1, R2, and R3 on XR, YR, and ZR we first take the total derivative of the expressions for R1, R2, and R3 to get the system of equations shown here. Note that the derivatives have been represented by DX, DY, and DZ.

\[ R_{1DR} = (XR - X_1)DX + (YR - Y_1)DY + (ZR - Z_1)DZ \]
\[ R_{2DR} = (XR - X_2)DX + (YR - Y_2)DY + (ZR - Z_2)DZ \]
\[ R_{3DR} = (XR - X_3)DX + (YR - Y_3)DY + (ZR - Z_3)DZ \]

In this system of equations, R1, R2, and R3 are measured by the radars. XR, YR, and ZR are computed as above, X1, Y1, Z1, etc. are known from the range
survey data, and DR is assumed to be known from range calibration data. DR may be deterministic, as in a fixed range bias, or may vary statistically. A general model for DR is:

\[ DR = DR + U \]

where U is a random variable with zero mean and some specified variance and probability density function. For the majority of data here, DR will be assumed to be a constant. Existing data from WSMR indicates that DR is approximately 10 feet.

By solving this system of equations, DX, DY, and DZ may be found as a function of the range errors associated with the three radars. The DX, DY, and DZ values may subsequently be used to correct the values XR, YR, and ZR, and thus improve the range estimate RR.

D-2.3 Experimental Data

Figures D3 and D4 are plots of range error computed from two sets of WSMR experimental data. Figure D3 is the range error observed from tracking a target which was close to the Brass Cap location, and at a relatively low altitude. Note that the range errors are large, on the order of 150 feet. Figure D4 is the range error computed for a target which was at a higher altitude and considerably longer range. Note that in this instance, the range errors are approximately 20 feet.

These results are consistent with the example presented at the beginning of this section. They were computed assuming the radar range errors were the same for all radars, and were equal to 10 feet.

D-3 RANGE RATE (VELOCITY) ERRORS

D-3.1 Example

For the example below, refer to Figure D1. Given a target close to the Brass Cap at a very low altitude the range rate measurement of the 3
FIGURE D3  RANGE ERRORS DUE TO GDOP
FOR TARGET NEAR BRASS CAP
SIM DATA PROFILE HJ146AD
TEST DATE 10-1-85

FIGURE D4 RANGE ERRORS DUE TO GDOP
FOR TARGET AWAY FROM BRASS CAP

D-8
TMR radars would not be affected significantly by the Vz component of the target. Inversely a small uncertainty in range rate translates into a large uncertainty in the TMR predicted Vz component. With this scenario, the target is practically above the Ku-Band Radar and the actual Vz component affects the Ku-Band range rate measurement significantly. In this case one would expect the GDOP effect to be large. A target whose location was not close to the Brass Cap and had a large altitude one would expect the GDOP effect to be small. Examples of real range data which support this example will be presented in Section D-3.3.

D-3.2 Mathematical Analysis

The range rate of the target relative to a radar can be determined by taking the time derivative of the range equation which is repeated below for reference

\begin{equation}
R_l^2 = (X_R - X_l)^2 + (Y_R - Y_l)^2 + (Z_R - Z_l)^2
\end{equation}

When this is done, we obtain equations of the form shown below. The equation shown is for radar R1.

\begin{equation}
\frac{R_l}{dt} \frac{DR}{dt} = (X_R - X_l) \frac{DX}{dt} + (Y_R - Y_l) \frac{dY}{dt} + (Z_R - Z_l) \frac{dZ}{dt}
\end{equation}

The sensitivity of X, Y, and Z to small errors in the TMR range rate measurements can be determined by taking the total derivative of the range rate equations and simultaneously solving the set of equations which result. The form of the range rate equation is shown below. The equation shown is for radar R1. The dot (.) superscript denotes derivative.

\begin{equation}
\dot{R_1} + \ddot{R_1} = (X_R - X_l) \dot{DX} + (Y_R - Y_l) \dot{DY} + (Z_R - Z_l) \dot{DZ}
\end{equation}

Regrouping the terms of equation 10, we obtain a more convenient form:

\begin{equation}
\dot{R_1} + \ddot{R_1} = (X_R - X_l) DX + (Y_R - Y_l) DY + (Z_R - Z_l) DZ
\end{equation}
For compactness, we will adopt matrix notation to write the complete set of equations for the three radars. Rewrite equation 11 as:

(12) \quad G \times K = H \times \text{DEL}

Where G is given by:

\[
G = \begin{bmatrix}
R_1 & R_1 & X_R & Y_R & Z_R \\
R_2 & R_2 & X_R & Y_R & Z_R \\
R_3 & R_3 & X_R & Y_R & Z_R
\end{bmatrix}
\]

(13)

and K is:

\[
K = \begin{bmatrix}
D_R \\
D_R \\
-D_X \\
-D_Y \\
-D_Z
\end{bmatrix}
\]

(14)

H is the matrix:

\[
H = \begin{bmatrix}
(X_R - X_1) & (Y_R - Y_1) & (Z_R - Z_1) \\
(X_R - X_2) & (Y_R - Y_2) & (Z_R - Z_2) \\
(X_R - X_3) & (Y_R - Y_3) & (Z_R - Z_3)
\end{bmatrix}
\]

(15)

and DEL is the vector:

\[
\text{DEL} = \begin{bmatrix}
D_X \\
D_Y \\
D_Z
\end{bmatrix}
\]

(16)
DX, DY, and DZ are determined from the equations derived in the Range Error Section above. DR and its derivative DDr are the range error and range rate error associated with the radars. They are assumed to be the same for all three radars, and known from independent measurements. The range error (DR) is assumed to be constant, while the range rate error (DDr) is assumed to be stochastic with zero mean and a standard deviation of 0.2 ft/sec.

The quantities of interest in the above equations are the range rate errors DX, DY, and DZ. Normal values of XR, YR, ZR, XR, YR, and ZR are available as data from the TMR radar solution.

To calculate the variance of DX, DY, and DZ we form the covariance matrix of DEL as shown in the equations below. The diagonal elements of the matrix P are the variances of DX, DY, and DZ respectively.

\[
(17a) \quad P = \text{VAR} [\text{DEL}]
\]

\[
(17b) \quad P = H^{-1} G E[KK^T] G^T (H^{-1})^T - E[\text{DEL}]E[\text{DEL}^T]
\]

where

\[
E[K] = \begin{bmatrix} DR \\ 0.0 \\ -DX \\ -DY \\ -DZ \end{bmatrix}
\]

and

D-11
(19) \[
\begin{bmatrix}
\mathbf{E}[\mathbf{K}^T] = \\
\begin{bmatrix}
\begin{array}{cccc}
\mathbf{D}R^2 & \mathbf{DR} & -\mathbf{DX} & -\mathbf{DY} & -\mathbf{DZ} \\
\mathbf{DR} & \mathbf{D}R^2 & -\mathbf{DX} & -\mathbf{DY} & -\mathbf{DZ} \\
-\mathbf{DX} & -\mathbf{DX} & \mathbf{D}X^2 & \mathbf{DX} & \mathbf{DZ} \\
-\mathbf{DY} & -\mathbf{DY} & \mathbf{DX} & \mathbf{DY}^2 & \mathbf{DY} \\
-\mathbf{DZ} & -\mathbf{DZ} & \mathbf{DX} & \mathbf{DY} & \mathbf{DZ}^2
\end{array}
\end{bmatrix}
\end{bmatrix}
\]

Since \( \mathbf{D}R \) is 0 mean with .2 ft/sec standard deviation, the \( \mathbf{E}[\mathbf{K}^T] \) reduces to:

\[
\begin{bmatrix}
\mathbf{E}[\mathbf{K}^T] = \\
\begin{bmatrix}
\begin{array}{cccc}
\mathbf{D}R^2 & 0 & -\mathbf{DX} & -\mathbf{DY} & -\mathbf{DZ} \\
0 & 0.04 & 0 & 0 & 0 \\
-\mathbf{DX} & 0 & \mathbf{D}X^2 & \mathbf{DX} & \mathbf{DZ} \\
-\mathbf{DY} & 0 & \mathbf{DX} & \mathbf{DY}^2 & \mathbf{DY} \\
-\mathbf{DZ} & 0 & \mathbf{DX} & \mathbf{DY} & \mathbf{DZ}^2
\end{array}
\end{bmatrix}
\end{bmatrix}
\]

The effect of the TMR range rate measurement errors on the predicted range rate at the Ku-Band Radar site is approximated by taking the dot product of the Brass Cap range unit vector with the velocity error vector. This approximation is valid because the Brass Cap and the Ku-Band radar were separated by only a few feet and the coordinate transformations involved would not affect the results significantly.

D-3.3 Examples

Figures D5 through D12 show two cases where velocity errors were computed from WSMR experimental data. Figures D5 through D8 demonstrate that for a target at low altitude and close to the Brass Cap the GDOP effect is significant. Figures D5, D6, and D7 show the X, Y, and Z range from the Brass Cap. Figure D8 shows the range rate errors which were computed from the data, using the procedures above. Figures D9 through D12 show that at high altitudes the GDOP effect is minimal. Figures D9 through D11 show the X, Y and Z coordinates which were measured, and Figure D12 the range rate error. Note that the errors in the longer range case, shown in Figure D12 are less than those shown in D8.
TEST DATA PROFILE HJ146AD
TO-62415, GMT-17.20.14.

FIGURE D5 X COORDINATES OF TARGET

D-13
TEST DATA PROFILE HJ146AD
TO-62415. GMT-17.20.14.

FIGURE D6 Y COORDINATES OF TARGET

D-14
FIGURE D7 Z COORDINATES OF TARGET
SIM DATA PROFILE MJ146AD
TEST DATE 10-5-85

TIME (SECONDS)

MEAN = 0.142

FIGURE D8  RANGE RATE STANDARD DEVIATION DUE TO GDOP
TEST DATA PROFILE H30SKAG

TO-60657, GMT-16.50.57.

FIGURE D9 X COORDINATES OF TARGET

D-17
TEST DATA PROFILE H30SKAG

TO-60657, GMT-16.50.57.

FIGURE D 10 Y COORDINATES OF TARGET

D-18
FIGURE D11  Z COORDINATES OF TARGET
FIGURE D12  RANGE RATE STANDARD DEVIATION DUE TO GDOP
Figure D13 is a listing of the program which was used to perform the GDOP analysis. Its inputs are the same WSMR data files used by the other analysis programs, and similar output plots are available.
SUBROUTINE GDOP(RDOTSD)

COMMON/TMR/X,Y,Z,XDOT,YDOT,ZDOT
REAL H(3,3),G(3,5),DK(5,1),DKT(1,5),GT(5,3)
REAL HINV(3,3),HINVT(3,3),M(3,1),MT(1,3),R(3,1)
REAL XYZMAT(3,3),DELT(3,1),HINVG(3,5),DKDKT(5,5)
REAL TEMP(3,5),GTINVT(5,3),MMT(3,3),MDKT(3,5)
REAL DKMT(5,3),COV1(3,3),COV2(3,3),COV3(3,3),COV(3,3)
DATA ILOOP=1/

READ X Y Z POSITIONS OF VECTORS
IF (ILOOP.EQ.1) THEN
OPEN(UNIT=8,FILE='POS.DAT',STATUS='OLD')
DO I=1,3
  READ(8,*), XYZMAT(I,1),XYZMAT(I,2),XYZMAT(I,3)
END DO
READ IN DELTA RANGE, DELTA RANGE RATE MEAN AND
DELTA RANGE RATE VARIANCE
READ(8,*), DR,DRDOT,DRDOTTQ
ILOOP=0
END IF
DO I=1,3
  H(I,1)=X-XYZMAT(I,1)
  H(I,2)=Y-XYZMAT(I,2)
  H(I,3)=Z-XYZMAT(I,3)
  G(I,3)=XDOT
  G(I,4)=YDOT
  G(I,5)=ZDOT
  R(I,1)=SQRTH(1,1)**2+H(I,2)**2+H(I,3)**2)
  G(I,1)=(H(I,1)*XDOT+H(I,2)*YDOT+H(I,3)*ZDOT)/R(I,1)
  G(I,2)=R(I,1)
END DO
DO MATRIX TRANSPOSES INVERSES AND MULTIPICATIONS
CALL MATINV(H,HINV)
CALL MATMULT(HINV,R,DELT,3,3,1)

SOLVE FOR DELTA X DELTA Y AND DELTA Z

FIGURE D13 SOURCE LISTING OF GDOP ANALYSIS PROGRAM
PAGE 1

D-22
DO I=1,3
  DELT(I,1)=DELT(I,1)*DR
END DO

C

SET UP MATRIX DELTA K

C

DK(1,1)=DR
DK(2,1)=DRDOT
DK(3,1)=DELT(1,1)
DK(4,1)=DELT(2,1)
DK(5,1)=DELT(3,1)
CALL MATMULT(HINV,G,HINVG,3,3)
CALL MATMULT(HINV,DK,M,3,5,1)
CALL MATTRAN(M,MT,3,1)
CALL MATTRAN(G,GT,3,5)
CALL MATTRAN(DK,DKT,5,1)
CALL MATTRAN(HINV,HINVT,3,3)
CALL MATMULT(DK,DKT,5,1,5)
C

SET DKDKT(2,2) TO VARIANCE OF VELOCITY ERROR

C

DKDKT(2,2)=DRDOTSQ
CALL MATMULT(GT,HINVT,GTHINVT,5,3)
CALL MATMULT(M,DKT,MDKT,3,1,5)
CALL MATMULT(DK,MT,DKMT,5,1,3)
CALL MATMULT(HINVG,DKDKT,TEMP,3,5,5)
CALL MATMULT(TEMP,GTHINVT,COVI,3,5,3)
CALL MATMULT(MDKT,GTHINVT,COV2,3,5,3)
CALL MATMULT(HINVG,DKMT,COV3,3,5,3)
CALL MATMULT(M,MT,MMT,3,1,3)
C

FORM COVARIANCE MATRIX

C

DO I=1,3
  DO J=1,3
    COV(I,J)=COV1(I,J)-COV2(I,J)-COV3(I,J)+MMT(I,J)
  END DO
END DO

XDSD=SQRT(COV(1,1))
YDSD=SQRT(COV(2,2))
ZDSD=SQRT(COV(3,3))
RANGE=SQR((X+Y**2+Z**2)/RANGE)
RDOTSD=abs((X*XDSD+Y*YDSD+Z*ZDSD)/RANGE)
RETURN

END SUBROUTINE MATTRAN

DIMENSION A(M,N),B(N,IP),C(M,IP)
C

DO I=1,M
  DO J=1,IP
    C(I,J)=0.0
  DO K=1,N
    C(I,J)=C(I,J)+A(I,K)*B(K,J)
  END DO
END DO
END SUBROUTINE MATTRAN

DIMENSION A(IROW,ICOL),B(1COL,IROW)

FIGURE D13 SOURCE LISTING OF GDOP ANALYSIS PROGRAM

PAGE 2

D-23
C
DO I=1,IROW
   DO J=1,ICOL
      B(J,I)=A(1,J)
   END DO
END DO
RETURN
END

SUBROUTINE MATINV(A,B)
DIMENSION A(3,3),B(3,3)

C
DET=A(1,1)*A(2,2)*A(3,3)+A(1,2)*A(2,3)*A(3,1)
+ A(1,3)*A(2,1)*A(3,2)-A(1,1)*A(2,2)*A(3,3)
- A(1,2)*A(2,1)*A(3,3)-A(1,1)*A(2,3)*A(3,2)

B(1,1)=A(2,2)*A(3,3)-A(2,3)*A(3,2)
B(2,1)=A(3,1)*A(2,3)-A(2,1)*A(3,3)
B(3,1)=A(2,1)*A(3,2)-A(3,1)*A(2,2)
B(1,2)=A(1,3)*A(3,2)-A(1,2)*A(3,3)
B(2,2)=A(1,1)*A(3,3)-A(1,3)*A(3,1)
B(3,2)=A(1,2)*A(3,1)-A(1,1)*A(3,2)
B(1,3)=A(2,1)*A(3,2)-A(2,2)*A(3,1)
B(2,3)=A(2,3)*A(3,1)-A(2,1)*A(3,3)
B(3,3)=A(1,1)*A(2,2)-A(1,2)*A(2,1)

DO I=1,3
   DO J=1,3
      B(J,I)=B(J,I)/DET
   END DO
END DO
RETURN
END

FIGURE D13  SOURCE LISTING OF GDOP ANALYSIS PROGRAM
PAGE 3
D-24
APPENDIX E: EFFECTS OF COORDINATE MISALIGNMENT ON DELTA ROLL ANGLES

If we start with two coordinate systems that have the same origins but are not aligned a point in space will have two sets of coordinates; \((X,Y,Z)\) and \((X',Y',Z')\), as shown in Figure E-1. In our particular case, we let the \((X,Y,Z)\) system represent where the TMR2KU subroutine says the radar is pointing and the \((X',Y',Z')\) system represent where the shuttle radar actually is pointing.

It is possible to go from the \((X,Y,Z)\) system to the \((X',Y',Z')\) system using the rotation matrices:

\[
\begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos A & \sin A \\
0 & -\sin A & \cos A
\end{bmatrix} \begin{bmatrix}
\cos B & 0 & \sin B \\
0 & 1 & 0 \\
-\sin B & 0 & \cos B
\end{bmatrix} \begin{bmatrix}
\cos C \sin C & 0 \\
-\sin C \cos C & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

**Figure E-1** TWO COORDINATE SYSTEMS HAVING SAME ORIGIN, BUT UNALIGNED AXES
where A, B, and C are the rotation angles about the coordinate axes. Multiplying through, we obtain

\[
\begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix} =
\begin{bmatrix}
\cos B \cos C & \cos B \sin C & \sin B \\
- \sin C \cos A - \sin B \cos C \sin A & \cos A \cos C - \sin A \sin B \sin C & \cos A \sin B \\
\sin A \cos B & \sin A \sin C - \cos A \sin B \cos C & \sin A \cos C
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

If we use the small angle approximations of
\[
\sin u = u \\
\cos u = 1
\]

we get

\[
\begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix} =
\begin{bmatrix}
1 & C & B \\
- C - AB & 1 - ABC & A \\
AC - B & - A - BC & 1
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

or

\[
\begin{align*}
X' &= X + YC + ZB \\
Y' &= X(- C - AB) + Y(1 - ABC) + ZA \\
Z' &= X(AC - B) + Y(- A - BC) + Z
\end{align*}
\]

if we let

\[
\begin{align*}
X' &= X + \Delta X \\
Y' &= Y + \Delta Y \\
Z' &= Z + \Delta Z
\end{align*}
\]

we find

\[
\begin{align*}
\Delta X &= YC + ZB \\
\Delta Y &= ZA - XC - XAB - YABC \\
\Delta Z &= - XB - YA + XAC - YBC
\end{align*}
\]
where these deltas are the errors caused by the rotation angles.

We can now see how the rotation angles would affect delta roll.

By definition,

\[
\text{Roll} = \arctan(Y/Z)
\]

\[
\text{Roll}' = \arctan(Y'/Z')
\]

\[
\text{Roll} = \text{Roll}' - \text{Roll}
\]

If we use small roll angle data, we can approximate

\[
\arctan(Y/Z) = \frac{Y}{Z}
\]

This makes

\[
\text{Roll} = \frac{Y' - Y}{Z'} - \frac{Z}{Z'}
\]

\[
= \frac{Y + \Delta Y - Y}{Z + \Delta Z} - \frac{Z \Delta Y - Y \Delta Z}{Z \left(1 + \frac{\Delta Z}{Z}\right)}
\]

using the approximation

\[
\frac{1}{1 + x} = 1 - x \quad \text{for small } x
\]

\[
\Delta \text{Roll} = \frac{Z \Delta Y - Y \Delta Z - Y \Delta Z + \Delta Z^2 Y}{Z^2}
\]

The last term in the numerator, \(Z^2 Y_1\), is negligible for the trajectories used. This leaves

\[
\Delta \text{Roll} = \frac{Z \Delta Y - Y \Delta Z}{Z^2}
\]
If we now substitute in delta Y and Z from before

$$\Delta \text{Roll} = -XZAB - YZA^2 + XZA^2C - YZAB$$
$$+ X^2BC + YZAC - Z^2AC^2 + XYBC^2$$
$$+ X^2AB^2 + XYA^2B - X^2A^2BC + XYAB^2C$$

If we keep second order and above terms

$$\Delta \text{Roll} = A(I + Y^2/Z^2) + B(XY/Z^2) + C(-X/Y)$$
$$+ A^2(Y/Z) - ACXY/Z^2 + BC((Y^2 - X^2/Z^2)$$

for the available trajectories, the last two terms become negligible and we have

$$\Delta \text{Roll} = A(I + Y^2/Z^2) + B(XY/Z^2) + C(-X/Y) + A^2(Y/Z)$$
APPENDIX F - TARGET ACCELERATION EFFECTS

F-1  INTRODUCTION AND ANALYSIS

In order to predict the current velocity of a target, the Ku-Band Radar computes a velocity discriminant. This computation is made assuming that there is no target acceleration. The presence of target acceleration has an adverse effect on the ability of the radar to measure velocity because bias errors are introduced into the velocity discriminant calculation.

In order to form the velocity discriminant, the radar performs the following steps:

1) Transmits 16 pulses per time slot, 4 time slots per frequency, over 5 frequencies with a null time equal to 1/PRF between each time slot.

2) For each time slot the radar performs a DFT from the early range gate and a DFT from the late range gate.

3) Sums up the magnitudes of the "low" filter bin for all range gates, time slots, and frequencies. Similar processing is performed for the "high" filter bin.

4) A velocity discriminant is formed by computing log (low/high).

5. Computes fractional position within a filter by an inverse mapping of the velocity discriminant.

6. Computes velocity estimate from knowledge of the center filter number and the fractional displacement from the center.

If a target is accelerating during a time slot, the effect of the acceleration is to "slide" the target across DFT frequency bins. The contents of a bin are thus the average of the outputs from the various frequencies which the acceleration produced during a time slot. The practical effect of this phenomenon is minimal in many cases because the acceleration
values likely to be encountered, and the time slot are both small - the latter is 16 times the reciprocal of the PRF.

Averaging of the individual DFT responses prior to using the velocity discriminant function produces a smoothing effect, which damps acceleration effects. It has been observed that the combination of this averaging, and the inverse mapping which is used to form the velocity estimate are approximately linear for the accelerations which would likely be encountered. The end result is that the velocity error is given by:

\[ \text{VEL ERROR} = \frac{(\text{Final Velocity} - \text{Initial Velocity})}{2} \]

Note that the velocity error is a "bias" error, that is, it is proportional only to the velocity difference and not the values of the individual velocities. Note that if the target's acceleration was oscillating between positive and negative values this bias error could cause the velocity estimate of the radar to have an error standard deviation which exceeded the specification.

F-2 SIMULATION RESULTS

A computer simulation of the velocity estimation signal processing portion of the radar was written to validate the conclusions drawn in the above section. The radar return from a target accelerating at a constant rate was modelled by using a linear ramped FM wave. This simulated signal was processed by the DFT and velocity processor and resulted in velocity errors approximately one half the velocity change over the update period as shown in Fl.

The results shown in Figure F1 confirm that the effects of acceleration on velocity computation are linearly predictable as described in Section F1 above.
FIGURE P1 VELOCITY CHANGE OVER UPDATE INTERVAL

(Fps)

VELOCITY ERROR
APPENDIX G
WHITE SANDS MISSILE RANGE FLIGHT TEST DATA SUMMARY

This appendix provides a brief summary of all official flight
tests of the space shuttle radar at the White Sands Missile Range (WSMR). The
information in the summary was obtained from two sources: (1) the 24 hour
reports written by Andy Lindberg of Lockheed Engineering and Management
Services Company (LEMSCO), and (2) the reduced flight data provided by NASA
JSC and LEMSCO personnel.

This Appendix is structured as follows. Each subsection provides
the summaries of all test flights flown on a particular test date. The
introduction of each subsection provides the flight conditions for the day,
the targets used, and the trajectories flown. The format of each individual
flight summary is as follows.

Trajectory: Name of the profile flown.

Range Equipment: WSMR tracking equipment employed.

Flight Profile: The initial and final X, Y and Z coordinates of
the target in the Brasscap coordinate system. In this system X is
North and -Z is vertical. The arrows (→) indicate the direction
of travel.

Duration: The length of the flight in seconds.

Comments: Documentation of large trends, means or standard
deviations in the difference data or other anomalies.

In addition, Table G-1 provides a list of all flight tests by
trajectory name and the corresponding page numbers within this appendix where
a summary of the flight test can be found. Table G-2 contains the statistics
for the delta range, range rate, roll, roll rate, pitch, pitch rate and alpha
and beta angles for each test.

G-1
### TABLE G-1 CONTENTS OF THE APPENDIX

<table>
<thead>
<tr>
<th>Test Date</th>
<th>Flight Name</th>
<th>Summary Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/1/85</td>
<td>HL546AC</td>
<td>G-7</td>
</tr>
<tr>
<td></td>
<td>HL246AD</td>
<td>G-7</td>
</tr>
<tr>
<td></td>
<td>HL446AC</td>
<td>G-8</td>
</tr>
<tr>
<td></td>
<td>HL146AC</td>
<td>G-8</td>
</tr>
<tr>
<td></td>
<td>HL346AD</td>
<td>G-8</td>
</tr>
<tr>
<td></td>
<td>HJ146AC</td>
<td>G-9</td>
</tr>
<tr>
<td>10/3/85</td>
<td>HEL30AE</td>
<td>G-11</td>
</tr>
<tr>
<td></td>
<td>HEL30AF</td>
<td>G-11</td>
</tr>
<tr>
<td></td>
<td>H30SKAE</td>
<td>G-11</td>
</tr>
<tr>
<td></td>
<td>H30SKAF</td>
<td>G-12</td>
</tr>
<tr>
<td></td>
<td>HEL30AG</td>
<td>G-12</td>
</tr>
<tr>
<td></td>
<td>HEL30AH</td>
<td>G-12</td>
</tr>
<tr>
<td></td>
<td>H30SKAG</td>
<td>G-13</td>
</tr>
<tr>
<td></td>
<td>H30SKAH</td>
<td>G-13</td>
</tr>
<tr>
<td></td>
<td>H30SKAI</td>
<td>G-13</td>
</tr>
<tr>
<td></td>
<td>HEL30AI</td>
<td>G-14</td>
</tr>
<tr>
<td></td>
<td>HEL30AJ</td>
<td>G-14</td>
</tr>
<tr>
<td>10/5/85</td>
<td>HL546AE</td>
<td>G-15</td>
</tr>
<tr>
<td></td>
<td>HL346AE</td>
<td>G-15</td>
</tr>
<tr>
<td></td>
<td>HL446AD</td>
<td>G-17</td>
</tr>
<tr>
<td></td>
<td>HL146AD</td>
<td>G-17</td>
</tr>
<tr>
<td></td>
<td>HJ146AD</td>
<td>G-17</td>
</tr>
<tr>
<td></td>
<td>HL546AF</td>
<td>G-18</td>
</tr>
<tr>
<td></td>
<td>HL246AE</td>
<td>G-18</td>
</tr>
<tr>
<td>10/16/85</td>
<td>GEM1</td>
<td>G-20</td>
</tr>
<tr>
<td></td>
<td>GEM2</td>
<td>G-20</td>
</tr>
<tr>
<td></td>
<td>GEM3</td>
<td>G-20</td>
</tr>
<tr>
<td>Test Date</td>
<td>Flight Name</td>
<td>Summary Location</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td>10/19/85</td>
<td>SAT1</td>
<td>G-22</td>
</tr>
<tr>
<td></td>
<td>SAT2</td>
<td>G-22</td>
</tr>
<tr>
<td></td>
<td>SAT3</td>
<td>G-23</td>
</tr>
<tr>
<td></td>
<td>SAT4</td>
<td>G-23</td>
</tr>
<tr>
<td></td>
<td>SAT6</td>
<td>G-23</td>
</tr>
<tr>
<td></td>
<td>SAT8</td>
<td>G-23</td>
</tr>
<tr>
<td>11/4/85</td>
<td>BAL1</td>
<td>G-25</td>
</tr>
<tr>
<td></td>
<td>BAL2</td>
<td>G-25</td>
</tr>
<tr>
<td></td>
<td>BAL5</td>
<td>G-26</td>
</tr>
<tr>
<td></td>
<td>BAL6</td>
<td>G-26</td>
</tr>
<tr>
<td></td>
<td>BAL7</td>
<td>G-27</td>
</tr>
<tr>
<td></td>
<td>HL546AG</td>
<td>G-27</td>
</tr>
<tr>
<td></td>
<td>HL246AF</td>
<td>G-27</td>
</tr>
<tr>
<td></td>
<td>HL446AE</td>
<td>G-28</td>
</tr>
<tr>
<td></td>
<td>HL146AE</td>
<td>G-28</td>
</tr>
<tr>
<td></td>
<td>HL346AF</td>
<td>G-28</td>
</tr>
<tr>
<td></td>
<td>HJ146AE</td>
<td>G-29</td>
</tr>
<tr>
<td>PROFILE</td>
<td>RUN</td>
<td>NAME</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td>BAL1 THR</td>
<td>MEAN</td>
<td>19.3236</td>
</tr>
<tr>
<td>BAL2 THR</td>
<td>MEAN</td>
<td>19.3236</td>
</tr>
<tr>
<td>BAL3 THR</td>
<td>MEAN</td>
<td>24.7628</td>
</tr>
<tr>
<td>BAL4 THR</td>
<td>MEAN</td>
<td>22.3772</td>
</tr>
<tr>
<td>BAL7 THR</td>
<td>MEAN</td>
<td>22.3772</td>
</tr>
<tr>
<td>GED THR</td>
<td>MEAN</td>
<td>32.2348</td>
</tr>
<tr>
<td>GEDO THR</td>
<td>MEAN</td>
<td>12.0248</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>43.5256</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM BEST</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
<tr>
<td>KDSM CNE</td>
<td>MEAN</td>
<td>10.5472</td>
</tr>
</tbody>
</table>
### TABLE G-2  DIFFERENCE MEANS AND STANDARD DEVIATIONS

**BY TEST RUN AND REFERENCE (continued)**

<table>
<thead>
<tr>
<th>PROFILE</th>
<th>BEST</th>
<th>MEAN</th>
<th>RANGE</th>
<th>RANGE RATE</th>
<th>ROLL</th>
<th>PITCH</th>
<th>ROLL RATE</th>
<th>ALPHA</th>
<th>BETA</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL.46AC</td>
<td>CINE</td>
<td>MEAN</td>
<td>13.4669</td>
<td>-0.1477</td>
<td>0.6438</td>
<td>-0.639</td>
<td>-0.312</td>
<td>0.0118</td>
<td>0.4015</td>
</tr>
<tr>
<td></td>
<td>ST.DEV</td>
<td>18.3177</td>
<td>0.0373</td>
<td>0.1228</td>
<td>0.3715</td>
<td>0.0875</td>
<td>0.2251</td>
<td>0.1021</td>
<td>0.0860</td>
</tr>
<tr>
<td>HL.46AC</td>
<td>THR</td>
<td>MEAN</td>
<td>5.2928</td>
<td>0.6798</td>
<td>0.6359</td>
<td>-0.639</td>
<td>-0.414</td>
<td>0.0117</td>
<td>0.3812</td>
</tr>
<tr>
<td></td>
<td>ST.DEV</td>
<td>36.3080</td>
<td>0.0443</td>
<td>0.1228</td>
<td>0.3715</td>
<td>0.0875</td>
<td>0.2251</td>
<td>0.1021</td>
<td>0.0860</td>
</tr>
<tr>
<td>HL.46AC</td>
<td>BEST</td>
<td>MEAN</td>
<td>12.4325</td>
<td>0.1180</td>
<td>-0.2998</td>
<td>-0.417</td>
<td>0.3306</td>
<td>0.0764</td>
<td>0.2664</td>
</tr>
<tr>
<td></td>
<td>ST.DEV</td>
<td>19.3269</td>
<td>0.0473</td>
<td>0.1228</td>
<td>0.3715</td>
<td>0.0875</td>
<td>0.2251</td>
<td>0.1021</td>
<td>0.0860</td>
</tr>
<tr>
<td>HL.46AC</td>
<td>CINE</td>
<td>MEAN</td>
<td>15.7964</td>
<td>0.0286</td>
<td>-0.2926</td>
<td>-0.414</td>
<td>0.3273</td>
<td>0.0757</td>
<td>0.2525</td>
</tr>
<tr>
<td></td>
<td>ST.DEV</td>
<td>24.1269</td>
<td>0.0494</td>
<td>0.1228</td>
<td>0.3715</td>
<td>0.0875</td>
<td>0.2251</td>
<td>0.1021</td>
<td>0.0860</td>
</tr>
<tr>
<td>HL.46AC</td>
<td>BEST</td>
<td>MEAN</td>
<td>8.0726</td>
<td>0.3114</td>
<td>0.1634</td>
<td>0.3749</td>
<td>0.2148</td>
<td>0.2023</td>
<td>0.0871</td>
</tr>
<tr>
<td></td>
<td>ST.DEV</td>
<td>46.2292</td>
<td>0.0312</td>
<td>0.1634</td>
<td>0.3749</td>
<td>0.2148</td>
<td>0.2023</td>
<td>0.0871</td>
<td>0.1223</td>
</tr>
<tr>
<td>HL.46AC</td>
<td>CINE</td>
<td>MEAN</td>
<td>10.8358</td>
<td>0.2922</td>
<td>0.4329</td>
<td>0.6298</td>
<td>0.4227</td>
<td>0.6302</td>
<td>0.2397</td>
</tr>
<tr>
<td></td>
<td>ST.DEV</td>
<td>46.0489</td>
<td>0.0273</td>
<td>0.3866</td>
<td>0.1110</td>
<td>0.0012</td>
<td>0.0262</td>
<td>0.0304</td>
<td>0.0071</td>
</tr>
<tr>
<td>HL.46AC</td>
<td>BEST</td>
<td>MEAN</td>
<td>15.7205</td>
<td>0.1813</td>
<td>-0.3964</td>
<td>-0.1196</td>
<td>0.2051</td>
<td>0.0323</td>
<td>0.2014</td>
</tr>
<tr>
<td></td>
<td>ST.DEV</td>
<td>27.1848</td>
<td>0.0463</td>
<td>0.2393</td>
<td>0.3583</td>
<td>0.3523</td>
<td>0.2794</td>
<td>0.2992</td>
<td>0.3202</td>
</tr>
<tr>
<td>HL.46AC</td>
<td>CINE</td>
<td>MEAN</td>
<td>12.9801</td>
<td>0.2121</td>
<td>-0.4256</td>
<td>-0.4251</td>
<td>0.2324</td>
<td>0.6364</td>
<td>0.1832</td>
</tr>
<tr>
<td></td>
<td>ST.DEV</td>
<td>25.1886</td>
<td>0.0397</td>
<td>0.3937</td>
<td>0.3937</td>
<td>0.0875</td>
<td>0.2251</td>
<td>0.1021</td>
<td>0.0860</td>
</tr>
<tr>
<td>HL.46AC</td>
<td>BEST</td>
<td>MEAN</td>
<td>10.1874</td>
<td>0.5123</td>
<td>-0.4767</td>
<td>-0.8386</td>
<td>0.0753</td>
<td>0.1514</td>
<td>0.2714</td>
</tr>
<tr>
<td></td>
<td>ST.DEV</td>
<td>18.9782</td>
<td>0.0323</td>
<td>0.1787</td>
<td>0.1944</td>
<td>0.0875</td>
<td>0.2251</td>
<td>0.1021</td>
<td>0.0860</td>
</tr>
</tbody>
</table>

**RANGE MEASURED IN FEET**

**RANGE RATE MEASURED IN FEET/SECOND**

**ROLL MEASURED IN DEGREES**

**PITCH MEASURED IN DEGREES**

**ROLL RATE MEASURED IN DEGREES/SECOND**

**PITCH RATE MEASURED IN DEGREES/SECOND**

**ALPHA MEASURED IN DEGREES**

**BETA MEASURED IN DEGREES**

---

**G-5**
Table G-2 summarizes the flight conditions, targets used, and trajectories flown on 10/1/85.

TABLE G-2: 10/1/85 FLIGHT CONDITION SUMMARY

Flight Conditions: Heavy clouds in spots, hindering cinetheodolites. Ceiling was 2500 ft. and cover 2500 ft. thick.

Target: Helicopter

Trajectories flown:

- HL546AC
- HL246AD
- HL446AC
- HL346AD
- HJ146AC
Individual Flight Test Summaries

Trajectory: HL546AC
Range Equipment: 3 radars (R-350, R-393, R-394), no cines

Flight Profile (Brasscap)
\[ X: 47000 \rightarrow 27000 \text{ ft} \]
\[ Y: 1500 \rightarrow 30000 \text{ ft} \]
\[ -Z: \text{oscillates, } 3500 \rightarrow 11000 \text{ ft} \]
Duration: 450 s

Comments: Delta roll mimics the Z profile, delta pitch is the inverse of the Z profile, large \(-Z\) changes, downtrend in delta range, delta roll rate oscillates.

-----------------------------------

Trajectory: HL246AD
Range Equipment: 3 radars (R-350, R-393, R-394),
5 minutes of cine data.

Flight Profile (Brasscap)
\[ X: 42000 \rightarrow 30000 \text{ ft} \]
\[ Y: 8000 \rightarrow 36000 \text{ ft} \]
\[ -Z: \text{oscillates, } 5890 \rightarrow 6030 \text{ ft} \]
Duration: 500 s

Comments: V-shaped trend in delta roll, down to \(-.65 \text{ deg}\), oscillating delta pitch, bias of \(-.4 \text{ deg}\).
Trajectory: HL446AC
Range Equipment: 3 radars (R-350, R-393, R-394),
7 minutes of cine data.

Flight Profile (Brasscap) X: 48000 → 35000 ft
Y: 1250 → 31000 ft
-Z: stable around 6100 ft,
drops to 5550 at t=300 s

Duration: 425 s

Comments: Delta roll seems to follow the Z profile, as in HL546AC
delta range mean=22.46, std. dev.=25.36
delta pitch skews up.

----------------------------------------

Trajectory: HL146AC not available

----------------------------------------

Trajectory: HL346AD
Range Equipment: 3 radars (R-350, R-393, R-394),
small amounts of cine data.

Flight Profile (Brasscap) X: 45500 → 32500 ft
Y: 14000 → 36000 ft
-Z: oscillates, 5820 → 5900 ft

Duration: 400 s

Comments: Delta roll mean= -.38 deg, oscillates, delta pitch
mean= -.43 deg, oscillates.
Trajectory: HJI46AC
Range Equipment: 3 radars (R-350, R-393, R-394), small amounts of cine data.

Flight Profile (Brasscap)  X: 64000 → 30000 ft  
Y: 0 → 32500 ft  
-Z: oscillates, 5500 → 6200 ft

Duration: 600 s

Comments: Delta roll appears to mimic the Z profile, delta pitch has a similar pattern, delta range mean = 23.8, std.dev. = 40.39
Table G-3 summarizes the flight conditions, targets used, and trajectories flown on 10/3/85.

**TABLE G-3: 10/3/85 FLIGHT CONDITION SUMMARY**

Flight Conditions: Good weather, winds tended to increase the target's velocity and slightly altered the flight path.

Target: Helicopter

Trajectories Flown:
- HEL30AE
- HEL30AF
- H30SKAE
- H30SKAF
- HEL30AG
- HEL30AH
- H30SKAG
- H30SKAH
- H30SKAI
- HEL30AI
- HEL30AJ
Individual Flight Test Summaries

Trajectory: HEL30AE  
not available

Trajectory: HEL30AF  
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines  
Flight Profile (Brasscap)  
X: 9500 → 3250 ft  
Y: 5200 → 5400 → 3400 ft  
Z: 6000 → 5100 ft  
Duration: 160 s

Comments: Large trends in delta roll and delta pitch, on the order of .2 deg, discontinuity of 30 ft in delta range at t=50 s.

Trajectory: H30SKAE  
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines  
Flight Profile (Brasscap)  
X: 2800 → 1000 ft  
Y: oscillates, 1580 → 1340 ft  
Z: 1600 → 2000  
Duration: 100 s

Comments: Delta roll and delta pitch mimic the Y profile, sinusoidal delta range has 30 ft deflections, delta range rate deflections of 5 ft, trends in delta pitch and roll rates.
Trajectory: H30SKAF
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  
X: 3200 → 1400  
Y: 2100 → 800  
Z: 1550 → 1450

Duration: 130 s

Comments: Large trends in delta range (40-50 ft), delta range rate std. dev. = 1.72 large deflections in delta roll and pitch, up to 1.6 deg.

---

Trajectory: HEL30AG
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  
X: 10250 → 4600 ft  
Y: 5000 → 6200 → 1500 ft  
Z: 6400 → 5200

Duration: 260 s

Comments: Large trend in delta roll (.5 deg), delta pitch discontinuity at t=150 s

---

Trajectory: HEL30AH  not available

---

G-12
Trajectory: H30SKAG
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  
X: 1590 → 1360 ft  
Y: 860 → 620 ft  
-Z: 1636 → 1652 → 1644 ft

Duration: 25 s

Comments: Large trend in delta roll and pitch (.7 deg), trends in delta range, 35 ft deflections.

-----------------------------

Trajectory: H30SKAH
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  
X: 3100 → 1400 ft  
Y: 1900 → 900 ft  
-Z: 1700 → 1800 → 1580 ft

Duration: 110 s

Comments: Large downtrend in delta range (50 ft), delta roll has 1.4 deg deflections, delta pitch has .7 deg deflections, large delta roll and pitch rates (.6 deg/s).

-----------------------------

Trajectory: H30SKAI
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  
X: 2700 → 2300 ft  
Y: 1900 → 1890 → 1930 ft  
-Z: 1620 → 1660 → 1550 ft

Duration: 40 s

Comments: Trends in delta roll, pitch and range.
Trajectory: HEL30AI
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  
X: 10000 → 4000 ft  
Y: 6000 → 2100 ft  
- Z: 6500 → 3300 ft

Duration: 375 s

Comments: Large delta roll skew (.8 deg), delta pitch skew of .3 deg, discontinuity of 40 ft at t=100 s.

Trajectory: HEL30AJ
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  
X: 10500 → 4000 ft  
Y: 6000 → 2000 ft  
- Z: 6200 → 4000 ft

Duration: 400 s

Comments: Large trend in delta range (90 ft deflections), trends in delta roll (2.5 deg), trends in delta pitch (.8 deg), oscillating delta roll rate.
Table G-4 summarizes the flight conditions, targets used, and trajectories flown on 10/5/85.

**TABLE G-4: 10/5/85 FLIGHT CONDITION SUMMARY**

Flight Conditions: Good weather, slight winds.

Target: Helicopter

Trajectories Flown:
- HL546AE
- HL346AE
- HL446AD
- HL146AD
- HJ146AD
- HL546AF
- HL246AE
Individual Flight Test Summaries

Trajectory: HL546AE
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  
X: 47000 → 24000 ft
Y: 1000 → 32500 ft
-Z: oscillates, 5900 → 6300 ft

Duration: 475 s

Comments: Small trends in delta roll and pitch (.1 deg).

Trajectory: HL346AE
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  
X: 47000 → 32000 ft
Y: 0 → 37000 ft
-Z: oscillates, 5940 → 6170 ft

Duration: 550 s

Comments: Small spikes (.06 deg) in delta roll and pitch.
Trajectory: HL446AD
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  X: 49000 → 31000 ft
                        Y: 0 → 35000 ft
                        Z: oscillates, 5850 → 6250 ft

Duration: 650 s

Comments: Oscillations in delta roll (.25 deg), and delta pitch (.2 deg).

---------------------------------------------

Trajectory: HL146AD  not available

---------------------------------------------

Trajectory: HJ146AD
Range Equipment: 3 radars (R-350, R-393, R-394), 5 cines

Flight Profile (Brasscap)  X: 64000 → 36000 ft
                        Y: 0 → 29000 ft
                        Z: oscillates, 5775 → 6050 ft

Duration: 575 s

Comments: Large delta range std. dev. = 69 ft, oscillations in delta roll and pitch (.4 deg).

---------------------------------------------
Trajectory: HL546AF
Range Equipment: 3 radars (R-350, R-393, R-394), no cines

Flight Profile (Brasscap)  
X: 46000 → 24000 ft  
Y: 0 → 32500 ft  
-Z: 6850 → 7200 → 6900 ft

Duration: 600 s

Comments: Delta range std. dev. = 51 ft, large bias in delta roll (1.48 deg), and delta pitch (-1.17 deg), V-shaped trends in data.

Trajectory: HL246AE
Range Equipment: 3 radars (R-350, R-393, R-394)

Flight Profile (Brasscap)  
X: 42000 → 29000 ft  
Y: 0 → 36000 ft  
-Z: oscillates, 5900 → 6175 ft

Duration: 700 s

Comments: Slight trend in delta roll, mean = .02, std. dev. = .09  
delta pitch mean = -.65, std. dev. = .06
Table G-5 summarizes the flight conditions, targets used, and trajectories flown on 10/16/85.

**TABLE G-5: 10/16/85 FLIGHT CONDITION SUMMARY**

Flight Conditions: Drizzling rain, low ceiling (3000 ft).

Target: Gemspheres (free floating).

Trajectories Flown:
- GEM1
- GEM2
- GEM3
G.4.1

Individual Flight Test Summaries

Trajectory: GEM1
not available

-----------

Trajectory: GEM2
Range Equipment: 3 radars (R-350, R-393, R-394)

Flight Profile (Brasscap)  
X: 2000 → 28000 ft
Y: -1500 → 2000 ft
-Z: 2000 → 11000 ft

Duration: 500 s

Comments: Track lost at first, but picked up at range of 4000 ft.  
downtrend in delta range, flat delta range rate,  
but large std. dev.=2.17,  
large trends in delta roll and pitch (1.6 deg)  
trends in delta roll and pitch rate.

-----------

Trajectory: GEM3
Range Equipment: 3 radars (R-350, R-393, R-394)

Flight Profile (Brasscap)  
X: 1000 → 24000 ft
Y: -500 → -3500 → 500 ft
-Z: 1500 → 10000 ft

Duration: 500 s

Comments: Initially lost track but required,  
large downtrend in delta range (175 ft),  
delta range rate std. dev.=1.83,  
large trends in delta roll and pitch ( 2 deg),  
also in delta roll and pitch rates.
Table G-6 summarizes the flight conditions, targets used, and trajectories flown on 10/19/85.

TABLE G-6: 10/19/85 FLIGHT CONDITION SUMMARY

Flight Conditions: Good conditions

Target: 2m Gemsphere suspended below 2, 10 ft balloons, tethered flight.

Trajectories Flown:
SAT1
SAT2
SAT3
SAT4
SAT6
SAT8
G.5.1

Individual Flight Test Summaries

Trajectory: SAT1
Range Equipment: 3 radars (R-350, R-393, R-394), 10 cines

Flight Profile (Brasscap)  
X: 450 → 300 → 700 ft
Y: -1340 → -1440 → -1220 ft
Z: 2180 → 2050 → 2120 ft

Duration: 600 s

Comments: Large trend in delta roll (.6 deg), and delta pitch (.8 deg) trend in delta range (80 ft). Delta range rate std. dev. = 2.33. Trajectory had large roll angles up to -74 deg.

------------------------------------------------------------------------

Trajectory: SAT2
Range Equipment: 3 radars (R-350, R-393, R-394), 10 cines

Flight Profile (Brasscap)  
X: 725 → 450 → 1075 ft
Y: -750 → -100 ft
Z: oscillates 2290 → 2390 ft

Duration: 600 s

Comments: Large trends in delta roll (1.8 deg), and delta pitch (.6 deg), roll angles up to -62 deg, large trends in delta range (80 ft).

------------------------------------------------------------------------
Trajectory: SAT3
Range Equipment: 3 radar (R-350, R-393, R-394), 10 cines

Flight Profile (Brasscap)  
X: 1000 → -800 ft  
Y: 0 → 700 ft  
-Z: 2200 → 200 ft  

Duration: 600 s  

Comments: The balloon tether broke on this flight. 
trends in delta range (160 ft),
delta range rate std. dev.=6.78,
anomalies in delta roll and pitch.

Trajectory: SAT4
Range Equipment: 3 radars (R-350, R-393, R-394), 10 cines

Flight Profile (Brasscap)  
X: 6400 → 8600 ft  
Y: 5600 → 7800 → 6300 ft  
-Z: 6750 → 5100 ft  

Duration: 600 s  

Comments: Sporadic delta range (60 ft deflections), trends in 
delta roll and pitch (.4 deE), deltaroll and pitch rates have damped oscillations.

Trajectory: SAT6  
not available

Trajectory: SAT8  
not available
Table G-7 summarizes the flight conditions, targets used, and trajectories flown on 11/4/85.

TABLE G-7: 11/4/85 FLIGHT CONDITION SUMMARY

Flight Conditions: Higher altitude winds caused the target balloons to drift back over the Pearl site.

Target: Gemspheres (free floating) and helicopters.

Note: The antenna servo gain had been increased on 11/2.

Trajectories Flown:

BAL1
BAL2
BAL5
BAL6
BAL7
HL546AG
HL246AF
HL446AE
HL146AE
HL346AF
HJ146AE
G.6.1 Individual Flight Test Summaries

Trajectory: BAL1
Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap)  
X: 500 $\rightarrow$ 3300 $\rightarrow$ 900 ft
Y: 300 $\rightarrow$ 1900 ft
-Z: 500 $\rightarrow$ 10000 ft
Duration: 600 s

Comments: Large bias and initial skew on delta roll and pitch (2 deg), discontinuity in delta range at t=250 s, delta range rate std. dev.=1.39.

Trajectory: BAL2
Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap)  
X: 750 $\rightarrow$ 3500 ft
Y: 200 $\rightarrow$ 1000 ft
-Z: 300 $\rightarrow$ 4100 ft
Duration: 300 s

Comments: Large trends in delta roll and pitch (1 deg), oscillations in delta range (7 ft), delta range rate std. dev.=3.08, large delta roll and pitch rate deflections at t=75 s (.6 deg).
Trajectory: BAL5
Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap)  
X: 3500 → 1600 ft  
Y: 1000 → 2000 ft  
-Z: 7100 → 10000 ft  

Duration: 170 s

Comments: Large bias in delta roll (.42 deg) and delta pitch (-.64 deg), trends in these of .15 deg.  
delta range rate std. dev.=1.2 deg/s.

-------------------------------------------------------------------------------------------------

Trajectory: BAL6
Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap)  
X: 750 → 3750 → 1750 ft  
Y: 300 → 2400 ft  
-Z: 500 → 10000 ft  

Duration: 600 s

Comments: Large initial skew in delta pitch and roll (1.6 deg), and delta pitch and roll rate (.55 deg/s)  
delta range rate std. dev.=1.78.

-------------------------------------------------------------------------------------------------
Trajectory: BAL7
Range Equipment: 1 radar (R-394), no cines

Flight Profile (Brasscap)  
X: 900 → 3800 → 2250 ft  
Y: 300 → 2400 ft  
Z: 500 → 10000 ft

Duration: 600 s

Comments: Large initial skew in delta roll and pitch (1.6 deg), delta range rate std. dev.=2.91, also an initial skew in delta roll and pitch rates

Trajectory: HL546AG
Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap)  
X: 45000 → 25000 ft  
Y: 8000 → 31000 ft  
Z: oscillates 6040 → 6300 ft

Duration: 500 s

Comments: Delta roll within .1 deg std. dev., delta pitch still has mean of -.64 deg.

Trajectory: HL246AF  
not available
Trajectory: HL446AE
Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap)  
X: 48000 → 30000 ft  
Y: 0 → 35000 ft  
-Z: oscillates 6000 → 6400 ft

Duration: 550 s

Comments: Large down spike (2 deg) in delta roll and pitch at t= 225-275 s. Due to glitch in KU angles.

------------------------------------------------------------------------------------------------------------------

Trajectory: HL146AE
Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap)  
X: 46000 → 25000 ft  
Y: 0 → 35000 ft  
-Z: oscillates 6120 → 6340 ft

Duration: 600 s

Comments: Delta roll is fairly flat, mean=.07, std. dev.=.12, delta pitch is still biased mean=-.64 deg, std. dev.=.07.

------------------------------------------------------------------------------------------------------------------

Trajectory: HL346AF
Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap)  
X: 47000 → 31000 ft  
Y: 0 → 37000 ft  
-Z: oscillates 6125 → 6255 ft

Duration: 550 s

Comments: Delta range rate std. dev.=.55, delta roll is fair (mean=.09, std. dev.=.1) delta pitch mean=-.63 std. dev.=.07
Trajectory: HJ146AE
Range Equipment: 1 radar (R-394), 5 cines

Flight Profile (Brasscap)  
X: 65000 → 35000 ft  
Y: 0 → 30000 ft  
Z: oscillates 6080 → 6300 ft

Duration: 600 s

Comments: Delta roll mean=-0.08, std. dev.=0.24, delta pitch mean=-0.62, std. dev.=0.1
APPENDIX H
ADDENDUM TO SORTE ANGLE RATE DATA ANALYSIS

The purpose of this appendix is to augment the angle rate data analysis presented in Section 3.6. In particular, in the one case (H30SKAF) that was analyzed in detail, it was found that the principal error source was angle acceleration. Furthermore, the bias-effect on the angle rate was exactly predictable from a knowledge of the acceleration and the natural frequency of the loop, $f_n$. As noted there, this was the first corroboration that the angle rate loop is properly represented by the model in Figure 3.6-4, and that the $f_n$ value for the widest bandwidth case has been properly implemented in the hardware. Since there are two other bandwidth values for the angle rate tracker, the purpose of this appendix is to verify that the other two $f_n$ values are implemented properly through the use of the angle acceleration data.

Table H-1 summarizes the values of $f_n$ for the different range intervals in the passive tracking mode. As noted earlier, the H30SKAF data was used to analyze the wide bandwidth case. Here, the first 150 seconds of the HEL30AG profile is used to analyze the medium bandwidth case, and HL446AC profile is used to analyze the narrow bandwidth case.

<table>
<thead>
<tr>
<th>RANGE INTERVAL</th>
<th>$f_n$, Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&lt; 11,510$</td>
<td>0.120</td>
</tr>
<tr>
<td>$11,520$ to $23,020$</td>
<td>0.070</td>
</tr>
<tr>
<td>$&gt; 23,030$</td>
<td>0.027</td>
</tr>
</tbody>
</table>

Medium Bandwidth Case. Figures H-1 and H-2 compare the angle rate difference data and the corresponding angle acceleration for pitch and roll rate, respectively. As was done in Section 3.6, a time interval was selected in each data set and the angle rate bias formula of equation 3-12 was applied to determine if the relation was satisfied. Table H-2 summarizes the results.
FIGURE H-1  A COMPARISON OF THE CINE PITCH ANGLE ACCELERATION PROFILE AND THE CINE PITCH RATE DIFFERENCE PROFILE FOR THE HEL30AG FLIGHT
FIGURE H-2  A COMPARISON OF THE CINE ROLL ANGLE ACCELERATION PROFILE AND THE CINE ROLL RATE DIFFERENCE PROFILE FOR THE HEL30AC FLIGHT
of these selections and computations. It should be pointed out that the average angle acceleration and the measured angle rate bias are "eyeball" estimates taken from Figure H-1 and H-2. The data of Table H-2 shows a very close match between computed and measured angle rate bias. It can be concluded that the value of $f_n$ (0.07) for this range interval has been correctly implemented in the hardware.

**TABLE H-2 EVALUATION OF ANGLE ACCELERATION BIAS EFFECTS IN THE MEDIUM BANDWIDTH CASE**

<table>
<thead>
<tr>
<th>TIME INTERVAL, SEC</th>
<th>AVERAGE ANGLE ACCELERATION</th>
<th>COMPUTED ANGLE RATE BIAS</th>
<th>MEASURED ANGLE RATE BIAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll Rate 40 to 50</td>
<td>0.0060 deg/sec²</td>
<td>-0.027 deg/sec</td>
<td>-0.023 deg/sec</td>
</tr>
<tr>
<td>Roll Rate 85 to 95</td>
<td>-0.0125</td>
<td>0.056</td>
<td>0.050</td>
</tr>
<tr>
<td>Pitch Rate 20 to 30</td>
<td>0.0030</td>
<td>-0.0136</td>
<td>-0.015</td>
</tr>
<tr>
<td>Pitch Rate 82 to 92</td>
<td>-0.0070</td>
<td>0.0318</td>
<td>0.028</td>
</tr>
</tbody>
</table>

Narrow Bandwidth Case. Figures H-3 and H-4 give the angle rate difference and the corresponding angle acceleration for pitch and roll rate, respectively. Table H-3 provides the results of how the angle acceleration bias affects computations. In this case, it is very hard to identify the angle rate bias because it appears to be buried in the thermal noise and other effects. There were some time intervals where the acceleration effects were prominent. In those cases, there was good agreement between the predicted bias and the measured bias.

**TABLE H-3 EVALUATION OF ANGLE ACCELERATION BIAS EFFECTS IN THE NARROW BANDWIDTH CASE**

<table>
<thead>
<tr>
<th>TIME INTERVAL, SEC</th>
<th>AVERAGE ANGLE ACCELERATION</th>
<th>COMPUTED ANGLE RATE BIAS</th>
<th>MEASURED ANGLE RATE BIAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll Rate 150 to 170</td>
<td>0.001 deg/sec²</td>
<td>-0.0118 deg/sec</td>
<td>-0.009 deg/sec</td>
</tr>
<tr>
<td>Roll Rate 25 to 50</td>
<td>-0.0005</td>
<td>0.0059</td>
<td>0.005</td>
</tr>
<tr>
<td>Pitch Rate 25 to 35</td>
<td>-0.0008</td>
<td>0.0094</td>
<td>0.008</td>
</tr>
</tbody>
</table>
FIGURE H-3  A COMPARISON OF THE CINE PITCH ANGLE ACCELERATION PROFILE AND THE CINE PITCH RATE DIFFERENCE PROFILE FOR THE HL446AC FLIGHT
FIGURE H-4 A COMPARISON OF THE CINE ROLL RATE ACCELERATION PROFILE AND THE CINE ROLL RATE DIFFERENCE PROFILE FOR THE HL446AC FLIGHT