NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE
KU-BAND RENDEZVOUS RADAR PERFORMANCE COMPUTER SIMULATION MODEL

FINAL REPORT

Radar Systems Group
Hughes Aircraft Company
2000 E. Imperial Highway
El Segundo, California 90245

July 1980

Contract NAS9-15840

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Johnson Space Flight Center
Houston, Texas 77058
J. W. Griffin, Technical Officer
KU-BAND RENDEZVOUS RADAR PERFORMANCE

COMPUTER SIMULATION MODEL

FINAL REPORT

CONTRACT NAS 9-15840

Prepared for: National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas 77058
J. W. Griffin, Technical Monitor

Prepared by: Radar Subsystem Staff
Engineering Division
Radar Systems Group
Hughes Aircraft Company
El Segundo, CA 90245

July 1980
# TABLE OF CONTENTS

1. INTRODUCTION AND OVERVIEW. ........................................ 1
   1.1 Introduction .................................................. 1
   1.2 Overview of Radar Performance Computer Model ................ 2
      1.2.1 Target Scattering Model Summary ......................... 2
      1.2.2 General Computer Model Structure ....................... 3
      1.2.3 Radar Search and Acquisition Performance Model Summary .. 3
      1.2.4 Radar Tracking Performance Model Summary ............... 7
   1.3 Report Organization ............................................ 14

2. DEFINITION OF COORDINATE SYSTEMS AND VECTOR NOTATIONS. ....... 17
   2.1 Coordinate System Definitions ................................ 17
   2.2 Definition of Vector and Transformation Notation .......... 22

3. RADAR SIMULATION/PARENT SIMULATION INTERFACE DESCRIPTION AND REQUIREMENTS ........................................ 27
   3.1 Input Data Required From the Parent Simulation ............ 28
      3.1.1 Required Radar Controls ................................ 28
      3.1.2 Required Target/Orbiter Position and Motion Data ....... 28
   3.2 Output Data to the Parent Simulation .......................... 32
   3.3 Input/Output Data Format ..................................... 36
   3.4 Interface Timing Requirements .................................. 36
      3.4.1 Simulation Cycle Time Requirements ..................... 36
      3.4.2 Maximum Computation Time Requirements ................ 38
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. TARGET MODELING METHOD</td>
<td>41</td>
</tr>
<tr>
<td>4.1 General Approach</td>
<td>41</td>
</tr>
<tr>
<td>4.2 Scattering Centers and Cross-Sections for Simple and</td>
<td>41</td>
</tr>
<tr>
<td>Representative Shapes</td>
<td></td>
</tr>
<tr>
<td>4.2.1 Smoothly Curved Bodies</td>
<td>42</td>
</tr>
<tr>
<td>4.2.2 Other Shapes</td>
<td>45</td>
</tr>
<tr>
<td>4.2.3 Reflector Antennas</td>
<td>47</td>
</tr>
<tr>
<td>4.3 STAS Model</td>
<td>51</td>
</tr>
<tr>
<td>4.3.1 Satellite and its Coordinate System</td>
<td>51</td>
</tr>
<tr>
<td>4.3.2 Scatterer Selection Strategy</td>
<td>51</td>
</tr>
<tr>
<td>4.3.3 Point-Scatterer Model</td>
<td>55</td>
</tr>
<tr>
<td>4.3.4 Effect of Thermal Blanket</td>
<td>58</td>
</tr>
<tr>
<td>4.3.5 Recommendation</td>
<td>58</td>
</tr>
<tr>
<td>4.4 Mathematical Description of Target Return Signal</td>
<td>58</td>
</tr>
<tr>
<td>4.4.1 Antenna Weighting Factor Computation</td>
<td>60</td>
</tr>
<tr>
<td>4.4.2 Computation of Scatterer Phase</td>
<td>62</td>
</tr>
<tr>
<td>4.5 Computer Algorithm Details</td>
<td>64</td>
</tr>
<tr>
<td>5. SEARCH AND ACQUISITION MODE COMPUTER MODEL DESCRIPTION</td>
<td>72</td>
</tr>
<tr>
<td>5.1 Summary of Ku-Band Radar Search Mode Operation</td>
<td>74</td>
</tr>
<tr>
<td>5.1.1 General Antenna Steering Mode Operation</td>
<td>74</td>
</tr>
<tr>
<td>5.1.2 Display Meters</td>
<td>75</td>
</tr>
<tr>
<td>5.1.3 Search Mode Waveforms and Signal Processing.</td>
<td>75</td>
</tr>
<tr>
<td>5.1.4 Antenna Scan Operation</td>
<td>77</td>
</tr>
<tr>
<td>5.2 Search Mode Control Algorithm Description</td>
<td>83</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3 Gimbal Pointing Loop Model Description</td>
<td>90</td>
</tr>
<tr>
<td>5.3.1 Basic Servo Loop Model Definition</td>
<td>91</td>
</tr>
<tr>
<td>5.3.2 Computer Algorithm Details</td>
<td>95</td>
</tr>
<tr>
<td>5.4 Scan Model Description</td>
<td>100</td>
</tr>
<tr>
<td>5.4.1 Summary of Scan Operation.</td>
<td>103</td>
</tr>
<tr>
<td>5.4.2 Computer Algorithm Details</td>
<td>103</td>
</tr>
<tr>
<td>5.5 Detection Model Description</td>
<td>108</td>
</tr>
<tr>
<td>5.5.1 Model Assumption</td>
<td>108</td>
</tr>
<tr>
<td>5.5.2 CFAR Detection Model</td>
<td>110</td>
</tr>
<tr>
<td>5.5.3 Single-Hit Detection Model</td>
<td>112</td>
</tr>
<tr>
<td>5.5.4 Determination of $P_d$(SNR) Data</td>
<td>112</td>
</tr>
<tr>
<td>5.5.5 Computer Algorithm Details</td>
<td>114</td>
</tr>
<tr>
<td>6. TRACK MODE COMPUTER MODEL DESCRIPTION</td>
<td>123</td>
</tr>
<tr>
<td>6.1 Summary of Ku-Band Radar Track Mode Operation.</td>
<td>123</td>
</tr>
<tr>
<td>6.1.1 General Antenna Steering Mode Operation.</td>
<td>123</td>
</tr>
<tr>
<td>6.1.2 Data Valid Flags</td>
<td>125</td>
</tr>
<tr>
<td>6.1.3 Display Meters</td>
<td>125</td>
</tr>
<tr>
<td>6.1.4 Break-Track Algorithm.</td>
<td>127</td>
</tr>
<tr>
<td>6.1.5 Track Waveforms.</td>
<td>127</td>
</tr>
<tr>
<td>6.1.6 Tracking Loops and Signal Processor Operation.</td>
<td>129</td>
</tr>
<tr>
<td>6.2 Track Mode Control Algorithm Description</td>
<td>129</td>
</tr>
<tr>
<td>6.2.1 Track Mode Initialization Control</td>
<td>129</td>
</tr>
<tr>
<td>6.2.2 Tracking Loop Update Control</td>
<td>134</td>
</tr>
<tr>
<td>6.3 Tracking Loop Initialization Algorithm Description</td>
<td>138</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3.1 Break-Track Algorithm Initialization</td>
<td>140</td>
</tr>
<tr>
<td>6.3.2 Angle and Angle Rate Tracking Model Initialization</td>
<td>140</td>
</tr>
<tr>
<td>6.3.3 Range Tracking Model Initialization</td>
<td>141</td>
</tr>
<tr>
<td>6.3.4 Signal Processor Parameter Initialization</td>
<td>141</td>
</tr>
<tr>
<td>6.3.5 Velocity Processor Model Initialization</td>
<td>142</td>
</tr>
<tr>
<td>6.3.6 Signal Strength Algorithm Initialization</td>
<td>142</td>
</tr>
<tr>
<td>6.4 Signal Generation and Processing Model Description</td>
<td>144</td>
</tr>
<tr>
<td>6.4.1 Model Assumptions</td>
<td>147</td>
</tr>
<tr>
<td>6.4.2 Target Position and Motion Computation Model</td>
<td>148</td>
</tr>
<tr>
<td>6.4.3 Angle Discriminant Computation Model</td>
<td>149</td>
</tr>
<tr>
<td>6.4.4 Range Discriminant Computation Model</td>
<td>155</td>
</tr>
<tr>
<td>6.4.5 Velocity Discriminant Computation Model</td>
<td>159</td>
</tr>
<tr>
<td>6.4.6 On-Target Discriminant Computation Model</td>
<td>163</td>
</tr>
<tr>
<td>6.4.7 Radar Signal Strength Computation Model</td>
<td>163</td>
</tr>
<tr>
<td>6.4.8 Computer Model Details</td>
<td>164</td>
</tr>
<tr>
<td>6.5 Break-Track Algorithm Description</td>
<td>175</td>
</tr>
<tr>
<td>6.5.1 Noise-Free Discriminant Response Functions</td>
<td>175</td>
</tr>
<tr>
<td>6.5.2 Determination of a No-Target Condition</td>
<td>178</td>
</tr>
<tr>
<td>6.5.3 Break-Track Determination</td>
<td>181</td>
</tr>
<tr>
<td>6.5.4 Computer Algorithm Details</td>
<td>181</td>
</tr>
<tr>
<td>6.6 Angle and Angle Rate Tracking Loop Model Description</td>
<td>181</td>
</tr>
<tr>
<td>6.6.1 The Model</td>
<td>181</td>
</tr>
<tr>
<td>6.6.2 Model Assumptions and Approximations</td>
<td>187</td>
</tr>
<tr>
<td>6.6.3 Error Sources Modeled</td>
<td>193</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.6.4 Model Performance</td>
<td>195</td>
</tr>
<tr>
<td>6.6.5 Computer Model Details</td>
<td>199</td>
</tr>
<tr>
<td>6.7 Range and Range Rate Tracking Loop Model Description</td>
<td>202</td>
</tr>
<tr>
<td>6.7.1 Range Tracker Model Description</td>
<td>204</td>
</tr>
<tr>
<td>6.7.2 Velocity Processor Model Description</td>
<td>209</td>
</tr>
<tr>
<td>6.7.3 Computer Algorithm Details</td>
<td>221</td>
</tr>
<tr>
<td>7. RECOMMENDATIONS FOR FURTHER STUDY AND DEVELOPMENT</td>
<td>229</td>
</tr>
<tr>
<td>7.1 System Analysis</td>
<td>229</td>
</tr>
<tr>
<td>7.2 Radar model Fidelity Improvement</td>
<td>229</td>
</tr>
<tr>
<td>7.3 Target Model Fidelity Improvement</td>
<td>230</td>
</tr>
<tr>
<td>References and Bibliography</td>
<td>231</td>
</tr>
<tr>
<td>APPENDIX A DERIVATION OF ANGLE AND ANGLE RATE TRACKING LOOP MODEL INITIALIZATION</td>
<td>234</td>
</tr>
<tr>
<td>A.1 Derivation of Target Inertial LOS Azimuth and Elevation Rate Initializations</td>
<td>234</td>
</tr>
<tr>
<td>A.2 Derivation of $\alpha$ and $\beta$ Gimbal Rate Initialization</td>
<td>236</td>
</tr>
<tr>
<td>APPENDIX B DERIVATION OF TARGET PITCH ANGLE, ROLL ANGLE, INERTIAL ROLL RATE, AND INERTIAL PITCH RATE TRANSFORMATIONS</td>
<td>238</td>
</tr>
<tr>
<td>B.1 Definitions and Assumptions</td>
<td>238</td>
</tr>
<tr>
<td>B.2 Derivation of Target Roll and Pitch Angle Transformations</td>
<td>240</td>
</tr>
<tr>
<td>B.3 Derivation of Target Inertial Roll and Pitch Rate Transformation</td>
<td>241</td>
</tr>
<tr>
<td>APPENDIX C DERIVATION OF NOISE-FREE DISCRIMINANT COMPONENT COMPUTATION MODELS</td>
<td>243</td>
</tr>
<tr>
<td>C.1 Model Assumptions</td>
<td>243</td>
</tr>
<tr>
<td>C.2 Noise-Free Magnitude-Squared Detector Response Derivation</td>
<td>243</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>C.3 Discriminant Component Computation Models.</td>
<td>249</td>
</tr>
<tr>
<td>C.3.1 Angle Discriminant Component Computation</td>
<td>249</td>
</tr>
<tr>
<td>C.3.2 Range Discriminant Component Computation</td>
<td>250</td>
</tr>
<tr>
<td>C.3.4 Velocity Discriminant Component Computation.</td>
<td>250</td>
</tr>
<tr>
<td>APPENDIX D DERIVATION OF THERMAL NOISE MODEL</td>
<td>252</td>
</tr>
<tr>
<td>D.1 Model Assumptions.</td>
<td>252</td>
</tr>
<tr>
<td>D.2 Noise Model Derivation</td>
<td>254</td>
</tr>
<tr>
<td>D.2.1 Derivation of Mean and Variance at PDI Output</td>
<td>254</td>
</tr>
<tr>
<td>D.2.2 Derivation of PDF for Z.</td>
<td>256</td>
</tr>
<tr>
<td>D.3 Practical Aspects of Model Implementation.</td>
<td>257</td>
</tr>
<tr>
<td>APPENDIX E CROSS SECTION CALCULATION NOTES</td>
<td>259</td>
</tr>
<tr>
<td>APPENDIX F A MODEL FOR CENTRIC WANDER IN ROUGH SURFACE MODELS</td>
<td>261</td>
</tr>
<tr>
<td>APPENDIX G LISTING OF SIMULATION MODEL COMPUTER CODE</td>
<td>264</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Simplified Block Diagram of the Computer Model</td>
<td>4</td>
</tr>
<tr>
<td>1-2</td>
<td>Outline of Search and Acquisition Mode Computer Algorithm</td>
<td>5</td>
</tr>
<tr>
<td>1-3</td>
<td>Simplified Block Diagram of the Gombal Pointing Loop Model</td>
<td>6</td>
</tr>
<tr>
<td>1-4</td>
<td>Fundamental Detection Model Configuration</td>
<td>8</td>
</tr>
<tr>
<td>1-5</td>
<td>Outline of Track Mode Computer Algorithm</td>
<td>9</td>
</tr>
<tr>
<td>1-6</td>
<td>Track Mode Signal Processor Computer Model</td>
<td>10</td>
</tr>
<tr>
<td>1-7</td>
<td>Simplified Block Diagram of Break-Track Algorithm</td>
<td>12</td>
</tr>
<tr>
<td>1-8</td>
<td>Angle and Angle Rate Discrete-Time Tracking Loop Model</td>
<td>13</td>
</tr>
<tr>
<td>1-9</td>
<td>Range Discrete-Time Tracking Loop Model</td>
<td>15</td>
</tr>
<tr>
<td>1-10</td>
<td>Ku-Band Radar Velocity Processor</td>
<td>16</td>
</tr>
<tr>
<td>2-1</td>
<td>Definition of Positive Rotation About a Coordinate Axis</td>
<td>18</td>
</tr>
<tr>
<td>2-2</td>
<td>Examples of Possible Target (T) Frame Orientations</td>
<td>19</td>
</tr>
<tr>
<td>2-3</td>
<td>Orbiter Body (B) Frame Definition</td>
<td>20</td>
</tr>
<tr>
<td>2-4</td>
<td>Radar (R) Frame Orientation with Respect to the Orbiter Body Frame</td>
<td>21</td>
</tr>
<tr>
<td>2-5</td>
<td>Outer Gimbal (G) Frame Orientation with Respect to the Radar Frame</td>
<td>24</td>
</tr>
<tr>
<td>2-6</td>
<td>Antenna Los (L) Frame Orientation with Respect to the Outer Gimbal Frame</td>
<td>24</td>
</tr>
<tr>
<td>3-1</td>
<td>Illustration of Orbiter - Point Target Geometry</td>
<td>39</td>
</tr>
<tr>
<td>3-2</td>
<td>Example of Effects of Different Sample Intervals</td>
<td>39</td>
</tr>
<tr>
<td>4-1</td>
<td>Toruspherical - Ended Cylinder Geometry</td>
<td>44</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4-2</td>
<td>Cylinder Geometry</td>
<td>46</td>
</tr>
<tr>
<td>4-3</td>
<td>Dehedral Corner Reflector Geometry</td>
<td>48</td>
</tr>
<tr>
<td>4-4</td>
<td>Reflector Geometry</td>
<td>50</td>
</tr>
<tr>
<td>4-5</td>
<td>SPAS Isometric View</td>
<td>52</td>
</tr>
<tr>
<td>4-6</td>
<td>SPAS Spacecraft</td>
<td>53</td>
</tr>
<tr>
<td>4-7</td>
<td>Antenna Sum Pattern</td>
<td>61</td>
</tr>
<tr>
<td>4-8</td>
<td>Antenna Difference Pattern</td>
<td>63</td>
</tr>
<tr>
<td>4-9</td>
<td>Target Model Computer Algorithm (1 of 3)</td>
<td>65</td>
</tr>
<tr>
<td>4-9</td>
<td>Target Model Computer Algorithm (2 of 3)</td>
<td>66</td>
</tr>
<tr>
<td>4-9</td>
<td>Target Model Computer Algorithm (3 of 3)</td>
<td>67</td>
</tr>
<tr>
<td>5-1</td>
<td>Outline of Search and Acquisition Mode Computer Algorithm</td>
<td>73</td>
</tr>
<tr>
<td>5-2</td>
<td>Ku-Band Radar Single-Hit Detector</td>
<td>76</td>
</tr>
<tr>
<td>5-3</td>
<td>Ku-Band Radar CFAR Detector</td>
<td>78</td>
</tr>
<tr>
<td>5-4</td>
<td>Passive GPC Search Mode Waveform</td>
<td>79</td>
</tr>
<tr>
<td>5-5</td>
<td>Passive Auto and Manual Search Mode Waveform</td>
<td>81</td>
</tr>
<tr>
<td>5-6</td>
<td>Search Mode Control Computer Algorithm (1 of 5)</td>
<td>85</td>
</tr>
<tr>
<td>5-6</td>
<td>Search Mode Control Computer Algorithm (2 of 5)</td>
<td>86</td>
</tr>
<tr>
<td>5-6</td>
<td>Search Mode Control Computer Algorithm (3 of 5)</td>
<td>87</td>
</tr>
<tr>
<td>5-6</td>
<td>Search Mode Control Computer Algorithm (4 of 5)</td>
<td>88</td>
</tr>
<tr>
<td>5-6</td>
<td>Search Mode Control Computer Algorithm (5 of 5)</td>
<td>89</td>
</tr>
<tr>
<td>5-7</td>
<td>Simplified Block Diagram of the Gimbal Pointing Loop Model</td>
<td>92</td>
</tr>
<tr>
<td>5-8</td>
<td>Antenna Gimbal Servo Loop Model</td>
<td>93</td>
</tr>
<tr>
<td>5-9</td>
<td>Discrete-Time Approximation of Antenna Gimbal Servo Loop Model</td>
<td>94</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5-10</td>
<td>Antenna Gimbal Pointing Loop Computer Algorithm</td>
<td>96</td>
</tr>
<tr>
<td>5-11</td>
<td>Antenna Obscuration Computation Algorithm</td>
<td>98</td>
</tr>
<tr>
<td>5-12</td>
<td>Antenna Obscuration Profile</td>
<td>99</td>
</tr>
<tr>
<td>5-13</td>
<td>Definition of Scan Rings.</td>
<td>101</td>
</tr>
<tr>
<td>5-14</td>
<td>Illustration of Antenna Boresight Ring Assignment Method</td>
<td>102</td>
</tr>
<tr>
<td>5-15</td>
<td>Scan Procedure Computer Algorithm</td>
<td>104</td>
</tr>
<tr>
<td>5-16</td>
<td>Boresight Ring Position as a Function of the Scan Time Parameter</td>
<td>105</td>
</tr>
<tr>
<td>5-17</td>
<td>Target Ring Location as a Function of Angle ($\theta_{SN}$) Off</td>
<td>107</td>
</tr>
<tr>
<td>5-18</td>
<td>Detection Model Computer Algorithm</td>
<td>109</td>
</tr>
<tr>
<td>5-19</td>
<td>CFAR Detection Model</td>
<td>111</td>
</tr>
<tr>
<td>5-20</td>
<td>Single-Hit Detection Model</td>
<td>113</td>
</tr>
<tr>
<td>5-21</td>
<td>Example of $P_D$ Versus SNR$_D$ Data</td>
<td>115</td>
</tr>
<tr>
<td>5-22</td>
<td>CFAR Detection Model Computer Algorithm</td>
<td>118</td>
</tr>
<tr>
<td>5-23</td>
<td>Single-Hit Detection Model Computer Algorithm</td>
<td>122</td>
</tr>
<tr>
<td>6-1</td>
<td>Outline of Track Mode Computer Algorithm</td>
<td>124</td>
</tr>
<tr>
<td>6-2</td>
<td>Waveform for Passive Track Modes</td>
<td>128</td>
</tr>
<tr>
<td>6-3</td>
<td>Waveform for Active Track Modes</td>
<td>131</td>
</tr>
<tr>
<td>6-4</td>
<td>Track Mode Control Computer Algorithm (1 of 2)</td>
<td>133</td>
</tr>
<tr>
<td>6-4</td>
<td>Track Mode Control Computer Algorithm (2 of 2)</td>
<td>134</td>
</tr>
<tr>
<td>6-5</td>
<td>Data Valid Flag Control Algorithm</td>
<td>136</td>
</tr>
<tr>
<td>6-6</td>
<td>System Initialization Algorithm</td>
<td>137</td>
</tr>
<tr>
<td>6-7</td>
<td>Tracking Loops Initialization Algorithm</td>
<td>139</td>
</tr>
</tbody>
</table>
### LIST OF ILLUSTRATIONS (continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-8</td>
<td>Simplified Diagram of Ku-Band Radar Track Mode Signal Processing</td>
<td>145</td>
</tr>
<tr>
<td>6-9</td>
<td>Track Mode Signal Processor Computer Model</td>
<td>146</td>
</tr>
<tr>
<td>6-10</td>
<td>Noise-Free Angle Discriminant Component Computation Model</td>
<td>151</td>
</tr>
<tr>
<td>6-11</td>
<td>Noise-Free Range Discriminant Component Computation Model</td>
<td>157</td>
</tr>
<tr>
<td>6-12</td>
<td>Track Mode Doppler Filter Configuration (Only Mainlobe Response Shown)</td>
<td>160</td>
</tr>
<tr>
<td>6-13</td>
<td>Noise-Free Velocity Discriminant Component Computation Model</td>
<td>162</td>
</tr>
<tr>
<td>6-14</td>
<td>Signal Generation and Processing Model Computer Algorithm (1 of 3)</td>
<td>165</td>
</tr>
<tr>
<td>6-14</td>
<td>Signal Generation and Processing Model Computer Algorithm (2 of 3)</td>
<td>166</td>
</tr>
<tr>
<td>6-14</td>
<td>Signal Generation and Processing Model Computer Algorithm (3 of 3)</td>
<td>167</td>
</tr>
<tr>
<td>6-15</td>
<td>Simplified Block Diagram of Break-Track Algorithm</td>
<td>176</td>
</tr>
<tr>
<td>6-16</td>
<td>Noise-Free Velocity Discriminant Frequency Response</td>
<td>177</td>
</tr>
<tr>
<td>6-17</td>
<td>Noise-Free On-Target Discriminant Frequency Response</td>
<td>178</td>
</tr>
<tr>
<td>6-18</td>
<td>No-Target Determination Algorithm</td>
<td>180</td>
</tr>
<tr>
<td>6-19</td>
<td>Break-Track Computer Algorithm</td>
<td>182</td>
</tr>
<tr>
<td>6-20</td>
<td>Simplified Diagram of Ku-Band Angle Rate and Angle Tracker</td>
<td>183</td>
</tr>
<tr>
<td>6-21</td>
<td>a-Loop Model</td>
<td>184</td>
</tr>
<tr>
<td>6-22</td>
<td>b-Loop Model</td>
<td>185</td>
</tr>
<tr>
<td>6-23</td>
<td>Discretized a-Loop</td>
<td>190</td>
</tr>
<tr>
<td>6-24</td>
<td>Discretized b-Loop</td>
<td>191</td>
</tr>
<tr>
<td>6-25</td>
<td>Discrete Representation of Integrator</td>
<td>192</td>
</tr>
<tr>
<td>6-26</td>
<td>Angle Discriminant Test Results</td>
<td>194</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>6-27</td>
<td>Angle Rate Loop Step Response (R &lt; 1.9 nmi)</td>
<td>196</td>
</tr>
<tr>
<td>6-28</td>
<td>Angle Rate Loop Step Response (1.9 nmi &lt; R &lt; 3.8 nmi)</td>
<td>197</td>
</tr>
<tr>
<td>6-29</td>
<td>Angle Rate Loop Step Response (3.8 nmi &lt; R &lt; 9.5 nmi)</td>
<td>198</td>
</tr>
<tr>
<td>6-30</td>
<td>Angle and Angle Rate Track Loop Filter Computer Algorithm</td>
<td>200</td>
</tr>
<tr>
<td>6-31</td>
<td>Simplified Diagram of Range and Range Rate Tracking Loop</td>
<td>203</td>
</tr>
<tr>
<td>6-32</td>
<td>Range Tracking Loop Discrete-Time Filter</td>
<td>205</td>
</tr>
<tr>
<td>6-33</td>
<td>Range Discriminant Test Results</td>
<td>208</td>
</tr>
<tr>
<td>6-34</td>
<td>Range Tracking Loop Transient Response for Ranges Less Than 0.42 NM</td>
<td>210</td>
</tr>
<tr>
<td>6-35</td>
<td>Range Tracking Loop Transient Response for Ranges 0.42 NM ≤ R &lt; 3.95 NM</td>
<td>211</td>
</tr>
<tr>
<td>6-36</td>
<td>Range Tracking Loop Transient Response for Ranges 0.95 ≤ R &lt; 3.8 NM</td>
<td>212</td>
</tr>
<tr>
<td>6-37</td>
<td>Range Tracking Loop Transient Response for Ranges 3.8 NM ≤ R ≤ 9.5 NM</td>
<td>213</td>
</tr>
<tr>
<td>6-38</td>
<td>Ku-Band Radar Velocity Processor</td>
<td>214</td>
</tr>
<tr>
<td>6-39</td>
<td>Simplified Diagram of Ambiguous Velocity Estimation Process</td>
<td>215</td>
</tr>
<tr>
<td>6-40</td>
<td>Simplified Diagram of Velocity Resolution Process</td>
<td>217</td>
</tr>
<tr>
<td>6-41</td>
<td>Filter Position Update Algorithm</td>
<td>219</td>
</tr>
<tr>
<td>6-42</td>
<td>Velocity Discriminant Test Results</td>
<td>220</td>
</tr>
<tr>
<td>6-43</td>
<td>Range and Range Rate Tracking Loop Computer Algorithm (1 of 2)</td>
<td>222</td>
</tr>
<tr>
<td>6-43</td>
<td>Range and Range Rate Tracking Loop Computer Algorithm (2 of 2)</td>
<td>223</td>
</tr>
<tr>
<td>6-44</td>
<td>Fractional Filter Width as a Function Velocity Discriminant Value</td>
<td>225</td>
</tr>
<tr>
<td>B-1</td>
<td>Definition of Target Roll and Pitch Angles</td>
<td>239</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>C-1</td>
<td>Simplified Diagram of the Noise-Free Discriminant Component Computation Model</td>
<td>244</td>
</tr>
<tr>
<td>C-2</td>
<td>Illustration of the Result of the Filtering, Sampling, and Range Gating Process</td>
<td>246</td>
</tr>
<tr>
<td>D-1</td>
<td>Illustration of Model Which Generates Noisy Discriminant Component</td>
<td>253</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>Radar Controls Required From Parent Simulation.</td>
<td>29</td>
</tr>
<tr>
<td>3-2</td>
<td>Radar Controls Required From Parent Simulation.</td>
<td>29</td>
</tr>
<tr>
<td>3-3</td>
<td>Other Inputs Required From the Parent Simulation.</td>
<td>31</td>
</tr>
<tr>
<td>3-4</td>
<td>Radar Simulation Output</td>
<td>34</td>
</tr>
<tr>
<td>3-4</td>
<td>Radar Simulation Outputs (continued)</td>
<td>37</td>
</tr>
<tr>
<td>3-6</td>
<td>Summary of Allowed Computation Time per Cycle for Each Simulator</td>
<td>40</td>
</tr>
<tr>
<td>3-7</td>
<td>Maximum Number of Allowed Target Points for Each Simulator</td>
<td>40</td>
</tr>
<tr>
<td>4-1</td>
<td>SPAS Point Scatterer Model.</td>
<td>56</td>
</tr>
<tr>
<td>4-1</td>
<td>SPAS Point Scatterer Model (continued)</td>
<td>57</td>
</tr>
<tr>
<td>5-1</td>
<td>Parameters for GPC Passive Search Modes</td>
<td>80</td>
</tr>
<tr>
<td>5-2</td>
<td>Parameters for Auto and Manual Passive Search Mode.</td>
<td>82</td>
</tr>
<tr>
<td>5-3</td>
<td>Scan Switch (From Outward to Inward Scan) Points in GPC-ACQ Mode.</td>
<td>84</td>
</tr>
<tr>
<td>6-1</td>
<td>Data Valid Flag Timeouts (After Closing Tracking Loops) for Active and Passive Modes</td>
<td>126</td>
</tr>
<tr>
<td>6-2</td>
<td>Waveform and Signal Processing Parameters for Passive Track Modes</td>
<td>130</td>
</tr>
<tr>
<td>6-3</td>
<td>Waveform and Signal Processing Parameters for Active Track Modes</td>
<td>132</td>
</tr>
<tr>
<td>6-4</td>
<td>Definition of Internal Control Parameters</td>
<td>143</td>
</tr>
<tr>
<td>6-5</td>
<td>Angle Tracking Loop Constants $f_n$ and $\tau$</td>
<td>186</td>
</tr>
<tr>
<td>Table</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6-6</td>
<td>Equivalent Angle Tracking Loop Constants $k_{eq}$ and $k^*$</td>
<td>188</td>
</tr>
<tr>
<td>6-7</td>
<td>Equivalent Range Tracking Loop Constants $m_a$ and $m_b^*$</td>
<td>206</td>
</tr>
</tbody>
</table>
INTRODUCTION AND OVERVIEW

1.1 INTRODUCTION

The objective of this program is the preparation of a real time computer simulation model of the Ku-Band Rendezvous Radar to be integrated into the Shuttle Mission Simulator (SMS), the Shuttle Engineering Simulator (SES), and the Shuttle Avionics Integration Laboratory (SAIL) simulator. Primary requirements of the simulation model are to provide crew training and to provide mission planners with representative predictions of the Ku-Band Radar tracking capability against selected candidate targets. The crew training requirement imposes the following design objectives with respect to the track and search modes:

1. to provide a real time simulation,
2. to provide accurate timing of discrete events appearing on the radar cockpit display,
3. to provide accurate operation of cockpit display meters,
4. to provide accurate responses to all cockpit radar controls.

In addition, the design objectives generated by a desire for accurate prediction of track mode operation against candidate targets are as follows:

5. to provide representative scattering models for all targets of interest,
6. to provide accurate processing of the target return signal,
7. to provide accurate models of all tracking loops.

Based upon our present knowledge of the capabilities of the three simulators, design goal (1) will conflict with design goals (5) through (7). Therefore, some sacrifices were made in target model accuracy and track signal processing accuracy to maintain a real time simulation. The sacrifices in track model accuracy and target scattering model accuracy and the performance limits they impose are discussed in detail in the sequel.

The development of the Ku-Band Rendezvous Radar performance computer model
that meets the requirements stated above has been divided into three tasks: (1) development of the radar tracking performance model, (2) development of the radar search and acquisition performance model, and (3) development of a target modeling method. This report documents the results obtained in these three areas. It includes:

- a detailed description of the parent simulation/radar simulation interface requirements,
- a detailed description of the method selected to model target scattering properties, including an application of this method to the SPAS spacecraft,
- a detailed description of the radar search and acquisition mode performance model.
- a detailed description and supporting analysis of the radar track mode signal processor model,
- a detailed description and supporting analysis of the angle, angle rate, range, and range rate tracking loops.

1.2 OVERVIEW OF RADAR PERFORMANCE COMPUTER MODEL

In all of the material that follows the reader's background knowledge of the Ku-Band Rendezvous Radar system is assumed to be on or above the level given in [1] or [2].

1.2.1 TARGET SCATTERING MODEL SUMMARY

Since virtually all target effects work (References 3-9) deals with point scatterer models, our approach is to represent the target as a collection of point scatterers. More specifically, this approach to modeling consists of:

- identifying strong scattering centers ("bright spots") and modeling them as point scatterers with associated cross section functions to express the angular variation,
- modeling intricate or rough-surfaced areas of the target as a random scatterer field, in turn, modeled by point scatterers with random amplitudes and specified angular variation functions.
It is remarked that these angular variation functions account for the shadowing effects due to a point scatterer's position relative to the other scatterers. Also these angular variation functions do not include the phasing terms given in the cross section literature. These factors are reflected in the spatial separation of the model's point scatterers. An example of this modeling method applied to the SPAS spacecraft is described in Section 4.0.

1.2.2 General Computer Model Structure

Figure 1-1 illustrates the general configuration of the computer model. It consists of three major parts: the executive program, the search and acquisition program, and the track program. The functions of the executive program are to initialize the system and target data when the program is first entered, to determine the system operating mode each update period and pass control to the appropriate subprogram, and to initialize the system appropriately when changes in the system controls have occurred. Search and track program details are summarized below.

1.2.3 Radar Search and Acquisition Performance Model Summary

An outline of the search and acquisition performance computer model is given in Figure 1-2. Main elements of this model are:

- antenna gimbal pointing loop model,
- scan model,
- detection model.

Antenna Gimbal Pointing Loop. The antenna \( \alpha \) and \( \beta \) gimbal pointing loops were both represented by the second order model shown in Figure 1-3. This model responds to (1) angle designates input from the General Purpose Computer (GPC) and (2) slew rate commands input by the crew from the cockpit. In the present configuration, the loop constants are chosen to give a loop damping factor \( \rho \) of 0.7 and a crossover frequency \( \omega_c \) of 1 hz.

Scan Model. This algorithm models radar system performance when a spiral antenna scan is in progress. The model is invoked by a search initiate command from either the GPC or the crew and operates as follows. It tracks the antenna position,
Figure 1-2. Outline of search and acquisition mode computer algorithm.
Figure 1-3 SIMPLIFIED BLOCK DIAGRAM OF THE GIMBAL POINTING LOOP MODEL
during the scan, to the nearest scan ring (see Figure 5-10) and tracks the target position exactly. It attempts detection if the target and the boresight are in the same scan ring in the present data cycle and were not in the same scan ring in the previous data cycle. The scan model continues in this manner until either the target is detected or an end-of-scan is reached.

Detection Model. This model contains a constant false alarm rate (CFAR) detection algorithm and a single-hit detection algorithm. These two models have the same fundamental construction which is shown in Figure 1-4 with the processing differences between the two detectors being absorbed in the SNR computation and SNR versus $P_D$ curves used in each case. The inaccuracies of these models occur in the beamshape and scan loss computations and in the target radar cross section value. More specifically, an average beamshape/scan loss value is used when the antenna is scanning and the beamshape loss at the beginning of the data cycle is used for the entire data cycle when the antenna is being slewed. The target cross section is inaccurate because it is modeled as a fixed, predetermined value independent of aspect angle.

1.2.4 Radar Tracking Performance Model Summary

Figure 1-5 gives a simplified illustration of the track mode computer model. This model is comprised of:

- a signal generation and processing model,
- a break-track algorithm,
- an angle and angle rate tracking model,
- a range tracking model,
- a velocity processor algorithm.

The key features of each of these models are summarized below.

Signal Generation and Processing Model. A simplified diagram of the computer model used to generate the target return signal, process this signal, and produce the discriminants for the tracking loops is shown in Figure 1-6. This model is based upon several assumptions about the system and the target motion. Of these, the ones
Figure 1-4  FUNDAMENTAL DETECTION MODEL CONFIGURATION

*SNR computed at doppler filter output for CFAR detector and at video filter output for single-hit detector.
I ENTER I TRACKING LOOP INITIALIZATION ALGORITHM

I TRACK MOOR INITIALIZATION CONTROL

I DATA VALID FLAG CONTROL ALGORITHM

I CONTROL ISIGNAL GENERATION AND PROCESSING

I MODEL SET APPROPRIATE CONTROL FLAGS

I BREAK-TRACK ALGORITHM

I ANGLE AND ANGLE RATE TRACKING LOOP MODEL

I RANGE AND RANGE RATE TRACKING LOOP MODEL

Figure 1-5. Outline of track mode computer algorithm.
Figure 1-6  TRACK MODE SIGNAL PROCESSOR COMPUTER MODEL

Return from
Point Target

Return from
Point Target

Appropriate Magnitude-Squared Data

NT
(Thermal Noise)

Discriminant Formation
that will have the most impact can be stated as follows:

- any radial acceleration of the point targets over a data cycle is ignored,
- the antenna does not move with respect to the target during the data cycle,
- the receiver's RF and IF electronics work perfectly, (i.e. no coupling loss, the down conversion is error-free, and the filters don't distort the return signal, but the receiver maintains the correct noise figure and noise bandwidth),
- quantization noise contributed by the signal processing chain from the A/D to the log converter is neglected,
- Automatic Gain Control (AGC) is not implemented.

A complete list of model assumptions and approximations is given in Section 6.4.1 and Appendix C. It is noted that this model also generates an estimate of the radar signal strength which is sent to the cockpit display. This value is taken as the SNR referenced to the video filter output and is very accurate for SNR > > 1, but will not be valid for SNR \( \lesssim 1 \).

**Break-Track Algorithm.** The computer model of the break-track algorithm is identical to the algorithm used in the Ku-Band Radar system. A simplified block diagram of this algorithm is given in Figure 1-7.

**Angle and Angle Rate Tracking Loop Model.** This model is used for estimating the target inertial roll and pitch rate and tracking the target roll and pitch angles in the GPC-ACQ and the Auto Track Modes. It consists of two tracking loops: one for each antenna gimbal. The basic loop model adopted for each gimbal is the second order loop shown in Figure 1-8 for the \( \alpha \) - loop. These loops are inertially stabilized, as required, and include the following error sources: target error effects (to the extent that the target scattering model is correct), thermal noise, boom deployment error, radar offset error, discriminant error, and gimbal bias error.

**Range Tracking Loop Model.** A simplified block diagram of the range tracking
Figure 1-7  SIMPLIFIED BLOCK DIAGRAM OF BREAK-TRACK ALGORITHM

Velocity Discriminant → No-Target Condition Determination For present Update period

→ No-Target Condition in 5 of last 8 Update periods?

If yes, break-track
If no, continue

On-Target Discriminant
Figure 1-8 ANGLE AND ANGLE RATE DISCRETE-TIME TRACKING LOOP MODEL

NOTE: The β-gimbal loop is similar in form to the α-gimbal loop shown above.
loop computer model is given in Figure 1-9. The loop filter equations and the loop constants for the model are identical to those used in the Ku-Band Radar system. Error sources incorporated into the model include target-effects, thermal noise, discriminant distortion, and a fixed average range bias error that accounts for unknown and time varying time delays.

**Velocity Processor Model.** The velocity processor computer model is shown in Figure 1-10. This model of the velocity processor is functionally identical to the algorithm used in the Ku-Band Radar system. That is, the equations, the logic and the number of bits of accuracy at each step are identical. Error sources modeled include target-effects, thermal noise, and discriminant distortion.

### 1.3 REPORT ORGANIZATION

The remainder of the report is organized in the following manner. In Section 2 all of the coordinate systems and the vector notation required for the description and analysis of the Ku-Band Rendezvous Radar simulation model are defined. In Section 3 the parent simulation/rendezvous radar simulation interface requirements are defined. Presented in this discussion are a definition of the data required from the parent simulation by the rendezvous radar simulation, the effects of different computer cycle times on rendezvous radar model tracking accuracies, and the effects of different allowed computing times per cycle on the point target model complexity. Section 4 gives complete details of the target modeling method. In Section 5, a detailed description of the radar search and acquisition performance model is presented and Section 6 gives a complete description plus supporting analysis of the radar tracking performance model.
Figure I-9  RANGE DISCRETE-TIME TRACKING LOOP MODEL
Figure -10  KU-BAND RADAR VELOCITY PROCESSOR

![Diagram of KU-band radar velocity processor]

- Return Signal Data
- Form Velocity/On Target Discriminants
- Update Doppler Filter Position
- Ambiguous Velocity Estimate
- Velocity Resolver
- Digital Smoothing Filter
- (From Range Tracker)
2. DEFINITION OF COORDINATE SYSTEMS AND VECTOR NOTATIONS

Since vectors, transformation operators, and a variety of coordinate systems pervade the description and analysis of the Ku-Band Radar performance computer model, we begin with definitions of all coordinate systems and vector and operator notation used in this report.

2.1 COORDINATE SYSTEM DEFINITIONS

In all, there are five coordinate systems that are useful in the description of the computer model. All of these coordinate systems have the following properties. Each reference frame is a right-handed coordinate system and positive rotation about a coordinate axis of a given frame is defined by the illustration in Figure 2-1.

Target(T) Frame. This coordinate system is defined to be fixed in the target. It will be most convenient to assume that the frame origin is coincident with the target c.g. and to choose an orientation that most easily accommodates the target description in the computer. Examples of possible target frame orientations for a multiple-point target are given in Figure 2-2.

Orbiter Body(B) Frame. Definition of this reference frame is the same as that given in [10]. The origin of this frame lies at the c.g. of the Shuttle Orbiter. Its x-axis lies along the body with the nose in the positive x-region and its y-axis lies along the wings with the right wing in the positive y-region. This reference frame is shown in Figure 2-3.

Radar(R) Frame. The Radar Frame origin is located at the B-frame coordinates (48, 11, -6), which corresponds to the center of the antenna gimbals. The x-y plane of the Radar frame is parallel to the x-y plane of the Body frame, but the x-y axes of the Radar frame are rotated with respect to the Body frame x-y axes by +67° about the z-axis. This arrangement is illustrated in Figure 2-4.
Figure 2-1. Definition of Positive Rotation about a Coordinate Axis.
Figure 2-2. Examples of Possible Target (T) Frame Orientations.
Figure 2-3. Orbiter Body (B) Frame Definition.
Figure 24. Radar (R) Frame Orientation with Respect to the Orbiter Body Frame.
Outer-Gimbal(G) Frame. This frame is fixed in the outer(or α) gimbal. Its origin is coincident with the Radar frame origin and its x-axis is coincident with the Radar frame x-axis. The y-z axes of the outer-gimbal frame are rotated by an amount α (variable) about the x-axis of the Radar frame. The angle α is measured from the minus z-axis of the Radar frame as shown in Figure 2-5.

Antenna LOS(L) Frame. This frame is fixed in the inner(or β) gimbal. Its origin is coincident with the G-frame origin and its y-axis is coincident with the G-frame y-axis. The x-z axes of this frame are rotated by an amount β (variable) about the y-axis of the G-frame. As shown in Figure 2-6 the angle β is measured from the minus z-axis of the G-frame. It should also be noted the z-axis of the antenna LOS frame is coincident with the antenna boresight.

Other Useful Frames. The only other useful frames for the present discussion are the Body, Radar, Outer-Gimbal, or Antenna LOS frames translated to the origin of the Target frame. These frames will be denoted by their usual letter and a zero subscript. For example, a frame centered at the target origin with its axes aligned with the antenna LOS frame will be denoted L₀.

2.2 DEFINITION OF VECTOR AND TRANSFORMATION NOTATION

In this subsection, the vector notation used to describe (1) a point scatterer's position and velocity measured in a given frame, (2) the target's inertial angular velocity, and (3) the orbiter's inertial angular velocity are defined. Also the notation for the various operations on these vectors is defined. We start with the vector description of a point scatterer's position and velocity. These are

\[ \mathbf{r}_K^{P} = \text{k th point scatterer position expressed in } P\text{-frame coordinates}. \]

and

\[ \mathbf{v}_K^{P} = \text{k th point scatterer velocity measured in the } P\text{-frame and expressed in } P\text{-frame coordinates}. \]
Figure 2-5. Outer Gimbal (G) Frame Orientation with Respect to the Radar Frame.
Figure 2-8. Antenna Los (L) Frame Orientation with Respect to the Outer Gimbals Frame.
where \( k = 1, 2, 3, \ldots, N \). Similarly, the vector description of the position and velocity of the target c.g. is given by

\[
\begin{align*}
\mathbf{r}_o^P &= \text{target c.g. position expressed in } P\text{-frame coordinates}, \\
\mathbf{v}_o^P &= \text{target c.g. velocity measured in the } P\text{-frame coordinates}.
\end{align*}
\]

and

\[
\begin{align*}
\mathbf{r}_o^k = & \quad \text{target c.g. position expressed in } P\text{-frame coordinates}, \\
\mathbf{v}_o^k = & \quad \text{target c.g. velocity measured in the } P\text{-frame coordinates},
\end{align*}
\]

where the subscript \( o \) will always be associated with the target c.g. The inertial angular velocity for the target and the orbiter are defined by the notation.

\[
\begin{align*}
\omega_T^P &= \text{inertial angular velocity of the target about a specified point expressed in } P\text{-frame coordinates.} \\
\omega_B^P &= \text{inertial angular velocity of the Shuttle Orbiter about a specified point expressed in } P\text{-frame coordinates.}
\end{align*}
\]

In component form, any of the above vectors can be expressed as a 3 x 1 column vector. For example,

\[
\begin{pmatrix}
\mathbf{r}_k^P \\
\mathbf{r}_k^P \\
\mathbf{r}_k^P
\end{pmatrix}
\]

where \( r_{kx}^P, r_{ky}^P, \) and \( r_{kz}^P \) are the components along the \( x, y, z \) \( P\)-frame axes, respectively. Also, it should be pointed out that if the reference frame under consideration is clear from the text, then the superscript will be dropped from the vector.

The next set of definitions describe the notation used for various vector operations of interest. A primary vector operation used throughout the development is the one that transforms a vector expressed in coordinate system \( A \) to a vector expressed in coordinate system \( B \) where \( A \) and \( B \) have the same origin.
This operation will be denoted $\mathbf{r}_{BA}$ and has the following features. Combining $T_{BA}$ with the vector notation from the previous paragraph, we obtain

$$\mathbf{r}_{X} = T_{BA} \mathbf{r}_{K}$$

Also, this transformation notation has the useful property that

$$T_{BA} = T_{CB} T_{BA}$$

There are two other vector operations that are of use in this report. They are the vector dot product, denoted by $\mathbf{a} \cdot \mathbf{b}$, and the vector cross-product, denoted by $\mathbf{a} \times \mathbf{b}$. These two products have the usual meaning.
1. RADAR SIMULATION/PARENT SIMULATION INTERFACE DESCRIPTION AND REQUIREMENTS

Development of the interface between the parent simulation and the Ku-Band Radar performance simulation is based upon the following assumptions:

(1) the amount of information passed across the interface should be kept to a minimum.

(2) the parent simulation (NASA) responsibilities are

- to define and generate all shuttle orbiter and target motion, including translational and rotational motion,
- to provide all cockpit and GPC radar control information to the radar simulation,
- to accept all radar tracking and status data generated by the radar simulation.

(3) the radar simulation (Hughes) responsibilities are

- to define the modeling method that best represents the scattering characteristics of all targets of interest,
- to generate the target return signal and process it during the tracking phase,
- to accept GPC and cockpit control information from the parent simulation,
- to provide target tracking data and radar status data to the parent simulation.

Assumption (1) was motivated by a desire to achieve integration of the radar performance simulation computer model into the three proposed parent simulations, the SMS, the SES, and the SAIL simulator, with relative ease. Assumptions (2) and (3) were partially generated from the following reasoning. All definitions of rendezvous missions, target trajectories, and orbiter trajectories fall under the heading of NASA expertise and, thus, these quantities should be provided by the parent simulation. However, definition of a target scattering model and
generation of radar return signals fall in the domain of Hughes expertise and should be provided by the radar simulation. In the following paragraphs, we shall define the radar/parent simulation interface, which is based upon the above assumptions, in detail.

3.1 **INPUT DATA REQUIRED FROM THE PARENT SIMULATION**

There are two types of input data required from the parent simulation. The first type is radar control data such as the desired operating mode and the target position designates that would normally be passed to the Ku-Band Radar over the modulation-demodulation (MDM) interface in the actual system. The second type of information required from the parent simulation is the data associated with target and orbiter motion, including both rotational and translational motions. This data is required to generate the target return signal and to simulate inertially stabilized tracking.

3.1.1 **Required Radar Controls**

Table 3-1 and Table 3-2 defines the radar control words required by the radar simulation that must be supplied by the parent simulation. In the actual hardware, each of the controls listed is sent to the radar either in discrete or serial word form through the MDM interface. It should be noted that the list of controls in Table 3-1 and Table 3-2 represents only those controls required in the search, acquisition, and tracking phases.

3.1.2 **Required Target/Orbiter Position and Motion Data**

All data associated with target and orbiter motion required by the radar simulation from the parent simulation is summarized in Table 3-3. A rationale for each of these data requirements is offered below.

In order to generate the target return signal as described in Section 4, the following information is required: (1) position of each point target and
<table>
<thead>
<tr>
<th>SYSTEM CONTROL FUNCTION</th>
<th>CONTROL NAME</th>
<th>CONTROL VALUE</th>
<th>CONTROL STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Power Switch</td>
<td>IPWR</td>
<td>1</td>
<td>Power Off</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>System On</td>
</tr>
<tr>
<td>System Mode Switch</td>
<td>IMODE</td>
<td>1</td>
<td>Radar Active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>Radar Passive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Communications</td>
</tr>
<tr>
<td>Transmitter Power</td>
<td>ITXP</td>
<td>1</td>
<td>High Power</td>
</tr>
<tr>
<td>Level Switch</td>
<td></td>
<td>2</td>
<td>Medium Power</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Low Power</td>
</tr>
<tr>
<td>Antenna Steering</td>
<td>IASM</td>
<td>1</td>
<td>GPC-ACQ</td>
</tr>
<tr>
<td>Mode Switch</td>
<td></td>
<td>2</td>
<td>GPC-DES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>Auto</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>Manual</td>
</tr>
<tr>
<td>Search Initiate Switch</td>
<td>ISRCHC</td>
<td>0</td>
<td>Inhibit Scan</td>
</tr>
<tr>
<td>(From Control Panel)</td>
<td></td>
<td>1</td>
<td>Enable Scan</td>
</tr>
<tr>
<td>Search Initiate Switch</td>
<td>ISRCHG</td>
<td>0</td>
<td>Inhibit Scan</td>
</tr>
<tr>
<td>(From GPC)</td>
<td></td>
<td>1</td>
<td>Enable Scan</td>
</tr>
<tr>
<td>Slew Antenna</td>
<td>IAZS</td>
<td>-1</td>
<td>Slew Left</td>
</tr>
<tr>
<td>Left/Right</td>
<td></td>
<td>0</td>
<td>No Slew</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Slew Right</td>
</tr>
<tr>
<td>Slew Antenna</td>
<td>IELS</td>
<td>-1</td>
<td>Slew Down</td>
</tr>
<tr>
<td>Up/Down</td>
<td></td>
<td>0</td>
<td>No Slew</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Slew Up</td>
</tr>
<tr>
<td>Antenna Slew Rate</td>
<td>ISLR</td>
<td>0</td>
<td>0.4 degrees/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>20.0 degrees/sec</td>
</tr>
</tbody>
</table>
### TABLE 3-2  RADAR CONTROLS REQUIRED FROM PARENT SIMULATION

<table>
<thead>
<tr>
<th>SYSTEM CONTROL FUNCTION</th>
<th>CONTROL NAME</th>
<th>CONTROL DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designated Target Range</td>
<td>EDRNG</td>
<td>Estimated Target Range From GPC</td>
<td>Feet</td>
</tr>
<tr>
<td>Designated Target Pitch Angle</td>
<td>EDPA</td>
<td>Estimated Target Pitch Angle From GPC</td>
<td>Degrees</td>
</tr>
<tr>
<td>Designated Target Roll Angle</td>
<td>EDRA</td>
<td>Estimated Target Roll Angle From GPC</td>
<td>Degrees</td>
</tr>
</tbody>
</table>
Table 3-3 OTHER INPUTS REQUIRED FROM THE PARENT SIMULATION

<table>
<thead>
<tr>
<th>INPUT</th>
<th>INPUT NAME</th>
<th>INPUT DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rightarrow_i r_0$</td>
<td>ERTO(I) I=1,2,3</td>
<td>Components of T-Frame Origin Position in B-frame</td>
<td>Feet</td>
</tr>
<tr>
<td>$\rightarrow_v v_0$</td>
<td>EVTO(I) I=1,2,3</td>
<td>Components of T-frame Origin Velocity Measured With Respect to B-frame and Expressed in B-frame Coordinates</td>
<td>Feet Per Second</td>
</tr>
<tr>
<td>$T_{B_0 T}$</td>
<td>TBT (I,J) I,J = 1,2,3</td>
<td>Elements of Transformation Matrix that aligns T-frame axes with B-frame axes.</td>
<td>No Units</td>
</tr>
<tr>
<td>$T_{B_0 T}$</td>
<td>TBTD (I,J) I,J=1,2,3</td>
<td>Elements of Matrix which is time derivative of $T_{B_0 T}$ Matrix</td>
<td>Seconds⁻¹</td>
</tr>
<tr>
<td>$W^B_W$</td>
<td>EWB(I) I=1,2,3</td>
<td>Orbiter inertial angular velocity expressed in B-frame Coordinates</td>
<td>Radians Per Second</td>
</tr>
</tbody>
</table>
(2) velocity of each point target as measured in the B-frame. (It should be pointed out that, ultimately, we want the point target position and velocity as measured in the L-frame but, since the radar simulation is tracking the antenna gimbal motion with respect to the B-frame, the radar simulation can easily perform the transformation from the B-to-L frame.) For the k th point target these data can be described as follows. Position of the k th scatterer at a fixed time t can be expressed as

\[ \mathbf{r}_k^B = \mathbf{r}_0^B + T_B^T \mathbf{r}_k^T \]

where Figure 3-1 illustrates the relation between these three vectors. Velocity of the k th scatterer as measured in the B-frame is given by

\[ \dot{\mathbf{r}}_k^B = \dot{\mathbf{r}}_0^B + T_B^T \dot{\mathbf{r}}_k^T \]

where the dot above a quantity represents time differentiation of that quantity. It is noted that equation (3.2) is obtained by time differentiating equation (3.1) and observing that \( \dot{\mathbf{r}}_k^T \) is fixed from the rigid lattice assumption (See Section 4). Since \( \dot{\mathbf{r}}_0^B \) and \( \dot{\mathbf{r}}_0^B \) are associated with target translational motion and since \( T_B^T \) and \( \dot{T}_B^T \) are associated with target rotational motion, they will be provided by the parent simulation under assumption (2). \( \dot{\mathbf{r}}_k^T \) is part of the target model definition and will be provided by the radar simulation under assumption (3).

Orbiter inertial angular velocity \( \dot{\omega}_B^B \) is required to perform tracking of the target inertial azimuth and elevation rates. The reason for this requirement is shown in Section 6.

3.2 OUTPUT DATA TO THE PARENT SIMULATION

All data output to the parent simulation are defined in Table 3-4. This data includes all cockpit radar display responses and the target tracking
Figure 3-1. Illustration of Orbiter - Point Target Geometry.
<table>
<thead>
<tr>
<th>OUTPUT DATA DESCRIPTION</th>
<th>OUTPUT NAME</th>
<th>OUTPUT VALUE</th>
<th>OUTPUT STATES</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan Warning Flag</td>
<td>MSWF</td>
<td>0</td>
<td>Scan Warning False</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Scan Warning True</td>
<td></td>
</tr>
<tr>
<td>Track Flag</td>
<td>MTF</td>
<td>0</td>
<td>Target Track False</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Target Track True</td>
<td></td>
</tr>
<tr>
<td>Search Flag</td>
<td>MSF</td>
<td>0</td>
<td>Target Search False</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Target Search True</td>
<td></td>
</tr>
<tr>
<td>Estimated Target Range</td>
<td>SRNG</td>
<td>Variable</td>
<td></td>
<td>Feet</td>
</tr>
<tr>
<td>Estimated Target Range Rate</td>
<td>SRDOT</td>
<td>Variable</td>
<td></td>
<td>Feet Per Second</td>
</tr>
<tr>
<td>Estimated Target Pitch Angle</td>
<td>SPANG</td>
<td>Variable</td>
<td></td>
<td>Degrees</td>
</tr>
<tr>
<td>Estimated Target Roll Angle</td>
<td>SRANG</td>
<td>Variable</td>
<td></td>
<td>Degrees</td>
</tr>
<tr>
<td>Estimated Target Pitch Rate</td>
<td>SPRTE</td>
<td>Variable</td>
<td></td>
<td>mrad Per Second</td>
</tr>
<tr>
<td>Estimated Target Roll Rate</td>
<td>SRRTE</td>
<td>Variable</td>
<td></td>
<td>mrad Per Second</td>
</tr>
<tr>
<td>Estimated Radar Signal Strength</td>
<td>SRSS</td>
<td>Variable</td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Angle Data Valid Flag</td>
<td>MADVF</td>
<td>0</td>
<td>Angle Data Invalid</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Angle Data valid</td>
<td></td>
</tr>
<tr>
<td>OUTPUT DATA DESCRIPTION</td>
<td>OUTPUT NAME</td>
<td>OUTPUT VALUE</td>
<td>OUTPUT STATES</td>
<td>UNITS</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
<td>--------------</td>
<td>--------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Angle Rate Data</td>
<td>MARDVF</td>
<td>0</td>
<td>Angle Rate Data Invalid</td>
<td></td>
</tr>
<tr>
<td>Valid Flag</td>
<td></td>
<td>1</td>
<td>Angle Rate Data Valid</td>
<td></td>
</tr>
<tr>
<td>Range Data</td>
<td>MRDVF</td>
<td>0</td>
<td>Range Data Invalid</td>
<td></td>
</tr>
<tr>
<td>Valid Flag</td>
<td></td>
<td>1</td>
<td>Range Data Valid</td>
<td></td>
</tr>
<tr>
<td>Range Rate Data</td>
<td>MRRDVF</td>
<td>0</td>
<td>Range Rate Data Invalid</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Range Rate Data Valid</td>
<td></td>
</tr>
</tbody>
</table>
data required by the guidance and navigation computer.

3.3 **INPUT/OUTPUT DATA FORMAT**

The technique used to pass data between the controlling program (parent simulation) and the subprogram (radar simulation) is to establish several labeled common storage areas. Labeled common is useful because it allows one to break a large common block into several smaller, independent common blocks which are distinguished by assigning them different labels. Thus, one can modify a section of common without having to perform bookkeeping on the whole array. Further information about labeled common can be found in [11].

In the development of the radar simulation the common block used for the interface between the two programs is divided into three parts. These are labeled: CNTL, INPUT, and OUTPUT. CNTL contains the radar control data required from the parent simulation and defined in Table 3-1 and Table 3-2. INPUT contains the target/orbiter motion data required from the parent simulation and defined in Table 3-3. OUTPUT contains the radar data output to the parent simulation and defined in Table 3-4.

3.4 **INTERFACE TIMING REQUIREMENTS**

Parent/Radar simulation interface timing involves (1) the length of the parent simulation update period called the (cycle time) and (2) the fraction of the period allotted to the radar simulation for computation of required radar outputs. The details of these two topics are summarized below.

3.4.1 **Simulation Cycle Time Requirements**

Table 3-5 summarizes the different update periods for the various Ku-Band Radar tracking modes and the update periods for the three simulators. These data show that the sample interval for each of the tracking modes differs from the update periods of the three parent simulators. This would imply that the radar discrete time tracking loops must operate in an asynchronous-fashion
<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>UPDATE INTERVAL, ms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ku-Band Radar</strong></td>
<td></td>
</tr>
<tr>
<td>7 khz PRF modes</td>
<td>51.</td>
</tr>
<tr>
<td>3 khz PRF modes</td>
<td>120.</td>
</tr>
<tr>
<td>268 hz PRF</td>
<td>250.</td>
</tr>
<tr>
<td><strong>Parent Simulators</strong></td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>TBS</td>
</tr>
<tr>
<td>SES (UNIVAC 1108)</td>
<td>200.</td>
</tr>
<tr>
<td>SAIL</td>
<td>TBS</td>
</tr>
</tbody>
</table>
with the parent simulator. However, rather than attempt this type of operation, the radar simulator is designed to run synchronously with the parent simulation. This means that the sample interval of the discrete time tracking loops will be an integral number of update periods of the simulation computer. Then the primary question is, what is the impact of this design decision on the tracking performance? Observe that the minimum update rate of the three simulators is approximately 4 Hz and the maximum loop bandwidth for any of the servos in any of the tracking modes is well under 1 Hz. Therefore the minimum sample rate of the computer is at least four to five times the tracking loop bandwidth and the fidelity of the loop response should not be affected. We offer an example to illustrate this point. Consider a target at a range of 0.4 nm (largest bandwidth) which is not moving at time \( t = 0 \) and is being tracked by the radar. At time \( t = 0 \), the target is given a step of 10 mrad/sec in roll rate with respect to the radar. The angle rate loop step response is then generated using update intervals of 50, 100, 200, and 400 milliseconds and plotted in Figure 3-2. These results show only slight error in the response for sample intervals as large as 200 ms.

3.4.2 Maximum Computation Time Requirements

Table 3-6 gives the computation time allotted to the radar simulation per cycle for each of the simulation computers. Assuming the present multiple point scatterer target model, these computation times can be converted to the maximum number of points allowed using empirically determined conversion factor. The maximum number of points and the conversion factors for each simulator are listed in the Table 3-7.
Figure 3-2. Example of Effects of Different Sample Intervals.

- RANGE = 0.4 NM
- $\dot{\mathbf{r}} = 0$ FPS
- PITCH ANGLE = 0 DEGREES

SAMPLE INTERVAL (MSEC)
- 50, 100
- 200
- 400

$\theta$, RAD/SEC

TIME, SEC.
Table 3-6  SUMMARY OF ALLOWED COMPUTATION TIME PER CYCLE
 FOR EACH SIMULATOR

<table>
<thead>
<tr>
<th>SIMULATOR</th>
<th>COMPUTATION TIME PER CYCLE, ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>TBS</td>
</tr>
<tr>
<td>SES (Univac 1108)</td>
<td>200.</td>
</tr>
<tr>
<td>SAIL</td>
<td>TBS</td>
</tr>
</tbody>
</table>

Table 3-7 MAXIMUM NUMBER OF ALLOWED TARGET POINTS FOR EACH SIMULATOR

<table>
<thead>
<tr>
<th>SIMULATOR</th>
<th>TIME PER TARGET, ms</th>
<th>ALLOWED COMPUTATION TIME, ms</th>
<th>MAXIMUM NUMBER OF POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>TBS</td>
<td>TBS</td>
<td>TBS</td>
</tr>
<tr>
<td>SES</td>
<td>5.7</td>
<td>200</td>
<td>35</td>
</tr>
<tr>
<td>SAIL</td>
<td>TBS</td>
<td>TBS</td>
<td>TBS</td>
</tr>
</tbody>
</table>
The purpose of target modeling is to predict target effects on the radar measurement accuracies. In this section, the general modeling approach is described, an example of the method applied to the SPAS spacecraft is provided, and a mathematical description of the resultant target return signal at the radar is given.

4.1 GENERAL APPROACH

As stated in the proposal [12], virtually all of the target effects analyses in the literature treat the target as a collection of point scatterers. This approach was adopted for the computer simulation described in this report. More specifically, our modeling method divides the spacecraft scatterers into two distinct classes: (1) those associated with simple geometric shapes and (2) those which are not. Simple shapes are modeled as point scatterers with the appropriate locations and their associated cross section functions to express the angular variation. (A review of the quantitative cross section results, taken from the literature, for several useful geometric shapes is provided in the next subsection.) Intricate or rough-surfaced areas of the spacecraft are modeled as random scatterer fields, which in turn are modeled by point scatterers with random amplitudes and specified angular variation functions. For both types of scatterers, the angular variation of the cross section amplitude includes the approximate effects of shadowing caused by neighboring elements. These cross-section functions do not include phasing. Instead, phase effects are accounted for via the spatial separation of the target's scatterers; this is shown in quantitative terms in section 4.4.

Details of the modeling method, especially the rough-surfaced modeling, are best illustrated by the SPAS modeling example of section 4.3.

4.2 SCATTERING CENTERS AND CROSS SECTIONS FOR SIMPLE AND REPRESENTATIVE SHAPES
The cross section literature can be used to extract point-scatterer models for simple geometric shapes, as follows.

4.2.1 Smoothly Curved Bodies (Reference 13)

A well-known result of the geometrical theory of diffraction is that the main RCS contribution from a curved surface comes from the "specular point" at which the radar line of sight (LOS) is normal to the surface. The cross section is

\[ \sigma = \pi R_1 R_2 \]

where \( R_1, R_2 \) are the surface's principal radii of curvature at the specular point. This principle is illustrated by the following examples.

**Sphere.** Here \( \sigma = \pi \alpha^2 \) where \( \alpha \) is the sphere's radius. The specular point lies on the sphere's surface at the intersection of the LOS.

**Hemispherical-Ended Cylinder.** The specular point is on the upper hemisphere when the LOS is from above. One has

\[ \sigma = \pi \alpha^2 \]

for all \( \theta \) except \( \theta = 90^\circ \); for the latter, the result for the cylinder (reference 14, p.9) yields

\[ \sigma = \frac{2\pi aL^2}{\lambda} = 291 \text{ } aL^2 \]

with dimensions in meters and \( \lambda = 0.0216 \) meters
The width of the "flare" at 90° can be taken to be \( \pm \frac{L}{L} = \pm 1.24 \text{ degrees} \).

The specular point lies on the intersection of the LOS with the cylinder's surface in the xy-plane.

**Toruspherical-Ended Cylinder.** In the toruspherical-ended cylinder, the ends consist of a section of large radius joined tangentially to a toroidal section that in turn is joined tangentially to the cylindrical section (See Figure 4–1).

Here we have

\[
\sigma = \frac{\pi a_0^2}{\sin \theta}, \quad 0 < |\theta| < \theta_o
\]

\[
= \pi a_1 \left[ \left( a_o - a_1 \right) \frac{\sin \theta}{\sin \theta_o} + a_1 \right], \quad \theta_o \leq |\theta| < 90^\circ
\]

\[
= \frac{2\pi a I^2}{\lambda}, \quad \theta = 90^\circ \pm \frac{1.24}{L} \text{ degrees}
\]

where we have used the results of Ref 13, p.114 for the toroid.

When the end is designed for maximum strength (everywhere equally stressed), as appears the case on the SPAS MOMS cannister,

\[
a_o = 2a_c
\]

\[
a_1 = a_c/3
\]

and

\[
\sin \theta_o = 0.4 \quad (\theta_o = 23.6^\circ)
\]

\[
\sigma = 4\pi a c^2, \quad |\theta| < \theta_o
\]

\[
= \frac{\pi a c^2}{9} \left[ \frac{2}{\sin \theta} + 1 \right], \quad \theta_o \leq |\theta| < 90^\circ
\]

\[
= \frac{2\pi ha}{\lambda}, \quad |\theta| = 90^\circ
\]
Figure 4.1. Toruspherical-Ended Cylinder Geometry.

\[ \sin \theta = \frac{a \cdot a_1}{a_2 \cdot a_1} \]
4.2.2 Other Shapes

Cylinders. Reference 14 provides cross sections for cylinders and discs. The flat-ended cylinder has three specular points at the intersection of the plane containing the LOS and the visible edges of the ends (Figure 4-2). The cross sections associated with these points are

\[ \sigma_1 = \frac{0.0046m^2}{\sin \theta} \left[ \frac{-1}{1 + 2 \cos \frac{2}{3}(\pi + \Theta)} \pm 1 \right]^2 \]

Note: \( \{ + \rightarrow \text{Vertical Polarization} \} \)

\[ \sigma_2 = \frac{0.0046m^2}{\sin \theta} \left[ \frac{-1}{1 + 2 \cos \frac{4\Theta}{3}} \pm 1 \right]^2 \]

\[ \sigma_3 = \frac{0.0046m^2}{\sin \theta} \left[ \frac{-1}{1 + 2 \cos (\pi - 2\Theta)} \pm 1 \right]^2 \]

These relations indicate negligible contributions except near normal incidence \((\Theta = 0^\circ, 90^\circ)\). For \(\Theta = 0^\circ\), one has

\[ \sigma_0 = 265,000 \ a^4 \quad |\Theta| < \frac{1.24^\circ}{2a} \]

And at \(\Theta = 90^\circ\), one has

\[ = 291 \ a^2 \quad |\Theta| = 90^\circ \pm \frac{1.24^\circ}{L} \]

Wire, Struts. A typical spacecraft has structural elements that are typically modeled as wires, i.e., long thin elements. Reference 13 (p.107) indicates that for a long thin wire (or edge) that

\[ \sigma = \frac{\lambda}{16\pi^3} \tan^2 \theta \cos^4 \phi \quad \theta < 90^\circ \]

\[ = 9.4 \times 10^{-7} \tan^2 \theta \cos^4 \phi \quad \text{m}^2 \]

\[ = \frac{L^2}{\pi} \cos^4 \phi \quad 45 \quad \theta = 90^\circ \]
Figure 4-2. Cylinder Geometry.
where $\phi$ is the angle of polarization incidence and $\theta$ is the angle between the LOS and the wire axis. Thus a significant contribution is seen only at broadside, reflecting the conclusions of reference 15 that edges don't provide significant RCS contributions.

**Corner Reflectors - Dihedrals.** The RCS for a dihedral reflector shown in Figure 4-3 is (Reference 16, p. 589)

$$\sigma = 16\pi a^2 b^2 \sin^2 \left( \frac{\pi}{4} + \phi \right)$$

at incidence perpendicular to the reflector axis and falls off rapidly away from normal.

**Corner Reflectors - Trihedrals.** Square trihedrals have cross section

$$\sigma = 4\pi \frac{A^2}{\lambda^3}$$

with $A$ the area normal to the LOS for which energy is redirected (Ref. 13, p.239), and for a square reflector,(Ref. 16, p.591)

$$\sigma = \frac{12\pi L^4}{\lambda^2} = 80,802 \frac{L^4}{\lambda^2} \text{ m}^2$$

with $L$ the width of each face. This RCS is maintained over a 23 degree cone about the symmetry axis. A 1-inch corner reflector thus has .033 m$^2$ cross section.

### 4.2.3 Reflector Antennas

On boresight, an antenna provides an enormous RCS. Let $G(\theta)$ be the antenna power gain pattern. Then

$$\sigma = \frac{\lambda^2 \rho G^2(\theta)}{4\pi}$$

where $\rho$ is the antenna power reflection coefficient, and usually approaches unity out of band. One has

$$G(\rho) = \eta \frac{4\pi A}{\lambda^2}$$

so

$$\sigma = \eta A \frac{4\pi \frac{A^2}{\lambda^2}}{G_N(\theta)}$$
Figure 4-3. Dihedral Corner Reflector Geometry.
where $G_n(\theta)$ is normalized to its maximum value. Taking $\phi = 1$, $\lambda = 0.0216$ m yields

$$\sigma = 26934 \Lambda^2 G^2_N(\theta)$$

or for a circular aperture of diameter $D$

$$\sigma = 16611 D^4 G^2_N(\theta)$$

and taking $n = 40\%$ yields

$$\sigma = 10774 \Lambda^2 G^2_N(\theta)$$

$$= 6645 D^4 G^2_N(\theta)$$

The width (first null) of this flare is about $\pm \lambda/D$ radians or $\pm 1.24/D$ degrees.

For a parabolic antenna, the reflector surface provides a significant return over a broad angle. For a body of revolution, Reference 13 gives

$$\sigma = \pi R_1 R_2$$

$$= \pi \left| \frac{\frac{d^2 x}{dz^2}}{\frac{x}{2} \sin^4 \theta} \right|$$

where the geometry is as shown in Figure 4.4. For the reflector

$$x = 2\sqrt{fz}$$

with $f$ the focal length; then one obtains

$$\sigma = \pi f^2 \cos^4 \theta$$

and for $f/D = .5$,

$$\sigma = \frac{\pi D^2}{4} \cos^4 \theta$$

$$= .785 D^2 \cos^4 \theta$$
Figure 4-4. Reflector Geometry.
This RCS contribution is seen so long as the LOS intersects the reflector at normal incidence somewhere. This occurs if

\[ |\theta| \leq \tan^{-1} \frac{dz}{dx} = \theta_o \]

\[ \theta_o = \tan^{-1} \frac{D}{4f} \]

\[ = 26.6 \text{ degrees for} \frac{f}{D} = 0.5 \]

4.3 SPAS MODEL

4.3.1 Satellite and its Coordinate System

Figure 4-5 shows the SPAS satellite in isometric view and identifies our coordinate system. Figure 4-6 shows a drawing of the satellite. Define the following angles:

\[ \phi_x = \text{Angle between LOS and x-axis} \]

\[ = \cos^{-1}(\hat{u}_L \cdot \hat{u}_x) \]

\[ \phi_y = \text{Angle between LOS and y-axis} \]

\[ = \cos^{-1}(\hat{u}_L \cdot \hat{u}_y) \]

\[ \phi_z = \cos^{-1}(\hat{u}_L \cdot \hat{u}_z) \]

where \( \hat{u}_x, \hat{u}_y, \hat{u}_z \) are unit vectors aligned with the x, y and z axes, and \( \hat{u}_L \) is a unit vector aligned with the LOS.

4.3.2 Scatterer Selection Strategy

Two classes of scatterers may be identified: those that arise due to geometric shapes discussed in Section 4.2, and those that do not. Among the former are tanks, experiment cannisters, mounting pallets, and the S-band antenna. Among the latter are complex areas such as are seen on the SPAS electronics pallets or structural areas, where multiple bounces and corner-reflector-like areas can give rise to significant and relatively orientation-free return. We model the former explicitly, and attempt to model the latter by associating point scatterers with the major complex areas, choosing the scatter cross section using a rough-surface model.
Figure 4.8. IPR ISOMETRIC VIEW
Figure 4-6. Drawing of SPAS Spacecraft.
4.3.3 **Point-Scatterer Model**

Table 4-1 lists the point scatterers that comprise the SPAS model. The angular region of applicability for each scatterer is indicated by the $\phi_x$, $\phi_y$, and $\phi_z$ columns and these entries provide an approximate inclusion of shadowing effects. Notes in calculating the cross sections are included as appendix E.

Scatterers 1 through 34 reflect geometries discussed previously. Specular flares due to plates have been limited to $700-1200m^2$ to reflect the fact that these surfaces are not usually good enough to provide the several thousand square meters predicted theoretically. Scatterers 35 through 54 are intended to model complex areas. The cross section for each area can only be guessed. The rationale for our guess is as follows. The area of each complex surface is about $0.5 m^2$. Taking a rough surface model (Models 9A4, 9A5 of Ref. 16, p.678) yields

$$\sigma = A \sigma_0$$

$$A = 0.5 m^2$$

$$\sigma_0 = \eta \cos \phi_{\perp}$$

$\eta$ = Backscatter coefficient

$\phi_{\perp}$ = incidence angle (angle from LOS to surface normal)

The constant $\eta$ has been determined experimentally for terrain and ranges from -30 to -15 dB for vegetation and ranges up to +10 or +20 dB for cultural areas. We take $\eta = -10$ dB to obtain $\sigma = 0.05m^2$ at normal incidence and allow the RCS to fall off as the cosine of the incidence angle. This value should be randomized to avoid interference effects.

At ranges for which the radar beam encompasses the target, modeling these areas as points still allows the radar's range and angle trackers to wonder over the target since variation in the relative phasing among scattering areas,
### Table 4-1: SPAS Point Scatterer Model

<table>
<thead>
<tr>
<th>Feature</th>
<th>XY Plane</th>
<th>Xk,m</th>
<th>Yk,m</th>
<th>Zk,m</th>
<th>r_k,m</th>
<th>$\phi_x$,deg.</th>
<th>$\phi_y$,deg.</th>
<th>$\phi_z$,deg.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tank</strong></td>
<td>1</td>
<td>2.6</td>
<td>.24</td>
<td>-0.83</td>
<td>0.15</td>
<td>-0.1</td>
<td>&lt;90</td>
<td>90 ±150</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.6</td>
<td>.24</td>
<td>-1.05</td>
<td>0.25</td>
<td>-0.1</td>
<td>&lt;90</td>
<td>90 ±150</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2.6</td>
<td>.24</td>
<td>-1.27</td>
<td>0.15</td>
<td>-0.1</td>
<td>&lt;90</td>
<td>90 ±150</td>
</tr>
<tr>
<td><strong>Cannister</strong></td>
<td>4</td>
<td>61</td>
<td>.37</td>
<td>1.05</td>
<td>0</td>
<td>-0.29</td>
<td>&lt;90</td>
<td>90-180</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>61</td>
<td>.37</td>
<td>0.35</td>
<td>0</td>
<td>-0.29</td>
<td>&lt;90</td>
<td>90-180</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>61</td>
<td>.37</td>
<td>0.35</td>
<td>0</td>
<td>-0.29</td>
<td>&lt;90</td>
<td>90-180</td>
</tr>
<tr>
<td><strong>Dome</strong></td>
<td>7</td>
<td>25.7</td>
<td>-0.35</td>
<td>-1.05</td>
<td>-0.8</td>
<td>-0.315</td>
<td>—</td>
<td>-145-145</td>
</tr>
<tr>
<td><strong>Plates</strong></td>
<td>8</td>
<td>13100</td>
<td>.12</td>
<td>1.9</td>
<td>0</td>
<td>0</td>
<td>&lt;2.1</td>
<td>90 ±150</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>13900</td>
<td>.12</td>
<td>-1.05</td>
<td>0</td>
<td>0</td>
<td>&lt;2.1</td>
<td>90 ±150</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>13100</td>
<td>.12</td>
<td>-1.8</td>
<td>0</td>
<td>0</td>
<td>&lt;2.1</td>
<td>90 ±150</td>
</tr>
<tr>
<td><strong>Viewed from +Y</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Viewed from +Y</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Viewed from -Y</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Comm Ant</strong></td>
<td>11</td>
<td>850</td>
<td>-0.3</td>
<td>2.0</td>
<td>-0.67</td>
<td>0</td>
<td>0</td>
<td>90 ±2.6</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>1200</td>
<td>-0.3</td>
<td>-2.0</td>
<td>-0.67</td>
<td>0</td>
<td>180 ±2.6</td>
<td>90 ±2.6</td>
</tr>
<tr>
<td><strong>Viewed from +Y</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Viewed from +X</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Viewed from -X</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Comm Ant</strong></td>
<td>13</td>
<td>3322</td>
<td>-0.35</td>
<td>-2.0</td>
<td>0</td>
<td>0</td>
<td>180 ±2.5</td>
<td>90 ±2.5</td>
</tr>
<tr>
<td><strong>Cylinder</strong></td>
<td>14</td>
<td>1117</td>
<td>-0.3</td>
<td>0</td>
<td>0.67</td>
<td>0.24</td>
<td>—</td>
<td>90 ±3</td>
</tr>
<tr>
<td><strong>Plates</strong></td>
<td>15</td>
<td>760</td>
<td>-0.35</td>
<td>1.7</td>
<td>-0.48</td>
<td>0</td>
<td>90 ±1.5</td>
<td>90 ±1.5,180 ±1.5</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>800</td>
<td>-0.35</td>
<td>1.05</td>
<td>-0.48</td>
<td>0</td>
<td>90 ±1.5</td>
<td>90 ±1.5,180 ±1.5</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>1000</td>
<td>-0.35</td>
<td>0.35</td>
<td>-0.48</td>
<td>0</td>
<td>90 ±1.5</td>
<td>90 ±1.5,180 ±1.5</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>900</td>
<td>-0.35</td>
<td>-0.35</td>
<td>-0.48</td>
<td>0</td>
<td>90 ±1.5</td>
<td>90 ±1.5,180 ±1.5</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>850</td>
<td>-0.35</td>
<td>-1.75</td>
<td>-0.48</td>
<td>0</td>
<td>90 ±1.5</td>
<td>90 ±1.5,180 ±1.5</td>
</tr>
<tr>
<td><strong>Cannister</strong></td>
<td>20</td>
<td>750</td>
<td>.37</td>
<td>1.05</td>
<td>.425</td>
<td>0</td>
<td>90 ±2</td>
<td>90 ±2</td>
</tr>
<tr>
<td><strong>Cannister</strong></td>
<td>21</td>
<td>850</td>
<td>.37</td>
<td>1.35</td>
<td>-0.425</td>
<td>0</td>
<td>90 ±2</td>
<td>90 ±2</td>
</tr>
<tr>
<td><strong>Cannister</strong></td>
<td>22</td>
<td>850</td>
<td>.37</td>
<td>.35</td>
<td>.425</td>
<td>0</td>
<td>90 ±2</td>
<td>90 ±2</td>
</tr>
<tr>
<td><strong>Cannister</strong></td>
<td>23</td>
<td>750</td>
<td>.37</td>
<td>.35</td>
<td>.425</td>
<td>0</td>
<td>90 ±2</td>
<td>90 ±2</td>
</tr>
<tr>
<td><strong>Cannister</strong></td>
<td>24</td>
<td>92</td>
<td>.37</td>
<td>-0.35</td>
<td>.425</td>
<td>0</td>
<td>90 ±2</td>
<td>90 ±2</td>
</tr>
<tr>
<td><strong>Cannister</strong></td>
<td>25</td>
<td>750</td>
<td>.37</td>
<td>-0.35</td>
<td>-0.425</td>
<td>0</td>
<td>90 ±2</td>
<td>90 ±2</td>
</tr>
</tbody>
</table>

56
| TABLE 4-1 SPAN POINT SCATTERER MODEL (Continued) |
|------------------|---|---|---|---|---|---|
| Wide-Angle Scatterers | | | | | | |
| Reflector | | | | | | |
| 26 | 0.2 | - .35 | -2.25 | 0 | 0 | 180 ± 26.6 |
| Task Hemispheres | | | | | | |
| 27 | .03 | .24 | - .83 | - .02 | -0.1 | 0 - 90 |
| 28 | .03 | .24 | - .01 | + .3 | -0.1 | 0 - 90 |
| 29 | .03 | .24 | -1.05 | - .02 | -0.1 | 0 - 90 |
| 30 | .03 | .24 | -1.05 | + .3 | -0.1 | 0 - 90 |
| 31 | .03 | .24 | -1.27 | - .02 | -0.1 | 0 - 90 |
| 32 | .03 | .24 | -1.27 | + .3 | -0.1 | 0 - 90 |
| Dome | | | | | | |
| 33 | 1.25 | - .35 | -1.05 | - .8 | - .35 | 156 - 180 |
| 34 | 0.17 | - .35 | -1.05 | - .8 | - .35 | 90 - 156 |
| Cold Gas Panel | | | | | | |
| 35 | 0.02 | - .72 | +1.75 | 0 | 0 | 0 - 90 |
| Data Handling Panel | | | | | | |
| 36 | 0.12 | -1.75 | 0 | 0 | 0 - 90 | |
| Power Panel | | | | | | |
| 37 | 0.12 | +1.75 | 0 | 0 | 0 - 90 | |
| +F Still Fitting Assy. | | | | | | |
| 38 | 0.38 | - .35 | +2.15 | 0 | 0 | 0 - 90 |
| Ant Assy 39 | | | | | | |
| 39 | 0.38 | - .35 | -2.15 | 0 | 0 | 90 - 180 |
| MOWS Module | | | | | | |
| 40 | 0.30 | - .35 | -1.75 | - .9 | 0 | |
| LISA-D Grapple Fixture | | | | | | |
| 41 | 0.24 | - .3 | 0 | 0 | 0 - 90 |
| RITA-D 42 | | | | | | |
| 42 | .34 | 0 | 0 | 0 | 0 | 0 - 90 |
| Structure | | | | | | |
| +2 43 | 0.45 | - .35 | 1.75 | 0 | 0 | 0 - 90 |
| +2 44 | 0.45 | - .35 | 1.05 | 0 | 0 | 0 - 90 |
| +2 45 | 0.45 | - .35 | .35 | 0 | 0 | 0 - 90 |
| +2 46 | 0.45 | - .35 | - .35 | 0 | 0 | 0 - 90 |
| +2 47 | 0.45 | - .35 | -1.05 | 0 | 0 | 0 - 90 |
| +2 48 | 0.45 | - .35 | -1.75 | 0 | 0 | 0 - 90 |
| -X 49 | 0.45 | - .35 | 1.75 | 0 | 0 | 90 - 180 |
| -X 50 | 0.50 | - .35 | 1.05 | 0 | 0 | 90 - 180 |
| -X 51 | 0.50 | - .35 | .35 | 0 | 0 | 90 - 180 |
| -X 52 | 0.50 | - .35 | - .35 | 0 | 0 | 90 - 180 |
| -X 53 | 0.50 | - .35 | 1.05 | 0 | 0 | 90 - 180 |
| -X 54 | 0.50 | - .35 | 1.75 | 0 | 0 | 90 - 180 |

57
that causes the wander, is modeled by the physical separation of multiple scattering areas.

At short ranges, the radar beam may encompass only one of these areas, and thus the wander effect will not be observed. Appendix F develops a simple model for wander that adds a "wander vector" to the scattering points given for the complex areas.

4.3.4 Effect of Thermal Blanket

Several, if not most, of the spacecraft will be wrapped by multi-layer insulation. The RF properties of this material are not known at present. If it is effectively conductive, it will tend to reduce flares and promote diffuse returns. The effects are almost impossible to predict analytically and measurements would be very desirable.

4.3.5 Recommendation

The validation of an analytical model of as complex an object as a spacecraft requires measurements. It would be very desirable if

a. Data can be taken with the Ku-Band system tracking a spacecraft - like target in the planned White Sands tests.

b. The RCS of a SPAS mockup could be measured with and without thermal blankets.

4.4 MATHEMATICAL DESCRIPTION OF TARGET RETURN SIGNAL

If we assume a single pulse was transmitted, then the expression for the noise-free return signal from a single point scatterer at the antenna sum (difference) channel output is given by

\[
S_k(t) = \sigma_k \rho_k A_k \cos \left[ 2\pi (f_c + f_k) (t - t_k) \right] p \left( \frac{t - t_k}{\tau} \right)
\]

(4.1)
where

\[ A_k = \left( \frac{R}{R_k} \right)^4 C_0 \]

\[ \sigma_k = \text{RCS of } k\text{th scatterer}, \]

\[ C_0 = \left[ \frac{P_T G_0^2 \lambda_c^2}{\left( 4 \pi \frac{3}{2} R_0^4 L_T \right)^2} \right] \]

\[ R_k = \text{Range of } k\text{th scatterer} \]

\[ R_o = \text{Range of target c.g.} \]

\[ \rho_k = \text{antenna sum (difference) pattern weighting normalized to the peak gain,} \]

\[ L_T = \text{transmit losses,} \]

\[ P_T = \text{Peak transmit power,} \]

\[ G_0 = \text{Peak one-way antenna gain,} \]

\[ \lambda_c = \text{wavelength of carrier frequency,} \]

\[ f_c = \text{carrier frequency,} \]

\[ f_k = -\frac{2v_k}{c} = \text{doppler shift of } k\text{th scatterer,} \]

\[ t_k = \frac{2(R_k - R_0)}{c} = \text{delay of target return relative to the target c.g. return,} \]

\[ c = \text{speed of light} \]

\[ \psi(\frac{t}{t_t}) = \begin{cases} 1, & 0 \leq t \leq t_t \\ 0, & \text{otherwise} \end{cases} \]

\[ t_t = \text{transmit pulsewidth} \]
Then, assuming the antenna is linear, by applying the principle of linear superposition the resultant return signal for the entire collection of point scatterers at the sum (difference) channel output terminal can be written as

\[
S(t) = \sum_{k=1}^{N} S_k(t)
\]

(4.1a)

where the target is composed of N point scatterers. A nice feature of the present target model hidden in equations (4.1) and (4.1a) is that this model easily handles the spatial integration of the return signal performed by the antenna.

In the rest of this subsection, additional details of the antenna weighting factor and scatterer phase computation models are given. Computation models for the other terms in equation (4.1) have either been explained earlier or the computation is clear from the definition of the term.

4.4.1 Antenna Weighting Factor Computation

Computation of the antenna sum and difference pattern weighting factors makes the assumption that the return signal from a single point target at the radar is a plane wave propagating from the direction of the scatterer. The sum and difference pattern weights can then easily be determined from the antenna sum and difference pattern models described below.

The sum pattern weighting is computed with the following expression

\[
\rho_s(\theta) = \frac{\sin x}{x} \quad \text{(sum pattern weighting)}
\]

(4.2)

where

\[
x = 93.80 \ \theta,
\]

\[
\theta = \text{target angle off boresight}.
\]

Figure 4-7 illustrates the pattern given by equation (4.2). This pattern has a 3 dB two-sided beamwidth of 1.7° and is assumed to be symmetric about the bore. For the k th target, the angle off boresight is computed with

\[
\theta_{ks} = \cos^{-1} \left( \frac{r_k^L}{|r_k^L|} \right).
\]

(4.3)
Figure 4-7. Antenna Sum Pattern.
The azimuth difference pattern weighting is computed from

\[ \delta_{\text{AZ}} = 1.1465 \left[ \frac{y \cos y - \sin y}{y^2} \right] \] (difference pattern weighting)

where \( y = 93.88 \). This pattern is assumed to be symmetric about the y-axis of the LOS frame and is illustrated in Figure 4-8. The angle \( \Delta \) for the \( k \)th target is obtained from

\[ \Delta = \phi_{kaz}^{-1} = -\sin \left( \frac{r_{ky}}{|r_k|} \right). \]

The elevation difference pattern weighting is also computed using equation (4.4) only in this case the angle \( \Delta \) is given by

\[ \Delta = \phi_{kel}^{-1} = \sin \left( \frac{r_{kx}}{|r_k|} \right). \]

and the pattern is assumed to be symmetric about the x-axis.

4.4.2 Computation of Scatterer Phase

From equation (4.1) the initial (\( t = 0 \)) phase associated with the \( k \)th scatterer is given by

\[ \phi_k = 2\pi (f_c + f_k) t_k. \]

If we choose the time origin appropriately, then \( f_k t_k < < 1 \) for all \( k \) and as a result

\[ \phi_k = 2\pi f_c t_k. \]

For example, the time origin can be located at the center of the range gate or
Figure 4-8. Antenna Difference Pattern.
at the leading edge of the return from the target c.g. which would give either

$$\phi_k = \frac{4\pi (R_k - R)}{\lambda c}$$

or

$$\phi_k = \frac{4 (R_k - R_0)}{c}$$

4.5 COMPUTER ALGORITHM DETAILS

Figure 4-9 illustrates the target scattering model computer algorithm. This algorithm computes the value of the RCS in the direction of the radar for each scatterer and the location of the scattering center for each scatterer. Using the modeling description given in sections 4.1 through 4.3 and adhering to the real-time computation constraint, the algorithm was structured as follows:

1. determine all scatterers with nonzero RCS in the direction of the radar,
2. determine the specular point location for those scatterers where the geometric optics approximation applies,
3. compute the RCS for all rough surface scattering areas that are illuminated,
4. if at close range, determine the scattering center for the rough surface (or diffuse) scattering area using the method presented in Appendix F.

Details of each of these steps are given in the remainder of this subsection.

The purpose of the first step is to weed out all of those scattering areas which are not illuminated or have, for all practical purposes, no RCS in the direction of the radar. Towards this end, we first compute the direction to the radar from each of the scattering centers using the expression
FIGURE 4-9 TARGET MODEL COMPUTER ALGORITHM (1 of 3)

1. ENTER
   - Perform Required Variable Initializations
   - Compute Unit Vector In Direction of Radar From ith Scatterer (eqn. 4.7)
   - ith Scatterer illuminated? (eqn. 4.8)
     - Yes: Add ith scatterer to Vector of illuminated Scatterers
     - No: Last Scatterer?
       - Yes: Compute Range to Radar in Target Frame (eqn 4.11)
       - No: Set Hysteresis Monitoring Variable (Relation 4.12)

2. Compute Location of kth illuminated scatterer (eqn 4.9)
   - Last Scatterer?
     - Yes: Compute RCS For kth illuminated diffuse scatterer (eqn 4.10)
     - No: Last Scatterer?
       - Yes: Compute Range to Radar in Target Frame (eqn 4.11)
       - No: Set Hysteresis Monitoring Variable (Relation 4.12)
In Short Range Interval? (Relation 4.13)

Yes

First Time Through?

Yes

Compute Angle Off Normal For kth diffuse Scatterer (eqn 4.14)

Compute Location Of kth diffuse Scattering Center (eqn 4.15)

No

Last Scatterer?

Yes

Compute Angle Off Normal for kth Diffuse Scatterer (eqn 4.14)

Compute Angular Change Over an Update Period For kth Scatterer.

Last Diffuse Scatterer?

Yes

Update Position Of kth diffuse Scattering Center (eqn 4.16)

Update Old Offset Vector:

\[ X_{old}(n+1) = X(n) \]

No

Last scatterer?

Yes
FIGURE 4-9 TARGET MODEL COMPUTER ALGORITHM (3 of 3)

4

Update Parameter used to monitor target position on Hysteresis Curve

EXIT
where \( \mathbf{x}_R^T \) = location of the radar in the target coordinate system.

The \( k \)th scatterer is declared to have a nonzero RCS in the direction of the radar if the components of the direction vector \( \hat{\mathbf{u}}_k^T \) satisfy the following inequalities

\[
(4.8) \quad m_{ki}^T \leq u_{ki}^T \leq M_{ki}^T \quad \text{i = x, y, z}
\]

where the \( m_{ki}^T \)'s and the \( M_{ki}^T \)'s are determined using the appropriate method outlined previously.

Step (2) of the algorithm is to determine the location of the specular point (or scattering center) for those scatterers where the geometric optics approximation applies and with nonzero RCS in the direction of the radar. For the SPAS scattering model, all of the specular scatterers have circular or spherical symmetry. In these cases, the specular point can easily be calculated from the simple expression

\[
(4.9) \quad \mathbf{s}_k^T = \mathbf{r}_k^T + a_k \hat{\mathbf{u}}_k^T \quad \text{for all } k
\]

where \( \mathbf{r}_k^T \) = location of the centroid of the simple shape in the target frame,

\( a_k \) = represents the appropriate radius for the \( k \)th scatterer.

It is remarked that for those scatterers where specular reflection does not apply, the \( a_k \) are set equal to zero.
The third step is to compute the RCS for all scatterers which were found in step (1) to have a nonzero RCS in the direction of the radar. Scatterers representing simple geometric shapes require no work since the model for these scatterers are assumed to have a constant RCS over the region where its theoretical RCS is significant and zero where it is not significant. However, the rough-surface scatterers require some calculation to obtain the proper RCS value. In section 4.3, this calculation was given as

\[(4.10) \quad \sigma_k = \eta_k \cos \phi_{ki}\]

where

- \(\eta_k\) = backscatter coefficient for the kth scatterer,
- \(\cos \phi_{ki} = \frac{\eta_k^T \cdot \eta_k}{\eta_k^T \cdot \eta_k}\),
- \(\eta_k^T = \text{normal to the kth rough surface scatterer.}\)

The fourth and final step is to determine whether the target is at close range (defined below) and, if it is at close range, to compute the position of the rough surface scattering center using the method of Appendix F. The idea is that one wants to avoid using a nonfluctuating scattering model when the target is close enough so that only one (rough surface) scatterer occupies the full 3 dB antenna beamwidth. Since all rough surface scatterers in the SPAS model have the same dimensions of 2.3 feet by 2.3 feet and the 3 dB beamwidth is taken to be 1.68 degrees, the criterion for closeness is easily computed to be a range of 78 feet. As an added measure of safety, the boundary for close range was established as approximately 300 feet. Also a hysteresis loop (shown below) is used so that the close range model is not swapped in
and out rapidly when the target range is jittering about the close range boundary.

To determine whether the short range model applies we first compute the range to the radar in the target frame using

\[ r_R(n) = \frac{\sqrt{x_R(n)^T x_R(n)}} {r(n)} \]

Next, the output of the hysteresis loop for the present update period is obtained from the following relations:

\[ r_R(n) \leq 290 + h(n) = 1 \]
\[ r_R(n) \geq 300 + h(n) = 0 \]

(4.12)

\[ 290 < r_R(n) < 300 \text{ and } h(n-1) = 1 \rightarrow h(n) = 1 \]
\[ 290 < r_R(n) < 300 \text{ and } h(n-1) = 0 \rightarrow h(n) = 0 \]

The short range model is invoked if

(4.13) \[ h(n) = 1. \]
If it has been determined from the above procedure that the short range model should be used, the computation of the "wander" in the rough surface scattering center is performed in the following manner. (To facilitate the explanation it is assumed that the normal to the kth rough surface scatterer is parallel to the z-axis of the target frame.) First, the incidence angle is computed with the expression

$$\phi_{ki}(n) = \cos^{-1}(u_k^T(n) \cdot \hat{z})$$

and is then used in the update of the components of the wander vector as follows

$$x_k(n) = \alpha(n) x_k(n-1) + \sigma_o \left[ 1 - \frac{2}{\alpha(n)} \right] \frac{1}{2} u \left[ -\frac{1}{2}, \frac{1}{2} \right]$$

where

$$\alpha(n) = \exp \left[ \frac{2D_x \delta \phi_{ki}(n) \cos \phi_{ki}(n)}{\lambda} \right]$$

$$\delta \phi_{ki}(n) = \phi_{ki}(n) - \phi_{ki}(n-1),$$

$$D_x = \text{length of the x-dimension of the rough surface scatterer},$$

$$\sigma_o^2 = \frac{D_x^2}{12 N_F}$$

and $$u \left[ -\frac{1}{2}, \frac{1}{2} \right]$$ represents a selection from a population which is uniformly distributed over the interval $$\left[ -\frac{1}{2}, \frac{1}{2} \right]$$. The y-component of the wander vector is obtained by replacing all x's by y's in equation (4.15).

The only detail that remains is the initialization of the difference equation given in (4.15), i.e. determining the value of $$x_k(o)$$ and $$y_k(o)$$, when the close range model is first invoked. This is accomplished by choosing the $$x_k(o)$$ and $$y_k(o)$$ from a random population with the appropriate statistics. Quantitatively, we have

$$x_k(o) = \sigma_o u \left[ -\frac{1}{2}, \frac{1}{2} \right].$$
5. **SEARCH AND ACQUISITION MODE COMPUTER MODEL DESCRIPTION**

As stated in the introduction, the search and acquisition mode performance computer model is provided for the purpose of crew training only which dictates the following design objectives:

1. to provide a real-time simulation,
2. to provide accurate timing of discrete events appearing on the cockpit radar display,
3. to provide accurate operation of cockpit radar display meters,
4. to provide accurate responses to all cockpit radar controls.

Since the model is not required for critical engineering evaluation of the Ku-Band Radar search mode performance, the design objectives above can be met while providing only a representative model of the target and detection processor.

Figure 5-1 illustrates the basic structure of the search and acquisition computer model. This model consists of a main control program and three major subprograms dedicated to (1) the gimbal pointing loop model, (2) the spiral scan model, and (3) the target detection model. The functions of the main program are to decide which antenna steering mode has been requested and then update the search sequence for that steering mode. Updating the search sequence requires a check of internal and external controls to determine which of the three models listed above should be invoked. In the remainder of this section the details of the main algorithm and the point, scan, and detection models will be presented.

Before launching a detailed description of the algorithm, we must state a fundamental assumption that was made in the development of the search and acquisition mode computer model: all of the acquisition mode logic was ignored since it is transparent to the crew. Impact of this assumption is to introduce some error into discrete event timing under certain conditions. For example, neglecting the mini-scan will cause a noticeable timing error.
Figure 5-1. Outline of search and acquisition mode computer algorithm.
5.1 SUMMARY OF KU-BAND RADAR SEARCH MODE OPERATION

5.1.1 General Antenna Steering Mode Operation

This subsection provides a brief description of the Ku-Band Radar search mode procedure for each antenna steering mode. For a given antenna steering mode, the general procedure is the same for active and passive targets; the only difference between active and passive are in the waveforms and processing as discussed in the sequel.

GPC-ACQ Search and Acquisition Mode. In this mode, the radar accepts angle designates from the GPC. The antenna then slews towards these designated angles and attempts detection once inside zone 0 (within 3° of the designated angles). If the antenna moves into zone I (within 0.3° of the designated angles) without a detection and the search initiate is low, the antenna stops at the designates, awaits new angle designates or a search initiate from the GPC, and still attempts detection. If the antenna is in zone I and the search initiate command has been given, the antenna begins scanning using a spiral pattern, centered at the inertially held target angle designates. The scan will last for 60 seconds or until a target has been detected, which ever comes first. If a detection does not occur, the antenna returns to the designated angles and awaits new designates or another search initiate command. If a detection occurs the system progresses to the acquisition mode where a mini-scan (if required) and a sidelobe avoidance test are performed. Depending upon the outcome, the system proceeds to the rack mode or returns to the search mode. Details of the acquisition mode are deliberately sketchy because this mode is not modeled as noted earlier.

GPC-DES Search and Acquisition Mode. Search operation in this antenna steering mode is identical to the GPC-ACQ mode minus the spiral scan capability. That is, the antenna only moves if it receives new angle designates from the GPC. Rules for when target detection is allowed are the same as GPC-ACQ and the waveforms and processing for active and passive operation are identical.
Auto Search and Acquisition Mode. In this mode the crew moves the antenna to the desired position using the antenna slew switches on the radar console. Using these slew switches, the antenna can be slewed up or down and left or right at either 20 degrees per second or 0.4 degrees per second. When the antenna is being manually slewed, target detection is only allowed if the slew rate is less than or equal to 0.4 degrees per second. Once the antenna has been slewed to the desired position and no target detection has occurred, the crew can initiate a spiral scan search. After a scan is initiated, the antenna will continue to spiral outwardly for one minute (to 30° off the body-stabilized scan center) or until a target is detected whichever comes first. If a target is not detected then the antenna returns directly to the scan center and awaits either a slew command or another search initiate command. If a target is detected the system proceeds to the acquisition mode.

Manual Search and Acquisition Mode. The manual search mode is identical to the Auto search mode minus the spiral scan capability. That is, the antenna position can only be changed via the slew switches on the radar console and target detection is only allowed if the commanded antenna slew rate is less than 0.4 degrees per second. The transmit waveforms and signal processing for this mode are identical to the Auto mode. Manual control of the antenna is also maintained during the acquisition and tracking phases.

5.1.2 Display Meters

The only meters that are operational during the search and acquisition mode are the roll and pitch angle meters. These meters monitor the antenna position during search and acquisition. All other meters, including the signal strength meter, are zeroed during this phase.

5.1.3 Search Mode Waveforms and Signal Processing.

Two types of detectors are used in the search mode: a single-hit detector shown in Figure 5-2 and a constant false alarm rate (CFAR) detector shown in
Figure 5-2  KU-BAND RADAR SINGLE-HIT DETECTOR

I channel

signal

Baseband Filter

A/D

Comparator

Hit/Miss

Fixed Threshold $T$

Q channel

signal

Baseband Filter

A/D

Comparator

Hit/Miss

Fixed Threshold $T$
Figure 5-3. The situations where these two detectors are used are summarized below.

**Passive GPC Modes.** The passive GPC modes use a single hit detector when the designated range is less than 0.42 nm, and the CFAR detector when the designated range is greater than 0.42 nm. In single-hit detection, returns from the first 3000 feet are processed through the hit detector. In CFAR detection two overlapped range gates centered at the target range designate are used to obtain a detection. (We note that the range gates are of width $3/2 \tau_{t}$ and overlapped by $\tau_{t}$ where $\tau_{t}$ is the transmit pulse width). Figure 5-4 gives the general waveform used for all designated ranges and Table 5-1 summarizes the waveform and processor parameters used at each designated range.

**Passive Auto and Manual Modes.** These modes use the relatively complex waveform shown in Figure 5-5. As noted in the figure, this waveform requires both types of detectors during an update period. That is, for a given transmit frequency the first three pulses are processed through the hit detector and the last 16 pulses are used in the CFAR detection process. In single-detection, returns from the first 3000 feet are processed through the hit detector. In CFAR detection, four juxtaposed range gates, of width $\tau_{t}$ and covering the interpulse period are used to obtain a target detection. Table 5-2 gives the waveform and processing parameters for these modes.

**All Active Modes.** Single-hit detection is employed in all active search modes. Only one transmit frequency is used, the PRF is fixed at 268 Hz, the transmit pulsewidth is 4.15 microseconds, and the sample interval is 2.075 microseconds under all conditions in the active mode. Target returns from up to 300 nm are processed through the single-hit detector. Also it is noted that the target range designate is ignored in the GPC active search modes.

5.1.4 **Antenna Scan Operation**

**GPC-ACQ Passive or Active Modes.** In this mode, the scan can only be
Figure 5-3  KU-BAND RADAR CFAR DETECTOR

*NOTE: Range gates are overlapped in GPC modes and are juxtaposed in Auto and Manual Modes.
Figure 5.4. Passive GPC Search Mode Waveform.
**TABLE 5-1  PARAMETERS FOR GPC PASSIVE SEARCH MODES**

<table>
<thead>
<tr>
<th>Range Interval NH</th>
<th>Frequency, GHz</th>
<th>PRF, Hz</th>
<th>Pulsewidth, μsec</th>
<th>Number of Range Bins</th>
<th>Range Bin Type</th>
<th>Sample Interval, μsec</th>
<th>Number of Samples Per Pulse</th>
<th>Detector Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 0.42</td>
<td>13.883</td>
<td>6969.7</td>
<td>0.122</td>
<td>50</td>
<td>Adjacent</td>
<td>0.122</td>
<td>1</td>
<td>Single-Hit</td>
</tr>
<tr>
<td>0.47 to 0.95</td>
<td>13.883</td>
<td>6969.7</td>
<td>4.15</td>
<td>2</td>
<td>Overlapped</td>
<td>2.075</td>
<td>2</td>
<td>CFAR</td>
</tr>
<tr>
<td>0.95 to 1.9</td>
<td>13.883</td>
<td>6969.7</td>
<td>8.3</td>
<td>2</td>
<td>Overlapped</td>
<td>2.075</td>
<td>4</td>
<td>CFAR</td>
</tr>
<tr>
<td>1.9 to 3.8</td>
<td>13.883</td>
<td>6969.7</td>
<td>16.6</td>
<td>2</td>
<td>Overlapped</td>
<td>2.075</td>
<td>8</td>
<td>CFAR</td>
</tr>
<tr>
<td>3.8 to 7.2</td>
<td>13.883</td>
<td>6969.7</td>
<td>33.2</td>
<td>2</td>
<td>Overlapped</td>
<td>2.075</td>
<td>16</td>
<td>CFAR</td>
</tr>
<tr>
<td>7.2</td>
<td>13.883</td>
<td>2987.0</td>
<td>66.4</td>
<td>2</td>
<td>Overlapped</td>
<td>2.075</td>
<td>32</td>
<td>CFAR</td>
</tr>
</tbody>
</table>

* These numbers correspond to frequency number 3 of a 5 frequency sequence. The complete sequence is given below.

<table>
<thead>
<tr>
<th>Transmit Frequency, GHz</th>
<th>PRF(“7000”) Hz</th>
<th>PRF(“3000”) Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.779</td>
<td>7017.4</td>
<td>3009.0</td>
</tr>
<tr>
<td>13.831</td>
<td>6993.5</td>
<td>2998.0</td>
</tr>
<tr>
<td>13.883</td>
<td>6969.7</td>
<td>2987.0</td>
</tr>
<tr>
<td>13.935</td>
<td>6946.1</td>
<td>2976.2</td>
</tr>
<tr>
<td>13.988</td>
<td>6822.6</td>
<td>2965.4</td>
</tr>
</tbody>
</table>
Figure 5-5. Passive Auto and Manual Search Mode Waveform.
### TABLE 5-2. PARAMETERS FOR AUTO AND MANUAL PASSIVE SEARCH MODE

<table>
<thead>
<tr>
<th>RANGE INTERVAL</th>
<th>FREQUENCY, GHz</th>
<th>PRF, Hz</th>
<th>PULSEWIDTH, μsec</th>
<th>NUMBER OF RANGE BINS</th>
<th>RANGE BIN TYPE</th>
<th>SAMPLE INTERVAL, μsec</th>
<th>NUMBER OF SAMPLES PER PULSE</th>
<th>DETECTOR TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 3000 ft</td>
<td>13.883</td>
<td>15,060</td>
<td>.122</td>
<td>50</td>
<td>Adjacent</td>
<td>.122</td>
<td>1</td>
<td>Single-Hit</td>
</tr>
<tr>
<td>3000 ft to 27 nm</td>
<td>13.883</td>
<td>2987.0</td>
<td>66.4</td>
<td>4</td>
<td>Adjacent</td>
<td>2.075</td>
<td>32</td>
<td>CFAR</td>
</tr>
</tbody>
</table>

* These numbers correspond to frequency number 3 in a 5 frequency sequence.

The complete sequence is given in Table 5-1.
commanded by a search initiate command from the GPC. The scan is centered at the angle designates received in the frame when the initiate command is given. The antenna begins executing the spiral scan pattern when the antenna has moved to within 0.3 degrees (Zone I) of these angle designates which are inertially held. Once the scan has been initiated the antenna spirals outwardly to a predetermined angle off the scan center, which depends on the target designated range, and begins to spiral inwardly. (These predetermined angles off scan center are called switch points and are summarized in Table 5-3). All scans will last 60 seconds or until a target is detected which ever comes first. It is also noted that the scan will terminate if the system mode or the antenna steering mode is changed.

Auto Passive or Active Modes. In this mode, the crew selects the scan center by slewing the antenna with the switches on the cockpit control panel. Once a scan center is selected, the crew initiates the spiral scan using the search initiate switch on the control panel. The scan pattern is the same in all situations. That is, the antenna spirals outwardly to 30 degrees off scan center and terminates. This procedure lasts for 60 seconds or until a target is detected whichever comes first.

5.2 SEARCH MODE CONTROL ALGORITHM DESCRIPTION

Figure 5-1 provides an outline of the overall structure of the computer implementation of the search mode. The mainstay of this computer model is the search mode control algorithm (enclosed in dashed lines in Figure 5-1). The control sequence is (1) determine the antenna steering mode, (2) update the search operation using the proper antenna steering mode sequence, and (3) set the appropriate flags based upon the outcome of step (2). Figure 5-6 gives the detailed computer algorithm (called SEARCH) used to accomplish this task. Basically, this algorithm is partitioned into four sections where each section of code is dedicated to the complete search procedure for one of the antenna steering modes: GPC-ACQ, GPJ-DES, Auto, or Manual. The computer code for each of these
<table>
<thead>
<tr>
<th>DESIGNATED RANGE, nm</th>
<th>SWITCH POINT, degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 8</td>
<td>Outward Scan Only (to 30°)</td>
</tr>
<tr>
<td>8 to 9.2</td>
<td>27.7</td>
</tr>
<tr>
<td>9.2 to 10.3</td>
<td>24.4</td>
</tr>
<tr>
<td>10.3 to 11.8</td>
<td>21.7</td>
</tr>
<tr>
<td>11.9 to 15</td>
<td>19.6</td>
</tr>
<tr>
<td>15 to 25</td>
<td>16.5</td>
</tr>
<tr>
<td>25 to 40</td>
<td>13.4</td>
</tr>
<tr>
<td>40 to 65</td>
<td>11.0</td>
</tr>
<tr>
<td>65 to 145</td>
<td>8.0</td>
</tr>
<tr>
<td>145 to 300</td>
<td>6.2</td>
</tr>
</tbody>
</table>
Figure 5-6  SEARCH MODE CONTROL COMPUTER ALGORITHM
(1 of 5)

ENTER

GPC-ACQ MODE?
yes → 2
no

GPC-DES MODE?
yes → 3
no

AUTO MODE?
yes → 4
no

5

Antenna Steering
Mode Determination
Figure 5-6  SEARCH MODE CONTROL COMPUTER ALGORITHM

(2 of 5)

2

Scanning

yes  Call scan Model  Exit

no

In Zone I and Search Initiate On?

yes  Initialize Scan Model  Exit

no

Search Initiate On?

no  Update Roll/Pitch References (From GPC)

yes  Call Gimbal Pointing Loop Model

Inside Zone O2

no  Exit

yes  Call Detection Model

Exit

GPC-ACQ STEERING MODE
CONTROL LOGIC
Figure 5-6 SEARCH MODE CONTROL COMPUTER ALGORITHM
(3 of 5)

3

Update Roll/ Pitch Designates (From GPC)

Inside Zone 0? no EXIT

yes

Call Detection model

Exit

GPC-DES STEERING
MODE CONTROL LOGIC
Figure 5-6  SEARCH MODE CONTROL COMPUTER ALGORITHM (4 of 5)

Call Gimbal Pointing Loop Model

Slew Rate \leq 0.4^\circ/\text{sec}?

Search Initiate On?

Update Roll/Pitch Reference Angles (Eqns 5.1)

Call Scan Model

Initialize Scan Model

exit
Figure 5-6 SEARCH MODE CONTROL COMPUTER ALGORITHM
(5 of 5)

1. Update Roll/Pitch Angles (Eqns 5.1)
2. Call Gimbal Pointing Loop Model
3. Slew Rate $\leq 0.4^\circ$/sec?
   - no: EXIT
   - yes: Call Detection Model
4. EXIT

MANUAL STEERING MODE
CONTROL LOGIC
sections closely mimics the operation summary given for the corresponding antenna steering mode in section 5.1.1 and, with the exception of the gimbal pointing loop reference computation, requires no further description.

The gimbal pointing loop reference computation is performed as follows. When the antenna is being slewed manually in either Auto or Manual, the roll and pitch references are updated using the expressions

\[
\text{Roll}_{\text{Ref}}(n) = \text{Roll}_{\text{Ref}}(n-1) + T_s \theta_{AZ}(n) \\
\text{Pitch}_{\text{Ref}}(n) = \text{Pitch}_{\text{Ref}}(n-1) + T_s \theta_{EL}(n)
\]

(5.1)

where

- \(T_s\) = update interval,
- \(\theta_{AZ}(n)\) = commanded roll rate at \(n\)th time sample,
- \(\theta_{EL}(n)\) = commanded pitch rate at \(n\)th time sample.

In the GPC modes, the gimbal pointing loop references are set equal to the present angle designates if the search initiate command is low. But if the search initiate is high the pointing loop references are maintained at the angle designate values obtained in the update period when the initiate went high.

5.3 GIMBAL POINTING LOOP MODEL DESCRIPTION

A computer model of the antenna gimbal pointing loop is included in the search model to provide reasonable fidelity in the antenna motion response to

(1) angle designates from the GPC during GPC-ACQ and GPC-DES search modes,

(2) slew commands from the console during Auto and Manual search modes,

(3) slew commands from the console during Manual track mode.

It is noted that the present model does not contain gimbal stops or a cable unwrap capability. A simplified block diagram of the complete antenna gimbal
pointing loop computer model is given in Figure 5-7. The description of this model is divided into two parts: (1) a definition of the basic servo loop model and (2) a detailed description of the computer algorithm which implements the process illustrated in Figure 5-7.

5.3.1 Basic Servo Loop Model Definition.

Both antenna gimbal servos were modeled using the second order loop shown in Figure 5-8. This choice for the servo loop model is based on the antenna servo simulation material presented at the March 1978 preliminary design review (PDR) [17], the description of the baseline antenna servo design given in [10] and [18], and discussions with Mr. J. C. Riles the antenna servo system designer for the Ku-Band Radar. Rationale for each of the basic model components is provided below. The first stage of the integration represents smoothing and shaping of the error signal and the second integration stage represents the effect of the gimbal. A limiter was placed between integration stages to represent the fact that the commanded gimbal rate is limited to 58 degrees per second in the hardware. Loop constants $k_g$ and $t_g$ are chosen to best approximate the characteristics of the real antenna gimbal response to slewing and designate commands. At the present time these constants are chosen to give a damping factor of 0.7 and a crossover frequency of 1.0 Hertz.

In order to present the servo model of Figure 5-8 on the computer, it is approximated by the discrete-time model shown in Figure 5-9. This discrete-time model can be described mathematically as follows. The first step is to compute the error signal and update the output of the first integrator using the equations

$$\dot{\alpha}_g(n+1) = \dot{\alpha}_g(n) + T_g k_g \varepsilon(n) \tag{5.2}$$

where $\dot{\alpha}_g(n) =$ smoothed $\alpha$-gimbal rate estimate at the $n$th time sample,
Figure 5.1. Simplified Block Diagram of the Gimbal Pointing Loop Model.
Figure 5-9 DISCRETE-TIME APPROXIMATION OF ANTENNA GIMBAL SERVO LOOP MODEL
\[ \varepsilon(n) = \alpha_{\text{Ref}}(n) - \alpha(n) = \text{error signal at } n\ \text{th time sample}, \]
\[ \alpha_{\text{Ref}}(n) = \alpha\text{-gimbal reference position at time sample } n, \]
\[ \alpha(n) = \alpha\text{-gimbal position at time sample } n, \]
\[ k_g = \text{loop constant}. \]

Next, the gimbal rate is updated by the expression
\[ (5.3) \quad \dot{\alpha}(n+1) = \dot{\alpha}_g(n) + k_g \varepsilon(n) \]
where \[ \dot{\alpha}(n+1) = \text{commanded gimbal rate at the } n+1\ \text{th time sample}, \]
\[ t_g = \text{loop constant}. \]

The effect of limiting the commanded gimbal rate is given by
\[ (5.4) \quad \dot{\alpha}(n+1) = \begin{cases} -58, & \text{if } \dot{\alpha}(n+1) \leq -58. \\ \dot{\alpha}(n+1), & \text{if } -58 \leq \dot{\alpha}(n+1) \leq 58. \\ +58, & \text{if } \dot{\alpha}(n+1) \geq 58. \end{cases} \]

Finally, the \( \alpha\)-gimbal position at the \((n+1)\) time sample is obtained from
\[ (5.5) \quad \alpha(n+1) = \varepsilon(n) + T_g \dot{\alpha}(n+1). \]

### 5.3.2 Computer Algorithm Details

A flow chart of the antenna gimbal pointing loop computer model is given in Figure 5-10. The required inputs for this model at each update are the desired roll and pitch reference angles. In the GPC modes these references are just the target angle designates and in Auto and Manual the references are obtained from equation (5.1). Using these new roll and pitch angle references, the algorithm updates the \( \alpha \) and \( \beta \) gimbal positions using the procedure outlined below.

The first step of the gimbal pointing loop algorithm is to transform the roll and pitch reference angles expressed in body coordinates to \( \alpha_{\text{Ref}} \)
Figure 5-10  ANTENNA GIMBAL POINTING LOOP COMPUTER ALGORITHM

ENTER

Transform Roll/Pitch Reference Angles to $a_{\text{Ref}}, \beta_{\text{Ref}}$ (Eqn 5.6)

Update Outer ($\alpha$) gimbal position. (Eqns 5.2-5.5)

Update Inner ($\beta$) gimbal position. (Eqns 5.2-5.5)

Check Obscuration Zone (See Figure 5-11)

Transform $\alpha/\beta$ to Roll/Pitch (Eqns 5.8)

Resolve Ambiguity in Roll/Pitch Angles (Eqns 5.7)

EXIT
and $\beta_{\text{Ref}}$ angles (or, equivalently, roll and pitch angles expressed in the radar frame). This transformation can be expressed as

\[
\alpha_{\text{Ref}} = \tan^{-1}\left[ \frac{S_{g\,p} + C_{g\,p\,S_{r\,p}}}{C_{r\,C_p}} \right]
\]

(5.6)

\[
\beta_{\text{Ref}} = \sin^{-1}\left[ C_{g\,p} - S_{g\,S_{r\,p}} \right]
\]

where 

- $g$ = 6/ degrees,
- $p$ = -pitch$_{\text{Ref}}$,
- $r$ = -roll$_{\text{Ref}}$,
- $C$ = cos,
- $S$ = sin.

Also, it is noted that this transformation is identical to that used by the radar, i.e. it ignores the radar offset from the orbiter c.g. and the boom deployment error. In the next step, the $\alpha$ and $\beta$ gimbal positions are updated using the servo-loop model given by equations (5.2) through (5.5).

Then it is determined whether the new $\alpha$ and $\beta$ values lie in the obscuration zone. This task is accomplished using the algorithm given in Figure 5-11 which can be described as follows. First, the $\alpha$, $\beta$ angle ambiguity is resolved using the relations

\[
-90 < \beta \leq 90
\]

(5.7)

\[
-180 < \alpha \leq 180.
\]

Then a scan warning is determined by comparing the unambiguous $\alpha$ and $\beta$ position to a map of the scan warning area shown in Figure 5-12 which was digitized to an accuracy of 5 degrees-by-5 degrees and stored in computer memory. If the comparison shows that $\alpha$, $\beta$ are in the obscuration zone, the scan warning flag is raised.

The final step in the gimbal pointing loop algorithm is to transform
Figure 5-11  ANTENNA OBSCURATION COMPUTATION ALGORITHM

1. Enter
2. Resolve $\alpha / \beta$ ambiguity (Eqs. 5.7)
3. Quantize position (to $5^\circ$ accuracy):
   
   $I_\alpha = \left(\frac{\alpha + 180}{5} + 1\right)$
   $I_\beta = \left(\frac{90 - \beta}{5} + 1\right)$

4. Determine value of $m(I_\alpha, I_\beta)$ from lookup table
5. Set warning flag
   - High --- If $m = 1$
   - Low --- If $m = 0$

6. Exit
Figure 5-12. Antenna Obscuration Profile.
\[ \alpha \text{ and } \beta \text{ to roll and pitch using the expressions} \]

\[
\begin{align*}
\text{Roll angle} &= -\tan^{-1} \left( \frac{S_{\alpha} + C_{\alpha} \frac{S_{\beta} + C_{\beta}}{C_{\alpha} C_{\beta}}} \right) \\
\text{Pitch angle} &= -\sin^{-1} \left( C_{\alpha} \frac{S_{\beta} + S_{\alpha} C_{\beta}}{C_{\alpha} C_{\beta}} \right).
\end{align*}
\]

These transformations are identical to the Ku-Band Radar transformations. Also any angle ambiguity is resolved using the convention given in (5.7).

5.4 SCAN MODEL DESCRIPTION.

The primary function of the scan model is to provide a simulation of search mode operation whenever the antenna is performing a spiral scan. Description of the scan model makes abundant use of a quantity called a "scan ring". We offer the following definition of this entity. It is noted that in reality the antenna traces out a spiral pattern about the scan center, however, we will approximate the spiral pattern as a set of concentric rings and label these rings as shown in Figure 5-13. With this definition in mind, the scan model can be described as follows.

When a spiral scan has been initiated from the console or the GPC, the model tracks the antenna boresight position to the nearest scan ring (see Figure 5-14) and tracks the target position exactly. A target detection is attempted only if the boresight and target are in the same ring in the present update period and were not in the same ring in the previous update period. If an attempt at detection is successful, the scan procedure is terminated and control is handed over to the track routines. If no detection is obtained then the scan procedure continues until another target detection is allowed or the scan is completed.

The main advantage of the model is that it offers reasonably accurate estimates of elapsed time from scan initiate to target detection or an end-of-scan condition for an arbitrary rendezvous situation. Maximum error in elapsed
Figure 5-13. Definition of Scan Rings.
Figure 5-14. Illustration of Antenna Boresight Ring Assignment Method.

<table>
<thead>
<tr>
<th>ACTUAL BORESIGHT POSITION</th>
<th>RING ASSIGNMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. 1</td>
<td>RING 4</td>
</tr>
<tr>
<td>NO. 2</td>
<td>RING 3</td>
</tr>
<tr>
<td>NO. 3</td>
<td>RING 4</td>
</tr>
</tbody>
</table>
time for any situation should be no worse than ± 2 seconds. It is noted that there are some deficiencies in the model too. These are that (1) there is no inertial stabilization of the scan center during a scan in the present version and (2) the target detection capability is highly inaccurate under certain target motion conditions. For example, when the target is moving radially with respect to the scan center.

5.4.1 Summary of Scan Operation

Rules and conditions for scan initiation and termination in the various antenna steering modes are identical to the Ku-Band Radar scan rules summarized in Section 5.1.4.

5.4.2 Computer Algorithm Details

Figure 5-15 gives a flow chart of the scan model computer algorithm. Step one of the procedure is to determine whether or not the scan has just been initiated. If the scan has just been initiated, then the scan model must be initialized. This procedure consists of raising the search flag and resetting the scan clock and other time parameters to zero. In step two the scan clock is updated and checked for an end-of-scan condition. If no end-of-scan is obtained, then we proceed to the next set of steps which involve determination of target and boresight positions in the scan area.

In the third step, the antenna boresight position in the scan area is resolved to the nearest scan ring. When no switch point is involved, i.e. the antenna only spirals outwardly to 30 degrees and stops, determination of the boresight scan ring location is done in the following way. Since the elapsed scan time is known, this value can be used to address a lookup table which contains the boresight scan ring position versus scan time profile shown in Figure 5-16. (Data for this curve was obtained from a detailed simulation of the antenna scan process written by Mr. J.C. Riles of Hughes.) For those modes where a switch
Figure 5-15  SCAN PROCEDURE COMPUTER ALGORITHM

**ENTER**

- **New Scan?**
  - no
    - **Update Scan Clocks**
  - yes
    - **Reset Scan Clocks To Scan**

**Determine Antenna Position to Nearest Scan Ring**

- **Target Ring Equal Antenna Ring?**
  - no
    - **EXIT**
  - yes
    - **Both in same Ring Last Update Period?**
      - no
        - **Model**
      - yes
        - **EXIT**
Figure 5-16. Bore sight ring position as a function of the scan time parameter ($T_\Delta$, eqn 5.9).
point is involved, the following assumption is used: at the switch point, the boresight begins to retrace the profile given in Figure 5-16. With this assumption, the boresight position can be determined in all possible scan cases using the profile of Figure 5-16 and defining the time parameter

\[
T_{\Delta}(n) = \begin{cases} 
T_{SN}(n-1) + T_s & t_{SN}(n-1) \leq t_{\text{switch}} \\
T_{SN}(n-1) - T_s & t_{SN}(n-1) > t_{\text{switch}}.
\end{cases}
\]

where \( t_{sn}(n) \) = elapsed scan time at n th sample time,
\( T_s \) = update interval,
\( t_{switch} \) = time at which switch occurs (measured from scan initiation).

The fourth step in the scan model procedure is to determine in which scan ring the target is located. To do this, we first compute the target’s angle off scan center, call it \( \theta_{SN} \), using the expression

\[
\theta_{SN}(n) = \cos^{-1}\left[ \frac{\hat{r}_o^L \cdot \hat{r}_s^L}{r_o^L} \right]
\]

where \( \hat{r}_o^L \) = unit vector in the direction of the target, c.g. expressed in L-coordinates,
\( \hat{r}_s^L \) = unit vector in the direction of the scan center expressed in L-coordinates.

This value of \( \theta_{SN}(n) \) is used to obtain the target's scan ring position from the scan ring number versus \( \theta_{SN} \) curve shown in Figure 5-17 (Data for this curve was also provided by Mr. J. C. Riles) and stored in the computer. It is noted that in practice only the ring transition points, denoted by \( \theta_i \), are stored in memory. Then, one determines the target scan ring location by choosing \( \theta_i \) such that \( \theta_i \leq \theta_{SN} \leq \theta_{i+1} \).
Figure 5-17. Target Ring Location as a Function of Angle ($\theta_{SN}$) off Scan Center.
In the final step, it is determined whether a detection attempt (using the detection model described in Section 5.5) should be made. A detection will be attempted if the target ring number and the boresight ring number satisfy the following two conditions:

1. they are equal in the present update period,
2. they were not equal in the previous update period.

If a target detection is attempted the following logic governs the outcome of the process. If the target is not detected the scan is continued. If the target is detected the scan is halted, the target present flag is raised (MTP=1) the search flag is lowered (MSF=0), and control is handed to the track subroutine.

It is emphasized that no acquisition mode operations are modeled, including the mini-scan sequence. It is also assumed that any detection is a mainlobe detection and the track mode is initialized accordingly.

5.5 DETECTION MODEL DESCRIPTION

The detection model provides a simulation of the detection process for each of the search modes. Figure 5-18 gives a simplified flow chart of the detection model computer algorithm. The basic model consists of two types of detectors, a CFAR model and a single-hit model, and some control logic that decides which operating parameters and detector types should be used. In the remainder of this subsection, the modeling assumptions are listed, the two detector models are described, and the computer algorithm details are provided.

5.5.1 Model Assumptions

The following assumptions were used in the development of both detector models:

1. the target is a point scatterer with a slowly fluctuating (Swerling II) RCS which has a fixed, predetermined average value for all aspect angles,
Figure 5-18 DETECTION MODEL COMPUTER ALGORITHM

ENTER

Compute Req'd Target Parameters (Eqns 5.12-5.15)

Active Mode?

Call Single-Hit Detection Model

EXIT

yes

no

GPC Mode & DES > 0.42 nm?

Call Single-Hit Detection Model

yes

Target Detection?

EXIT

no

yes

GPC Mode?

EXIT

no

Call CFAR Detection Model

EXIT
(2) the target radial velocity does not change over the data cycle,

(3) for all nonscanning modes, the beamshape loss obtained at the beginning of the data cycle is used for the entire data cycle.

(4) for all scanning modes, an average beamshape/scan loss, based on the target position in the scan pattern and computed using a simulation documented in [19], is used instead of computing the loss on a pulse-by-pulse basis.

In some cases, assumption (1) can have a significant impact upon fidelity of the detection process. However, if the fixed average RCS is chosen carefully, then this model should provide the crew with a reasonable feel for the target detection capability. Assumptions (2) and (3) are forced on us by the constraints of real-time operation. That is, the motion is only updated at the sample rate. We are stuck with assumption (4) because of the method selected to model the scan process and the real-time constraint.

5.5.2 CFAR Detection Model

Figure 5-19 illustrates the CFAR detection process. For a given target range and velocity, target position in the scan area, and average target RCS value, the basic idea of the procedure is as follows. First, the SNR at the doppler filter output is computed. (This value excludes beamshape/scan loss when scanning but includes beamshape loss when not scanning.) This value of $SNR_D$ is then used to determine the probability of detection $P_D$ from a precomputed $P_D$ versus $SNR_D$ curve which is stored in computer memory. The statistical character of the detection process is injected by selecting a number $x$ from a population which is uniformly distributed on $[0,1]$ and deciding the outcome of the detection process using the algorithm.
Figure 5-19  CFAR DETECTION MODEL

*Nominal Means Beamshape and Scan Loss is not included.
\[ X \leq P_D \text{ target detected} \]
\[ X > P_D \text{ target not detected.} \]

It is noted that the \( P_D \) versus \( SNR_D \) data implicitly contains the beamshape/scan loss in those cases when the antenna is scanning. Further details of the \( P_D \) curves are provided in Section 5.5.4.

5.5.3 Single-Hit Detection Model

A block diagram of the single-hit detection model is given in Figure 5-20. Fundamentally, this procedure is identical to the CFAR detection model. However, in this case the SNR at the video filter output is used to obtain the required probability of detection \( P_D \) from the proper \( P_D \) versus \( SNR_v \) data stored in a lookup table. Once the \( P_D \) is selected the rest of the procedure is identical to the CFAR procedure.

As in the CFAR case, the \( SNR_v \) computation excludes the beamshape/scan loss when scanning but includes the beamshape loss when not scanning. The \( P_D \) curves will implicitly contain the average beamshape/scan loss in the scanning cases.

5.5.4 Determination of \( P_D(SNR) \) Data

All of the \( P_D \) curves required by the detection algorithm were generated using a very accurate model of the Ku-Band Radar search mode processor, modified appropriately for each case. This simulation model, documented in Reference [19], can be described as follows. It contains a very accurate model of the signal processor and a slowly fluctuating (Swerling II-type) target model. The model also contains the capability for including an average scan loss in the \( P_D \) computation. This capability is too involved to describe here. Therefore the reader is referred to [19] for complete details of the average scan loss model. It is noted that the scan model uses the scan parameters, i.e. target dwell time and beam overlap, associated with the outer edge of the scan pattern for all cases, regardless of target position in the scan pattern. If time permits, the scan loss will be modeled more accurately.
Figure 5-20  SINGLE-HIT DETECTION MODEL

- Select Uniform Random Number from [0,1]
- Determine $P_D$
- Add Beamshape Loss - if not scanning
- Compute Nominal $\text{SNR}_V$
- Comparator
- $X > P_D$ Miss
- $X \leq P_D$ Hit
For a given set of conditions, the simulation model of [19] computes the SNR versus $P_D$ profile in the following manner. For a given nominal SNR$_v$ (nominal means ignore the beamshape/scan loss), the Monte Carlo technique is employed to determine the associated $P_D$. Then, performing this computation for a range of SNR$_v$ values gives the desired $P_D$ versus SNR data. We note that the range of SNR is always chosen so that the data points adequately described the curve from 5% to 95% $P_D$. Figure 5-21 gives an example of the resulting data for the GPC-ACQ CFAR scanning case.

As a side remark we note that the data generated by this model for the CFAR case implicitly contains an average beamshape/scan loss (if scanning), the losses associated with the magnitude detector, and the losses due to the thresholding technique. In the single-hit detection cases this data will contain the average beamshape/scan loss implicitly (if scanning).

The $P_D$ (SNR) data will be modeled on the computer in the following way. For all CFAR cases, $P_D$ data will be generated by the simulation described above for SNR$_D$ values ranging from 0 dB to 20 dB spaced at 1/2 dB increments. For any SNR$_D$ values above 20 dB, the $P_D$ is set equal to 1 and for values of SNR$_D$ below 0 dB, the $P_D$ is set equal to 0. For all single-hit cases, $P_D$ data will be generated for SNR$_v$ values ranging from -25 dB to -5 dB spaced at 1/2 dB increments. SNR$_v$ values outside this range are treated in the same manner as the CFAR case.

5.5.5 Computer Algorithm Details

The detection computer model consists of a set of three algorithms:

1. the control algorithm shown in Figure 5-18,
2. the CFAR detection algorithm shown in Figure 5-19, and
3. the single-hit detection algorithm shown in Figure 5-20. The control algorithm first computes the target parameters required by the detection algorithms and then it decides which detector algorithm should be called based on the operating mode. In the first step of the control algorithm, the point target range, the point target radial velocity with respect to the
Figure 5-21. Example of $P_D$ Versus $SNR_D$ Data
radar, and the angle off the boresight are computed as follows. The range is given by the expression

\[(5.12)\]  
\[\begin{align*}
\mathbf{r}_o^L &= |T_{LB} (\mathbf{r}_o^B - \mathbf{x}^B)| \\
\text{(Range)}
\end{align*}\]

where \(\mathbf{r}_o^B\) is provided by the parent simulation, \(\mathbf{x}^B\) is the fixed radar offset from the orbiter C.G. and \(T_{LB}\) is given by

\[
T_{LB} = \begin{pmatrix}
C\beta & 0 & -S\beta & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & C\alpha & S\alpha \\
S\beta & 0 & C\gamma & 0 & -S\alpha & C\alpha
\end{pmatrix}
\]

where \(\alpha, \beta = \) most recent positions of the antenna gimbals,  
\(\gamma = \) yaw angle of R-frame with respect to B-frame,  
\(C = \cos,\)  
\(S = \sin.\)

The radial velocity is computed using the equation

\[(5.14)\]  
\[\dot{\mathbf{r}}_o^L = \frac{\mathbf{v}_o^L}{\mathbf{r}_o^L} \cdot \mathbf{T}_{LB} \mathbf{r}_o^B + \mathbf{T}_{LB} \mathbf{r}_o^B, \quad \text{(Radial velocity)}\]

where \(\dot{\mathbf{r}}_o^L = \mathbf{T}_{LB} \dot{\mathbf{r}}_o^B + \mathbf{T}_{LB} \mathbf{r}_o^B,\) 
\[\mathbf{r}_o^L = \frac{\mathbf{r}_o^L}{|\mathbf{r}_o^L|}, \quad \text{(direction)}\]

and \(\dot{\mathbf{r}}_o^B\) is provided by the parent simulation. Lastly, the angle-off boresight is computed using

\[(5.15)\]  
\[\theta_S = \cos^{-1} \left( \frac{\mathbf{r}_{oz}^L}{|\mathbf{r}_o^L|} \right).\]

The second step of the detection control logic is to decide which detection model should be invoked. The rules for this decision are identical to those for Ku-Band Radar summarized in Section 5.1.3. Once this decision is made control is passed to the proper detection model to attempt a target detection.
CFAR Detection Algorithm. In the CFAR algorithm of Figure 5-22, the first step is to compute the SNR at the video filter output, neglecting the beam-shape and/or scan loss. In the sequel, this will be referred to as the nominal SNR. It is computed using the expression

\[
\text{SNR}_v = \frac{P_T G^2 \lambda_c^2}{(4\pi)^3 R^4 k L T B R T} \quad \text{(All Passive Modes)}
\]

where

- \(P_T\) = peak transmit power,
- \(G\) = one-way antenna gain,
- \(\lambda_c\) = carrier wavelength,
- \(\sigma\) = average radar cross section,
- \(R\) = target range
- \(k\) = Boltzmann's constant
- \(L_T\) = transmit losses,
- \(B_n\) = receiver noise bandwidth,
- \(T\) = system noise temperature.

The next step is to compute the net gain of the processor from the baseband filter output to the doppler filter output and to combine it with the nominal SNR\(_v\) to form the nominal SNR\(_D\). The net gain is comprised of (1) range gate loss, (2) net presum gain, and (3) net doppler filter gain. Each of these budget entries is expressed quantitatively below.

Of the three budget entries only the range gate loss differs for the GPC modes and the Auto and Manual Modes. Range gate loss for the GPC modes where overlapped range gates are used, includes the effects of misdesignation loss and widened range gates and is given by
Figure 5-22  CFAR DETECTION MODEL COMPUTER ALGORITHM

1. Compute nominal SNR_v (Eqn 5.16)

2. Scanning?
   - Yes → Compute Net Processor Gain (eqns 5.17 and 5.19-5.21)
   - No → Compute Beam-Shape Loss (eqns 5.15, 4.7)

3. GPC Modes?
   - Yes → Compute Net Processor Gain (eqns 5.17 and 5.19-5.21)
   - No → Compute Net Processor Gain (eqns 5.18-5.21)

4. Determine P_D (using appropriate data)

5. Select Uniform Random Number, X, from [0,1].

6. X ≤ P_D?
   - Yes → Set Target Present Flag
   - No → EXIT

EXIT
\[ RGL = \begin{cases} 2/3, & \text{if } x \leq 1/2 \\ 2/3 (3/2-x)^2, & \text{if } 1/2 \leq x < 3/2 \\ 0, & \text{if } x > 3/2 \end{cases} \]  

(Range Gate Loss --- GPC Modes)

where 
\[ x = \frac{2 |R_G - R_o|}{ct}, \]

\( R_G \) = target designated range (center of gates),
\( c \) = speed of light,
\( t \) = transmitted pulse width.

For the Auto and Manual modes, where the range gates are juxtaposed, this loss is given by
\[ RGL = \begin{cases} X^2, & \text{if } X < 1 \\ (1 - (X-[X]))^2, & \text{if } 1 < X < 9/2 \text{ and } X-[X] < 1/2 \\ (X-[X])^2, & \text{if } 1 < X < 9/2 \text{ and } X-[X] > 1/2 \end{cases} \]

(Range Gate Loss --- Auto and Manual Modes)

where 
\[ X = \frac{2 R_o}{ct} \]

\([\cdot]\) = greatest integer in \(\cdot\).

The net presum gain computation is identical for all antenna steering modes and includes the coherent gain of the presumming process and the loss due to doppler mismatch. This value is computed using the expression
\[ PSG = \frac{\sin^2(N_p y)}{\sin^2(y) N_p} \]

(Net presum gain)

where 
\[ y = \pi f_d r_o \]
\[ f_d = \text{target doppler shift} = \frac{2 R_o}{\lambda c} \]
\[ r_s = \text{A/D sample interval,} \]
\[ N_p = \text{number of A/D samples per pulse width.} \]

The net doppler filter gain computation is identical for all antenna steering modes and includes the coherent gain of the doppler filter and the loss caused by filter straddling. It is computed from the expression

\[
(5.20) \quad DFG = \frac{\sin^2(16z)}{16 \sin^2(z)} \quad \text{(Doppler Filter Gain)}
\]

where
\[ z = r \left( \frac{m}{32} - f_d t_p \right), \]
\[ t_p = \text{PRI}, \]
\[ m = \text{filter nearest } f_d. \]

The net processor gain is obtained from

\[
(5.21) \quad \text{Net Processor Gain} = (RGL)(PSG)(DFG)
\]

in all cases where a CFAR detector is required.

The third step is to decide whether the antenna is scanning or not. If the antenna is scanning we proceed to step four. If the antenna is not scanning the beamshape loss is computed, using equations (5.15) and (4.2), and combined with the nominal SNR_D.

In the fourth step, the value of SNR_D is used to address the look-up table, containing the proper \( P_D \) profile for the given conditions, to determine the \( P_D \) value in the present case. It is noted that if the computed nominal SNR_D falls between stored data points then linear (in dB) interpolation is used to obtain the \( P_D \). This value of the \( P_D \) will implicitly contain an average beamshape and scan loss (if scanning), the losses associated with the magnitude detector, and the losses due to the thresholding technique.
The fifth and final step is to select a number $x$ from a population which is uniformly distributed on the interval $[0,1]$, compare this value of $x$ with the $P_D$ from step (4) and make a target detection/no detection decision based on the algorithm of (5.11).

**Single-Hit Detection Algorithm.** The single-hit detection algorithm of Figure 5-23 is similar in form to the CFAR algorithm. That is, first the nominal SNR$_v$ is computed using equation (5.16) for the passive modes or the expression

$$
(5.22) \quad \text{SNR}_v = \frac{P_B G_B \lambda^2}{(4\pi)^2 R^2 L_B k_B T}
$$

where

- $P_B$ = Peak beacon transmit power,
- $G_B$ = One-way gain of the beacon antenna,
- $L_B$ = Beacon transmit losses.

for the active modes. In the second step it is determined whether the antenna is scanning or not. If the antenna is scanning we proceed to step (3), but if it is not scanning then the beamshape loss is computed in the same manner as the CFAR case and combined with the nominal SNR$_v$. The fourth step is to determine the $P_D$ associated with present value of SNR$_v$. This determination is identical to the CFAR case. The last step is to select a uniform random number $x$, compare it to $P_D$, and decide hit or miss with the algorithm of (5.11) as in the CFAR case.
Figure 5-23  SINGLE-HIT DETECTION MODEL COMPUTER ALGORITHM

ENTER

ACTIVE?

yes

Compute Nominal SNR_v
(Eqn 5.22)

no

no

Compute Beamshape Loss
(Eqn 5.15, 4.7)

yes

Scanning?

Determine P_D
(Using Appropriate data)

Select Uniform Random Number, x, From [0,1]

X \leq P_D?

yes

Set Target Present Flag

EXIT

no

EXIT
5. TRACK MODE COMPUTER MODEL DESCRIPTION

An illustration showing the basic structure of the track mode computer algorithm is given in Figure 6-1. The structure of this algorithm is similar in form to the search mode computer algorithm of Figure 5-1. It consists of a main control program and several subprograms dedicated to various tracking functions. The main program, called the track mode control algorithm, has two purposes:
(1) it controls initialization of all tracking loops and updating of the status of all data valid flags when control is first passed from search to track and
(2) it controls the computation sequence required to update all tracking loops during the tracking phase. The initialization procedure requires two major subprograms. One is dedicated to initialization of the tracking loops when tracking first starts and the other subprogram decides when the various data valid flags should be raised, indicating the track estimates are accurate. The track loop update procedure requires several major subprograms which perform the following tasks: (1) target return signal generation and processing, (2) break-track determination, (3) angle and angle rate estimate updates, and (4) range and range rate estimate updates. In the following subsections, models for each of these functions are described, analysis is supplied whenever appropriate, and details of the computer algorithm for each function are given.

6.1 SUMMARY OF KU-BAND RADAR TRACK MODE OPERATION

6.1.1 General Antenna Steering Mode Operation

In this subsection, a short description of the Ku-Band Radar tracking procedure for each antenna steering mode is provided. For each steering mode, the general procedure is the same for active or passive targets. The only difference between active and passive modes are the transmit waveforms as discussed below.

GPC-ACQ Track Mode. In GPC-ACQ the radar performs target angle, inertial angle rate, range and range rate tracking. Angle and inertial angle rate tracking are accomplished using the amplitude comparison monopulse technique.
Figure 6-1. Outline of track mode computer algorithm.
Range tracking is achieved by maintaining the return pulse centered in two juxtaposed range gates. Velocity tracking is performed using the algorithm described in section 6.7.

**GPC-DES Track Mode.** In this mode the radar tracks the target range and range rate only. Target angle tracking is performed by the GPC which supplies target angle designates to the gimbal pointing loop during tracking. There is no tracking of the target's inertial angle rate. Range and range rate tracking are identical to the GPC-ACQ mode.

**Auto Track Mode.** This mode is identical to the GPC-ACQ mode.

**Manual Track Mode.** In this mode, the radar performs range and range rate tracking using the same method as the other three steering modes. Angle tracking is performed by the crew using the antenna slew switches on the cockpit radar console. There is no inertial angle rate tracking in this mode.

### 6.1.2 Data Valid Flags

The data valid flag representing a given quantity will be raised when all transients in the loop tracking that quantity have settled out. The time allotted from tracker initialization to raising the data valid flag is precomputed based on maximum allowable errors in the quantities tracked and linearized loop models. Precomputed times for the angle, angle rate, range, and range rate data valid flags as a function of the loop bandwidth are summarized in Table 6-1 for active and passive operation.

The only data valid flag that is allowed to drop during tracking, without a break-track condition is the range rate data valid flag. Conditions under which this flag is lowered are (1) when the PRF is switched from 7 kHz to 268 Hz in the active track mode or (2) when the predicted target velocity moves to a new filter in two out of the last five update periods, including the present update period. In either of these cases it is raised again if the predicted velocity remains in the same doppler filter for 15 consecutive update periods.

### 6.1.3 Display Meters

Display meters are provided on the cockpit radar console for
Table 6-1  DATA VALID FLAG TIMEOUTS (AFTER CLOSING TRACKING LOOPS) FOR ACTIVE AND PASSIVE MODES

<table>
<thead>
<tr>
<th>RANGE INTERVAL, nm</th>
<th>DATA VALID FLAG TIMEOUT, SECONDS</th>
<th>ANGLE</th>
<th>ANGLE RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RANGE &amp; RANGE RATE</td>
<td>ANGLE</td>
<td>ANGLE RATE</td>
</tr>
<tr>
<td>Passive Modes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R &lt; 3.8</td>
<td>6.97</td>
<td>1.02</td>
<td>8.2</td>
</tr>
<tr>
<td>3.8 &lt; R &lt; 7.2</td>
<td>6.97</td>
<td>1.02</td>
<td>26.23</td>
</tr>
<tr>
<td>7.2 &lt; R</td>
<td>29.76</td>
<td>2.33</td>
<td>29.76</td>
</tr>
<tr>
<td>Active Modes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R &lt; 9.5</td>
<td>6.15</td>
<td>1.02</td>
<td>8.20</td>
</tr>
<tr>
<td>9.5 &lt; R</td>
<td>28.69</td>
<td>5.12</td>
<td>28.69</td>
</tr>
</tbody>
</table>
The target signal strength meter which is zeroed during search and acquisition immediately becomes operational when the track mode is entered regardless of the antenna steering mode. Inertial roll and pitch rate meters operate only in GPC-ACQ and Auto steering modes; in these modes they become operational when track is first initialized. The target inertial rate meters are zeroed in GPC-DES and Manual steering modes. Range and range rate display meters become operational when the track mode is first entered. They are operational in all antenna steering modes. Roll angle and pitch angle display meters operate in all antenna steering modes and become operational when the track mode is first entered.

6.1.4 Break-Track Algorithm

The basic idea of the break-track algorithm is simple. If a no-target condition is obtained in five of the last eight update periods, including the present update period, then a break-track condition is declared and the system is returned to the search mode. The determination of a no-target condition is slightly more involved and is discussed in detail in section 6.5.

6.1.5 Track Waveforms

Passive Modes. The general track waveform for passive modes is illustrated in Figure 6-2. This waveform consists of five consecutive transmit frequency intervals with four time slots per frequency interval and 17 pulse repetition intervals (PRI) per time slot. For a given transmit frequency the receiver dedicates each of the four time slots to the following information:
Figure 6-2. Waveform for Passive Track Modes.
Slot 1 = (Sum channel output) + (Azimuth Difference Channel Output)
Slot 2 = (Sum Channel Output) - (Azimuth Difference Channel Output)
Slot 3 = (Sum Channel Output) + (Elevation Difference Channel Output)
Slot 4 = (Sum Channel Output) - (Elevation Difference Channel Output)

The receiver processes 16 pulses for each of these time slots. The waveform parameters are a function of range and are summarized in Table 6-2.

**Active Modes.** The general track waveform for all active modes is illustrated in Figure 6-3. For these modes only one transmit frequency interval is used. This interval is divided into four time slots with 17 PRI per time slot as in the passive modes. The waveform parameters are listed in Table 6-3.

6.1.6 Tracking Loops and Signal Processor Operation

The signal processor configuration is described in Section 6.4, the angle and angle rate tracking loops are described in Section 6.6, and the range and range rate tracking loops are described in Section 6.7.

6.2 TRACK MODE CONTROL ALGORITHM DESCRIPTION

The track mode algorithm is entered immediately upon detection of a target in the search mode. A detailed block diagram of the track mode control algorithm is given in Figure 6-4. As noted in the introduction to this section this subroutine has two functions: (1) to control tracking loop initializations and (2) to control the computation of tracking loop estimates. These two functions are described below.

6.2.1 Track Mode Initialization Control

The first task is to initialize each of the tracking loops. This means initial values for the target parameters being tracked must be computed to allow
<table>
<thead>
<tr>
<th>RANGE INTERVAL, ( \text{in} )</th>
<th>PULSE WIDTH, ( \mu \text{sec} )</th>
<th>SAMPLE INTERVAL, ( \mu \text{sec} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R &lt; 0.42 )</td>
<td>0.122</td>
<td>6970</td>
</tr>
<tr>
<td>( 0.42 \leq R &lt; 0.95 )</td>
<td>2.075</td>
<td>6970</td>
</tr>
<tr>
<td>( 0.95 \leq R &lt; 1.9 )</td>
<td>2.075</td>
<td>6970</td>
</tr>
<tr>
<td>( 1.9 \leq R &lt; 3.8 )</td>
<td>2.075</td>
<td>6970</td>
</tr>
<tr>
<td>( 3.8 \leq R &lt; 7.2 )</td>
<td>2.075</td>
<td>6970</td>
</tr>
<tr>
<td>( 7.2 \leq R &lt; 9.5 )</td>
<td>2.075</td>
<td>6970</td>
</tr>
<tr>
<td>( 9.5 \leq R &lt; 18.9 )</td>
<td>2.075</td>
<td>2987</td>
</tr>
</tbody>
</table>

Table 6-2  WAVEFORM AND SIGNAL PROCESSING PARAMETERS FOR PASSIVE TRACK MODES
NOTE: ONLY ONE TRANSMIT FREQUENCY IS USED IN ACTIVE MODE

Figure 8-3. Waveform for Active Track Modes.
Table 6-3  WAVEFORM AND SIGNAL PROCESSING PARAMETERS FOR ACTIVE TRACK MODES

<table>
<thead>
<tr>
<th>RANGE INTERVAL, nm</th>
<th>PULSE WIDTH, $\mu$sec</th>
<th>PRF, Hz</th>
<th>SAMPLE INTERVAL, $\mu$sec</th>
<th>SAMPLE PER RANGE BIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R &lt; 9.5$</td>
<td>0.122</td>
<td>6970.</td>
<td>0.122</td>
<td>1</td>
</tr>
<tr>
<td>$R \geq 9.5$</td>
<td>4.15</td>
<td>268.</td>
<td>2.075</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure 6-4  TRACK MODE CONTROL COMPUTER ALGORITHM

1. If Track Flag Up? no
2. Call Track Loop Initialization Algorithm
3. Call Data Valid Flag Control Algorithm
4. Yes, proceed to initialization flag up.
5. No, proceed to track flag up.
6. Generate Target Return and process to obtain all discriminants.
7. Call Break Track Algorithm.
8. Break Track? no
9. Call System Initialization Algorithm (Return to Search).
10. Yes, proceed to track.
11. Exit.
Figure 6-4  TRACK MODE CONTROL COMPUTER ALGORITHM  
(2 of 2)

![Flowchart](image)

- GPC-ACQ or auto? 
  - yes: Call Radar Angle and Angle Rate Track Model
  - no: GPC-ACQ or auto? (recursion)

- manual mode? 
  - yes: Update Roll/Pitch Angle Reference Using Antenna Slew Switch Positions (Eqs. 5.2)
  - no: Call Gimbal Pointing Loop Model

- Call Radar Range and Range Rate Track Model

- Call Target Signal Strength Model

Exit
the difference equation representations of the loops to begin tracking the parameter changes. This initialization is done during the first update period after control has been passed from the search algorithm to the track algorithm. The choice of parameter initialization and the equations that compute these values are discussed in detail in Section 6.3 and Appendix A.

The other task is to initialize the clock used to time the data valid flags and continue to update this clock until the appropriate data valid flags for a given antenna steering mode are all raised. Clock initialization is performed in the first update period after control has been passed to the track algorithms. The subprogram used by the track mode control algorithm to perform these tasks is shown in Figure 6-5 and the data valid flag timeout periods are given in Table 6-2.

6.2.2 Tracking Loop Update Control

The other responsibility of the track mode control algorithm is to control the computations leading to updated estimates by the various tracking loops. This is a complex procedure involving several steps and is outlined below. The first step is to generate a target return signal, based on the latest target - radar configuration and process this signal to produce error signals in the form of discriminants to be used by the appropriate tracking loop models. A set of four subprograms are required to perform this computation. Complete details of this package of algorithms will be given in Section 6.4.

The second step in the update procedure is to check for a break-track condition. This is done using the algorithm described in section 6.5. If a break-track condition is obtained, then the system is reset to the search mode using the algorithm shown in Figure 6-6. If a break-track condition is not obtained then the computation sequence proceeds to the third step which is to update the antenna position and the target inertial angle rates (if appropriate). If the antenna steering mode is GPC-ACQ or Auto, then the target roll and pitch angles, i.e.
Figure 6-5 DATA VALID FLAG CONTROL ALGORITHM

Enter

Update Data Valid Flag Clock

GPC-ACQ or Auto? yes

Angle Data Valid? yes

Put Angle Data Valid Flag Up.

no

Put Range & Range Rate Data Valid Flags Up

yes

Range & Range Rate Data Valid? yes

no

GPC-ACQ or Auto? yes

Angle Rate Data Valid? no

no

EXIT

yes

Put Angle Rate Data Valid Flag Up.

no

Range & Range Rate Flags Up? yes

Put Track Flag Up

no

EXIT

136
Figure 6-6 SYSTEM INITIALIZATION ALGORITHM

ENTER

Initialize All Internal Flags And Controls

Initialize All Internal Clocks

Initialize All Display Flags

Initialize All Display Meters Except Roll/Pitch

System Power Off?

yes
Align Antenna Boresight With R-Frame Z Axis

no

System In Standby?

yes
Hold Antenna At Present Position

no

Initialize Gimbal Pointing Loop

EXIT

137
the antenna position, and the inertial roll and pitch rate estimates
are updated using the model described in section 6.6. If the system is in
the GPC-DES mode, the antenna gimbal pointing loop (section 5.2) is updated using
the latest roll and pitch designates from the GPC. If the system is in the Manual
mode, the antenna gimbal pointing loop is updated using the latest positions of
the antenna slew switches on the cockpit radar console and equations (5.2). Target inertial roll and pitch rates are not tracked in the GPC-DES and Manual
modes.

In the fourth step, the target range and velocity estimates are updated
using the model described in section 6.7. This step is performed in all antenna
steering modes. The fifth and final step is to compute an estimate of the target
signal strength using the algorithm described in section 6.4.

6.3 TRACKING LOOP INITIALIZATION ALGORITHM DESCRIPTION

This subsection gives a detailed description of the algorithm, illus-
trated in Figure 6-7, used to compute the intial state of each tracking loop for
a given antenna steering mode. The basic philosophy is to set the initial states
of the angle, angle rate, range and range rate tracking loops equal to the respec-
tive values of the target c.g. parameters in the update period in which initiali-
ization takes place. The general procedure is to initialize each of the following
items:

- Break-track algorithm,
- Angle and angle rate tracking model (if required),
- Range tracking model,
- Parameters for signal processor,
- Velocity processor model,
- Signal strength algorithm,

in the order shown. Initialization of each item is described in detail below.
Figure 6-7  TRACKING LOOPS INITIALIZATION ALGORITHM

ENTER

Initialize Break-Track Algorithm

GPC-ACQ or Auto?

yes

Initialize Angle & Angle Rate Loops (Eqns 6.1 to 6.3)

Initialize Range Tracking Loop

Initialize Signal Processor Operating Parameters

Initialize Velocity Processor (Eqn 6.4)

EXIT
6.3.1  **Break-Track Algorithm Initialization**

This initialization requires setting the break-track flag low and zeroing the registers used to track the number of no-target conditions obtained in the previous 7 update periods.

6.3.2  **Angle and Angle Rate Tracking Model Initialization**

This model is used only in the GPC-ACQ and Auto antenna steering modes. The initial $\alpha$ and $\beta$ gimbals positions, the initial $\alpha$ and $\beta$ gimbal rates, and the initial target inertial LOS azimuth and elevation rate must be computed in order to start the tracking loops. These values are initialized using the following procedure. First, the positions of the $\alpha$ and $\beta$ gimbals are determined so that the antenna boresight points directly at the target c.g. using the equations

$$
\beta = -\tan^{-1}(\frac{r_{ox}}{s})
$$

(6.1)

$$
\alpha = -\tan^{-1}(\frac{r_{R}^{R}}{r_{oz}^{R}})
$$

where

$$
\frac{r_{R}^{R}}{r_{o}^{o}} = T_{RB}(\frac{r_{B}^{B}}{r_{o}^{o}} - x^{B}),
$$

$\frac{r_{B}^{B}}{x^{B}}$ = Radar offset from orbiter body C.G. expressed in body coordinates,

$$
s^{2} = (r_{oy}^{R})^{2} + (r_{oz}^{R})^{2}.
$$

In the next step, the target inertial LOS azimuth and elevation rate tracking loops are initialized using the expressions

$$
\omega_{TX}^{L} = \frac{v_{oy}^{L}}{|r_{o}^{o}|} + \omega_{Bx}^{L} + \omega_{L}^{L}
$$

(6.2)

$$
\omega_{TY}^{L} = \frac{-v_{ox}^{L}}{|r_{o}^{o}|} + \omega_{By}^{L}
$$
where \( \omega^L_{\text{vo}} \) = target velocity measured in the B-frame and expressed in L-frame coordinates,
\( \omega^L_{\text{wT}} \) = target inertial angular velocity expressed in L-frame coordinates,
\( \omega^L_{\text{wB}} \) = body inertial angular velocity expressed in L-frame coordinates.

These relations are derived in Appendix A.

In the final step, the initial rates of the \( \alpha \) and \( \beta \) gimbals are computed using the expressions

\[
\dot{\alpha} = \frac{v_{\text{vo}}^L}{|v_{\text{vo}}^L| \cos \beta}
\]

(6.3)

\[
\dot{\beta} = \omega^L_{\text{Ty}} - \omega^L_{\text{By}}
\]

which are also derived in Appendix A.

6.3.3 Range Tracking Model Initialization

The range tracker is an \( \alpha - \beta \) tracker (see section 6.7 for details) that generates an estimate of the target range and range rate at each update period. It is initialized by setting the first range and range rate estimates equal to the target c.g. range and velocity, respectively. For the range this is accomplished by digitizing the CG range so that the least significant bit (LSB) represents \( 5/16 \) feet and loading it into the digital integrator that produces the range estimate at its output (see figure 6-32). For the range rate it is accomplished by digitizing the CG velocity so that the LSB represents \( 5/(16t_s) \) feet per second, where \( t_s \) is the update interval, and loading it into the digital integrator that produces the smoothed range rate estimate at its output (as shown in Figure 6-32).

6.3.4 Signal Processor Parameter Initialization

Several signal processor and tracking loop constants change with a different target range interval, processor A/D sample rate, and PRF. Therefore,
the internal controls MRNG, MSAM, and MPRF were defined to apprise the system
of changes in the target range interval, the A/D sample rate, and the PRF,
respectively. These controls are defined in Table 6-4 and they are initialized
as follows. MRNG is determined using the CG range and MSAM and MPRF are determined
using MRNG and the system mode switch (IMODE) position.

6.3.5  Velocity Processor Model Initialization

The velocity processor tracks the target velocity by using five adjacent
doppler filters, always maintaining the target in the center filter. The predicted
velocity estimate in the present update period is averaged with the velocity
estimate from the three previous update periods to obtain the final velocity
estimate, i.e. the velocity is smoothed using the moving window average technique.
Thus, initialization of the velocity processor involves (1) setting each entry
of the array used for averaging equal to the C.G. velocity and determining the
location of the center filter (of the five filter bank) using the equation

\[(6.4) \quad m_c = \text{mod} \left( \left\lfloor \frac{\dot{V}_o}{\Delta} + 0.5 \right\rfloor, 32 \right) \]

where
\[\text{mod} (\cdot, 32) = \text{modulo } 32,\]
\[\left\lfloor \cdot \right\rfloor = \text{greatest integer in } \cdot,\]
\[\Delta = \frac{\lambda_c}{\text{PRF/64}}.\]
\[m_c = \text{number of the center doppler filter}.\]

See section 6.7 for complete details of the velocity processor model.

6.3.6  Signal Strength Algorithm Initialization

The model which is used to compute the radar signal strength in the
present version of the computer simulation is quite simple and does not require
initialization.
<table>
<thead>
<tr>
<th>RANGE INTERVAL, nm</th>
<th>MRNG</th>
<th>MSAM</th>
<th>MPRF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Passive Modes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0-120 ft</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>120-240 ft</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>240-720 ft</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>720 ft-0.42</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.42-0.95</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0.95-1.9</td>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1.9-3.8</td>
<td>7</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3.8-7.2</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>7.2-9.5</td>
<td>9</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>9.5-18.9</td>
<td>10</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>Active Modes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0-120 ft</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>120-240 ft</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>240-720 ft</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>720 ft-0.42</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.42-0.95</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.95-1.9</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1.9-3.8</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3.8-7.2</td>
<td>8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7.2-9.5</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9.5-18.9</td>
<td>10</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
This subsection gives a detailed description of the model used to generate the target return signal and process this signal to obtain all of the discriminants. The objective of this model is to generate the most accurate discriminant estimates possible given the present target scattering model selection and the constraint of real-time computer operation. The method selected to achieve this goal is heavily dependent upon the fact that the target is modeled as a collection of point scatterers and can be roughly outlined as follows. Instead of forming the target return signal at the antenna output and processing the resultant signal on a sample-by-sample basis using the exact Ku-band radar processing configuration shown in Figure 6-8, the processor model uses assumed linearity of the processor from the antenna to the doppler filter output and the assumptions listed in section 6.4.1 to compute the resultant signal at the doppler filter output in closed-form. Then, except for replacing the magnitude detector by a magnitude-squared detector, the remainder of the signal processing model is identical to the corresponding Ku-band radar processor functions. This computation model is illustrated in Figure 6-9. By using this model, sample-by-sample processing can be abandoned, thereby reducing the computation time per update cycle significantly without sacrificing signal processing model accuracy.

In the remainder of this subsection, we will present: (1) the model assumptions, (2) the technique for updating the position and motion of the point targets, (3) complete details of all discriminant generations, including the thermal noise model, (4) the radar signal strength computation algorithm, and (5) the computer model details.
Figure 6-8  SIMPLIFIED DIAGRAM OF KU BAND RADAR TRACK MODE SIGNAL PROCESSING

Target Return Signal

Antenna
Sum/Difference Channel

Video Filter

A/D

Range/Gate Presum

Doppler Filter

Magnitude Detector

PDI

Log Converter

Discriminant Formation
6.4.1 Model Assumptions

The signal generation and processing model is based upon the assumptions listed below. Target assumptions are that

(1) the target is composed of a collection of point scatterers with the properties described in section 4,

(2) radial acceleration of the point targets during a data cycle is ignored.

Radar assumptions are that

(3) the waveform described in Figures 6-2 and 6-3 are transmitted without any distortion,

(4) the antenna does not move with respect to the target during the data cycle,

(5) the receiver's RF and IF electronics work perfectly, (i.e. the down conversion is error-free and the filters do not distort the return signal, but the receiver maintains the same noise figure and noise bandwidth).

(6) the baseband (video) filter has a perfect rectangular impulse response of width equal to the A/D sample interval,

(7) the A/D is treated as an ideal zero-order sample and hold,

(8) quantization noise contributed by the signal processing chain from the A/D to the log converter (see Figure 6-8) is neglected,

(9) the magnitude detector was replaced by a magnitude-squared detector,

(10) Automatic Gain Control (AGC) is not implemented.
Motivation for assumption (1) was discussed in section 4. Assumptions (2) and (4) are forced on us by the real-time processing constraint which rules out sample-by-sample or even pulse-by-pulse processing of the target return signal. Assumption (3) will have little effect on processor accuracy. On the other hand, assumption (5) can have a significant impact upon the fidelity of the angle tracking estimates because the difference channel coupling losses are ignored. The baseband filter assumption will have little effect upon processing accuracy. Impact of assumptions (7) and (8) is not known at the present; if time permits, an equivalent quantization noise will be added to the discriminant computation model. The magnitude-squared detector assumption will have only a slight impact upon the model accuracy. The system AGC will be implemented if time permits.

6.4.2 Target Position and Motion Computation Model

Generation of the discriminants in a given data cycle requires a knowledge of each point target's position and radial velocity with respect to the LOS frame. To obtain these quantities, we utilize the following information. Firstly, the parent simulation provides (1) $\mathbf{r}_o^B$ and $\mathbf{v}_o^B$, the present position and velocity of the target C.G. with respect to the orbiter body frame, and (2) $T_{BoT}$ and $\dot{T}_{BoT}$, which describe the rotation of the T-frame with respect to the B-frame as discussed in sections 2 and 3. Secondly, the radar simulation tracks the angular position and rates of the antenna relative to the orbiter body frame. From these two facts, the position of the kth point target, located at an arbitrary but known position in the T-frame, can be computed in the L-frame using the expression,

$$\mathbf{r}_k^L = T_{LB}(\mathbf{r}_o^B + T_{BoT} \mathbf{r}_k^T - \mathbf{r}_B^T)$$
or, regrouping the terms,

\[(6.5) \quad \dot{\mathbf{r}}_k^L = T_{LB}(\dot{\mathbf{r}}_o^B - \mathbf{X}^B) + T_{LO}^T \dot{\mathbf{r}}_L^T \]

where \(\mathbf{X}^B\) = vector describing the offset of the radar from the orbiter C.G.,

\(T_{LB}(\dot{\mathbf{r}}_o^B - \mathbf{X}^B) = position\ of\ target\ C.G.\ in\ the\ LOS\ frame.\)

The velocity of the arbitrary point target as measured in the LOS frame, can be obtained by time differentiating equation (6.5) and noting that \(\mathbf{X}^B\) and \(\dot{\mathbf{r}}_k^B\) are constant vectors. This gives

\[(6.6) \quad \dot{\mathbf{v}}_k^L = \left[ \dot{T}_{LB}(\dot{\mathbf{r}}_o^B - \mathbf{X}) + T_{LB}^T \dot{\mathbf{r}}_o^B \right] + T_{LO}^T \dot{\mathbf{r}}_k^T \]

where the expression in the square brackets is the velocity of the target C.G. as measured in the L-frame and expressed in L-frame coordinates. Finally, the target radial velocity as measured in the L-frame is obtained by computing the component of velocity in the direction of the radar. Quantitatively, this can be expressed as

\[(6.7) \quad \dot{v}_k^L (radial) = \dot{\mathbf{v}}_k^L \cdot \mathbf{r}_k^L \]

where \(\mathbf{r}_k^L\) = unit vector in the direction of the target.

6.4.3 Angle Discriminant Computation Model

The angle discriminant is essentially formed by comparing the sum channel plus the difference channel signal to the sum channel minus the difference channel signal where both signals are appropriately integrated over the five
transmit frequencies and two range bins. In the sequel, these two quantities will be referred to as the components of the angle discriminant. With this in mind, we proceed to a description of the angle discriminant computation model which is divided into three parts:

1. computation of the noise-free discriminant components,
2. computation of the equivalent thermal noise,
3. computation of the angle discriminant.

Noise-Free Discriminant Component Computation. Figure 6-10 gives a block diagram of the model used to compute the noise-free discriminant components; this model is derived in Appendix C. Figure 6-10a shows the computation of the target response at the magnitude-squared detector output for a given transmit frequency and range bin. Figure 6-10b illustrates the post-detection integration (PDI) of the detector output over frequency and range bin to form the noise-free discriminant component. A detailed description of these steps is given below.

The total response of the target return at the doppler filter output is computed using the assumptions listed in section 6.4.1 and the assumption that the receiver/signal processor configuration is linear from the antenna to the doppler filter output. First, the response of each point target at the doppler filter output is computed in closed-form. Then, using the linearity assumption and the superposition principle, the resultant response for the complete target is computed by vectorially summing the individual responses.

Computation of the doppler filter response for a single point target requires a more detailed explanation. This computation is performed as follows. Using assumptions (1) through (5) and equation (4.1), the return
Figure 6-10  NOISE-FREE ANGLE DISCRIMINANT COMPONENT COMPUTATION MODEL

(a) Magnitude-Squared Detector Response for a Single Frequency, Time Slot, and Range Gate

(b) Formation of Noise-Free Discriminant Component

\[
\begin{align*}
&\text{Antenna Sum Pattern Weighting} \\
&\text{Video Filter} \\
&\text{Range Gate/Presum} \\
&\text{Center (c) Doppler Filter} \\
&\text{Magnitude-Squared Detector}
\end{align*}
\]
from the kth point target over a single time slot referenced at the input to the baseband filter is given by the expression,

\[ S_k(t) = c_k^2 A_k(\rho_{Sk} + \rho_{Dkj}) \sum_{n=0}^{15} \exp\left\{j \left[2\pi f_k t - \phi_{k1}\right]\right\} P\left(\frac{t-nt_p - t_k}{t_c}\right) \]

where \( \rho_{Sk}, \rho_{Dkj} \) = kth target sum and difference pattern weightings,

\[ \phi_{k1} = 2\pi f_c t_k \]

and the other terms are defined after equation (4.1).

After filtering, sampling, range gating and presumming the signal in equation (6.8), we obtain

\[ S_k = c_k^2 A_k R_k(\rho_{Sk} + \rho_{Dkj}) \exp\left\{j \left[2\pi n f_k t_p - \phi_{k1}\right]\right\} \]

where \( n=0, 1, 2, \ldots, 15 \). The factor \( R_k \) in the above expression represents the range gate and presum weighting. It is noted that this factor ignores the mismatch in the presummer due to the target doppler shift. This assumption will have no impact in the short pulse modes and slight effect in the long pulse modes. Quantitatively, the expression for the range gate/presum weighting is

\[ R_k = F(t_k) \]
where

\[
F(t_k) = \begin{cases} 
0, & \text{if } \Delta \geq \frac{3}{2} \text{ or } \Delta \leq -\frac{3}{2} \\
\frac{3 + \Delta}{2}, & \text{if } -\frac{3}{2} \leq \Delta \leq -\frac{1}{2} \\
\frac{1}{2}, & \text{if } -\frac{1}{2} \leq \Delta \leq \frac{1}{2} \\
\frac{3 - \Delta}{2}, & \text{if } \frac{1}{2} \leq \Delta \leq \frac{3}{2},
\end{cases}
\]

and \( \Delta = \frac{t_k}{t_t} \).

There is a very important assumption made at this step. The total energy of the pulse within the range gates is assumed to be exactly split between the early and late gate contribution. However, the phase associated with the point target's true position in the range gate is maintained. This is illustrated in quantitative terms in Appendix C. The assumption was made to enhance the computation speed but it may have a significant impact in those cases where the range tracker does not follow the target with fidelity. If that is the case, the assumption may have to be abandoned.

The final step in this sequence is to compute the response of the doppler filter to the signal given in equation (6.9). (As an aside, it is noted that the target velocity is tracked using 5 adjacent doppler filters where the velocity always seeks to maintain the target in the center (C) filter of the 5. Only information from filter C is used in the formation of the angle discriminant). Since the signal in (6.9) represents a pure doppler tone by assumption (2) of section 6.4.1, we can easily write the response of the kth target for a single frequency and range gate. It is
given by the expression

\[ S_k = \sigma_k \left| A_k R_k \left( \rho_{SK} + \rho_{DKj} \right) \sin \left( 16Z_k \right) \sin(z_k) \exp \left[ -j(15Z_k + \phi_{ki}) \right] \right| \]

where \( Z_k = \pi(m_c/32 - f_{kp}) \)

\[ m_c = \text{number of center filter.} \]

As noted earlier, the complete target return signal at the doppler filter output is obtained by assuming the processing channel from the antenna to the doppler filter output is linear and applying the linear superposition principle. For the \( i \)th transmit frequency and \( l \)th range gate, this gives

\[ S(i,l) = \sum_{k=1}^{N_T} S_k(i,l) \]

where \( N_T \) is the total number of point targets. Magnitude-squared detecting \( S(i,l) \) and PDIing over the appropriate number of transmit frequencies and range gates, we obtain the noise-free angle discriminant component

\[ A = \frac{2}{N_F} \sum_{k=1}^{N_F} \sum_{l=1}^{N_F} | S(i,l) |^2 = \frac{2}{N_F} \sum_{i=1}^{N_F} | S(i) |^2 \]

where \( N_F \) is the number of transmit frequencies and the summation over the range gate is replaced by the factor 2 using the assumption stated earlier.

**Equivalent Thermal Noise.** If we assume that the target signal plus white gaussian noise is introduced at the front end of the receiver, the noise appearing at the PDI output can be shown (see appendix D) to be additive and approximately gaussian with mean and variance given by
where \( N_A \) = PDI ratio for the angle discriminant,

\[ \text{SNR}_D = \text{Signal-to-Noise Ratio referenced to the doppler filter output}, \]

\[ \sigma^2_0 = \text{variance of noise at doppler filter output}. \]

**Angle Discriminant Computation.** The angle discriminant is then computed by the expression

\[
D_A = 10 \log \left( \frac{A_\sigma + \eta_\sigma}{A_\delta + \eta_\delta} \right)
\]

where \( A_\sigma \) represents the sum plus difference noise-free discriminant component, \\
\( A_\delta \) represents the sum minus difference noise-free discriminant component, \\
and \( \eta_\delta \) and \( \eta_\sigma \) are samples from statistically independent random sequences \nwhere each member has the statistics described above. It is noted that \( A_\sigma \) and \( A_\delta \) are computed using equation (6.13) with the appropriate antenna weighting factor.

**Range Discriminant Computation Model**

The range discriminant is formed by comparing the energy from the late range gate to the energy from the early range gate. Description of this computation model will follow the same format as the angle discriminant computation model description.
Noise-Free Discriminant Component Computation. A model for the computation of the noise-free range discriminant component is derived in Appendix C and is shown in Figure 6-11. The basic configuration of the model is identical to the corresponding angle discriminant model. However, there are some differences in the weighting factors between the two and for this reason we outline the processing of the range discriminant component below.

We start with a description of the computation of the single point target response at the doppler filter output. As in the angle discriminant case, the signal at the input to the baseband filter is given by equation (6.8) with $\rho_{Dk,j}$ set equal to zero even if the boresight is not pointing directly at the target. Proceeding to the filtering, sampling, range gating, and presumming process, we obtain the same expression as equation (6.9), but now separate range gate/presum weighting factors, $R_k$, must be computed for the early and late range gates. For the early gate the weighting factor is given by

\begin{equation}
R_{Ek} = F_E(t_k) \quad \text{(Early)}
\end{equation}

where

\[ F_E(t_k) = \begin{cases} 
0 & \text{if } \Delta < -3 \text{ or } \Delta > 1 \\
\frac{3 + \Delta}{2} & \text{if } -3 \leq \Delta \leq -1 \\
\frac{1 - \Delta}{2} & \text{if } -1 \leq \Delta \leq 1
\end{cases} \]

and for the late gate this factor is given by

\begin{equation}
R_{Lk} = F_L(t_k) \quad \text{(Late)}
\end{equation}
where

\[
F_L(t_k) = \begin{cases} 
\frac{1+\Delta}{2}, & \text{if } -1 < \Delta < 1 \\
\frac{3-\Delta}{2}, & \text{if } 1 \leq \Delta < 3 \\
0, & \text{if } \Delta \geq 3 \text{ or } \Delta < -1.
\end{cases}
\]

The range discriminant components only use information from the center doppler filter and therefore they have the same doppler filter weighting as the angle discriminant components. Thus, the kth target response for the early range gate at the doppler filter output is given by

\[
S_{\text{Ek}} = \sum_{k} A_k R_k \sqrt{\text{E}}_k \frac{\sin(16\theta_k)}{\sin 2\theta_k} \exp \left[ -j(15\theta_k + \phi_k) \right]
\]

and the complete target response for the ith transmit frequency, the jth time slot, and the early range gate at the doppler filter output is given by the expression,

\[
S_E(i,j) = \sum_{k=1}^{N_T} S_{\text{Ek}}(i,j).
\]

Expressions for the late gate single target doppler filter response and the total target doppler filter response are identical to equations (6.19) and (6.20), respectively, with E replaced by L.

The noise-free range discriminant component is obtained by magnitude-squared detecting the doppler filter response (6.20) and PDIing over transmit frequencies and time slots. This can be expressed as

\[
R_E = \sum_{i=1}^{N_F} \sum_{j=1}^{4} |S_E(i,j)|^2 = \sum_{i=1}^{N_F} |S_E(i)|^2
\]
where the last equality is obtained by assuming that all time slot components for a given frequency are equal.

**Equivalent Thermal Noise.** The thermal noise added to the noise-free range discriminant component has properties which are identical to the angle discriminant noise, except that the PDI ratio becomes $N_R$, representing the range PDI ratio, instead of $N_a$.

**Range Discriminant Computation.** The range discriminant is computed using the expression

\[
D_R = 10 \log \left( \frac{R_L + \eta_L}{R_E + \eta_E} \right)
\]

where $R_E$ and $R_L$ are computed from equation (6.21) and $\eta_E$ and $\eta_L$ are samples from statistically independent random sequences where each member has the statistics described above.

**6.4.5 Velocity Discriminant Computation Model**

Definition of the velocity and the on-target discriminants rely heavily upon the configuration of the doppler filters used to track the target velocity. The configuration is comprised of five adjacent filters as shown in Figure 6-12 where the tracker seeks to maintain target velocity in the center filter. In the sequel these filters will be labeled (from lowest frequency to highest frequency) Low Outrigger (LO), Low (L), Center (C), High (H), and High Outrigger (HO), respectively. The velocity discriminant is then formed by comparing all of the energy from the low (L) filter to all the energy from the high (H) filter. The form of this model is identical to the range and angle discriminant model and its description will follow the same format.
Figure 6-12. Track Mode Doppler Filter Configuration (Only Mainlobe Response Shown).
Noise-Free Discriminant Component Computation. Figure 6-13 gives
a block diagram of the computation model and Appendix C gives a derivation
of this model. Since the processing has the same form as the range and
angle discriminant models, we will only provide the various weighting
factors used in this case and point out any differences.

The antenna weighting factor is given by \( \rho_{sk} \) where \( \rho_{sk} \) is computed
from equations (4.7) and (4.8). As in the range discriminant case, the
difference pattern weighting is set to zero even though the antenna may
not be pointing directly at the target. The range gate weighting \( R_k \) is
computed as described in equation (6.10). The range gate weighting assumption
used in the angle discriminant computation, applies in this case, as well.
The doppler filter weighting factor is the same as the range and angle case
with \( m_L \) (or \( m_H \), depending on the component) replacing \( m_C \).

If we let the total response for the \( i \)th frequency, the \( j \)th time
slot, and the \( k \)th range gate at the \( L \)-doppler filter output be given by
\( V_L(i,j,k) \), then the noise-free velocity discriminant component is obtained
by magnitude-squared detecting and PDIing over frequencies, time slots, and
range gates. This is expressed as

\[
F_L = \sum_{i=1}^{N_F} \sum_{j=1}^{2} \sum_{k=1}^{2} |V_L(i,j,k)|^2 = 8 \sum_{i=1}^{N_F} |V_L(i)|^2
\]

(6.23)  

where the last equality is obtained by assuming that all time slot and
range gate components are equal for a given transmit frequency.

Equivalent Thermal Noise. The velocity discriminant noise has the
same form as the angle and range discriminant case with \( N_V \), the velocity
PDI ratio, replacing \( N_A \) in equations (6.14) and (6.15).
Figure 6-13
NOISE-FREE VELOCITY DISCRIMINANT COMPONENT COMPUTATION MODEL

(a) Magnitude-Squared Detector Response for a Single Frequency, Time Slot, Range Gate, and Doppler Filter

(b) Formation of Noise-Free Discriminant Components

1. Data from
   2 Frequencies
   4 Time Slots
   2 Range Gates
Velocity Discriminant Computation. The velocity discriminant is computed from the expression

\[ D_V = 10 \log \left( \frac{F_L + \eta_L}{F_H + \eta_H} \right) \]  

where \( F_L \) and \( F_H \) are computed using equation (6.23) and \( \eta_L \) and \( \eta_H \) are samples from statistically independent random sequences where each member has the statistics described above.

6.4.6 On-Target Discriminant Computation Model

The On-Target discriminant is formed by comparing the total energy from the center (C) doppler filter to the combined total energy from the LO and HO doppler filters over a data cycle. Computation of the noise-free discriminant components is identical to the velocity discriminant case with \( m_C, m_{LO}, \) or \( m_{HO} \) replacing \( m_L \) and \( m_H \) in the doppler filter weighting factor. The On-Target discriminant noise characteristics are identical to those given for the velocity discriminant case above. Therefore, the expression for the On-Target discriminant is

\[ D_{OT} = 10 \log \left( \frac{F_C + \eta_C}{F_{LO} + F_{HO} + \eta_{LO}} \right) \]  

where \( F_C, F_{LO}, \) and \( F_{HO} \) are computed from equations (6.23) and \( \eta_C \) and \( \eta_{LO} \) are samples from statistically independent random sequences where each member has the statistics described above.

6.4.7 Radar Signal Strength Computation Model

In the Ku-Band radar, the radar signal strength meter is designed to work in the tracking mode only. During the track mode the radar signal strength meter is determined from the following algorithm
(6.26) \[
\text{RSS} = \begin{cases} 
\text{AGC}_\text{initial}, & \text{if AGC} \geq \text{AGC}_\text{initial} \\
0, & \text{if AGC} < \text{AGC}_\text{initial}
\end{cases}
\]

where \(\text{AGC}_\text{initial}\) is the setting of the system AGC when the track mode is first entered. The AGC is designed to maintain the average signal plus noise voltage at 1.4 quantization levels at the video filter output.

Since the system AGC is not modeled in the present version of the track mode simulation, the algorithm of (6.26) is replaced by the following computation:

(6.27) \[\text{RSS} = \text{SNR}_V\]

where \(\text{SNR}_V\) is the signal-to-noise ratio at the video filter output and is computed from equation (5.16) for passive modes and equation (5.22) for active modes. This approximation will be highly accurate for \(\text{SNR}_V \gg 1\), but will break down for \(\text{SNR}_V \leq 1\). If time permits, the Ku-Band Radar AGC algorithm and signal strength algorithm will be simulated more accurately.

6.4.8 Computer Model Details

The computer model shown in Figure 6-14 consists of five subroutines. A separate subroutine is dedicated to each of the following functions:

1. updating of all transformation matrices,
2. updating of the LOS position, velocity and RCS value for each scatterer and updating of the LOS position and velocity for the target C.G.,
3. computation of all noise-free discriminant components,
4. computation of all discriminants (including thermal noise),
5. computation of target signal strength.

Each of these subroutines is described in detail below.
Figure 6-14 SIGNAL GENERATION AND PROCESSING MODEL COMPUTER ALGORITHM
(1 of 3)

ENTER

Compute \( T_{LB} \)  
(Eqn 6.28)

Compute \( T_{LoT} \)  
(Eqn 6.29)

Compute \( T_{LB} \)

Compute \( T_{LoT} \)  
(Eqn 6.30)

2

Compute Target C.G.  
Position, Range, &  
Direction in L-Frame

(Eqn 6.31)

Compute Target C.G.  
Velocity Measured in  
L-frame

(Eqn 6.32)

Compute Target  
Scattering Properties  
(see Figure 4-8)

Compute \( K \)th Target  
Position, Range, &  
Direction in L-Frame

(Eqn 6.33)

Compute \( K \)th Target  
Velocity, Radial Velocity  
Measured in L-Frame.  
(Eqn 6.34)

Last Target?  

no  

yes

3

3
Compute k th Target
Sum Channel Weighting
(Eqns 4.7, 4.9)

Compute k th Target
AZ and EL Difference Channel Weighting
(Eqns 4.8, 4.10, 4.11)

Compute k th Target
Range Gate/Presum Weighting
(Eqns 6.10, 6.17)

Compute k th Target Doppler Filter Weightings
(Eqn 6.35)

Compute k th Target Phase Factor Due To Range Difference
(Eqn 6.36)

Form Components of eqn 6.37 for k th Target and Vectorially Sum with K-1 Previous Values.

Form Discriminant Component For ith Frequency, \( |\cdot|^2 \) Detect and Sum with i-1 Previous Values
(see Eqn 6.38)

Compute Effective RCS
(Eqn 6.39)
Figure 6-14  SIGNAL GENERATION AND PROCESSING MODEL COMPUTER ALGORITHM  
(3 of 3)

5

- Compute Discriminant Component
- Scale Factor (Eqn 6.40 or 6.41)

- Compute SNR_D

- Compute Angle Discriminant
- Includes Noise (Eqns 6.42 to 6.44)

- Compute Velocity Discriminant
- (Similar to Angle with appropriate changes)

- Compute Velocity Discriminant
- (Similar to Angle with appropriate changes)

- Compute On-Target Discriminant
- (Similar to Angle with appropriate changes)

6

- Compute SNR_v
- (Use C.G. Range and Effective RCS)

- Set Radar Signal Strength equal to SNR_v

ENTER
Update of Transformation Matrices (TRNSFM). This subroutine updates $T_{LB}$, $T_{LT}$, $\dot{T}_{LB}$, and $\dot{T}_{LT}$. The transformation matrix $T_{LB}$ is computed with the expression

\[
T_{LB} = \begin{pmatrix}
C_B & 0 & -S_B \\
0 & 1 & 0 \\
S_B & 0 & C_B
\end{pmatrix}
\begin{pmatrix}
1 & 0 & 0 \\
0 & C_a & S_a \\
0 & -S_a & C_a
\end{pmatrix}
\begin{pmatrix}
C_Y & S_Y & 0 \\
-S_Y & C_Y & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

(6.28)

where $\alpha, \beta = \text{latest measurement of antenna gimbal position}$, $\gamma = \text{yaw angle of R-frame with respect to B-frame (nominally 67°)}$, $C = \cos$, $S = \sin$.

The transformation $T_{LT}$ is obtained from

\[
T_{LT} = T_{LB} \cdot T_{B_0T}
\]

(6.29)

where $T_{LB} = T_{LB}$ is computed in equation (6.28) and $T_{B_0T}$ is provided by the parent simulation.

The matrix $\dot{T}_{LB}$ is computed by time differentiating $T_{LB}$ as given in equation (6.28) and noting that $\alpha, \beta$ vary with time but $\gamma$ is fixed. Finally, the matrix $\dot{T}_{LT}$ is computed from the expression

\[
\dot{T}_{LT} = T_{LB} \cdot \dot{T}_{B_0T} + T_{LB} \cdot \dot{T}_{B_0T}
\]

(6.30)

where $T_{LB}$ and $\dot{T}_{LB}$ are defined above and $T_{B_0T}$ and $\dot{T}_{B_0T}$ are provided by the parent simulation.
Target Position and Velocity Update (PVTRAN). This subroutine computes
(1) the position, range and direction vector in LOS coordinates for each
scatterer and the target C.G., (2) the velocity and radial velocity component
as measured in the LOS frame for each scatterer and the target C.G., and
(3) the RCS value for each scatterer. The subroutine is organized as follows.
Since each scatterer's LOS position and velocity computation requires the
C.G. position and velocity in the LOS frame, the C.G. parameters are computed
first then the parameters for each point target are computed.

The target C.G. position, range, and direction vector in LOS coordi-
nates are computed from

\[
\begin{align*}
\hat{r}_o^L &= T_{LB}(\hat{r}_o^B - \hat{X}_B) & \text{(position)} \\
|\hat{r}_o^L| &= |\hat{r}_o^L| & \text{(range)} \\
\hat{r}_o^L &= \hat{r}_o^L / |\hat{r}_o^L| & \text{(direction)}
\end{align*}
\]

where \( \hat{r}_o^B \) is provided by the parent simulation, \( \hat{X}_B \) is the fixed radar
offset from the orbiter C.G., and \( T_{LB} \) is computed above. The C.G. velocity
and radial velocity component as measured in the LOS frame and expressed
in LOS coordinates are given by

\[
\begin{align*}
\hat{r}_o^L &= T_{LB}\hat{r}_o^B + T_{LB}\hat{r}_o^B & \text{(velocity)} \\
\hat{r}_o^L &= \hat{r}_o^L \cdot \hat{r}_o^L & \text{(radial component)}
\end{align*}
\]

where \( \hat{r}_o^B \) and \( \hat{r}_o^B \) are provided by the parent simulation and \( T_{LB} \) and \( \hat{T}_{LB} \)
are computed above.

The next step is to update the scatterer positions in the target frame
and the scatterer RCS values using the subprogram described in Section 4.5
(see Figure 4-9). Then, each of the scatterer's position and velocity parameters
are computed. LOS position, range, and direction for the kth scatterer are given by

\[
\begin{align*}
\dot{r}_k^L &= \dot{r}_0^L + T_{LO} \dot{T}_k^T \\
(6.33) \\
\dot{r}_k^L &= |\dot{r}_k^L| \\
\hat{r}_k^L &= \frac{\dot{r}_k^L}{|\dot{r}_k^L|}
\end{align*}
\]

where \( \dot{r}_k^T \) is fixed and \( \dot{r}_0^L \) and \( T_{LO} \) are computed above. LOS frame velocity of the kth scatterer is computed from the expression

\[
\begin{align*}
\dot{r}_k^L &= \dot{r}_0^L + T_{LO} \dot{T}_k^T \\
(6.34) \\
\dot{r}_k^L &= \dot{r}_k^L \cdot \dot{r}_k^L
\end{align*}
\]

where \( \dot{r}_0^L \) and \( T_{LO} \) are computed above.

**Signal Generation and Processing (SIGNAL).** The main purpose of this subroutine is to compute all of the noise-free discriminant components. It performs the calculations which are described in sections 6.4.3 through 6.4.6 and derived in Appendix C. This task is performed as follows.

First, for the kth target and ith transmit frequency, all of the weighting factors required to form the various responses at the doppler filter output are computed. These include the sum pattern weighting, difference pattern weighting, range gate/presum weighting, doppler filter weighting and initial phase computation. Antenna sum and difference pattern weightings are computed using equations (4.2) through (4.7) and the antenna models described in section 4. The range gate/presum weighting factor is given by equation (6.10) for non-range discriminant components and by equation (6.17) for range discriminant components. The doppler filter weighting is computed from the expression
(6.35) 

\[ F_k(m) = \frac{\sin(16Z_k)}{\sin(Z_k)} \exp\left[-j15Z_k\right] \]

where \( Z_k = \pi\left(\frac{m}{32} - f_{k_p}\right) \),

\( m = m_{LO}, m_{L}, m_{U}, m_H, m_{HO} \).

The doppler weights need only be computed for the first transmit frequency, since the PRF is adjusted at each new frequency to maintain a constant filter position for a nonaccelerating target. Finally, the phase of the kth target return referenced to the leading edge of the C.G. return is computed from

\[ \phi_{k1} = \frac{4\pi}{\lambda_{c1}} (r_k - r_G) \]

(6.36)

where \( \lambda_{c1} \) is the wavelength of the ith transmit frequency. It is remarked that one can just as well compute the phase using the expression

\[ \phi_{k1} = \frac{4\pi}{\lambda_{c1}} (r_k - r_L) \]

where \( r_G \) is the range to the center of the range gates.

The next step is to form the following responses for the kth target and ith frequency at the doppler filter output:

\[ A_{1k} = \text{Sum Component} = \sqrt{\sigma_0} \rho_{Sk} R_k F_k(m_c) \exp(j\phi_{k1}) \]

\[ A_{2k} = \text{Azimuth Difference Component} = \sqrt{\sigma_0} \rho_{AZk} R_k F_k(m_c) \exp(j\phi_{k1}) \]

(6.37)

\[ A_{3k} = \text{Elevation Difference Component} = \sqrt{\sigma_0} \rho_{ELk} R_k F_k(m_c) \exp(j\phi_{k1}) \]

\[ R_{1k} = \text{Early Component} = \sqrt{\sigma_0} \rho_{Sk} R_k E_k F_k(m_c) \exp(j\phi_{k1}) \]

\[ R_{2k} = \text{Late Component} = \sqrt{\sigma_0} \rho_{Sk} L_k F_k(m_c) \exp(j\phi_{k1}) \]

\[ V_{1,2,3,4k} L,H,LO,H0 \text{ Components} = \sqrt{\sigma_0} \rho_{Sk} R_k F_k(\cdot) \exp(j\phi_{k1}) \]

where \( . = m_{L}, m_H, m_{LO}, m_{HO} \), respectively. Once formed these components are
vectorially summed over the number of targets to form the complete target response for the component at the doppler filter output for the ith frequency.

Then, these components are combined appropriately to form the noise-free discriminant components at the doppler filter output for the ith frequency. After this step, the newly formed components are magnitude-squared detected and summed over NF transmit frequencies. Result of all of this processing gives the following noise-free discriminant components at the PDI output:

\[
AZ_o = \sum_{i=1}^{NF} \sum_{k=1}^{NT} \left| k^2 (A_{1k} + A_{2k}) \right|^2
\]

\[
AZ_s = \sum_{i=1}^{NF} \sum_{k=1}^{NT} \left| k^2 (A_{1k} - A_{2k}) \right|^2
\]

(for ELo, ELs replace the subscript 2 by 3)

\[
R_E = \sum_{i=1}^{NF} \sum_{k=1}^{NT} R_{1k}^2
\]

\[
R_L = \sum_{i=1}^{NF} \sum_{k=1}^{NT} R_{2k}^2
\]

\[
V_L = \sum_{i=1}^{NF} \sum_{k=1}^{NT} V_{1k}^2
\]

\[
V_H = \sum_{i=1}^{NF} \sum_{k=1}^{NT} V_{2k}^2
\]

( for VLo, VHo replace subscripts 1,2 by 3,4)

where each of the components above is within a constant scale factor of the actual noise-free discriminant component seen at the PDI output.

One final step is performed in this subroutine and that is to compute a quantity called the average effective cross-section. This
quantity is a measure of the average target cross-section weighted by the normalized antenna sum pattern value. This value is averaged over five frequencies and is given by the expression

\[
(6.39) \quad \text{Effective Cross Section} \quad A \left[ \frac{1}{N_F} \sum_{i=1}^{N_T} | \sum_{k=1}^{K} \beta_k e^{j\theta_k} |^2 \right].
\]

**Discriminant Component Computation (DISCRM).** This subroutine adds equivalent thermal noise to each of the discriminant components and computes the discriminants with the resulting component values. We first compute the scale factor alluded to earlier. This factor contains many of the range equation terms. For the passive mode it is given by

\[
(6.40) \quad S_1 = \frac{4 G^2 \lambda^2 P_\text{C} T N_P}{(4\pi)^3 (R_o)^{1/2} L_k T S B N_F}. \tag{Passive Mode}
\]

where \(N_P\) is the number of samples per pulse and the other terms are defined in Section 5. For the active mode this factor becomes

\[
(6.41) \quad S_1 = \frac{4 G^2 P_\text{B} T N_P}{(4\pi)^2 (R_o)^2 L_k T S B N_F}. \tag{Active Mode}
\]

where

\[P_\text{B} = \frac{P_B G_B}{L_B},\]

\(P_B = \text{peak transmit power of beacon},\)

\(G_B = \text{one-way beacon antenna gain},\)

\(L_B = \text{beacon transmit losses}.)\)
The next step is to tackle the angle discriminant computation. This includes computing the statistics of the noise for this discriminant. Mean and variance of the noise are computed as follows:

\[
\text{Mean}_{AZ} = N_A \quad \text{(same for } \sigma \text{ and } \delta \text{ components)}
\]
\[
\text{var}_{AZ\sigma} = \left[ 2 N_A S_{1AZ\sigma} + N_A \right]
\]
\[
\text{var}_{AZ\delta} = \left[ 2 N_A S_{1AZ\delta} + N_A \right]
\]

where \( S_1 \) is computed in equation (6.40 or 6.41), \( AZ \) and \( AZ \) are computed in equations (6.38) and \( N_A \) is the angle PDI ratio. It is important to note that the variance of the \( I, Q \) noise components at the doppler filter output are assumed to be equal to unity for convenience in the computation. In the next step, the equivalent noise is added to the angle discriminant components and we obtain

\[
D_{AZ\sigma} = |N_A S_{1AZ\sigma} + \text{Mean}_{AZ} + \sqrt{\text{Var}_{AZ\sigma}} N(0,1)|
\]
\[
D_{AZ\delta} = |N_A S_{1AZ\delta} + \text{Mean}_{AZ} + \sqrt{\text{Var}_{AZ\delta}} N(0,1)|
\]

where \( N(0,1) \) is defined as a random selection from a gaussian population with zero mean and unit variance. The last step is to compute the angle discriminant

\[
D_AZ = 10 \log (D_{AZ\sigma}/D_{AZ\delta})
\]

where the logarithm computation is assumed to be base 10.
The range, velocity and on-target discriminants are computed in an identical manner, making the appropriate changes in scale factors and components.

**Radar Signal Strength Computation (RSS).** As discussed in Section 6.4.7, the radar signal strength is computed very simply in the present version of the tracking simulation. The radar signal strength is set equal to the SNR at the video filter output where it is assumed that the transmitter is at full power. This computation is done using equation (5.16) for the passive mode where in this case $P_T$ is always the maximum peak transmitter power and $\sigma$ (the RCS) is computed using equation (6.39). For the active mode equation (5.22) is used to obtain the $SNR_v$.

### 6.5 Break-Track Algorithm Description

The break-track algorithm used in the track mode simulation is functionally identical to the logic used in the Ku-Band radar signal processor. Figure 6-15 gives a simplified block diagram of the break-track algorithm. Key components are the two discriminants and the no-target condition determination. In this subsection, we will describe the no-target condition and its determination, describe the break-track condition, and describe the implementation of this algorithm on the computer.

#### 6.5.1 Noise-Free Discriminant Response Functions

Both discriminants take advantage of the shape of the doppler filter's mainlobe and its relative position with respect to the other filter mainlobes in order to determine target location in the filter bank and absence or presence of the target. This is most clearly seen from plots of the noise-free velocity and on-target discriminants as a function of target doppler velocity given in Figures 6-16 and 6-17, respectively.
Figure 6-15  SIMPLIFIED BLOCK DIAGRAM OF BREAK-TRACK ALGORITHM

- Velocity
- Discriminant

No-Target Condition Determination For Present Update Period

No-Target Condition in 5 of last 8 update periods?

If yes, break-track

If no, continue

On-Target
Discriminant
Figure 6-16. Noise-Free Velocity Discriminant Frequency Response.
Figure 6-17. Noise-Free On-Target Discriminant Frequency Response.
6.5.2 Determination of a No-Target Condition

The basic idea behind the no-target determination is as follows. If a noise-only condition exists then the velocity discriminant value will be in the neighborhood of 0 dB and the on-target discriminant value will be in the vicinity of -3 dB. However, if a target with sufficient strength exists within the five filters, at least one of the discriminant values will not be near its noise only bias value as shown in Figures 6-16 and 6-17. Thus, the method for determination of a no-target condition is to establish thresholds about the bias values in each case (shown as dashed lines in Figures 6-16 and 6-17) and compare the discriminant values to their respective thresholds. A no-target condition is declared if both discriminants lie between their thresholds, i.e. in the region of their no-target bias values.

Quantitatively, the no-target condition can be described as follows. First, the following quantities (called discrete) are defined:

\[
F_{TH} = \begin{cases} 
1, & \text{if } |D_V| \leq T_v \\
0, & \text{if } |D_V| > T_v
\end{cases}
\]

(6.45)

\[
O_T = \begin{cases} 
1, & \text{if } D_{OT} \leq T_H \\
0, & \text{if } D_{OT} > T_H
\end{cases}
\]

\[
A_{OT} = \begin{cases} 
1, & \text{if } D_{OT} > T_L \\
0, & \text{if } D_{OT} \leq T_L
\end{cases}
\]

Then, target/no-target decision is based upon the product of the discrete as follows:

\[
(F_{TH}) (O_T) (A_{OT}) = \begin{cases} 
1, & \text{no target} \\
0, & \text{target}
\end{cases}
\]

(6.46)

This decision logic is illustrated in Figure 6-18.
Figure 6-18 NO-TARGET DETERMINATION ALGORITHM
6.5.3 Break-Track Determination

A break-track condition is declared if a no-target condition is obtained in the present update period and four of the last seven update periods.

5.4 Computer Algorithm Details

Figure 6-19 illustrates the computer algorithm used for the break-track determination. It implements the equations described above and requires no further description.

6.6 ANGLE AND ANGLE RATE TRACKING LOOP MODEL DESCRIPTION

In the GPC-ACQ and Auto modes, the radar provides estimates of the target inertial roll and pitch rates and tracks the target roll and pitch angles in the Orbiter Body coordinate system. A simplified block diagram of its mechanization in the Ku-Band radar is illustrated in Figure 6-20. In this subsection, we shall describe (1) the mathematical model used to represent this tracking system, (2) the major assumptions and approximations underlying this model, (3) the system and target error effects incorporated into the model, and (4) the computer implementation of the model.

6.6.1 The Model

As noted in Figure 6-20, the tracking system is composed of an α and β gimbal tracking loop. It was suggested in reference [20] that these loops be approximated by the second order continuous-time models shown in Figures 6-21 and 6-22. The loop constants \( w_n \) and \( \tau \) were designed (see reference [21]) so that the angle rate estimator is critically damped and so that the loop transient response is damped out as quickly as possible while still meeting the loop noise specification. The design values of \( w_n \) and \( \tau \) are given in Table 6-5 for reference.
ENTER

Initialize All Discretes to Zero

$|D_v| > T_v$^\text{?}$

no Set FTH=1

yes

$D_{OT} > T_H$?

no Set AOT=1

yes

$D_{OT} < T_L$?

no

yes

Compute Product $P(n) = (FTH)(OT)(AOT)$

Update Break-Track Sum:
$S(n) = S(n-1) + P(n) - P(n-8)$

Update Registers: 8 most recent values of $P$. 

EXIT
Figure 6-20  SIMPLIFIED DIAGRAM OF KU-BAND ANGLE RATE AND ANGLE TRACKER
Figure 6-22: θ - LOOP MODEL
Table 6-5  ANGLE TRACKING LOOP CONSTANTS $\nu_n$ AND $\tau$

<table>
<thead>
<tr>
<th>RANGE INTERVAL, nm</th>
<th>$\nu_n$, Hz</th>
<th>$\tau^*$, SEC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Passive Mode</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R &gt; 9.5$</td>
<td>0.027</td>
<td>11.8</td>
</tr>
<tr>
<td>$3.8 \leq R \leq 9.5$</td>
<td>0.027</td>
<td>11.8</td>
</tr>
<tr>
<td>$1.9 \leq R \leq 3.8$</td>
<td>0.075</td>
<td>4.2</td>
</tr>
<tr>
<td>$R \leq 1.9$</td>
<td>0.12</td>
<td>2.7</td>
</tr>
<tr>
<td><strong>Active Mode</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R &gt; 9.5$</td>
<td>0.027</td>
<td>11.8</td>
</tr>
<tr>
<td>$R \leq 9.5$</td>
<td>0.075</td>
<td>4.2</td>
</tr>
</tbody>
</table>

* $\rho = 1$ in the angle rate loop design. Therefore $\tau = 2/\nu_n$. 
The α and β loop models shown in Figures 6-21 and 6-22 were adopted, with one modification, as the basis for the angle tracking performance computer model. The one modification is that the return signal is actually generated and processed to produce the angle discriminants (see section 6.4). This provides more flexibility and accuracy in the modeling of angle tracker error sources as discussed below. Introduction of the discriminant generation into the model requires calculation of the equivalent loop constant $k_{eq}$. This is done using the expression

$$k_{eq} = \frac{\omega_n^2}{4 k_a \rho}$$  \hspace{1cm} (6.46)$$

where

- $\omega_n$ = loop natural frequency given in Table 6-5,
- $k_a$ = slope of normalized antenna difference pattern given in Figure 4-8,
- $\rho = \frac{1}{1+SNR^{-1}}$

and it is assumed that $SNR >>1$ so that $\rho=1$. Table 6-6 summarizes the results of the $k_{eq}$ calculation for each value of $\omega_n$ listed in Table 6-5.

6.6.2 Model Assumptions and Approximations

The analog models of the α and β tracking loops are based upon the following assumptions. In the area of the antenna electronics, the antenna gimbal motors are treated as perfect analog integrators and any filtering used for signal shaping, predistortion, or smoothing is assumed to work ideally. The rate stabilization loop is assumed to act instantaneously to remove the body inertial angular velocity from the estimates of the target inertial LOS azimuth and elevation rates. This is a reasonable assumption, since the rate stabilization loop bandwidth is much wider than the angle rate loop bandwidth. Also, any errors such as gyro drift or thermal noise introduced by the antenna electronics, are ignored.
### Table 6-6  EQUIVALENT ANGLE TRACKING LOOP CONSTANTS $k_{eq}$ and $k'$

<table>
<thead>
<tr>
<th>RANGE INTERVAL, nm</th>
<th>$k_{eq}$, deg/sec$^2$</th>
<th>$k'$, deg/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Passive Mode</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R &gt; 0.5$</td>
<td>$2.0106 \times 10^4$</td>
<td>$2.3725 \times 10^3$</td>
</tr>
<tr>
<td>$3.8 \leq R \leq 9.5$</td>
<td>$2.0106 \times 10^4$</td>
<td>$2.3725 \times 10^3$</td>
</tr>
<tr>
<td>$1.9 \leq R \leq 3.8$</td>
<td>$1.5529 \times 10^3$</td>
<td>$6.5907 \times 10^3$</td>
</tr>
<tr>
<td>$R &lt; 1.9$</td>
<td>$3.9750 \times 10^3$</td>
<td>$1.0546 \times 10^2$</td>
</tr>
<tr>
<td><strong>Active Mode</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R &gt; 9.5$</td>
<td>$2.0106 \times 10^4$</td>
<td>$2.3725 \times 10^3$</td>
</tr>
<tr>
<td>$R \leq 9.5$</td>
<td>$1.5529 \times 10^3$</td>
<td>$6.5907 \times 10^3$</td>
</tr>
</tbody>
</table>

$k_{eq} = \frac{w_n^2}{4}$ (normalized difference pattern slope)

$k' = k_{eq}^r$
The signal generation and processing assumptions that affect the present version of the angle tracking model are (1) the choice of antenna sum and difference patterns, (2) ignoring the quantization noise introduced by the digital signal processor, and (3) ignoring the difference in microwave loss between the sum and difference channel. Neglecting the microwave loss difference will have a noticeable impact upon angle tracker performance at low SNR. This error source will be included if time permits. Quantization effects are ignored in all processor steps except the last one: computation of the angle discriminants. These discriminants are quantized to 3/16 dB accuracy. Thus, one of the major sources of quantization is included in the model and the impact of assumption (2) is not too significant. However, quantization of the discriminant does increase the importance of the antenna difference pattern selection, as this choice will affect the resolution capability of the angle rate estimator and the angle tracker.

The last major assumption involves the implementation of the continuous-time loops of Figures 6-21 and 6-22 on the computer. These models are approximated by the discrete-time loops shown in Figures 6-23 and 6-24. The fundamental assumption used in the discretization process was to replace the analog integrators by the digital integration model of Figure 6-25. Effect of this approximation is to constrain the antenna to move in a stair-case fashion. That is, the antenna position remains constant during an update period and is moved to the new predicted $\alpha$ and $\beta$ at the beginning of the next update period. Therefore, the digital integrator approximation will be practical provided the commanded $\alpha$ and $\beta$ rates are not too large.
Figure 6-23 DISCRETIZED $\alpha$-LOOP
Figure 6-24  DISCRETIZED $\beta$-LOOP
Figure 6-25 DISCRETE REPRESENTATION OF INTEGRATOR
6.6.3 Error Sources Modeled

Error sources in the Ku-Band radar angle and angle rate tracking loop that are modeled in the simulation include

- Target error effects (to the extent that the target scattering model is correct),
- Thermal noise,
- Boom deployment error,
- Radar offset error,
- Discriminant error,
- Gimbal bias error.

All of the errors listed above, except the gimbal bias error, are included in the generation of the angle discriminant. Target-induced angle and angle rate measurement errors are included by virtue of the fact that the return signal is generated and processed. Then, provided the target scattering model is accurate, the target-induced errors will be accurately represented. Thermal noise is based upon the receiver and signal processor configuration; the exact method of computation is derived in Appendix D. Boom deployment and radar offset errors occur because the radar transforms the radar estimates of angle and angle rate to body coordinates assuming the radar is located at the orbiter body C.G. and the radar frame is yawed +67° with respect to the body frame. These two errors are included by computing the target return signal based upon a radar offset from the orbiter C.G. and then transforming the resulting radar estimates with the same equations that are used by the radar microprocessor. Discriminant error is the distortion introduced by the method of discriminant computation. This distortion is induced by (1) large target angle errors or (2) low SNR as illustrated in Figure 6-26. The last error source included in the model is gimbal bias error, i.e., the error in the gimbal position.
Figure 6-28. Angle Discriminant Test Results.
reading. This source is easily incorporated into the model as illustrated in Figure 6-21.

6.6.4 Model Performance

This subsection presents the results of tests of the angle and angle rate tracking model. Two tests of the angle tracking model were performed. In the first test, the response of the angle discriminant generator was computed as a function of angle error and SNR. This test was performed with the quantization removed. Results of this test are given in Figure 6-26 and they agree reasonably well with the theoretically predicted discriminant function given in [2] and [22]. The second test checked the step response of the angle rate estimator to determine whether it is behaving as a second order critically damped loop. This test made the following assumptions: (1) the SNR was much greater than unity, (2) the loop was run in an "analog" fashion, i.e. all quantization and discretization of the quantities involved was ignored, and (3) the α-loop was used for the test with β set equal to zero. Results of this test are shown in Figures 6-27 through 6-29 for each loop bandwidth. The 2% convergence times obtained from the computer model agree quite well with the theoretical values as obtained from the solution of the following transcendental equation

\[(\alpha + 1) = 0.02 e^a\]

where

\[\alpha = \omega_n t_2\],
\[\omega_n = \text{loop natural radial frequency,}\]
\[t_2 = 2\% \text{ convergence time} .\]

We make the following disclaimer about the tests and tests results discussed above. These tests show only that the simulation model provides accurate representation of the theoretical design of the angle tracking loops. It does not prove that the model response will closely approximate the actual hardware response under all conditions.
Figure 8-27. Angle Rate Loop Step Response (R < 1.9 n.mi.).
Figure 6.26. Angle Rate Loop Step Response (1.9 n.mi. < R < 3.8 n.mi.).
Figure 6-29. Angle Rate Loop Step Response, (3.8 nmi<R<9.5 nmi).

PITCH ANGLE = 0.0 DEGREES
RANGE = 9.5 nm
ACTUAL 2%
CONVERGENCE TIME = 37.6 SEC
THEORETICAL 2%
CONVERGENCE TIME = 34.4 SEC
6.6.5 **Computer Model Details**

The computer model of the $\alpha$ and $\beta$ tracking loops is broken into two distinct parts: (1) the generation of the discriminants and (2) using these new discriminants to generate new target inertial roll and pitch rates and new target roll and pitch angles. The first step is included in the signal generation and processing algorithm and was described in section 6.4. In this subsection we present a detailed description of step (2).

The algorithm used to update the angle rates and angles is shown in Figure 6-30. A stepwise description of the algorithm follows below. A preliminary step that is required prior to updating the angle and angle rate filter equation is to quantize the angle discriminants to 3/16 dB accuracy:

\[
D_{AZ}(n) = \left\lceil \frac{16}{3} D_{AZ}(n) \right\rceil
\]

\[
D_{EL}(n) = \left\lceil \frac{16}{3} D_{EL}(n) \right\rceil
\]

where \(\lceil \cdot \rceil\) means the greatest integer in \(\cdot\). Using this preliminary computation, we have

\[
\hat{\dot{\theta}}_{AZ}(n) = \hat{\dot{\theta}}_{AZ}(n-1) + T_s K_{eq} D_{AZ}(n)
\]

\[
\hat{\dot{\theta}}_{EL}(n) = \hat{\dot{\theta}}_{EL}(n-1) + T_s K_{eq} D_{EL}(n)
\]

where

$\hat{\dot{\theta}}_{EL}$ \(\triangleq\) smoothed target inertial LOS elevation rate,

$\hat{\dot{\theta}}_{AZ}$ \(\triangleq\) smoothed target inertial LOS azimuth rate,

$T_s$ = update interval,

$K_{eq}$ = loop constant computed from equation 6-46.
Figure 6-30 \textit{ANGLE AND ANGLE RATE TRACK LOOP FILTER COMPUTER ALGORITHM}

\begin{itemize}
  \item \textbf{ENTER}
  \item Update $T_{BL}$ (Eqn B.3)
  \item Quantize AZ and EL Discriminant to $3/16$ dB (Eqn 6.48)
  \item Transform Body Angular Velocity to LOS Coordinates
  \item Update Smoothed Estimate of Target Inertial LOS AZ EL Rates. (Eqn 6.49)
  \item Update $a$ and $b$ gimbal rates (Eqn 6.50)
  \item Update $a$ and $b$ gimbal positions (Eqn 6.51)
  \item Check Obscuration Zone. (See Figure 5-11)
  \item Transform $\phi$ and $\theta$ to Roll and Pitch rates. (Eqn 6.52)
  \item Transform $a$ and $b$ to roll and pitch angles. (Eqn 6.53)
  \item Check for Angle Ambiguity. (Eqn 5.8)
  \item EXIT
\end{itemize}
The second step is to update the \( \alpha \) and \( \beta \) gimbal rates, \( \dot{\alpha} \) and \( \dot{\beta} \). This is accomplished by subtracting the body inertial angular velocity from the new estimate of the target inertial rate and transforming appropriately. Quantitatively, the new \( \dot{\alpha} \) and \( \dot{\beta} \) estimates are obtained from the expression

\[
\dot{\alpha}(n) = \left[ \omega_{\text{TX}}(n) - \omega_{\text{BX}}(n) \right] / \cos \beta \\
\dot{\beta}(n) = \omega_{\text{TY}}(n) - \omega_{\text{BY}}(n)
\]

where

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

Equations (6.50) are derived in Appendix A, section A.2 (see equations (A.7) and (A.9), respectively).

In the fourth step, we update the \( \alpha \) and \( \beta \) gimbal positions to be used for the next update period. This is easily accomplished by using the digital approximation of the analog integrator illustrated in Figure 6-25 to obtain the expression

\[
\alpha(n) = \alpha(n-1) + T_e \dot{\alpha}(n) \\
\beta(n) = \beta(n-1) + T_e \dot{\beta}(n)
\]

The fifth step is to transform the smoothed estimates of the target inertial LOS azimuth and elevation rates to target inertial roll and pitch rates. This is done using the present values of \( \alpha \) and \( \beta \) and the expression

201
Target Inertial Roll Rate
\[ = -1000.0 \left( T_{BL}(1,1) \hat{8}_{AZ}(n) + T_{BL}(1,2) \hat{8}_{EL}(n) \right) \]

Target Inertial Pitch Rate
\[ = -1000.0 \left( T_{BL}(2,1) \hat{8}_{EL}(n) + T_{BL}(2,2) \hat{8}_{EL}(n) \right) \]

where \( T_{BL} \) is computed using \( \alpha(n-1) \) and \( \beta(n-1) \). The above equations are derived in Appendix B section B.3. The final step is to transform the present values of \( \alpha \) and \( \beta \) gimbal position, \( \alpha(n-1) \) and \( \beta(n-1) \), to target roll and pitch angle in the orbiter body (B) frame. This is accomplished by the following expressions:

Target Roll Angle
\[ = -\tan^{-1} \left[ \frac{T_{BL}(2,3)}{T_{BL}(3,3)} \right] 57.29576 \]

Target Pitch Angle
\[ = -\sin^{-1} \left[ T_{BL}(1,3) \right] 57.29576 \]

which are derived in Appendix B section B.2. Once the new target roll and pitch angles have been computed, any ambiguity in these angles is removed using the relations (5.7).

6.7 RANGE AND RANGE RATE TRACKING MODEL DESCRIPTION

The range and range rate tracking simulation model is functionally identical to the Ku-Band radar range and range rate tracker. Figure 6-31 provides a simplified block diagram of the range and range rate tracking loop model. It is composed of three major algorithms: (1) the signal processor which generates the range and velocity discriminants, (2) a tracking loop filter which uses the range discriminant to produce estimates of the range and range rate, and (3) a velocity processor which uses the velocity discriminant and the rough range rate estimate to produce a very accurate estimate of the target.
Figure 6-31  SIMPLIFIED DIAGRAM OF RANGE AND RANGE RATE TRACKING LOOP

Signal Processor

Range Gates

Doppler Filters

Discriminant Generator

Range Track Loop Filter

Velocity Processor
velocity. The signal processing algorithm which generates the range and velocity discriminants has already been described in section 6.4. Therefore, this subsection will focus on the details of the track filter model and the velocity processor model.

6.7.1 Range Tracker Model Description

The range tracker algorithm is composed of a signal processing and a discriminant generator algorithm and a discrete-time range tracking filter algorithm. The signal processing and range discriminant generation algorithm closely approximate the corresponding function in the Ku-Band radar as discussed in section 6.4. The discrete-time tracking loop filter shown in Figure 6-32 is modeled exactly. This includes quantizing the range discriminant to 3/16 dB, quantizing the output range estimate to 5/16 feet, quantizing the output range rate to \( \frac{5}{(16 \frac{T_a}{T_b})} \) feet per second where \( T_a \) is the update interval, and using the same values for \( m_a \) and \( m_b \), the loop constants. These loop constants were calculated in [23] and are summarized in Table 6-7 for the various operating conditions.

Assumptions. One of the major simplifications in the range tracker involves the filtering at IF and baseband. It is assumed that the IF filters pass the perfect rectangular target return pulses without distortion. Also the baseband (or video) filter impulse response is assumed to be perfectly rectangular and of width equal to the A/D sample interval. Impact of these simplifications should be minimized. The only other assumption that might have some impact on model fidelity is neglecting the quantization noise contributed by the signal processing chain from the A/D to the discriminant generator. This assumption will have varying impact upon the model fidelity, depending upon target return signal strength.
Table 6-7  EQUIVALENT RANGE TRACKING LOOP CONSTANTS $m_a$ and $m_b^*$

<table>
<thead>
<tr>
<th>RANGE INTERVAL, nm</th>
<th>$m_a$</th>
<th>$m_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Passive Mode</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R &gt; 9.5$</td>
<td>16.0</td>
<td>0.25</td>
</tr>
<tr>
<td>$3.8 \leq R \leq 9.5$</td>
<td>16.0</td>
<td>0.5</td>
</tr>
<tr>
<td>$1.9 \leq R \leq 3.8$</td>
<td>16.0</td>
<td>2.0</td>
</tr>
<tr>
<td>$0.95 \leq R \leq 1.9$</td>
<td>8.0</td>
<td>1.0</td>
</tr>
<tr>
<td>$0.42 \leq R \leq 0.95$</td>
<td>8.0</td>
<td>2.0</td>
</tr>
<tr>
<td>$R &lt; 0.42$</td>
<td>0.5</td>
<td>0.125</td>
</tr>
<tr>
<td><strong>Active Mode</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R &gt; 9.5$</td>
<td>4.0</td>
<td>0.25</td>
</tr>
<tr>
<td>$R \leq 9.5$</td>
<td>0.5</td>
<td>0.125</td>
</tr>
</tbody>
</table>
Error Sources. Error sources incorporated into the range tracking model include

- Target-induced errors,
- Thermal noise,
- Discriminant Distortion
- Range bias.

Target induced errors, thermal noise, and discriminant error are included in the range discriminant computation. As noted in the angle tracker model discussion, the target induced errors will be accurate to the extent that the scattering model for the target is accurate. Thermal noise is computed using the model discussed in section 6.4 and derived in Appendix D. Range bias is treated as a fixed number representing errors in the system time delay calibration and time-varying system delays.

Model Performance. Testing of the range tracker model is analogous to the angle tracker model testing. That is, the range discriminant was checked for accurate performance and the tracking loop was tested for proper design and loop constant values. The rules for the range discriminant test were the same as the angle discriminant test: quantization was ignored and the discriminant was computed for several values of range error and SNR. Results of the discriminant computations are shown in Figure 6-33 and agree with the theoretical calculations shown in reference [2].

The second test verifies that the loop is operating as designed and that the constants are correct. It is performed by applying a constant range acceleration to the target and computing the range response. The range tracker should respond with a steady state range estimate bias error that is related to the value of $\beta$ (or $m_B$) and the target range acceleration by the expression (taken from reference [24]).
Figure 6-33. Range Discriminant Test Results.
The test described above was performed under the following assumption: the range tracker was operated in "analog fashion". That is, the discriminant was not quantized and all multiplications and additions were done in floating point arithmetic. The results of accelerating the target at 10 fps² are shown in Figures 6-34 through 6-37 for range intervals possessing different β values. These data are in excellent agreement with the theoretically predicted results, indicating proper operation of the range tracking loop.

6.7.2 Velocity Processor Model Description

Model. The simulation velocity processor algorithm is functionally identical to the Ku-Band radar velocity processor algorithm. A simplified block diagram of the algorithm is shown in Figure 6-38. It is composed of two major tasks: (1) determination of the unambiguous velocity estimate and (2) updating the position of the doppler filter bank.

As shown in Figure 6-38, the first task is accomplished by computing the ambiguous velocity estimate and then using this estimate and the rough range rate estimate \( r \) from the range tracker to determine the unambiguous velocity estimate. Figure 6-39 gives a block diagram of the algorithm used to compute the ambiguous velocity estimate. The basic idea of this algorithm

\[
\text{(6.54) Range Bias Error} = - \frac{a T^2}{\beta}
\]

where

\[
a = \text{value of target range acceleration}, \\
\beta = \frac{m_b \cdot 80}{c t_e \cdot \ln 2}
\]

\[
\rho = \frac{1}{1 + \text{SNR}^{-1}}
\]

\( T_S \) = update interval.
RANGE \( (at \ t = 0) = 0.4\text{nm} \)
ACCELERATED AT \(-10000\text{m}/\text{s}^2\) FOR 5 SECONDS

THEORETICAL RANGE BIAS ERROR = 0.175 FEET
ACTUAL RANGE BIAS ERROR = 0.175 FEET

Figure 8-34. Range Tracking Loop Transient Response for Ranges Less Than 0.42 NM.
Figure 8-35. Range Tracking Loop Transient Response for Ranges 0.42NM<CR<2.95NM.
Figure 6-38. Range Tracking Loop Transient Response for Ranges 0.95<CR<3.8 NM.
Figure 6-37. Range Tracking Loop Transient Response for Ranges 3.8 NM < R < 9.5 NM.
Figure 6-38  KU-BAND RADAR VELOCITY PROCESSOR

- Update Doppler Filter Position
- Form Velocity/On-Target Discriminants
- Ambiguous Velocity Estimate
- Velocity Resolver
- Digital Smoothing Filter

(From Range Tracker)
Figure 6-39  SIMPLIFIED DIAGRAM OF AMBIGUOUS VELOCITY ESTIMATION PROCESS

Integral Number of Filter Widths

\[ \frac{\mu L+1}{n_L} \]

Fractional Number of Filter Widths.

(see Eqn 6.59 and 6.60)
is as follows: compute the integral number of filter widths between zero frequency and the target location, combine it with the fractional filter width that remains, and scale appropriately to obtain the ambiguous velocity estimate. It is noted that the fractional part is determined to an accuracy of $1/128$ of a filter width in all cases.

The ambiguous velocity resolver algorithm is given in Figure 6-40. This algorithm operates using the following principle. The ambiguous velocity is used to give a very accurate location of the target in the doppler filter bank and the rough range rate estimate is used to estimate the integral number $N_a$ of filter banks that the target velocity is removed (either up or down) from zero frequency. The unambiguous velocity is then obtained by combining the fractional filter bank width with the integral number of filter bank widths and scaling appropriately.

It is worth mentioning here that the resolver has some additional protection against inaccurate determinations of $N_a$ (the number of filter bank ambiguities) caused by noisy $\hat{r}$ values, especially when the target velocity falls near either edge of the ambiguous filter bank. The portion of the resolver algorithm that provides this protection is enclosed in dashed lines in Figure 6-40 and works in the following way. If the computed position of the rough range estimate in the ambiguous filter bank, call it $\hat{r}_a$, is more than half a filter bank (16 filters) from the ambiguous velocity estimate, then the ambiguity number $N_a$ is increased or decreased by one, depending upon the sign of the difference between $\hat{v}_a$ and $\hat{r}_a$.

The other major task of the velocity processor algorithm is to update the position of the five adjacent doppler filters, always maintaining the target in the center filter (provided target acceleration is not too great). The initial position of the filter set is determined by the filter in which
target detection occurred. This position is then updated during track using the algorithm shown in Figure 6-41. Depending on the values of the velocity and on-target discriminants, the position can be moved by 0, ± 1, or ± 2 filter widths. The exact decision algorithm is given in the figure.

**Assumptions.** Modeling assumptions that affect the velocity processor are that acceleration of the target is not allowed during a data cycle and quantization error contributed by the signal processing chain from the A/D to the discriminant generator is ignored. Target acceleration during a data cycle causes broadening of the signal energy spectrum (a spreading over the doppler filter outputs), causing some degradation in velocity processor performance. Thus, the zero-acceleration constraint will give an optimistic estimate of performance in those cases where the target is accelerating. The effects of neglecting the quantization error has not been analyzed yet.

**Error Sources Modeled.** Velocity processor error sources include

- Target-induced errors,
- Thermal noise,
- Discriminant distortion.

All three of these errors are included in the computation of the velocity and the on-target discriminants. Target-induced error modeling is achieved in the same manner as in the angle and range tracker models. Thermal noise is injected using the method described in section 6.4 and Appendix D. Discriminant error is generated by using an accurate discriminant computation model.

**Model Performance.** Performance of the ambiguous velocity estimator was tested in the following way. The target estimated ambiguous velocity was computed as a function of target position over a filter width for high and low SNR values. Results of this test are shown in Figure 6-42 and agree with the theoretically predicted performance given in [25].
Figure 6-41  FILTER POSITION UPDATE ALGORITHM

1. Enter
2. Do $P_T$ equal T?
   - Yes
   - No $P_V < 51$
   - No $P_V > 51$
3. No $P_V < 0$
   - Yes $n_L(i+1) = n_L(i) + 1$
   - No $n_L(i+1) = n_L(i) - 1$
4. No $n_L(i+1) = n_L(i) + 2$
   - Yes $n_L(i+1) = n_L(i) - 2$
5. Exit
Figure 6-42. Velocity Discriminant Test Results.

200 TRIALS
POINT TARGET
\( R_0 = 1 \text{m}^2 \)
SNR REFERENCED TO DOPPLER FILTER OUTPUT
The ambiguity resolver has not been tested. However, it is noted that for ranges less than 12 nm (7 kHz PRF) the error in the rough range rate estimate would have to exceed ±123 feet per second before the ambiguity number is in error.

6.7.3 Computer Algorithm Details

Construction of the range and range rate tracking loop computer model is identical to the angle and angle rate tracking loop computer model. That is, the computer model is broken into two distinct parts. One set of algorithms is dedicated to generation and processing of the target return signal to produce the required discriminants. These algorithms were described in section 6.4. The other part of the model is dedicated to updating of the range and velocity estimates and updating of internal control parameters. This part is described in this subsection.

Figure 6-43 gives the range tracker and velocity processor computer model. This algorithm is divided into four tasks: (1) updating the tracking loop filter difference equations which give the latest estimate of the range and rough range rate, (2) ambiguous velocity determination, (3) unambiguous velocity determination and (4) updating of the system internal control parameters. Each of these tasks are described in detail below.

Range and Rough Range Rate Estimate Update. The first step is to quantize the range discriminant to 3/16 dB using

\[
D_R(n) = \left\lfloor \frac{16}{3} D_R(n) + \frac{1}{2} \right\rfloor
\]

(6.55)

where \(\lfloor \cdot \rfloor\) means take the greatest integer in \(\cdot\). Then, the range and range rate estimates are updated using the difference equations.
Figure 6-43 RANGE AND RANGE RATE TRACKING LOOP COMPUTER ALGORITHM
(1 of 2)

1. **ENTER**
   - Quantize Range Discriminant to 3/16 dB Accuracy
   - Update Range Rate Estimate (Eqn 6.56)
   - Update Range Estimate (Eqn 6.57)
   - Scale Range Estimate to Feet
   - Add Fixed Range Bias
   - Range Track Loop Filter

2. **Quantize Velocity Discriminant To 3/16 dB Accuracy**
   - Compute Integral Number of Filter Widths In Ambiguous Velocity (Eqn 6.58)
   - Compute Fractional Portion of Ambiguous Velocity. (Eqn 6.59)
   - Combine Integral And Fractional Parts To Form Ambiguous Velocity Est. Scaled to 1/128 of a Filter Width

3. **Ambiguous Velocity Estimator**

---

222
Figure 6-43  RANGE AND RANGE RATE TRACKING LOOP COMPUTER ALGORITHM
(2 of 2)

3

Compute Number Of Filter Bank Ambiguities in R. (Eqn 6.61)

Compute Position Of $\hat{R}$ in Ambiguous Filter Bank. (Eqn 6.62)

Form: $\Delta = \frac{R_A - V_A}{V_B}$

$|\Delta| < \frac{1}{4}$?

no

$\Delta \Delta = N_a - N_a - 1$

yes

$\Delta \Delta = N_a - N_a + 1$

Compute Unambiguous Velocity Estimate. (Eqn 6.64)

Compute Smoothed Unambiguous Velocity Estimate. (Eqn 6.65)

4

Update Position Of 5 Track Filters (Eqn 6.66 or Figure 6-41)

Update Range Interval Control Parameter (Eqn 6.67)

Update Sample Rate Control Parameter (Eqn 6.68)

Update PRF Control Parameter (Eqn 6.69)

EXIT

System Control Parameter Update
\( R(n) = \dot{R}(n-1) + m \Delta R(n) \)  
\( R(n) = \dot{R}(n-1) + \dot{R}(n) + m_a \Delta R(n) \)

where it should be noted that \( R \) is scaled to 5/16 feet and \( \dot{R} \) is scaled to 5/(16 \( T_s \)) feet per second. The last step is to scale the range estimate to feet and add a fixed range bias to form the radar predicted range estimate.

**Ambiguous Velocity Estimator.** In the first step the velocity discriminant is quantized to 3/16 dB by replacing \( D_R \) by \( D_V \) in equation (6.55). Next the integral number of filter widths between zero frequency and the target location in the doppler filter bank is updated using the equation

\[
\text{Integral Number of Filter Widths} = \begin{cases} 
m_L, & \text{if } D_V > 0 
m_c, & \text{if } D_V < 0 
\end{cases}
\]

where

\[ m_L = \text{filter number of low filter} \]  
(see Figure 6-12),

\[ m_c = \text{filter number of center filter}. \]

Then, the fractional filter width remainder is determined to 1/128 of a filter width accuracy in the following way:

\[
\text{Fractional Filter Width Remainder} = \begin{cases} 
F(D_V), & \text{if } D_V \geq 0 
1-F(D_V), & \text{if } D_V < 0 
\end{cases}
\]

where the function \( F \) is shown in Figure 6-44. This function is predetermined using the expression

\[
D_V = \frac{160}{3} \log \left( \frac{\sin 16X_L \sin X_H}{\sin X_L \sin 16X_H} \right)
\]
Figure 5-44. Fractional Filter Width as a Function Velocity Discriminant Value.
where
\[ x_L = \tau(-\frac{1}{32} - ft_p) \]
\[ x_H = \tau(-\frac{1}{32} - ft_p) \]
\[ t_p = PRI \]
\[ \lceil \cdot \rceil = \text{greatest integer in } \cdot \]

to associate a \( D_v \) value with each of the following values of \( f \):
\[
(0, \frac{1}{256t_p}, \frac{1}{256t_p}, ..., \frac{127}{256t_p})
\]
These values are stored in the computer in look-up table fashion. The last step is to compute the ambiguous velocity estimate by adding the results of equations (6.58) and (6.59).

**Velocity Resolver.** The first task is to compute the number \( N_a \) of ambiguous doppler filter bank widths in \( \hat{R}(n) \). This is achieved using

\[
N_a = \left\lfloor \frac{\hat{R}(n)}{V_B} \right\rfloor
\]

where \( V_B \) is the maximum unambiguous velocity. Then, \( N_a \) is checked for accuracy using the following procedure: the position of \( \hat{R}(n) \) in the ambiguous filter bank, call it \( \hat{R}_a(n) \), is computed using the equation

\[
\hat{R}_a(n) = \text{mod} (\hat{R}(n), V_B)
\]

and is compared to the ambiguous velocity \( V_a \) obtained from the first step. The ambiguity number is corrected, depending upon the result of this comparison, as follows

\[
N_a = \begin{cases} 
N_a + 1, & \text{if } \hat{R}_a - V_a \leq -\frac{V_B}{2} \\
N_a, & \text{if } -\frac{V_B}{2} < (\hat{R}_a - V_a) < \frac{V_B}{2} \\
N_a - 1, & \text{if } \hat{R}_a - V_a \geq \frac{V_B}{2}
\end{cases}
\]
Once the ambiguity number has been correctly determined, it is combined with the result from step one to obtain the unsmoothed, unambiguous velocity estimate, $V_u(n)$, i.e.

$$V_u(n) = V_a(n) + N_a V_B.$$  \hspace{1cm} \text{(6.64)}

The final step is to pass this value of $V_u(n)$ through a digital smoothing filter. This filter is a moving window average which averages the previous three $V_u$ values with the present value. Quantitatively, we have

$$V_a(n) = \frac{1}{3} \sum_{i=n-3}^{n} V_u(i).$$  \hspace{1cm} \text{(6.65)}

**Internal Control Parameter Update.** Based on the new estimates of the range, the velocity discriminant and on-target discriminant the following internal controls are updated: (1) filter bank position, (2) the range interval parameter, MRNG, (3) the PRF parameter, MPRF and (4) the sample rate parameter, MSAM. The filter position update requires the on-target and velocity discriminant values and the following algorithm:

$$m_c = \begin{cases} 
  m - 2 & \text{if } D_V > 0 \text{ and } D_{OT} < T, \\
  m - 1 & \text{if } D_V > 51 \text{ and } D_{OT} \geq T, \\
  m + 0 & \text{if } |D_V| \leq 51 \text{ and } D_{OT} \geq T, \\
  m + 1 & \text{if } D_V < -51 \text{ and } D_{OT} \geq T, \\
  m + 2 & \text{if } D_V < 0 \text{ and } D_{OT} < T.
\end{cases}$$  \hspace{1cm} \text{(6.66)}

The range interval parameter MRNG is determined by finding the integer $i$ such that
\[ R_{i-1} \leq R(n) < R_i \]

where the \( R_i \) are listed in Table 6-4. MSAM is computed using the following algorithm:

\[
MSAM = \begin{cases} 
1, & \text{if } \text{MRNG} < 9 \text{ and } \text{IMODE} = 1 \\
& \text{or } \text{MRNG} < 4 \text{ and } \text{IMODE} = 2 \\
2, & \text{if } \text{MRNG} > 9 \text{ and } \text{IMODE} = 1 \\
& \text{or } \text{MRNG} > 4 \text{ and } \text{IMODE} = 2
\end{cases}
\]

(6.68)

Finally, the PRF parameter, MPRF, is updated by:

\[
MPRF = \begin{cases} 
1, & \text{if } \text{MRNG} < 9 \text{ and } \text{IMODE} = 1, \\
& \text{or } \text{MRNG} < 9 \text{ and } \text{IMODE} = 2, \\
2, & \text{if } \text{MRNG} > 9 \text{ and } \text{IMODE} = 2, \\
3, & \text{if } \text{MRNG} > 9 \text{ and } \text{IMODE} = 1
\end{cases}
\]

(6.69)

The values for MRNG, MSAM and MPRF as a function of range interval and system mode are summarized in Table 6-4.
7. RECOMMENDATIONS FOR FURTHER STUDY AND DEVELOPMENT

7.1 SYSTEM ANALYSIS

The present computer simulation model is a very useful tool for evaluation of the Ku-Band Radar track mode design. As an example of a useful system analysis where the model can immediately be applied, consider the following problem. At the present, it is not clear that one should PDI over all five frequencies in the track mode. Instead, it has been conjectured that performance would be improved by selecting the largest return of the five frequencies, especially when the return signal is weak and the target scattering properties are sensitive to small changes in transmit frequency. In this case, the computer simulation model can easily be adapted to perform an analysis of this problem.

7.2 RADAR MODEL FIDELITY IMPROVEMENT

Some of the areas where the radar simulation model may be improved are

- reducing computation time,
- discriminant model accuracy,
- AGC model accuracy,
- search model fidelity.

Reducing computation time is always desirable, since it will provide room for improvement in the model accuracy. For example, a reduction in computation time would allow us to use a more accurate discriminant generation model (see Appendix C). An accurate AGC model will not consume an appreciable amount of computation time, but it will require a significant amount of time to develop, install, and test an accurate algorithm. Accurate AGC estimates would be useful in predicting radar performance when a target fades rapidly and providing accurate signal strength estimates under weak target (low SNR) conditions. Although the search model has enough fidelity to provide adequate crew training, significant improvements can be made in this area if desired.
7.3 TARGET MODEL FIDELITY IMPROVEMENT

If the target scattering measurements recommended in section 4.3.5 are performed, then it would be very useful to correlate these data with the predictions of the present target model and, if feasible, make the necessary adjustments in the present model.
REFERENCES AND BIBLIOGRAPHY


APPENDIX A

DERIVATION OF ANGLE AND ANGLE RATE TRACKING LOOP

MODEL INITIALIZATION

The purpose of this appendix is to derive the equations used to initialize (1) the target inertial LOS azimuth and elevation rates and (2) the α1 and β gimbal rates. Fundamental to both derivations is the following fact taken from [28]. Consider two reference frames A and B with a common origin. Suppose B is rotating uniformly with angular velocity \( \mathbf{\omega} \) with respect to A. Then the velocity of a free point target as measured by an observer fixed in frame A is related to the velocity of an observer fixed in frame B by the equation

\[
\mathbf{v}^{|A|} = \mathbf{v}^{|B|} + \mathbf{\omega} \times \mathbf{r}
\]

where
- \( \mathbf{v}^{|A|} \) = velocity measured in the A frame,
- \( \mathbf{v}^{|B|} \) = velocity measured in the B frame,
- \( \mathbf{r} \) = position vector of the point target,

and all of the vectors in equation (A.1) are expressed in the same, but arbitrary, coordinate system centered at origin of the A (or B) frame.

An important assumption that is used in both derivations is that the target c.g. is assumed to be on the antenna boresight axis, or, equivalently, the negative z axis of the L-frame at the time of initialization. Thus, the position vector \( \mathbf{r}^{|L|}_o \) has the form

\[
\mathbf{r}^{|L|}_o = \begin{pmatrix} 0 \\ 0 \\ -z^{|L|}_o \end{pmatrix}
\]

A.1 DERIVATION OF TARGET INERTIAL LOS AZIMUTH AND ELEVATION RATE INITIALIZATIONS

Using the assumption stated in the previous paragraph, we can define the target inertial LOS azimuth and elevation rates by the expressions
Inertial LOS \( \Delta w^L_{TX} = \frac{v^L_{OY}}{r^L_o} \)

Azimuth Rate \( w^L_{TX} = \frac{v^L_{OY}}{r^L_o} \)

(Equation A.3)

Inertial LOS \( \Delta w^L_{LY} = -\frac{v^L_{OX}}{r^L_o} \)

Elevation Rate \( w^L_{LY} = -\frac{v^L_{OX}}{r^L_o} \)

where \( v^L_o \) = velocity of target c.g. as measured in the inertial frame and expressed in LOS coordinates.

We can now begin the derivation. Given that the orbiter body has the inertial angular velocity \( w_B \), equation (A.1) can be written

\[ \dot{v}^L_o = \dot{v}^L_B + \omega_B \times r^L_o \]

Using the assumption given in equation (A.2), the x and y components of \( \dot{v}^L_o \) can be written

\[ v^L_{OX} = v^L_{OX} - w_{By} \frac{r^L_o}{r^L_o} \]

(Equation A.4)

\[ v^L_{OY} = v^L_{OY} + w_{Bx} \frac{r^L_o}{r^L_o} \]

Dividing equations (A.4) by \( r^L_o \) and using the definitions of target inertial LOS rates given in equations (A.3), we obtain

Inertial LOS \( \Delta w^L_{TY} = \frac{v^L_{OX} B}{r^L_o} + w^L_{By} \)

Elevation Rate \( w^L_{TY} = \frac{v^L_{OX} B}{r^L_o} + w^L_{By} \)

(Equation A.5)

Inertial LOS \( \Delta w^L_{TX} = \frac{v^L_{OY} B}{r^L_o} + w^L_{Bx} \)

Azimuth Rate \( w^L_{TX} = \frac{v^L_{OY} B}{r^L_o} + w^L_{Bx} \)
A.2 DERIVATION OF $\alpha$ AND $\beta$ GIMBAL RATE INITIALIZATIONS

The $\alpha$ gimbal rate is defined as the rate of rotation of the outer gimbal (or G) frame about the x-axis of the R frame. If we assume the rotation is uniform, then from equation (A.1) we have

\[ \dot{v}_o^G = \dot{v}_o^G + \left( \begin{array}{c} \dot{\alpha} \\ 0 \\ 0 \end{array} \right) \times \vec{r}_o^G. \]

Noting that

\[ \vec{r}_o^G = T_{GL} \vec{r}_o^L = \left( \begin{array}{ccc} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{array} \right) \vec{r}_o^L, \]

or

\[ \vec{r}_o^G = \left( \begin{array}{c} -|\vec{r}_o^L| \sin \beta \\ 0 \\ -|\vec{r}_o^L| \cos \beta \end{array} \right), \]

and writing out the y-component of equation (A.6), we then have

\[ v_{oy}^G = v_{oy}^G + \dot{\alpha} |\vec{r}_o^L| \cos \beta. \]

But the y-component of the target velocity as measured in the G-frame and expressed in the G-frame coordinates, i.e. $v_{oy}^G$, is zero. Therefore

\[ \dot{\alpha} = \frac{v_{oy}^G}{|\vec{r}_o^L| \cos \beta} = \frac{v_{oy}^L}{|\vec{r}_o^L| \cos \beta} = \frac{\omega_{lx} - \omega_{bx}}{\cos \beta}, \]

The $\beta$ gimbal rate, $\dot{\beta}$, is defined as the rate of rotation of the inner gimbal (or L) frame about the outer gimbal (or G) frame y-axis. Using this fact and equation (A.1), we obtain

\[ \dot{v}_o^G = \dot{v}_o^G + \left( \begin{array}{c} 0 \\ \dot{\beta} \\ 0 \end{array} \right) \times \vec{r}_o^G. \]

Noting that $v_{o_L}^G = 0$ by assumption and substituting the resultant expression
for $v_0^G$ into equation (A.6), gives

$$v_0^G = \begin{pmatrix} \frac{\dot{\alpha}}{\alpha} \\ \frac{\dot{\beta}}{\beta} \\ 0 \end{pmatrix} \cdot \vec{r}_o^G$$

Transforming to L-frame coordinates,

$$(A.8) \quad \dot{v}_o^L = T_{LG} \left( \begin{array}{c} \frac{\dot{\alpha}}{\alpha} \\ \frac{\dot{\beta}}{\beta} \\ 0 \end{array} \right) \times T_{LG} \vec{r}_o^G$$

$$= \begin{pmatrix} -\frac{\dot{\beta}}{\beta} \frac{\dot{r}_o^L}{\alpha} \\ \frac{\dot{r}_o^L}{\alpha} \cos \beta \\ 0 \end{pmatrix}$$

The expression for $\beta$ can be obtained from the x-component of equation (A.8).

It is

$$(A.9) \quad \dot{\beta} = -\frac{v_{ox}^L}{\alpha} \vec{r}_o^L = \frac{L}{\alpha} T_y^L - \omega_{By}$$

where equation (A.5) was used to obtain the last equality.
DERIVATION OF TARGET PITCH ANGLE, ROLL ANGLE, INERTIAL ROLL RATE, AND INERTIAL PITCH RATE TRANSFORMATIONS

This appendix presents the derivations of (1) the transformation of $\alpha$ and $\beta$, which are tracked by the radar, to roll and pitch angles in the Orbiter Body (B) frame and (2) the transformation of the target inertial LOS azimuth and elevation rates, which are estimated by the radar, to target inertial roll and pitch rates in the B frame.

B.1 DEFINITIONS AND ASSUMPTIONS

We first provide definitions of all quantities which are pertinent to the derivations given below. The $\alpha$ and $\beta$ gimbal angles were defined in Section 2.1, while the roll and pitch angles are defined as follows:

- **Target Roll Angle** is the angle between the $-Z_B$ axis and the projection of the target direction vector on the $Z_B-Y_B$ plane as shown in Figure B-1.

- **Target Pitch Angle** is the angle between the target direction vector and the projection of the target direction vector on the $Z_B-Y_B$ plane.

Quantitatively, these definitions can be expressed as

\[
\text{Roll angle } \theta = \tan^{-1} \left( \frac{\hat{r}_o \cdot \hat{z}_B}{\hat{r}_o \cdot \hat{y}_B} \right) \\
\text{Pitch angle } \phi = \sin^{-1} \left( \frac{\hat{r}_o \cdot \hat{x}_B}{\hat{r}_o \cdot \hat{z}_B} \right)
\]

where $\hat{r}_o$ = unit vector in direction of the target, $\hat{x}_B, \hat{y}_B, \hat{z}_B$ = unit vectors along the $X_B, Y_B, Z_B$ axis of the B-frame, respectively.

The target inertial LOS azimuth and elevation rates were defined in
Appendix A equation (A.3). Inertial roll and pitch rate are defined as

- **Inertial Roll Rate** is the projection of the target inertial angular velocity (estimated by the radar) along the $X_B$-axis.

- **Inertial Pitch Rate** is the projection of the target inertial angular velocity along the $Y_B$-axis.

Again, mathematically we have

\[
\text{Inertial Roll Rate } \hat{\omega}_T \cdot \hat{X}_B,
\]

\[
\text{Inertial Pitch Rate } \hat{\omega}_T \cdot \hat{Y}_B.
\]

There are two basic assumptions that were made in the development of the required transformations. These are that

1. the radar is located at the origin (or C.G.) of the B-frame, i.e. no offset,
2. the $67^\circ$ yaw angle between the B and R frames is assumed to be exact, i.e. no boom deployment error.

With these assumptions under our belt we can define one last, but very useful, quantity. The transformation matrix $T_{BL}$, which transforms a vector expressed in L coordinates to a vector expressed in B coordinates (see section 2), is defined by

\[
T_{BL} = \begin{pmatrix}
c y & -s y & 0 \\
s y & c y & 0 \\
0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
1 & 0 & 0 \\
0 & c a & -s a \\
0 & s a & c a
\end{pmatrix} \begin{pmatrix}
c a & 0 & s b \\
s a & 0 & c b \\
0 & 1 & 0
\end{pmatrix}
\]

\[
(B.3)
\]

where $c = \cos$, $s = \sin$, $y = +67^\circ$.

B.2 DERIVATION OF TARGET ROLL AND PITCH ANGLE TRANSFORMATIONS

As mentioned in the introduction we are given the $\alpha$, $\beta$ angles and
desire to convert these angles to roll and pitch. Consider the following argument. \( \hat{r}_o \), the unit vector in the direction of the target, lies along the \( Z_L \)-axis in the LOS frames. This can be written

\[
\hat{r}_o^L = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}
\]

Transferring this vector to the Body Frame coordinates, we have

\[
T_{BL} \hat{r}_o^L = \begin{pmatrix} -T_{BL} (1,3) \\ -T_{BL} (2,3) \\ -T_{BL} (3,3) \end{pmatrix}
\]

Using equation (B.4) and the definitions of roll and pitch angles given in equations (B.1), we obtain

\[
\begin{align*}
\text{Roll angle} &= -\tan^{-1} \left[ \frac{T_{BL}(2,3)}{T_{BL}(3,3)} \right] \\
\text{Pitch angle} &= -\sin^{-1} \left[ \frac{T_{BL}(1,3)}{} \right].
\end{align*}
\]

B.3 DERIVATION OF TARGET INERTIAL ROLL AND PITCH RATE TRANSFORMATIONS

In this case, the radar estimates the components of the target inertial angular velocity vector in the LOS frame, and we would like this vector transformed to the B-frame coordinates. The argument begins by noting that in the LOS frame the \( Z_L \)-component is always zero. That is,

\[
\hat{w}_T^L = \begin{pmatrix} \hat{w}_T^T \\ \hat{w}_T^{LX} \\ \hat{w}_T^{LY} \end{pmatrix} = \begin{pmatrix} \theta_{AZ} \\ \theta_{EL} \\ 0 \end{pmatrix}.
\]

Transferring this vector to body coordinates, we have
Using equation (B.6) and the definition of roll and pitch rate given in equations (B.2), target roll and pitch rates can be written as

\[
\begin{align*}
\text{Target Roll Rate} &= - \left[ T_{BL} (1,1) \dot{\theta}_{AZ} + T_{BL} (1,2) \dot{\theta}_{EL} \right] \\
\text{(B.7)}
\end{align*}
\]

\[
\begin{align*}
\text{Target Pitch Rate} &= - \left[ T_{BL} (2,1) \dot{\theta}_{AZ} + T_{BL} (2,2) \dot{\theta}_{EL} \right] .
\end{align*}
\]
APPENDIX C

DERIVATION OF THE NOISE-FREE DISCRIMINANT COMPONENTS

COMPUTATION MODEL

This appendix provides a derivation of the noise-free discriminant component (see section 6.4.3 for a definition of discriminant component) computation model. A simplified diagram of the model, illustrated in Figure C-1, shows that each of the noise-free discriminant components is computed at the PDI output. Derivation of this model is structured as follows. First, the complete target response representing the i th frequency, the j th time slot, the 1 th range gate, and the n th doppler filter is computed at the magnitude-detector output and then the individual noise-free discriminant components are formed by summing (PDIing) the magnitude-squared detected response over the appropriate i, j, 1, and n indices.

C.1 MODEL ASSUMPTIONS

Assumptions used in the development of the computational model are listed in section 6.4.1. Rather than repeating the list here, use of each of the assumptions will be noted at the appropriate point in the derivation.

C.2 NOISE-FREE MAGNITUDE-SQUARED DETECTOR RESPONSE DERIVATION

The development of the magnitude-squared detector response is broken into several steps. These are (1) compute the doppler filter response for a single point scatterer, (2) using the assumed linearity of the processor from the antenna to the doppler filter, compute the complete target response by vectorial summation of the individual responses, and (3) compute the magnitude-square of the result. These steps are illustrated in Figure C-1 and described in detail below.

Doppler Filter Response For a Single Scatterer. We first write the expression for the k th target response at the baseband filter input. This response represents that portion of the received waveform associated with the entire j th time slot at the i th frequency (see Figures 6-2 and 6-3). The expression
Figure C-1  SIMPLIFIED DIAGRAM OF THE NOISE-FREE DISCRIMINANT COMPONENT COMPUTATION MODEL

Return From Point Target \#1

Antenna Sum/Diff Weighting → Video Filter → Range Gate/Presum Weighting → Doppler Filter Weighting

...\...

Return From Point Target \#N T

Antenna Sum/Diff Weighting → Video Filter → Range Gate/Presum Weighting → Doppler Filter Weighting

Magnitude-Squared Detector → PDI

[Appropriate Magnitude-Squared Data] → PDI → [Noise-Free Discriminant Component] → To PDI
for this response is obtained by starting with the point scatterer's single pulse return at the antenna output terminals given in equation (4.1) and applying assumptions (1) through (5) of section 6.4.1. This gives

\[ S_k(t) = A_k \sigma_k^i \left( \rho_{sk} + \rho_{dkj} \right) \sum_{m=0}^{15} \exp \left( j \left( 2\pi f_k t - \delta_{ki} \right) \right) \exp \left( \frac{t - t_p - t_k}{\tau_c} \right) \]

where

- \( A_k \) is defined in equation (4.1),
- \( \rho_{sk} \) = antenna sum pattern weighting for \( k \)th scatterer,
- \( \rho_{dkj} \) = antenna difference pattern weighting for \( k \)th scatterer and \( j \)th time slot,
- \( \sigma_k \) = RCS for \( k \)th scatterer,
- \( f_k \) = \( k \)th target doppler shift,
- \( \tau_c \) = transmit pulse width,
- \( t_p \) = PRI,
- \( \tau_k \) = \( 2 \left( R_k^L - R_0^L \right) / c \),
- \( \delta_{ki} \) = \( 4\pi(R_k^L - R_0^L) / \lambda_i \),
- \( \lambda_i \) = wavelength associated with \( i \)th transmit frequency,
- \( P(t) = \begin{cases} 1, & 0 < t \leq 1 \\ 0, & \text{otherwise} \end{cases} \)

The next step is to compute the response of the presummer to the \( m \)th pulse in the above expression. This computation includes several intermediate steps: baseband filtering, sampling, range gating, and, finally, presumming. Assumptions (6) through (8) are used in the filtering, sampling, and ranging gating process. The result of this process is best described by the illustration provided in Figure C-2, showing the sampled pulse response with respect to the early and...
Figure C-2. Illustration of the Result of the Filtering, Sampling, and Range Gating Process
late range gates. With these assumptions, the general response of the presummer for the $i$ th frequency, the $j$ th time slot, the $l$ th range gate, and the $m$ th pulse is

\begin{equation}
S_k(i,j,l,m) = A_k e^{j\frac{\theta_k}{d_k}} e^{j2\pi f_k m t_p + \phi_k}
\end{equation}

\begin{equation}
= \sum_{n=1}^{N_p} r_{k,l}(n) \exp\left(j2\pi f_k \left(n + (f-3/2) N_p - 1/2\right) \tau - \tau_k\right)
\end{equation}

where $r_{k,l}(n)$ is the magnitude of the $n$ th sample in the $l$ th range gate and depends upon the position of the filtered pulse in the range gate as illustrated in Figure C-2. Quantitative expression of each $r_{k,l}(n)$ is delayed until the following approximation is stated:

**Approximation:** It is assumed that the phase progression over a pulsewidth can be ignored.

Using this approximation, the summation in (C.2) simplifies to

\begin{equation}
\left[\text{Summation in C.2}\right] = \sum_{n=1}^{N_p} r_{k,l}(n)
\end{equation}

\begin{equation}
= N_p R_l(t_k)
\end{equation}

and $R_l(t_k)$ is defined by

\begin{equation}
R_l(t_k) = \begin{cases} 
0, & \text{if } \Delta \leq -3 \text{ or } \Delta \geq 1 \\
\frac{3+\Delta}{2}, & \text{if } -3 \leq \Delta \leq -1 \\
\frac{1-\Delta}{2}, & \text{if } -1 \leq \Delta \leq 1 
\end{cases}
\end{equation}

(Early Gate)
or

\[
R_2(t_k) = \begin{cases} 
\frac{1+\Delta}{2}, & \text{if } -1 \leq \Delta \leq 1 \\
\frac{3-\Delta}{2}, & \text{if } 1 \leq \Delta \leq 3 \\
0, & \text{if } \Delta > 3 \text{ or } \Delta < -1
\end{cases}
\]  

(Late Gate)

where \( \Delta = t_k/t_t \). It is noted that the approximation given above is excellent for all short pulse modes. However, it may introduce some degradation in the long range case where large pulsewidths are used.

Calculation of the \( n \)th doppler filter response to the \( k \)th scatterer is easily accomplished by using equation (C.2) and (C.3)(or (C.4)) and forming the summation

\[
S_k(i,j,l,n) = \sum_{m=0}^{15} S_k(i,j,l,m) \exp \left(-j \frac{2\pi mn}{32}\right).
\]

Performing the summation, we obtain

\[
S_k(i,j,l,n) = C_k(i,j,l) \frac{\sin(16z_k)}{\sin(z_k)} \exp \left(-j15z_k\right)
\]

where

\[
C_k(i,j,l) = A_k \sigma_k \delta_{sk} \delta_{dkj} \frac{\rho}{p} \exp \left(-j2\pi\phi_{k1}\right),
\]

\[
z_k = \pi\left(\frac{n}{32} - f_k f_p\right).
\]

**Magnitude-Squared Detector Response.** The magnitude-squared detector response is obtained by vectorially summing the doppler filter responses of all \( N_p \) scatterers using the assumed linearity of the processor, and then squaring the magnitude of the resultant sum. The result of these steps is the expression
C.3 DISCRIMINANT COMPONENT COMPUTATION

This subsection derives the closed-form expression used to model each of the three discriminant types: angle, range, and velocity.

C.3.1 Angle Discriminant Component Computation

The angle discriminant component corresponding to the $j$th time slot is obtained by performing a post-detection summation of the energy from the center doppler filter ($n = m_c$) over $N_F$ frequencies and both range gates. This gives the expression

$$D_{Aj} = \sum_{i=1}^{N_F} \sum_{l=1}^{2} S(i, j, l, m_c).$$

Practical Aspects of Computer Implementation. In order to reduce the amount of computation by a factor of two, the following approximation was used.

Approximation: For a given pulse from a single target, the early and late gate presum weights are equal and are given by $\frac{1}{2} \left[ R_1(t_k) + R_2(t_k) \right]$. However, the phase associated with the true position in the range gate is retained.

This approximation has the effect of altering the form of $C_k(i, j, l)$ used in equation (C.8). These coefficients now have the form

$$C_k(i, j, l) = A_k \sigma_k \left[ \rho_{sk} + \rho_{dkl} \right] N_p \left[ R_1(t_k) + R_2(t_k) \right] / 2 \exp(-j 2 \phi_{kl}).$$

We note that the approximation becomes exact when the target is composed of a single point scatterer. However, for a multiple point target, this approximation may be invalid, especially if the range tracker does not keep the return
pulses close to the center of the range gates.

As mentioned above the original motivation for this approximation was to insure adequate computation speed. If it turns out that there is room for additional computation after the target has been represented adequately, then this approximation will be abandoned.

C.3.2 Range Discriminant Component Computation

The range discriminant component corresponding to the 1th range gate is obtained by performing a post-detection summation of the energy from the center doppler filter over $N_F$ frequencies and four time slots. The expression for the range gate discriminant component is

\[
D_{R_f} = \sum_{i=1}^{N_F} \sum_{j=1}^{4} S(i,j,f,m_c).
\]

Practical Aspects of Computer Implementation As in the angle discriminant case, we desire to speed the computation by making approximations in $D_{R_f}$. In this case, we make the following approximation

**Approximation:** $p_{dkj}$ are identically zero for all $k$ and all $j$.

In effect, this approximation makes the assumption that the angle tracker is working perfectly. The result of the approximation is to alter the $C_k$'s as follows

\[
C_k(i,j,f) = A_k \sigma_k \rho_{sk} N_p R_l(t_i) \exp \left[ -j2\pi f_k i \right].
\]

C.3.3 Velocity Discriminant Component Computation

We note that the velocity discriminant components and the on-target discriminant components are computed in an identical manner. Therefore, only the velocity discriminant component computation is described. The velocity
discriminant component corresponding to the \( m_L \) (or \( m_H \)) filter is obtained by performing a post-detection summation of all energy from the \( m_L \) (or \( m_H \)) filter. This can be expressed as

\[
D_{VL} = \sum_{i=1}^{N_F} \sum_{j=1}^{4} \sum_{l=1}^{2} S(i, j, l, m_L).
\]

Practical Aspects of Computer Implementation. To enhance the computer speed in this case we use both approximations stated above for the angle discriminant and range discriminant. Therefore, the \( C_k(i, j, l) \) for equation (C.12) are given by

\[
C_k(i, j, l) = A_k \sigma_k^2 \rho_k N_p \left[ R_1(t_k) + R_2(t_k) \right] / 2 \left\{ \exp(-j2\pi \phi_{kl}) \right\}. \]
APPENDIX D

DERIVATION OF THE THERMAL NOISE MODEL

As described in section 6.4, the computational model for the noisy discriminant values generates the noise-free target response at the PDI output and adds the equivalent thermal noise sample, obtained from the appropriate statistics, to the noise-free value. This model is illustrated in Figure D-1. Motivation for injecting the noise at this point, rather than at the signal processor input or some intermediate point was to enhance the real-time processing capability of the track mode. That is, it was desired to maximize the number of point scatterers allowed in the target model. The purpose of this appendix is to demonstrate that the equivalent noise can be represented as an additive noise process and to derive the statistical characteristics, i.e. the mean, the variance, and the probability density function (pdf) for each member of this random sequence.

D.1 MODEL ASSUMPTIONS

Derivation of the noise model is based upon the following set of assumptions. The primary assumption is that the form of the signal, including thermal noise, at the doppler filter output is given by the expression

\[ v(n) = v_I(n) + v_Q(n) = (S_I(n) + n_I(n)) + j (S_Q(n) + n_Q(n)). \]

\( S_I(n) \) and \( S_Q(n) \) are the in-phase and quadrature components of the noise-free target response at the doppler filter output for the \( n \)th time sample. The quantities \( n_I(n) \) and \( n_Q(n) \) are the in-phase and quadrature components of the thermal noise process for the \( n \)th time sample. These components are assumed to have the following statistical characteristics:

1. both are Gaussian random sequences,
2. \( n_I(n), n_Q(n) \) are statistically independent for all values of \( n \),
3. \( n_I(i), n_I(j) \) (and \( n_Q(i), n_Q(j) \) are statistically independent
Figure D-1  ILLUSTRATION OF MODEL WHICH GENERATES NOISY DISCRIMINANT COMPONENT

[Diagram showing the process from Antenna Weighting to Doppler Filter, with additional blocks for Data From, PDI, and Noisy discriminant Component to log Converter, and a note for equivalent additive noise.]
for all values of $i, j$ such that $i \neq j$.

(4) the mean and variance of $n_I, n_q$ are

$$m_I = m_q = 0$$

$$\sigma^2_I = \sigma^2_q = \sigma^2_o$$

The last assumption is that all signal processor quantization effects are ignored.

D.2 NOISE MODEL DERIVATION

D.2.1 Derivation of Mean and Variance at PDI Output

We begin the derivation by calculating the output of the magnitude-squared detector when the sequence of equation (D.1) is applied at the input. The resulting output is given by the expression,

$$X(n) = \left| v(n) \right|^2 = v_I^2(n) + v_q^2(n)$$

$$= (S^2_I + 2S_I n_I + n_I^2) + (S^2_q + S_q n_q + n_q^2)$$

$$= (S^2_I + S^2_q) + (2S_I n_I + 2S_q n_q + n_I^2 + n_q^2)$$

Computing the mean of $X(n)$, we have

$$\bar{X}(n) = \left| \overline{v(n)} \right|^2 = \overline{S^2_I} + \overline{S^2_q} + 2\overline{S_I n_I} + 2\overline{S_q n_q} + \overline{n_I^2} + \overline{n_q^2}$$

where the bar over a quantity means to compute the expected value of the quantity.

Using the assumptions given in section D.1, this expression reduces to

(D.2) $$\bar{X}(n) = S^2_I + S^2_q + 2\sigma^2_o = \left| S \right|^2 + 2\sigma^2_o$$

Calculation of the variance of $X(n)$ is straightforward, but quite tedious, to perform. Therefore we will only provide the result of that computation:
The next step is to calculate the PDI output signal and its associated mean and variance. Assuming the PDI ratio is $N$, the output signal has the form

\begin{equation}
\text{(D.4)} \quad y(n) = \sum_{n=1}^{N} X(n).
\end{equation}

The mean of $y(n)$ is computed from

\begin{equation}
\overline{y(n)} = \sum_{n=1}^{N} \overline{X(n)} = \sum_{n=1}^{N} |S(n)|^2 + 2\sigma_o^2
\end{equation}

or, defining $|S(n)|^2 = \frac{1}{N} \sum_{n=1}^{N} |S(n)|^2$, we have

\begin{equation}
\overline{y(n)} = N|S(n)|^2 + 2\sigma_o^2.
\end{equation}

Calculation of the variance of $y(n)$ is based upon the following fact which is stated without proof. (The proof is straightforward, but quite messy.) Since it was assumed that $n_1(i)$, $n_1(j)$ (and $n_q(i)$, $n_q(j)$) are statistically independent for all $i$, $j$ such that $i \neq j$, it can be shown that $(x(i), x(j))$ are uncorrelated (and statistically independent). Using this fact, and the well-known relation,

\begin{equation}
\text{var} (x + y) = \text{var} x + \text{var} y
\end{equation}

where $x$ and $y$ are uncorrelated, one can easily write the expression for the variance of $y(n)$ as
We can define the new random variable

\[ Z = y - N|S|^2 \]

which has the mean and variance

\[ \mu_Z = y - N|S|^2 \]
\[ \text{var} Z = \text{var} y \]

Thus, from equation (D.7), it is seen that the output of the PDI can be expressed as the sum of the noise-free target response \((N|S|^2)\) and a sample from the random variable \(Z\) which has the mean and variance given in equation (D.8) and the pdf, \(P_Z\), which is derived in the next subsection.

2.2 Derivation of the PDF for \(Z\)

The pdf for the random variable \(Z\) can be derived as follows. Define the random variable

\[ w = \frac{1}{\sqrt{N}} \left[ Z - 2N_o \sigma^2 \right] \]

\[ = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \left[ x(n) - |S(n)|^2 - 2N_o \sigma^2 \right] \]

or

\[ w = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} w_n \]
where the $w_n$ are continuous, mean zero, and statistically independent. When $N$ is reasonably large (we note that $N \geq 10$ for passive tracking modes), the pdf for $w$ approaches a normal distribution of the form

$$P_w(w) = \frac{1}{\sqrt{2\pi \sigma_w^2}} e^{-\frac{(w)^2}{2\sigma_w^2}}$$

where $\sigma_w^2 = \frac{1}{N} \sum_{n=1}^{N} w_n^2$ from the central limit theorem [2]. Now, from equation (D.9), we have that

$$Z = \sqrt{N} w + 2N \sigma_0^2$$

and thus the pdf for $Z$ is normal with mean and variance

$$\bar{Z} = \sqrt{N} \bar{w} + 2N \sigma_0^2 = 2N \sigma_0^2$$

$$\text{var } Z = N \text{ var } w = \text{var } y$$

as shown in section D.2.1.

D.3 PRACTICAL ASPECTS OF MODEL IMPLEMENTATION

Given the value of $|S|^2$, the PDI output can be generated using the model of the previous section as follows:

$$y = N|S|^2 + 2N \sigma_0^2 + 2N \sigma_0^2 \left( \frac{|S|^2}{\sigma_0^2} + 1 \right) N(0,1)$$

where $N(0,1)$ denotes a sample from a normally distributed population with zero mean and unit variance. It is important to point out that, although the probability is very small, in some instances the resulting value of $y$ obtained from equation (D.12) can be negative. This result is totally unacceptable since it does not in practice and it will cause the log conversion process to become undefined.
Therefore, to prevent this situation, we make our final approximation: we simply set $y$ equal to the absolute value of the quantity on the right hand side of equation (D.12).

It is also noted that since the discriminant formation process computes the ratio of the noisy discriminant components, any scale factors that are common to both components can be ignored. We chose to ignore the factor, $2\sqrt{N} \sigma_0^2$.

Combining this fact with the absolute value approximation explained above, equation (D.12) becomes

\[
\text{(D.13)} \quad y = \left| \frac{\sqrt{|s|^2}}{2\sigma_0^2} + \sqrt{N} \left[ \frac{2|s|^2}{2\sigma_0^2} + 1 \right]^{\frac{1}{2}} \right| \mathcal{N}(0,1) .
\]
APPENDIX E

CROSS SECTION CALCULATION NOTES

Features 1-3

\[ \sigma = 291 \text{ al}^2 = 2.6 \text{m}^2 \]

Features 4-6

\[ \sigma = 291 \text{ al}^2 = 61 \text{m}^2 \]

Feature 7

\[ \sigma = 291 \text{ al}^2 = 25.7 \text{m}^2 \]

Features 8-10

\[ \sigma = \frac{4 \pi A}{\lambda^2} = 26934 \text{ A} \]

\[ A_e = 0.2 \text{m}^2 \]

\[ \sigma = 5000 \text{m}^2 - \text{Limit to } 1000 \text{m}^2 \]

Features 11-12

\[ A = (0.24 \text{m})^2 \times \pi = \]

\[ \sigma = 5000 \text{m}^2 - \text{Limit to } 1000 \text{m}^2 \]

Features 13-26

Take \( D = 0.5 \text{m} \)

\[ f/D = 0.5 \]

Mainlobe (13)

\[ \gamma = 6645D^2 = 3322 \text{m}^2 \]

Reflector (20)

\[ \sigma = 0.785D^2 \cos^4 \theta - 0.2 \cos^4 \theta \]

Take \( \cos \theta = 1. \)

259
Feature 14

\[ a = 0.24 \]
\[ L = 4m \]
\[ \sigma_1 = 291 aL^2 = 1117m^2 \]

Features 15-25

\[ \sigma = 4\pi A/\lambda^2 \] (See text)

Features 27-32

\[ a = 0.1m \]
\[ \sigma = \pi a^2 = 0.03 \text{ m}^2 \]

Features 33, 34

\[ a_c = 0.315m \]
\[ \theta_0 = 23.6^\circ \]
\[ \sigma(\theta < \theta_0) = 4\pi a_c^2 = 1.25m^2 \]
\[ \sigma(\theta > \theta_0) = \frac{5\pi a_c^2}{9} = 0.17m^2 \]
APPENDIX F

A MODEL FOR CENTROID WANDER IN
ROUGH SURFACE MODELS

The areas modeled as rough surfaces can be expected to experience wander of the apparent center. We take the mean position to be \( x_k, y_k, z_k \) in Table 4-1 and add a vector \( \bar{v} \) that reflects the wander. Consider features 35, perpendicular to the x-axis, 0.7 x 0.7 m in extent. Then we take \( \bar{v} \) in the yz plane with y and z components to be random variables with

\[
\bar{y} = \bar{z} = 0 \\
E yz = 0 \\
\sigma_y^2 = \sigma_z^2 = \frac{D^2}{12 N_f} \\
D = \text{Area dimension} = 0.7m \\
N_f = \text{No. of frequencies averaged} = 5
\]

The vector changes as the target aspect changes. We model this behavior as follows. Let \( v_m \) be the wander vector at the \( m \) th simulation update. Take \( \bar{v}_m \) to be a first order Markov process (Ref. 28 p. 324) with uncorrelated components and with, for example, the z component given by

\[
z_{m+1} = z_m + w_m
\]

where \( w_m \) is zero mean, uniform, of variance

\[
\sigma_w^2 = (1 - \alpha) \sigma_z^2
\]

We now choose \( \alpha \) to match the correlation time of the model to that of the target. One has

\[
\rho_k = \frac{E z_{m+1} \sigma_z^2}{E z_m^2} \sigma_z^2
\]
and taking the "correlation interval" as

\[ N_c = \frac{1}{\ln \alpha} \]

\[ \alpha = \exp(-1/N_c) \]

This is the number of iterations for which the correlation \( \rho_N \) falls to 1/e.

Now consider the target

The target return becomes decorrelated every time its aspect changes enough to cause another \( 2\pi \) propagation phase change from one edge of the area to the other. This corresponds to half wavelength differential range:

Change in \( \Delta r = \text{Change in } D \sin \phi_x = \frac{\lambda}{2} \)

Let \( \phi_x \) change by \( \Delta \phi_x \). Then

\[ \Delta(\Delta r) = \Delta \phi \frac{d}{d\phi} (\Delta R) \]

\[ = \Delta \phi \ D \cos \phi_x = \frac{\lambda}{2} \]

so that the angle change corresponding to decorrelation is

\[ \Delta \phi_x = \frac{\lambda}{2D \cos \phi_x} \]
Let the change in one simulation update period be \( \Delta \phi_x \); then the number of update cycles required for decorrelation is

\[
N_u = \frac{\Delta \phi_x}{\delta \phi_x} = \frac{\lambda}{2D \delta \phi_x \cos \phi_x}
\]

Matching \( N_u \) to \( N_c \) then yields

\[
\alpha = \exp \left[ \frac{-2D \delta \phi_x \cos \phi_x}{\lambda} \right].
\]
APPENDIX G

LISTING OF SIMULATION MODEL COMPUTER CODE
* EXECUTIVE PROGRAM: INTERFACE WITH PARENT SIMULATION *

** +

***

* SUBROUTINE EXEC

** +

* COMMON /CNTL/

** +

* IPWR, IMODE, ITP, IASM, IDUMC(5), DUMC(3)

** +

* COMMON /OUTPUT/ MSWF, MTF, MSF, DUM(7), IOUM2(4)

** +

* COMMON /ICNTL/ IOLDPW, IOLDMD, IOLDSM, ICHOLD, KMSCLK, KWMUP, IDUM1(3)

** +

* DATA DATINT/1.0/

** +

* KWMUP=1

** +

* IF(DATINT*NE.1.0) GO TO 1

** +

* CALL DATA

** +

* IOLDPW=IPWR

** +

* DATINT=0.0

** +

* IF(I*EQ.1) GO TO 30

** +

* IF(IPWR*GT*1) GO TO S

** +

* CALL SYSINT

** +

* RETURN

** +

* IF NOT EXCEEDED --- INITIALIZE ALL SYSTEM FLAGS AND CLOCKS.

** +

* KMSCLK=0

** +

* CALL SYSINT

** +

* RETURN

** +

* IF SYSTEM IN COMM(IMODE=3) --- INITIALIZE ALL SYSTEM FLAGS.

** +

* CALL SYSINT

** +

* RETURN

** +

* IF SYSTEM IN RADAR MODE --- CHECK FOR CHANGE IN MODE (I.E. ACTIVE-TO

** +

* PASSIVE OR PASSIVE-TO-ACTIVE).

** +

* IF(IMODE*EQ.IOLDMD) GO TO 10

** +

* CALL SYSINT

** +

* UPDATE STATUS OF IOLDMD.

** +

* 10 IOLDMD=IMODE

** +

* CALL SYSINT

** +

* RETURN

** +

* IF SYSTEM IN STANDBY --- DETERMINE WHETHER SYSTEM IN STANDBY

** +

* IF(IPWR*GT.2) GO TO 15

** +

* CALL SYSINT

** +

* RETURN

** +

* IF WARMUP PERIOD EXCEEDED --- DETERMINE WARMUP PERIOD EXCEEDED

** +

* IF(KMSCLK*GT.KWMUP) GO TO 20

** +

* CALL SYSINT

** +

* RETURN

** +

* IF EXCEEDED --- CONTINUE SYSTEM OPERATING MODE DETERMINATION.

** +

* ORIGINAL PAGE IS

* OF POOR QUALITY
* STEP 5: DETERMINE IF THERE HAS BEEN AN ANTENNA STEERING MODE CHANGE.

20 IF(IASM.EQ.IOLDSM) GO TO 25

C IF CHANGE HAS OCCURRED --- RESET ALL FLAGS AND GO TO NEW MODE.

25 IOLDSM=IASM

C *************************************************************************************************

C * STEP 5: DETERMINE WHETHER SYSTEM IS IN SEARCH AND ACQUISITION OR TRACK MODE. *

C *************************************************************************************************

C IF TRACK FLAG DOWN --- GO TO SEARCH MODE.

IF MTF.EQ.1 OR MTP.EQ.1) GO TO 30

CALL SEARCH

RETURN

30 CALL TRACK

RETURN

FUNCTION

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

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,ITXP,IASM,IDUMC(5),DUMD(3)
COMMON /OUTPUT/MSWF,MWF,MSF,SRNS,SRDOT,SRANG,SRANG,SRPRTE,SRRTE
2 COMMON /ICNTL/IOLDPW,IOLDM,IOLDSM,IISHOLD,KMSCLK,KWUP,KSNCLK,
3 COMMON /ATDAT/DUMI(4),ALRATE,ATRTE,ATRATE,DUM2(2),ATBT,ATREF,ATRREF

C *************************************************************************************************

C * STEP 1: INITIALIZE ALL INTERNAL FLAGS AND CONTROLS *

IOLDM=IMODE

IOLDSM=IASM

ISHOLD=0

MTF=0

M2=0

MSS=0

MTKINT=0

C *************************************************************************************************

C * STEP 2: INITIALIZE ALL INTERNAL CLOCKS *

KACCLK=0

KSNCLK=0

C *************************************************************************************************

C * STEP 3: INITIALIZE ALL DISPLAY FLAGS *

MSWF=0

MSF=0

MDF=0

MRDF=0

C *************************************************************************************************

C * STEP 4: INITIALIZE ALL DISPLAY METERS *

SRNG=0.0

SRDOT=0.0

SPRTE=0.0

END
**STEP 5:** INITIALIZE GIMBAL POINTING LOOP

- **STEP 5-1:** IF SYSTEM POWER OFF THEN ALIGN BORESIGHT WITH ZENITH.
  - \( P_{REF} = 0.0 \)
  - \( R_{REF} = 0.0 \)
  - \( AL = 0.0 \)
  - \( BT = 0.0 \)
  - \( SPANG = 0.0 \)
  - \( SRANG = 0.0 \)
  - \( IOLDPW = IPWR \)
  - RETURN

- **STEP 5-2:** IF SYSTEM IN STANDBY THEN HOLD GIMBALS AT POSITION WHEN STANDBY ENTERED AND ZERO DISPLAYS.
  - IF \( IOLDPW \neq IPWR \) GO TO 10
  - \( P_{REF} = \pi \times SPANG \)
  - \( R_{REF} = \pi \times SRANG \)
  - \( IOLDPW = IPWR \)
  - RETURN

- **STEP 5-3:** PREPARE GIMBAL LOOP FOR ENTRY INTO ANY OF SEARCH MODES.
  - \( P_{REF} = \pi \times SPANG \)
  - \( R_{REF} = \pi \times SRANG \)
  - \( IOLDPW = IPWR \)
  - RETURN

**END**

---

**THIS SUBROUTINE COMPUTES THE RESPONSE TO ALL DISPLAYS AND CONTROLS WHEN THE RADAR IS IN ANY OF THE SEARCH MODES.**
**STEP 1: 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**

* IF(ISHOLD.EQ.1.AND.ISRCHG.EQ.1) GO TO 12
  RREF=EDRA
  PREF=EDPA
  ISHOLD=ISRCHG

**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**

* CALL POINT

**STEP 2-3: DETERMINE WHETHER BORESIGHT IN ZONE I AND/OR ZONE 0 AND TAKE APPROPRIATE ACTION**

* CALL ZONECK

**STEP 2-4: CHECK FOR TARGET DETECTION — IF IN ZONE 0**

* CALL DETECT

**STEP 3: CHECK FOR TARGET DETECTION — IF IN ZONE 0**

* CALL DETECT

**STEP 4: PERFORM SCAN SEQUENCE**

* CALL SCAN
AUTO SEARCH AND ACQUISITION MODE

STEP 1: DETERMINE WHETHER SEQUENCING THRU POINT OR SCAN

IF(ISRCHEQ.EQ.1) GO TO 32

STEP 2: PERFORM GIMBAL POINTING SEQUENCE

STEP 2-1: UPDATE ROLL/PITCH REFERENCE ANGLES.

PREF=PREF+FLOAT(ELS)*SLWRTE(ISLR+1)*TS

RREF=RREF+FLOAT(IAZS)*SLWRTE(ISLR+1)*TS

STEP 2-2: UPDATE POSITION OF GIMBALS.

CALL POINT

STEP 2-3: 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 3: CHECK FOR TARGET DETECTION -- IF SLEW RATE <0.4 DEG/SEC.

CALL DETECT

RETURN

MANUAL SEARCH AND ACQUISITION MODE

STEP 1: UPDATE ANTENNA POSITION

STEP 1-1: UPDATE ROLL/PITCH REFERENCE ANGLES.

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 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/SEC.

CALL DETECT

RETURN

END
SUBROUTINE DETECT
COMMON /CNTL/IPWR*I140DE*ITXPolASM*IDUMC(5)9EDRNG,EDMC(2)
COMMON /ICNTL/IDUMC(2),TGTSEG,GPSEG,UGRNG,UGVEL
COMMON /SYSOL/OUMC(12)•TGTSIG•GPS•GAS
COMMON /TGTIAT/ITT
COMMON /DET1/TAT/SIGMA*CLANG

* STEP 1: COMPUTE TARGET PARAMETERS WRT RADAR *
CALL TRNSFM
CALL PVTRNSM

* 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.
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 *

* STEP 4: PASSIVE AUTO/MANUAL MODE DETECTION PROCESS *

* STEP 4-1: CHECK SHORT RANGE FIRST — CALL SINGLE-HIT DETECTION
MODEL.
10 CALL SINGLE

* STEP 4-2: CHECK FOR SUCCESS IN SINGLE-HIT DETECTION — IF NOT SUC
CESSFUL, THEN TRY LONG RANGE SEARCH.
IF(MTP.EQ.0) CALL CFAR
RETURN

* STEP 5: PASSIVE GPC MODES DETECTION PROCESS *

* STEP 5-1: CHECK DESIGNATED RANGE.
15 IF(EDRNG.GT.2552.) GO TO 20.
**STEP 5-2:** IF DESIGNATED RANGE < 0.42 NM -- USE SINGLE-HIT DETECTION MODEL.

CALL SINGLE
RETURN

**STEP 5-3:** IF DESIGNATED RANGE > 0.42 NM -- USE CFAR DETECTION MODEL.

CALL CFAR
RETURN

END

* THIS SUBROUTINE CONTAINS SINGLE-HIT DETECTION MODEL *

SUBROUTINE SINGLE

**STEP 1:** COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT

**STEP 1-1:** SET SAMPLE RATE TO OBTAIN CORRECT NOISE LEVEL IN SNR COMP.

**STEP 1-2:** COMPUTE NOMINAL SNR.

MSAM=1

SNR=SNR(SIGMA,CGRNGE )

**STEP 2:** IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNR

**STEP 2-1:** CHECK SCAN FLAG.

**STEP 2-2:** COMPUTE BEAMSHAPE LOSS -- BASED UPON C.G. POSITION

**STEP 2-3:** ADD BEAMSHAPE LOSS TO NOMINAL. I.E. COMPUTE ACTUAL SNR

**STEP 3:** DETERMINE PROBABILITY OF DETECTION. PD. BASED UPON SNR CURVE.

**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 NCRV=3
  10 NCRV=5
  15 NCRV=NCRV+MSF

STEP 3-2: CONVERT SNRV TO DB.
  IF(SNR.LT.1E-08) GO TO 20
  GO TO 35

STEP 3-3: SNR OUTSIDE (-30 DB, 0 DB) INTERVAL? — IF SO, SET
OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.

STEP 3-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR
INTERPOLATION.

STEP 3-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION.

*STEP 4: DETERMINE OUTCOME OF DETECTION ATTEMPT*

*STEP 5: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT*

STEP 5-1: IF NO DETECTION — SET TARGET PRESENT FLAG LOW.

STEP 5-2: IF DETECTION SUCCESSFUL — SET TARGET PRESENT FLAG
HIGH AND INITIALIZE ACQUISITION CLOCK.
**THIS SUBROUTINE CONTAINS THE CFAR DETECTION MODEL**

**SUBROUTINE CFAR**

COMMON /CNTL/IPIW,INODE,ITXP,IASM,IDUM(5),EDRNG,DUMC(2)
COMMON /OUTPUT/MSW,MTP,MSF,DUM(7),IDUM(4)
COMMON /ICNTL/DUM(8),KACILK,HTP,IDUM(4),MRNG,MEAN,MPRF
COMMON /TGDATA/NT,DUM3(800),RQ(3),DQ(3),CGANG,CGVEL
COMMON /EDTDA/SIGMA,CGANG

DIMENSION RI(6),NP(6),PV(6),TPR(3),TS(2),P(41)
DATA ... ,882...918...937...951...967...999...997...996/
PI=3.1414...5

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.

Determine range interval.

DO 5 I=1,NRI

IF(RI(I).GT.EDRNG) GO TO 10

CONTINUE

SET SAMPLE RATE

10 MSAM=2

Determine PRF

MPRF=1

IF(MRNG.GE.RI(6)) MPRF=2

GO TO 20

CONTROL SETTINGS FOR AUTO/MANUAL MODES.

SET RANGE INTERVAL.

15 MRNG=6

SET SAMPLE RATE.

16 MSAM=2

SET PRF.

MPRF=1

STEP 2: COMPUTE NOMINAL SNR AT VIDEO FILTER OUTPUT

SNR=SNR(SIGMA,CGANG)
* STEP 3: IF NOT SCANNING ADD BEAMSHAPE LOSS TO SNRV *

STEP 3-1: CHECK SCAN FLAG.
IF(MSF.EQ.1) GO TO 25
STEP 3-2: COMPUTE BEAMSHAPE LOSS --- BASED UPON C.G. POSITION OFF
SORESIGHT.
BETAX=SPAT(CGANG)/2
STEP 3-3: ADD BEAMSHAPE LOSS TO NOMINAL SNRV, I.E. COMPUTE ACTUAL
SNRV:
SNR=SNRBETA2

* STEP 4: COMPUTE NET PROCESSOR GAIN AND COMBINE WITH SNRV TO FORM
* SNR0

STEP 4-1: COMPUTE RANGE GATE LOSS (RGL) --- DIFFERS FOR GPC AND
AUTO/MANUAL MODES.
COMPUTE EQUIVALENT RANGE OF XMIT PULSEWIDTH.
25 CTDE=C*MRNG/2.
Determine operating mode
IF(IL.AGE.3) GO TO 30
COMPUTE RGL FOR GPC MODES.
DEL=ABS(CRANGE-CTDE)/CTD2
IF(DEL.LE.1.0) RGL=0.0
IF(DEL.GE.0.5.AND. DEL.LT.1.5) RGL=0.6666666*1.5-DEL)**2
IF(DEL.LT.0.5) RGL=0.6666666
GO TO 35
COMPUTE RGL FOR AUTO/MANUAL MODES
30 DEL=DEL-INT(DEL)
IF(DEL.LE.1.0) RGL=DEL**DEL
IF(DEL.GE.1.0.AND. DEL.LT.4.5.AND. DEL.LT.1.5) RGL=DEL*DEL
IF(DEL.GE.1.0.AND. DEL.LT.4.5.AND. DEL.LT.1.5) RGL=0.6666666
COMPUTE NET PROCESSOR GAIN --- SAME FOR ALL PASSIVE ANTENNA
STEERING MODES.

STEP 4-2: COMPUTE DOPPLER FREQUENCY ASSOCIATED WITH TARGET RADIAL VELOCITY
35 FDOP=-2.0*CGVEL/ALMDA/1.0E-06
COMPUTE ARGUMENT ASSOCIATED WITH TARGET VELOCITY
ARG1=FLOAT(MFIL-INT(MFIL))
COMPUTE NET PRESUM GAIN
PSG=SUM(ARG1*NP(MANG))

STEP 4-3: COMPUTE NET DOPPLER FILTER GAIN --- SAME FOR ALL PASSIVE
ANTENNA STEERING MODES.

STEP 4-4: COMPUTE NET DOPPLER FILTER GAIN

STEP 4-5: COMPUTE NET PROCESSOR GAIN.
**STEP 4-5: COMPUTE SNR AT DOPPLER FILTER OUTPUT**

\[ \text{SNR} = \text{SNR} \times \text{NPG} \]

**STEP 5-1: DETERMINE INDEX TO ACCESS APPROPRIATE CURVE**

\[
\text{IF} (\text{SNR} \geq 3) \text{ GO TO 40}
\]

\[ \text{NCRV} + 1 \]  
\[ \text{GO TO 45} \]

\[ \text{NCRV} = 3 \]

**ADJUST INDEX FOR SCANNING**

\[ \text{NCRV} = \text{NCRV} + \text{MSF} \]

**STEP 5-2: CONVERT SNR TO DB**

\[
\text{IF} (\text{SNR} \leq 1.0E-08) \text{ GO TO 50}
\]

\[ \text{SNR} = 10 \times \text{ALOG10} (\text{SNR}) \]  
\[ \text{GO TO 45} \]

\[ \text{SNR} = -100. \]

**STEP 5-3: SNR OUTSIDE (0 DB, +20 DB) INTERVAL? --- IF SO, SET OUTCOME APPROPRIATELY AND SKIP REMAINING STEPS.**

\[
\text{IF} (\text{SNRD} < 0, \text{DB} \rightarrow \text{DECLARE A MISS.})
\]

\[ \text{IF} (\text{SNR} \leq 0.) \text{ GO TO 60} \]

\[ \text{IF} (\text{SNR} > 20. \text{DB} \rightarrow \text{DECLARE A HIT.})
\]

\[ \text{IF} (\text{SNR} \leq 50. \text{GO TO 65} \]

**STEP 5-4: COMPUTE INDEX FOR LOOKUP TABLE AND FACTORS FOR LINEAR INTERPOLATION**

\[
\text{SCALE} = (\text{SNR} + 0.) \times 2. \times 1.00000001
\]

\[
\text{ISNR} = \lfloor \text{SCALE} \rfloor
\]

\[
\text{NEW} = \text{SCALE} - \text{FLOAT} (\text{ISNR})
\]

**STEP 5-5: DETERMINE PD USING TABLE AND LINEAR (IN DB) INTERPOLATION**

\[
\text{PROB} = \text{P} (\text{ISNR}) + \text{REMAIN} \times (\text{P} (\text{ISNR} + 1) - \text{P} (\text{ISNR}))
\]

**STEP 6: DETERMINE OUTCOME OF DETECTION ATTEMPT**

\[
\text{X} = \text{RNDU} (\text{NSRCM})
\]

\[
\text{IF} (\text{X} \leq \text{PROR}) \text{ GO TO 65}
\]

**STEP 7: SET CONTROLS BASED UPON OUTCOME OF DETECTION ATTEMPT**

**STEP 7-1: IF NO DETECTION --- SET TARGET PRESENT FLAG 'LOW.'**

\[ \text{MTP} = 0 \]  
\[ \text{RETURN} \]

**STEP 7-2: IF DETECTION SUCCESSFUL --- SET TARGET PRESENT FLAG 'HIGH AND INITIALIZE ACQUISITION CLOCK.'**

\[ \text{MTP} = 1 \]  
\[ \text{KACCLK} = 0 \]  
\[ \text{RETURN} \]

**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

FUNCTION SNRV(SIGMA•RANGE)  
COMMON /CNTL/IPMR•I MODE, ITXP•OM, DUMC(6), DUMC(3)  
COMMON /ICNTL/IDUM(12),MS•MTKINT•MRNG•MSAN•MPRF•IDUM2(10)  
COMMON /SYSDAT/DUM(12),TGTSIG•GPS•GAS  
DIMENSION PT(3),BN(2)  
DATA PT/47.923,7.7/  
B/W 69.5  
DATA AK/2.0/  
TAU/1.414/  
PI/3.141592653/  
\* DETERMINE WHETHER ACTIVE OR PASSIVE MODE *  
** IF(MODE.EQ.1) GO TO 10  
** PASSIVE MODE VIDEO SNR CALCULATION *  
SNRV=GAS-20.*ALOG10(RANGE)  
SNRV=10.**(0.1*SNRV)  
RETURN

** ACTIVE MODE VIDEO SNR CALCULATION *  
10 SNRV=GAS-20.*ALOG10(RANGE)  
SNRV=10.**(0.1*SNRV)  
RETURN  
END

** THIS SUBROUTINE UPDATES THE POSITION OF THE ANTENNA GIMBALS *  
SUBROUTINE POINT  
COMMON /OUTPUT/IDUM1(3),DUM(2),SPANG,SRANG•DUM5(3),IDUM2(4)  
COMMON /SYSDAT/T•DUM(3),CG•SG•DUM2(9)  
COMMON /ATDAT/DUM(4),SALRTE•SBRTE•DUM3(2),AL•BT•PREF•RREF/  
DATA AK/2.0/  
TAU/1.414/  
PI/3.141592653/  
\* STEP 1: PRELIMINARY COMPUTATIONS *  
CR=COS(-RREF)  
SR=SIGN(-RREF)  
CP=COS(-PREF)  
SP=SIGN(-PREF)
STEP 2: COMPUTE ANTENNA REFERENCE ROLL/PITCH ANGLES IN THE RADAR FRAME.

```
XX = CG * SP - SG * SR * CP
YY = SG * SP + CG * SR * CP
ZZ = CR * CP
```

IF (YY.LT.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 RREF = ASIN(XX)
```

STEP 3: UPDATE OUTER (ALPHA) GIMBAL RATE AND POSITION

```
COMPUTE ALPHA LOOP POSITION ERROR.
ERRA = AREF - AL
```

STEP 4: UPDATE INNER (BETA) GIMBAL RATE AND POSITION

```
COMPUTE BETA LOOP POSITION ERROR.
ERRB = SREF - BT
```

STEP 5: ANTENNA IN OBSCURATION REGION?

```
CALL SCNWTRN
```

STEP 6: COMPUTE ANTENNA ROLL/PITCH ANGLES IN THE BODY FRAME

```
CA = COS(AL)
SA = SIN(AL)
CR = 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
AREF = ATAN2(YY, ZZ)
GO TO 4
3 IF (XX.GT.0.0) SRANG = 90.0
IF (XX.LT.0.0) SRANG = -90.0
4 SPANG = SP - 20576 * ASIN(XX)
RESOLVE POSSIBLE ANGLE AMBIGUITIES, VIZ., -90.<SPANG<90. AND -180.<SRANG<180.
IF (SRANG.LE.90.) GO TO 10
```

END
**THIS SUBROUTINE DETERMINES WHETHER THE ANTENNA IS IN THE 08 — 00011620
* SCURATION ZONE AND SETS THE SCAN WARNING FLAG APPROPRIATELY. **

SUBROUTINE SCNWRM 00011680
COMMON /OUTPUT/MSWF,DUMO(2),DUMO(7),DUMO1(4) 00011690
COMMON /ATDAT/DUM(8),A8,DUM(4) 00011700
DIMENSION ICLEAR(36) 00011710
DATA ICLEAR /17*1913*096*1918*1912*096*191861 *12*096*1919*1911*096*1919*1911*096*19 00011720
2 19*1.11*09611919*1911*09601919*1911*096*192001910*096*19 00011730
3 20*1 *10*0*6*1920*1 *10*096*19-20*1910*096*1920*1910*096*19 00011740
4 6*1920*1910*096*1919*1 *11*096*1918*1912*0 *6*1*17*1.13*09 00011750
5 6*1916*1914*096*1 91501 91500 96*1914*l 9-16*0*6*19-14*1,16*09- 00011760
6 6*1913*1917*096*1912*1 *18*0*6*1*11*1919*O96*1910*1920*09019 00011770
7 9*1921*096*199*1921*0 *6*198*I *22*O *6*1 *4*1909-3*19-22*096*I9 00011780
8 4*1926*096*194*1926*096*1 *2690 *6*1 *29*1 *0*6*1929619096*1928#1.26*096*1 9-000 11 800
4*1926*0*6*194*1926#09-6*194*19-26*09-6*1900011810
A 4*1 *26*096*194*1 *26*096*194*1926*096*194*197*092*1917*096619- 00011840
B 4*197*096*1924*1 9600 9 601 9-26*1*460* 00011860
E 21*199*096*1924*19600 9 601 9-26*1*460* 00011880
F 6*1927*193*096*1928*l92*096*1 *29*1 *0*6*1929619096*1928#19 00011790
G 2*O *6*1927*0693*096*1 *26*194*0 *6*1925*195*096*1923 422 00011890
H 23*197*096*1922*198*096*1919*1-11*0 *6*1*18*1912*09-6*1/ 00011900

ALPHA=A 00011910
BETA=B 00011920
IF(ABS(BETA)=.LE.90.) GO TO 1 00011930
BETA=-(180.-ABS(B)) * (B/ABS(B)) 00011940
1 CONTINUE 00011950
IA=INT((ALPHA+180.)/5.+.1) 00011960
IB=INT((90.-BETA)/5.+1.) 00011970
MSWF=ICLEAR(IB,IA) 00011980
RETURN 00012000
END 00012010

**THIS SUBROUTINE DETERMINES WHETHER ANTENNA IS IN ZONE 1 AND/OR ZONE 0 (FOR GPC-AC0 AND GPC-DES POINTING MODES ONLY). **

SUBROUTINE ZONECK 00012100
COMMON /CNTL/IDUMC(9),EDRNG,EDPA,EDRA 00012110
COMMON /OUTPUT/IDUM1(3),DUM1(2),SPANG,SRANG,DUM3(3),IDUM3(4) 00012120
COMMON /ICNTRL/IDUM2(10),MZ1,MZ0,EXUM4(15) 00012130
MZO=0 00012140
MZ1=1 00012150
PI=3.141592653/180. 00012160
RB=PI*SRANG 00012170
PB=PI*SPANG 00012180
P=EDPA 00012190
R=EDRA 00012200
CPB=COS(PB) 00012210
SPB=SIN(PB) 00012220
CRB=COS(RB) 00012230
SRB=SIN(RB) 00012240
CP=COS(PI) 00012250
SP=SIN(PI) 00012260
CR=COS(R) 00012270
SR=SIN(R) 00012280

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

***************************************************************************
***************************************************************************
***************************************************************************
***************************************************************************
***************************************************************************
**THIS SUBROUTINE CYCLES THRU THE LOGIC FOR ANY SCAN GENERATION.**

```
ANGDF=ACOS((SPB+CRB+SP+CR+RB+SR+CPB+CR+CP)/PII)
ANGDF=ABS (ANGDFF)
IF(ANGDF.GT.3.0) RETURN
ANGDF=ABS (ANGDF)
MZD=1
IF(ANGDF.GT.0.3) RETURN
MZD=1
RETURN
END
```

SUBROUTINE SCAN
COMMON /CNTL/IDUM(4),ISRCHC,ISRCHG.IDUMC(3),EDRNG.DUMC(2)
COMMON /OUTPUT/MSWF,MTF,MSF.IDUM(7)
COMMON /ICNTL/IDUM3(4),KSNCLK.IDUM4(2),MTP.IDUM5(17),MSWTCH
2
KSN,JAROLD,ITROLD
COMMON /SYSDAT/TSAM.DUMS(14)
COMMON /TDAT/NT,DUM2(SO3).ROU(3),DUM3(2)
COMMON /DATAT/DUM4(3).AI,BT.DUM5(2),ARF.BREF
DIMENSION TIMINT(31),ANGINT(31),RSW(10),TSW(10)
DATA TIMINT=7,1.4,1.9,2.6,3.4,4.3,5.1,6.7,8.9,10.4,11.6,
13.3,14.9,16.9,18.9,21.4,23.4,25.9,28.6,31.5,33.5,35.6,37.9,40.8,
2
43.2,46.8,50.5,54.3,58.4,60.0
DATA ANGINT=7,1.5,2.2,3.3,4.4,5.6,6.4,7.6,8.8,9.9,10.9,9.8,10.9,
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
25.7,27.8,29.9,30.1
DATA TSW=60.0,54.3,43.2,33.2,23.2,13.2,1.2,1.1,1.2
DATA RSW=46.809.2,55900.6,62584.3,71698.6,91142.5,151903.8
3
MZI=0.9,39490.8,88104.8,1622845.0,
PII=180./3.141592653
```

**STEP 1: DETERMINE WHETHER TO PERFORM SCAN Initialization(MSF=0) OR SCAN UPDATE(MSF=1).**

```
IF(MSF.EQ.0) GO TO 15
```

**STEP 2: PERFORM SCAN INITIALIZATION.**

```
INITIALIZE ALL FLAGS.*
MSF=1
C INITIALIZE RING MONITORS.
JAROLD=0
ITROLD=10
C INITIALIZE SCAN CLOCK.
KSNCLK=0
C INITIALIZE SCAN TIME PARAMETER.
KSN=0
C DETERMINE SWITCH POINT PARAMETER.
DO 5 I=1,10
IF(EDRNG.LT.RSW(I)) GO TO 10
5 CONTINUE
10 MSWTCH=I
C
```
* STEP 3: UPDATE SCAN CLOCKS *

STEP 3-1: UPDATE SCAN CLOCK (TRACKS TOTAL ELAPSED TIME FROM SCAN INITIATION).
15 KSNCLK=KSNCLK+1
T=FLOAT(KSNCLK)*TSAM

STEP 3-2: UPDATE SCAN TIME PARAMETER (USED TO DETERMINE BORESIGHT POSITION IN SCAN PATTERN).
IF(T .LE. TSW(MSWSCH)) KSN=KSN+1
IF(T .GT. TSW(MSWSCH)) KSN=KSN-1
TSN=FLOAT(KSN)*TSAM

* STEP 4: DETERMINE ANTENNA POSITION TO NEAREST SCAN RING *
DO 20 T=1,3
IF(TSN.LT.TIMINT(1)) GO TO 25
20 CONTINUE
25 IARNG=I

* STEP 5: DETERMINE TARGET POSITION IN SCAN PATTERN (SCAN RING NUMBER FOR TARGET) *

STEP 5-1: DETERMINE TARGET POSITION EXACTLY.
AL0LD=AL
BTOLD=BT
AL=AREF
BT=BREF
CALL TRANSFM
CALL PVTTRAN
AL=AL0LD
BT=BTOLD

STEP 5-2: DETERMINE TARGET SCAN RING NUMBER.
Determine target angle off scan designates (degrees).
CGANG=ACOS(ROU(3))

STEP 6: DETERMINE IF A DETECTION SHOULD BE ATTEMPTED *

STEP 6-1: CHECK CONDITION.
IF(IARNG.EQ.ITRNG.AND.IAROLD.NE.ITROLD) CALL DETECT

STEP 6-2: UPDATE RING NUMBER MONITOR.
IAROLD=IARNG
ITROLD=ITRNG

* STEP 7: CHECK FOR SCAN TERMINATION CONDITIONS *

STEP 7-1: CHECK ALL POSSIBLE TERMINATION CONDITIONS.
CONDITION # 1: T > 60 SECONDS?
IF(T.GE.60.) GO TO 40
CONDITION # 2: NEXT SCAN TIME PARAMETER < 0.?
ITEMP=KSN-1
IF (ITEMP.LT.0) GO TO 40

CONDITION # 3: DETECT A TARGET?
IF (MTF.EQ.0) RETURN

STEP 7-2: PERFORM SCAN TERMINATION STEPS --- IF TERMINATION CONDITION OBTAINED.

40 MSF=0
KSNCLK=0
KSN=0
ISRCHG=0
ISRCHC=0
RETURN
END

******************************************************************************
THIS SUBROUTINE SIMULATES THE TRACKING MODES OF THE KU-BAND
******************************************************************************

SUBROUTINE TRACK
COMMON /CNTL/IDUM(3),IASH,ISRCHG,ISRCHC,IAZS,IELS,ISLR,EORNG,
EDPA,EDRA,2
COMMON /OUTPUT/MSWP,MTF,MSF,DUMQ(7),IDUMQ(4)
COMMON /ICNTL/IIDUM(13),MTKINT,MRNG,MSAM,MPRF,MBKTRK,IDUM(9)
COMMON /SYSDAT/TSAM,DUM2(14)
COMMON /ATOA T/DUMO(14),PREF,RREF,DUMA(2)
DIMENSION SLMRTE(2)
DATA SLMRTE/6EQ8I4E-3,3.4907E-1/

******************************************************************************
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.EQ.1) GO TO 6
IF (MTKINT.NE.0) GO TO 5

STEP 1-1: INITIALIZE RANGE ANGLE AND VELOCITY TRACK LOOPS --- ASSUMES STEADY STATE TRACKING OF TARGET C.G.
CALL TKINI

STEP 2-1: UPDATE DATA VALID FLAG STATUS --- ONLY WHEN ENTERING TRACK FROM SEARCH.
5 CALL TGTACQ

******************************************************************************
STEP 2-1: UPDATE TRANSFORMATION MATRICES AND MATRICE RATES.
6 CALL TRNSFM

STEP 2-2: TRANSFORM TARGET POSITION AND VELOCITY COMPONENTS FROM ORBITER BODY FRAME-TO-ANTENNA LOS FRAME.
CALL PRTRAN

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: DETERMINE IF A BREAK TRACK CONDITION HAS OCCURRED.
CALL BRKTRK

CHECK STATUS OF BREAK-TRACK FLAG (MBKTRK=1 --- BREAK-TRACK).
IF(MBKTRK.EQ.1) GO TO 7

IF BREAK-TRACK HAS OCCURRED --- RESET THE SYSTEM AND RETURN TO
SEARCH.
CALL SYSINT
RETURN

STEP 2-6: UPDATE ANTENNA GIMBAL POSITIONS AND RATES AND TARGET
ANGLES AND ANGLE RATES FOR DISPLAY (GPC-AOQ AND AUTO
MODES ONLY).
7 IF(IASM.EQ.2 OR IASM.EQ.4) GO TO 10

FOR GPC-AOQ OR AUTO USE RADAR ESTIMATED TARGET ANGLES FOR
TRACK SERVO INPUT.
8 IF(SMBKTRK.EQ.1) GO TO 15

FOR GPC-DES MODE USE GPC-SUPPLIED ANGLE DESIGNATES FOR TRACK SERVO
INPUT.
PREF=EDPA
RREF=EDRA
CALL POINT
GO TO 15
10 IF(IASM.EQ.4) GO TO 12

FOR MANUAL MODE USE CREW-SUPPLIED SLEW RATES TO DETERMINE TRACK
SERVO INPUT.
12 PREF=PREF+FLOAT(IELS)*SLWRTE(ISLR+1)*TSAM
RREF=RREF+FLOAT(IAS)*SLWRTE(ISLR+1)*TSAM
CALL POINT

STEP 2-7: UPDATE THE RANGE AND RANGE RATE ESTIMATES.
15 CALL RTRACK

STEP 2-8: DETERMINE RADAR SIGNAL STRENGTH (FOR DISPLAY METER)
CALL RSS
20 RETURN

END

*************************************************************************
* THIS SUBROUTINE INITIALIZES THE ANGLE TRACKING LOOPS, THE * 
* RANGE TRACKING LOOP, AND THE VELOCITY PROCESSOR --- STEADY * 
* STATE CONDITIONS ARE ASSUMED. *
*************************************************************************

SUBROUTINE TKINIT
COMMON /CNTRL/IMODE,ITXP,IASM,IDUMC(5),DUMC(3)
COMMON /INPUT/ERT(3),EVT(3),EWB(3),DUM(18)
COMMON /OUTPUT/IDUM(3),SRNG,DUM(6),IDUM(4)
COMMON /CNTRL/IIDUM(13),MTKINT,SRNG,MSAN,MPRF,MBKTRK,MBSUM.*
2 COMMON /MT(8) 
COMMON /SYSDAT/TSAN,OR(3),CP,SP,PSI,PSBIAS,DUM2(7)
COMMON /TOAT/NUT,DUM(500),RO(3),RUI(3),CRANGE,EVEL
COMMON /SADAT/RADAR(3),KTR,RT(70),S1G,ROLD,ICLOSE,ICOLD
COMMON /ATDAT/CA,SA,CB,SB,AZRATE,ELRATE,BRATE,AL,BT
2 COMMON /DUM(2)
COMMON /RTDAT/ORGN,RGBIAS,VEST(4),MDF(5)
COMMON /XFORMS/TLB(3,3),TLB(3,3),TLT(3,3),TLT(3,3)
COMMON /AGCDAT/AGC,AGCOLD
DIMENSION TRB(3,3),ER(3),ERT(3),FLTV(3),RI(10)
DATA FLTV/7.7215,3.390062969/
DATA RI/120.,240.,780.,7552.,5772.,11544.,23089.,43747.
2 5772.,1.8228E+6,JRI/10.,PI/3.141592653/

282
**STEP 0: INITIALIZE BREAK-TRACK ALGORITHM**

**STEP 0-1: INITIALIZE MOVING WINDOW-OF-8 REGISTERS.**

DO 3 I=1,8
3 WBT(I)=0

**STEP 0-2: INITIALIZE SUM REGISTER.**

WTSUM=0

**STEP 0-3: SET BREAK-TRACK FLAG TO LOW (OR 0) STATE.**

NBTRK=0

**STEP 1: INITIALIZE ANGLE TRACKING LOOP**

**STEP 1-1: COMPUTE INITIAL INNER AND OUTER GIMBAL POSITIONS.**

IF(IASM.EQ.0 OR IASM.EQ.4) GO TO 5

**STEP 1-2: COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH AND ELEVATION RATES.**

PRELIMINARY TRIGONOMETRIC COMPUTATIONS.

* CG=cos(AL)
  * SG=sin(AL)
  * CB=cos(BT)
  * SB=sin(BT)
  * TRANSFORM BODY ANGULAR VELOCITY VECTOR FROM BODY TO OUTER GIMBAL(G) REFERENCE FRAME.
    * WG1=CP*ER2+CSP*EW1(2)
    * WG2=-SC*EW1(1)+CP*EW1(2)+CSP*EW1(3)
    * COMPUTE THE RANGE TO TARGET.
      * R=SQRT(ER(1)**2+ER(2)**2+ER(3)**2)
    * COMPUTE INITIAL TARGET INERTIAL LOS AZIMUTH RATE(AZRATE).
      * VGY=CA*EV(2)+SA*EV(3)
    * COMPUTE INITIAL TARGET INERTIAL LOS ELEVATION RATE(ELRATES).
      * ELRATE=(CB*EV(1)-SB*SA*EV(2)+CA*EV(3))/R+VGY
STEP 1-31: COMPUTE INITIAL INNER AND OUTER GIMBAL RATES.

* COMPUTE INITIAL OUTER GIMBAL RATE(ALRATE).*

RCB=RCB

IF(ABS(RCB).LT.1.0E-6) GO TO 2

ALRATE=0

CONTINUE

STEP 1-31: COMPUTE INITIAL INNER GIMBAL RATE(STRATE).

BTRATE=ELRATE-WGY

STEP 2-1: TRANSFORM TARGET C.G. POSITION AND C.G. VELOCITY FROM BODY TO ANTENNA LOS FRAME.

CALL TNSFM

CALL PVTRAN

STEP 2-2: INITIALIZE THE RANGE ESTIMATE REGISTER.

SRNG=CRNGE

RNG=INTT(SRNG+3.2)

STEP 2-3: INITIALIZE THE RANGE RATE ESTIMATE REGISTER.

RNGT=INTT(CGVEL+TSAM+3.2)

STEP 3: DETERMINE CORRECT RANGE INTERVAL.

DO 30 I=1,NRNG

SRNG=I

CONTINUE

STEP 3-2: DETERMINE CORRECT SAMPLE RATE.

IF(MODE.GE.2) GO TO 44

IF(MRNG.GT.9) GO TO 42

MSAM=1

CONTINUE

STEP 3-3: DETERMINE CORRECT PRF.

IF(MODE.GE.2) GO TO 54

IF(MRNG.GT.9) GO TO 50

MSAM=1

CONTINUE

STEP 4: INITIALIZE VELOCITY PROCESSOR.
STEP 4-1: INITIALIZE MOVING WINDOW VELOCITY AVERAGING.

DO 10 I=1,4

VEST(I)=CGVEL(I)*

10

STEP 4-2: SET INITIAL POSITION OF 5 DOPPLER FILTERS.

VR=CGVEL/FLTWID(MPRF)

IVR=INTT(VR+0.5)+32000

MDF(3)=MOD(IVR,32)

DO 20 J=1,5

MD=MDF(J)+1+3*32000

MDF(1)=MOD(MD,32)

20

STEP 5: INITIALIZE SIGNAL STRENGTH ALGORITHM PARAMETERS

AGCOLD=0.0

ITXP=1

STEP 6: SET TRACK INDICATOR TO ALLOW OPERATION OF TRACK LOOP

MTKINT=1

NOTE: DEBUGGING PRINT STATEMENTS.

WRITE(6,899)

WRITE(6,900) AZRATE, ELRATE, ALRTE, STRATE, AL, BT

WRITE(6,901) IRNG, IRODT, SRNG

WRITE(6,902) (VEST(I), I=1,4), MDF(J), J=1,5

WRITE(6,903) IRNG, IRODT, SRNG

WRITE(6,904) IMODE, IROD, MSAM, MRRF

899 FORMAT(/' TRACKER INITIALIZATION: */ TRACK: AZRAT

2 ,' ELRTE, ALRTE, STRAE, AL, BT')

900 FORMAT(6F14.6)

901 FORMAT(' ITRACK: IRNG, IRODT, SRNG')

902 FORMAT(218F14.6)

903 FORMAT(' VTRACK: VEST, MDF')

904 FORMAT(6F14.6,8I8)

905 FORMAT(' CNTL: IMDRE, IROD, MSAM, MRRF')

906 FORMAT(818F14.6)

RETURN

END

SUBROUTINE TGTACO

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

COMMON /OUTPL/: /MSF, MTIP, MSF, IDUM(7), MDF, MRRDF, MRRDF, MRRDF

COMMON /ICNTL/: IDUM(3), IMODLL, MTP, MZ1, M20, MSAM, MTKINT

COMMON /SYSDAT/: TS, DUM(14)

DIMENS IDM ADV(10,2), RDV(10,2), ARDV(10,2)

DATA ADV/98.10,2,11,2,81,02,242.33/

DATA RDV/98,15,26,69,86,97,229,76/

DATA ARDV/98.2,26.69,78.2,26.23,2*29.76/

285
STEP 1: UPDATE ACQUISITION CLOCK
KACCLK = KACCLK + 1
ACCLK = KACCLK * TS

STEP 2: PERFORM ANGLE DATA VALID TEST — GPC-ACQ & AUTO ONLY
IF (IASM.EQ.2 OR IASM.EQ.4) GO TO 10
MADVF = 1

STEP 3: PERFORM RANGE AND RANGE RATE DATA VALID TEST
IF (ACCLK.LT.ADV(MRNG.IMODE)) GO TO 15
MRDVF = 1
MADVF = 1

IF GPC-DES OR MANUAL INITIATE RADAR TRACKING PARAMETERS.
15 IF (IASM.EQ.2 OR IASM.EQ.4 AND MRDVF.EQ.1) GO TO 20

STEP 4: PERFORM ANGLE RATE DATA VALID TEST — GPC-ACQ & AUTO
MADVF = 1

STEP 5: PERFORM STEADY STATE RADAR TRACKING INITIALIZATION
KACCLK = 0
MIF = 1
RETURN
END

THIS SUBROUTINE UPDATES ALL REQUIRED TRANSFORMATION
MATRICES AND TRANSFORMATION MATRIX RATES.

SUBROUTINE TRANSFM
COMMON /INPUT/DUM(9), TBT(3,3), TTD(3,3)
COMMON /SYSDAT/DUM2(4), CP, SP, DUM(9)
COMMON /ATDAT/CA, SA, CB, SB, DUM(2), ALRATE, BTRATE, AL, BT, DUM3(4)
COMMON /XFORMS/TLB(3,3), TLD(3,3), TLT(3,3), TLD(3,3)

STEP 1: UPDATE TRANSFORMATION MATRICES
STEP 1-1: PRELIMINARY COMPUTATIONS.
CA = COS(BT)
SB = SIN(BT)
SA = SIN(AL)

STEP 1-2: COMPUTE TRANSFORMATION MATRIX TLB (BODY-TO-LOS FRAME).
TLB(1,1) = CB*CP - SB*SA*SP
TLB(1,2) = CB*SP - SB*SA*CP
TLB(1,3) = SB*CA
TLB(2,1) = -CA*SP
TLB(2,2) = CA*CP
TLB(2,3) = SA
TLB(3,1) = SB*CP - CB*SA*SP
TLB(3,2) = SB*SP - CB*SA*CP
TLB(3,3) = CB*CA
STEP 1-3: COMPUTE TRANSFORMATION MATRIX TLT (TARGET-TO-LOS FRAME).

```
DO 10 I=1,3
    TL(I,J)=0.0
DO 10 K=1,3
10 TLT(I,J)=TLT(I,J)+TLD(I,K)*TBT(K,J)
```

STEP 2-1: COMPUTE TLD-DOT.

```
TLD(I,J)=-STRATE(I,J)*TLD(I,J)+ALRATE(I,J)*TBT(I,J)
```

STEP 2-2: COMPUTE TLT-DOT.

```
TLD(I,J)=-STRATE(I,J)*TLD(I,J)+ALRATE(I,J)*TBT(I,J)
```

RETURN

END

SUBROUTINE PVTRAN

```plaintext
COMMON /CNTL/IPWR(3),IMODE
COMMON /INPUT/ERT(3),EVT(3),DUM(21)
COMMON /MULT/TLB,MSL,TBT(3,3),TBD(3,3),TLT(3,3),TLD(3,3)
COMMON /ICNTL/IJUM6(91,MTP,MDUM7(3,MTKINT)
COMMON /SYSOAT/TSAM,DR(3),SO12(11)
COMMON /TGTOAT/NT,RAU(3),RANGE(100),RADVEL(100),RO(3)
COMMON /SATOR/A,RADAR(3),N20,RT(70),SIG(70),RLOD,ICLOSE,ICOLD
COMMON /XFORMS/TLB(3,3),TLB(3,3),TLT(3,3),TLD(3,3),RO(3),CGRNGE
DIMENS N=70,R=30 V=10,RV=100,RS=30
```

STEP 1: ADD RADAR OFFSET IN ORBITER BODY FRAME.

```
DO 5 I=1,3
    ROR(I)=ERT(I)-OR(I)
5 RETURN
```

STEP 1-2: TRANSFORM TARGET C.G. POSITION FROM BODY FRAME TO ANTE\r\n```
CALL MULT31(TLB,ROR,RO)
```

STEP 1-3: COMPUTE RANGE OF TARGET C.G. WRT RADIUS.

```
CGRANGE=SQRT((RO(1)**2+RO(2)**2+RO(3)**2)**2+R(1)**2+R(2)**2+R(3)**2)
```

STEP 1-4: COMPUTE UNIT VECTOR IN DIRECTION OF TARGET C.G. WRT ANTE\r\n```
DO 10 I=1,3
    ROU(I)=RO(I)/CGRANGE
10 RETURN
```

END
****STEP 2: COMPUTE TARGET C.G. RADIAL VELOCITY WRT ANTENNA LOS****

**STEP 2-1: COMPUTE TARGET C.G. VELOCITY COMPONENTS WRT ANTENNA LOS FRAME**
- CALL MULT31(TLB,R,ROU,V1)
- CALL MULT31(TLB,EVT,ROD)

**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 TARGET N**

**STEP 3-1: IF IN ACTIVE MODE, SEARCH MODE, OR TRACKER INITIALIZATION -- ASSUME SINGLE SCATTERER LOCATED AT TARGET FRAME ORIGIN.**
- MODING.EQ.1.AND.MTKINT.EQ.0.AND.MTP.EQ.0) GO TO 30
- NT=1
- SIG(I)=1.0
- DO 25 I=1,3
  - 25 R(I)=0.0

**STEP 3-2: COMPUTE LOCATION OF RADAR IN TARGET FRAME.**
- DO 35 J=1,3
  - 35 RADAR(I)=RADAR(I)+TLT(I,J)*ROU(J)

**STEP 3-3: COMPUTE TARGET SCATTERING CHARACTERISTICS.**
- CALL SPAS
  - NT=20

**STEP 4: COMPUTE KTH SCATTERER POSITION, RANGE, AND DIRECTION VECTOR WRT ANTENNA LOS FRAME (OR RADAR).**

**STEP 4-1: COMPUTE KTH SCATTERER POSITION WRT ANTENNA LOS FRAME.**
- DO 45 J=1,3
  - 45 RL(J)=0.0
- DO 45 J=1,3
  - 45 RL(J)=RL(J)+TLT(J,I)*RT(K)*1
- DO 50 I=1,3
  - 50 RA(I)=ROI(I)+RL(I)

**STEP 4-2: COMPUTE RANGE OF KTH SCATTERER WRT RADAR.**
- RANGE(K)=SQRT(RA(1)**2+RA(2)**2+RA(3)**2)

**STEP 4-3: COMPUTE UNIT VECTOR IN DIRECTION OF KTH SCATTERER WRT...**
C DO 55 I=1,3
55 RAU(I,K)=RA(I)/RANGE(K)

**********************************************************************
STEP 5-1: COMPUTE KTH SCATTERER VELOCITY COMPONENTS WRT ANTENNA
LOS FRAME.
CALL MULT31I(TLTD,RT,RLD)
DO 60 I=1,3
60 RAD(I)=ROD(I)*RLD(I)

STEP 5-2: COMPUTE KTH SCATTERER RADIAL VELOCITY WRT TO RADAR.
RAUVEL(K)=0.0
DO 65 I=1,3
65 RAUVEL(K)=RAU(I,K)*RAD(I)*RADVEL(I)
70 CONTINUE

NOTE: DEBUGGING PRINT STATEMENTS.
WRITE(6,900) R0(I),R02(I),R03(I),CGKE,CGVEL
WRITE(6,902) WRITE(6,902) WRITE(6,902)
WRITE(6,903) I,[RT(I),I+1,3],SIG(I),I=1,N20)
900 WRITE(6,900) WRITE(6,900) WRITE(6,900)
901 WRITE(6,901) RAU,R0AU,R03,R0V=SF10.2)
902 WRITE(6,902) SPAS, RCS DATA
903 WRITE(6,903)(I,(RT(I)),I=1,2),,SIG(I) ,I,MOD(I,1)
RETURN
END

*************************************************************************
* THIS SUBROUTINE GENERATES THE NOISE-FREE ANGLE, RANGE, VELOCITY *
* AND ON-TARGET DISCRIMINANT COMPONENTS. *
*************************************************************************
SUBROUTINE SIGNAL
COMMON /CNTL/IPWR,IMODE,ITXP,IASM,IDUMC(5),IDUMC(3)
COMMON /CNTL/IDUM(5),IKINT,MANG,EBAM,MBRF,MBKTRK,MBSUM,
2 MBT(8)
COMM0N /TGTDAT/NRAU3(100),RANGE(100),RADVEL(100),R03(3),
2 R0A3(CGKE,CGVEL)
COMM0N /SATDAT/RADAR(3),N20,RTT0),SIG(70)
COMMON /TDAT/IDUM(2),DUN(5),MDF(5)
COMMON /SIGDAT/SPAZ,SM2,SPH,SM,EARLY,LATE,DF1,DF5,
2 DF2,DF4,SGBAR
COMMON /XFORMS/TLBD(3,3),TLB(3,3),TLT(3,3),TLTD(3,3)
COMPLEX CSUM,CDIFAZ,CDF23,CDF4,CDF5,CDF2,CDF4,
2 DF5,DF1,DF2,DF4,DF5
DIMENSION CT(102),DFTS,100),ALAM(5),ALAMO(5),NFREQ(2),
2 GRLD(3)
REAL CTP
2 DATA CTP/9.033189,799E-4,4.03316,1.9599E-3,9.8E-4,4.9E-4,
2 26.4S-4.1.251-4/
2 DATA NFREO/1.5,ALAM/177,439,178.05,178.71,178.71,178.04/,
2 DATA ALAM/1.221,2.9512,2.9512,2.9512,2.9512/,
2 REAL LATE

*************************************************************************
* 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
SPHEL = 0.0
SMEG = 0.0
EARY = 0.0
DFE = 0.0
DFE = 0.0
SIGBAR = 0.0

NFMAX = NREF(I-MODE)
ND 55 = 1, NFMAX

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

CSUM = 0.0
CSFEL = 0.0
CUNT = 0.0
CUTE = 0.0
CUNF = 0.0
CUTE = 0.0
CUT = 0.0

STEP 2-1: COMPUTE SUM CHANNEL MULTIPLICATION FACTOR FOR KTH SCATTERER.

PSI = ACOS(ABS(RAU3.K))

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

X = SPAT(PSI)

STEP 2-3: COMPUTE RX CHANNEL MULTIPLICATION FACTOR.

XX = SIG(K)*X

NOTE: IF IN ACTIVE MODE SET XX; 1.0.
IF(I-MODE.EQ.1) XX = 1.0

STEP 2-4: CHECK ANTENNA STEERING MODE (IF IN GPD-DES OR MANUAL

IF(I-MASE.EQ.2 .OR. I-ASM.EQ.4) GO TO 20

STEP 3-1: COMPUTE AZ AND EL DIFFERENCE CHANNEL MULTIPLICATION FACTORS FOR KTH SCATTERER.

DELAZ = ASIN(RA[U2.K])
DEE = ASIN(RAU[1.K])
STEP 3-2: COMPUTE AZ AND EL DIFFERENCE PATTERN MULTIPLICATION FACTORS.

Y=|DATAZ|;
Z=|DATDELZ|

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

DZ=Z*X;
DEL=AZ*Z

STEP 3-4: COMPUTE RANGE GATE LOCATION WRT RANGE GATE CENTER.

20 DELX=CTPMING, IMODE*(RANGE(K)-RANG)

STEP 4-1: COMPUTE RANGE GATE WEIGHTING FOR KTH SCATTERER.

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

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

II=INT((DELX+20)/21);
IF(II<6) II=6;
GO TO [21,22,23,24,21], II

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

25 RGWT=O.5*(RG+RGE)

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

RGE=S*RG;
RGL=S*RGL;
S=R+RGWT;
DAZ=DAZ*RGWT;
DEL=DEL*RGWT

STEP 5-1: COMPUTE DOPPLER FILTER PHASE SHIFT AND WEIGHTING FOR KTH SCATTERER.

DEFINITION: ALAMD(MPFR)=2.*PI/(PRF*LAMBDAA)

DEFINITION: THE CONSTANT 0.196346=PI/16.

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

FOT=2.*ALAMD(MPFR)*RADVEL(K)

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

C

DO 30 J=1,5
ARG=0.19646*MDF(J)-FDTC
30 DFWTS(J,K)=UGPFIL(ARG)

C

C ****^s**s*s**ssss*s*s*#***ss*s^s***ssss*****ss*ss***#*ss^ssss***#

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

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

C * STEP 6-2: COMPUTE PHASE FACTOR, I.E. EXP(J*DELPHI).

C * STEP 6-3: COMBINE RANGE PHASE FACTOR AND DOPPLER FILTER #3

C WEIGTH AND PHASE FACTOR.

C PHASE=PHASE*DFWTS(3,K)

C

C

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

C

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

STEP 7-3: ADD KTH SCATTERER CONTRIBUTION TO AZ AND EL DIFFERENCE

CHANNELS SIGNALS:

CDF2=CDF2+PHASE*DFWTS(2,K)

CDF4=CDF4+PHASE*I*DFWTS(4,K)

STEP 7-4: ADD KTH SCATTERER CONTRIBUTION TO RANGE DISCRIMINANT

COMPONENT SIGNALS:

CLATE=CLATE+RGL*PHASE

STEP 7-5: ADD KTH SCATTERER CONTRIBUTION TO VELOCITY DISCRIMINANT

COMPONENT SIGNALS:

CDF2=CDF2+PHASE*I*DFWTS(2,K)

CDF4=CDF4+PHASE*I*DFWTS(4,K)

STEP 7-6: ADD KTH SCATTERER CONTRIBUTION TO ON-TARGET DISCRIMINANT

COMPONENT SIGNALS:

CDF1=CDF1+PHASE*DFWTS(1,K)

CDF5=CDF5+PHASE*DFWTS(5,K)

STEP 7-7: CHECK ANTENNA STEERING MODE --- ADD STRONG DISCRIMINANT COMPONENTS AT 7TH FREQUENCY AND SQUARE

STEP 7-8: LAW DETECT THESE COMPONENTS.

STEP 7-9: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET

STEP 7-10: IF IN GPC-DES OR MANUAL

IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 40

STEP 8-1: CHECK ANTENNA STEERING MODE --- SKIP STEPS 9-2 AND 9-3

IF(IASM.EQ.2.OR.IASM.EQ.4) GO TO 50

STEP 8-2: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET

STEP 8-3: LAW DETECT THESE COMPONENTS.

STEP 8-4: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET

STEP 8-5: LAW DETECT THESE COMPONENTS.

STEP 8-6: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET

STEP 8-7: LAW DETECT THESE COMPONENTS.

STEP 8-8: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET

STEP 8-9: LAW DETECT THESE COMPONENTS.

STEP 8-10: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET

STEP 8-11: LAW DETECT THESE COMPONENTS.

STEP 8-12: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET

STEP 8-13: LAW DETECT THESE COMPONENTS.

STEP 8-14: FORM NOISE-FREE ANGLE, RANGE, VELOCITY, AND ON-TARGET

STEP 8-15: LAW DETECT THESE COMPONENTS.
SUBROUTINE CISCRM
COMMON /CNTI/1PMR,IMODE,ITXP,IASM9IOUMC(S),DUMC(3)
COMMON /ICNTTL/13DUM114)
2 COMMON /SYSDAT/DUM1(2),MARC,HAS,MPRF,1DUM4(10)
2 COMMON /TGIDAT/NT,DUM5(506),CGRANGE,CGVEL
COMMON /DSCRM/A2DISC,ELDISC,RD1SC,V01SC,RRTE,OD1SC,SIGBR1,SRND,2
2 COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,OF5,1DF5,2
2 COMMON /NOISE/NS1,NS2,NN(10),GAUSS1(200)
DIMENSION NFREQ(2),PO1A(2),POR1(2),PDIV(2),PS(10,2),SN(2),PT(3)
DATA NFREQ/1.5,8/9100.52b./PS/4*1.92.1.2.4.*8.16./
2 PT/50000.,3122.,195.3/
REAL LATE,MEAN

**STEP 8-1: COMPUTE AZ DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.**

\[ \text{SMAZ} = \text{SMAZ} + \text{CAB} \times \text{CSUM} \times \text{CFAFAZ} \times 2 \]

**STEP 8-3: COMPUTE EL DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.**

\[ \text{SPEL} = \text{SPEL} + \text{CAB} \times \text{CSUM} \times \text{CFELE} \times 2 \]

**STEP 8-6: COMPUTE ON-TARGET DISCRIMINANT COMPONENTS AND SQUARE-LAW DETECT.**

\[ \text{DF} = \text{DF} + \text{CAB} \times \text{CDEF} \times 2 \]

**STEP 9: COMPUTE EFFECTIVE CROSS-SECTION AVERAGED OVER PROPER NUMBER OF TRANSMIT FREQUENCIES.**

\[ \text{SIGBAR} = \text{SIGBAR} + \text{CAB} \times \text{CSUM} \times 2 \]

**NOTE: DEBUGGING PRINT STATEMENTS**

\[ \text{WRITE}(6,900) (I, SIG(I)), 1=1, NT) \]

\[ \text{WRITE}(6,902) NT,S1,DAZ,DEL,AGE,RGL, RGWGT, MDF(3) \]

\[ \text{WRITE}(6,901) DFMTS(1K), DFMTS(2K), DFMTS(3,1),DFMTS(4,1) \]

**SUBROUTINE DSCRM**

COMMON /CINTL/1PR,IMODE,ITXP,IASM9IOUMC(S),DUMC(3)
COMMON /ICINTL/13DUM1(4),MARC,HAS,MPRF,1DUM4(10)
COMMON /SYSDAT/TSA,SR1(3),CP,SP,PS,PSIAS,ALBIAS,BTBIAS,GP,GA, 2
DUM3(3)
COMMON /TGIDAT/NT,DUM5(506),CGRANGE,CGVEL
COMMON /DSCRM/A2DISC,ELDISC,RD1SC,V01SC,RRTE,OD1SC,SIGBR1,SRND,2
COMMON /SIGDAT/SPAZ,SMAZ,SPEL,SMEL,EARLY,LATE,OF5,1DF5,2
COMMON /NOISE/NS1,NS2,NN(10),GAUSS1(200)
DIMENSION NFREQ(2),PO1A(2),POR1(2),PDIV(2),PS(10,2),SN(2),PT(3)
DATA NFREQ/1.5,8/9100.52b./PS/4*1.92.1.2.4.*8.16./
2 PT/50000.,3122.,195.3/
REAL LATE,MEAN

293
NOTE: Debugging print statements.
WRITE(6,900) SPAZ, SNAZ, SMEL, EARLY, LATE
WRITE(6,901) DF1, DF5, DF2, DF4, SIG, BAR
900 FORMAT(DF1, DF5, DF2, DF4, SIG *)
901 FORMAT(DF1, DF5, DF2, DF4, SIG */)

***********************************************************************
* STEP 1: Compute constant used in signal scaling and computation *
* of noise statistics.                                      *
***********************************************************************

STEP 1-1: Compute constant (NOTE: IT IS DIFFERENT FOR ACTIVE AND *
PASSIVE MODES).                                                  *
IF(IMUDE.EQ.2) GO TO 5                                           *

STEP 1-1-1: Compute constant (NOTE: IT IS DIFFERENT FOR ACTIVE AND *
PAS SIVE MODES).                                                  *

NOTE: THIS IS THE CONSTANT USED IN ACTIVE MODE.                    *
YY=CAPS(MENG, IMODE)/(CGRNGE*2*BNSAM))                           *

NOTE: THIS IS THE CONSTANT USED IN PASSIVE MODE.                   *

YY=CAPS(MENG, IMODE)*PI1/(CGRNGE*4*BNSAM))                        *

SI=YY/FLAT(FREQ(IMODE))                                          *

STEP 1-2: Compute the SNR at the output of the Doppler filter      *
(Note: THIS IS USED FOR DEBUGGING PURPOSES ONLY).                  *

10 SNR=YY*SIGBAR                                                  *
SNRD=10.*ALOG10(SNRD)                                            *
SIGDB=10.*ALOG10(SIGBAR)                                         *
SIGBAR1=SIGBAR                                                   *

STEP 1-3: Update noise sequence.                                  *

DO 15 I=2,6                                                       *
15 NN(I)=MOD(NN(I-1)+29,200)+1                                   *
GAUS(SG)=AONR(N,NS2)                                             *

***********************************************************************
* STEP 2: Compute angle discriminant (includes noise) = *
***********************************************************************

STEP 2-1: Check antenna steering mode --- SKIP STEP 2 IF IN *
GPE-DES OR MANUAL.                                               *
IF(IASM.GE.2.OR. IASM.LE.4) GO TO 20                              *

STEP 2-2: Compute angle discriminant component scale factor.       *

ASCAL=SI*PDIA(IMODE)                                             *

STEP 2-3: Compute statistics of additive noise for angle          *
discriminant components.                                         *

MEAN=PDIA(IMODE)                                                 *
VARPA=SQRT(2.*SI*SPAZ+1.)                                        *
VARMZ=SQRT(2.*SI*SNAZ+1.)                                       *
VARPE=SQRT(2.*SI*SMEL+1.)                                       *
VARMLE=SQRT(2.*SI*SMEL+1.)                                      *

STEP 2-4: Add equivalent noise to angle discriminant component     *
signals.                                                        *

ID=NN(4)                                                        *
SPAZ=ABS(ASCAL*SPAZ+MEAN+VARPA*GAUS(SG))                         *
SNAZ=ABS(ASCAL*SNAZ+MEAN+VARMZ*GAUS(SG))                         *
SMEL=ABS(ASCAL*SMEL+MEAN+VARMLE*GAUS(SG))                        *

STEP 2-5: Compute A and El discriminant components.               *

ELDISC=10.*ALOG10(SPAZ/SNAZ)                                     *
ELDISC=10.*ALOG10(SPEL/SMEL)                                     *
**STEP 31** COMPUTE RANGE DISCRIMINANT (INCLUDES NOISE)

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

\[ \text{RSCALE} = \text{S1} \times \text{POIR} (\text{IMODE}) \]

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

\[ \text{MEAN} = \text{POIR} (\text{IMODE}) \]
\[ \text{VARELY} = \text{SQRT}(2.0 \times \text{EARY} + 1.1) \]
\[ \text{VARLTE} = \text{SQRT}(2.0 \times \text{LATE} + 1.1) \]

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

\[ \text{ID3} = \text{NN16} \times \text{EARLY} \]
\[ \text{ID6} = \text{NN16} \times \text{LATE} \]
\[ \text{ID3} = \text{ABS} \left( \text{RSCALE} \times \text{EARLY} \right) + \text{MEAN} + \text{VARELY} \times \text{GAUSS} (\text{ID3}) \]
\[ \text{ID6} = \text{ABS} \left( \text{RSCALE} \times \text{LATE} \right) + \text{MEAN} + \text{VARLTE} \times \text{GAUSS} (\text{ID6}) \]

*STEP 3-4: COMPUTE RANGE DISCRIMINANT.*

\[ \text{RDISC} = 10.0 \times \text{ALOGIO} (\text{LATE} / \text{EARLY}) \]

**STEP 4** COMPUTE VELOCITY DISCRIMINANT (INCLUDES NOISE)

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

\[ \text{VSCALE} = \text{S1} \times \text{POIV} (\text{IMODE}) \]

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

\[ \text{MEAN} = \text{POIV} (\text{IMODE}) \]
\[ \text{VAROF2} = \text{SQRT}(2.0 \times \text{OF2} + 1.1) \]
\[ \text{VARDF4} = \text{SQRT}(2.0 \times \text{DF4} + 1.1) \]

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

\[ \text{DF2} = \text{ABS} \left( \text{VSCALE} \times \text{DF2} \right) + \text{MEAN} + \text{VARDF2} \times \text{GAUSS} (\text{ID3}) \]
\[ \text{DF4} = \text{ABS} \left( \text{VSCALE} \times \text{DF4} \right) + \text{MEAN} + \text{VARDF4} \times \text{GAUSS} (\text{ID6}) \]

*STEP 4-4: COMPUTE VELOCITY DISCRIMINANT.*

\[ \text{VDISC} = 10.0 \times \text{ALOGIO} (\text{DF2} / \text{DF4}) \]

**STEP 5** COMPUTE ON-TARGET DISCRIMINANT — USED FOR BREAK-TRACK AND VELOCITY DETERMINATION

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

\[ \text{VARDF1} = \text{SQRT}(2.0 \times \text{DF1} + 1.1) \]
\[ \text{VARDF5} = \text{SQRT}(2.0 \times \text{DF5} + 1.1) \]

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

\[ \text{DF1} = \text{ABS} \left( \text{VSCALE} \times \text{DF1} \right) + \text{MEAN} + \text{VARDF1} \times \text{GAUSS} (\text{ID2}) \]
\[ \text{DF5} = \text{ABS} \left( \text{VSCALE} \times \text{DF5} \right) + \text{MEAN} + \text{VARDF5} \times \text{GAUSS} (\text{ID6}) \]

*STEP 5-3: COMPUTE ON-TARGET DISCRIMINANT.*

\[ \text{ODISC} = 10.0 \times \text{ALOGIO} (\text{EARY} + \text{LATE}) / \text{SQRT}(2.0 \times (\text{DF1} + \text{DF5})) \]

*NOTE: DEBUGGING PRINT STATEMENTS.*
* THIS SUBROUTINE IMPLEMENTS THE BREAK-TRACK ALGORITHM *

SUBROUTINE BKTRK
COMMON /ICNL/IDUM2,MBKTRK,MBTUM,MBT(8)
COMMON /DSCAM/DUM(3),VDISC,DUM1,ODISC,DUM2(3)
INTEGER THRSHO,THRSHC
DATA IVMAX,NTHSHO/51,5,-1/
NOTE= USED IN THE RADAR. THESE VALUES MUST BE CHANGED TO THE RADAR VALUES.

* STEP 1: DETERMINE STATUS OF L-H DISCRETE (FTH) *
STEP1-1: QUANTIZE THE VELOCITY DISCRIMINANT TO 3/16 DB STEPS.
IVDISC=INT(IVDISC*5.333333333333+0.5)
STEP1-2: DETERMINE STATUS OF L-H DISCRETE.
IFTH=0 IF(IABS(IVDISC).GT.IVMAX) IFTH=1

* STEP 2: DETERMINE STATUS OF ON-TARGET DISCRETE (OT) *
STEP2-1: QUANTIZE THE 0-DISCRIMINANT TO 3/16 DB STEPS.
IODISC=INT(IODISC*5.333333333333+0.5)
STEP2-2: DETERMINE STATUS OF ON-TARGET DISCRIMINANT.
IOT=0 IF(IODISC.GT.THRSHC) IOT=1

* STEP 3: DETERMINE STATUS OF ADJACENT ON-TARGET DISCRETE (AOT) *
STEP3-1: QUANTIZE THE 0-DISCRIMINANT TO 3/16 DB STEPS.
IAOT=0 IF(IODISC.LT.THRSHO) IAOT=1

* STEP 4: COMBINE ABOVE DISCRETE 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)

WRITE(6,903) SNRD,SLDQ,SIGBAR
WRITE(6,904) SNRD,SLDQ,SIGBAR
WRITE(6,905) DF1,DF2,DF4,SIGBAR
902 FORMAT(7,A6,4E16.6,F10.2)
903 FORMAT(7,A6,4E16.6,F10.2)
904 FORMAT(1,A6,4E16.6,F10.2)
RETURN
END
**STEP 5**: DETERMINE STATUS OF BREAK-TRACK FLAG (MBKTRK).

**DEFINITIONS**

### ssfw ## + M####i#####ss#i## * i## Epp

RD #### i**s####s####G

### ssww### ss## ss###

LEA

AT  ST 5 KA OF THENO RECENT 8 DATA CYCLES 1 FOR AT

STEP M5-1: UMBTSUM MOVING G WIM DDW-O; SUN (MBTSUM).

STEP 5-2: UPDATE STORAGE REGISTERS.

```
DO 10 I=17
10 MBT(I)=MBT(I+1)
MBT(18)=NOTARG
```

STEP 5-3: DETERMINE STATUS OF BREAK-TRACK FLAG (1=BREAK-TRACK).

```
MBKTRK=MBTSUM/5
```

**END**

**SUBROUTINE ATRACK COMMON /CNTL/IPMR (MODE 1**

```
COMMON /INPUT/DMC(7),DMC(3)
COMMON /OUTPUT/DMC(7),DMC(18)
COMMON /ICNTL/DMC(1),MRNG,DMC(12)
COMMON /CROS/PSI,PSIAS,ALBIAS,OTBIAS,
2 COMMON /ATOT/CA,SA,SB,AZRATE,ELRATE,ALRATE,BTRATE,AL,AT
2 COMMON /DUM/DMC(3)
2 COMMON /DUM/DMC(7)
DIMENSION AT1,AT2,AT3,AT(5),AT(12),TX(3,3),TB(1,3)
DATA AT1,-1.3529E-3,2.0106E-1,GO3,9750E-3,1.5529E-3,
3.20.0106E-1,AT1/2.99E-6,1.907E-2,3.1225E-3,
6.9.046E-2,6.3907E-3,3.325E-3/7
DEFINITION: AT1=QT=Q^(MN**2)/14^DIFERENCE PATTERN SLOPE) WHERE
WN IS NATURAL FREQUENCY OF THE LOOP.
DEFINITION: AT2=Q^TAU WHERE Tau IS PROPORTIONAL TO STEP RESPONSE
CONVERGENCE TIME.
```

**STEP 1**: UPDATE ANTENNA LOS-TO-BODY TRANSFORMATION (NOTE: TRANS-
FORMATION INCLUDES GIMBAL BIAS ERRORS AND RADAR YAM-
ANGLE ERROR WRIT BODY FRAME).

CALL GAMMA(TX1,-(BT+BTIAS))
CALL THEETA(TX2,-(AT+ALIAS))
CALL MULT33(TX2,TX1,TX3)
CALL PHII(TX2,PSI1)
CALL MULT33(TX2,TX3,TBL)

**STEP 2**: UPDATE ESTIMATED TARGET INERTIAL AZIMUTH AND ELEVATION
RATES IN ANTENNA LOS FRAME.

**QUANTIZE THE ANGLE DISCRIMINANTS TO 3/16 DB**

```
IADSC=INT(3.33333333*AIDSC)
IELDSC=INT(3.33333333*ELDISC)
ADSC=0.0431*FLOAT(AIDSC)
TBLDSC=INT(3.33333333*ELDISC)
```

**UPDATE ESTIMATED TARGET INERTIAL AZIMUTH RATE**

```
A#RATE=AZRATE+TSAMAT11(MRNG,IMODE)*ADSC
```

```
```
UPDATE ESTIMATED TARGET INERTIAL ELEVATION RATE.
ELRATE=ELRATE+TSAM*AT2(MRNG,IMODE)*EDSC

* STEP 3: UPDATE INNER AND OUTER GIMBAL RATES.

OUTER GIMBAL FRAME:
WGX=CP*EWB(1).*SEWB(2)
WGY=CA*(SP*EBW(1)+CP*EBW(2))
WGZ=SA*(-SP*EBW(1)+CP*EBW(2))

OUTER GIMBAL RATE:
IF (TBL(1,-1)<=0) GO TO 2
ALRATE=(AZRATE+AT2(MRNG,IMODE)*ADSC+WGZ*SB1/CB-WGX
GO TO 4
2 ALRATE=0
4 CONTINUE

INNER GIMBAL RATE:
BTRATE=(ELRATE+AT2(MRNG,IMODE)*EDSC)-WGY

* STEP 4: UPDATE INNER AND OUTER GIMBAL POSITIONS.

OUTER GIMBAL POSITION (ALPHA ANGLE):
AL=AL+TSAM*ALRATE

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

* STEP 5: ANTENNA IN OBSCURATION REGION?

CALL SCRNRN

* STEP 6: TRANSFORM TARGET ANGLES AND INERTIAL ANGLE RATES TO

OUTER GIMBAL 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.

UPDATE TARGET INERTIAL PITCH RATE IN ORBITER BODY COORDINATES
FOR DISPLAY.
SPRTE=-100.(TBL(2,1))*AZRATE+TBL(2,2)*ELRATE

UPDATE TARGET INERTIAL ROLL RATE IN ORBITER BODY COORDINATES
FOR DISPLAY.
SRRTE=-100.-(TBL(1,1))*AZRATE+TBL(1,2)*ELRATE

UPDATE ANTEenna PITCH ANGLE IN ORBITER BODY COORDINATES FOR DISPLAY.
SPANG=-ASIN(TBL(1,3))=57.29576

UPDATE ANTEenna ANGLES IN ORBITER BODY COORDINATES FOR DISPLAY.
IF(TBL(1,3).EQ.0) GO TO 5
SRANG=ATAN2(-TBL(2,3),TBL(3,3))*EDSC
GO TO 7
5 IF(TBL(1,3).LT.0.01) SRANG=90.0
IF(TBL(1,3).GT.0.01) SRANG=-90.0
STOP
7 IF(SPANG.LE.90) GO TO 10
SRANG=180.-ABS(SPANG) IF(SPANGABS(SPANG)
SRANG=180.-ABS(SRANG) IF(SRANGABS(SRANG)
10 CONTINUE

NOTE: DEBUGGING PRINT STATEMENTS.

* 899 FORMAT(99)
WRITE(6,899)
WRITE(6,900) ALRATE,STRATE,ELRATE,SRATE,SPRTE
WRITE(6,901) TBL(1,1),TBL(1,2),TBL(2,1),TBL(2,2)
WRITE(6,902) ADSC,ELDISC,ADSC,EDSC
900 FORMAT(99)
WRITE(6,900) ALRE,TRATE,ELRE,SRRE,SPRTE
WRITE(6,901) TBL(1,1),TBL(1,2),TBL(2,1),TBL(2,2)
WRITE(6,902) ADSC,EIDSC,ADSC,EDSC

* THIS SUBROUTINE UPDATES RANGE AND RANGE RATE ESTIMATES:
* UPDATES DOPPLER FILTER BANK POSITION; AND SYSTEM PARA-
* METERS UPON RANGE INTERVAL.

SUBROUTINE RTRACK

COMMON /CNTL/PMRT,MODE,DMC(7),DNC(3)
COMMON /OUTPUT/DMC(2),SRNG,SRD,NMC(3),IMODE
COMMON /SYSDAT/TM,DMC(14)
COMMON /RTDAT/IRNG,RBIAS,VEST(4),MDF(9)
COMMON /OMNI/DMC(3)

DIMENSION IPROM(128),RT(101),R(110),V(13),V2(13)

DATA IPROM/1,2,7,12,14,22,12,12,12,12,12,12,12,12,12,12,12,12
1,2,7,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12

DATA IR/71/124.780,0.3,7,17,1.7,1.7,1,1.7,1
2,21.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7
3,21.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7,1.7
4,0.51638,0.4,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}

END

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

* R-TRACKER MODEL ********** RANGE TRACER MODEL **********************

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

* STEP 1: UPDATE ROUGH RANGE RATE ESTIMATE *

INTEGRATE RANGE DISCRIMINANT AND CHECK FOR SATURATION.
IRDISC=3.333333*IRDISC
IF(IRDISC.LT.250) IRDISC=250,
IF(IRDISC.LT.256) IRDISC=256
ROUGH RANGE RATE PREDICTION FROM ALPHA-BETA TRACKING EQUATIONS.
RR=FLOAT(IRDISC)*RT(MNG. IMODE)
RDOT=IRDOT+INT(RR)*0.5,
STEP 2: UPDATE RANGE ESTIMATE *

DEFINITION: RT2 CORRESPONDS TO ALPHA IN ALPHA-BETA TRACKER.
R=FLOAT(IRDISC)*RT(MNG. IMODE)
RR=RR+IRG+RDOT+INT(RR)
CONVERT RANGE ESTIMATE (IRG) TO FEET USING THE FACT THAT THE LSB
OF IRG REPRESENTS 0.3125 FT.
RNG=0.3125*FLOAT(IRG)
ADD FIXED RIAS TO FINAL RANGE ESTIMATE.
SRNG=RNG+RBIAS

END
STEP 1: GENERATE AMBIGUOUS VELOCITY ESTIMATE.

1. INTEGRATE VELOCITY DISCRIMINANT AND CHECK FOR SATURATION.
   - \( V_{out} = \int V_{in} \, dt \) if \( V_{in} > 0 \), \( V_{out} = 0 \) otherwise.
   - \( V_{out} \leq V_{max} \)

2. COMPUTE INTEGRAL FILTER NUMBER PORTION OF AMBIGUOUS VELOCITY ESTIMATE.
   - \( \text{INT} = \text{MOD}(\text{INT}, \text{MOD}(\text{INT} + 1, 2)) \)

3. COMPUTE FRACTIONAL FILTER PORTION OF AMBIGUOUS VELOCITY.
   - \( \text{FRAC} = \text{Frac}(\text{INT}) \)

4. COMPUTE AMBIGUOUS VELOCITY ESTIMATE.
   - \( V_{amb} = \text{INT} + \text{FRAC} \)

STEP 2: SCALE ROUGH VELOCITY ESTIMATE.

1. SCALE LSB OF ROUGH VELOCITY TO 4 TIMES A DOPPLER FILTER WIDTH
   - \( V_{scale} = V_{amb} \times 4 \)

2. COMPUTE AMBIGUOUS VELOCITY ESTIMATE.

STEP 3: RESOLVE AMBIGUITY.

1. COMPUTE LSBS OF SCALED ROUGH VELOCITY ESTIMATE.

2. CONTINUE.

STEP 4: COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE.

1. SCALE LSB TO 0.05 FEET/SEC.

2. COMPUTE UNAMBIGUOUS VELOCITY ESTIMATE.

3. VERIFY CALCULATIONS.
C  ***********************************************************************
C  * STEP 31: COMPUTE SMOOTHED UNAMBIGUOUS VELOCITY *                 
C  ***********************************************************************
C DO 20 I=1,3
     20 VEST(I-1)=VEST(I-1)
C AVERAGE AND SCALE ANSWER INTO FEET/SEC plus 0.05 FEET/SEC.
     SMOOTH=0.0125*(VEST(1)+VEST(2)+VEST(3)+VEST(4))
C***********************************************************************
C  ***********************************************************************
C  * STEP 3: SET PRF PARAMETER *                                       
C  ***********************************************************************
C IF(MODE.GE.2) GO TO 84
     IF(MRNG.GT.9) GO TO 72
     MSAM=1
     GO TO 80
     MSAM=2
     GO TO 80
     IF(MRNG.GT.4) GO TO 76
     MSAM=1
     GO TO 80
     MSAM=2
C***********************************************************************
C  ***********************************************************************
C  * STEP 2: SET SAMPLE RATE PARAMETER *                                
C  ***********************************************************************
C IF(MODE.GE.2) GO TO 74
     IF(MRNG.GT.9) GO TO 72
     MSAM=1
     GO TO 80
     MSAM=2
     GO TO 80
     IF(MRNG.GT.4) GO TO 76
     MSAM=1
     GO TO 80
     MSAM=2
C***********************************************************************
C  ***********************************************************************
C  * STEP 1: SET RANGE INTERVAL PARAMETER *                             
C  ***********************************************************************
C DO 60 I=1,NRI
     IF(RNG.LE.RRI(I)) GO TO 70
     CONTINUE
     MRNG=1
     IF(MRNG.GT.NRI) STOP
C***********************************************************************
C  ***********************************************************************
C  * STEP 0: RE; ET DOPPLER FILTER BANK *                               
C  ***********************************************************************
C CHECK ON-TARGET DISCRIMINANT FOR LARGE ACCELERATION DURING A DATA
C CYCLE OR LOSS OF SIGNAL
     IF(MODE.GE.2) GO TO 30
     IF(MRNG.GT.2) MDF(I)=MOD(MDF(I)+36,32)
     GO TO 40
     30 IF(MRNG.GT.51) MDF(I)=MOD(MDF(I)+31,32)
     40 IF(MRNG.GT.51) MDF(I)=MOD(MDF(I)+31,32)
C  50 MDF(I+1)=MOD(MDF(I+1)+32)
C***********************************************************************
C  ***********************************************************************
C  * STEP 2: SET RANGE INTERVAL PARAMETER *                             
C  ***********************************************************************
C DO 60 I=1,NRI
     IF(RNG.LE.RRI(I)) GO TO 70
     CONTINUE
     MRNG=1
     IF(MRNG.GT.NRI) STOP
C***********************************************************************
C  ***********************************************************************
C  * STEP 3: SET PRF PARAMETER *                                       
C  ***********************************************************************
C IF(MODE.GE.2) GO TO 84
     IF(MRNG.GT.9) GO TO 72
     HPRF=1
     GO TO 80
     HPRF=2
     GO TO 80
     IF(MRNG.GT.9) GO TO 84
     HPRF=1
     GO TO 90
     HPRF=2
     CONTINUE
     RETURN
     END
* This subroutine computes the radar signal strength — it is set equal to the SNR in the present version.

**SUBROUTINE RSS**
COMMON /CTRL/DUM(2),ITXP,ITDUMC(6),DUMC(3)
COMMON /OUTPUT/DUMA1(I),IOM16(3),ASS,ITDUM(4)
COMMON /SIGDUM/DUM(2),SIGMAA,DUMA2(2)
COMMON /ACCDAY/AGC,ACCOLD

* STEP 1: DEFINE TARGET PARAMETERS *
SET THE RANGE TO TARGET C.G.
RANGE=C_RANGE
SET TARGET RADAR CROSS-SECTION (INCLUDES BEAMSHAPE LOSS).
SIGMA=SIGBAR

**SNR=SNR(SIGMA,RANGE)**

* STEP 2: COMPUTE THE SNR AT VIDEO OUTPUT *
$RSS=10^{ALOG10(SNR)}$
RETURN
END

**SUBROUTINE DATA**
COMMON /R/DUM(2),RBIAS,DUMC(9)
COMMON /SYS/DUMA1M(3),CP,SP,PSI,PSBIAS,ALBIAS,BTBIAS,GP,GAS,

* SYSTEM PARAMETERS *

**REAL LT,RTS**

**PI=1415926.**
**PI1=PI/180.**

**RADIUS FRAME YAW ANGLE IN BODY COORDINATES (DEGREES).**
**RPI=PI/180.**
**CP=COS(RPI).**

**RADIUS LOCATION OFFSET FROM ORBITER C.G. IN BODY COORD. (FEET)**
**DA[1]=48.0**
**DA[2]=1.0**
**DA[3]=1.0**

**RADIUS BIAS ERROR.**
**RBIAS=0.0**

**ALPHA GIMBAL BIAS.**
**ALBIAS=0.0**

**BETA GIMBAL BIAS.**
**BTBIAS=0.0**

**RADIUS YAW ANGLE ERROR WRT BODY FRAME.**
**PSBIAS=0.0**
**SYSTEM SAMPLE INTERVAL**

\[ TSAM = 0.2 \]

**COMPUTE SNR CONSTANT**

\[ \text{TSAM} = 0.2 \]

\[ LT = 10.0 \times (0.1 \times LT) \]

**EQUIVALENT ONE-SIDED NOISE POWER SPECTRAL DENSITY (MW/KHZ)**

\[ KTS = 136.6 \]

**SYSTEM LOSSES ON TRANSMIT (DB).**

\[ LT = 10.0 \times (0.1 \times LT) \]

**ONE-WAY ANTENNA GAIN (DB).**

\[ G = 38.5 \]

\[ G = 10.0 \times (0.1 \times G) \]

\[ \text{ALMBDA} = 0.070845 \]

**CONSTANT FOR PASSIVE TRACKING SNR COMPUTATION.**

\[ GP = 4.0 \times (\text{ALMBDA} \times 2)/(4.0 \times \pi \times 3 \times LT \times KTS) \]

**BEACON PARAMETER (DBM).**

\[ BCN = 44.0 \]

\[ BCN = 10.0 \times (0.1 \times BCN) \]

**CONSTANT FOR ACTIVE TRACKING SNR COMPUTATION.**

\[ GA = 6.0 \times (\text{ALMBDA} \times 2 \times BCN)/(4.0 \times \pi \times 2 \times KTS) \]

**CONSTANT FOR PASSIVE MODE VIDEO SNR COMPUTATION (DB).**

\[ GPS = 183.9 \]

**CONSTANT FOR ACTIVE MODE VIDEO SNR COMPUTATION (DB).**

\[ GAS = 146.9 \]

**RANDOM NUMBER GENERATOR SEEDS**

\[ NS = 48 \]

\[ NN(1) = 0 \]

**INITIALIZE NOISE SEQUENCE.**

\[ \text{DO 2 I=1,920} \]

\[ \text{2 GAUSS(I) = ANORM(NS1,NS2)} \]

**DEFINE TARGET PARAMETERS**

**TARGET SEARCH CROSS-SECTION (FIXED TEMPORARILY).**

\[ \text{TGTSIG} = 10.0 \]

RETURN

END

**FUNCTION SPAT(X)**

\[ \text{NOTE: THE FOLLOWING VALUE OF B GIVES THE SUM PATTERN A SINGLE-SIDED 3 DB BEAMWIDTH OF 0.85 DEGREES.} \]

\[ B = 93.80 \times X \]

\[ \text{TEMP} = \text{ABS}(Y) \]

\[ \text{IF} (\text{TEMP G.T.1.0E-06}) \text{GO TO 10} \]

\[ \text{SPAT} = 1.0 \]

RETURN

\[ \text{10 SPAT} = \text{SIN(Y)}/Y \]

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.E-4, GOTO 10)
  DPAT=0.0.228*X
  RETURN
10  Y=.93.80*X
  DPAT=1.1465*(Y*COS(Y)-SIN(Y))/(Y+Y)
  RETURN
END

* THIS FUNCTION GENERATES A RANDOM NUMBER FROM A GAUSSIAN PDF *
* WITH ZERO MEAN AND UNIT VARIANCE. *

FUNCTION ANORM(K1,K2)
  Y1=RNDU(K1)
  Y2=RNDU(K2)
  TPI=0.2031552
  ANORM=-2.*ALOG(Y1)*COS(TPI*Y2)
  RETURN
END

* THIS FUNCTION GENERATES A RANDOM NUMBER FROM A UNIFORM 90,11 *
* DISTRIBUTION. *

FUNCTION RNDU(IRAN)
  DATA MU=524287./I ETA/997./
  IF(IRAN) 20,10,20
  CONTINUE
  IRAN=I ETA*IRAN
  IKEEP=IRAN
  IRAN=IRAN—IKEEP*MU
  IRAN=IRAN
  RNDU=IRAN
  10 RETURN
END

* THIS FUNCTION COMPUTES THE DOPPLER FILTER OUTPUT AMPLITUDE *
* AND PHASE FOR AN INPUT SIGNAL OF FREQUENCY X. *

COMPLEX FUNCTION DOPFIL(X)
  COMPLEX DENOM,NUMER
  DENOM=1.—CEXP(CMPLX(0.,X))
  DENOM=16.—DENOM
  C CHECK FOR DENOMINATOR EQUAL TO ZERO.
  XX=CABS(DENOM)
  IF(xx.CT.1.0E-06) GO TO 10
  DOPFIL=(1.0,0.0)
  RETURN
10  NUMER=1.—CEXP(CMPLX(0.,16.*X))
  DOPFIL=NUMER/DENOM
  RETURN
END
** This function checks for negative argument for INT function

```
INTEGER FUNCTION INTT(Y)
  KEY
  IF (Y .LT. 0.0) X = Y - 1.0
  INT = INT(X)
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.

```
SUBROUTINE PHID(TPHD,PH,W)
  DIMENSION TPHD(3,3)
  DO 10 I=1,3
    TPHD(I,1) = 0.0
    TPHD(I,2) = W*SIN(PH)
    TPHD(I,3) = W*COS(PH)
  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
    DO 10 J=1,3
      C(I,J) = A(I,K)*B(K,J)
  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) = A(I,J)*B(J)
  END
```
* THIS SUBROUTINE GENERATES A (3X3) MATRIX TTH THAT PRODUCES A ROTATION OF TH RADIANS ABOUT THE X-AXIS.

```fortran
SUBROUTINE TMETA(TTH, TH)
DIMENSION TTH(3,3)
DO 10 I = 1, 3
    DO 10 J = 1, 3
        TTH(I,J) = 0.0
10     TTH(I,1) = 1.0
    TTH(2,2) = COS(TH)
    TTH(2,3) = SIN(TH)
    TTH(3,2) = -TTH(2,3)
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 10 J = 1, 3
        TPH(I,J) = 0.0
10     TPH(3,3) = 1.0
    TPH(1,1) = COS(PH)
    TPH(2,1) = -SIN(PH)
    TPH(1,2) = SIN(PH)
    TPH(2,2) = TPH(1,1)
RETURN
END
```

* THIS SUBROUTINE GENERATES A (3X3) MATRIX TGA THAT PRODUCES A ROTATION OF GA RADIANS ABOUT THE Y-AXIS.

```fortran
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(1,1) = 1.0
    TGA(1,3) = COS(GA)
    TGA(2,3) = SIN(GA)
    TGA(3,3) = TGA(1,1)
    TGA(1,3) = -TGA(3,3)
RETURN
END
```
SUBROUTINE SPAS
COMMON / SATAT / RADAR (3), RSTAR (570, 3), SIG (570), ROLD, ICLOSE, ICLUD
DIMENSION SIGMA(63), TARG(63, 3), PHI (MIN(63, 3), PHIMAX(63, 3))
DIMENSION OFFSET(63), JHOT(63), JHOT20(63), PNL(63, 3)
DIMENSION VECT(31, COSPHI(63, 3))
DIMENSION DPAT(20, 3), DPAT20(20, 3), WSCALE(101), OPHI(101, 0), VOLA(20, 31)
DIMENSION VSCALE(101, 101, 0)

* 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 RSCALE /14679896787, 9867896789, 6789678967, 6789678967, 6789678967,
2 51, 23, 32, 43, 55, 67, 55, 89, 55,
3 71, 27, 37, 47, 57, 67, 77, 87, 97,
4 97, 24, 34, 44, 54, 64, 74, 84, 94,
5 20, 30, 40, 50, 60, 70, 80, 90, 100,

DATA DESCRIBING DIMENSIONS OF WIDE-ANGLE SCATTERERS
DEFINITION: WSCALE = SQRT (D^2/(12* NF)) (UNITS = FEET, NF = # OF FREQUENCY)
DATA DIM /60*64.8/
DATA WSCALE /60*0.2965/

FOR EACH DIFFUSE SCATTERER, SPECIFY NORMAL COMPONENT
DATA NORMAL /3*1929299*3, 6*1/

SQUARE ROOT OF RCS VALUES (FEET)
DATA SIGMA/20*.73493*5929 6*25.6 16.6 109.96.4 104.95.69 3.69/.2/.99 3.69
A-COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
DATA TARG /3*.39177*-.1593*.79, 6*1.21
B-COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
DATA TARG /3*.39177*-.1593*.79, 6*1.21
C-COORDINATES OF SCATTERERS IN SPAS FRAME (FEET)
DATA TARG /3*.39177*-.1593*.79, 6*1.21

ORIGINAL PAGE IS OF POOR QUALITY
307
MINIMUM SUBTENDED ANGLE IN X-DIRECTION
DATA PHIMIN /18.60, 9.10, 9.10, 0.10, 0.10, 9.10, 10.00, 0.26, 0.03/.

MINIMUM SUBTENDED ANGLE IN Y-DIRECTION
2 4 1.13, 0.42, 0.81, 0.30, 0.29, 2.29, 9.06, 0.68, 9.06/.
3 4 0.50, 0.50, -0.99, 0.99, -0.99, 0.99, 1.00, 0.05, 1.00/.
4 4 -0.94, 1.8, 1.00/.

MINIMUM SUBTENDED ANGLE IN Z-DIRECTION
5 5 1.3, 0.12, 0.71, 0.02, 0.02, 2.04, 0.86, 0.0, 0.86/.
6 4 -0.43, 0.43, 0.90, 0.1, -0.99, 0.99, 0.99, 1.0, -0.99/.
7 1 -0.99, 0.99, -0.99, 0.99, -0.99, 0.99, 0.47, 0.0, 1.0/.
8 0 0, -9.15, 0, 0/.

MAXIMUM SUBTENDED ANGLE IN X-DIRECTION
DATA PHIMAX /3.00, 1.17, 0.10, 0.10, 1.30, 9.99, 0.50, 1.00, 0.26/.
2 10 -0.02, 0.03, 0.0, 0.0, 2.0, 3.0, 0.99, 0.05, 0.0/.

MAXIMUM SUBTENDED ANGLE IN Y-DIRECTION
3 3 -1.0, 4.2, 8.6, 9.0, 9.0, 9.0, -0.19, 1.3, 0.9, 0.9/.
4 -0.70, 0.91, 0.91, 0.91, 0.91, 0.91, -0.05, 1.0, 0.0, 0.0/.
5 9 1.0/.

MAXIMUM SUBTENDED ANGLE IN Z-DIRECTION
6 6 -1.6, 0.8, -1.1, 0.3, -0.71, 0.5, -0.02, 0.0, -0.40, 0.3, -0.02, 0.0/.
7 7 -0.45, -0.45, -0.45, -0.45, 0.5, 0.5, -0.90, 0.0, -0.90, 0.0/.
8 -0.99, 0.99, -0.99, 0.99, -0.99, 0.99, -0.99, 0.99, -0.99, 0.99/.
9 -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0/.

RADII OF THE SCATTERERS (FEET)
DATA OFFSET /2.0, 0.0, 3.3, 0.95, 1.03, 0.70, 0.79, 1.70, 0.0/.

MISCELLANEOUS DATA
DATA NPAR /63/, KWIDTH /20/, PI /3.141592653/.

* STEP 1: DETERMINE WHICH SCATTERER ARE ILLUMINATED AND HAVE A * NONZERO RCS IN THE DIRECTION OF THE RADAR.

* STEP 1-1: PERFORM REQUIRED INITIALIZATIONS.
NWIDTH = 0
KNTAR = 0

* STEP 1-2: COMPUTE UNIT VECTOR IN DIRECTION OF RADAR FOR ITH SCATTERING CENTER.
DO 15 I = 1, NPAR
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
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.
IF (COSPHI (I, J) .LT. PHIMAX (I, J) .OR. COSPHI (I, J) .GT. PHIMIN (I, J))
2 GO TO 15
10 CONTINUE

308
STEP 1-4: IF ITH SCATTERER RCS IS NONZERO THEN ADD TO VECTOR OF ILLUMINATED SCATTERERS.

```
STEP 2: COMPUTE LOCATION OF SPECULAR POINTS THAT ARE ILLUMINATED.
```

```
DO 20 K=1,KTAR
   IF(JHOT(K))
      DO 20 J=1,3
         RK(J)=TARG(I,J)+OFFSET(I)*COSPHI(I,J)
      CONTINUE
20 CONTINUE
```

```
STEP 3: COMPUTE SQUARE ROOT OF RCS FOR ALL ILLUMINATED WIDE AREAS.
```

```
22 SIG(K)=SQRT(ABS(COSPHI(I,NORMAL(I11)))*SIGMA(I))
```

```
STEP 4: CHECK FOR SHORT RANGE CONDITION.
```

```
IF(ICLOSE.EQ.0.OR.RANGE.GT.300.) GO TO 55
```

```
STEP 4-1: DETERMINE RANGE TO RADAR IN TARGET FRAME.
```

```
RANGE=SQRT(RADAR(I11)*2+RADAR(I22)*2+RADAR(I33)*2)
```

```
STEP 4-2: SET MURAGSIS LOOP MONITORING VARIABLE.
```

```
IF(RA0.0.DT.RANGE.LE.0.1.AND.RANGE.LE.270.) ICLOSE=1
```

```
STEP 4-3: CHECK MONITORING VARIABLE TO DETERMINE IF SHORT RANGE CONDITION EXISTS.
```

```
IF(ICLOSE.EQ.0.AND.RANGE.GT.0.) GO TO 55
```

```
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 ICLOLQ.EQ.0 GO TO 35
```

```
DO 30 I=1,NWIDE
   PHIOLD(I)=ACOS(COSPHI(I,NORMAL(I))
   DO 25 J=1,3
      V(I,J)=NORMAL(I11)*RND(UKSEED(I,J)-.5)
      V(I,J)=V(I,J)*SCALE(I,J)
      V(I,J)=V(I,J)+V(I,J)
      CONTINUE
30 CONTINUE
```

```
STEP 5-2: UPDATE ANGULAR INCREMENT FOR EACH DIFFUSE SCATTERER --- CHANGE IN ANGLE FROM SAMPLE-TO-SAMPLE.
```

```
DO 40 I=1,NWIDE
   PHII(NORMAL(I))=ACOS(COSPHI(I,NORMAL(I))
   DPHII(I)=(PHII(NORMAL(I))-PHII(I))
   PHII(I)=PHII(I)+DPHII(I)
   CONTINUE
```

END
STEP 5-4: UPDATE SCATTERER LOCATION FOR ALL ILLUMINATED DIFFUSE SCATTERER — UPDATE DIFFERENCE EQUATIONS.

DO 50 K=1,NWIDE
   I=JHOT(K)  
   DO 45 J=1,3
      IF(J.NE.NORMAL(I)) GOTO 45
      ALPHA(I,J)=EXPRSIN(J)*ABS(DPHI(I,J)*COSPH(I,NORMAL(I)))+
      WSCALE(I,J)*RND(UKSEED(I,J))-5  
      V(I,J)=ALPHA(I,J)*VOLD(I,J)+MRAN(I,J)
      W(I,J)=ALPHA(I,J)*WRAN(I,J)
      END
   CONTINUE
50 CONTINUE

STEP 6: UPDATE PARAMETERS USED TO MONITOR TARGET POSITION.

S.MEAN=ICLOSE  
SCATTERER UPDATE obese  
DIFFUSE IK=1.NWIDE
   Q=NA`L
   IQ=SQRT(F11VOLD(I)**2)*WSCALE(I,J)*RND(UKSEED(I,J))
      _S LK+^J	 IrJ)
   END

STEP 6: UPDATE PARAMETERS USED TO MONITOR TARGET POSITION.

S.MEAN=ICLOSE  
SCATTERER UPDATE obese  
DIFFUSE IK=1.NWIDE
   Q=NA`L
   IQ=SQRT(F11VOLD(I)**2)*WSCALE(I,J)*RND(UKSEED(I,J))
      _S LK+^J	 IrJ)
   END

NOTE: THE FOLLOWING STATEMENTS ARE PRINT STATEMENTS USED IN THE DEBUGGING PROCESS.

NOTE: DEBUGGING PRINT STATEMENTS.

WRITE(6,908) K,TAR,NWIDE,ICLOSE,ROLD
908 FORMAT(/,11,MT,ICL,R=318,F12.4)

PRINT TABULAR LISTING OF ALL DATA ASSOCIATED WITH SPAS SCATTERERS.

WRITE(6,901)(I,SIGN(I),TARG(I,1),TARG(I,2),TARG(I,3),OFFSET(I),
     8 PHIMIN(I,1),
     1 PHIMAX(I,1),PHIMIN(I,2),PHIMAX(I,2),PHIMIN(I,3),PHIMAX(I,3),
     2 I=1,KTAR)

PRINT TOTAL # OF SCATTERERS AND # OF DIFFUSE SCATTERERS.

WRITE(6,902) K,TAR,NWIDE

PRINT INFORMATION ASSOCIATED WITH ILLUMINATED SCATTERERS.

WRITE(6,903) (I,JHOT(I),SIG(I),(R(I,J),J=1,3),
     1 I=1,KTAR)

PRINT DATA ASSOCIATED WITH DIFFUSE SCATTERER DIFFERENCE EQUATION.

WRITE(6,905) PHIMOLD(I),
     1 (V(I,J),J=1,3),R(I,J),J=1,3)
     (W(I,J),J=1,3),PHIMOLD(I),PHIOLD(I)
     WRITE(6,907) I,JHOT(I),VOLD(I,J),J=1,3,ALPH(I,J),J=1,3,
     1 (R(I,J),J=1,3),V(I,J),J=1,3)

PRINT TOTAL # OF TARGETS = K

900 FORMAT(/,10,(5X,10.5,4F10.2))
901 FORMAT(/,10,(5X,10.5,4F10.2))
902 FORMAT(/,10,(5X,10.5,4F10.2))

ALL PRINT FORMAT STATEMENTS.

903 FORMAT(/,9X,’X’,3X,’JHOT(I)’,7X,’RCS’,5X,’PHI-X’,5X,’PHI-Y’,
     1 5X,’PHI’,3F10.3)
904 FORMAT(21F10.3)
905 FORMAT(21F10.3)
906 FORMAT(3F10.3)
907 FORMAT(21F10.3)
RETURN
END