

99162-01

The Engineering Analysis and Design
of the Aircraft Dynamics Model For the
FAA Target Generation Facility

Mark Peters
Michael A. Konyak

Prepared for:

Dan Warburton
ACB-860
NAS Simulation Branch

Federal Aviation Administration
William J. Hughes Technical Center
Atlantic City, NJ 08405

Under:

System Resources Corporation
5218 Atlantic Avenue
Mays Landing, NJ 08330
FAA Prime Contract No. DTFA03-94-C-00042
Subcontract No. 97- 020

November 2003

Acknowledgments

This project was supported under contract No. DTFA03-94-C-0042 (CADSS) with the Federal Aviation Administration William J. Hughes Technical Center, Atlantic City, NJ, under subcontract to System Resources Corporation (SRC). The author acknowledges the insightful project direction provided by Mr. Dan Warburton of the FAA and Mr. George Chachis of SRC, during the course of this work. The author also wishes to acknowledge the various technical contributions and reviews from the members of the Seagull team. Specifically, the author acknowledges Dr. John Sorensen and Tysen Mueller for their contributions to the error modeling sections of the Navigator development, and George Hunter and JC Gillmore for their leadership and guidance throughout the project.

- Mark Peters
September 1999

Change History

This document is a work in progress. This section of the document tracks the revisions to the document in table format.

Rev. #	Date	Description
1	11/5/2001	Added <i>Change History</i> section. Removed selected paragraphs from <i>Section 12</i> that pertained to approach and landing.
2	11/5/2003	<ul style="list-style-type: none">• Updated take-off and landing sequences.• Added control regions outside of speed-altitude plane.• Updated the supporting functional logic of the longitudinal control system.• Removed flight restriction on back side of thrust curve.• Removed compressibility drag.• Removed chapter on terminal flight phases, incorporating these phases into the supporting functional logic of the longitudinal control system.• Abandoned the previous design of the longitudinal guidance system (i.e., chapter 10, formerly called the vertical constraint logic). A new design is pending.• Corrected errors in text, equations, and graphics.

Table of Contents

1. INTRODUCTION.....	1
1.1 BACKGROUND	1
1.2 SCOPE	2
1.3 ORGANIZATION	2
2. THE AIRCRAFT EQUATIONS OF MOTION	5
2.1 THE DEFINITION OF THE BODY FRAME AND THE INERTIAL FRAME.....	5
2.2 DEFINITION OF FLIGHT MECHANICS NOMENCLATURE	8
2.3 THE WIND AND STABILITY REFERENCE FRAMES.....	9
2.4 THE DERIVATION OF THE SIX DEGREE OF FREEDOM EQUATIONS OF MOTION.....	12
2.5 THE MODAL PROPERTIES OF THE SIX DEGREE OF FREEDOM MODEL .	14
2.5.1 <i>The Short Period Mode</i>	15
2.5.2 <i>The Phugoid Mode</i>	16
2.5.3 <i>The Dutch Roll Mode</i>	16
2.5.4 <i>The Roll Mode</i>	17
2.5.5 <i>The Spiral Mode</i>	17
2.6 SIMPLIFYING THE EQUATIONS OF MOTION TO FOUR DEGREES OF FREEDOM	18
2.7 THE ADDITION OF WINDS	22
2.8 ELLIPSOIDAL EARTH TRAJECTORY PROPAGATION EQUATIONS	24
2.8.1 <i>Elliptic Earth Reference Frames</i>	25
2.8.2 <i>The Ellipsoidal Earth Definition</i>	27
2.8.3 <i>Geocentric Latitude and Geodetic Latitude</i>	28
2.8.4 <i>Determining Geocentric Latitude in terms of Geodetic Latitude</i>	29
2.8.5 <i>Kinematics for an Ellipsoidal Earth</i>	31
2.8.6 <i>X, Y, Z Data for DIS Criteria</i>	37
2.9 THE DERIVED STATE VARIABLES	40
2.10 THE AIRFRAME MODEL	41

2.11 THE ENGINE MODEL.....	43
2.12 THE STANDARD DAY ATMOSPHERE MODEL.....	45
2.13 INTEGRATION TECHNIQUES	49
2.13.1 <i>The Second Order Runge-Kutta Method</i>	50
2.13.2 <i>The First Order Euler Method</i>	51
2.13.3 <i>Analytic Integration</i>	52
2.14 THE INTEGRATION OF THE ROLL EQUATION.....	52
2.14.1 <i>The open loop roll rate and roll angle equations</i>	52
2.14.2 <i>The closed loop system</i>	53
2.14.3 <i>Zero Order Hold</i>	54
2.14.4 <i>The State Transition Matrix</i>	54
2.14.5 <i>The Inverse Laplace Transform of the State Transition Matrix</i>	55
2.14.6 <i>The Inverse of the State Transition Matrix</i>	56
2.14.7 <i>The Integration of the Zero Order Hold</i>	56
2.14.8 <i>Initial Preparation for Integration</i>	57
2.14.9 <i>The Two Common Integrations</i>	58
2.14.10 <i>The Complete Integrated Terms</i>	59
2.14.11 <i>Complete Integrated Portion of the Zero Order Hold</i>	59
2.14.12 <i>The Complete Solution</i>	60
3. THE EXAMINATION OF THE LONGITUDINAL DYNAMICS.....	63
3.1 THE LINEAR MODEL OF THE LONGITUDINAL DYNAMICS.....	63
3.2 THE ANALYSIS OF LONGITUDINAL AIRCRAFT MODAL PROPERTIES ..	72
4. THE FEEDBACK CONTROL SYSTEM FOR LONGITUDINAL CONTROL..	79
4.1 THE GENERAL CONTROL LAW	79
4.2 MANIPULATING AN LTD STATE-SPACE WITH INTEGRAL CONTROL....	81
4.3 AN ANALYSIS OF THE EFFECTS OF FEEDBACK CONTROL ON THE MODAL PROPERTIES	83
4.4 FEEDBACK CONTROLLER DESIGN	88
4.4.1 <i>Lift Coefficient Control of Altitude Rate</i>	90
4.4.2 <i>Lift Coefficient Control of Speed</i>	96

4.4.3 Controlling Speed and Altitude Rate Simultaneously	105
5 THE SUPPORTING FUNCTIONAL LOGIC OF THE LONGITUDINAL CONTROL SYSTEM.....	117
5.1 CONTROL STRATEGIES.....	118
5.1.1 Feeding Back Altitude Rate Only.....	118
5.1.2 Feeding Back Speed Only	118
5.1.3 Feeding Back Speed and Altitude Rate	119
5.1.4 Altitude Capture.....	119
5.1.5 Speed Capture	121
5.2 DIVIDING THE FLIGHT ENVELOPE INTO REGIONS.....	123
5.2.1 The Speed-Altitude Plane.....	124
5.2.2 Taking off	133
5.2.3 Landing Region Management	136
5.2.4 Landing Flare (Region 10).....	138
5.2.5 Landing Ground Run (Region 11).....	139
5.3 THROTTLING IN REGIONS 1 THROUGH 6	139
5.4 AIRCRAFT DEVICE DEPLOYMENT	140
5.4.1 Flaps.....	140
5.4.2 Speed Brakes	140
5.4.3 Spoiler	140
5.4.4 Landing Gear	141
5.4.5 Ground Braking.....	141
6. THE SELECTION OF GAINS.....	143
6.1 THE AIRCRAFT'S FLIGHT ENVELOPE.....	143
6.2 DETERMINING ACCEPTABLE MODAL PROPERTIES.....	145
6.3 CHOOSING A SINGLE REFERENCE CONDITION FOR GAIN CALCULATION	146
6.4 EVALUATING SYSTEM PERFORMANCE WITH SCHEDULED GAINS	147
7. THE LATERAL DIRECTIONAL CONTROL LAWS	153
7.1 THE BANK ANGLE CAPTURE ALGORITHM.....	153

7.2 THE HEADING CAPTURE ALGORITHM	155
7.3 USING THE BANK ANGLE CAPTURE AND HEADING CAPTURE ALGORITHMS TO EXECUTE A TURN	156
7.4 DECIDING WHICH WAY TO TURN	157
7.5 CAPTURING A HEADING WHEN THE DIRECTION OF TURN IS SPECIFIED	158
8. THE LATERAL GUIDANCE SYSTEM.....	161
8.1 GROUND TRACK GUIDANCE	161
8.2 FIX CAPTURE GUIDANCE	162
8.3 ROUTE FOLLOWING.....	163
8.3.1 <i>Maintaining Ground Track Along a Segment</i>	163
8.3.2 <i>Transitioning to the Next Segment</i>	166
8.4 CAPTURING A ROUTE	168
8.4.1 <i>Determining a Capture Segment</i>	169
8.4.2 <i>Determining if it is Time to Merge onto the Route</i>	172
8.4.3 <i>Automatic Route Capture</i>	174
8.4.4 <i>Vectored Route Capture</i>	178
8.4.5 <i>Initial Fix Route Capture</i>	180
8.5 BASIC ALGORITHMS REQUIRED FOR COMPLETE FUNCTIONALITY ...	180
8.5.1 <i>Calculating the Aircraft Turn Radius</i>	180
8.5.2 <i>Determining the Aircraft's Lateral Distance from a Segment</i>	181
8.5.3 <i>Determining Distance to go Along a Segment</i>	182
8.5.4 <i>Rhumb Line Bearing</i>	182
8.5.5 <i>Rhumb Line Distance</i>	185
8.5.6 <i>Creating Vectors Representing Segments</i>	185
9. NAVIGATION ERROR MODELING	187
9.1 VOR/DME NAVIGATION.....	187
9.1.1 <i>Determining Aircraft Position from a VOR/DME Station</i>	187
9.1.2 <i>Determining the Proper VOR/DME Station to use for Navigation</i>	193
9.2 GPS NAVIGATION.....	196

9.3 DISCRETIZING THE CONTINUOUS 2ND ORDER GAUSS MARKOV PROCESS	200
9.4 ILS LOCALIZER ERROR MODEL	202
10. THE LONGITUDINAL GUIDANCE SYSTEM	207
11. FLIGHT TECHNICAL ERROR	209
11.1 OPERATIONAL DETAILS	210
<i>11.1.1 Piloted Flight Technical Error.....</i>	<i>210</i>
<i>11.1.2 FMS Flight Technical Error</i>	<i>210</i>
<i>11.1.3 ILS Flight Technical Error.....</i>	<i>210</i>
12. MODEL VERIFICATION AND VALIDATION.....	213
12.1 CONSTANT AIRSPEED CLIMBS AND DESCENTS	214
12.2 MACH/CAS DESCENTS AND CAS/MACH CLIMBS	215
12.3 SPEED CHANGES	217
12.4 SPEED CHANGES DURING CLIMBS AND DESCENTS	220
12.5 AUTOMATIC ROUTE CAPTURE.....	222
12.6 VECTORED ROUTE CAPTURE.....	226
12.7 INITIAL FIX ROUTE CAPTURE.....	228
12.8 SEGMENT TRANSITION.....	229
12.9 FLIGHT TECHNICAL ERROR	230
12.10 NAVIGATION ERRORS.....	234
<i>12.10.1 GPS Navigation Error.....</i>	<i>234</i>
<i>12.10.2 VOR/DME Error</i>	<i>235</i>
12.11 TERMINAL FLIGHT PHASES.....	237
<i>12.11.1 Take-Off.....</i>	<i>237</i>
<i>12.11.2 Landing.....</i>	<i>238</i>
12.12 CONCLUSIONS.....	240
APPENDIX A ANALYSIS OF THE TRANSFER FUNCTIONS OF THE LONGITUDINAL DYNAMICS.....	243
GLOSSARY.....	251
ACRONYMS.....	253

REFERENCES.....255

List of Figures

<i>Figure 2.1. The body fixed reference frame aligned with an aircraft</i>	<i>5</i>
<i>Figure 2.2. The relationship between body and inertial reference frames</i>	<i>6</i>
<i>Figure 2.3. The 3-2-1 Euler sequence of rotations used to quantify the aircraft's orientation.....</i>	<i>6</i>
<i>Figure 2.4. The forces, moments, velocity components and angular rates of an aircraft...8</i>	
<i>Figure 2.5. Illustration of the stability and wind coordinate systems</i>	<i>9</i>
<i>Figure 2.6. Illustration of the Short Period mode causing oscillations about the aircraft's center of gravity.....</i>	<i>16</i>
<i>Figure 2.7. Illustration of the Phugoid mode</i>	<i>16</i>
<i>Figure 2.8. Illustration of the Dutch Roll mode</i>	<i>16</i>
<i>Figure 2.9. Illustration of the Roll mode</i>	<i>17</i>
<i>Figure 2.10. Illustration of an unstable Spiral mode</i>	<i>17</i>
<i>Figure 2.11. The Earth Centered Earth Fixed (ECEF) reference frame and the surface frame</i>	<i>26</i>
<i>Figure 2.12. The longitude and latitude rotations.....</i>	<i>26</i>
<i>Figure 2.13. The ellipsoidal Earth model.....</i>	<i>28</i>
<i>Figure 2.14. Geodetic and geocentric latitudes</i>	<i>29</i>
<i>Figure 2.15. Elliptic Earth terminology</i>	<i>29</i>
<i>Figure 2.16. The position vector from the center of the Earth to the aircraft.....</i>	<i>31</i>
<i>Figure 2.17. Ellipsoidal Earth terminology with the addition of the 'c' frame and the angle ϵ.....</i>	<i>32</i>
<i>Figure 2.18. The rotation between the surface frame and the 'c' frame.....</i>	<i>33</i>
<i>Figure 2.19. The ECEF and DIS Coordinate Frames.....</i>	<i>38</i>
<i>Figure 2.20. Maximum thrust vs altitude for a DC-9/MD80.....</i>	<i>43</i>
<i>Figure 2.21. Fuel Consumption at maximum thrust (both engines).....</i>	<i>45</i>
<i>Figure 2.22. Temperature vs altitude for the standard day atmosphere.....</i>	<i>46</i>
<i>Figure 2.23. The speed of sound variation with altitude for the standard day atmosphere.....</i>	<i>47</i>

Figure 2.24. Pressure variation with altitude for the standard atmosphere	48
Figure 2.25. Density variation with altitude for the standard atmosphere	49
Figure 2.26. The open loop response to an aileron unit step	53
Figure 2.27. Roll mode response to a 10 degree desired bank angle.....	62
Figure 3.28. Comparison of linear and nonlinear models with a 0.01 perturbation from the reference lift coefficient	71
Figure 3.29. Comparison of linear and nonlinear models with a 0.2 perturbation from the reference lift coefficient	72
Figure 4.1. Block diagram for the longitudinal control law	80
Figure 4.2. Effects of proportional feedback to the lift coefficient.....	86
Figure 4.3. Effects of integral feedback to the lift coefficient	87
Figure 4.4 Effects of Proportional Feedback to the Thrust.....	88
Figure 4.5 Effects of Integral Feedback to the Thrust.....	89
Figure 4.6. System response to a 0.1 step in lift coefficient	91
Figure 4.7. Root Loci of $k_{p_{14}}$ (left) and $k_{i_{14}}$ (right) successive loop closures.....	92
Figure 4.8. The effects of an increased proportional gain $k_{p_{14}}$	92
Figure 4.9. System response to a 1000 ft/min rate commanded rate of climb.....	93
Figure 4.10. $k_{b_{14}}$ root locus for the Mach Capture controller	98
Figure 4.11. $k_{p_{12}}$ root locus for the Mach Capture controller.....	98
Figure 4.12. Simulated Mach Capture	99
Figure 4.13. $k_{i_{12}}$ (left) and $k_{b_{14}}$ (right) root loci for Mach capture	99
Figure 4.14. Simulation of the Completed Mach Capture.....	100
Figure 4.15. Simulation of Mach capture with slower dynamics	106
Figure 4.16. The initial $k_{p_{14}}$ (left plot) and $k_{i_{14}}$ (right plot) loop closures for region 7	107
Figure 4.17. The final increase in $k_{p_{14}}$ to achieve adequate damping.....	107
Figure 4.18. Simulation of a commanded 1000ft/min rate of climb without any feedback to thrust.....	108

Figure 4.19. Root loci for k_{p22} (left plot) and k_{i22} (right plot)	109
Figure 4.20. Simulation of a commanded 1000ft/min climb rate using the final controller for region 7	110
Figure 5.1: Illustration of an aircraft capturing an altitude.....	120
Figure 5.2: Altitude Rate During Region Transition.....	121
Figure 5.3. Flow diagram for calculating K_h	122
Figure 5.4. The Speed-Altitude Plane.....	123
Figure 5.5. The constant energy line on the speed-altitude plane.....	125
Figure 5.6. Illustration of the approximation for the constant energy line using the diagonal cut across the steady, level flight region (region 7) of the speed-altitude plane.....	126
Figure 5.7. The speed-altitude plane in terms of indicated airspeed	128
Figure 5.8: Desired Landing Speed Profile.....	138
Figure 5.9: Flap Deployment Algorithm	140
Figure 6.1. An example flight envelope	143
Figure 6.2. The flight envelope for a DC-9 in terms of CL and True Airspeed.....	144
Figure 6.3. The locus of Phugoid poles for the entire flight envelope of a DC-9 in the clean configuration	145
Figure 6.4. The locus of closed loop phugoid poles in for altitude-rate-only feedback for the entire flight envelope of a DC-9 in the clean configuration	149
Figure 6.5. The locus of closed loop Phugoid poles for speed-only feedback for the entire flight envelope of a DC-9 in the clean configuration	149
Figure 6.6. The locus of closed loop Phugoid poles in the steady, level flight region for the entire flight envelope of a DC-9 in the clean configuration	150
Figure 6.7. The locus of closed loop Phugoid poles in the steady, level flight region for the entire flight envelope of a DC-9 with full flaps deployed	151
Figure 7.1. Block diagram for the rolling dynamics	154
Figure 7.2. Block diagram for heading feedback	156
Figure 7.3. Simulation of Aircraft Executing a Turn.....	157

<i>Figure 7.4. An illustration of the dilemma of whether to make a right or left turn to a heading.....</i>	<i>158</i>
<i>Figure 7.5. Algorithm for capturing a heading</i>	<i>159</i>
<i>Figure 8.1. Illustration of the difference between ground track and heading.....</i>	<i>161</i>
<i>Figure 8.2. An aircraft turning to a fix.....</i>	<i>163</i>
<i>Figure 8.3. Fix capture guidance algorithms</i>	<i>164</i>
<i>Figure 8.4. Illustration of the aircraft in route following mode.....</i>	<i>164</i>
<i>Figure 8.5. Logic for insuring desired ground track is within proper boundaries</i>	<i>166</i>
<i>Figure 8.6. Illustration of segment transition distance</i>	<i>167</i>
<i>Figure 8.7. The geometry of segments which intersect at obtuse angles</i>	<i>167</i>
<i>Figure 8.8. Geometric representation of segments adjoined at an acute angle.....</i>	<i>168</i>
<i>Figure 8.9. Scenario of an aircraft determining which segment to capture.....</i>	<i>170</i>
<i>Figure 8.10. A scenario demonstrating the failure of criterion #1</i>	<i>171</i>
<i>Figure 8.11. Regions where both criterion #1 and criterion #2 fail</i>	<i>171</i>
<i>Figure 8.12. Flow chart detailing segment determination logic</i>	<i>172</i>
<i>Figure 8.13. Illustration of geometry associated with an aircraft merging onto a segment when aircraft is heading in the direction of the segment.....</i>	<i>173</i>
<i>Figure 8.14. An aircraft merging onto a segment which is pointed in a direction opposite of the aircraft's current velocity.....</i>	<i>174</i>
<i>Figure 8.15. Logic for Automatic Route Capture Guidance</i>	<i>175</i>
<i>Figure 8.16. Illustration of aircraft using a 45 degree intercept</i>	<i>176</i>
<i>Figure 8.17. Illustration of two aircraft capturing a segment using a dynamic fix</i>	<i>176</i>
<i>Figure 8.18. Determining an offset fix (dynamic fix) location</i>	<i>177</i>
<i>Figure 8.19. Route capture with fixed heading guidance.....</i>	<i>179</i>
<i>Figure 8.20. Logic for determining whether or not a heading will intercept a segment.....</i>	<i>179</i>
<i>Figure 8.21. Illustration of distance calculation geometry</i>	<i>181</i>
<i>Figure 8.22. Cylindrical mapping of spherical Earth model. Shown are two fixes and the constant heading route between the fixes.</i>	<i>182</i>
<i>Figure 8.23 Geometry relating latitude, longitude, and bearing on a spherical earth..</i>	<i>183</i>

<i>Figure 9.1. An illustration of the range and bearing from the station</i>	<i>188</i>
<i>Figure 9.2. Illustration of the quadrants of the compass rose with respect to a VOR/DME station</i>	<i>190</i>
<i>Figure 9.3. Logic for determining if the current VOR/DME used for navigation should be changed</i>	<i>192</i>
<i>Figure 9.4. Logic for determining the appropriate VOR/DME for the next segment</i>	<i>194</i>
<i>Figure 9.5. An illustration of the geometry used to determine which VOR/DME should be used for segments without a VOR/DME</i>	<i>195</i>
<i>Figure 9.6. Logic For determining which VOR/DME to use when no VOR/DME lies along route.....</i>	<i>196</i>
<i>Figure 9.7 GPS Receiver Measurement Geometry.....</i>	<i>197</i>
<i>Figure 9.8. Monte Carlo Simulated SA Position Errors.....</i>	<i>199</i>
<i>Figure 9.9. Monte Carlo Simulated SA Velocity Errors.....</i>	<i>200</i>
<i>Figure 9.10. Measured ILS Localizer Bearing Deviation Angle Errors</i>	<i>204</i>
<i>Figure 9.11. Simulated ILS Localizer Bearing Deviation Angle Errors</i>	<i>205</i>
<i>Figure 12.1. Simulation Window for TGF-test.....</i>	<i>213</i>
<i>Figure 12.2. Comparison of Pseudocontrol (black) and TGF-test (gray) in a constant indicated airspeed climb and 280kt.....</i>	<i>215</i>
<i>Figure 12.3. A comparison of Pseudocontrol (black) and TGF-test (gray) in a descent at a constant indicated airspeed of 300 kts.....</i>	<i>216</i>
<i>Figure 12.4. Comparison of Pseudocontrol (black) and TGF-test (gray) performing a Mach/CAS descent from 30,000 ft to 10,000ft using an MD80 at 130,000lb.....</i>	<i>217</i>
<i>Figure 12.5. A deceleration of an MD80 from 350kts to 250kts while at 10,000ft using Pseudocontrol (black) and TGF-test (gray) simulation tools.....</i>	<i>218</i>
<i>Figure 12.6. An MD80 accelerating from 250kts to 350kts while maintaining 10,000ft using Pseudocontrol (black) and TGF-test (gray) simulation tools.....</i>	<i>219</i>
<i>Figure 12.7. An MD80 decelerating from Mach 0.8 to Mach 0.6 while maintaining 30,000ft using Pseudocontrol (black) and TGF-test (gray) simulation tools.....</i>	<i>220</i>
<i>Figure 12.8. An MD80 accelerating from Mach 0.6 to Mach 0.8 while maintaining 25,000ft using Pseudocontrol (black) and TGF-test (gray) simulation tools.....</i>	<i>221</i>
<i>Figure 12.9. An MD80 in a climb with various speed changes using the TGF-test simulation.....</i>	<i>222</i>

<i>Figure 12.10. Automatic route capture with the aircraft close to the route but headed in the wrong direction</i>	223
<i>Figure 12.11. Automatic route capture with an aircraft far from the capture segment.</i>	224
<i>Figure 12.12. Automatic route capture with the aircraft headed perpendicular to route</i>	225
<i>Figure 12.13. Automatic route capture with aircraft in an ambiguous region between segments</i>	226
<i>Figure 12.14. Vectored route capture from an ambiguous position</i>	227
<i>Figure 12.15. Vectored route capture when the vectored heading tends to be in the opposite direction of the route</i>	228
<i>Figure 12.16. The initial fix capture algorithm being used to vector an aircraft back to the beginning of the route</i>	229
<i>Figure 12.17. An aircraft capturing a route from behind using the initial fix capture algorithm</i>	230
<i>Figure 12.18. Piloted flight technical error of an MD80 traveling at 250kts and 5000ft</i>	232
<i>Figure 12.19. Piloted flight technical error of an MD80 traveling at 300kts and 30,000ft</i>	232
<i>Figure 12.20. FMS flight technical error of an MD80 traveling at 250kts and 5000ft.</i>	233
<i>Figure 12.21. FMS flight technical error of an MD80 traveling at 300kts and 30,000ft</i>	233
<i>Figure 12.22. An aircraft trajectory using GPS navigation</i>	234
<i>Figure 12.23. An aircraft flying a segment using VOR/DME navigation where both endpoints are VOR/DME stations</i>	235
<i>Figure 12.24. An aircraft flying a route comprised of 2 VOR/DME stations with 2 intersections between the VOR/DME stations</i>	236
<i>Figure 12.25. An MD80 at 130,000lbs taking off with a rotation speed of 150KIAS</i>	238
<i>Figure 12.26. Longitudinal view of an MD80 on final approach and landing</i>	239
<i>Figure 12.27. A top view of an MD80 on final approach to landing</i>	240
<i>Figure A.1: The root loci of proportional and integral lift coefficient control of altitude rate considering the thrust control of speed</i>	249

List of Tables

<i>Table 2.1. Definition of flight mechanics nomenclature.....</i>	<i>8</i>
<i>Table 2.2. Summary of kinematic and dynamic equations of motion</i>	<i>15</i>
<i>Table 2.3. The equations of motion for the 4-DOF model.....</i>	<i>22</i>
<i>Table 2.4. The equations of motion including wind gradients.....</i>	<i>24</i>
<i>Table 2.5. The Latitude and Longitude Trajectory Propagation Equations</i>	<i>37</i>
<i>Table 3.6. Stability and control derivatives of f_{V_a}</i>	<i>68</i>
<i>Table 3.7. Stability and control derivatives of f_{γ_a}</i>	<i>68</i>
<i>Table 3.8. Stability and control derivatives of f_h</i>	<i>68</i>
<i>Table 4.9. The gain scheduling equations for lift coefficient control of speed.....</i>	<i>105</i>
<i>Table 5.1: Speed - Altitude Plane Booleans</i>	<i>130</i>
<i>Table 6.1 Marker key to Figure 6.4 - Figure 6.7.....</i>	<i>148</i>
<i>Table 9.1. Observed Local Coordinate Position Root-Mean-Square (rms) Errors</i>	<i>197</i>
<i>Table 9.2 Simplified vs. Observed SA Position and Velocity Model Parameters.....</i>	<i>198</i>
<i>Table 0.1 Error Statistics Summary.....</i>	<i>231</i>

List of Symbols

The purpose of this reference list is to define the symbols used throughout this document. In many cases they appear without definition, so the reader can refer to this list for clarification. Some symbols are used in more than one context, but it is assumed that its relevant definition is clear from the context. Any deviations from this list are defined within the context of the relevant section.

Superscripts and Subscripts

$[\hat{x}]$	<i>a unit vector along the axis defined by $[x]$</i>
$[\dot{x}]$	<i>derivative of $[x]$ with respect to time</i>
$[\ddot{x}]$	<i>second derivative of $[x]$ with respect to time</i>
$[x]_a$	<i>relative to the air mass</i>
$[x]_{amb}$	<i>ambient</i>
$[x]_{AP}$	<i>approach flight condition</i>
$[x]_b$	<i>in the body axis coordinate system</i>
$[x]_b$	<i>proportional portion of input signal in the feedback path</i>
$[x]_c$	<i>commanded value</i>
$[x]_c$	<i>in the geocentric surface frame coordinate system</i>
$[x]_{cl}$	<i>closed loop value of $[x]$</i>
$[x]_{CR}$	<i>cruise flight condition</i>
$[x]_d$	<i>desired value</i>
$[x]_{ecf}$	<i>in the earth-centered, earth-fixed coordinate system</i>
$[x]_G$	<i>ground</i>
$[x]_{GS}$	<i>glide slope</i>
$[x]_{GT}$	<i>ground track</i>
$[x]_i$	<i>in the inertial, flat-earth coordinate system</i>
$[x]_i$	<i>integral portion of input signal</i>
$[x]_{IC}$	<i>initial climb flight condition</i>
$[x]_{LD}$	<i>landing flight condition</i>
$[x]_{max}$	<i>maximum</i>
$[x]_{min}$	<i>minimum</i>
$[x]_p$	<i>proportional portion of input signal in the feed-forward path</i>
$[x]_s$	<i>in the surface frame coordinate system (geodetic)</i>

$[x]_{sl}$	<i>at sea level</i>
$[x]_{st}$	<i>in the stability axis coordinate system</i>
$[x]_t$	<i>total</i>
$[x]_{TO}$	<i>take-off flight condition</i>
$[x]_w$	<i>in the wind axis coordinate system</i>
$[x]_x$	<i>the x-axis component of $[x]$</i>
$[x]_y$	<i>the y-axis component of $[x]$</i>
$[x]_z$	<i>the z-axis component of $[x]$</i>

Primary Symbols

Variables are in *italic* type-face, vectors and matrices are in **bold** type-face. The only exception is unit vectors, denoted with a '^' in *italic* type-face.

a	<i>Acceleration</i> <i>Semi-minor axis of the Earth</i> <i>Temperature lapse rate in the standard atmosphere</i> <i>Angle between intersecting segments</i>
a^*	<i>Speed of sound in the atmosphere</i>
A	<i>State matrix of a state space representation</i>
AR	<i>Aspect Ratio, b^2/S_w</i>
b	<i>Wing span</i> <i>Semi-major axis of the Earth</i>
B	<i>Input matrix of a state space representation</i>
C	<i>Output matrix of a state space representation</i>
C_L	<i>Lift coefficient</i>
C_{L_0}	<i>Zero-lift lift coefficient</i>
C_{L_R}	<i>Rotation lift coefficient</i>
C_{L_α}	<i>Lift curve slope, $\frac{\partial C_L}{\partial \alpha}$</i>
$C_{L_{\delta_e}}$	<i>Variation of lift coefficient with elevator deflection, $\frac{\partial C_L}{\partial \delta_e}$</i>
C_{D_0}	<i>Zero lift drag coefficient</i>
C_D	<i>Drag coefficient</i>

C_p	<i>Specific heat at constant pressure</i>
C_V	<i>Specific heat at constant pressure</i>
$C_{f_1} - C_{f_2}$	<i>BADA coefficients for determining fuel burn</i>
$C_{T_{e,1}} - C_{T_{e,3}}$	<i>BADA coefficients for determining maximum thrust</i>
C_θ	<i>Cosine of θ</i>
D	<i>Drag</i>
D	<i>Carry through term matrix of a state space representation</i>
e	<i>error vector, $\mathbf{y}_d - \mathbf{y}$</i>
e	<i>specific energy</i>
$e_1 - e_5$	<i>Error between desired & actual values. In order, they are IAS, Mach, Altitude, Altitude Rate, and Heading</i>
$e_{\text{left_turn}}$	<i>Left turn error between desired heading and actual heading</i>
$e_{\text{right_turn}}$	<i>Right turn error between desired heading and actual heading</i>
e^{At}	<i>State transition matrix</i>
E	<i>energy</i>
f	<i>Fuel burn rate</i> <i>Flattening parameter for the Earth</i>
g	<i>Gravitational acceleration</i>
h	<i>Altitude</i>
\dot{h}	<i>Altitude rate</i>
h_{error}	<i>The altitude error used to define the speed altitude plane</i>
H	<i>Angular momentum vector</i>
I_{C_L}	<i>State variable representing the integrated portion of the lift coefficient</i>
I_T	<i>State variable representing the integrated portion of the throttle</i>
\mathbf{I}_u	<i>The integrated control vector</i>
I_x, I_y, I_z	<i>Moments of inertia about the x-, y-, and z-axes</i>
I_{yz}, I_{xz}, I_{xy}	<i>Products of inertia about the x-, y-, and z-axes</i>
IAS_{error}	<i>The indicated airspeed error used to define the speed altitude plane</i>
J	<i>Jacobian for solving systems of equations</i>
k_{lag}	<i>A term used to characterize engine spooling lags</i>
k_ψ	<i>Feedback gain for heading capture</i>
K	<i>Induced drag factor $\frac{1}{\pi e AR}$</i>

<i>K.E.</i>	<i>Kinetic energy</i>
K_h	<i>A term used to smooth transitions into Region 7 from climbs and descents</i>
k_p	<i>Gain feeding roll rate back to ailerons</i>
k_ϕ	<i>Gain feeding roll angle back to ailerons</i>
l	<i>Longitude</i>
L	<i>Lift</i>
\bar{L}	<i>Rolling moment</i>
L_p	<i>Rolling moment variation with roll rate, $\frac{\partial L}{\partial p}$</i>
L_{δ_a}	<i>Rolling moment variation with elevator deflection, $\frac{\partial L}{\partial \delta_a}$</i>
M	<i>Mach number</i> <i>Pitching moment</i>
M_{error}	<i>The Mach error used to define the Mach speed altitude plane</i>
M_0	<i>Zero angle of attack pitching moment</i>
M_α	<i>Pitching moment variation with angle of attack, $\frac{\partial M}{\partial \alpha}$</i>
M_{δ_e}	<i>Pitching moment variation with elevator deflection</i>
m	<i>Aircraft mass</i>
m_{IAS}	<i>Slope of the constant energy line for IAS based flight</i>
m_{Mach}	<i>Slope of the constant energy line for Mach based flight</i>
N	<i>Yawing moment</i>
n	<i>Load factor</i>
n_B	<i>Scaled Gaussian white noise</i>
\hat{n}_s	<i>A unit vector normal to a segment</i>
p	<i>Roll rate</i> <i>Pressure</i> <i>transfer function poles</i>
<i>P.E.</i>	<i>Potential energy</i>
q	<i>Pitch rate</i> <i>Dynamic pressure</i>
r	<i>Yaw rate</i> <i>radius</i> <i>range distance</i>

r_e	<i>Radius of the Earth</i>
r_l	<i>The distance a point on the Earth's surface is from the polar axis</i>
\mathbf{r}_{lf}	<i>A vector from the aircraft's location to the leading fix of a segment</i>
r_t	<i>Aircraft radius of turn</i>
R	<i>Ideal gas constant</i>
\mathbf{R}_s	<i>A vector describing a segment along a route</i>
\mathbf{R}_e	<i>A vector from the center of the Earth to a point on the surface</i>
\mathbf{R}	<i>A position vector in earth-centered, earth-fixed coordinates</i>
s	<i>Laplace domain time analog</i>
S_w	<i>Aircraft reference or wing area</i>
SP_{error}	<i>Generic speed error boundary used to define speed altitude plane</i>
S_θ	<i>Sine of θ</i>
t	<i>Time</i>
T	<i>Thrust</i> <i>Temperature</i>
μ	<i>Velocity of aircraft in the x-body direction</i>
\mathbf{u}	<i>control vector</i>
u_{FTE}	<i>Zero mean unity variance Gaussian white noise</i>
\mathbf{V}	<i>Body frame velocity vector</i>
V_r	<i>Rotation speed</i>
V_2	<i>Climb out speed</i>
V_a	<i>True airspeed</i>
V_{stall}	<i>Stall speed</i>
V_{IAS}	<i>Indicated airspeed</i>
V_w	<i>Wind velocity</i>
v	<i>Body frame side velocity in the y-body direction</i>
w	<i>Body frame velocity in the z-body direction</i> <i>Gaussian white noise</i>
W	<i>Aircraft weight</i>
W_{igf}	<i>Wind in-track gradient factor</i>
W_{cgf}	<i>Wind cross-track gradient factor</i>
X	<i>X-body force</i>
X_{DIS}	<i>X DIS position</i>

\mathbf{y}	<i>Output vector</i>
Y	<i>Y force in the y-body direction</i>
Y_{DIS}	<i>Aircraft position along the y-DIS axis</i>
y_r	<i>Easterly distance of a VOR in miles from a reference point</i>
Z	<i>Z force in the z-body direction</i>
Z_{DIS}	<i>The position of the aircraft along the z-DIS axis</i>
z	<i>Transfer function zeros</i>
α	<i>Angle of attack</i>
β	<i>Sideslip angle</i>
	<i>Damping factor for Gauss-Markov process</i>
$\beta_B(s)$	<i>Spatial damping factor for localizer error modeling</i>
δ	<i>Distance from a radial</i>
δ_a	<i>Aileron deflection</i>
δ_e	<i>Elevator deflection</i>
δ_r	<i>The lateral position error (navigation)</i>
δv	<i>The rate of change of the lateral position error, δr</i>
δ_r	<i>Lateral offset from flight technical error</i>
$\delta\psi$	<i>Heading bias; due, for example to wind or FTE</i>
Δt	<i>Time step</i>
ε	<i>Difference between geodetic and geocentric latitude</i>
γ	<i>Ratio of specific heats</i>
γ_a	<i>Flight path angle relative to the air mass</i>
λ	<i>Geocentric latitude</i>
μ	<i>Geodetic latitude</i>
η	<i>Thrust specific fuel consumption</i>
ρ	<i>Air density</i>
	<i>Range to a NAV-AID (used in navigation error modeling)</i>
$\Delta\rho$	<i>The total range error (used in navigation error modeling)</i>
$\delta\rho_{VD_A}$	<i>Airborne receiver range bias</i>
$\delta\rho_{VD_G}$	<i>Ground based range bias</i>
σ	<i>Standard deviation</i>
ψ	<i>Heading, bearing, or azimuth angle</i>

θ	<i>Pitch angle</i>
ϕ	<i>Roll angle</i>
ω_n	<i>Natural frequency</i>
ω_p	<i>Natural frequency of the phugoid mode</i>
ω	<i>Angular velocity of a rotating frame with respect to a fixed frame</i>
ζ	<i>Damping ratio</i>
ζ_p	<i>Damping ratio of the Phugoid mode</i>
τ	<i>Time constant associated with landing altitude tracking</i>

THIS PAGE INTENTIONALLY LEFT BLANK

1. Introduction

This document presents the analytical development of the aircraft dynamics model to support Seagull Technology's contract No. DTFA03-94-C-00042 (CADSS) with the FAA for the enhancement of the functional capabilities of the Target Generation Facility (TGF). Specifically, this document discusses the detailed engineering design and software implementation of an Aircraft Dynamics Model (ADM) suitable for incorporation into the FAA Target Generation Facility simulations at the FAA William J. Hughes Technical Center, Atlantic City, NJ. The model is designed to be implemented on computers located within the facility, and to work in conjunction with software models of radar, data links, and other Air Traffic Management (ATM) equipment to provide real-time simulation of aircraft operating within the National Airspace System (NAS). This introductory section provides a brief background into the project as well as discusses the scope and organization of the document.

1.1 Background

The FAA Technical Center conducts research and development to investigate emerging Air Traffic Control (ATC & ATM) technologies, associated applications, and ATC processes and procedures. Inherent in these efforts is the requirement to emulate real-world operational conditions in laboratory environments. This requirement extends across the operational domains (e.g., Terminal, En Route, and Oceanic). Much of this work requires the establishment of operational test beds encompassing current operational as well as emerging prototype ATC systems. These test beds are frequently used to conduct studies that simulate the operational conditions found or desired in the associated domain. In the majority of cases, it is necessary to provide realistic representation of air traffic scenarios to evaluate the system, process, or procedure being evaluated. The Technical Center's TGF provides this capability by producing simulated primary and secondary radar targets to the system under test. To the greatest extent possible, TGF produced targets must accurately reflect the flight dynamics of the aircraft which they represent.

Currently, the aircraft dynamics incorporated in the TGF are based on a set of simple aircraft models which base aircraft performance on empirical estimators rather than dynamics models based on the first principles of physics and aeronautics. The current models do not provide the performance characteristics needed to support certain high fidelity simulations. The TGF does not incorporate fuel burn models or the capabilities necessary to introduce environmental (weather) effects. Furthermore, these models do not include many of the aircraft in the current United States inventory.

Many of these missing functions limit the current TGF capabilities. As other simulation models are developed or brought to the Technical Center, higher fidelity will be required to identify NAS operational safety and performance issues. The TGF is prepared to increase its fidelity and operational connectivity required to meet the demands by the other FAA programs and simulators. The goal of this project has been to develop aircraft

models based on aircraft dynamics models and engineering first principles to enhance the TGF modeling capabilities.

1.2 Scope

This document provides a defensible, theoretical foundation for the engineering theory, principles and algorithmic design of the Aircraft Dynamics Model. The engineering analysis starts with the first principles of aircraft flight mechanics and derives a 6-degree-of-freedom model. Next, simplifying assumptions are discussed and the model is reduced to 4 degrees of freedom. The propagation of the aircraft on the surface of the Earth is discussed along with all the necessary reference frames to support all current conventions and interfaces. Modeling the effects of wind is discussed. Next, the flight control system necessary to fly the aircraft through the fundamental maneuvers of climb, descent, turning to a heading, and speed changes is discussed. The control theory necessary to implement the flight control system is discussed in detail. Next, the aircraft guidance system is discussed. The guidance system enables the aircraft to capture and follow routes. Different types of route capture methods are discussed such as the automatic and vectored route capture methods. Navigation systems and their error models are discussed next. This discussion includes the modeling of VOR/DME, ILS, and GPS navigation. With the completion of guidance and navigation, the algorithms that meet speed and altitude constraints at fixes are discussed. Pilot modeling and pilot flight technical error are also included. Finally, the document concludes with a section on verification and validation, the process by which the various features of the simulation are tested and verified.

1.3 Organization

Since much of the analysis is of a highly technical nature, an effort has been made to organize the document so that specific topics are easy to access. This is done to avoid the need to read the entire document to find a specific point. The document is organized into 13 sections, each of which is now discussed briefly.

- Section 2 provides a detailed analysis of the aircraft equations of motion. The 4 degree-of-freedom aircraft model is derived from first principles. All trajectory propagation material is also covered. The numerical integration techniques are also discussed.
- Section 3 develops a linear model of the longitudinal dynamics and analyzes the longitudinal modal properties of the system. Transfer function analysis of the longitudinal dynamics is also performed
- Section 4 provides a detailed analysis of the non-feedback control aspects of the longitudinal control system. There is non-feedback control related logic that determines which feedback control algorithm should be used at a given time. Non-feedback control related logic also manages how the inputs are given to the feedback control algorithms.

- Section 5 deals with the design of the feedback control algorithms which are the algorithms which stabilize the aircraft and drive it to the desired state. There are different feedback control algorithms for different flight phases; each of these control systems is discussed, and a strategy for calculating the required gains is developed.
- Section 6 documents the decision process which led to the final conclusion that gain scheduling would not be necessary. By carefully choosing the reference flight condition, it is possible to choose one set of gains which will work for the aircraft's entire flight envelope.
- Section 7 discusses the lateral directional control system
- Section 8 discusses the lateral guidance system. The purpose of the lateral guidance system is to steer the aircraft to follow routes or other simulation pilot commands.
- Section 9 discusses the navigation systems and navigation error modeling. The purpose of navigation error modeling is to model the variances which occur in aircraft flight paths as a result of imperfect navigation information.
- Section 10 discusses the logic used to make aircraft meet speed and altitude constraints at fixes.
- Section 11 documents the flight technical error algorithms which model the inability of the pilot or autopilot to steer the aircraft perfectly along the desired course.
- Section 12 documents the terminal flight phases which are take-off and landing.
- Section 13 describes verification and validation of the aircraft simulation. The section starts with testing of climbs, descents, and speed changes and continues with testing of the guidance algorithms. The navigation error and flight technical error are also tested. Finally, the terminal flight phases are tested.

THIS PAGE INTENTIONALLY LEFT BLANK

2. The Aircraft Equations of Motion

The purpose of this section is to provide a theoretical foundation for the aircraft equations of motion that are used in the TGF aircraft simulation. The foundation will start with the definition of reference frames. Once the reference frames have been defined, we will derive the equations of motion for the full six-degree-of-freedom (DOF) model. Then, we will apply several assumptions to the equations of motion to reduce the 6-DOF model to a 4-DOF model.

2.1 The Definition of the Body Frame and the Inertial Frame

As discussed in Nelson [N89], the two major reference frames used in the derivation of the aircraft equations of motion are the aircraft body fixed reference frame (denoted with a b subscript) and the inertial reference frame (denoted with an i subscript). The aircraft body frame's origin is fixed at the aircraft's center of gravity. The body frame has its \hat{x}_b -axis aligned with the nose of the aircraft so that the aircraft's nose points in the positive \hat{x}_b direction. The positive \hat{y}_b direction points out along the aircraft's starboard wing. The \hat{z}_b axis points down to complete a right handed coordinate frame. Figure 2.1 shows the body fixed reference frame.

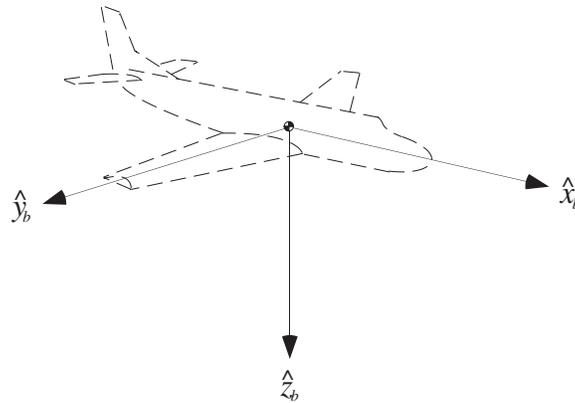


Figure 2.1. The body fixed reference frame aligned with an aircraft

The inertial reference frame is fixed on a point on the Earth's surface and is aligned so that the positive \hat{x}_i axis points in the true North direction and the positive \hat{y}_i axis points in the true East Direction. The \hat{z}_i axis points down and is normal to the surface of the Earth. The body reference frame can assume any orientation with respect to the inertial frame. Figure 2.2 illustrates the relationship between the body and inertial reference frames.

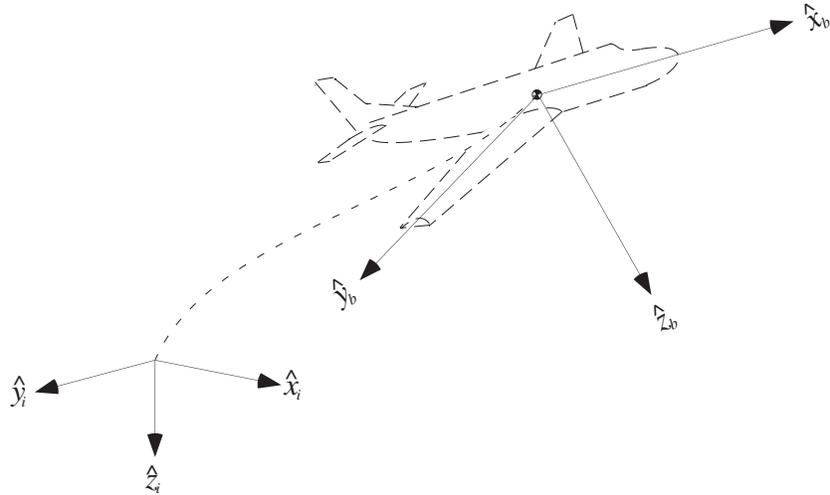


Figure 2.2. The relationship between body and inertial reference frames

The orientation of the body frame with respect to the inertial frame is usually described by an Euler angle (3-2-1) sequence of rotations. The ordering of the rotations is critical to the orientation of the body frame. It is difficult to visualize the actual sequence of rotations in a single drawing, so the sequence is illustrated with 3 separate drawings. Figure 2.3 shows the Euler sequence of rotations which is used to quantify the aircraft's orientation.

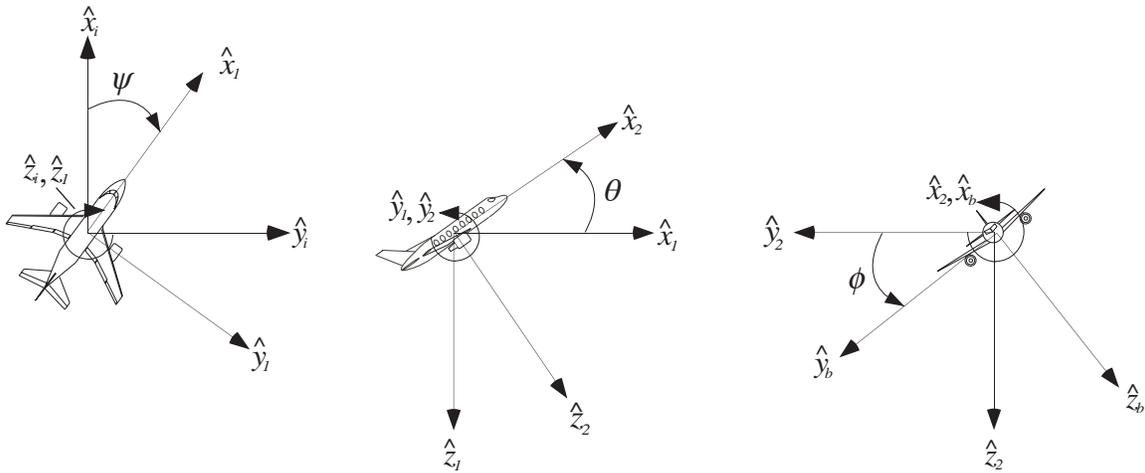


Figure 2.3. The 3-2-1 Euler sequence of rotations used to quantify the aircraft's orientation

The first rotation is through the angle ψ about the \hat{z}_i axis to an intermediate reference frame which is arbitrarily denoted with a '1' subscript. The second rotation is through the angle θ about the \hat{y}_1 axis to another intermediate reference frame which is denoted with a '2' subscript. The final rotation is through the angle ϕ about the \hat{x}_2 axis to the body frame. The angles ψ , θ , and ϕ are referred to as the heading, pitch, and roll angles, respectively.

The conversion between the inertial frame and the body frame of the aircraft is accomplished using direction cosine matrices. The first cosine matrix establishes the relationship between the inertial frame and the first intermediate reference frame. Equations (2.1), (2.2), and (2.3) quantify the relationship between the individual rotations. The nomenclature C_θ and S_θ are simplified notation for $\cos \theta$ and $\sin \theta$. This is done for all trigonometric manipulations to simplify the ultimate expression. From the rotation sequence shown in Figure 2.3, one can write the following direction cosine matrices shown in Equations (2.1) through (2.3).

$$\begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{z}_1 \end{bmatrix} = \begin{bmatrix} C_\psi & S_\psi & 0 \\ -S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{bmatrix} \quad (2.1)$$

$$\begin{bmatrix} \hat{x}_2 \\ \hat{y}_2 \\ \hat{z}_2 \end{bmatrix} = \begin{bmatrix} C_\theta & 0 & -S_\theta \\ 0 & 1 & 0 \\ S_\theta & 0 & C_\theta \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{z}_1 \end{bmatrix} \quad (2.2)$$

$$\begin{bmatrix} \hat{x}_b \\ \hat{y}_b \\ \hat{z}_b \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix} \begin{bmatrix} \hat{x}_2 \\ \hat{y}_2 \\ \hat{z}_2 \end{bmatrix} \quad (2.3)$$

The product of these three direction cosine matrices results in the complete conversion between the inertial frame and the body frame as shown in Equation (2.5).

$$\begin{bmatrix} \hat{x}_b \\ \hat{y}_b \\ \hat{z}_b \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix} \begin{bmatrix} C_\theta & 0 & -S_\theta \\ 0 & 1 & 0 \\ S_\theta & 0 & C_\theta \end{bmatrix} \begin{bmatrix} C_\psi & S_\psi & 0 \\ -S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{bmatrix} \quad (2.4)$$

$$\begin{bmatrix} \hat{x}_b \\ \hat{y}_b \\ \hat{z}_b \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\ -C_\phi S_\psi + S_\phi S_\theta C_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & S_\phi C_\theta \\ S_\phi S_\psi + C_\phi S_\theta C_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{bmatrix} \quad (2.5)$$

The inverse of Equation (2.5) is shown in Equation (2.6).

$$\begin{bmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & -C_\phi S_\psi + S_\phi S_\theta C_\psi & S_\phi S_\psi + C_\phi S_\theta C_\psi \\ C_\theta S_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} \hat{x}_b \\ \hat{y}_b \\ \hat{z}_b \end{bmatrix} \quad (2.6)$$

2.2 Definition of Flight Mechanics Nomenclature

Next, certain flight mechanics nomenclature must be defined. This nomenclature consists of the various linear and angular velocities associated with the motion of the aircraft as well as the forces and moments which are applied. Figure 2.4 provides an illustration of the nomenclature as it applies to the aircraft.

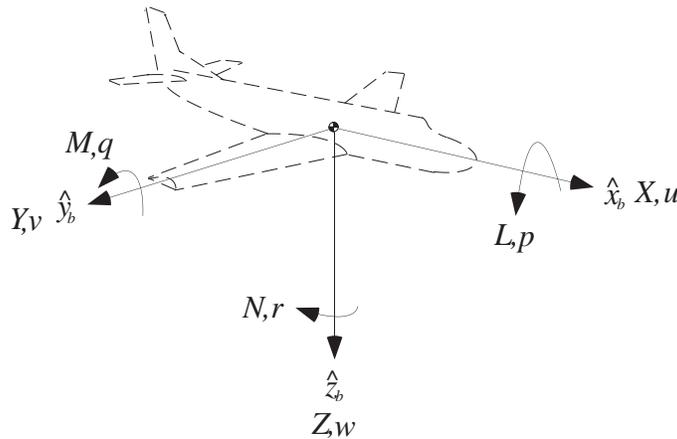


Figure 2.4. The forces, moments, velocity components and angular rates of an aircraft

Table 2.1 summarizes the nomenclature definition so that the mathematical symbols can be associated with the proper terminology. The reader will note that the term L is used to notate the rolling moment. Further along in the text, L will also be used for lift. This is an unfortunate consequence of the merging of two engineering disciplines, dynamics and control and aerodynamics. To avoid confusion, this document will notate the rolling moment using \bar{L} instead of L which is reserved for lift.

Table 2.1. Definition of flight mechanics nomenclature

	Roll Axis \hat{x}_b	Pitch Axis \hat{y}_b	Yaw Axis \hat{z}_b
Angular rates	p	q	r
Velocity components	u	v	w
Aerodynamic Force Components	X	Y	Z
Aerodynamic Moment Components	\bar{L}	M	N
Moments of Inertia	I_x	I_y	I_z
Products of Inertia	I_{yz}	I_{xz}	I_{xy}

It is important to note that the forces defined in Table 2.1 are aligned with the body frame. These forces do not directly coincide with the more commonly known aerodynamic forces of lift and drag. The forces of lift and drag are defined with respect to another reference frame, the wind frame, discussed in the following section.

2.3 The Wind and Stability Reference Frames

We need to derive two additional reference frames to resolve the relationship between the commonly known aerodynamic forces of lift and drag and the body forces of the 6-DOF model. These reference frames are the *stability frame* and the *wind frame* as defined by Stevens [SL92]. These reference frames characterize the relationship of the angle of attack, α , and the side-slip angle, β , to the body frame velocities. These aerodynamic angles are defined by means of coordinate rotations. The first rotation, about the \hat{y}_b axis, defines the stability frame and the angle is the angle of attack, α . With no sideslip, α is the angle between the aircraft \hat{x}_b axis and the aircraft velocity vector relative to the surrounding air mass. The angle of attack is positive if the rotation about the \hat{y}_b axis was negative. This ‘backwards’ definition is the unfortunate result of merging the disciplines of aerodynamics and classical kinematics.

The second rotation leads to the wind frame, and the side-slip angle is the angle between the stability frame and the wind frame. An aircraft has sideslip if its velocity vector relative to the air mass is not in the plane defined by \hat{x}_b - \hat{z}_b . The rotation is about the z-axis of the stability frame, \hat{z}_{st} , and β is defined as positive if the rotation about the \hat{z}_{st} axis is positive. The wind frame’s x-axis, \hat{x}_w , is aligned with the aircraft’s velocity vector which is the vector sum of the body frame velocities, $\mathbf{V} = u\hat{x}_b + v\hat{y}_b + w\hat{z}_b$. The other axes, \hat{y}_w and \hat{z}_w , are orthogonal to \hat{x}_w and to each other. Figure 2.5 illustrates the orientation of the x-axes of the stability and wind frames with respect to the body frame.

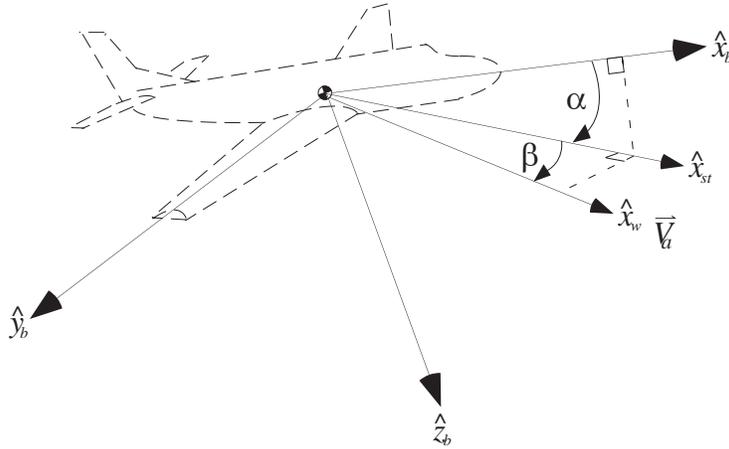


Figure 2.5. Illustration of the stability and wind coordinate systems

Equations (2.7) - (2.9) show the direction cosine matrices which define the transformations between the coordinate frames.

$$\begin{bmatrix} \hat{x}_{st} \\ \hat{y}_{st} \\ \hat{z}_{st} \end{bmatrix} = \begin{bmatrix} C_\alpha & 0 & S_\alpha \\ 0 & 1 & 0 \\ -S_\alpha & 0 & C_\alpha \end{bmatrix} \begin{bmatrix} \hat{x}_b \\ \hat{y}_b \\ \hat{z}_b \end{bmatrix} \quad (2.7)$$

$$\begin{bmatrix} \hat{x}_w \\ \hat{y}_w \\ \hat{z}_w \end{bmatrix} = \begin{bmatrix} C_\beta & S_\beta & 0 \\ -S_\beta & C_\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{st} \\ \hat{y}_{st} \\ \hat{z}_{st} \end{bmatrix} \quad (2.8)$$

$$\begin{bmatrix} \hat{x}_w \\ \hat{y}_w \\ \hat{z}_w \end{bmatrix} = \begin{bmatrix} C_\beta C_\alpha & S_\beta & C_\beta S_\alpha \\ -S_\beta C_\alpha & C_\beta & -S_\beta S_\alpha \\ -S_\alpha & 0 & C_\alpha \end{bmatrix} \begin{bmatrix} \hat{x}_b \\ \hat{y}_b \\ \hat{z}_b \end{bmatrix} \quad (2.9)$$

Using the direction cosine matrices, we can derive expressions for the angles, α and β . We start with the definition of true airspeed. The true airspeed of an aircraft, V_a , is defined as the magnitude of the aircraft's velocity relative to the air mass surrounding the aircraft. By definition, the only component of this velocity is along the \hat{x}_w axis of the wind frame. That is to say the total aircraft velocity is aligned with the \hat{x}_w axis. Written in equation form, $\mathbf{V} = V_a \hat{x}_w$. Using the inverse of the direction cosine matrix in equation (2.9), we can define the body frame velocities in terms of the true airspeed and the angles α and β .

$$\begin{bmatrix} u \hat{x}_b \\ v \hat{y}_b \\ w \hat{z}_b \end{bmatrix} = \begin{bmatrix} C_\beta C_\alpha & -S_\beta C_\alpha & -S_\alpha \\ S_\beta & C_\beta & 0 \\ C_\beta S_\alpha & -S_\beta S_\alpha & C_\alpha \end{bmatrix} \begin{bmatrix} V_a \hat{x}_w \\ 0 \\ 0 \end{bmatrix} \quad (2.10)$$

The three resulting scalar equations are shown below in Equations (2.11)-(2.13).

$$u = V_a C_\beta C_\alpha \quad (2.11)$$

$$v = V_a S_\beta \quad (2.12)$$

$$w = V_a C_\beta S_\alpha \quad (2.13)$$

Rearranging Equation (2.12) gives us an expression for side-slip.

$$\beta = \sin^{-1} \frac{v}{V_a} \quad (2.14)$$

Taking the quotient of w/u , we can derive an expression for angle of attack as shown in Equations (2.15)-(2.16).

$$\frac{w}{u} = \frac{V_a C_\beta S_\alpha}{V_a C_\beta C_\alpha} = \frac{S_\alpha}{C_\alpha} = \tan \alpha \quad (2.15)$$

$$\alpha = \tan^{-1} \frac{w}{u} \approx \frac{w}{u} \quad (2.16)$$

Assuming that the angle of attack is small, it can be approximated as just the ratio w/u . Often, this expression is used to substitute α for w .

Using the wind reference frame, we can resolve the relationship between the commonly known aerodynamic forces of lift, drag and thrust (L , D , and T , respectively), and the body forces of the 6 DOF model. We can see from the direction cosine matrix (2.9) that if we model the aerodynamic forces on an aircraft in terms of lift, drag, and thrust, equations (2.17) through (2.21) are expressions for X , Y , and Z forces in the body frame. The aircraft weight is not included because it is not an aerodynamic force.

$$\sum \mathbf{F}_a = T\hat{x}_b - L\hat{z}_w - D\hat{x}_w \quad (2.17)$$

$$\begin{bmatrix} X\hat{x}_b \\ Y\hat{y}_b \\ Z\hat{z}_b \end{bmatrix} = \begin{bmatrix} C_\beta C_\alpha & -S_\beta C_\alpha & -S_\alpha \\ S_\beta & C_\beta & 0 \\ C_\beta S_\alpha & -S_\beta S_\alpha & C_\alpha \end{bmatrix} \begin{bmatrix} -D\hat{x}_w \\ 0\hat{y}_w \\ -L\hat{z}_w \end{bmatrix} + \begin{bmatrix} T\hat{x}_b \\ 0\hat{y}_b \\ 0\hat{z}_b \end{bmatrix} \quad (2.18)$$

$$X = T - DC_\beta C_\alpha + LS_\alpha \quad (2.19)$$

$$Y = -DS_\beta \quad (2.20)$$

$$Z = -DC_\beta S_\alpha - LC_\alpha \quad (2.21)$$

Note that the actual forces and moments of the full 6 DOF model include much more than simply lift, drag, and thrust. Unsteady aerodynamics plays a large role in the determination of the complete force and moment model.

Before continuing, there is one more term which must be formally defined. This term is the flight path angle. The flight path angle, γ_a , is the angle that the \hat{x}_w axis makes with the x-y plane of the inertial frame (horizontal). This angle characterizes the vertical flight path of the aircraft and is formally defined in Equation (2.22).

$$\gamma_a = \sin^{-1} \left(C_\alpha C_\beta S_\theta - (S_\phi S_\beta + C_\phi S_\alpha C_\beta) C_\theta \right) \quad (2.22)$$

The 'a' subscript on the flight path angle denotes that it is an aerodynamic flight path angle. This is to say that it is the aircraft's flight path angle relative to the air mass. The aircraft's flight path angle relative to the ground is generally different because of the influence of winds.

We can see from Equation (2.22) that if both α and β are zero, the Euler angle θ reduces to γ_a .

2.4 The Derivation of the Six Degree of Freedom Equations of Motion

Once the reference frames and nomenclature are defined, the derivation of the equations of motion is straightforward. The linear equations of motion are derived by summing the forces to the time rate of change of linear momentum (*mass* \times *acceleration*). The acceleration of the aircraft's velocity is determined using the basic kinematic equation which states that the total acceleration of the aircraft with respect to the inertial frame is equal to the derivative of the velocity vector with respect to the body frame plus the cross product of the angular velocity between the inertial and body frames and the velocity vector. The basic kinematic equation is shown in Equation (2.23)

$$\mathbf{a}_i = \frac{d\mathbf{V}}{dt} = \left(\frac{\partial \mathbf{V}}{\partial t} \right)_b + \boldsymbol{\omega} \times \mathbf{V} \quad (2.23)$$

where

- \mathbf{a}_i is the acceleration of the aircraft with respect to the inertial frame
- $\frac{d\mathbf{V}}{dt}$ is the total time derivative of the velocity vector
- $\left(\frac{\partial \mathbf{V}}{\partial t} \right)_b$ is the derivative of the velocity vector as seen in the body frame
- $\boldsymbol{\omega}$ is the angular velocity of the body frame relative to the inertial frame: $\boldsymbol{\omega} = p\hat{x}_b + q\hat{y}_b + r\hat{z}_b$
- \mathbf{V} is the velocity vector in the body frame: $\mathbf{V} = u\hat{x}_b + v\hat{y}_b + w\hat{z}_b$.

The expression for the aircraft's acceleration is shown in Equation (2.24) and simplified in Equation (2.25).

$$\frac{d\mathbf{V}}{dt} = \dot{u}\hat{x}_b + \dot{v}\hat{y}_b + \dot{w}\hat{z}_b + (p\hat{x}_b + q\hat{y}_b + r\hat{z}_b) \times (u\hat{x}_b + v\hat{y}_b + w\hat{z}_b) \quad (2.24)$$

$$\frac{d\mathbf{V}}{dt} = (\dot{u} + qw - rv)\hat{x}_b + (\dot{v} + ru - pw)\hat{y}_b + (\dot{w} + pv - qu)\hat{z}_b \quad (2.25)$$

To complete the equations of motion, we must equate the acceleration terms to the applied forces according to Newton's second law ($F=ma$). Table 2.1 summarizes the aerodynamic forces applied to the aircraft; however, the aircraft weight must also be considered. The aircraft's weight always acts downward in the \hat{z}_i direction. Using the direction cosine matrix, the aircraft's weight (mg) can be represented in body frame coordinates as shown in Equation (2.26)

$$mg\hat{z}_i = -mgS_\theta\hat{x}_b + mgC_\theta S_\phi\hat{y}_b + mgC_\theta C_\phi\hat{z}_b \quad (2.26)$$

where

- m is the aircraft's mass
- g is the gravitational acceleration.

Summing the forces and equating the force terms yields the final expression as shown in Equation (2.27).

$$\begin{aligned} (X - mgS_\theta)\hat{x}_b + (Y + mgC_\theta S_\phi)\hat{y}_b + (Z + mgC_\theta C_\phi)\hat{z}_b \\ = m(\dot{u} + qw - rv)\hat{x}_b + m(\dot{v} + ru - pw)\hat{y}_b + m(\dot{w} + pv - qu)\hat{z}_b \end{aligned} \quad (2.27)$$

Equation (2.27) can be broken down into its individual components to yield the 3 force equations of motion as shown in Equations (2.28)-(2.30).

$$X - mgS_\theta = m(\dot{u} + qw - rv) \quad (2.28)$$

$$Y + mgC_\theta S_\phi = m(\dot{v} + ru - pw) \quad (2.29)$$

$$Z + mgC_\theta C_\phi = m(\dot{w} + pv - qu) \quad (2.30)$$

The moment equations are equal to the time rate of change of angular momentum. The angular momentum of the aircraft is equal to the inertia matrix multiplied by the angular velocities. The expression for angular momentum is shown in Equation (2.31) where the symbol H is used to denote the angular momentum.

$$\begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{zx} & 0 & I_z \end{bmatrix} \begin{bmatrix} p\hat{x}_b \\ q\hat{y}_b \\ r\hat{z}_b \end{bmatrix} \quad (2.31)$$

Because aircraft are symmetric, two products of inertia, I_{yz} and I_{xy} , are zero and therefore are eliminated from the angular momentum expression. Equation (2.31) can be expanded to three scalar equations as shown in Equations (2.32)-(2.34).

$$H_x = I_x p\hat{x}_b - I_{xz} r\hat{z}_b \quad (2.32)$$

$$H_y = I_y q\hat{y}_b \quad (2.33)$$

$$H_z = -I_{zx} p\hat{x}_b + I_z r\hat{z}_b \quad (2.34)$$

The time rate of change of each of these expressions is calculated using the ‘Basic Kinematic Equation’ of the form shown in Equation (2.35).

$$\frac{d\mathbf{H}}{dt} = \left(\frac{\partial \mathbf{H}}{\partial t} \right)_b + \boldsymbol{\omega} \times \mathbf{H} \quad (2.35)$$

When the kinematic expressions are summed to their respective moments, the three moment equations are derived. The three moment equations are shown in Equations (2.36)-(2.38). For convenience Table 2.2 summarizes the fundamental kinematic and dynamic equations of motion.

$$\bar{L} = I_x \dot{p} - I_{xz} (\dot{r} + pq) + qr(I_z - I_y) \quad (2.36)$$

$$M = I_y \dot{q} + rp(I_x - I_z) + I_{xz} (p^2 - r^2) \quad (2.37)$$

$$N = -I_{xz} \dot{p} + I_z \dot{r} + pq(I_y - I_x) + I_{xz} qr \quad (2.38)$$

So far we have developed the full 6-DOF equations of motion which are quite involved. The next step in developing a 6-DOF model, well beyond the scope of this discussion, would be to derive expressions for the forces and moments which act on the aircraft. For a complete discussion, refer to Nelson [N89]. The forces and moments which are used in the 6 DOF model are quite different from the simplified subset presented in Equations (2.19) through (2.21). The true forces and moments are complicated expressions which require estimates of unsteady aerodynamic data to handle properly. Our next task is to simplify the 6 DOF equations of motion to 4 DOF equations using some simplifying assumptions.

However, first we will briefly discuss the modal characteristics of the 6 DOF model so we better understand which characteristics are most likely going to influence trajectory propagation.

2.5 The modal Properties of the Six Degree of Freedom Model

Before simplifying the equations of motion to 4-DOF, it is useful to discuss the five modes of motion associated with the 6-DOF model. Three of the modes are second order and two of the modes are first order making up an 8th order system. These modes are:

- Short Period (Longitudinal plane)
- Phugoid (Longitudinal plane)
- Dutch Roll (Lateral-Directional plane)
- Roll (Lateral-Directional plane)
- Spiral (Lateral-Directional plane).

Table 2.2. Summary of kinematic and dynamic equations of motion

<i>Grouping</i>	<i>Equations</i>
Force Equations	$X - mgS_\theta = m(\dot{u} + qw - rv)$ (2.28)
	$Y + mgC_\theta S_\phi = m(\dot{v} + ru - pw)$ (2.29)
	$Z + mgC_\theta C_\phi = m(\dot{w} + pv - qu)$ (2.30)
Moment Equations	$\bar{L} = I_x \dot{p} - I_{xz}(\dot{r} + pq) + qr(I_z - I_y)$ (2.36)
	$M = I_y \dot{q} + rp(I_x - I_z) + I_{xz}(p^2 - r^2)$ (2.37)
	$N = -I_{xz} \dot{p} + I_z \dot{r} + pq(I_y - I_x) + I_{xz} qr$ (2.38)
Body Angular Velocities in terms of Euler angles and Euler rates	$p = \dot{\phi} - \dot{\psi} S_\theta$ (2.39)
	$q = \dot{\theta} C_\phi + \dot{\psi} C_\theta S_\phi$ (2.40)
	$r = \dot{\psi} C_\theta C_\phi - \dot{\theta} S_\phi$ (2.41)
Euler rates in terms of Euler angles and body angular velocities	$\dot{\theta} = qC_\phi - rS_\phi$ (2.42)
	$\dot{\phi} = p + qS_\phi T_\theta + rC_\phi T_\theta$ (2.43)
	$\dot{\psi} = (qS_\phi + rC_\phi) \sec \theta$ (2.44)

The longitudinal dynamics control the forward speed and altitude of the aircraft. There are two second-order oscillatory modes comprising the longitudinal dynamics. These modes are referred to as the short period and the Phugoid mode.

The lateral-directional dynamics consist of one second-order mode and two first-order modes. These modes control the turning dynamics of the aircraft within the lateral plane.

2.5.1 The Short Period Mode

The short period mode is named because it is the faster of the two modes. It is the mode which defines the aircraft's pitching about its center of gravity. The short period mode controls the dynamics between elevator deflection and the aircraft's resulting lift coefficient. Generally, the short period mode is over ten times faster than the Phugoid mode.

2.5.2 The Phugoid Mode

The Phugoid mode is the slower of the two longitudinal modes. We can think of the Phugoid mode as a gradual interchange between potential and kinetic energy about some equilibrium altitude and airspeed. The Phugoid mode is characterized by changes in pitch attitude, altitude, and velocity at a nearly constant lift coefficient. Usually, the Phugoid is over ten times slower than the short period mode and therefore the Phugoid

will have the dominant influence over the aircraft's trajectory. This is illustrated in Figure 2.7.

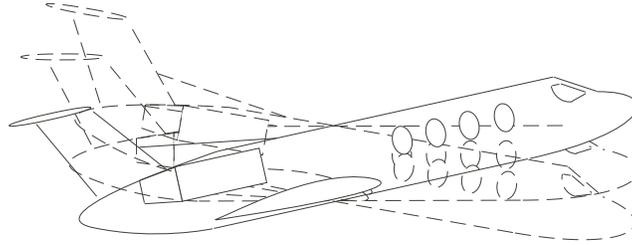


Figure 2.6. Illustration of the Short Period mode causing oscillations about the aircraft's center of gravity

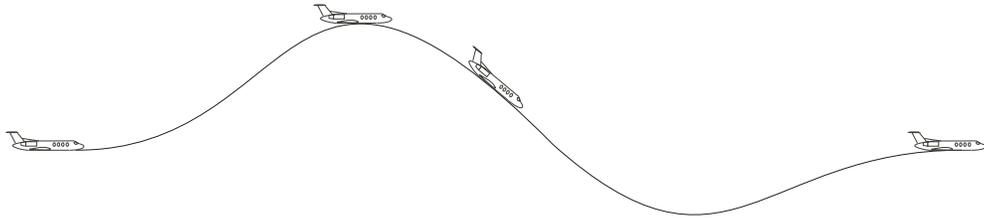


Figure 2.7. Illustration of the Phugoid mode

2.5.3 The Dutch Roll Mode

The Dutch Roll mode is the only oscillatory mode of the lateral directional dynamics and is a combination of yawing and rolling oscillations. The Dutch Roll gets its name from its resemblance to the weaving motion of an ice skater. The Dutch Roll mode is mostly an annoyance to the pilot and passengers. The pilot can easily damp out the motion of the Dutch roll. The Dutch roll is illustrated in Figure 2.8.

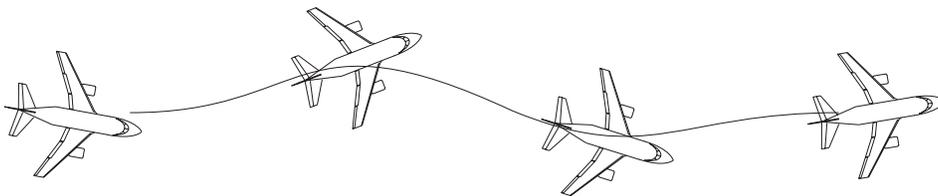


Figure 2.8. Illustration of the Dutch Roll mode

2.5.4 The Roll Mode

The roll mode characterizes how fast an aircraft can achieve a steady state roll rate after an aileron deflection. It is a first-order mode and therefore does not oscillate. The roll mode can influence the trajectory of an aircraft by causing a delay between the time a

turn is commanded and when a steady state turn rate is achieved. The Roll mode is illustrated in Figure 2.9.

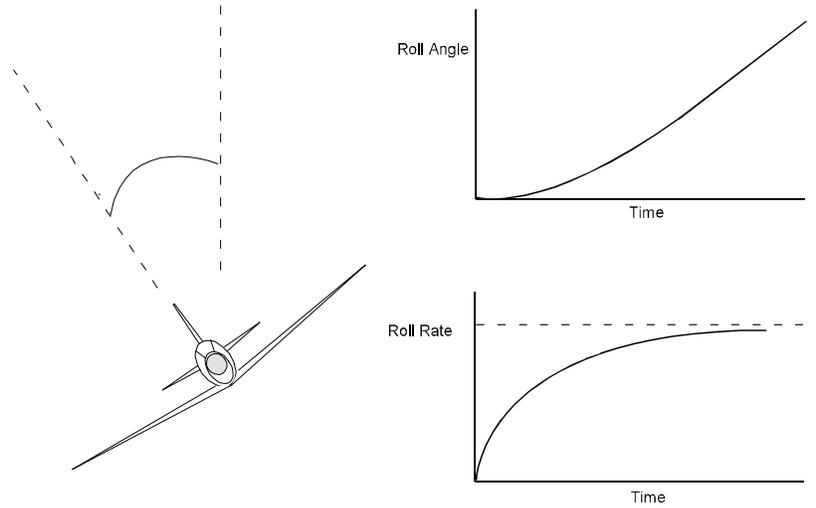


Figure 2.9. Illustration of the Roll mode

2.5.5 The Spiral Mode

The spiral mode characterizes an aircraft's spiral stability about the vertical axis. This mode controls whether or not an aircraft returns to level flight after a small perturbation in roll angle. When this mode is unstable, the aircraft will have a tendency to depart from level flight and enter a spiral dive.

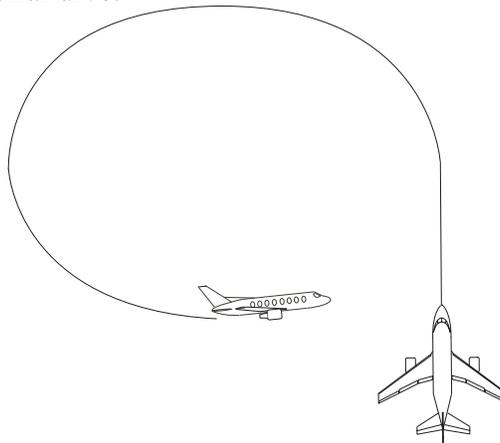


Figure 2.10. Illustration of an unstable Spiral mode

If the mode is stable, the aircraft remains in level flight. Usually the mode is stable. Even if the mode is not stable, the pilot will compensate to maintain straight and level flight.

The next section shows how the 6-DOF model is simplified to yield a model with four degrees of freedom.

2.6 Simplifying the Equations of Motion to Four Degrees of Freedom

The first step to simplifying the 6-DOF equations of motion is to make two assumptions about the aircraft in flight. These assumptions are that:

1. The aircraft's pitch dynamics, characterized by the short period mode, are fast enough to be assumed instantaneous.
2. The pilot maintains 'coordinated flight'.

We first concentrate on the implications of coordinated flight. Constraining the aircraft to coordinated flight implies that the side-slip angle is always zero. This in turn implies that there is never any side velocity, v , any side-force Y , or any yawing moment N . This reduces Equation (2.29), the side force equation, to Equation (2.45). The reduced Equation (2.45) is no longer a differential equation. Furthermore, the yaw rate derivative, \dot{r} , is neglected reducing the yawing moment differential equation to an algebraic expression as shown in Equation (2.46). The rolling moment derivative is removed by the substitution of the rolling moment equation in for \dot{p} .

$$gC_\theta S_\phi = (ru - pw) \quad (2.45)$$

$$0 = -I_{xz} \frac{\bar{L} + I_{xz}(pq) - qr(I_z - I_y)}{I_x} + pq(I_y - I_x) + I_{xz}qr \quad (2.46)$$

Similarly, the side velocity, v , drops out of the other force equations as well. Equations (2.47) and (2.48) show the modified force equations.

$$X - mgS_\theta = m(\dot{u} + qw) \quad (2.47)$$

$$Z + mgC_\theta C_\phi = m(\dot{w} - qu) \quad (2.48)$$

Now, concentrate on the first assumption that the aircraft's pitch dynamics are fast enough to be neglected. This implies that the aircraft is able to command an angle of attack instantaneously. Therefore, we neglect the derivatives, \dot{w} and \dot{q} . The net effect is that the Z force equation and the pitching moment equation are reduced to algebraic expressions.

$$Z + mgC_\theta C_\phi = m(-qu) \quad (2.49)$$

$$M = rp(I_x - I_z) + I_{xz}(p^2 - r^2) \quad (2.50)$$

The terms q and α can be solved explicitly by rearranging Equations (2.49) and (2.50). First, q is solved in Equation (2.51).

$$q = \frac{-Z - mgC_\theta C_\phi}{mu} \quad (2.51)$$

To solve for angle of attack, the pitching moment, M , must be expanded into its individual terms as shown in Equation (2.52). This is merely a formality because we will shortly show how we can remove angle of attack entirely.

$$M = M_o + M_\alpha \alpha + M_{\delta_e} \delta_e = rp(I_x - I_z) + I_{xz}(p^2 - r^2) \quad (2.52)$$

The terms in (2.52) are:

- M_o : The zero angle of attack pitching moment
- M_α : A derivative relating pitching moment changes to changes in angle of attack
- M_{δ_e} : A derivative relating the effect of elevator deflection on angle of attack.

Solving for angle of attack yields Equation (2.53).

$$\alpha = \frac{rp(I_x - I_z) + I_{xz}(p^2 - r^2) - M_o - M_{\delta_e} \delta_e}{M_\alpha} \quad (2.53)$$

Therefore, the angle of attack is an algebraic function of the elevator deflection, roll rate, and yaw rate. The terms $(p^2 - r^2)$ and rp are second order effects and can be neglected.

Therefore, the angle of attack is very nearly a function of the elevator deflection exclusively. This implies that the lift coefficient, C_L , also is very nearly an exclusive function of the elevator deflection as shown in Equations (2.54) and (2.55).

$$C_L = C_{L_o} + C_{L_\alpha} \alpha + C_{L_{\delta_e}} \delta_e \quad (2.54)$$

$$C_L = C_{L_o} + C_{L_\alpha} \left(\frac{-M_o - M_{\delta_e} \delta_e}{M_\alpha} \right) + C_{L_{\delta_e}} \delta_e \quad (2.55)$$

where

- C_{L_o} is the zero angle of attack lift coefficient
- C_{L_α} is the lift curve slope with respect to angle of attack
- $C_{L_{\delta_e}}$ is the effect of elevator deflection on lift coefficient.

Although the angle of attack has been removed from our equations, we can still calculate it for animation purposes. Furthermore, the elevator deflection can be completely bypassed in favor of the aircraft's lift coefficient as the primary longitudinal control input. This is convenient because the terms relating lift coefficient, angle of attack and elevator deflection are not provided in commonly available aircraft models.

At this point, we assume that the angle of attack is small. This assumption combined with the previous assumption of coordinated flight implies that the wind and body frames are very nearly aligned with each other. The alignment of the wind and body frames implies the following:

1. True airspeed, V_a , and u are aligned. Therefore V_a can be substituted for u in the differential equations.
2. The lift force, defined as pointing in the $-\hat{z}_w$ direction, is now aligned with the Z force in Equation (2.30).
3. The drag force, defined as pointing in the $-\hat{x}_w$, is aligned with the Thrust, defined as being aligned with the \hat{x}_b axis.
4. The Euler angle θ reduces to γ_a , the flight path angle.

These simplifications greatly reduce the equations of motion. Equations (2.56) through (2.58) show what remains of our differential equations.

$$T - D - mgS_\theta = m\dot{V}_a \quad (2.56)$$

$$q = \frac{L - mgC_\theta C_\phi}{mV_a} \quad (2.57)$$

$$\bar{L} = I_x \dot{p} - I_{xz}(pq) + qr(I_z - I_y) \quad (2.58)$$

These equations need to be rearranged into a useful form. We start with Equation (2.56) which can easily be rearranged as an expression for true airspeed as shown in Equation (2.59).

$$\dot{V}_a = \frac{T - D}{m} - gS_\theta \quad (2.59)$$

Rearranging Equation (2.57) into an expression for flight path angle takes more steps. We start by relating the pitch rate, q , to the Euler angle θ by using the relations from Table 2.2 yielding Equation (2.60).

$$\dot{\theta} = \frac{L - mgC_\theta C_\phi}{mV_a C_\phi} - \dot{\psi} \frac{C_\theta S_\phi}{C_\phi} \quad (2.60)$$

Using Table 2.2 again, we can substitute for yaw rate, r , in terms of Euler angles in Equation (2.45) yielding Equation (2.61).

$$\dot{\psi} = \frac{gC_\theta S_\phi}{V_a C_\theta C_\phi} + \dot{\theta} \frac{S_\phi}{C_\theta C_\phi} \quad (2.61)$$

Combining equations (2.60) and (2.61) result in the final expressions for $\dot{\theta}$ and $\dot{\psi}$ as shown in equations (2.62) and (2.63).

$$\dot{\theta} = \frac{LC_{\phi} - mgC_{\theta}}{mV_a} \quad (2.62)$$

$$\dot{\psi} = \frac{LS_{\phi}}{mV_a C_{\theta}} \quad (2.63)$$

Using the fact that the flight path angle and the pitch angle are identical for this model, we substitute γ_a for θ . This protects us from confusion that might arise as a result of our simplified Euler angle expressions.

$$\dot{\gamma}_a = \frac{LC_{\phi} - mgC_{\gamma_a}}{mV_a} \quad (2.64)$$

$$\dot{\psi} = \frac{LS_{\phi}}{mV_a C_{\gamma_a}} \quad (2.65)$$

The final equation of motion to manipulate is the rolling moment equation. This equation governs the rate at which an aircraft can establish a bank angle. The rolling moment equation as written is a function of roll and pitch rates and the moments of inertia.

$$\bar{L} = I_x \dot{p} - I_{xz} (pq) + qr(I_z - I_y) \quad (2.66)$$

We choose to neglect the higher order terms and reduce the rolling moment equation to its linear form of $\bar{L} = I_x \dot{p}$. We will use standard stability and control derivatives to define the rolling moment as shown in Equation (2.67).

$$L_p p + L_{\delta_a} \delta_a = I_x \dot{p} \quad (2.68)$$

where

- L_p is the rolling moment derivative with respect to roll rate
- L_{δ_a} is the rolling moment derivative with respect to aileron deflection
- δ_a is the aileron deflection.

We will take one further step to redefine our derivatives to include $1/I_x$ so that we can write our differential equation in first order form.

$$\dot{p} = L_p p + L_{\delta_a} \delta_a \quad (2.69)$$

Table 2.3. The equations of motion for the 4-DOF model

<i>Name</i>	<i>Equation</i>
True Airspeed Equation	$\dot{V}_a = \frac{T - D}{m} - gS_{\gamma_a} \quad (2.59)$
The Flight Path Angle Equation	$\dot{\gamma}_a = \frac{LC_\phi - mgC_{\gamma_a}}{mV_a} \quad (2.64)$
The Heading Equation	$\dot{\psi} = \frac{LS_\phi}{mV_a C_{\gamma_a}} \quad (2.65)$
The Roll Rate Equation	$\dot{p} = L_p p + L_{\delta_a} \delta_a \quad (2.69)$

Since we do not have much data for L_p and L_{δ_a} , we will have to use engineering judgment as to the best values for a specific aircraft. These numbers will be terms that must be ‘tweakable’ so that the user can tune them to suit. The final form of the equations of motion, without wind effects, is shown in Table 2.3. An alternative derivation of the equations of motion may be found in Mukai [Mu92].

2.7 The Addition of Winds

The addition of winds into the system is done by creating yet another reference frame. This frame, the air mass frame, is inserted in between the inertial frame and the wind and body frames. Recall that the wind and body frames are equivalent for our simplified model. The air mass frame’s orientation is aligned with the inertial frame, but moves at a constant velocity with respect to the inertial frame. The aircraft’s velocity motion as described in the previous sections is now considered to be with respect to the air mass frame and not the inertial frame. Knowing that the aircraft’s velocity is aligned with the \hat{x}_w axis we can determine the aircraft’s speed relative to the air mass in inertial coordinates using Equation (2.70).

$$\begin{bmatrix} V_a C_{\gamma_a} C_{\psi} \hat{x}_i \\ V_a C_{\gamma_a} S_{\psi} \hat{y}_i \\ -V_a S_{\gamma_a} \hat{z}_i \end{bmatrix} = \begin{bmatrix} C_{\gamma_a} C_{\psi} & -C_\phi S_{\psi} + S_\phi S_{\gamma_a} C_{\psi} & S_\phi S_{\psi} + C_\phi S_{\gamma_a} C_{\psi} \\ C_{\gamma_a} S_{\psi} & C_\phi C_{\psi} + S_\phi S_{\gamma_a} S_{\psi} & -S_\phi C_{\psi} + C_\phi S_{\gamma_a} S_{\psi} \\ -S_{\gamma_a} & S_\phi C_{\gamma_a} & C_\phi C_{\gamma_a} \end{bmatrix} \begin{bmatrix} V_a \hat{x}_w \\ 0 \hat{y}_w \\ 0 \hat{z}_w \end{bmatrix} \quad (2.70)$$

The aircraft’s speed relative to the inertial frame then includes the velocity of the air mass which we will write as shown in Equation (2.71)

$$\mathbf{V}_w = V_{wx}\hat{x}_i + V_{wy}\hat{y}_i \quad (2.71)$$

where

- V_w is the air mass velocity with respect to the inertial frame
- V_{wx} is the x component of the air mass velocity aligned with true North
- V_{wy} is the y component of the air mass velocity aligned with true East.

The total aircraft velocity with respect to the inertial frame is then the sum of the air mass velocity with respect to the inertial frame and the aircraft's velocity with respect to the air mass. The term \mathbf{V}_i is the aircraft's velocity with respect to the inertial frame or the sum of the aircraft's true airspeed and the wind velocity.

$$\mathbf{V}_i = (V_{wx} + V_a C_{\gamma_a} C_{\psi})\hat{x}_i + (V_{wy} + V_a C_{\gamma_a} S_{\psi})\hat{y}_i - V_a S_{\gamma_a}\hat{z}_i \quad (2.72)$$

The terms in Equation (2.72) can be rewritten as shown in Equation (2.73) to introduce the terms V_x , V_y , and \dot{h} .

$$\mathbf{V}_i = V_x\hat{x}_i + V_y\hat{y}_i - \dot{h}\hat{z}_i \quad (2.73)$$

These terms are defined as follows:

- V_x is the velocity of the aircraft with respect to the inertial frame in the true North direction
- V_y is the velocity of the aircraft with respect to the inertial frame in the true East direction
- \dot{h} is the altitude rate or vertical speed of the aircraft.

Later, the terms V_x and V_y are used when defining the latitude rate and longitude rate of the aircraft. Assuming that the winds are constant, Equation (2.72) is sufficient for modeling the dynamics. However, if the winds are not constant, then wind gradient terms must be added to the differential equations. Often, winds vary with altitude. If this is the case, the equations of motion must be re-derived accounting for the variations in winds with respect to altitude. The resulting equations of motion including wind terms are shown in Table 2.4.

The terms W_{igf} and W_{cgf} are called the in-track gradient factor and the cross-track gradient factors respectively. They are defined as follows:

$$W_{igf} = \frac{dV_{wx}}{dh} S_{\psi_a} - \frac{dV_{wy}}{dh} C_{\psi_a} \quad (2.74)$$

$$W_{cgf} = \frac{dV_{wx}}{dh} C_{\psi_a} + \frac{dV_{wy}}{dh} S_{\psi_a} \quad (2.75)$$

Table 2.4. The equations of motion including wind gradients

<i>Equation</i>	<i>Name</i>
True Airspeed Equation	$\dot{V}_a = \frac{T - D}{m} - gS_{\gamma_a} - W_{igf} V_a S_{\gamma_a} C_{\gamma_a} \quad (2.76)$
Flight Path Angle Equation	$\dot{\gamma}_a = \frac{LC_\phi - mgC_{\gamma_a}}{mV_a} + W_{igf} S_{\gamma_a}^2 \quad (2.77)$
Heading Equation	$\dot{\psi} = \frac{LS_\phi}{mV_a C_{\gamma_a}} - W_{cgf} T_{\gamma_a} \quad (2.78)$
Roll Rate Equation	$\dot{p} = L_p p + L_{\delta_a} \delta_a \quad (2.79)$

2.8 Ellipsoidal Earth Trajectory Propagation Equations

Once the dynamics of the aircraft have been determined, it is necessary to propagate the aircraft's trajectory on the surface of the Earth. The equations which characterize this motion are derived in three steps. These steps are:

1. Develop a set of reference frames
2. Develop an elliptic Earth model
3. Develop the kinematic expressions relating the aircraft's velocity to changes in latitude and longitude.

2.8.1 Elliptic Earth Reference Frames

There are two major reference frames used for the analysis. These reference frames are

- The Earth Centered Earth Fixed (ECEF). We will use an 'e' subscript for brevity of notation when referring to ECEF when writing equations.
- A reference frame on the surface of the earth aligned such that the x-y plane is tangent to the surface. This frame is often referred to the North-East-Down (NED) frame. We will use an 's' subscript for 'surface' for brevity of notation when writing Equations.

The observant reader will notice a difference between the aircraft equations of motion and the trajectory propagation equations in the inertial reference frame. Basically, the aircraft equations of motion were calculated assuming a flat Earth. This is normally how the aircraft equations of motion are handled. This implies necessarily that the aircraft dynamics, which are directly a result of propagating on an elliptic Earth, are ignored. This simplification limits our mathematical model to the flight of aircraft only. The model will not properly handle the flight of sub-orbital craft and spacecraft such as intercontinental ballistic missiles, satellites, or the space shuttle. The model is adequate for all vehicles traveling under Mach 3.

For trajectory propagation, since we can not assume a flat Earth, the original inertial reference frame, denoted with an 'i' subscript, is modified for the elliptic Earth. Thus, we align our newly defined ECEF 'surface' frame, with the inertial frame 'i'. The surface frame then moves with the aircraft so as to provide a frame which is tangent to the Earth's surface as well as providing the aircraft's horizontal plane of flight. All velocities that were originally defined with respect to the 'i' frame, are now taken to be with respect to the 's' frame.

Figure 2.11 shows the relationship between the two reference frames on a spherical earth. The inertial (ECEF) frame is fixed in the center of the earth with the \hat{x}_e axis out the North pole. The Plane described by the \hat{y}_e and the \hat{z}_e passes through the plane of the equator. The negative \hat{z}_e axis is through zero degrees longitude. The second frame is tangent to the surface of the earth and is centered in an object propagating along the Earth's surface. It is represented with an 's' subscript for surface. The \hat{z}_s axis points downward and is normal to the surface of the earth. The \hat{x}_s and \hat{y}_s axes define a plane tangent to the surface of the earth. The longitude and latitude angles describe the rotation between the two frames.

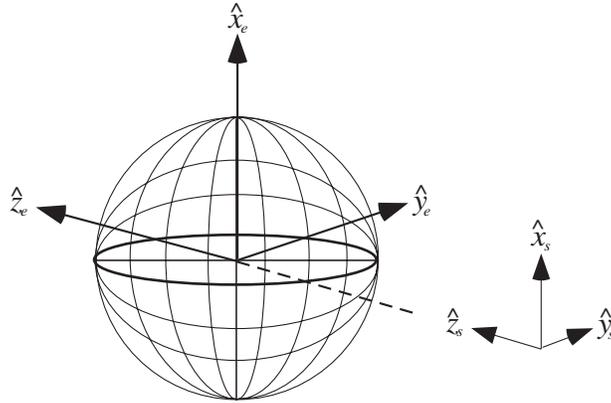


Figure 2.11. The Earth Centered Earth Fixed (ECEF) reference frame and the surface frame

The orientation of reference frames is chosen so that at zero degrees longitude and latitude, the two reference frames coincide. Furthermore, the surface frame conforms to the orientation normally used with aircraft as shown in Figure 2.2. Figure 2.2 shows the relationship between the surface frame and the aircraft's body fixed frame. The angular rotation between the two frames can be thought of as an Euler sequence of rotation. The ordered rotations are illustrated in Figure 2.12.

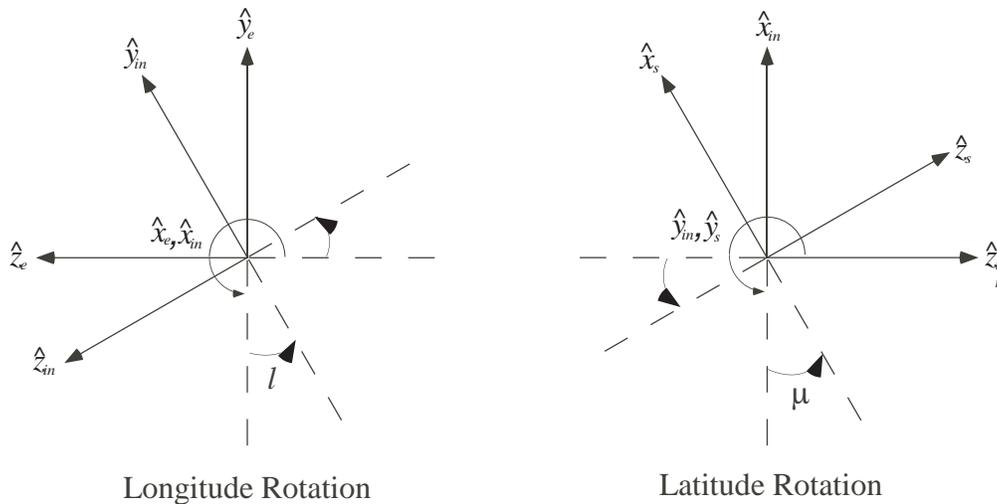


Figure 2.12. The longitude and latitude rotations

The first rotation is longitude. Longitude (l) is rotated about the positive \hat{x}_e axis in a right handed sense from the 'e' frame to an intermediate frame noted with an 'in'. The rotation yields a direction cosine matrix in Equation (2.80).

$$\begin{bmatrix} \hat{x}_{in} \\ \hat{y}_{in} \\ \hat{z}_{in} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos l & \sin l \\ 0 & -\sin l & \cos l \end{bmatrix} \begin{bmatrix} \hat{x}_e \\ \hat{y}_e \\ \hat{z}_e \end{bmatrix} \quad (2.80)$$

The latitude angle (μ) is rotated about the positive \hat{y}_e axis in a right handed sense from the 'in' frame to the surface frame denoted with a 's'. The transformation to the surface frame is shown in Equation (2.81). The reader will note that this latitude angle rotation is opposite of normal mapping convention. That is to say, an aircraft with a negative latitude will be in the Northern hemisphere and an aircraft with a positive latitude will be in the Southern hemisphere. This is an unfortunate consequence of merging the two disciplines of cartography and aircraft dynamics. The negative latitude angle is used to maintain a right handed system of coordinate frames. The software design will be required to reverse the sign of the latitude in the simulation.

$$\begin{bmatrix} \hat{x}_s \\ \hat{y}_s \\ \hat{z}_s \end{bmatrix} = \begin{bmatrix} \cos \mu & 0 & -\sin \mu \\ 0 & 1 & 0 \\ \sin \mu & 0 & \cos \mu \end{bmatrix} \begin{bmatrix} \hat{x}_{in} \\ \hat{y}_{in} \\ \hat{z}_{in} \end{bmatrix} \quad (2.81)$$

The product of these two matrices is the complete direction cosine matrix between the two reference frames.

$$\begin{bmatrix} \hat{x}_s \\ \hat{y}_s \\ \hat{z}_s \end{bmatrix} = \begin{bmatrix} \cos \mu & 0 & -\sin \mu \\ 0 & 1 & 0 \\ \sin \mu & 0 & \cos \mu \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos l & \sin l \\ 0 & -\sin l & \cos l \end{bmatrix} \begin{bmatrix} \hat{x}_e \\ \hat{y}_e \\ \hat{z}_e \end{bmatrix} \quad (2.82)$$

or

$$\begin{bmatrix} \hat{x}_s \\ \hat{y}_s \\ \hat{z}_s \end{bmatrix} = \begin{bmatrix} \cos \mu & \sin l \sin \mu & -\sin \mu \cos l \\ 0 & \cos l & \sin l \\ \sin \mu & -\sin l \cos \mu & \cos l \cos \mu \end{bmatrix} \begin{bmatrix} \hat{x}_e \\ \hat{y}_e \\ \hat{z}_e \end{bmatrix} \quad (2.83)$$

The inverse of the matrix can be calculated as well.

$$\begin{bmatrix} \hat{x}_e \\ \hat{y}_e \\ \hat{z}_e \end{bmatrix} = \begin{bmatrix} \cos \mu & 0 & \sin \mu \\ \sin l \sin \mu & \cos l & -\sin l \cos \mu \\ -\sin \mu \cos l & \sin l & \cos l \cos \mu \end{bmatrix} \begin{bmatrix} \hat{x}_s \\ \hat{y}_s \\ \hat{z}_s \end{bmatrix} \quad (2.84)$$

2.8.2 The Ellipsoidal Earth Definition

The real earth is not a perfect sphere but rather ellipsoidal, as the radius at the equator is slightly larger than the radius at the poles. The WGS-84 earth model is an ellipsoidal earth model [SL92] as shown in Figure 2.13 where:

- b is the equatorial radius which is $2.092565 \times 10^7 \text{ ft}$ (ref. [SL92])
- a is the semi-minor axis given by $a = b(1 - f)$ where f is the *Earth Flattening Parameter*. ($f = 1/298.257$)

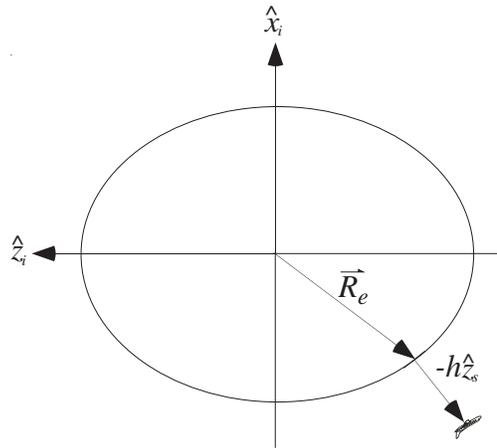


Figure 2.13. The ellipsoidal Earth model

There are two complications created by the ellipsoidal earth. These are:

- The term \mathbf{R}_e , the position vector from the center of the earth to a point on the surface, does not have a constant magnitude. (With a spherical Earth \vec{R}_e is equal to the sea level radius)
- \mathbf{R}_e is not normal to the surface of the Earth.

The fact that \mathbf{R}_e is not normal to the surface of the Earth creates the need for two different latitude definitions: *geocentric* and *geodetic* latitude.

2.8.3 Geocentric Latitude and Geodetic Latitude

To aid in the discussion of latitude, consider the drawing in Figure 2.14. The angle between \vec{R}_e and the equatorial plane is called *geocentric* latitude, λ . Note that the line \mathbf{R}_e is not normal to the earth's surface. Most maps and other navigational references use *geodetic* latitude, μ , which describes a line normal to the Earth's surface. As can be seen in Figure 2.14, geodetic latitude describes a line normal to the Earth's surface that does not go through the center of the Earth but rather intersects the equatorial plane at some other location.

While each latitude definition has its advantages, our definition of coordinate systems requires the use of geodetic latitude because the surface frame is defined such that its \hat{z}_s axis is normal to the surface. Geodetic latitude or some permutation of it must be used for reference frame definition.

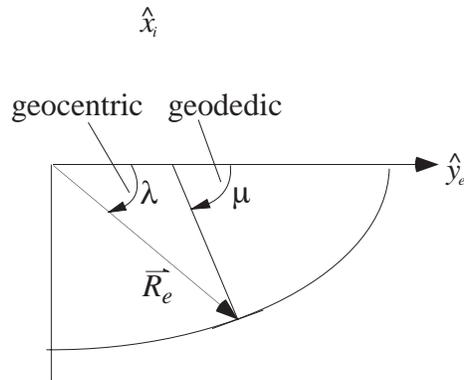


Figure 2.14. Geodetic and geocentric latitudes

2.8.4 Determining Geocentric Latitude in terms of Geodetic Latitude

Because geodetic latitude defines the rotation between coordinate systems, a choice is made to derive all of the equations in terms of geodetic latitude. Therefore, a relation between geodetic latitude and geocentric latitude must be determined. Consider the representation of geocentric latitude, λ , and geodetic latitude, μ , on Figure 2.15.

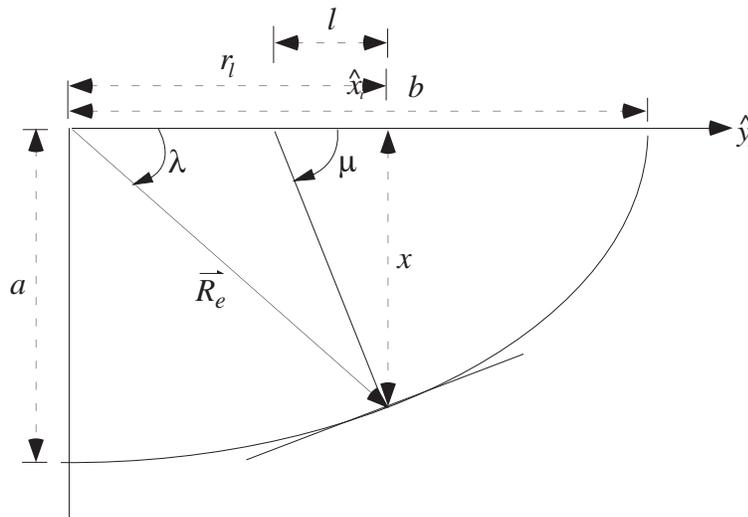


Figure 2.15. Elliptic Earth terminology

By referring to Figure 2.15 for nomenclature definition, the equation for an ellipse is given simply by Equation (2.85).

$$\frac{x^2}{a^2} + \frac{r_l^2}{b^2} = 1 \quad (2.85)$$

and can be expanded to

$$\frac{b^2 x^2 + a^2 r_l^2}{a^2 b^2} = 1 \quad (2.86)$$

Solving for x in terms of r_l leads to

$$\begin{aligned} x &= \sqrt{\frac{a^2 b^2 - a^2 r_l^2}{b^2}} \\ &= \sqrt{a^2 - \left(\frac{a^2}{b^2}\right) r_l^2} \end{aligned} \quad (2.87)$$

From observation we can see that the latitudes are described as

$$\tan \lambda = \frac{x}{r_l} \quad \text{and} \quad \tan \mu = \frac{x}{l}$$

where l is an unknown distance. Thinking of x as a dependent variable and r_l as the independent variable, it is possible to take the derivative of the expression for x with respect to r_l and determine the slope of the ellipse at any given point.

$$\begin{aligned} \frac{dx}{dr_l} &= \frac{1}{2} \left(\frac{a^2 b^2 - a^2 r_l^2}{b^2} \right)^{-\frac{1}{2}} \left(-2 \frac{a^2}{b^2} r_l \right) \\ &= \frac{-\frac{a^2}{b^2} r_l}{\sqrt{\frac{a^2 b^2 - a^2 r_l^2}{b^2}}} \\ \frac{dx}{dr_l} &= \frac{-\frac{a^2}{b^2} r_l}{\sqrt{a^2 - \left(\frac{a^2}{b^2}\right) r_l^2}} \end{aligned} \quad (2.88)$$

The slope of the line normal to the point on the ellipse is the negative reciprocal of the slope of the ellipse, or

$$slope_{normal} = \frac{\sqrt{a^2 - \frac{a^2}{b^2} r_l^2}}{\frac{a^2}{b^2} r_l} \quad (2.89)$$

Noticing that the numerator of the expression is equivalent to x (see Equation (2.87)) allows us to write Equation (2.90).

$$slope_{normal} = \frac{x}{\frac{a^2}{b^2} r_l} \quad (2.90)$$

Observing that $l = \frac{a^2}{b^2} r_l$ allows us to write μ in terms of the slope

$$\tan \mu = \frac{x}{\frac{a^2}{b^2} r_l}$$

which leads to Equation (2.91),

$$r_l \frac{a^2}{b^2} \tan \mu = x \quad (2.91)$$

and finally Equation (2.92).

$$\frac{a^2}{b^2} \tan \mu = \tan \lambda \quad (2.92)$$

2.8.5 Kinematics for an Ellipsoidal Earth

As we start the derivation for the ellipsoidal earth kinematics, we immediately see a difficulty in defining the direction of the vector \vec{R}_e . The vector is neither aligned with an axis of the ECEF frame nor the NED 's' frame as seen Figure 2.16.

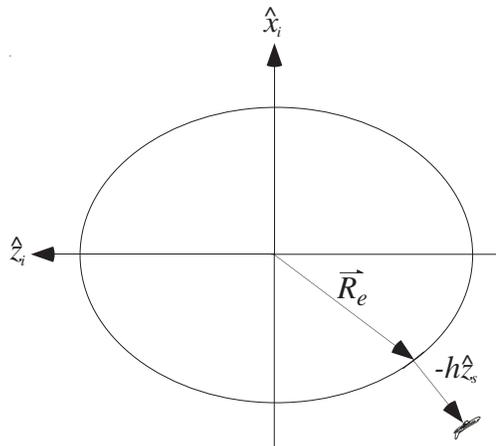


Figure 2.16. The position vector from the center of the Earth to the aircraft

Therefore, we choose to define a new reference frame, the geocentric surface frame, denoted with a subscript 'c', which is defined so that the vector \mathbf{R}_e points along the $-\hat{z}_c$ axis. The geocentric surface frame is defined as one rotation away from the 'geodetic' surface frame through an angle ε about the positive \hat{y}_s axis. Figure 2.17 and Figure 2.18 characterize graphically the relationship between the ellipsoidal earth and the reference frames.

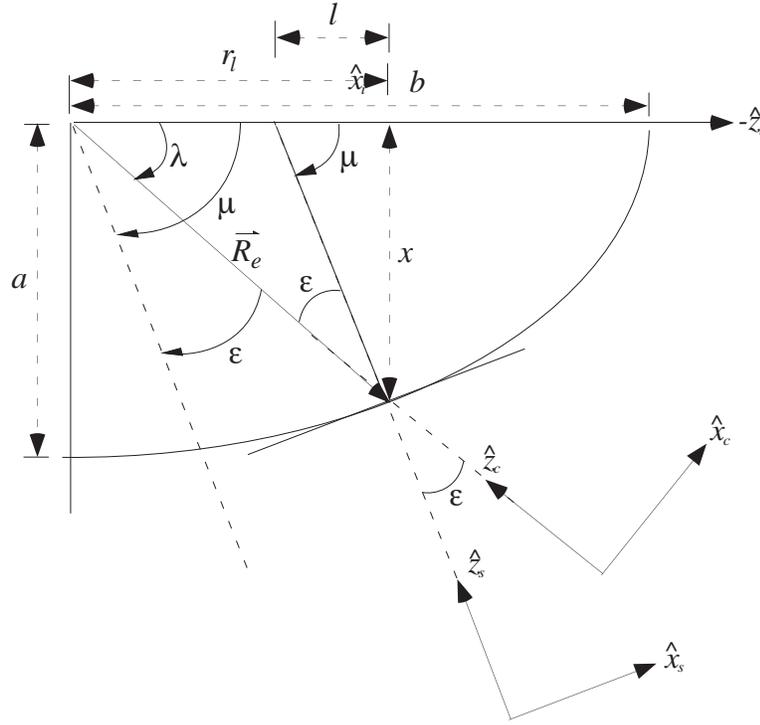


Figure 2.17. Ellipsoidal Earth terminology with the addition of the 'c' frame and the angle ε

The angle ε is defined as the difference between the geodetic latitude and the geocentric latitude as shown in Equation (2.93). The direction cosine matrix relating the surface frame and the 'c' frame is shown in Equation (2.94).

$$\varepsilon = \mu - \lambda \quad (2.93)$$

$$\begin{bmatrix} \hat{x}_s \\ \hat{y}_s \\ \hat{z}_s \end{bmatrix} = \begin{bmatrix} C_\varepsilon & 0 & -S_\varepsilon \\ 0 & 1 & 0 \\ S_\varepsilon & 0 & C_\varepsilon \end{bmatrix} \begin{bmatrix} \hat{x}_c \\ \hat{y}_c \\ \hat{z}_c \end{bmatrix} \quad (2.94)$$

Using the new reference frame, we can define our position vector \mathbf{R}_e as $\mathbf{R}_e = -r_e \hat{z}_c$ where r_e is the magnitude of \mathbf{R}_e . Then the complete position vector from the center of the earth to the aircraft can be written in Equation (2.95).

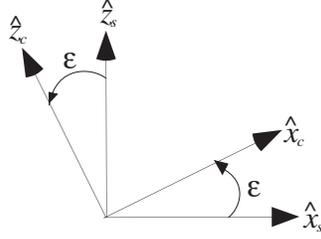


Figure 2.18. The rotation between the surface frame and the 'c' frame

$$\mathbf{R} = -r_e \hat{z}_c - h \hat{z}_s \quad (2.95)$$

Using (2.94) to remove 'c' frame terms, we can write the position vector completely in 's' coordinates.

$$\begin{aligned} \mathbf{R} &= r_e S_\epsilon \hat{x}_s - r_e C_\epsilon \hat{z}_s - h \hat{z}_s \\ &= r_e S_\epsilon \hat{x}_s - (r_e C_\epsilon + h) \hat{z}_s \end{aligned} \quad (2.96)$$

We need to take the derivative of the position vector to get an expression for the aircraft's velocity. The derivative is defined in (2.97) where the angular rotation between the ECEF ('e') frame and the surface frame is defined in (2.98). The altitude, h , is assumed to be constant for these calculations.

$$\frac{d\mathbf{R}}{dt} = (\dot{r}_e S_\epsilon + r_e \dot{\epsilon} C_\epsilon) \hat{x}_s + (r_e \dot{\epsilon} S_\epsilon - \dot{r}_e C_\epsilon) \hat{z}_s + (\boldsymbol{\omega} \times \mathbf{R}) \quad (2.97)$$

$$\begin{aligned} \boldsymbol{\omega} &= \dot{\lambda} \hat{x}_e + \dot{\mu} \hat{y}_s \\ &= \dot{\lambda} \cos \mu \hat{x}_s + \dot{\mu} \hat{y}_s + \dot{\lambda} \sin \mu \hat{z}_s \end{aligned} \quad (2.98)$$

Expanding the cross product term,

$$\begin{aligned} \boldsymbol{\omega} \times \mathbf{R} &= (\dot{\lambda} \cos \mu \hat{x}_s + \dot{\mu} \hat{y}_s + \dot{\lambda} \sin \mu \hat{z}_s) \times (r_e S_\epsilon \hat{x}_s - (r_e C_\epsilon + h) \hat{z}_s) \\ \boldsymbol{\omega} \times \mathbf{R} &= \dot{\lambda} C_\mu (r_e C_\epsilon + h) \hat{y}_s - (\dot{\mu} r_e S_\epsilon) \hat{z}_s - \dot{\mu} (r_e C_\epsilon + h) \hat{x}_s + (\dot{\lambda} r_e S_\epsilon S_\mu) \hat{y}_s \\ \boldsymbol{\omega} \times \mathbf{R} &= -\dot{\mu} (r_e C_\epsilon + h) \hat{x}_s + \dot{\lambda} (C_\mu (r_e C_\epsilon + h) + r_e S_\epsilon S_\mu) \hat{y}_s - (\dot{\mu} r_e S_\epsilon) \hat{z}_s \end{aligned}$$

we end up with the final expression for velocity in Equation (2.99).

$$\begin{aligned} \frac{d\mathbf{R}}{dt} &= (\dot{r}_e S_\epsilon + r_e \dot{\epsilon} C_\epsilon - \dot{\mu} (r_e C_\epsilon + h)) \hat{x}_s + \\ &\quad \dot{\lambda} (C_\mu (r_e C_\epsilon + h) + r_e S_\epsilon S_\mu) \hat{y}_s + \\ &\quad (r_e \dot{\epsilon} S_\epsilon - \dot{r}_e C_\epsilon - \dot{\mu} r_e S_\epsilon) \hat{z}_s \end{aligned} \quad (2.99)$$

We know from previous sections that the velocity of the aircraft in the surface frame is represented in two components in the surface frame as shown in Equation (2.100).

$$\mathbf{V} = V_x \hat{x}_s + V_y \hat{y}_s \quad (2.100)$$

We can set the velocity components of Equation (2.100) equal to the expressions of (2.99) so that we can eventually solve for the latitude and longitude rates.

$$V_x = (\dot{r}_e S_\varepsilon + r_e \dot{\varepsilon} C_\varepsilon - \dot{\mu}(r_e C_\varepsilon + h)) \quad (2.101)$$

$$V_y = \dot{l}(C_\mu(r_e C_\varepsilon + h) + r_e S_\varepsilon S_\mu) \quad (2.102)$$

While Equation (2.99) is a final expression for velocity, it leaves several terms only implicitly defined. To integrate (2.99), we need expressions for the terms r_e , \dot{r}_e , ε , and $\dot{\varepsilon}$ in terms of l , μ , \dot{l} and $\dot{\mu}$. We start with an expression for r_e . Using Figure 2.17 and the classic relationship for an ellipse, we can define r_e in terms of the geocentric latitude.

$$\frac{x^2}{a^2} + \frac{r_l^2}{b^2} = 1 \quad (2.103)$$

$$r_e \cos \lambda = r_l \quad (2.104)$$

$$r_e \sin \lambda = x \quad (2.105)$$

Combining the relations we have Equation(2.106). With some simplification, we can solve for r_e directly in (2.107).

$$\frac{r_e^2 \sin^2 \lambda}{a^2} + \frac{r_e^2 \cos^2 \lambda}{b^2} = 1 \quad (2.106)$$

$$\frac{b^2 r_e^2 \sin^2 \lambda + a^2 r_e^2 \cos^2 \lambda}{a^2 b^2} = 1$$

$$r_e^2 (b^2 \sin^2 \lambda + a^2 \cos^2 \lambda) = a^2 b^2$$

$$r_e = \sqrt{\frac{a^2 b^2}{(b^2 \sin^2 \lambda + a^2 \cos^2 \lambda)}} \quad (2.107)$$

The geocentric latitude, λ , is not a primary angle of concern, so it is convenient to use (2.108) as a substitution for λ .

$$\lambda = \tan^{-1} \left(\frac{a^2}{b^2} \tan \mu \right) \quad (2.108)$$

The angle ε is the difference between the geodetic and geocentric latitudes as shown in Equation (2.109) which can be expressed completely in terms of μ by substituting Equation (2.108) into (2.109) as is done in (2.110).

$$\varepsilon = \mu - \lambda \quad (2.109)$$

$$\varepsilon = \mu - \tan^{-1} \left(\frac{a^2}{b^2} \tan \mu \right) \quad (2.110)$$

Using the chain rule we can establish the time derivative of r_e .

$$\dot{r}_e = \frac{dr_e}{dt} = \frac{dr_e}{d\lambda} \frac{d\lambda}{d\mu} \frac{d\mu}{dt} \quad (2.111)$$

$$\frac{dr_e}{d\lambda} = \frac{1}{2} \left(\frac{a^2 b^2}{(b^2 \sin^2 \lambda + a^2 \cos^2 \lambda)} \right)^{-\frac{1}{2}} \left(\frac{(2b^2 - 2a^2) \cos \lambda \sin \lambda}{(b^2 \sin^2 \lambda + a^2 \cos^2 \lambda)^2} \right) \quad (2.112)$$

which can be reduced to

$$\frac{dr_e}{d\lambda} = \left(\frac{(2b^2 - 2a^2) \cos \lambda \sin \lambda}{2ab(b^2 \sin^2 \lambda + a^2 \cos^2 \lambda)^{\frac{3}{2}}} \right) \quad (2.113)$$

Similarly, we can take the derivative of λ with respect to μ .

$$\frac{d\lambda}{d\mu} = \frac{\frac{a^2}{b^2} \sec^2 \mu}{1 + \left(\frac{a^4}{b^4} \tan^2 \mu \right)} \quad (2.114)$$

$$\dot{r}_e = \left(\frac{(2b^2 - 2a^2) \cos \lambda \sin \lambda}{2ab(b^2 \sin^2 \lambda + a^2 \cos^2 \lambda)^{\frac{3}{2}}} \right) \left(\frac{\frac{a^2}{b^2} \sec^2 \mu}{1 + \left(\frac{a^4}{b^4} \tan^2 \mu \right)} \right) \dot{\mu} \quad (2.115)$$

Finally, we can write an expression for $\dot{\varepsilon}$ using Equations (2.114) and (2.109).

$$\dot{\varepsilon} = \left(1 - \frac{\frac{a^2}{b^2} \sec^2 \mu}{1 + \left(\frac{a^4}{b^4} \tan^2 \mu \right)} \right) \dot{\mu} \quad (2.116)$$

To simplify the final expressions, we choose to make the following substitutions:

$$K_{i_e} = \left(\frac{(2b^2 - 2a^2) \cos \lambda \sin \lambda}{2ab(b^2 \sin^2 \lambda + a^2 \cos^2 \lambda)^{\frac{3}{2}}} \right) \left(\frac{\frac{a^2}{b^2} \sec^2 \mu}{1 + \left(\frac{a^4}{b^4} \tan^2 \mu \right)} \right) \quad (2.117)$$

$$K_{\dot{\varepsilon}} = \left(1 - \frac{\frac{a^2}{b^2} \sec^2 \mu}{1 + \left(\frac{a^4}{b^4} \tan^2 \mu \right)} \right) \quad (2.118)$$

We can then substitute into Equations (2.101) and (2.102) to obtain expressions for $\dot{\varepsilon}$ and \dot{r}_e in terms of $\dot{\mu}$.

$$V_x = (K_{i_e} \dot{\mu} S_\varepsilon + r_e K_{\dot{\varepsilon}} \dot{\mu} C_\varepsilon - \dot{\mu} (r_e C_\varepsilon + h)) \quad (2.119)$$

$$V_y = \dot{l} (C_\mu (r_e C_\varepsilon + h) + r_e S_\varepsilon S_\mu) \quad (2.120)$$

These expressions can be rearranged in terms of \dot{l} and $\dot{\mu}$ as shown in Equations (2.121) and (2.122).

$$\dot{\mu} = \frac{V_x}{(K_{i_e} S_\varepsilon + r_e K_{\dot{\varepsilon}} C_\varepsilon - (r_e C_\varepsilon + h))} \quad (2.121)$$

$$\dot{l} = \frac{V_y}{(C_\mu (r_e C_\varepsilon + h) + r_e S_\varepsilon S_\mu)} \quad (2.122)$$

Table 2.5 summarizes the important results from this section.

2.8.6 X, Y, Z Data for DIS Criteria

It is necessary that the ADM aircraft conform to the DIS standard for representing aircraft trajectory propagation. The DIS standard [DIS98] is similar to the ECEF ‘*i*’ frame which we have defined; however, the DIS standard has the z-axis out the North pole and the x-axis out the zero degrees longitude line. The difference between the two frames is illustrated in Figure 2.19.

Table 2.5. The Latitude and Longitude Trajectory Propagation Equations

<i>Grouping</i>	<i>Equations</i>
Latitude Equation	$\dot{\mu} = \frac{V_x}{(K_{i_\varepsilon} S_\varepsilon + r_e K_\varepsilon C_\varepsilon - (r_e C_\varepsilon + h))} \quad (2.121)$
Longitude Equation	$\dot{i} = \frac{V_y}{(C_\mu (r_e C_\varepsilon + h) + r_e S_\varepsilon S_\mu)} \quad (2.122)$
Dependencies	$r_e = \sqrt{\frac{a^2 b^2}{(b^2 \sin^2 \lambda + a^2 \cos^2 \lambda)}} \quad (2.107)$
	$\varepsilon = \mu - \tan^{-1} \left(\frac{a^2}{b^2} \tan \mu \right) \quad (2.110)$
	$K_{i_\varepsilon} = \left(\frac{(2b^2 - 2a^2) \cos \lambda \sin \lambda}{2ab(b^2 \sin^2 \lambda + a^2 \cos^2 \lambda)^{\frac{3}{2}}} \right) \left(\frac{\frac{a^2}{b^2} \sec^2 \mu}{1 + \left(\frac{a^4}{b^4} \tan^2 \mu \right)} \right) \quad (2.117)$
	$K_\varepsilon = \left(1 - \frac{\frac{a^2}{b^2} \sec^2 \mu}{1 + \left(\frac{a^4}{b^4} \tan^2 \mu \right)} \right) \quad (2.118)$

The Conversion between the surface frame and the ECEF is accomplished with direction cosine matrix multiplication as defined in Equations (2.123) and (2.124), where the position vector, \mathbf{R}_e , is expressed in ECEF coordinates.

$$\begin{bmatrix} (r_e S_\varepsilon C_\mu - (r_e C_\varepsilon + h) S_\mu) \hat{x}_e \\ (r_e S_\varepsilon S_l S_\mu + (r_e C_\varepsilon + h) S_l C_\mu) \hat{y}_e \\ (-r_e S_\varepsilon S_\mu C_l - (r_e C_\varepsilon + h) C_l C_\mu) \hat{z}_e \end{bmatrix} = \begin{bmatrix} C_\mu & 0 & S_\mu \\ S_l S_\mu & C_l & -S_l C_\mu \\ -S_\mu C_l & S_l & C_l C_\mu \end{bmatrix} \begin{bmatrix} r_e S_\varepsilon \hat{x}_s \\ 0 \hat{y}_s \\ -(r_e C_\varepsilon + h) \hat{z}_s \end{bmatrix} \quad (2.123)$$

$$\begin{aligned} \mathbf{R} = & (r_e S_\varepsilon C_\mu - (r_e C_\varepsilon + h) S_\mu) \hat{x}_e \\ & + (r_e S_\varepsilon S_l S_\mu + (r_e C_\varepsilon + h) S_l C_\mu) \hat{y}_e \\ & + (-r_e S_\varepsilon S_\mu C_l - (r_e C_\varepsilon + h) C_l C_\mu) \hat{z}_e \end{aligned} \quad (2.124)$$

To get the position vector from ECEF coordinates to DIS coordinates, we use the direction cosine matrix in Equation (2.125) as shown in Equation (2.126).

$$\begin{bmatrix} \hat{x}_{DIS} \\ \hat{y}_{DIS} \\ \hat{z}_{DIS} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_e \\ \hat{y}_e \\ \hat{z}_e \end{bmatrix} \quad (2.125)$$

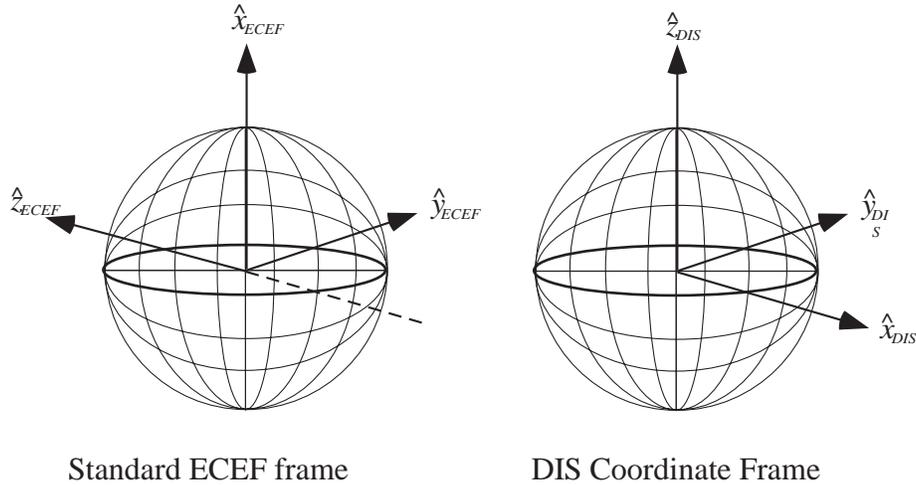


Figure 2.19. The ECEF and DIS Coordinate Frames

Using the direction cosine matrix in (2.125) and Equation (2.96), we can write the following conversions from the surface frame coordinate system to the DIS coordinate frame.

$$\begin{bmatrix} (r_e S_\varepsilon S_\mu C_l + (r_e C_\varepsilon + h) C_l C_\mu) \hat{x}_{DIS} \\ (r_e S_\varepsilon S_l S_\mu + (r_e C_\varepsilon + h) S_l C_\mu) \hat{y}_{DIS} \\ (r_e S_\varepsilon C_\mu - (r_e C_\varepsilon + h) S_\mu) \hat{z}_{DIS} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} (r_e S_\varepsilon C_\mu - (r_e C_\varepsilon + h) S_\mu) \hat{x}_e \\ (r_e S_\varepsilon S_l S_\mu + (r_e C_\varepsilon + h) S_l C_\mu) \hat{y}_e \\ (-r_e S_\varepsilon S_\mu C_l - (r_e C_\varepsilon + h) C_l C_\mu) \hat{z}_e \end{bmatrix} \quad (2.126)$$

$$\begin{aligned} \bar{R}^{op} &= (r_e S_\varepsilon S_\mu C_l + (r_e C_\varepsilon + h) C_l C_\mu) \hat{x}_{DIS} \\ &+ (r_e S_\varepsilon S_l S_\mu + (r_e C_\varepsilon + h) S_l C_\mu) \hat{y}_{DIS} \\ &+ (r_e S_\varepsilon C_\mu - (r_e C_\varepsilon + h) S_\mu) \hat{z}_{DIS} \end{aligned} \quad (2.127)$$

For convenience, we will refer to the DIS coordinates as shown in Equations (2.128) through (2.130).

$$X_{DIS} = (r_e S_\varepsilon S_\mu C_l + (r_e C_\varepsilon + h) C_l C_\mu) \quad (2.128)$$

$$Y_{DIS} = (r_e S_\varepsilon S_l S_\mu + (r_e C_\varepsilon + h) S_l C_\mu) \quad (2.129)$$

$$Z_{DIS} = (r_e S_\varepsilon C_\mu - (r_e C_\varepsilon + h) S_\mu) \quad (2.130)$$

Similarly, the DIS velocities can be derived from Equation (2.99) using the same coordinate transformations as used with the DIS position. The final expressions for the DIS velocities are shown in Equations (2.131) through (2.133).

$$\begin{bmatrix} \dot{X}_{DIS} \hat{x}_{DIS} \\ \dot{Y}_{DIS} \hat{y}_{DIS} \\ \dot{Z}_{DIS} \hat{z}_{DIS} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} C_\mu & 0 & S_\mu \\ S_l S_\mu & C_l & -S_l C_\mu \\ -S_\mu C_l & S_l & C_l C_\mu \end{bmatrix} \begin{bmatrix} (\dot{r}_e S_\varepsilon + r_e \dot{\varepsilon} C_\varepsilon - \dot{\mu}(r_e C_\varepsilon + h)) \hat{x}_s \\ (C_\mu (r_e C_\varepsilon + h) + r_e S_\varepsilon S_\mu) \hat{y}_s \\ (r_e \dot{\varepsilon} S_\varepsilon - \dot{r}_e C_\varepsilon - \dot{\mu} r_e S_\varepsilon) \hat{z}_s \end{bmatrix}$$

$$\begin{aligned} \dot{X}_{DIS} &= (\dot{r}_e S_\varepsilon + r_e \dot{\varepsilon} C_\varepsilon - \dot{\mu}(r_e C_\varepsilon + h)) S_\mu C_l - \dot{l} (C_\mu (r_e C_\varepsilon + h) + r_e S_\varepsilon S_\mu) S_l \\ &\quad - (r_e \dot{\varepsilon} S_\varepsilon - \dot{r}_e C_\varepsilon - \dot{\mu} r_e S_\varepsilon) C_l C_\mu \end{aligned} \quad (2.131)$$

$$\begin{aligned} \dot{Y}_{DIS} &= (\dot{r}_e S_\varepsilon + r_e \dot{\varepsilon} C_\varepsilon - \dot{\mu}(r_e C_\varepsilon + h)) S_l S_\mu + \dot{l} (C_\mu (r_e C_\varepsilon + h) + r_e S_\varepsilon S_\mu) C_l \\ &\quad - (r_e \dot{\varepsilon} S_\varepsilon - \dot{r}_e C_\varepsilon - \dot{\mu} r_e S_\varepsilon) S_l C_\mu \end{aligned} \quad (2.132)$$

$$\dot{Z}_{DIS} = (\dot{r}_e S_\varepsilon + r_e \dot{\varepsilon} C_\varepsilon - \dot{\mu}(r_e C_\varepsilon + h)) C_\mu + (r_e \dot{\varepsilon} S_\varepsilon - \dot{r}_e C_\varepsilon - \dot{\mu} r_e S_\varepsilon) S_\mu \quad (2.133)$$

The terms r_e , \dot{r}_e , ε , and $\dot{\varepsilon}$ are defined using the same relations developed in Section 2.8.5.

2.9 The Derived State Variables

The equations of motion are referred to as the state equations because they are the fundamental equations which govern the aircraft's motion. Each state equation is named for the state variable for which it calculates a derivative. In our case, we have four state variables $[V_a \ \gamma_a \ \psi \ p]$ which are governed by the equations in Table 2.3. There are also other important values which are not state variables but rather functions of the state variables. We call these values derived state variables. There are five important derived state variables:

1. V_{IAS} : The indicated Airspeed
2. M : The Mach number
3. ψ_{GT} : The ground track heading
4. V_{GS} : The ground speed
5. $\dot{\psi}$: The turn rate
6. f : The fuel flow/burn rate
7. W : The aircraft weight
8. h : The aircraft altitude

The indicated airspeed is the speed measurement indicated on an Aneroid type airspeed indicator which is hooked to an aircraft's pitot static system. The airspeed indicator measures the difference between the static and ram-air pressures and approximates an airspeed from the pressure difference. The indicated airspeed is not a good estimate of the true airspeed. At higher altitudes the difference between indicated and true airspeed may be in error by as much as 100 kts. To simulate the reading on an airspeed indicator, Equation (2.134) is used to convert from Mach number to indicated airspeed.

$$V_{IAS} = a_0^* \sqrt{\frac{2}{\gamma-1} \left\{ \left[\frac{p}{p_o} \left[\left(1 + \frac{\gamma-1}{2} M^2 \right)^{\frac{\gamma}{\gamma-1}} - 1 \right] + 1 \right]^{\frac{\gamma-1}{\gamma}} - 1 \right\}} \quad (2.134)$$

The terms in Equation (2.134) are defined as follows:

- a^* : The speed of sound.
- γ : The ratio of specific heats for air (not to be confused with the flight path angle). $\gamma = 1.4$ under normal conditions.
- p : The ambient pressure.
- p_o : The sea-level pressure.

To convert indicated airspeed to Mach number, requires rearranging Equation (2.134). Since this algebraic manipulation is not trivial, only the result is provided here.

$$M = \sqrt{\frac{2}{\gamma-1} \left(\left(\frac{p_o}{p} \left(1 + \frac{\gamma-1}{2} \left(\frac{V_{IAS}}{a_o^*} \right)^2 \right)^{\frac{\gamma}{\gamma-1}} - 1 \right) + 1 \right)^{\frac{\gamma-1}{\gamma}} - 1} \quad (2.135)$$

The Mach number is the ratio of the true airspeed and the speed of sound as shown in Equation (2.136).

$$M = \frac{V_a}{a^*} \quad (2.136)$$

The ground speed and the ground track heading are derived from the velocity terms first presented in Equation (2.73).

$$V_G = \sqrt{V_x^2 + V_y^2} \quad (2.137)$$

$$\psi_{GT} = \tan^{-1} \left(\frac{V_y}{V_x} \right) \quad (2.138)$$

The turn rate of the aircraft is calculated using the heading equation. Equation (2.63) is reprinted here for convenience.

$$\dot{\psi} = \frac{L S_{\phi}}{m V_a C_{\gamma_a}} \quad (2.63)$$

There are two ‘derived’ state variables which are not merely functions of the integrated states. These values must be integrated; however, we separate them from the formal integration of the differential equations because they do not require the rigorous integration procedure used to numerically integrate the state equations. These two ‘derived’ states are altitude, h , and aircraft weight, W . The altitude is simply the integration of the altitude rate and the aircraft weight is the integration of the fuel burn rate. The method of integration is discussed in the numeric integration section.

2.10 The Airframe Model

The airframe model is adapted from Seagull Technology’s AMT (Aircraft Modeling Tool) which is adapted from [Bo96]. The primary purpose of the airframe model is to calculate the aerodynamic forces applied to the aircraft. These forces are lift and drag as defined in Anderson [A89].

The lift of the aircraft is calculated using Equation (2.139).

$$L = qS_w C_L \quad (2.139)$$

The drag of the aircraft is calculated using Equation (2.140).

$$D = qS_w C_D \quad (2.140)$$

The terms for these equations are as follows:

1. L Lift
2. D Drag
3. S_w Wing Reference Area
4. q Dynamic pressure
5. C_D Drag coefficient
6. C_L Lift coefficient

To calculate the dynamic pressure Equation (2.141) is used.

$$q = \frac{1}{2} \rho V_a^2 \quad (2.141)$$

where the terms in the equation are defined as follows:

7. ρ air density
8. V_a true airspeed

The air density is obtained from the atmosphere model which is discussed in Section 2.12. The lift coefficient is an input which is usually generated by the control laws. The drag coefficient is calculated using Equation (2.142).

$$C_D = (C_{D_o} + K C_L^2) \quad (2.142)$$

where

1. C_{D_o} The zero lift drag coefficient
2. K The induced drag coefficient

The drag polar equation comes from classical incompressible aerodynamics. The compressibility effects of high speed flight are currently neglected.

Each aircraft in the simulation has five flap settings and a spoiler that can be deployed when needed. The flap settings are named for their respective flight phases.

- Clean configuration

- Initial climb configuration
- Take off configuration
- Approach configuration
- Landing configuration

Each flap setting is described by its own C_{D_0} and K pair.

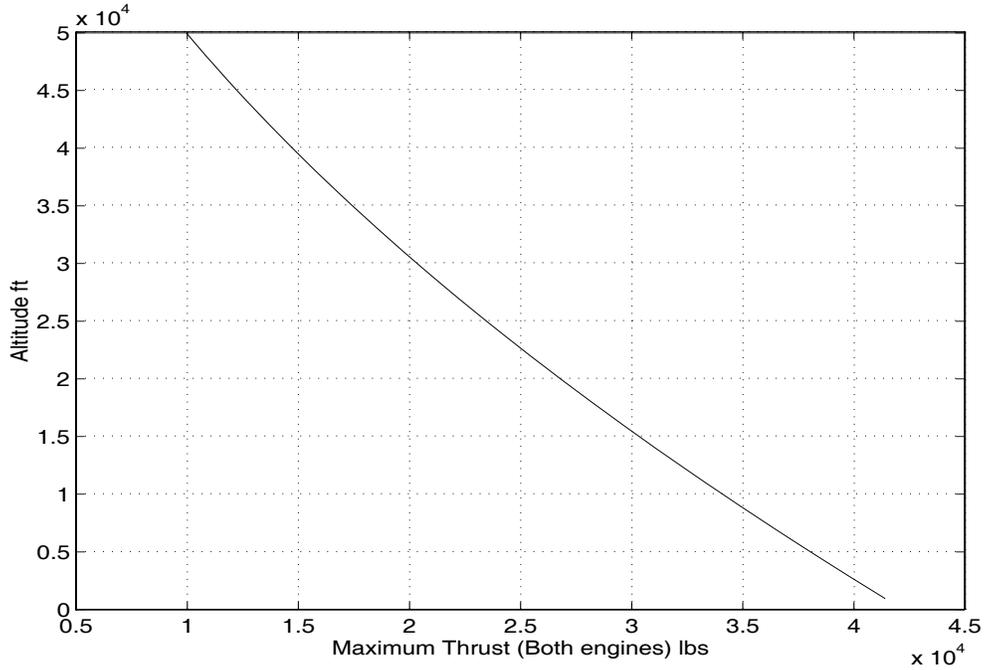


Figure 2.20. Maximum thrust vs altitude for a DC-9/MD80

2.11 The Engine Model

The engine model is responsible for providing two important parameters to the rest of the model. These parameters are the maximum thrust available and the fuel burn rate. We use the AMT data [Bo96] to perform these calculations. The maximum thrust available to the aircraft at any given time is a function of the air's density. Figure 2.20 shows the maximum thrust available for a DC-9 aircraft as the altitude is increased. The AMT data represent the air density in terms of pressure altitude. The maximum available thrust is computed using Equation (2.143).

$$T_{\max} = C_{T_{c,1}} \left(1 - \frac{h}{C_{T_{c,2}}} + C_{T_{c,3}} h^2 \right) \quad (2.143)$$

where:

1. h is altitude

2. T_{\max} is the maximum thrust

There are three thrust coefficients which are used. None of the coefficients have any physical meaning or any name other than their symbolic representation. These coefficients are:

1. $C_{T_c,1}$; Used for calculation of maximum thrust
2. $C_{T_c,2}$; Used for calculation of maximum thrust
3. $C_{T_c,3}$; Used for calculation of maximum thrust

The fuel burn rate is calculated using the following equations.

$$\eta = C_{f_1} \left(1 + \frac{V_a}{C_{f_2}} \right) \quad (2.144)$$

$$f = \eta T \quad (2.145)$$

$$f_{\min} = C_{f_3} \left(1 - \frac{h}{C_{f_4}} \right) \quad (2.146)$$

where:

1. η is the thrust specific fuel consumption
2. V_a is the true airspeed
3. T is the Thrust
4. f the fuel flow rate
5. f_{\min} is the minimum fuel rate
6. h is the altitude

There are four Thrust specific fuel consumption coefficients which comprise the fuel burn model. None of the coefficients have any physical meaning or any name other than their symbolic representation. These coefficients are:

1. C_{f_1} ; Used to calculate thrust specific fuel consumption
2. C_{f_2} ; Used to calculate thrust specific fuel consumption
3. C_{f_3} ; Used to calculate minimum thrust specific fuel consumption
4. C_{f_4} ; Used to calculate minimum thrust specific fuel consumption.

The fuel flow rate is normally calculated using Equations (2.144) and (2.145); however, there is a lower bound on the fuel burn which is calculated using Equation (2.146). If the fuel burn calculated using Equations (2.144) and (2.145) is lower than the minimum fuel

burn rate, the minimum fuel burn rate is returned as the fuel burn rate. Figure 2.21 shows the fuel burn rate for a DC-9 at maximum thrust for various airspeeds.

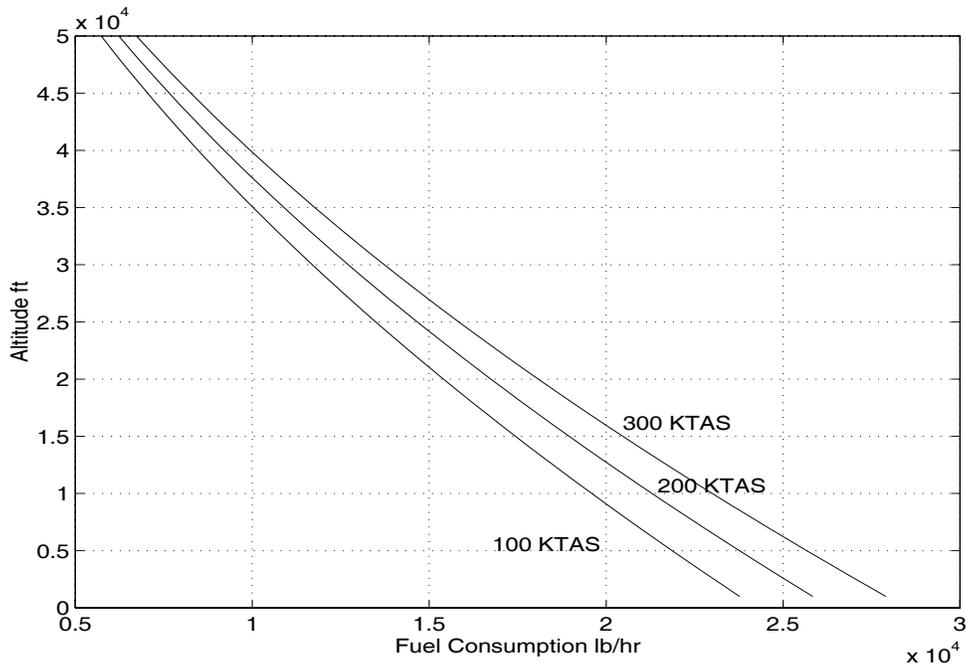


Figure 2.21. Fuel Consumption at maximum thrust (both engines)

2.12 The Standard Day Atmosphere Model

Since aircraft operate in the Earth's atmosphere and their lift and drag characteristics depend on the properties of that atmosphere, it is essential to be able to define these properties. To do this, the Standard Day Atmosphere model as defined by *The ARDC Model Atmosphere*, 1959 by Minzner, R.A., and Champion, K.S. W., and Pond, H.L. is implemented. The derivation of the governing equations is omitted since they are commonly available in Anderson [A89].

There are 2 separate regions to the Earth's atmosphere which we are concerned with. The first region is the gradient region where temperature drops off linearly with altitude. The gradient region spans from the Earth's surface to 36150 ft. The second is an isothermal region where the temperature is constant. The isothermal region spans from 36150 ft to 80000 ft. Figure 2.22 illustrates the temperature variation of the standard atmosphere.

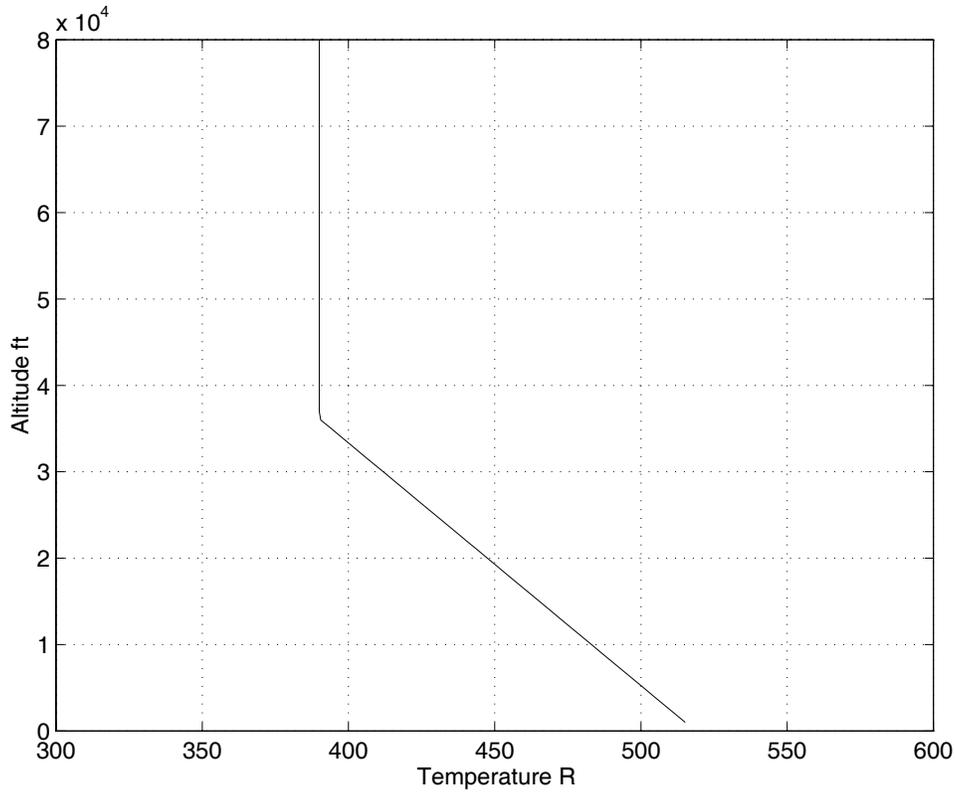


Figure 2.22. Temperature vs altitude for the standard day atmosphere

If the altitude is less than 36150 ft, the temperature of the ambient air surrounding the aircraft is calculated using Equation (2.147).

$$T_{amb} = T_{sl} - ah \quad (2.147)$$

where

1. T_{sl} : The sea level temperature
2. T_{amb} : The ambient temperature
3. h : The altitude
4. a : The temperature lapse rate

If the altitude is 36150 ft or greater, the temperature stays constant. Therefore the ambient temperature above 36150 ft is a constant 389.97 °R.

$$T_{amb} = T_{sl} - a(36150 \text{ ft}) \quad (2.148)$$

The speed of sound is strictly a function of the ambient air temperature. It is calculated using the thermodynamic relation in Equation (2.149)

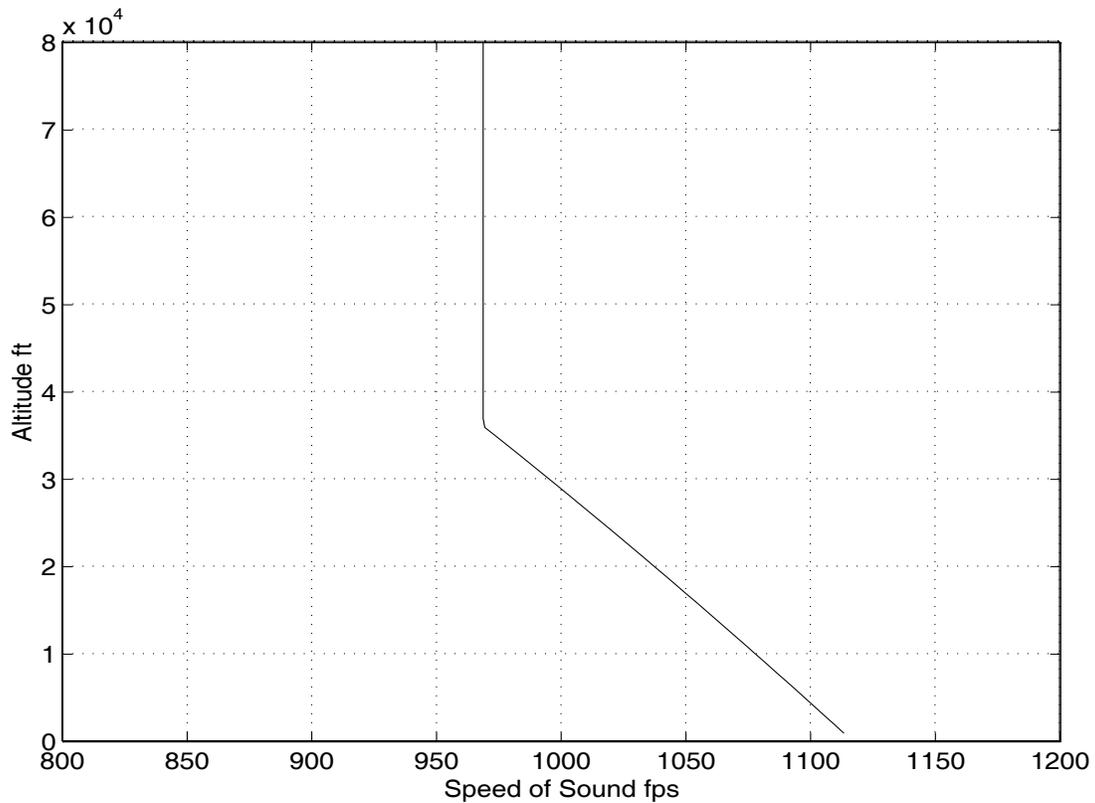


Figure 2.23. The speed of sound variation with altitude for the standard day atmosphere

$$a^* = \sqrt{\gamma RT} \quad (2.149)$$

where a^* is the speed of sound, and γ is the ratio of specific heats for air.

$$\gamma = \frac{c_p}{c_v} \quad (2.150)$$

where c_p is the constant pressure specific heat and c_v is the constant volume specific heat. The term R is the ideal gas constant and T is the absolute ambient air temperature. Figure 2.23 shows the relationship between the speed of sound and altitude for a standard day.

Figure 2.24 illustrates the pressure variation with altitude for the standard day. If the aircraft is below 36150 ft, the pressure ratio of the aircraft is calculated using Equation (2.151).

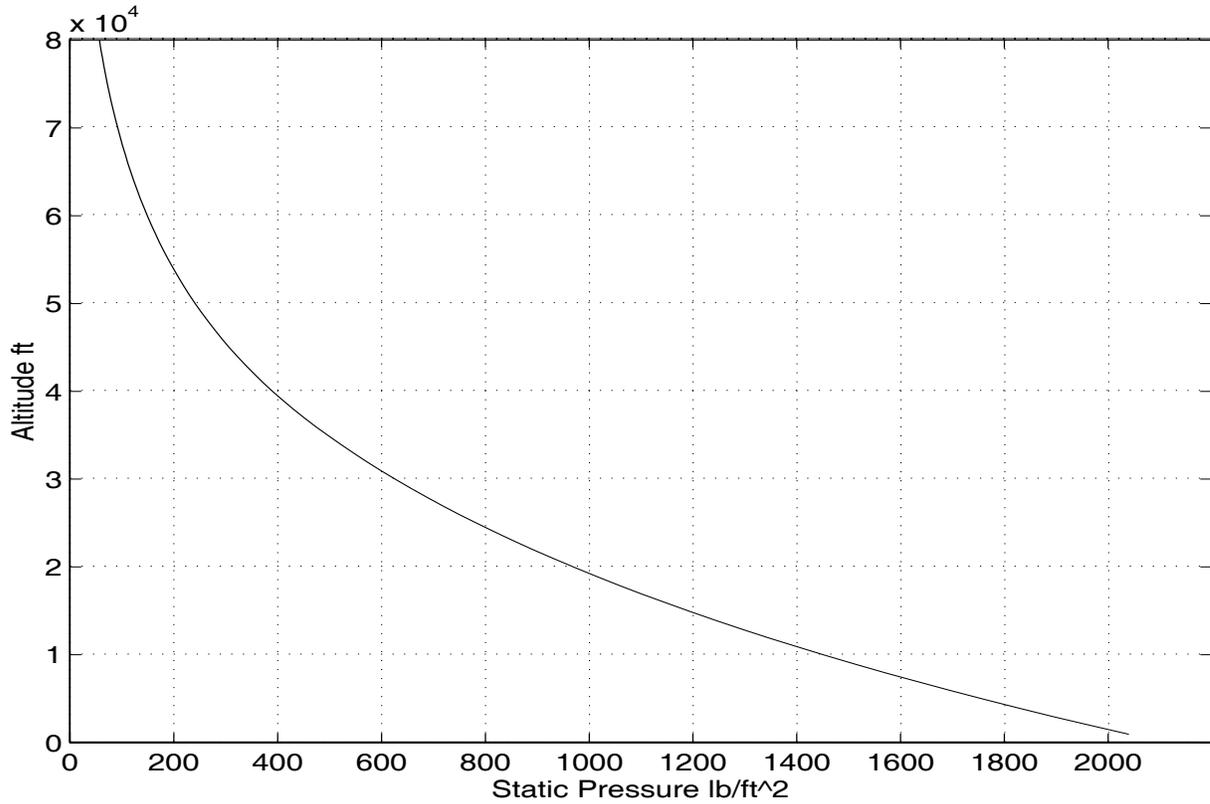


Figure 2.24. Pressure variation with altitude for the standard atmosphere

$$\left(\frac{P_{amb}}{P_{sl}} \right) = \left(\frac{T_{amb}}{T_{sl}} \right)^{\frac{-g}{aR}} \quad (2.151)$$

where

1. P_{amb} : The ambient pressure.
2. $\left(\frac{P_{amb}}{P_{sl}} \right)$: The pressure ratio.
3. g : Gravitational acceleration.
4. a : Temperature lapse rate.
5. R : Ideal gas constant.

If the aircraft is above 36150 ft, the pressure ratio of the aircraft is calculated using Equation (2.152).

$$\left(\frac{P_{amb}}{P_{sl}} \right) = \frac{P_{36150}}{P_{sl}} e^{-\left(\frac{g}{RT_{36150}}\right)(h-36150)} \quad (2.152)$$

where

1. p_{36150} : Pressure at 36150 ft.
2. T_{36150} : Temperature at 36150 ft.

The final equation, which calculates density, is valid regardless of the atmospheric region. It is the equation of state.

$$\rho_{amb} = \frac{P_{amb}}{RT_{amb}} \quad (2.153)$$

where

1. ρ_{amb} : The ambient density

The relationship between density and altitude is illustrated in Figure 2.25.

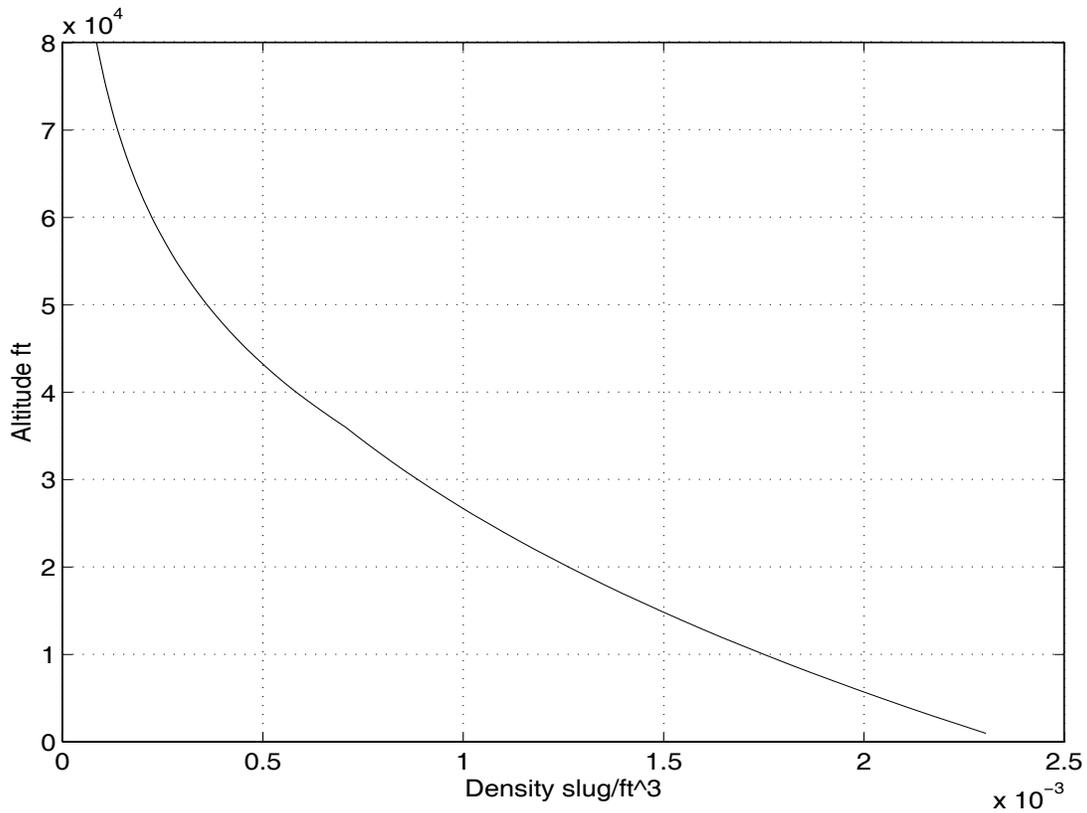


Figure 2.25. Density variation with altitude for the standard atmosphere

2.13 Integration Techniques

The TGF simulation as designed requires the real time integration of a series of nonlinear differential equations and one linear differential equation, Equation (2.79). Since nonlinear differential equations can not be solved analytically, some type of numerical integration method must be employed. There are many techniques available, so it is

important to find a technique well suited to the needs of a particular problem. There are several items to consider when choosing a numerical algorithm. These are:

1. Accuracy required.
2. Frequency of the dynamics to be simulated.
3. The computational efficiency required.
4. The stability of the algorithm.

The most demanding integration requirements for the TGF project stem from the Phugoid mode of the longitudinal dynamics. This mode generally has a period of 30 sec which is not very fast. Therefore, a sophisticated numerical algorithm need not be applied. Furthermore, the integrations which are not influenced by the Phugoid mode require even less computational precision. For the Phugoid influenced equations, a good second order method should suffice. For the non-Phugoid influenced equations, a first order method is quite adequate.

2.13.1 The Second Order Runge-Kutta Method

A second order Runge-Kutta method is chosen for those equations which are influenced by the Phugoid mode. This method, is simple and stable. It is self starting and does not require information from previous time steps. It is slightly more computationally expensive than other methods such as an Adams-Bashforth method, but the use of the Adams-Bashforth method did not prove as stable as the Runge-Kutta method and required a more complex algorithm because it is not self starting. These methods are discussed in detail in Hoffman [H92], or any other numerical method text.

The second order Runge-Kutta algorithm is summarized below. Consider a state vector, $\mathbf{X}(k)$, at time step, k , and, $t(k)$, the time at step, k . These are the inputs to the numerical integrator.

$$\mathbf{X}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} \quad (2.154)$$

It is our objective to update the state vector to the next time step at $(k+1)$. To do this we must first calculate \mathbf{K}_0 , the initial term of the Runge-Kutta integration sequence. The numerical integration routine does not actually do this. Instead, it uses a series of functions of the states and the independent variable, time, as shown in Equation(2.155). For our problem, the functions $f_1 - f_n$ are the state equations (2.59), (2.64), and (2.65).

$$\mathbf{K}_0 = \begin{bmatrix} k_{o_1} \\ k_{o_2} \\ \vdots \\ k_{o_n} \end{bmatrix} = \begin{bmatrix} f_1(x_1(k), x_2(k), \dots, x_n(k), t(k), \dots) \\ f_2(x_1(k), x_2(k), \dots, x_n(k), t(k), \dots) \\ \vdots \\ f_n(x_1(k), x_2(k), \dots, x_n(k), t(k), \dots) \end{bmatrix} \Delta t \quad (2.155)$$

The numerical integration routine will take \mathbf{K}_0 and add it to the original state vector at time step (k) and then send the results back to the derivative functions. This results in \mathbf{K}_1 .

$$\mathbf{K}_1 = \begin{bmatrix} k_{1_1} \\ k_{1_2} \\ \vdots \\ k_{1_n} \end{bmatrix} = \begin{bmatrix} f_1(x_1(k) + k_{o_1}, x_2(k)k_{o_2}, \dots, x_n(k) + k_{o_n}, t(k) + \Delta t, \dots) \\ f_2(x_1(k) + k_{o_1}, x_2(k)k_{o_2}, \dots, x_n(k) + k_{o_n}, t(k) + \Delta t, \dots) \\ \vdots \\ f_n(x_1(k) + k_{o_1}, x_2(k)k_{o_2}, \dots, x_n(k) + k_{o_n}, t(k) + \Delta t, \dots) \end{bmatrix} \Delta t \quad (2.156)$$

The final step is to determine, $\mathbf{X}(k+1)$.

$$\mathbf{X}(k+1) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \frac{1}{2} \left(\begin{bmatrix} k_{o_1} \\ k_{o_2} \\ \vdots \\ k_{o_n} \end{bmatrix} + \begin{bmatrix} k_{1_1} \\ k_{1_2} \\ \vdots \\ k_{1_n} \end{bmatrix} \right) \quad (2.157)$$

The equations which are integrated using the second order Runge-Kutta technique are Equations (2.59), (2.64), and (2.65) or the true airspeed, flight path angle, and the heading angle state variable equations.

2.13.2 The First Order Euler Method

The first order Euler method is arguably the simplest numerical integration routine available. Under most conditions, it is not considered adequate for actual simulation, but rather is used only as an instructional example. However it is very inexpensive computationally, and is more than adequate for the very slow changes in altitude, position and weight changes occurring in the TGF model. Using the same $X(k)$ vector defined in (2.154), the next time step, $X(k+1)$, is easily calculated using Equation (2.158).

$$\mathbf{X}(k+1) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} f_1(x_1(k), x_2(k), \dots, x_n(k), t(k), \dots) \\ f_2(x_1(k), x_2(k), \dots, x_n(k), t(k), \dots) \\ \vdots \\ f_n(x_1(k), x_2(k), \dots, x_n(k), t(k), \dots) \end{bmatrix} \Delta t \quad (2.158)$$

The quantities which are integrated by this method are as follows:

1. Latitude
2. Longitude
3. Weight
4. Altitude

2.13.3 Analytic Integration

For the one linear differential equation, the roll Equation (2.69), an analytic integration method is used. The equation is integrated and then discretized using the state transition matrix and a zero order hold on the inputs.

2.14 The Integration of the Roll Equation

The roll equation is unique in our simulation because it is the only differential equation which is linear. Because the equation is linear, no numerical integration technique need be applied. Furthermore, we can perform the loop closures of our lateral directional control logic within the analytic solution itself. The method of solution uses the state transition matrix and a zero order hold.

2.14.1 The open loop roll rate and roll angle equations

The roll mode is governed by Equation (2.69) which is reprinted below. It is convenient to assign $\dot{\phi}$, the derivative of the roll angle, to p , the roll rate. This yields Equation (2.159). These two equations yield the second order dynamics which characterize an aircraft's roll angle in response to deflection of the ailerons.

$$\dot{p} = L_p p + L_{\delta_a} \delta_a \quad (2.69)$$

$$\dot{\phi} = p \quad (2.159)$$

It is necessary to arrange the equations into state space representation as shown in Equation (2.161).

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} L_p & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} + \begin{bmatrix} L_{\delta_a} \\ 0 \end{bmatrix} \delta_a \quad (2.160)$$

We can examine the response of the open loop dynamics to a unit step in aileron deflection as shown in Figure 2.26. Predictably, we see the aircraft's roll rate rise to a steady state value and the bank angle rise at a constant rate. This is exactly the response one would expect from the open loop roll dynamics.

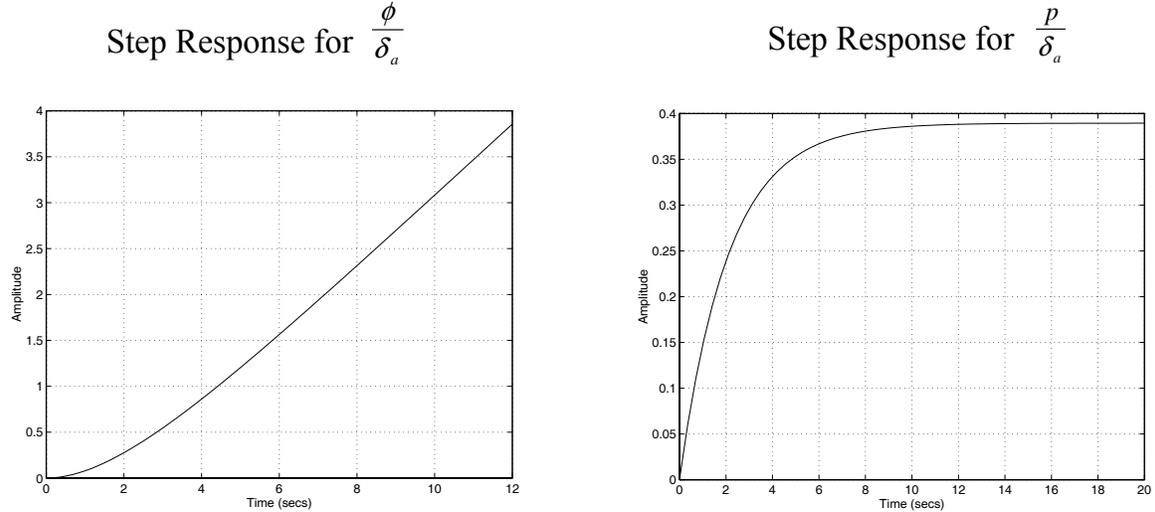


Figure 2.26. The open loop response to an aileron unit step

For our aircraft simulation, the control logic will need to command particular aircraft roll or bank angle by manipulating the ailerons. The desired system is one where a desired roll angle is commanded and the aircraft responds accordingly.

2.14.2 The closed loop system

To create a closed loop system, we use full state feedback of the form shown in Equation (2.161) where \mathbf{A}_{cl} is the closed loop A-matrix and \mathbf{B}_{cl} is the closed loop B-matrix.

$$\begin{aligned} \mathbf{A}_{cl} &= \mathbf{A} - \mathbf{BK} \\ \mathbf{B}_{cl} &= \mathbf{BK} \end{aligned} \tag{2.161}$$

The matrix nomenclature of Equation (2.161) can be expressed in terms of Equation (2.160) as shown in Equation (2.162).

$$\begin{aligned} \mathbf{A}_{cl} = \mathbf{A} - \mathbf{BK} &= \begin{bmatrix} L_p & 0 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} L_{\delta_a} \\ 0 \end{bmatrix} \begin{bmatrix} k_p & k_\phi \end{bmatrix} \\ \mathbf{B}_{cl} = \mathbf{BK} &= \begin{bmatrix} L_{\delta_a} k_p & L_{\delta_a} k_\phi \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{2.162}$$

The closed loop state space system is

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} L_p - L_{\delta_a} k_p & -L_{\delta_a} k_\phi \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} + \begin{bmatrix} L_{\delta_a} k_p & L_{\delta_a} k_\phi \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_{des} \\ \phi_{des} \end{bmatrix} \quad (2.163)$$

where p_{des} is the desired roll rate and ϕ_{des} is the desired roll angle. We do not actually allow a command p_{des} so we can eliminate it from Equation (2.163) as shown in Equation (2.164).

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} L_p - L_{\delta_a} k_p & -L_{\delta_a} k_\phi \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} + \begin{bmatrix} L_{\delta_a} k_\phi \\ 0 \end{bmatrix} \phi_{des} \quad (2.164)$$

2.14.3 Zero Order Hold

It is our intent to create a zero order hold for the system. Before continuing, we substitute in for the coefficients to simplify the expressions as much as possible.

$$\begin{aligned} a_{11} &= L_p - L_{\delta_a} k_p \\ a_{12} &= -L_{\delta_a} k_\phi \end{aligned} \quad (2.165)$$

The state equations can be expressed in terms of a_{11} and a_{12}

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} + \begin{bmatrix} L_{\delta_a} k_1 & L_{\delta_a} k_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_{des} \\ \phi_{des} \end{bmatrix} \quad (2.166)$$

The equation for the discrete-time system with a zero order hold, as defined by Ogata[O70], is shown in Equation (2.167).

$$\mathbf{x}(k+1) = e^{\mathbf{A}\Delta t} \mathbf{x}(k) + e^{\mathbf{A}\Delta t} \int_0^{\Delta t} e^{-\mathbf{A}\tau} \mathbf{B} \mathbf{u}(k) d\tau \quad (2.167)$$

where $\mathbf{x}(k+1)$ is the state vector at the $k+1$ time step, $e^{\mathbf{A}\Delta t}$ is the state transition matrix, $\mathbf{x}(k)$ is the state vector at time step k , and $\mathbf{u}(k)$ is the input at time step k .

2.14.4 The State Transition Matrix

The state transition matrix, $e^{\mathbf{A}t}$, is defined as $\mathcal{L}^{-1}\{(s\mathbf{I} - \mathbf{A})^{-1}\}$. Working through the mathematics we have

$$s\mathbf{I} - \mathbf{A} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} a_{11} & a_{12} \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} s - a_{11} & -a_{12} \\ -1 & s \end{bmatrix} \quad (2.168)$$

$$\det \begin{bmatrix} s - a_{11} & -a_{12} \\ -1 & s \end{bmatrix} = s^2 - a_{11}s - a_{12} \quad (2.169)$$

$$e^{At} = \mathcal{L}^{-1} \begin{bmatrix} \frac{s}{s^2 - a_{11}s - a_{12}} & \frac{a_{12}}{s^2 - a_{11}s - a_{12}} \\ \frac{1}{s^2 - a_{11}s - a_{12}} & \frac{s - a_{11}}{s^2 - a_{11}s - a_{12}} \end{bmatrix} \quad (2.170)$$

2.14.5 The Inverse Laplace Transform of the State Transition Matrix

Again, lets redefine some of our coefficients so that the mathematics is easier.

$$\begin{aligned} \omega_n^2 &= -a_{12} \\ 2\zeta\omega_n &= -a_{11} \end{aligned} \quad (2.171)$$

The state transition matrix is nearly complete except for the transition between the Laplace and time domains.

$$e^{At} = \mathcal{L}^{-1} \begin{bmatrix} \frac{s}{s^2 + 2\zeta\omega_n s + \omega_n^2} & \frac{-\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2} & \frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{bmatrix} \quad (2.172)$$

Expanding terms to allow the conversion between the Laplace and the time domain, we have

$$e^{At} = \mathcal{L}^{-1} \begin{bmatrix} \frac{s}{s^2 + 2\zeta\omega_n s + \omega_n^2} & (-1) \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ \left(\frac{1}{\omega_n^2} \right) \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} & \frac{s}{s^2 + 2\zeta\omega_n s + \omega_n^2} + \left(\frac{2\zeta}{\omega_n} \right) \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{bmatrix} \quad (2.173)$$

Notice that there are two distinct forms. From Laplace Transform tables we know the form of the solution as shown in Equation (2.174).

$$\mathcal{L}^{-1} \left(\frac{s}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right) = -\frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \left(\omega_n \sqrt{1-\zeta^2} t - \phi \right) \quad (2.174)$$

where

$$\phi = \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} \quad (2.175)$$

and

$$\mathcal{L}^{-1} \left(\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right) = \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \omega_n \sqrt{1-\zeta^2} t \quad (2.176)$$

The final conversion to the time domain is shown in Equation (2.177).

$$e^{At} = \begin{bmatrix} -\frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \left(\omega_n \sqrt{1-\zeta^2} t - \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} \right) & -\frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \omega_n \sqrt{1-\zeta^2} t \\ \left(\frac{1}{\omega_n^2} \right) \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \omega_n \sqrt{1-\zeta^2} t & -\frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \left(\omega_n \sqrt{1-\zeta^2} t - \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} \right) + \left(\frac{2\zeta}{\omega_n} \right) \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \omega_n \sqrt{1-\zeta^2} t \end{bmatrix} \quad (2.177)$$

2.14.6 The Inverse of the State Transition Matrix

We need to calculate the inverse of the state transition matrix in the time domain. Theory states that one property of the state transition matrix is $\Phi^{-1}(t) = \Phi(-t)$. Therefore, the inverse is calculated simply by reversing the sign on all of the t 's as shown in Equation (2.178).

$$e^{At} = \begin{bmatrix} -\frac{1}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n t} \sin \left(-\omega_n \sqrt{1-\zeta^2} t - \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} \right) & -\frac{\omega_n}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n t} \sin \left(-\omega_n \sqrt{1-\zeta^2} t \right) \\ \left(\frac{1}{\omega_n^2} \right) \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n t} \sin \left(-\omega_n \sqrt{1-\zeta^2} t \right) & -\frac{1}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n t} \sin \left(-\omega_n \sqrt{1-\zeta^2} t - \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} \right) + \left(\frac{2\zeta}{\omega_n} \right) \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n t} \sin \left(-\omega_n \sqrt{1-\zeta^2} t \right) \end{bmatrix} \quad (2.178)$$

2.14.7 The Integration of the Zero Order Hold

For a zero order hold we integrate from 0 to Δt , the size of the time step. The integration of the inverse state transition matrix is the most difficult operation.

$$\int_0^{\Delta t} e^{-A\tau} \mathbf{B} \mathbf{u}(\tau) d\tau \quad (2.179)$$

Equation (2.180) shows the multiplication of $e^{-A\tau} B$

$$\begin{bmatrix} -\frac{1}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n \tau} \sin\left(-\omega_n \sqrt{1-\zeta^2} \tau - \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta}\right) & -\frac{\omega_n}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n \tau} \sin(-\omega_n \sqrt{1-\zeta^2} \tau) \\ \left(\frac{1}{\omega_n^2}\right) \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n \tau} \sin(-\omega_n \sqrt{1-\zeta^2} \tau) & -\frac{1}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n \tau} \sin\left(-\omega_n \sqrt{1-\zeta^2} \tau - \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta}\right) - \left(\frac{2\zeta}{\omega_n}\right) \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n \tau} \sin(-\omega_n \sqrt{1-\zeta^2} \tau) \end{bmatrix} \begin{bmatrix} 0 & L_{\delta_a} k_2 \\ 0 & 0 \end{bmatrix} \quad (2.180)$$

Carrying out the matrix multiplication of $e^{-\Lambda\tau} \mathbf{B}$ leaves Equation (2.181).

$$\begin{bmatrix} 0 & -\frac{L_{\delta_a} k_2}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n \tau} \sin\left(-\omega_n \sqrt{1-\zeta^2} \tau - \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta}\right) \\ 0 & \left(\frac{L_{\delta_a} k_2}{\omega_n^2}\right) \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n \tau} \sin(-\omega_n \sqrt{1-\zeta^2} \tau) \end{bmatrix} \quad (2.181)$$

Next, we add the integration nomenclature and prepare to integrate from 0 to Δt . The $u(\tau)$ term is constant during a zero order hold and can be moved outside of the integration.

$$u \begin{bmatrix} 0 & \int_0^{\Delta t} -\frac{L_{\delta_a} k_2}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n \tau} \sin\left(-\omega_n \sqrt{1-\zeta^2} \tau - \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta}\right) d\tau \\ 0 & \int_0^{\Delta t} \left(\frac{L_{\delta_a} k_2}{\omega_n^2}\right) \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{\zeta\omega_n \tau} \sin(-\omega_n \sqrt{1-\zeta^2} \tau) d\tau \end{bmatrix} \quad (2.182)$$

2.14.8 Initial Preparation for Integration

We must integrate the two terms. Again, we choose to make some substitutions which make our work easier.

$$\begin{aligned} a &= \zeta\omega_n \\ b &= -\omega_n \sqrt{1-\zeta^2} \\ c &= -\tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} \\ K_1 &= \left(\frac{-L_{\delta_a} k_2}{\sqrt{1-\zeta^2}} \right) \end{aligned} \quad (2.183)$$

We can ‘simplify’ our 1st matrix term to

$$\int_0^{\Delta t} -K_1 e^{a\tau} \sin(b\tau + c) d\tau \quad (2.184)$$

We need to break the sine term down and split the integration.

$$\begin{aligned} \int_0^{\Delta t} -K_1 e^{a\tau} \sin(b\tau + c) d\tau = \\ \int_0^{\Delta t} -K_1 e^{a\tau} \sin(b\tau) \cos(c) d\tau + \int_0^{\Delta t} K_1 e^{a\tau} \cos(b\tau) \sin(c) d\tau \end{aligned} \quad (2.185)$$

Pulling out the constants leaves us with Equation (2.186).

$$\int_0^{\Delta t} -K_1 e^{a\tau} \sin(b\tau + c) d\tau = -K_1 \cos(c) \int_0^{\Delta t} e^{a\tau} \sin(b\tau) d\tau + K_1 \sin(c) \int_0^{\Delta t} e^{a\tau} \cos(b\tau) d\tau \quad (2.186)$$

The second matrix term is simplified as follows:

$$\int_0^{\Delta t} \left(\frac{-K_1}{\omega_n} \right) e^{a\tau} \sin(b\tau) d\tau \quad (2.187)$$

Removing the constants from the integration leaves Equation (2.188).

$$\int_0^{\Delta t} \left(\frac{-K_1}{\omega_n} \right) e^{a\tau} \sin(b\tau) d\tau = \left(\frac{-K_1}{\omega_n} \right) \int_0^{\Delta t} e^{a\tau} \sin(b\tau) d\tau \quad (2.188)$$

2.14.9 The Two Common Integrations

There are two common integrations which are required throughout the solution. Therefore we perform them here.

The sine integration, the integration of Equation (2.189), is the first.

$$\int_0^{\Delta t} e^{a\tau} \sin(b\tau) d\tau \quad (2.189)$$

$$= \frac{e^{a\tau} [a \sin(b\tau) - b \cos(b\tau)]}{a^2 + b^2} \Big|_0^{\Delta t} \quad (2.190)$$

$$= \frac{e^{a\Delta t} [a \sin(b\Delta t) - b \cos(b\Delta t)]}{a^2 + b^2} - \frac{e^{a0} [a \sin(b0) - b \cos(b0)]}{a^2 + b^2} \quad (2.191)$$

$$= \left(\frac{e^{a\Delta t} [a \sin(b\Delta t) - b \cos(b\Delta t)]}{a^2 + b^2} - \frac{-b}{a^2 + b^2} \right) \quad (2.192)$$

$$= \left(\frac{e^{a\Delta t} [a \sin(b\Delta t) - b \cos(b\Delta t)] + b}{a^2 + b^2} \right) \quad (2.193)$$

The cosine integration, the integration of Equation (2.194), is the second.

$$\int_0^{\Delta t} e^{a\tau} \cos(b\tau) d\tau \quad (2.194)$$

$$= \left(\frac{e^{a\tau}}{a^2 + b^2} [a \cos(b\tau) + b \sin(b\tau)] \Big|_0^{\Delta t} \right) \quad (2.195)$$

$$= \left(\frac{e^{a\Delta t}}{a^2 + b^2} [a \cos(b\Delta t) + b \sin(b\Delta t)] - \frac{e^{a0}}{a^2 + b^2} [a \cos(b0) + b \sin(b0)] \right) \quad (2.196)$$

$$= \left(\frac{e^{a\Delta t}}{a^2 + b^2} [a \cos(b\Delta t) + b \sin(b\Delta t)] - \frac{a}{a^2 + b^2} \right) \quad (2.197)$$

$$= \left(\frac{e^{a\Delta t} [a \cos(b\Delta t) + b \sin(b\Delta t)] - a}{a^2 + b^2} \right) \quad (2.198)$$

2.14.10 The Complete Integrated Terms

Next we substitute the integrated solutions into the matrix terms.

Matrix term #1:

$$\begin{aligned} & -K_1 \cos(c) \left(\frac{e^{a\Delta t} [a \sin(b\Delta t) - b \cos(b\Delta t)] + b}{a^2 + b^2} \right) \\ & + K_1 \sin(c) \left(\frac{e^{a\Delta t} [a \cos(b\Delta t) + b \sin(b\Delta t)] - a}{a^2 + b^2} \right) \end{aligned} \quad (2.199)$$

Matrix term #2:

$$\left(\frac{-K_1}{\omega_n} \right) \int_0^{\Delta t} e^{a\tau} \sin(b\tau) d\tau \quad (2.200)$$

2.14.11 Complete Integrated Portion of the Zero Order Hold

The complete integrated portion of the zero order hold can now be written.

$$\begin{aligned}
& \int_0^{\Delta t} e^{-A\tau} Bu(\tau) d\tau \\
&= \begin{bmatrix} K_1(2\zeta + \cos(c)) \left(\frac{e^{a\Delta t} [a \sin(b\Delta t) - b \cos(b\Delta t)] + b}{a^2 + b^2} \right) \\ + K_1 \sin(c) \left(\frac{e^{a\Delta t} [a \cos(b\Delta t) + b \sin(b\Delta t)] - a}{a^2 + b^2} \right) \\ - K_1 \omega_n \left(\frac{e^{a\Delta t} [a \sin(b\Delta t) - b \cos(b\Delta t)] + b}{a^2 + b^2} \right) \end{bmatrix} u(k)
\end{aligned} \tag{2.201}$$

2.14.12 The Complete Solution

The complete solution is shown below in Equation (2.202).

$$\begin{aligned}
\begin{bmatrix} p(k+1) \\ \phi(k+1) \end{bmatrix} &= \begin{bmatrix} -K_2 e^{-a\Delta t} \sin(-b\Delta t + c) & -K_2 \omega_n e^{-a\Delta t} \sin(-b\Delta t) \\ \frac{K_2}{\omega_n} e^{-a\Delta t} \sin(-b\Delta t) & -K_2 e^{-a\Delta t} \sin(-b\Delta t + c) \\ & + 2\zeta K_2 e^{-a\Delta t} \sin(-b\Delta t) \end{bmatrix} \begin{bmatrix} p(k) \\ \phi(k) \end{bmatrix} \\
&+ \begin{bmatrix} K_1(2\zeta + \cos(c)) \left(\frac{e^{a\Delta t} [a \sin(b\Delta t) - b \cos(b\Delta t)] + b}{a^2 + b^2} \right) \\ + K_1 \sin(c) \left(\frac{e^{a\Delta t} [a \cos(b\Delta t) + b \sin(b\Delta t)] - a}{a^2 + b^2} \right) \\ - K_1 \omega_n \left(\frac{e^{a\Delta t} [a \sin(b\Delta t) - b \cos(b\Delta t)] + b}{a^2 + b^2} \right) \end{bmatrix} \phi(k)
\end{aligned} \tag{2.202}$$

where

$$\begin{aligned}
a_{11} &= L_p - L_{\delta_a} k_1 & \omega_n^2 &= -a_{12} \\
a_{12} &= -L_{\delta_a} k_2 & 2\zeta \omega_n &= -a_{11}
\end{aligned}$$

$$\begin{aligned}
a &= \zeta \omega_n & K_1 &= \left(\frac{-L_{\delta_a} k_2}{\sqrt{1-\zeta^2}} \right) \\
b &= -\omega_n \sqrt{1-\zeta^2} \\
c &= -\tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} & K_2 &= \left(\frac{1}{\sqrt{1-\zeta^2}} \right)
\end{aligned}$$

It is important to note that the complete set of equations need only be calculated once for a given aircraft, set of feedback gains, and time steps size. Once the initial calculations are made, the relations used to actually update the state from one time step to another are simple. For instance, consider an aircraft with an open loop system as shown in Equation (2.203).

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.475 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} + \begin{bmatrix} 0.185 \\ 0 \end{bmatrix} \delta_a \quad (2.203)$$

Using feedback gains of $k_1 = 12.53$ and $k_2 = 21.62$ the resulting closed loop system is shown in Equation (2.204).

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -2.7930 & -3.9997 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} + \begin{bmatrix} 3.9997 \\ 0 \end{bmatrix} \phi_{des} \quad (2.204)$$

Discretizing the system for a 0.5 sec time step, using Equation (2.202), is shown in Equation (2.205).

$$\begin{bmatrix} p(k+1) \\ \phi(k+1) \end{bmatrix} = \begin{bmatrix} 0.0569 & -0.9120 \\ 0.2280 & 0.6938 \end{bmatrix} \begin{bmatrix} p(k) \\ \phi(k) \end{bmatrix} + \begin{bmatrix} 0.9120 \\ 0.3062 \end{bmatrix} \phi_{des}(k) \quad (2.205)$$

Equation (2.205) is the only calculation that must be made to update between time steps. Figure 2.27 shows the simulation results from Equation (2.205) using a 10 degree desired bank angle.

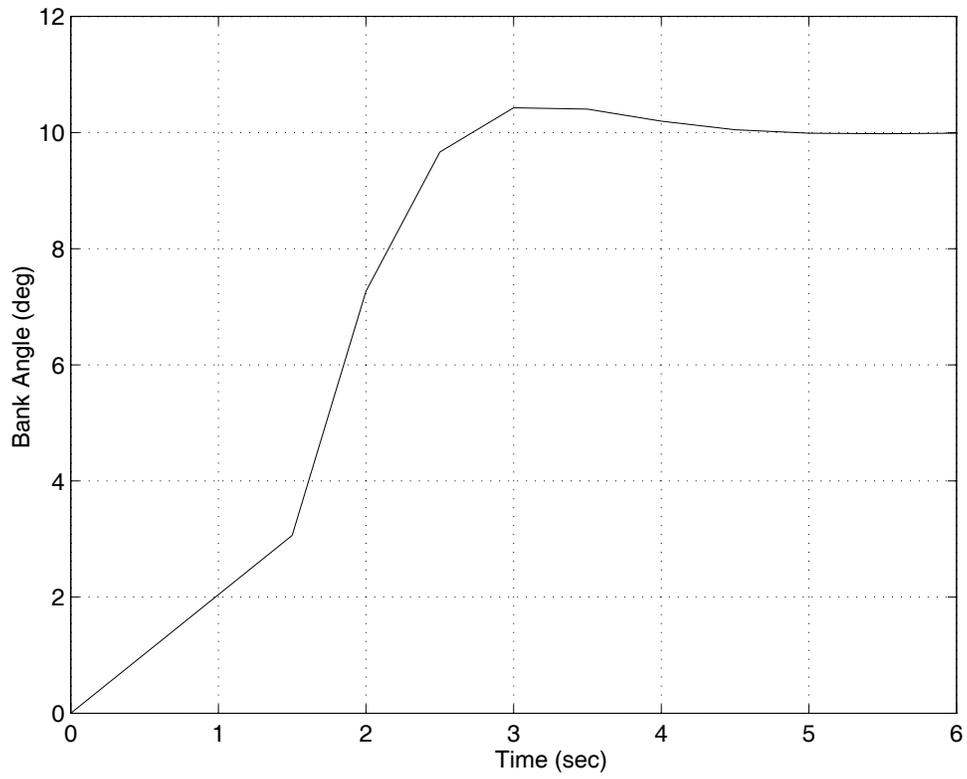


Figure 2.27. Roll mode response to a 10 degree desired bank angle

3. The Examination of the Longitudinal Dynamics

It can be argued that the majority of the effort put forth to build a successful feedback control system is spent trying to understand the plant which is to be controlled. This certainly is the case with the nonlinear longitudinal aircraft dynamics. The insight developed is a fundamental tool used to make intelligent decisions regarding feedback control strategy. Section 3 deals with the development of solid insight into the plant dynamics which in our case is the longitudinal dynamics of the aircraft. To develop insight several tasks are performed. These tasks are:

- Develop a linear model of the longitudinal dynamics
- Analysis of longitudinal modal properties
- Transfer function analysis of the longitudinal dynamics.

The linear model of the aircraft dynamics is the fundamental tool which allows for the modal analysis and the examination of the transfer functions. Therefore, it must be built first. Once the linear model is in place, the linear model is studied to determine what physical properties affect the modal properties of the system. Finally, certain transfer functions are created from the linear model that give insight into the different feedback control strategies that can be used.

3.1 The Linear Model of the Longitudinal Dynamics

The modeling equations for the aircraft dynamics, as presented in Table 2.3, are nonlinear with the exception of Equation (2.65). This nonlinearity limits our ability to perform an in depth study into the behavior of the system of equations and also precludes the design of a feedback control system. To overcome this limitation, a common approach in feedback control is to develop a linearized version of the system of equations.

In our linear modeling of the system of equations, we choose to separate the longitudinal dynamics from the lateral-directional dynamics. We can do this because the longitudinal modes and the lateral-directional modes are only lightly coupled [Nelson89]. For the longitudinal case, we constrain the aircraft to not turn.

In classical control theory, a linear, time-dependent (LTD) state-space is represented by the following system.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)\end{aligned}\tag{3.1}$$

In this system, \mathbf{x} is the system's (in our case, the aircraft's) state vector, \mathbf{u} is the system's control vector, \mathbf{y} is the system's output vector, and \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} are constant matrices. The state vector is a collection of variables that completely describes the system's state at

any given time. Our longitudinal state vector includes true airspeed, V_a , flight path angle, γ_a , and altitude, h . The control vector is the system's control inputs. Our longitudinal control vector includes lift coefficient, C_L and thrust, T . In discrete time, if the state and the control inputs at a given time-step are known, the state equation yields the state at the next time-step.

The output vector contains those parameters that are readily measurable by the controller. The aircraft (and, consequently, the aircraft's controller) may not know its true airspeed or its flight path angle, but it is able to measure its indicated airspeed, V_{IAS} , Mach number, M , altitude, h , and altitude rate, \dot{h} , and these parameters make up the output vector.

We wish to express the longitudinal dynamics as an LTD state-space, as in equation (3.1). This will facilitate an analysis of the modal properties. It will also facilitate the computation of feedback gains in Section 4.

Equations (2.59) and (2.64) are the focus of the longitudinal dynamics. These two equations characterize the Phugoid mode of the aircraft. To get the appropriate altitude information, we add the third term of Equation (2.72) to our system of equations as shown in Equation (3.2). Altitude rate, \dot{h} , and altitude, h , are both needed for the feedback control of the longitudinal dynamics. The equations for the longitudinal dynamics are repeated here.

$$\dot{V}_a = \frac{T - D - mg \sin \gamma_a}{m} \quad (2.59)$$

$$\dot{\gamma}_a = \frac{L \cos \phi - mg \cos \gamma_a}{mV_a} \quad (2.64)$$

$$\dot{h} = V_a \sin \gamma_a \quad (3.2)$$

Within these equations are the terms L and D , which are functions of the state variables. However, the terms L and D are not explicitly defined in terms of the state variables. To do this we need to know the aerodynamic characteristics.

Using the airframe equations of Section 2.10, we can express lift as shown in Equation (3.3). As discussed in Section 2.6, we know that the lift coefficient will be treated as a control input to the system, and will not be a function of the states.

$$L = \frac{1}{2} \rho V_a^2 S_w C_L \quad (3.3)$$

The aircraft drag can be similarly expressed.

$$D = \frac{1}{2} \rho V_a^2 S_w C_D \quad (3.4)$$

where the drag coefficient, C_D , of equation (2.143) is restated here.

$$C_D = (C_{D_0} + KC_L^2) \quad (2.143)$$

1. C_{D_0} is the zero lift drag coefficient.
2. K is the induced drag factor.

We need to express the total drag in terms of the drag coefficient.

$$D = \frac{1}{2} \rho V_a^2 S_w (C_{D_0} + KC_L^2) \quad (3.5)$$

These relations for lift and drag need to be substituted into the state equations. Equations (2.59) and (3.5) are combined to get the explicit state equation for true airspeed.

$$\dot{V}_a = \frac{T - \frac{1}{2} \rho V_a^2 S_w (C_{D_0} + KC_L^2) - mg \sin \gamma_a}{m} \quad (3.6)$$

Equations (2.73) and (3.3) are combined to get the explicit flight path angle equation.

$$\dot{\gamma}_a = \frac{\frac{1}{2} \rho V_a^2 S_w C_L \cos \phi - mg \cos \gamma_a}{m V_a} \quad (3.7)$$

We can represent Equations (3.6), (3.7), and (3.2) simply as functions of the state and control variables.

$$\dot{V}_a = f_{V_a}(V_a, \gamma_a, h, C_L, T)$$

$$\dot{\gamma}_a = f_{\gamma_a}(V_a, \gamma_a, h, C_L, T)$$

$$\dot{h} = f_h(V_a, \gamma_a, h, C_L, T)$$

This representation is convenient for representing the partial derivatives of the state equations. The symbolically represented, linearized perturbation equations are shown below.

$$\dot{V}_a|_o + \Delta \dot{V}_a = f_{V_a}|_o + \frac{\partial f_{V_a}}{\partial V_a}|_o \Delta V_a + \frac{\partial f_{V_a}}{\partial \gamma_a}|_o \Delta \gamma_a + \frac{\partial f_{V_a}}{\partial h}|_o \Delta h + \frac{\partial f_{V_a}}{\partial T}|_o \Delta T + \frac{\partial f_{V_a}}{\partial C_L}|_o \Delta C_L$$

$$\dot{\gamma}_a|_o + \Delta\dot{\gamma}_a = f_{\gamma_a}|_o + \frac{\partial f_{\gamma_a}}{\partial V_a}\bigg|_o \Delta V_a + \frac{\partial f_{\gamma_a}}{\partial \gamma_a}\bigg|_o \Delta\gamma_a + \frac{\partial f_{\gamma_a}}{\partial h}\bigg|_o \Delta h + \frac{\partial f_{\gamma_a}}{\partial T}\bigg|_o \Delta T + \frac{\partial f_{\gamma_a}}{\partial C_L}\bigg|_o \Delta C_L$$

$$\dot{h}|_o + \Delta\dot{h} = f_h|_o + \frac{\partial f_h}{\partial V_a}\bigg|_o \Delta V_a + \frac{\partial f_h}{\partial \gamma_a}\bigg|_o \Delta\gamma_a + \frac{\partial f_h}{\partial h}\bigg|_o \Delta h + \frac{\partial f_h}{\partial T}\bigg|_o \Delta T + \frac{\partial f_h}{\partial C_L}\bigg|_o \Delta C_L$$

Similarly, the parameters of the output vector are repeated here.

$$V_{IAS} = a_0^* \sqrt{\frac{2}{\gamma-1} \left\{ \frac{p}{p_o} \left[\left(1 + \frac{\gamma-1}{2} M^2 \right)^{\frac{\gamma}{\gamma-1}} - 1 \right] + 1 \right\}^{\frac{\gamma-1}{\gamma}} - 1} \quad (2.134)$$

$$M = \frac{V_a}{a^*} \quad (2.136)$$

$$\mathbf{h} = \mathbf{h}$$

$$\dot{h} = V_a \sin \gamma_a \quad (3.2)$$

Once again, we can represent these equations as functions of the state and control variables.

$$V_{IAS} = g_{V_{IAS}}(V_a, \gamma_a, h, C_L, T)$$

$$M = g_M(V_a, \gamma_a, h, C_L, T)$$

$$h = g_h(V_a, \gamma_a, h, C_L, T)$$

$$\dot{h} = g_{\dot{h}}(V_a, \gamma_a, h, C_L, T) = f_{\dot{h}}(V_a, \gamma_a, h, C_L, T)$$

The linearized perturbation equations of the output vector are,

$$V_{IAS}|_o + \Delta V_{IAS} = g_{V_{IAS}}|_o + \frac{\partial g_{V_{IAS}}}{\partial V_a}\bigg|_o \Delta V_a + \frac{\partial g_{V_{IAS}}}{\partial \gamma_a}\bigg|_o \Delta\gamma_a + \frac{\partial g_{V_{IAS}}}{\partial h}\bigg|_o \Delta h + \frac{\partial g_{V_{IAS}}}{\partial T}\bigg|_o \Delta T + \frac{\partial g_{V_{IAS}}}{\partial C_L}\bigg|_o \Delta C_L$$

$$M|_o + \Delta M = g_M|_o + \frac{\partial g_M}{\partial V_a}\bigg|_o \Delta V_a + \frac{\partial g_M}{\partial \gamma_a}\bigg|_o \Delta\gamma_a + \frac{\partial g_M}{\partial h}\bigg|_o \Delta h + \frac{\partial g_M}{\partial T}\bigg|_o \Delta T + \frac{\partial g_M}{\partial C_L}\bigg|_o \Delta C_L$$

$$h|_o + \Delta h = g_h|_o + \frac{\partial g_h}{\partial V_a} \Big|_o \Delta V_a + \frac{\partial g_h}{\partial \gamma_a} \Big|_o \Delta \gamma_a + \frac{\partial g_h}{\partial h} \Big|_o \Delta h + \frac{\partial g_h}{\partial T} \Big|_o \Delta T + \frac{\partial g_h}{\partial C_L} \Big|_o \Delta C_L$$

$$\dot{h}|_o + \Delta \dot{h} = f_h|_o + \frac{\partial f_h}{\partial V_a} \Big|_o \Delta V_a + \frac{\partial f_h}{\partial \gamma_a} \Big|_o \Delta \gamma_a + \frac{\partial f_h}{\partial h} \Big|_o \Delta h + \frac{\partial f_h}{\partial T} \Big|_o \Delta T + \frac{\partial f_h}{\partial C_L} \Big|_o \Delta C_L$$

A partial derivative of a system equation is called a stability derivative if it is with respect to a state variable and a control derivative if it is with respect to a control variable. The stability and control derivatives for the longitudinal model are organized in tabular form and presented in Table 3.1, Table 3.2, and Table 3.3. It is important to note that these derivatives are derived specifically for the 4 DOF model that we have constructed. These derivatives are similar, but not interchangeable with the classic stability and control derivatives of the full 6 DOF equations of motion such as those found in Nelson [N89] or Stevens [SL92].

By inspection, we see that the partial derivatives of $g_{V_{IAS}}$, g_M , and g_h are all zero except for $\frac{\partial g_{V_{IAS}}}{\partial V_a}$, $\frac{\partial g_M}{\partial V_a}$ and $\frac{\partial g_h}{\partial h}$. From equations (2.134) and (2.136),

$$\frac{\partial g_M}{\partial V_a} = \frac{\partial}{\partial V_a} \left(\frac{V_a}{a^*} \right) = \frac{1}{a^*} \quad (3.8)$$

$$\begin{aligned} \frac{\partial g_{V_{IAS}}}{\partial V_a} &= \frac{\partial}{\partial V_a} \left(a_0^* \sqrt{\frac{2}{\gamma-1} \left(\left\{ \frac{p}{p_o} \left[\left(1 + \frac{\gamma-1}{2} M^2 \right)^{\frac{\gamma}{\gamma-1}} - 1 \right] + 1 \right\}^{\frac{\gamma-1}{\gamma}} - 1 \right)} \right) \\ \frac{\partial g_{V_{IAS}}}{\partial V_a} &= \frac{\frac{a_0^*}{a^*} \frac{p}{p_o} M \left(1 + \frac{\gamma-1}{2} M^2 \right)^{\frac{1}{\gamma-1}} \left\{ \frac{p}{p_o} \left[\left(1 + \frac{\gamma-1}{2} M^2 \right)^{\frac{\gamma}{\gamma-1}} - 1 \right] + 1 \right\}^{\frac{-1}{\gamma}}}{\sqrt{\frac{2}{\gamma-1} \left(\left\{ \frac{p}{p_o} \left[\left(1 + \frac{\gamma-1}{2} M^2 \right)^{\frac{\gamma}{\gamma-1}} - 1 \right] + 1 \right\}^{\frac{\gamma-1}{\gamma}} - 1 \right)}} \end{aligned} \quad (3.9)$$

$$\frac{\partial g_h}{\partial h} = 1$$

Table 3.1. Stability and control derivatives of f_{V_a}

<i>State/Input</i>	<i>Derivative of f_{V_a}</i>
V_a	$\frac{\partial f_{V_a}}{\partial V_a} = \frac{-\rho V_a S_w (C_{D_0} + K C_L^2)}{m}$ (3.10)
γ_a	$\frac{\partial f_{V_a}}{\partial \gamma_a} = -g \cos \gamma_a$ (3.11)
h	$\frac{\partial f_{V_a}}{\partial h} = 0$ (3.12)
C_L	$\frac{\partial f_{V_a}}{\partial C_L} = \frac{-\rho V_a^2 S_w K C_L}{m}$ (3.13)
T	$\frac{\partial f_{V_a}}{\partial T} = \frac{1}{m}$ (3.14)

Table 3.2. Stability and control derivatives of f_{γ_a}

<i>State/Input</i>	<i>Derivative of f_{γ_a}</i>
V_a	$\frac{\partial f_{\gamma_a}}{\partial V_a} = \frac{\rho S_{ref} C_L}{2m} + \frac{g}{V_a^2} \cos \gamma_a$ (3.15)
γ_a	$\frac{\partial f_{\gamma_a}}{\partial \gamma_a} = \frac{g}{V_a} \sin \gamma_a$ (3.16)
h	$\frac{\partial f_{\gamma_a}}{\partial h} = 0$ (3.17)
C_L	$\frac{\partial f_{\gamma_a}}{\partial C_L} = \frac{\rho V_a S_{ref}}{2m}$ (3.18)
T	$\frac{\partial f_{\gamma_a}}{\partial T} = 0$ (3.19)

Table 3.3. Stability and control derivatives of f_h

<i>State/Input</i>	<i>Derivative of f_h</i>
V_a	$\frac{\partial f_h}{\partial V_a} = \sin \gamma_a$ (3.20)
γ_a	$\frac{\partial f_h}{\partial \gamma_a} = V_a \cos \gamma_a$ (3.21)
h	$\frac{\partial f_h}{\partial h} = 0$ (3.22)

$$C_L \quad \left| \quad \frac{\mathcal{F}_h}{\mathcal{C}_L} = 0 \quad (3.23)$$

$$T \quad \left| \quad \frac{\mathcal{F}_h}{\partial T} = 0 \quad (3.24)$$

The LTD state-space representation of the linearized longitudinal dynamics (per equation (3.1)) is shown in equations (3.25) and (3.26).

$$\begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{\mathcal{F}_{V_a}}{\mathcal{N}_a} & \frac{\mathcal{F}_{V_a}}{\partial \gamma_a} & \frac{\mathcal{F}_{V_a}}{\partial h} \\ \frac{\mathcal{F}_{\gamma_a}}{\mathcal{N}_a} & \frac{\mathcal{F}_{\gamma_a}}{\partial \gamma_a} & \frac{\mathcal{F}_{\gamma_a}}{\partial h} \\ \frac{\mathcal{F}_h}{\mathcal{N}_a} & \frac{\mathcal{F}_h}{\partial \gamma_a} & \frac{\mathcal{F}_h}{\partial h} \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \end{bmatrix} + \begin{bmatrix} \frac{\mathcal{F}_{V_a}}{\mathcal{C}_L} & \frac{\mathcal{F}_{V_a}}{\partial T} \\ \frac{\mathcal{F}_{\gamma_a}}{\mathcal{C}_L} & \frac{\mathcal{F}_{\gamma_a}}{\partial T} \\ \frac{\mathcal{F}_h}{\mathcal{C}_L} & \frac{\mathcal{F}_h}{\partial T} \end{bmatrix} \begin{bmatrix} \Delta C_L \\ \Delta T \end{bmatrix} \quad (3.25)$$

$$\begin{bmatrix} \Delta V_{IAS} \\ \Delta M \\ \Delta h \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{\partial V_{IAS}}{\partial V_a} & 0 & 0 \\ \frac{\partial M}{\partial V_a} & 0 & 0 \\ 0 & 0 & 1 \\ \frac{\partial f_h}{\partial V_a} & \frac{\partial f_h}{\partial \gamma_a} & \frac{\partial f_h}{\partial h} \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{\partial f_h}{\partial C_L} & \frac{\partial f_h}{\partial T} \end{bmatrix} \begin{bmatrix} \Delta C_L \\ \Delta T \end{bmatrix} \quad (3.26)$$

All of the derivatives in these constant matrices are evaluated at the reference condition; the notation indicating that explicitly has been removed. Numerous derivative are zero for all cases. These derivatives are:

$$\frac{\mathcal{F}_{V_a}}{\partial h} = \frac{\mathcal{F}_{\gamma_a}}{\partial T} = \frac{\mathcal{F}_{\gamma_a}}{\partial h} = \frac{\mathcal{F}_h}{\partial h} = \frac{\mathcal{F}_h}{\partial C_L} = \frac{\mathcal{F}_h}{\partial T} = 0$$

Furthermore, if we assume that the aircraft's reference condition is level flight ($\gamma_a = 0$), we can set other derivatives to zero.

$$\frac{\mathcal{F}_{\gamma_a}}{\partial \gamma_a} = \frac{\mathcal{F}_h}{\partial V_a} = 0$$

Modifying our state equations results in Equations (3.27) and (3.28). These equations represent the final form of the linearized model.

$$\begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_{V_a}}{\partial V_a} & \frac{\partial f_{V_a}}{\partial \gamma_a} & 0 \\ \frac{\partial f_{\gamma_a}}{\partial V_a} & 0 & 0 \\ 0 & \frac{\partial f_h}{\partial \gamma_a} & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \end{bmatrix} + \begin{bmatrix} \frac{\partial f_{V_a}}{\partial C_L} & \frac{\partial f_{V_a}}{\partial T} \\ \frac{\partial f_{\gamma_a}}{\partial C_L} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta C_L \\ \Delta T \end{bmatrix} \quad (3.27)$$

$$\begin{bmatrix} \Delta V_{LAS} \\ \Delta M \\ \Delta h \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{\partial V_{LAS}}{\partial V_a} & 0 & 0 \\ \frac{\partial M}{\partial V_a} & 0 & 0 \\ 0 & 0 & 1 \\ 0 & \left(\frac{\partial f_h}{\partial \gamma_a} \right) & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta C_L \\ \Delta T \end{bmatrix} \quad (3.28)$$

It is useful to compare the simulation results of the linear model to the results from the nonlinear model. The expected result is that the linear model will agree with the nonlinear model for very small perturbations from the reference condition. As the perturbations from the equilibrium condition become larger, the linear model will not follow the nonlinear dynamics. This behavior is seen in Figure 3.1 and Figure 3.2. Both figures show the time histories of the three longitudinal states along with the altitude rate as calculated by the linear and nonlinear models. Figure 3.1 shows the models' response to a small perturbation or change in the nominal or reference lift coefficient. As can be seen from the time histories of Figure 3.1, the match between the two models is good. However, as the perturbation or change in the nominal or reference lift coefficient becomes larger, the linear model fails to reflect accurately the behavior of the nonlinear dynamics.

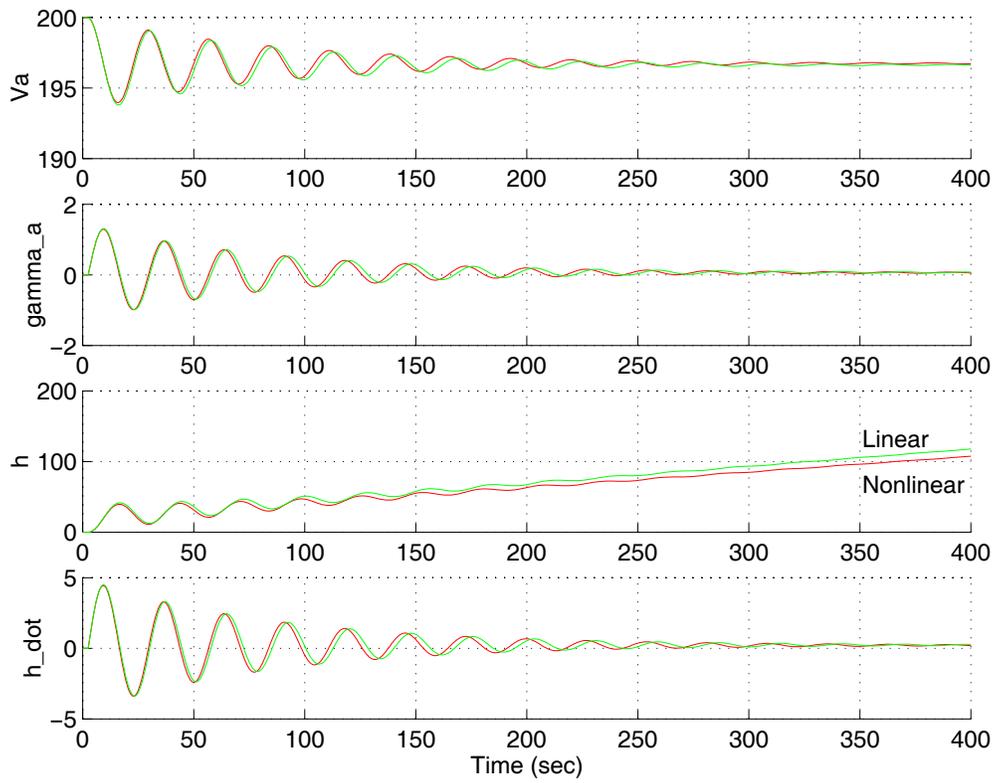


Figure 3.1. Comparison of linear and nonlinear models with a 0.01 perturbation from the reference lift coefficient

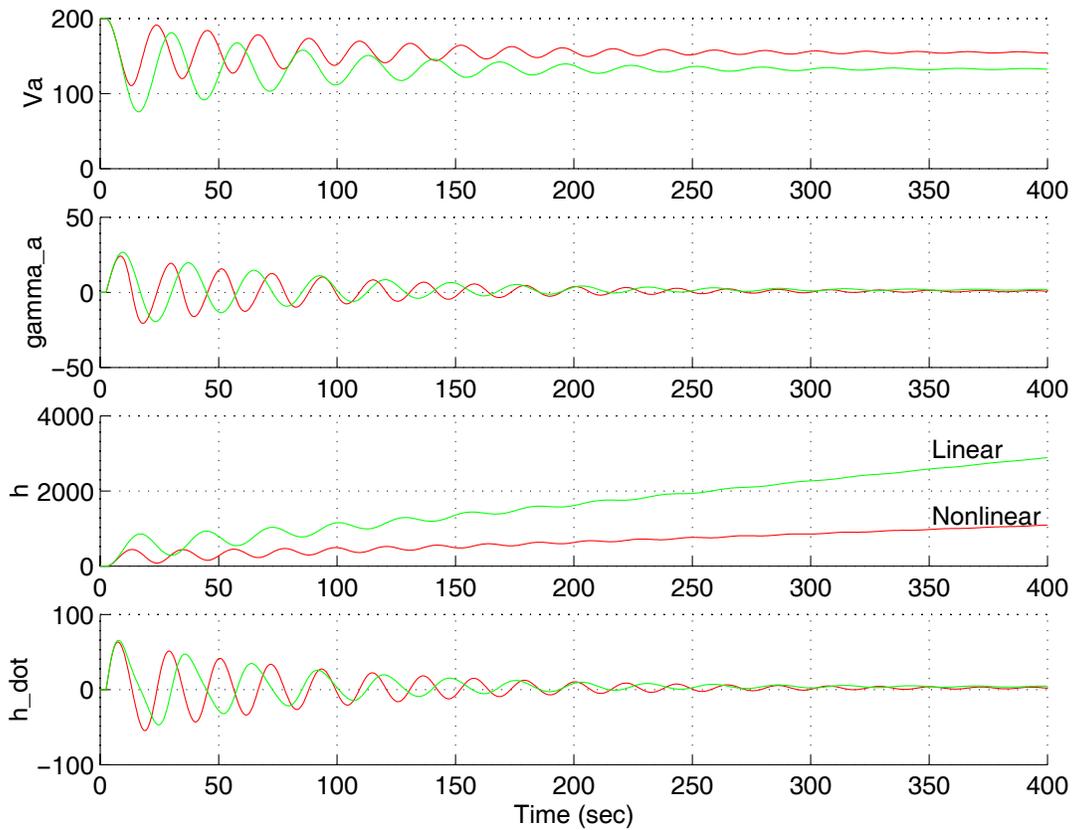


Figure 3.2. Comparison of linear and nonlinear models with a 0.2 perturbation from the reference lift coefficient

This is the limitation of using linear models. To account for this limitation, many linear models, all referenced about different reference conditions, are used to accurately model the aircraft's performance throughout the entire flight envelope.

3.2 The Analysis of Longitudinal Aircraft Modal Properties

The main limitation of linear models is that they are valid only for a limited range around the reference conditions which were used to create them. To model an entire flight envelope of an aircraft, many linear models, each having its own set of reference conditions, must be developed. Immediately, one can then see the advantage to having a linear model which is a function of as few reference values as possible. For instance, if the aircraft's linear model varied only with true airspeed, it would make for a simple one dimensional set of linear models, each with a different true airspeed reference. However, we can see through observation that the linear model of the longitudinal dynamics for a given aircraft is fundamentally a function of three varying parameters. These are:

- V_a : The aircraft's true airspeed

- ρ : The air density
- m : The mass of the aircraft

The fact that there are three terms immediately presents an inconvenience. Any set of linear models must be 3 dimensional. For instance, even a modest number of variations, say 10 true airspeeds, 10 masses, and 10 different air densities, would yield 1000 reference conditions and hence 1000 linear models. It is therefore very desirable to eliminate a varying parameter if possible. Elimination of a varying parameter is the attempt of this section.

Consider the state space representation of the system as shown in Equation (3.27). This system of equations contains three state equations, the first two of which, the ΔV_a equation and the $\Delta \gamma_a$ equation, characterize the Phugoid longitudinal mode. The third state equation, the Δh equation, only contributes to the calculation of altitude and does not affect the Phugoid mode.

$$\begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{\partial V_a}{\partial V_a} & \frac{\partial V_a}{\partial \gamma_a} & 0 \\ \frac{\partial \gamma_a}{\partial V_a} & 0 & 0 \\ 0 & \frac{\partial h}{\partial \gamma_a} & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \end{bmatrix} + \begin{bmatrix} \frac{\partial V_a}{\partial C_L} & \frac{\partial V_a}{\partial T} \\ \frac{\partial \gamma_a}{\partial C_L} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta C_L \\ \Delta T \end{bmatrix} \quad (3.27)$$

Because the Δh equation does not contribute to the Phugoid dynamics, we choose to ignore it in the following analysis. Ignoring the Δh equation reduces the state equations to Equation (3.29).

$$\begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \end{bmatrix} = \begin{bmatrix} \frac{\partial V_a}{\partial V_a} & \frac{\partial V_a}{\partial \gamma_a} \\ \frac{\partial \gamma_a}{\partial V_a} & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \end{bmatrix} + \begin{bmatrix} \frac{\partial V_a}{\partial C_L} & \frac{\partial V_a}{\partial T} \\ \frac{\partial \gamma_a}{\partial C_L} & 0 \end{bmatrix} \begin{bmatrix} \Delta C_L \\ \Delta T \end{bmatrix} \quad (3.29)$$

We can derive the characteristic equation for the A matrix of Equation (3.29) in terms of the derivatives using $\det(sI - A)$ as shown in Equations (3.30) and (3.31).

$$s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{\partial V_a}{\partial V_a} & \frac{\partial V_a}{\partial \gamma_a} \\ \frac{\partial \gamma_a}{\partial V_a} & 0 \end{bmatrix} = \begin{bmatrix} s - \frac{\partial V_a}{\partial V_a} & -\frac{\partial V_a}{\partial \gamma_a} \\ -\frac{\partial \gamma_a}{\partial V_a} & s \end{bmatrix} \quad (3.30)$$

$$s^2 - \frac{\partial^2 V_a}{\partial \mathcal{N}_a} s - \frac{\partial^2 V_a}{\partial \gamma_a} \frac{\partial \gamma_a}{\partial \mathcal{N}_a} \quad (3.31)$$

When we substitute for the actual derivatives we see that the characteristic equation expands to Equation (3.32).

$$s^2 + \frac{\rho V_a S_w (C_{D_0} + K C_L^2)}{m} \left(1 + \frac{9 C_{M_{16}}}{a^{*16}} V_a^{16} \right) s + g \cos \gamma_a \left(\frac{\rho S_{ref} C_L}{2m} + \frac{g}{V_a^2} \cos \gamma_a \right) \quad (3.32)$$

If we neglect the effects of compressibility, we can simplify Equation (3.32) to Equation (3.33).

$$s^2 + \frac{\rho V_a S_w (C_{D_0} + K C_L^2)}{m} s + g \cos \gamma_a \left(\frac{\rho S_{ref} C_L}{2m} + \frac{g}{V_a^2} \cos \gamma_a \right) \quad (3.33)$$

We know from classical control theory that the natural frequency and damping ratio are represented in the characteristic equation as $s^2 + 2\zeta\omega_n s + \omega_n^2$. Therefore we can assign the last term (s^0) to equal the square of the Phugoid frequency, ω_p .

$$\omega_p = \sqrt{g \cos \gamma_a \left(\frac{\rho S_{ref} C_L}{2m} + \frac{g}{V_a^2} \cos \gamma_a \right)} \quad (3.34)$$

We can gain insight from this equation. First, if the flight path angle is small, the relation can be reduced to Equation (3.35).

$$\omega_p = \sqrt{\left(\frac{\rho g S_{ref} C_L}{2m} + \frac{g^2}{V_a^2} \right)} \quad (3.35)$$

The ratio $\frac{g^2}{V_a^2}$ is likely to dominate this term at low speeds and be small at high speeds. It is a function of true airspeed squared which is proportional to dynamic pressure. Consider the other term. It is inversely proportional to the mass. This suggests that as the mass goes down, the frequency goes up. This is true providing that the lift coefficient does not change. However, it is likely that the lift coefficient will change with mass because the pilot will always tend to trim the aircraft for a given flight condition. If we assume that lift equals weight or is close to equaling weight for the vast number of flight conditions we can write the relation for mass as seen in Equation (3.36).

$$q S_w C_L = mg \quad (3.36)$$

Using (3.36) we can substitute for C_L in Equation (3.35) resulting the Phugoid expression in Equation (3.37).

$$\omega_p = \sqrt{\left(\frac{\rho g S_{ref}}{2m} \left(\frac{mg}{qS_w}\right) + \frac{g^2}{V_a^2}\right)} \quad (3.37)$$

Canceling terms leaves (3.38).

$$\omega_p = \sqrt{\left(\frac{\rho g^2}{2q} + \frac{g^2}{V_a^2}\right)} \quad (3.38)$$

Finally, noting that the dynamic pressure is a function of density and true airspeed,

$$\omega_p = \sqrt{\left(\frac{\rho g^2}{2\frac{\rho V_a^2}{2}} + \frac{g^2}{V_a^2}\right)} \quad (3.39)$$

we can see that the Phugoid frequency for the trimmed aircraft is entirely a function of true airspeed as shown in Equation (3.40).

$$\omega_p = \sqrt{\left(\frac{2g^2}{V_a^2}\right)} = \sqrt{2} \frac{g}{V_a} \quad (3.40)$$

This is an interesting result because it suggests that the frequency of the Phugoid is not a function of the weight of the aircraft or the altitude at which the aircraft is flying.

Moving to the damping ratio of the Phugoid mode, we can express the damping ratio using the middle (s^1) term of the characteristic equation if we divide by $2\omega_p$ as shown in Equation (3.41).

$$\zeta_p = \sqrt{\left(\frac{V_a^2}{2g^2}\right)} \frac{\rho V_a S_w (C_{D_0} + KC_L^2)}{2m} \quad (3.41)$$

Unfortunately, there is no simplification that reduces the damping ratio to a single function of any parameter that we have so far defined. This expression for damping implies that the only way to schedule gains to control the Phugoid is to have 3 dimensional tables consisting of aircraft weight, true airspeed and altitude (air density). This creates a large computational burden and requires the storage of many scheduled feedback gains. It is desirable to somehow reduce the schedule to a 2 dimensional table. One solution is to substitute dynamic pressure for the density and true airspeed. This

substitution effectively assumes that changes in speed and altitude (density) can be interchangeable. However, we can see that density and speed work independently of each other. While it may be acceptable to schedule vs dynamic pressure, it is only an approximation. It is better if another quantity can be found.

Working towards a simplified expression for damping we revisit Equation (3.41). Assuming a trimmed aircraft, we can substitute Equation (3.36) into Equation (3.42) for weight, mg . After some algebraic manipulation, the final result is shown in Equation (3.43).

$$\begin{aligned}\zeta_p &= \frac{V_a}{\sqrt{2}g} \frac{\rho V_a S_w (C_{D_0} + KC_L^2)}{2m} \\ \zeta_p &= \frac{\rho V_a^2}{2} \frac{S_w (C_{D_0} + KC_L^2)}{\sqrt{2}mg} \\ \zeta_p &= \frac{q S_w (C_{D_0} + KC_L^2)}{\sqrt{2} q S_w C_L} \\ \zeta_p &= \frac{(C_{D_0} + KC_L^2)}{\sqrt{2} C_L} \\ \zeta_p &= \frac{1}{\sqrt{2}(\frac{L}{D})}\end{aligned}\tag{3.43}$$

We see that the damping is a function of the lift to drag ratio of the aircraft. An alternative derivation is contained in Nelson [N89], which comes to the same basic conclusion. Unfortunately, it is impossible for the control logic or any sophisticated instrument to actually measure the lift to drag ratio of the aircraft. However, the lift to drag ratio is always a function of the lift coefficient at a given time. So, while the lift to drag ratio can not be known, we can approximately measure the trim lift coefficient at any given time in the flight and know that the lift coefficient corresponds to a particular location on the drag polar and hence a particular L/D. Therefore, the conclusion is that the modal properties for the aircraft are uniquely described by the trim lift coefficient and the true airspeed. However this does not consider the control derivatives.

There are three control derivatives to be considered as shown in Equations (3.44) through (3.46).

$$\frac{\partial \dot{\gamma}_a}{\partial C_L} = \frac{\rho V_a S_{ref}}{2m}\tag{3.44}$$

$$\frac{\partial f_{V_a}}{\partial C_L} = \frac{-\rho V_a^2 S_w K C_L}{m}\tag{3.45}$$

$$\frac{\partial f_{V_a}}{\partial T} = \frac{1}{m} \quad (3.46)$$

Consider the expression for the lift coefficient in Equation (3.47). If the lift coefficient and the true airspeed are known, it is possible to solve for the ratio between the air density and the aircraft's mass. This can be seen in Equation (3.48).

$$C_L = \frac{2mg}{\rho S_w V_a^2} \quad (3.47)$$

$$\frac{\rho}{m} = \frac{2g}{C_L S_w V_a^2} \quad (3.48)$$

While it is not possible to solve for the mass and density directly, we can solve for the ratio, which is then all that is needed to define two of the control derivatives as shown in Equations (3.49) and (3.50).

$$\frac{\partial f_{\gamma_a}}{\partial C_L} = \left(\frac{\rho}{m} \right) \frac{V_a S_{ref}}{2} \quad (3.49)$$

$$\frac{\partial f_{V_a}}{\partial C_L} = -V_a^2 S_w K C_L \left(\frac{\rho}{m} \right) \quad (3.50)$$

The final control derivative, the $\frac{\partial f_{V_a}}{\partial T}$ derivative, is obviously a function of mass only. Therefore, it is not completely possible to define the whole system using two parameters. However, having only one term which is a function of mass is far more convenient than having the mass term throughout the model. All gain calculations can be done using a nominal mass and varying true airspeed and trim lift coefficient. To accommodate different aircraft masses only one term need be changed.

THIS PAGE INTENTIONALLY LEFT BLANK

4. The Feedback Control System for Longitudinal Control

In the previous Chapters, the bulk of the analysis effort was spent on the derivation of the physical model of the aircraft. The physical model of the aircraft consisted of the dynamic equations, which model the aircraft's performance, and the kinematic equations, which characterize the aircraft's propagation over the surface of the Earth. The purpose of building such an intricate model is to insure the fidelity of the model for modeling actual aircraft in flight. The main advantage of a high fidelity aircraft model is that it accurately models the performance and handling characteristics of an aircraft.

Accordingly, the disadvantage of a high fidelity aircraft model is that it accurately models the performance and handling characteristics of the aircraft. To make the aircraft model follow a desired trajectory, the aircraft model must be 'flown' by a pilot in the same sense that the actual aircraft must be flown. Arguably, the longitudinal control system is the most complicated and sensitive part of the entire simulation. This Chapter is the first of three Chapters which covers the longitudinal control system.

The purpose of the longitudinal control system is to provide a means of automating two fundamental aircraft maneuvers. These maneuvers are:

- Altitude change and altitude capture
- Speed change and speed capture

Generally, the functionality of the longitudinal control system can be divided into two distinct classes of algorithms. These two classes are feedback control and supporting functional logic. This Chapter deals with the design of the feedback control algorithms which actually stabilize the aircraft and drive it to the desired state. There are different feedback control algorithms for different flight phases, and each of these is discussed along with a strategy for calculating the required gains.

4.1 The General Control Law

The general control law for the longitudinal dynamics is the same regardless of which region the aircraft is operating. The only real difference between regions from the feedback control standpoint is what gains are used. The general control law framework allows for any output variable to be fed back to any input variable.

The general framework for the longitudinal control law is shown in block diagram form in Figure 4.1. The terms in the block diagram are defined as follows:

- \mathbf{y}_d is the desired output vector
- \mathbf{y} is the actual output vector
- \mathbf{e} is the error vector equivalent to $\mathbf{y}_d - \mathbf{y}$
- \mathbf{K}_p is the proportional gain matrix in the feed-forward path.
- \mathbf{K}_i is the integral gain matrix (also in the feed-forward path)
- \mathbf{K}_b is the proportional gain matrix in the feedback path

- $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ is the linearized state equation for the longitudinal dynamics
- $\mathbf{y} = \mathbf{Cx} + \mathbf{Du}$ is the output equation for the longitudinal dynamics

From experience in dealing with the longitudinal dynamics, we have seen that proportional control using the appropriate output error, e , is sufficient to achieve the dynamic response desired. Integral control is then added to eliminate steady-state error. The final feedback loop, the one using \mathbf{K}_b , is designed to allow proportional feedback control of certain output variables without affecting the zeros of the transfer functions. Gains in the feedback path affect only the modal properties of a system; gains in the feedforward path affect the dynamics of the system while at the same time driving the state error to zero. In certain instances it is necessary to make use of the stabilization offered by feeding back a particular output while not driving that output to any particular value. The \mathbf{K}_b feedback path was added as a direct result of the analysis done in Chapter 2, where non-minimum phase behavior was observed.

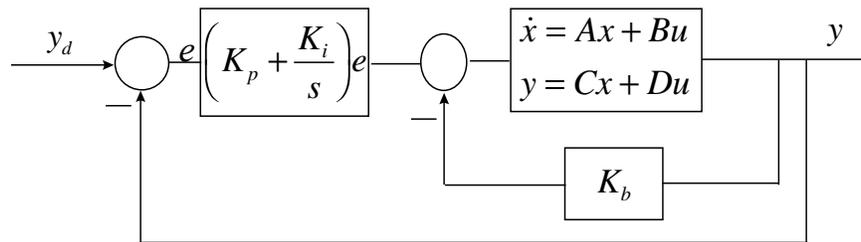


Figure 4.1. Block diagram for the longitudinal control law

The general form of a LTD state-space is restated here.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{Ax}(t) + \mathbf{Bu}(t) \\ \mathbf{y}(t) &= \mathbf{Cx}(t) + \mathbf{Du}(t)\end{aligned}\tag{4.1}$$

Our general control law is defined in the time-domain as,

$$\mathbf{u}(t) = \mathbf{K}_p \mathbf{e}(t) + \int_0^t \mathbf{K}_i \mathbf{e}(t) dt - \mathbf{K}_b \mathbf{y}(t)\tag{4.2}$$

The gain matrices, \mathbf{K}_p , \mathbf{K}_i , and \mathbf{K}_b , are the proportional, integral, and feedback gain matrices. Because the general control law allows for any output variable to be fed back to any input variable, a gain matrix is of dimension $(n \times l)$, and is of the form,

$$\mathbf{K}_p = \begin{bmatrix} k_{p_{11}} & k_{p_{12}} & \cdots & k_{p_{1,l}} \\ k_{p_{21}} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ k_{p_{n,1}} & \cdots & \cdots & k_{p_{n,l}} \end{bmatrix} \quad (4.3)$$

where, n is the number of control variables in the control vector, l is the number of output variables in the output vector, and the subscript, p , refers to the proportional gain matrix. The form the integral and feedback gain matrices is the same except that subscripts i and b are substituted for the subscript p .

4.2 Manipulating an LTD State-Space with Integral Control

Integral control adds system poles to our state-space. In effect, it changes the order of our LTD system. This can be seen explicitly if we modify our LTD system to include the differential equations added by our integral controller. Let's define proportional and integral control vectors as,

$$\begin{aligned} \mathbf{u}_p(t) &= \mathbf{K}_p \mathbf{e}(t) - \mathbf{K}_b \mathbf{y}(t) \\ \mathbf{u}_i(t) &= \mathbf{K}_i \mathbf{e}(t) \end{aligned} \quad (4.4)$$

and let's define the integrated control as,

$$\mathbf{I}_u(t) = \int_0^t \mathbf{u}_i(t) dt$$

Then the control vector can be written in terms of its separated proportional and integral terms.

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{I}_u(t)$$

Clearly, the derivative of the integrated control vector, \mathbf{I}_u , is the integral control vector, \mathbf{u}_i .

$$\dot{\mathbf{I}}_u(t) = \mathbf{u}_i(t) \quad (4.5)$$

The state equation can be rewritten in terms of the separated control vector.

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}[\mathbf{u}_p(t) + \mathbf{I}_u(t)] \\ \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}_p(t) + \mathbf{B}\mathbf{I}_u(t) \end{aligned} \quad (4.6)$$

Combining equations (4.5) and (4.6) and the output equation into one LTD system gives,

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{I}}_u(t) \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0}_{n,m} & \mathbf{0}_n \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{I}_u(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} & \mathbf{0}_{m,n} \\ \mathbf{0}_n & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{u}_p(t) \\ \mathbf{u}_i(t) \end{bmatrix} \\ \mathbf{y}(t) &= \begin{bmatrix} \mathbf{C} & \mathbf{0}_{l,n} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{I}_u(t) \end{bmatrix} + \begin{bmatrix} \mathbf{D} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{u}_p(t) \\ \mathbf{u}_i(t) \end{bmatrix} \end{aligned} \quad (4.7)$$

From equation (4.4), the control law can be written as,

$$\begin{bmatrix} \mathbf{u}_p(t) \\ \mathbf{u}_i(t) \end{bmatrix} = \begin{bmatrix} \mathbf{K}_p \\ \mathbf{K}_i \end{bmatrix} \mathbf{e}(t) - \begin{bmatrix} \mathbf{K}_b \\ \mathbf{0}_{n,m} \end{bmatrix} \mathbf{y}(t) \quad (4.8)$$

- m is the number of state variables in the state vector (3 in our current example),
- n is the number of control variables in the control vector (2),
- l is the number of output variables in the output vector (4),
- $\mathbf{0}_n$ is a square, $n \times n$ matrix of zeros, and
- $\mathbf{0}_{n,m}$ is a $n \times m$ matrix of zeros.

For the purposes of analyzing the modal properties of our linearized system, consider that the state, control, and output vectors are simply perturbations from a reference condition. Without affecting the system's characteristic equation, it can be assumed that our desired result is to return to that reference condition. Then the output vector, \mathbf{y}_d , is a vector of zeros and the error vector, \mathbf{e} , becomes the negative of the output vector, \mathbf{y} .

$$\mathbf{e} = \mathbf{y}_d - \mathbf{y} = (\mathbf{0}_l) - \mathbf{y} = -\mathbf{y}$$

Then the control law, defined by equation (4.4), can be written as,

$$\begin{bmatrix} \mathbf{u}_p(t) \\ \mathbf{u}_i(t) \end{bmatrix} = - \begin{bmatrix} \mathbf{K}_p + \mathbf{K}_b \\ \mathbf{K}_i \end{bmatrix} \mathbf{y}(t)$$

Since, in this linear approximation, there is no way to differentiate between the effects of proportional gain in the feed-forward path, \mathbf{K}_b , and proportional gain in the feedback path, \mathbf{K}_p , there is no use in considering both and so \mathbf{K}_b is neglected.

$$\begin{bmatrix} \mathbf{u}_p(t) \\ \mathbf{u}_i(t) \end{bmatrix} = - \begin{bmatrix} \mathbf{K}_p \\ \mathbf{K}_i \end{bmatrix} \mathbf{y}(t) \quad (4.9)$$

Equations (4.7) still follow the format of an LTD system: the new A-matrix is square, the new state and control are still time-dependent row vectors, and the new A-, B-, C-, and D-matrices are constant. Once combined with the standard output-based feedback

control law of equation (4.9), we have a complete linear, time-dependent (LTD) feedback control system conducive to modal analysis. It is a convenient form that adds the integral terms (and their resultant poles) to the state vector to facilitate the analysis of modal properties and calculation of gains.

4.3 An Analysis of the Effects of Feedback Control on the Modal Properties

Until this point, the analysis of this chapter is generalized to any LTD system with a control law defined by equation (4.2), but we now begin to tailor the analysis to our model. When we combine our LTD system of equations (4.7) with our original state, control, and output vectors and our A-, B-, C- and D-matrices as defined in Chapter 3, we get,

$$\begin{aligned}
 \begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \\ \Delta \dot{h} \\ \dot{I}_{C_L} \\ \dot{I}_T \end{bmatrix} &= \begin{bmatrix} \frac{\partial f_{V_a}}{\partial V_a} & \frac{\partial f_{V_a}}{\partial \gamma_a} & 0 \\ \frac{\partial f_{\gamma_a}}{\partial V_a} & 0 & 0 \\ 0 & \frac{\partial f_h}{\partial \gamma_a} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial f_{V_a}}{\partial C_L} & \frac{\partial f_{V_a}}{\partial T} \\ \frac{\partial f_{\gamma_a}}{\partial C_L} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \\ I_{C_L} \\ I_T \end{bmatrix} \\
 &+ \begin{bmatrix} \frac{\partial f_{V_a}}{\partial C_L} & \frac{\partial f_{V_a}}{\partial T} \\ \frac{\partial f_{\gamma_a}}{\partial C_L} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{p_{C_L}} \\ u_{p_T} \\ u_{i_{C_L}} \\ u_{i_T} \end{bmatrix}
 \end{aligned}$$

$$\begin{bmatrix} \Delta \dot{V}_{IAS} \\ \Delta M \\ \Delta h \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{\partial V_{IAS}}{\partial V_a} & 0 & 0 \\ \frac{\partial M}{\partial V_a} & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 60 \left(\frac{\partial f_h}{\partial \gamma_a} \right) & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \\ I_{C_L} \\ I_T \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{p_{C_L}} \\ u_{p_T} \\ u_{i_{C_L}} \\ u_{i_T} \end{bmatrix}$$

And the general form of our LTD system is,

$$\begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \\ \Delta \dot{h} \\ \dot{I}_{C_L} \\ \dot{I}_T \end{bmatrix} = \begin{bmatrix} \frac{\partial f_{V_a}}{\partial V_a} & \frac{\partial f_{V_a}}{\partial \gamma_a} & 0 & \frac{\partial f_{V_a}}{\partial C_L} & \frac{\partial f_{V_a}}{\partial T} \\ \frac{\partial f_{\gamma_a}}{\partial V_a} & 0 & 0 & \frac{\partial f_{\gamma_a}}{\partial C_L} & 0 \\ 0 & \frac{\partial f_h}{\partial \gamma_a} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \\ I_{C_L} \\ I_T \end{bmatrix} + \begin{bmatrix} \frac{\partial f_{V_a}}{\partial C_L} & \frac{\partial f_{V_a}}{\partial T} & 0 & 0 \\ \frac{\partial f_{\gamma_a}}{\partial C_L} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{p_{C_L}} \\ u_{p_T} \\ u_{i_{C_L}} \\ u_{i_T} \end{bmatrix}$$

$$\begin{bmatrix} \Delta V_{IAS} \\ \Delta M \\ \Delta h \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{\partial V_{IAS}}{\partial V_a} & 0 & 0 & 0 & 0 \\ \frac{\partial M}{\partial V_a} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 60 \left(\frac{\partial f_h}{\partial \gamma_a} \right) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \\ I_{C_L} \\ I_T \end{bmatrix} \tag{4.10}^1$$

Following the form of the general gain matrix of equation (4.3), our proportional, integral, and feedback gain matrices are,

¹ Note that the \dot{h} equation has added a factor of 60 to its $\frac{\partial f_h}{\partial \gamma_a}$ term. This is to convert the output from \dot{h}/sec to \dot{h}/min as would be read on a real vertical speed indicator.

$$\mathbf{K}_p = \begin{bmatrix} k_{p11} & k_{p12} & k_{p13} & k_{p14} \\ k_{p21} & k_{p22} & k_{p23} & k_{p24} \end{bmatrix} \quad (4.11)$$

$$\mathbf{K}_i = \begin{bmatrix} k_{i11} & k_{i12} & k_{i13} & k_{i14} \\ k_{i21} & k_{i22} & k_{i23} & k_{i24} \end{bmatrix} \quad (4.12)$$

$$\mathbf{K}_b = \begin{bmatrix} k_{b11} & k_{b12} & k_{b13} & k_{b14} \\ k_{b21} & k_{b22} & k_{b23} & k_{b24} \end{bmatrix} \quad (4.13)$$

Substituting into equation (4.9), we have our control law.

$$\begin{bmatrix} u_{p_{CL}} \\ u_{p_T} \\ u_{i_{CL}} \\ u_{i_T} \end{bmatrix} = \begin{bmatrix} k_{p11} & k_{p12} & k_{p13} & k_{p14} \\ k_{p21} & k_{p22} & k_{p23} & k_{p24} \\ k_{i11} & k_{i12} & k_{i13} & k_{i14} \\ k_{i21} & k_{i22} & k_{i23} & k_{i24} \end{bmatrix} \begin{bmatrix} \Delta V_{LAS} \\ \Delta M \\ \Delta h \\ \Delta \dot{h} \end{bmatrix} \quad (4.14)$$

There are 16 gains in our gain matrix, each representing the effect of feedback of a particular output parameter to a particular control parameter. To gain insight into the effect each feedback gain is likely to have, we examine the root locus of each gain. Consider a Boeing 767-300 at a reference weight of 198,000 lbs in steady, level flight at 300 knots indicated airspeed (KIAS) and 30,000 feet. For this example, the LTD system of equations (4.10) becomes,

$$\begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \\ \Delta \dot{h} \\ \dot{I}_{CL} \\ \dot{I}_T \end{bmatrix} = \begin{bmatrix} -0.0889 & -32.2 & 0 & -21.7 & 0.000163 \\ 0.000366 & 0 & 0 & 0.173 & 0 \\ 0 & 787 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \\ I_{CL} \\ I_T \end{bmatrix} + \begin{bmatrix} -21.7 & 0.000163 & 0 & 0 \\ 0.173 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{p_{CL}} \\ u_{p_T} \\ u_{i_{CL}} \\ u_{i_T} \end{bmatrix}$$

$$\begin{bmatrix} \Delta V_{LAS} \\ \Delta M \\ \Delta h \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} 0.593 & 0 & 0 & 0 & 0 \\ 0.00100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 47200 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \\ I_{CL} \\ I_T \end{bmatrix}$$

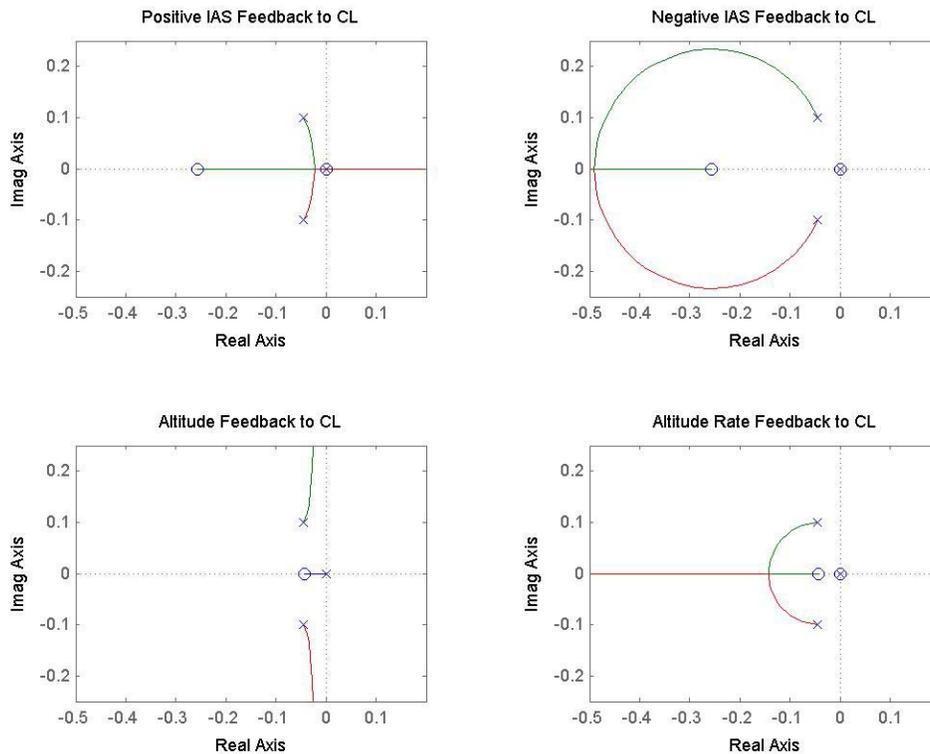


Figure 4.2. Effects of proportional feedback to the lift coefficient

Figure 4.2 - Figure 4.5 show how the phugoid mode poles are moved by various types of feedback. Figure 4.2 illustrates the system's behavior with different proportional feedback to the lift coefficient. The first subplot shows the positive feedback of indicated airspeed to the lift coefficient, which we see is unstable. A moment's reflection on the nature of the system provides intuitive verification. An increase in the lift coefficient results in higher drag, which serves to slow the aircraft. This implies that lowering the lift coefficient would serve to increase speed. The second subplot verifies this and shows negative speed feedback to the lift coefficient provides for a stable control.

The third subplot shows the effect of altitude feedback to lift coefficient which we see tends to shoot the phugoid poles up along the imaginary axis. This will increase the phugoid natural frequency while reducing the damping of the system.

The fourth subplot shows the effect of altitude rate feedback to lift coefficient. Here we see a well behaved loop. Feeding back the altitude rate tends to dampen the system. This effect can be intuitively verified by remembering that rate terms usually do increase the damping of a system.

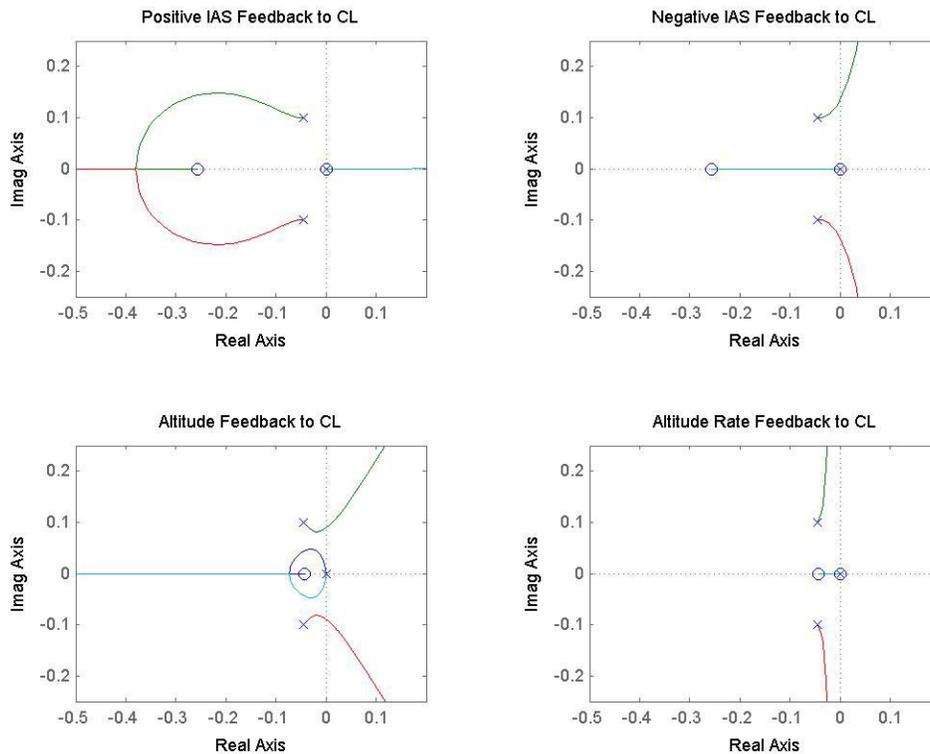


Figure 4.3. Effects of integral feedback to the lift coefficient

Figure 4.3 shows the effects of integral feedback to the lift coefficient. The second subplot shows a problem with instabilities of integral speed feedback to the lift coefficient. If we choose to use negative integral speed feedback to the lift coefficient, we will need to feed some output in the feedback path to keep it stable. The fourth subplot shows that we need to be careful with low damping of integral altitude rate feedback to the lift coefficient.

Figure 4.4 shows the effect of proportional feedback to the thrust. The first subplot shows positive feedback of indicated airspeed to the thrust. The locus is well behaved and tends to increase the damping of the system. Positive feedback of indicated airspeed or Mach to the thrust is a good choice for the control of speed. The fourth subplot shows altitude rate feedback to the thrust. The altitude rate here tends to have the same effect on the Phugoid poles that altitude feedback had to the lift coefficient. Altitude rate feedback to the thrust is not a good choice for controlling altitude rate or altitude. Similarly, the third subplot shows that altitude feedback to the thrust drives the system unstable

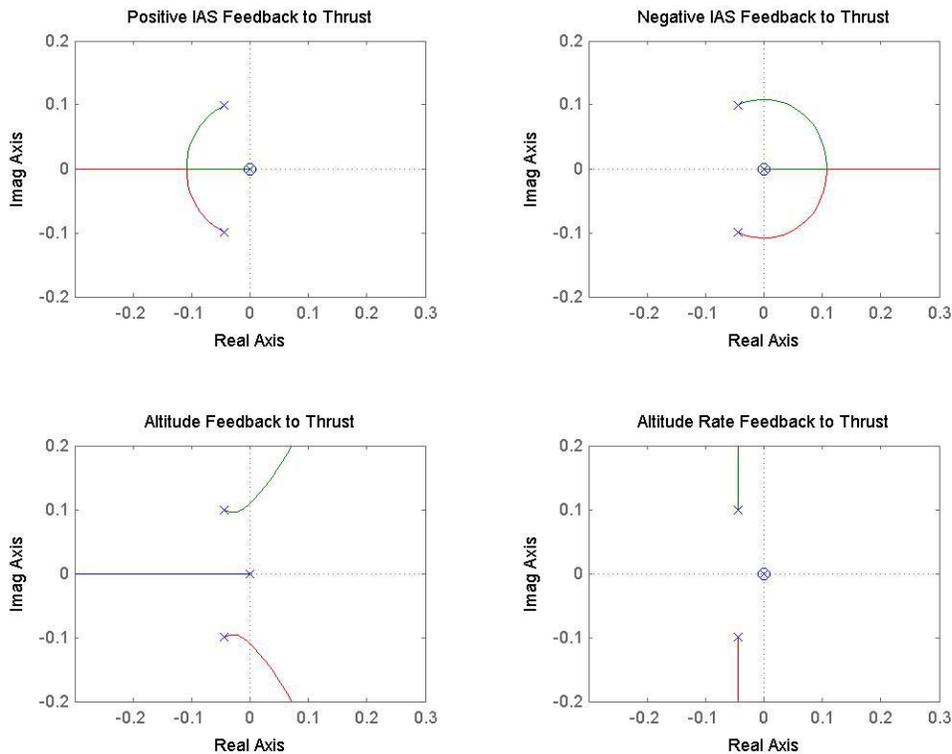


Figure 4.4 Effects of Proportional Feedback to the Thrust

Figure 4.5 shows the effects of integral feedback to the thrust. The third and fourth subplots confirm that it is not wise to control altitude or altitude rate with thrust because the system becomes unstable as soon as we apply control. The first subplot shows that integral feedback of speed to the thrust tends to shoot the phugoid poles up the imaginary axis, increasing the natural frequency and lowering the damping of the response. This implies that we need to be mindful of low damping when controlling speed with the thrust.

4.4 Feedback Controller Design

In the previous section, we analyzed the effects of feedback control on the modal properties of the LTD system in order to gain insight into how feedback control would affect our nonlinear, four-degree-of-freedom model. In this section, we continue to use the LTD system to design feedback controllers for several different regions of an aircraft's flight regime. The LTD system provides a suitable approximation to the nonlinear system so that feedback controllers designed using the LTD system will produce similar results in our nonlinear model. Each controller is customized to attain the desired flight configuration using smooth transitions that are typical of commercial aircraft flight.

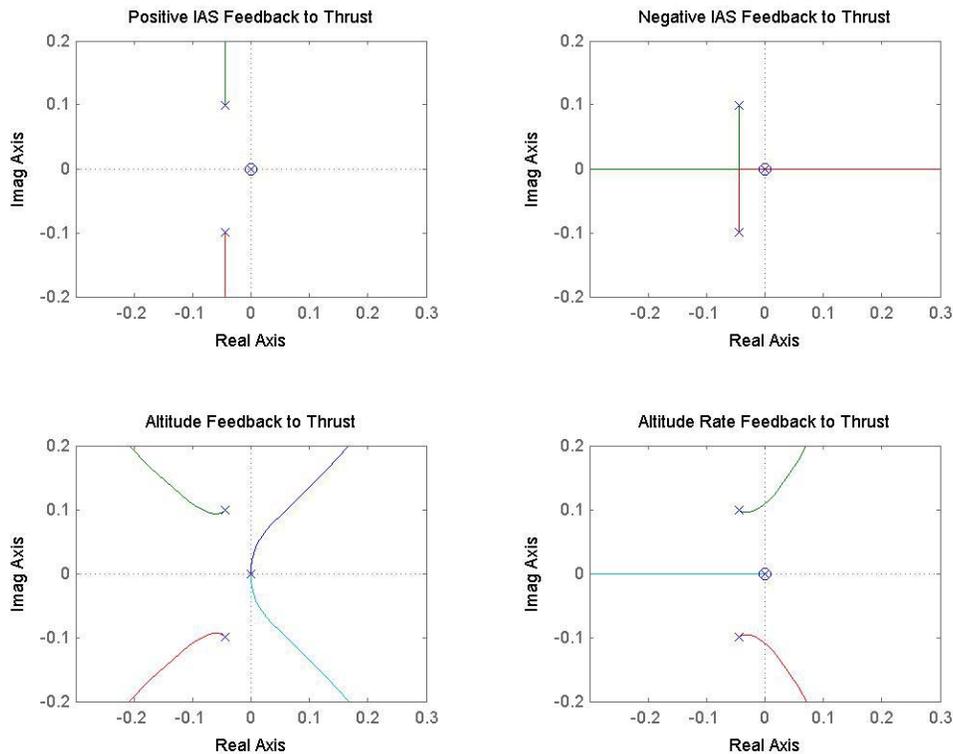


Figure 4.5 Effects of Integral Feedback to the Thrust

Let us summarize our conclusions from the analysis of the root loci of the LTD system. The primary control input in an aircraft is the control stick (i.e., the lift coefficient), so it should be the primary control in our controller as well. Figure 4.2 and Figure 4.3 show that we can successfully control speed with the lift coefficient, as long as we use negative feedback, and as long as we feed output (e.g., altitude rate) in the feedback path to keep the controller stable. Figure 4.2 and Figure 4.3 also show that we can successfully control altitude rate with the lift coefficient. Figure 4.4 and Figure 4.5 show that we can successfully control speed with the throttle (i.e., thrust). Our basic strategy will be to ...

- use lift coefficient to control altitude rate during speed changes;
- use lift coefficient to control speed during altitude changes, while feeding back altitude rate to keep the system stable; and
- use thrust to control speed and lift coefficient to control altitude rate when we need to control both.

Our control law allows us to feedback all outputs to both inputs; although only a fraction of the gains are used in a given feedback system. To design our controllers, we will use our full control law, as defined by equation (4.8) and expanded here using the gain matrix definitions of equations (4.11), (4.12), and (4.13).

$$\begin{bmatrix} u_{P_{C_L}} \\ u_{P_T} \\ u_{i_{C_L}} \\ u_{i_T} \end{bmatrix} = \begin{bmatrix} k_{p_{11}} & k_{p_{12}} & k_{p_{13}} & k_{p_{14}} \\ k_{p_{21}} & k_{p_{22}} & k_{p_{23}} & k_{p_{24}} \\ k_{i_{11}} & k_{i_{12}} & k_{i_{13}} & k_{i_{14}} \\ k_{i_{21}} & k_{i_{22}} & k_{i_{23}} & k_{i_{24}} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} - \begin{bmatrix} k_{b_{11}} & k_{b_{12}} & k_{b_{13}} & k_{b_{14}} \\ k_{b_{21}} & k_{b_{22}} & k_{b_{23}} & k_{b_{24}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_{IAS} \\ M \\ h \\ \dot{h} \end{bmatrix} \quad (4.15)$$

The control inputs to the system split into their proportional and integrated parts. The control inputs $u_{P_{C_L}}$ and u_{P_T} are the proportional portions of the lift coefficient and thrust respectively, and $u_{i_{C_L}}$ and u_{i_T} are the integrated portions of lift coefficient and thrust.

4.4.1 Lift Coefficient Control of Altitude Rate

There are several flight regimes in which a pilot uses the control stick to capture or maintain altitude rate while allowing the speed to change. During level flight accelerations and decelerations, a pilot will preset the thrust (maximum thrust for acceleration, idle thrust for deceleration) and allow the speed to change accordingly while using the control stick to maintain level flight (i.e., a zero altitude rate). Alternatively, the pilot may wish to capture and maintain a desired, non-zero altitude rate and let the speed change as it may (i.e., during accelerations in climb or decelerations in descent).

For these regions, we need a controller that uses lift coefficient to control altitude rate but does not modulate thrust. In the truest sense, the controller is not controlling speed; however, in actuality the speed is controlled because we preset the thrust according to the desired direction of the speed change. When the speed nears the desired steady, level flight condition, we switch to a controller that simultaneously controls speed and altitude rate.

To accomplish the goals of this controller, we need simply to command an altitude rate for all time. For this reason we can simplify the output equation of the LTD system of equation (4.10) to,

$$\Delta \dot{h} = \begin{bmatrix} 0 & 60 \left(\frac{\partial f_h}{\partial \gamma_a} \right) & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \end{bmatrix}$$

In doing so, we can see that the dynamics of the system are governed exclusively by the \dot{h}/C_L transfer function as defined in Chapter 3. Recall that since we control only the lift coefficient, the system is non-minimum phase at slow speeds; however, there is no need for a separate control law for the non-minimum phase system. Chapter 5 will show how we relieve the problems of the non-minimum phase system by ensuring that the thrust is sufficient to counter the changing drag.

Consider a DC-9 traveling at 15000 ft and 578 ft/sec and weighing 140,000lbs. The LTD system is shown below with its reduced control and output vectors. The Phugoid eigenvalues for the open loop system are located at $-0.0036 \pm 0.0786i$ which corresponds to a natural frequency of 0.0787 rad/sec and a damping ratio of 0.0445. A plot of the step response of the system to a 0.1 change in lift coefficient is shown in Figure 4.6.

$$\begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \\ \Delta \dot{h} \\ \dot{I}_{C_L} \end{bmatrix} = \begin{bmatrix} -0.0072 & -32.2 & 0 & -2.74 \\ 0.0002 & 0 & 0 & 0.1202 \\ 0 & 578.47 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \\ I_{C_L} \end{bmatrix} + \begin{bmatrix} -2.74 & 0 \\ 0.1202 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{P_{CL}} \\ u_{i_{CL}} \end{bmatrix} \quad (4.16)$$

$$\Delta \dot{h} = \begin{bmatrix} 0 & 578.47 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \end{bmatrix}$$

It is desired to increase the frequency and damping of the system while having zero steady state error. We start with proportional altitude rate feedback to the lift coefficient, $k_{p_{14}}$, shown in the left side of Figure 4.7. A value of $k_{p_{14}} = 2.6 \times 10^{-5}$ moves the Phugoid poles to location of $-0.0578 \pm 0.0562i$ with a damping ratio of 0.7173 and a natural frequency of 0.0806rad/sec. Since the frequency is rather low, we can increase it by use of integral control. Note that the purpose of integral control is to guarantee zero steady state error, not frequency changes; in this particular case, integral control can increase the system frequency. As seen on the right half of Figure 4.7, we move the Phugoid poles with

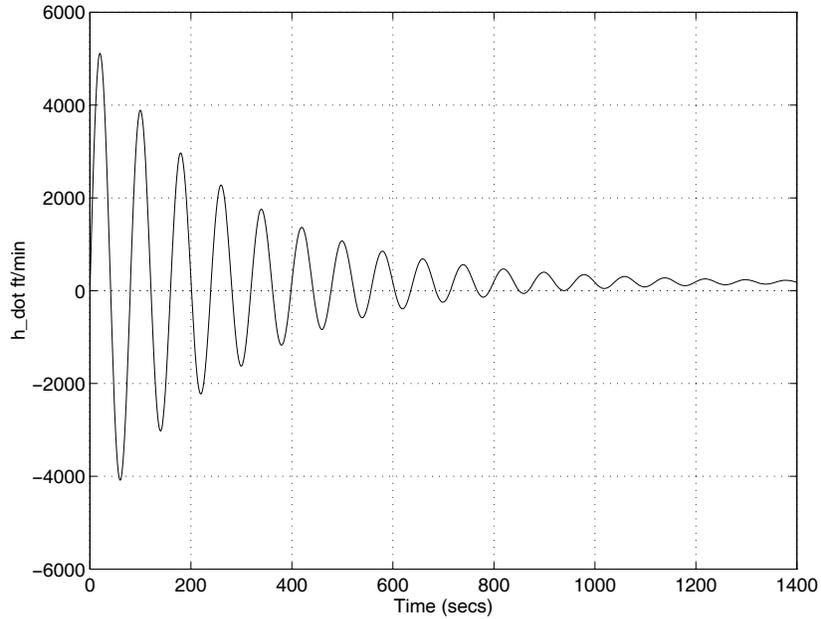


Figure 4.6. System response to a 0.1 step in lift coefficient

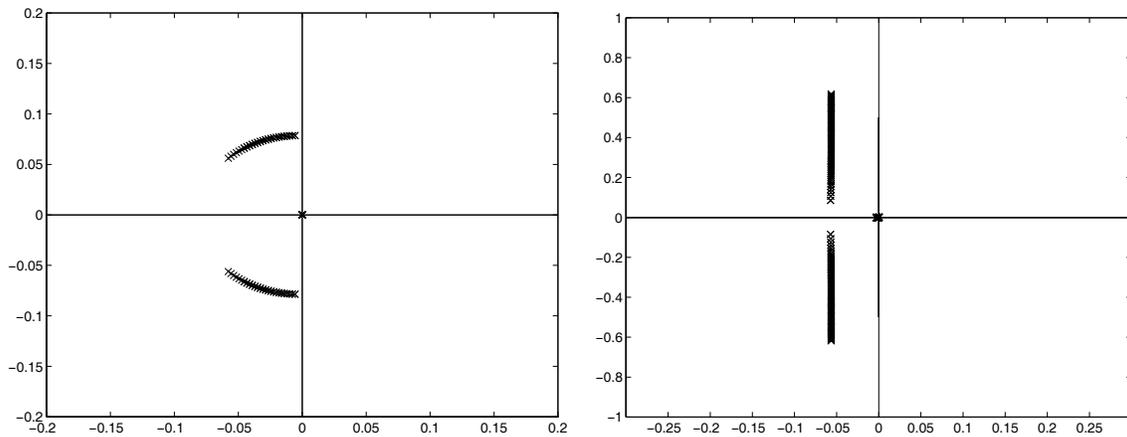


Figure 4.7. Root Loci of $k_{p_{14}}$ (left) and $k_{i_{14}}$ (right) successive loop closures

$k_{i_{14}} = 9.0 \times 10^{-5}$ to a location of $-0.0565 \pm 0.6152i$. In doing so, we increase the frequency of the mode to 0.6178 rad/sec but we reduce the damping to 0.0914. To correct for the low damping we can increase the gain $k_{p_{14}}$ once again and move the Phugoid poles to the left as shown in Figure 4.8.

The final values chosen for $k_{p_{14}}$ and $k_{i_{14}}$ are 2.08×10^{-4} and 9.0×10^{-5} . The final step response is shown in Figure 4.9. Notice that the system achieves the 1000 ft/min climb rate with zero steady state error. Furthermore, the time history for the lift coefficient is well within acceptable bounds. Note however that the lift coefficient initially is rather aggressive. In the next chapter, we define limits to the amount a control input can change in a time-step so that such aggressive control action is made more realistic.

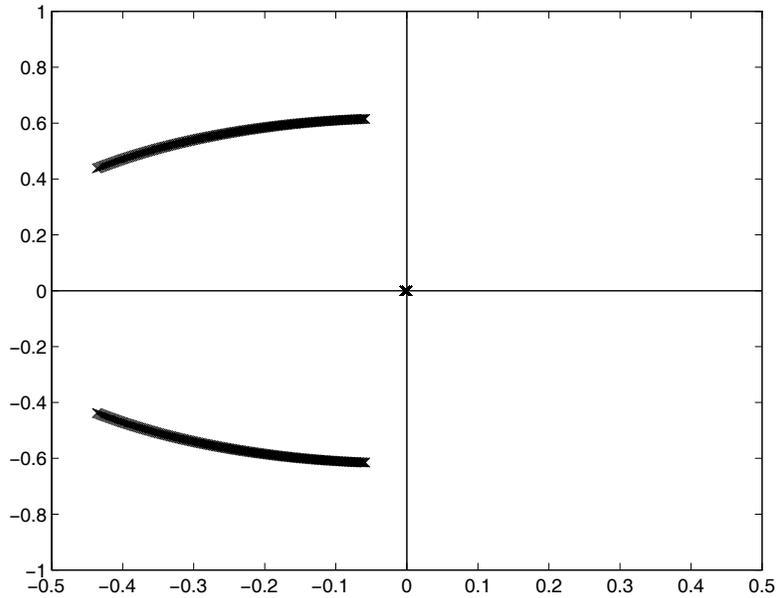


Figure 4.8. The effects of an increased proportional gain k_{p14}

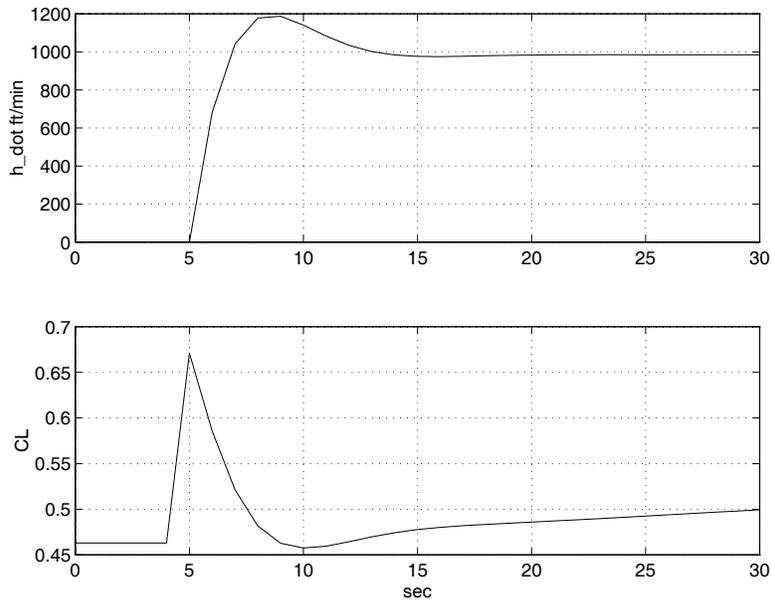


Figure 4.9. System response to a 1000 ft/min rate commanded rate of climb

The method of successive loop closures is good for initial work and illustration of the system dynamics, but it is tedious if many gains must be chosen or specific dynamic properties are desired quickly. Furthermore, for the purpose of scheduling gains, the method of choosing gains must be automated. Automating gain scheduling is reasonably straightforward once a control scheme has been established. Since the control logic is simple, the method of pole placement can be completely analytic. This explicit method of pole placement is outlined in Brogan [Br91]. Its limitation is that it is cumbersome and

useful only for low order systems, however its simplicity gives added flexibility in gain selection.

The process starts with replacing the individual terms in the system of equations with placeholders to simplify the final expressions. Furthermore, since we are not feeding back altitude, and it does not contribute to the Phugoid mode, we can remove it from the state equation.

$$\begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \\ \dot{I}_{C_L} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & 0 & b_{21} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ I_{C_L} \end{bmatrix} + \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{p_{C_L}} \\ u_{i_{C_L}} \end{bmatrix}$$

$$\Delta \dot{h} = \begin{bmatrix} 0 & c_{42} & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ I_{C_L} \end{bmatrix}$$

We want to be able to control the eigenvalues of the closed loop A-matrix, which is defined as follows.

$$\mathbf{A}_{cl} = \mathbf{A} - \mathbf{BKC}$$

$$\mathbf{A}_{cl} = \begin{bmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & 0 & b_{21} \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & k_{p_{14}} \\ 0 & k_{i_{14}} \end{bmatrix} \begin{bmatrix} 0 & c_{42} & 0 \end{bmatrix}$$

Simplifying,

$$\mathbf{A}_{cl} = \begin{bmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & 0 & b_{21} \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k_{p_{14}} \\ k_{i_{14}} \end{bmatrix} \begin{bmatrix} 0 & c_{42} & 0 \end{bmatrix}$$

$$\mathbf{A}_{cl} = \begin{bmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & 0 & b_{21} \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & c_{42}k_{p_{14}} & 0 \\ 0 & c_{42}k_{i_{14}} & 0 \end{bmatrix}$$

$$\mathbf{A}_{cl} = \begin{bmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & 0 & b_{21} \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & b_{11}c_{42}k_{p_{14}} & 0 \\ 0 & b_{21}c_{42}k_{p_{14}} & 0 \\ 0 & c_{42}k_{i_{14}} & 0 \end{bmatrix}$$

we have the final closed loop A-matrix in Equation (4.17).

$$\mathbf{A}_{cl} = \begin{bmatrix} a_{11} & a_{12} - b_{11}c_{42}k_{p_{14}} & b_{11} \\ a_{21} & -b_{21}c_{42}k_{p_{14}} & b_{21} \\ 0 & -c_{42}k_{i_{14}} & 0 \end{bmatrix} \quad (4.17)$$

Next, we need to calculate the characteristic equation for the A-matrix.

$$[s\mathbf{I} - \mathbf{A}_{cl}] = \begin{bmatrix} s - a_{11} & -a_{12} + b_{11}c_{42}k_{p_{14}} & -b_{11} \\ -a_{21} & s + b_{21}c_{42}k_{p_{14}} & -b_{21} \\ 0 & c_{42}k_{i_{14}} & s \end{bmatrix}$$

Finally, we have an expression for the characteristic equation shown in (4.18).

$$\begin{aligned} \det[s\mathbf{I} - \mathbf{A}_{cl}] &= (s - a_{11})(s^2 + c_{42}b_{21}k_{p_{14}}s) + a_{21}b_{11}c_{42}k_{i_{14}} \\ &\quad + (-a_{12} + c_{42}b_{11}k_{p_{14}})(a_{21}s) \\ &\quad + (s - a_{11})(b_{21})(c_{42}k_{i_{14}}) \\ &= s^3 + c_{42}b_{21}k_{p_{14}}s^2 - a_{11}s^2 - a_{11}c_{42}b_{21}k_{p_{14}}s + a_{21}b_{11}c_{42}k_{i_{14}} \\ &\quad - a_{12}a_{21}s + c_{42}b_{11}k_{p_{14}}a_{21}s + b_{21}c_{42}k_{i_{14}}s - a_{11}b_{21}c_{42}k_{i_{14}} \\ \det[s\mathbf{I} - \mathbf{A}_{cl}] &= s^3 + (c_{42}b_{21}k_{p_{14}} - a_{11})s^2 \\ &\quad + (c_{42}b_{11}k_{p_{14}}a_{21} + b_{21}c_{42}k_{i_{14}} - a_{11}c_{42}b_{21}k_{p_{14}} - a_{12}a_{21})s \\ &\quad + (a_{21}b_{11}c_{42}k_{i_{14}} - a_{11}b_{21}c_{42}k_{i_{14}}) \end{aligned} \quad (4.18)$$

From this we can determine what gains are necessary to achieve the desired characteristic equation. We have the following form of the characteristic equation:

$$\det[s\mathbf{I} - \mathbf{A}_{cl}] = c_1s^3 + c_2s^2 + c_3s + c_4$$

Therefore, we can set each coefficient equal to its corresponding term in Equation (4.18).

$$\begin{aligned} c_1 &= 1 \\ c_2 &= (c_{42}b_{21}k_{p_{14}} - a_{11}) \end{aligned}$$

$$c_3 = (b_{21}c_{42}k_{i_4} - a_{11}c_{42}b_{21}k_{p_{14}} - a_{12}a_{21} + c_{42}b_{11}k_{p_{14}}a_{21})$$

$$c_4 = (a_{21}c_{42}b_{11}k_{i_4} - a_{11}b_{21}c_{42}k_{i_4})$$

Being able to define the coefficients in terms of the gains is helpful; however, we really want to determine the gains from the coefficients to be sure of getting the correct response. This is a problem because we have three linearly independent equations and 2 unknown gains. This problem is alleviated because of our choice of a single equilibrium condition for gain calculation: steady, level flight at the speed for maximum lift-to-drag ratio (see chapter 6)². In this flight condition, the term, $(a_{21}b_{11} - a_{11}b_{21})$ is identically zero (see appendix for derivation).

$$(a_{21}b_{11} - a_{11}b_{21}) = 0$$

The system of coefficients becomes,

$$c_1 = 1$$

$$c_2 = (c_{42}b_{21}k_{p_{14}} - a_{11})$$

$$c_3 = (b_{21}c_{42}k_{i_4} - a_{12}a_{21})$$

$$c_4 = 0$$

With the c_4 equation eliminated, we have a system of two equations in 2 unknown gains and we can solve for the gains directly. Also, this corresponds to an integral pole at $s = 0$. The characteristic equation is now,

$$\det[s\mathbf{I} - \mathbf{A}_{cl}] = s(s^2 + c_2s + c_3)$$

The coefficients represent the second order phugoid dynamics, so we can easily specify the phugoid modal properties, ζ_p , the damping ratio of the phugoid, and ω_p , the natural frequency of the phugoid and rewrite the characteristic equation.

$$\det[s\mathbf{I} - \mathbf{A}_{cl}] = s(s^2 + 2\zeta_p\omega_p s + \omega_p^2)$$

The system of the gain equations is then rewritten in terms of the phugoid dynamics,

$$2\zeta_p\omega_p = (c_{42}b_{21}k_{p_{14}} - a_{11})$$

$$\omega_p^2 = (b_{21}c_{42}k_{i_4} - a_{12}a_{21})$$

² Also integral to this analysis is that $C_{M_{16}}$ is set to zero. This is because at the time of this revision, compressibility drag had been removed from the model.

We can now solve for the feedback gains.

$$k_{p_{14}} = \frac{2\zeta_p \omega_p + a_{11}}{c_{42} b_{21}} \quad (4.19)$$

$$k_{i_{14}} = \frac{\omega_p^2 + a_{12} a_{21}}{b_{21} c_{42}} \quad (4.20)$$

4.4.2 Lift Coefficient Control of Speed

During climbs and descents, a pilot typically maintains a constant airspeed or Mach number during the altitude change. The pilot will preset the thrust (climb thrust for climbs, idle thrust for descents) and allow the altitude rate to vary accordingly while using the control stick to maintain speed. For these regions, we need a controller that uses lift coefficient to control speed but does not modulate thrust. Since the aircraft has two measurements for speed, Mach and indicated airspeed, both speeds have to be considered in separate analyses. However, since the solutions are identical with the exception of a few changes in feedback gains, only the Mach case is discussed.

The goal of the feedback controller is to capture a given speed by adjusting the lift coefficient. This means that the system is mainly governed by the M/C_L transfer function discussed in Section 3.3.2. This is convenient because we know from previous analysis that this part of the system dynamics is not afflicted with non-minimum phase behavior. However, feedback of speed to the lift coefficient has a problem with low damping as can be seen in Figure 4.2. The feedback control strategy to fix the low damping problem has already been touched upon in Section 3.3.2. Our basic strategy is to build a proportional - plus- integral controller for capturing Mach, and then feedback altitude rate in the feedback path to increase the damping of the system.

We will need two of the four outputs to complete the design, so the output equation of the LTD system of equation (4.10) can be simplified to,

$$\begin{bmatrix} \Delta M \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{\partial M}{\partial V_a} & 0 & 0 \\ 0 & 60 \left(\frac{\partial f_h}{\partial \gamma_a} \right) & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \end{bmatrix}$$

Consider the same DC-9 traveling at 15000 ft and 578 ft/sec and weighing 140,000lbs. The open loop dynamics for the system are the same as shown in equation (4.16), and our new output equation is

$$\begin{bmatrix} \Delta M \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} 9.45 \times 10^5 & 0 & 0 \\ 0 & 3.47 \times 10^4 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \end{bmatrix}$$

Initially, the Phugoid eigenvalues are located at $-0.0036 \pm 0.0786i$ which correspond to a natural frequency of 0.0787 rad/sec and a damping ratio of 0.0445. We start the loop closures by closing the feedback path $k_{b_{14}}$ as shown in Figure 4.10. Setting $k_{b_{14}} = 2.7 \times 10^{-5}$ moves the Phugoid poles to $-0.0599 \pm 0.0540i$ where they have a damping ratio of 0.7173 and a natural frequency of 0.0806 rad/sec. Applying proportional Mach feedback to the lift coefficient, we increase the frequency of the Phugoid poles to 0.20 rad/sec, which we initially think is a good value as shown in Figure 4.11. The resulting feedback gain is $k_{p_{12}} = -9.5$. Note the negative value of $k_{p_{12}}$. This makes intuitive sense because an increase in speed should be the result of a lower lift coefficient. To test the partially built controller, we attempt a command to Mach 0.7. From the simulation shown in Figure 4.12, we find out that while we like the response, the required control effort is excessive. The lift coefficient drops nearly to -1. Realizing that too much control effort is required, we drop the frequency down to 0.1 rad/sec which corresponds to a gain of $k_{p_{12}} = -1.5$. With this reduction in gain, the Phugoid poles sit at $-0.0618 \pm 0.0904i$ with a frequency of 0.1095 rad/sec and a damping ratio of 0.5647.

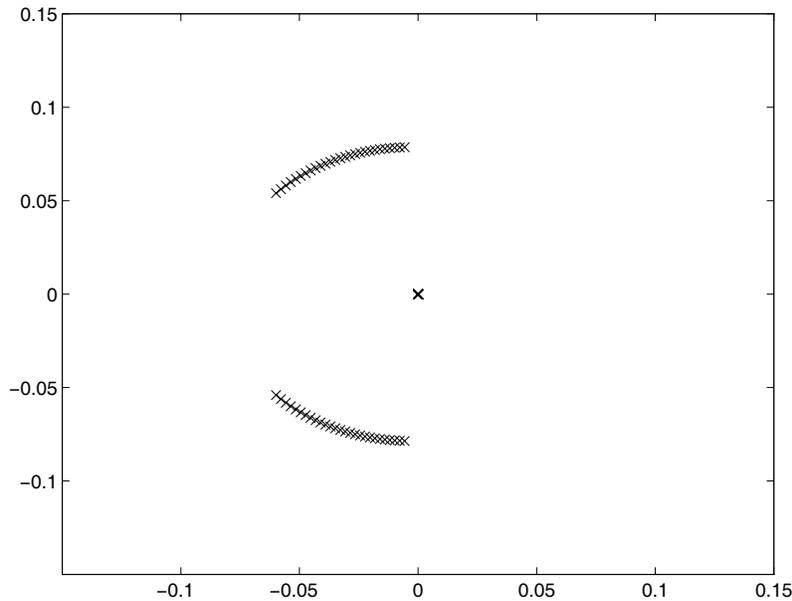


Figure 4.10. $k_{b_{14}}$ root locus for the Mach Capture controller

The next loop closure, the integral control, is tricky because the integral control tends to send the Phugoid poles towards the right half plane. Integral control also pulls the integrator pole away from zero along the negative real axis. A gain of $k_{i_{12}} = -0.1$ is

applied which tends to line up the real parts of the Phugoid pole with the integrator pole as shown in Figure 4.13. The Phugoid poles move to $-0.0412 \pm 0.0846i$ where the frequency is 0.0941 rad/sec and the damping is 0.4377. Since the damping is low, more altitude rate is applied. The altitude rate is applied until the locus starts to curve back

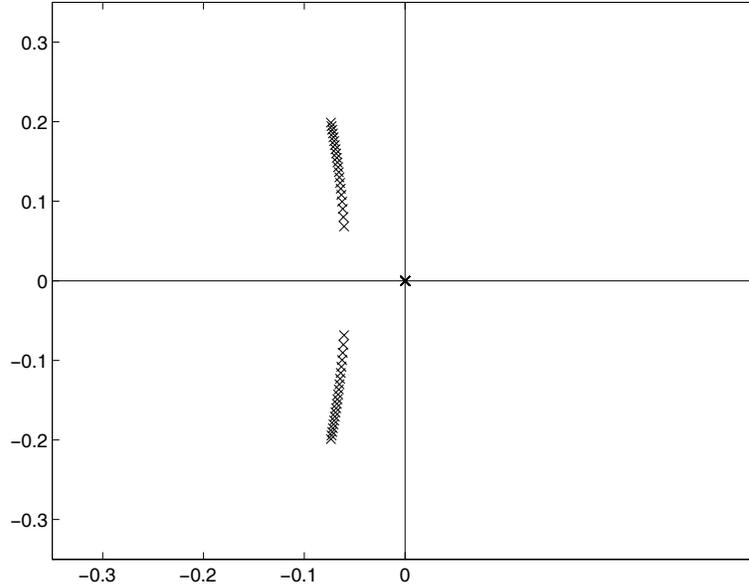


Figure 4.11. $k_{p_{12}}$ root locus for the Mach Capture controller

inward and head towards the imaginary axis. The gain is set to $k_{b_{14}} = 4.2 \times 10^{-5}$ where the frequency of the Phugoid is 0.0614rad/sec and the damping is 0.7265.

We also note that the integrator pole is moved to -0.0971. The complete system is simulated as shown in Figure 4.14. From Figure 4.14 we see that the controller captures a Mach of 0.7 and that the lift coefficient is not unreasonable. However, the speed of the response is low.

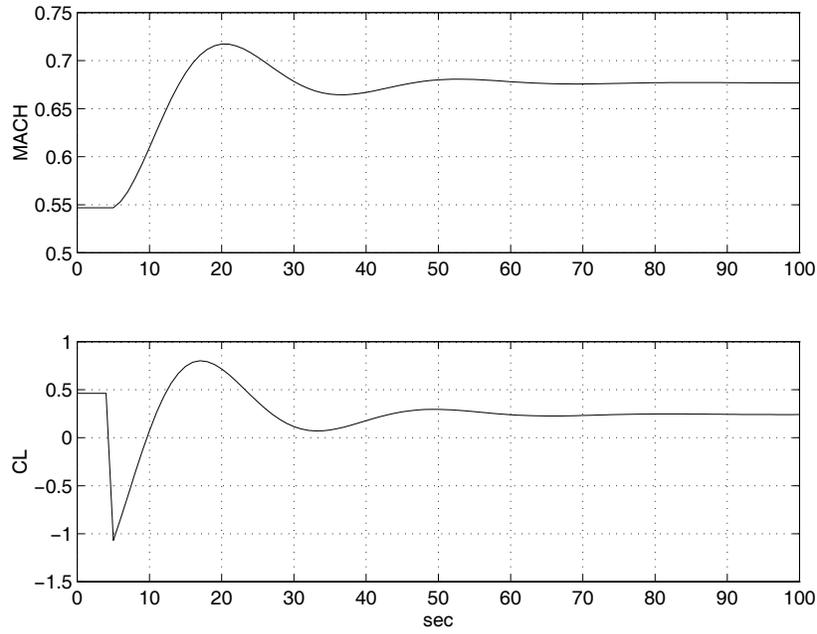


Figure 4.12. Simulated Mach Capture

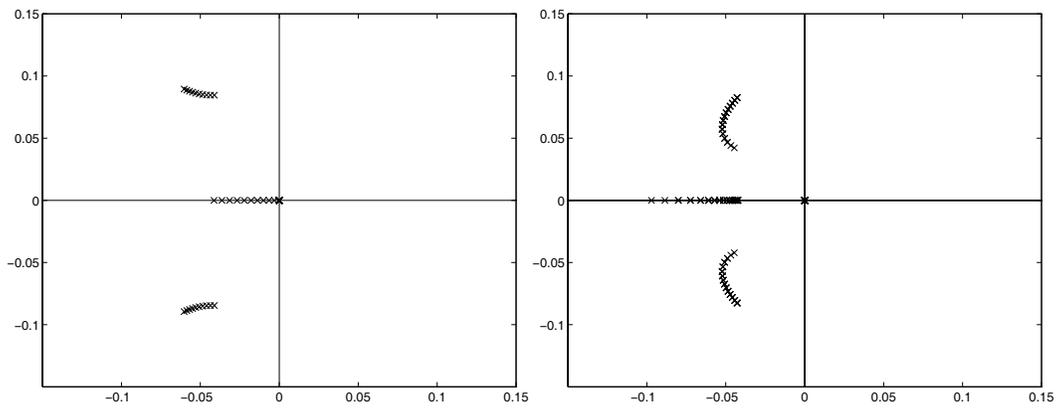


Figure 4.13. $k_{h_{12}}$ (left) and $k_{b_{14}}$ (right) root loci for Mach capture

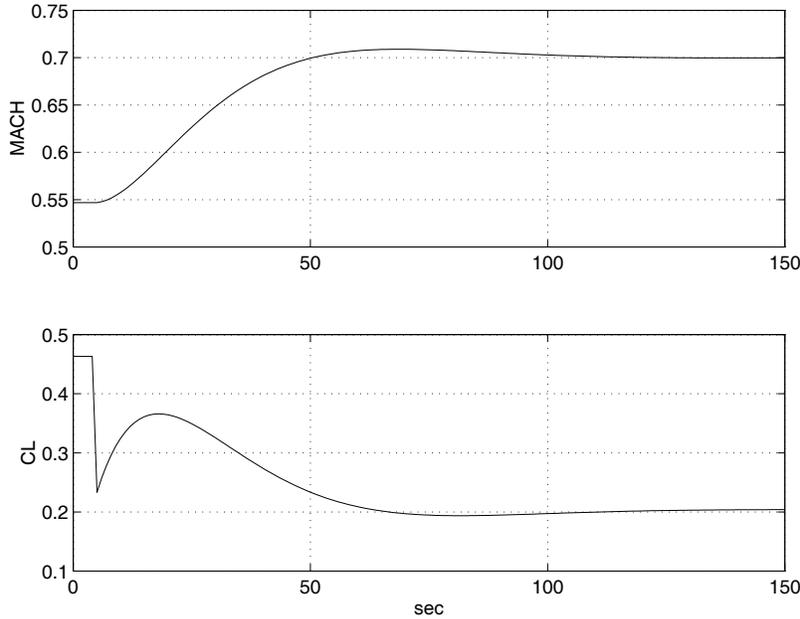


Figure 4.14. Simulation of the Completed Mach Capture

In adding the extra gain to $k_{b_{14}}$ we diminished our frequency so we would probably want to redo the design if this set of gains were actually going to be used. However, the actual gains used are calculated automatically by the gain scheduling algorithm where we can choose exactly the modal properties that we want.

We have seen from the control law design that successive loop closures can be difficult. In this example, one can see that just about any set of modal properties could be achieved; however, the trial and error approach is certain to take a considerable amount of time. It is still valuable to manually close the loops at least once because it helps to build our understanding of the system dynamics. For instance, we now know that high system frequencies require an excessive control force. For this system, it is best to schedule natural frequencies on the order of 0.1 rad/sec and no higher. For the purpose of scheduling the actual gains, the automated method is presented next.

Taking the LTD system of equation (4.10), we remove indicated airspeed and altitude from the output equation and we remove altitude and integrated thrust from the state equation.

$$\begin{bmatrix} \Delta \dot{V}_a \\ \Delta \dot{\gamma}_a \\ \dot{I}_{C_L} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & 0 & b_{21} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ I_{C_L} \end{bmatrix} + \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{p_{C_L}} \\ u_{i_{C_L}} \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} M \\ \dot{h} \end{bmatrix} = \begin{bmatrix} c_{21} & 0 & 0 \\ 0 & c_{42} & 0 \end{bmatrix} \begin{bmatrix} V_a \\ \gamma_a \\ I_{C_L} \end{bmatrix}$$

The basic control law as developed earlier is,

$$C_L = \left(k_{p_{12}} + \frac{k_{i_{12}}}{s} \right) e_2 + k_{b_{14}} \dot{h}$$

where

$$e_2 = (M_d - M)$$

Note that for the purpose of determining eigenvalues, the gains $k_{p_{14}}$ and $k_{b_{14}}$ have identical effects. Therefore we substitute $k_{b_{14}}$ into the matrix location reserved for $k_{p_{14}}$ for the purposes of gain scheduling *only*. The gain matrix becomes,

$$\mathbf{K} = \begin{bmatrix} k_{p_{12}} & k_{b_{14}} \\ k_{i_{12}} & 0 \end{bmatrix}$$

The closed loop form of the equations (A-BKC) is,

A-BKC

$$= \begin{bmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & 0 & b_{21} \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k_{p_{12}} & k_{b_{14}} \\ k_{i_{12}} & 0 \end{bmatrix} \begin{bmatrix} c_{21} & 0 & 0 \\ 0 & c_{42} & 0 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & 0 & b_{21} \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k_{p_{12}} c_{21} & k_{b_{14}} c_{42} & 0 \\ k_{i_{12}} c_{21} & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & b_{11} \\ a_{21} & 0 & b_{21} \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} k_{p_{12}} c_{21} & b_{11} k_{b_{14}} c_{42} & 0 \\ b_{21} k_{p_{12}} c_{21} & b_{21} k_{b_{14}} c_{42} & 0 \\ k_{i_{12}} c_{21} & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} - b_{11}k_{p_{12}}c_{21} & a_{12} - b_{11}k_{b_{14}}c_{42} & b_{11} \\ a_{21} - b_{21}k_{p_{12}}c_{21} & -b_{21}k_{b_{14}}c_{42} & b_{21} \\ -k_{i_2}c_{21} & 0 & 0 \end{bmatrix}$$

The characteristic equation is calculated next.

$$\det[s\mathbf{I} - \mathbf{A}_{cl}] = \det \begin{bmatrix} s - a_{11} + b_{11}k_{p_{12}}c_{21} & -a_{12} + b_{11}k_{b_{14}}c_{42} & -b_{11} \\ -a_{21} + b_{21}k_{p_{12}}c_{21} & s + b_{21}k_{b_{14}}c_{42} & -b_{21} \\ +k_{i_2}c_{21} & 0 & s \end{bmatrix}$$

$$\begin{aligned} \det[s\mathbf{I} - \mathbf{A}_{cl}] &= s(s - a_{11} + b_{11}k_{p_{12}}c_{21})(s + b_{21}k_{b_{14}}c_{42}) + (-a_{12} + b_{11}k_{b_{14}}c_{42})(-b_{21})(+k_{i_2}c_{21}) \\ &\quad - (-b_{11})(s + b_{21}k_{b_{14}}c_{42})(+k_{i_2}c_{21}) - s(-a_{12} + b_{11}k_{b_{14}}c_{42})(-a_{21} + b_{21}k_{p_{12}}c_{21}) \end{aligned}$$

Summing like terms yields the characteristic equation.

$$\begin{aligned} \det[s\mathbf{I} - \mathbf{A}_{cl}] &= s^3 + (c_{42}b_{21}k_{b_{14}})s^2 - (a_{11} - b_{11}k_{p_{12}}c_{21})s^2 - ((a_{11} - b_{11}k_{p_{12}}c_{21})c_{42}b_{21}k_{b_{14}})s \\ &\quad + (a_{12}(-a_{21} + b_{21}k_{p_{12}}c_{21}))s + (a_{12}b_{21}k_{i_2}c_{21}) \\ &\quad - (c_{42}b_{11}k_{b_{14}}(-a_{21} + b_{21}k_{p_{12}}c_{21}))s - (c_{42}b_{11}k_{b_{14}}b_{21}k_{i_2}c_{21}) \\ &\quad + (k_{i_2}b_{11}c_{21})s + (c_{42}b_{11}b_{21}k_{b_{14}}k_{i_2}c_{21}) \end{aligned}$$

$$\begin{aligned} \det[s\mathbf{I} - \mathbf{A}_{cl}] &= s^3 + [(c_{42}b_{21}k_{b_{14}}) - (a_{11} - b_{11}k_{p_{12}}c_{21})]s^2 \\ &\quad + [(a_{12}(-a_{21} + b_{21}k_{p_{12}}c_{21})) - ((a_{11} - b_{11}k_{p_{12}}c_{21})c_{42}b_{21}k_{b_{14}}) \\ &\quad - (c_{42}b_{11}k_{b_{14}}(-a_{21} + b_{21}k_{p_{12}}c_{21})) + (k_{i_2}b_{11}c_{21})]s \\ &\quad + [(a_{12}b_{21}k_{i_2}c_{21}) - (c_{42}b_{11}k_{b_{14}}b_{21}k_{i_2}c_{21}) + (c_{42}b_{11}b_{21}k_{b_{14}}k_{i_2}c_{21})] \end{aligned}$$

The coefficients of the terms are difficult to manage so some effort is applied to simplifying them. We wish to express the characteristic equation in the form,

$$\det[s\mathbf{I} - \mathbf{A}] = C_1s^3 + C_2s^2 + C_3s + C_4$$

Setting the two forms equal, we can solve for the coefficients.

$$s^3 : \quad C_1 = 1$$

$$s^2 : C_2 = \left[(c_{42} b_{21} k_{b_{44}}) - (a_{11} - b_{11} k_{p_{12}} c_{21}) \right]$$

$$C_2 = c_{42} b_{21} k_{b_{44}} + b_{11} k_{p_{12}} c_{21} - a_{11}$$

$$s^1 : C_3 = \left[(a_{12} (-a_{21} + b_{21} k_{p_{12}} c_{21})) + ((-a_{11} + b_{11} k_{p_{12}} c_{21}) c_{42} b_{21} k_{b_{44}}) \right. \\ \left. + (-a_{32} b_{11} k_{b_{44}} (-a_{21} + b_{21} k_{p_{12}} c_{21})) + (k_{i_2} b_{11} c_{21}) \right]$$

$$C_3 = [-a_{12} a_{21} + a_{12} b_{21} k_{p_{12}} c_{21} - a_{11} c_{42} b_{21} k_{b_{44}} + c_{42} b_{11} b_{21} k_{p_{12}} k_{b_{44}} c_{21} \\ + a_{21} c_{42} b_{11} k_{b_{44}} - c_{42} b_{11} b_{21} k_{p_{12}} k_{b_{44}} c_{21} + k_{i_2} b_{11} c_{21}]$$

$$C_3 = -a_{12} a_{21} + a_{12} b_{21} k_{p_{12}} c_{21} - a_{11} c_{42} b_{21} k_{b_{44}} + a_{21} c_{42} b_{11} k_{b_{44}} + k_{i_2} b_{11} c_{21}$$

$$s^0 : C_4 = \left[(a_{12} b_{21} k_{i_2} c_{21}) - (a_{32} b_{11} b_{21} k_{b_{44}} k_{i_2} c_{21}) + (a_{32} b_{11} b_{21} k_{b_{44}} k_{i_2} c_{21}) \right]$$

$$C_4 = a_{12} b_{21} k_{i_2} c_{21}$$

Assuming that the gain $k_{b_{44}}$ is known, we can solve for the other gains.

$$k_{p_{12}} = \frac{C_2 - c_{42} b_{21} k_{b_{44}} + a_{11}}{b_{11} c_{21}} \quad (4.21)$$

$$k_{i_2} = \frac{C_4}{a_{12} b_{21} c_{21}} \quad (4.22)$$

Plugging these gains back into the C_3 equation yields,

$$C_3 = -a_{12} a_{21} + a_{12} b_{21} \left(\frac{C_2 - c_{42} b_{21} k_{b_{44}} + a_{11}}{b_{11} c_{21}} \right) c_{21} \\ - a_{11} c_{42} b_{21} k_{b_{44}} + a_{21} c_{42} b_{11} k_{b_{44}} + \left(\frac{C_4}{a_{12} b_{21} c_{21}} \right) b_{11} c_{21}$$

Simplifying,

$$C_3 = -a_{12}a_{21} + a_{12}b_{21} \left(\frac{C_2 - c_{42}b_{21}k_{b_4} + a_{11}}{b_{11}c_{11}} \right) c_{21} - a_{11}c_{42}b_{21}k_{b_4} + a_{21}c_{42}b_{11}k_{b_4} + \left(\frac{C_4}{a_{12}b_{21}c_{21}} \right) b_{11}c_{21}$$

$$C_3 = -a_{12}a_{21} + \frac{a_{12}b_{21}C_2}{b_{11}} - \left(\frac{a_{12}b_{21}c_{42}b_{21}}{b_{11}} \right) k_{b_4} + \frac{a_{11}a_{12}b_{21}}{b_{11}} - (a_{11}c_{42}b_{21})k_{b_4} + (a_{21}c_{42}b_{11})k_{b_4} + \frac{C_4b_{11}}{a_{12}b_{21}}$$

$$C_3 = \left(a_{21}c_{42}b_{11} - \frac{a_{12}b_{21}c_{42}b_{21}}{b_{11}} - a_{11}c_{42}b_{21} \right) k_{b_4} + \left(\frac{a_{12}b_{21}C_2}{b_{11}} - a_{12}a_{21} + \frac{a_{11}a_{12}b_{21}}{b_{11}} + \frac{C_4b_{11}}{a_{12}b_{21}} \right)$$

and we can solve for the gain k_{b_4} .

$$k_{b_4} = \frac{C_3 - \left(\frac{a_{12}b_{21}C_2}{b_{11}} - a_{12}a_{21} + \frac{a_{11}a_{12}b_{21}}{b_{11}} + \frac{C_4b_{11}}{a_{12}b_{21}} \right)}{\left(a_{21}c_{42}b_{11} - \frac{a_{12}b_{21}c_{42}b_{21}}{b_{11}} - a_{11}c_{42}b_{21} \right)} \quad (4.23)$$

Because we have three gains, we see that we were able to specify all the coefficients of the characteristic equation. This enables us to place the poles arbitrarily. We can choose the frequency and damping of the Phugoid poles and the location of the integrator pole. From our manual loop closing, we also have some idea of what values make good gains. We can specify the coefficients in terms of the desired modal properties and the location of the integrator pole by expressing the characteristic equation in terms of its roots.

$$\det[s\mathbf{I} - \mathbf{A}] = C_1s^3 + C_2s^2 + C_3s + C_4 = (s + p)(s^2 + 2\zeta\omega_p s + \omega_p^2)$$

$$\begin{aligned} C_1 &= 1 \\ C_2 &= p + 2\zeta\omega_p \\ C_3 &= 2p\zeta\omega_p + \omega_p^2 \\ C_4 &= p\omega_p^2 \end{aligned} \quad (4.24)$$

Table 4.1 summarizes the gain scheduling equations used for this controller. Looking at Figure 4.14, we see that the lift coefficient is still commanded rather violently. Therefore we decide to specify slower dynamics. Using equations in Table 4.1 we can try many different modal properties quickly. After some experimentation, we choose the following modal properties:

- $\omega_p = 0.05 \text{ rad / sec}$
- $\zeta_p = 0.7$
- Integrator pole location: -0.05

Table 4.1. The gain scheduling equations for lift coefficient control of speed

$ \begin{aligned} C_1 &= 1 \\ C_2 &= p + 2\zeta\omega_p \\ C_3 &= 2p\zeta\omega_p + \omega_p^2 \\ C_4 &= p\omega_p^2 \end{aligned} \tag{4.24} $
$ k_{b_{14}} = \frac{C_3 - \left(\frac{a_{12}b_{21}C_2}{b_{11}} - a_{12}a_{21} + \frac{a_{11}a_{12}b_{21}}{b_{11}} + \frac{C_4b_{11}}{a_{12}b_{21}} \right)}{\left(a_{21}c_{42}b_{11} - \frac{a_{12}b_{21}c_{42}b_{21}}{b_{11}} - a_{11}c_{42}b_{21} \right)} \tag{4.23} $
$ k_{i_{12}} = \frac{C_4}{a_{12}b_{21}c_{21}} \tag{4.22} $
$ k_{p_{12}} = \frac{C_2 - c_{42}b_{21}k_{b_{14}} + a_{11}}{b_{11}c_{21}} \tag{4.21} $

The gains for this set of modal properties are $k_{p_{12}} = 0.1635$, $k_{i_{12}} = -0.0342$, and $k_{b_{14}} = 0.2715 \times 10^{-4}$. Figure 4.15 shows the simulation results. With the slower dynamics, the lift coefficient no longer has its initial sharp dip; however the system takes 40 seconds longer to capture the desired Mach number. This performance penalty is acceptable because an extra 40 seconds is not much time in the course of an entire flight.

4.4.3 Controlling Speed and Altitude Rate Simultaneously

Maintaining a specified speed and a specified altitude rate (as in steady, level flight) requires feedback control of both the thrust and lift coefficient. Because of this, the controller for this flight regime is the most complicated of our controllers. It is also the most used controller, because the aircraft spends most of its time controlling both speed and altitude (as in steady, level flight). The design goals of this controller are also the most ambitious. Here, we desire to drive both the speed of the aircraft and altitude rate to some commanded values.

Consider the same DC-9 traveling at 15000 ft and 578 ft/sec and weighing 140,000lbs. The open loop dynamics for system are the same as shown in equation (4.16); however, because of the addition of the throttle feedback control, we need to use the full state equation of the LTD system in equation (4.10). With the given flight condition, the state equation becomes,

$$\begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \\ \dot{I}_{C_L} \\ \dot{I}_T \end{bmatrix} = \begin{bmatrix} -0.0072 & -32.2 & 0 & -2.7445 & 0.0002 \\ 0.0002 & 0 & 0 & 0.1202 & 0 \\ 0 & 578.4794 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta V_a \\ \Delta \gamma_a \\ \Delta h \\ I_{C_L} \\ I_T \end{bmatrix} + \begin{bmatrix} -2.7445 & 0.0002 & 0 & 0 \\ 0.1202 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{P_{C_L}} \\ u_{P_T} \\ u_{i_{C_L}} \\ u_{i_T} \end{bmatrix} \quad (4.25)$$

Since we want to be able to drive the aircraft to a particular speed and altitude rate independently of each other, we must be very careful how we arrange the feedback in the feed-forward path. Feedback to any one input cannot be used to drive two independent errors to zero simultaneously. By observing effects of feedback in Figure 4.2 and Figure 4.3, we see that the throttle is much more adept at controlling speed than altitude rate. This is because the feedback of speed to the throttle tends to increase the damping of the Phugoid mode. Similarly, the lift coefficient is a much better control of altitude rate than speed. Therefore we only allow speed feedback to the throttle and altitude rate feedback to the lift coefficient. For purposes of this demonstration, we use Mach in the feedback. The control laws which use indicated airspeed are identical in form.

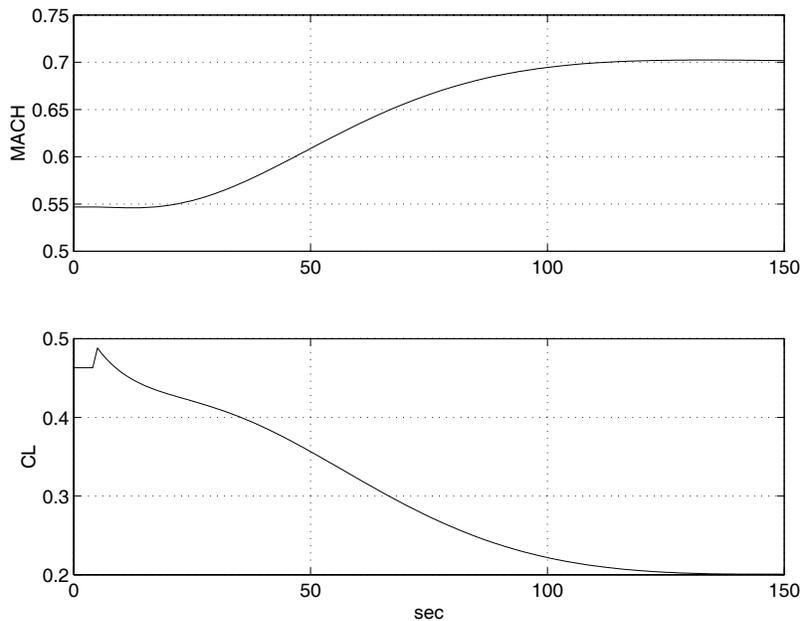


Figure 4.15. Simulation of Mach capture with slower dynamics

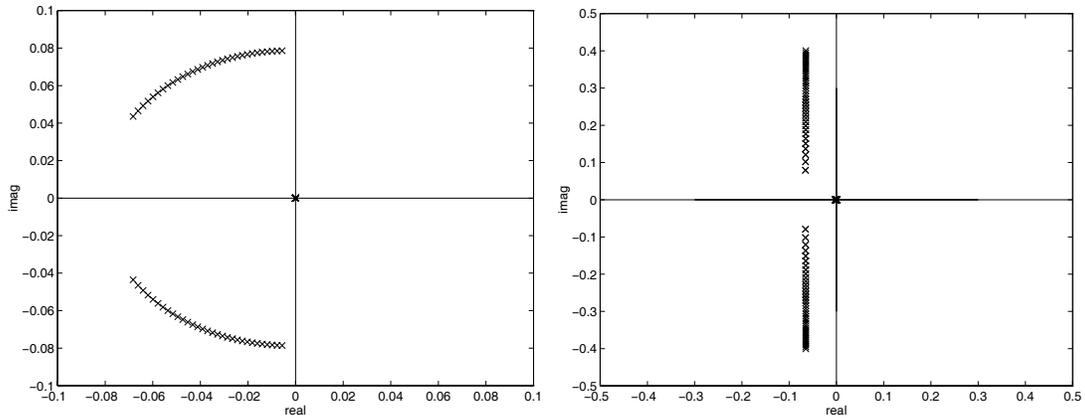


Figure 4.16. The initial $k_{p_{14}}$ (left plot) and $k_{i_{14}}$ (right plot) loop closures for region 7

Initially, the Phugoid eigenvalues are located at $-0.0036 \pm 0.0786i$ which corresponds to a natural frequency of 0.0787rad/sec and a damping ratio of 0.0445 . Initially, we close the lift coefficient feedback paths in a manner very similar to what was done in Section 4.4.1. We close the proportional loop first using $k_{p_{14}}$. We increase the proportional gain to $k_{p_{14}} = 3.0 \times 10^{-5}$ as shown in Figure 4.16. This results in Phugoid poles at $-0.0662 \pm 0.0465i$. Our modal properties are $\omega_p = 0.081\text{rad/sec}$ and $\zeta_p = 0.819$. Our damping ratio is acceptable; however, our frequency is low. We adjust the frequency

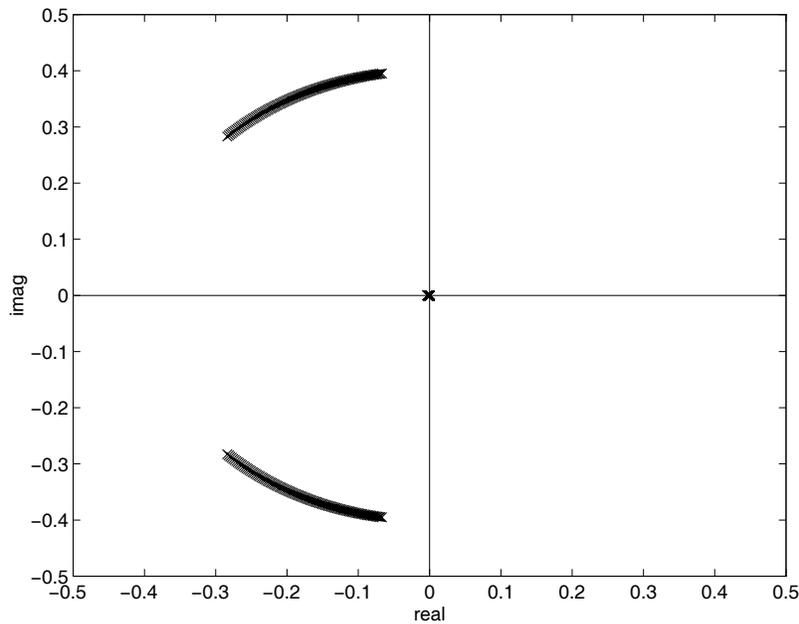


Figure 4.17. The final increase in $k_{p_{14}}$ to achieve adequate damping

when we add integral control. As shown in the second plot of Figure 4.16, we increase the integral gain to $k_{i_{14}} = 3.7 \times 10^{-5}$. This moves the Phugoid poles to $-0.0648 \pm 0.3954i$ where the modal properties are $\omega_p = 0.4\text{rad/sec}$ and $\zeta_p = 0.1618$. While this results in

an acceptable frequency, the damping ratio is too low. Therefore, we increase the proportional gain to $k_{p_{14}} = 1.34 \times 10^{-4}$ as shown in Figure 4.17. This results in Phugoid poles of $-0.2818 \pm 0.2850i$ with the corresponding modal properties of $\zeta_p = 0.7031$ and $\omega_p = 0.4 \text{ rad/sec}$. The position of the integrator pole is -0.0027 .

At this point we can illustrate the system dynamics by commanding a 1000 ft/min rate of climb. The simulation results, shown in Figure 4.18, show response of altitude rate and Mach number as well as the lift coefficient and thrust inputs. We can see that we approximately achieve the 1000 ft/min climb rate; however, we have a small error of approximately 25 ft/min. This error is due to the slow integrator pole. We ignore this small error for now. We also see that the Mach number tapers off during the climb. This is expected at the moment because there is no feedback to the throttle to maintain Mach (the thrust is constant).

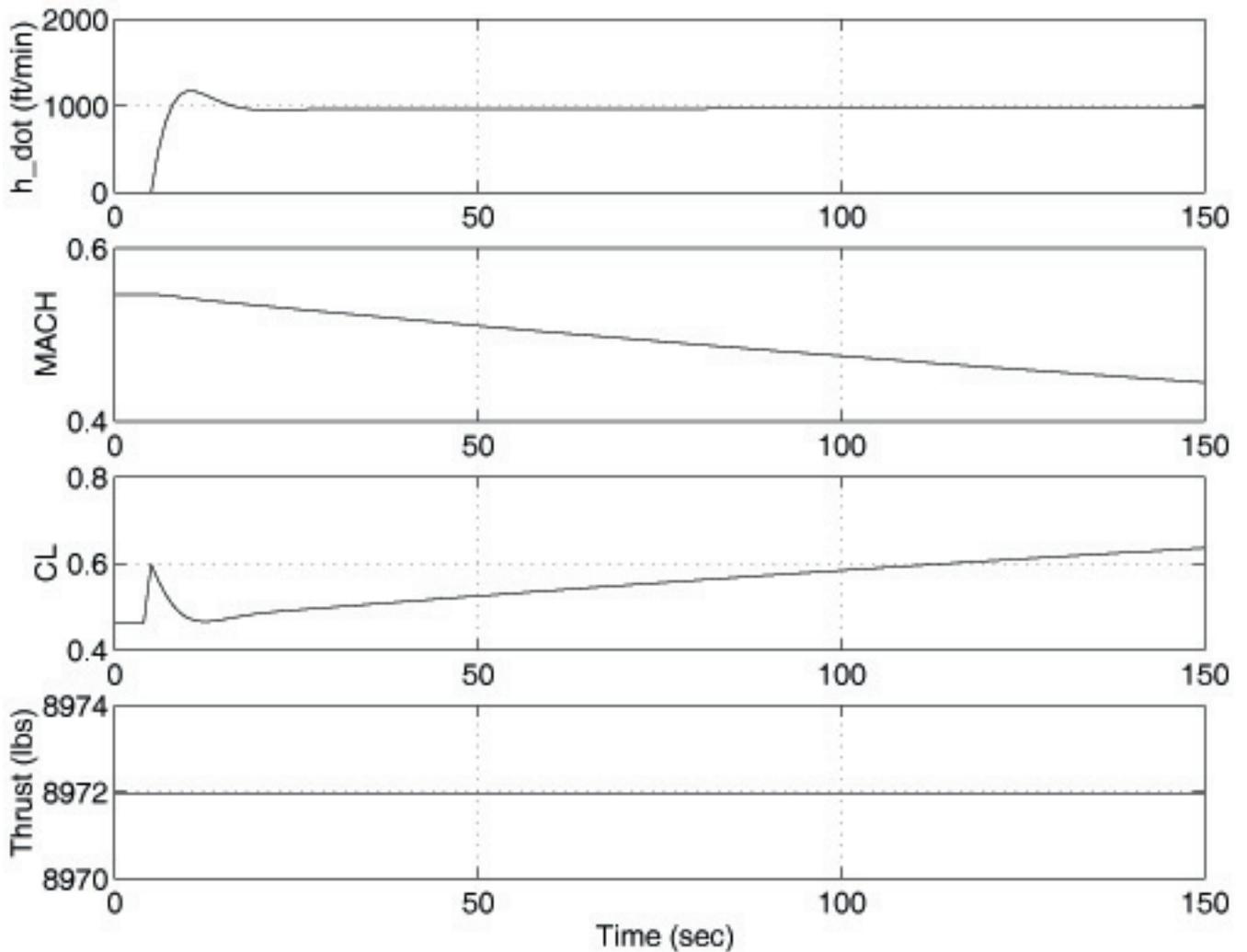


Figure 4.18. Simulation of a commanded 1000ft/min rate of climb without any feedback to thrust

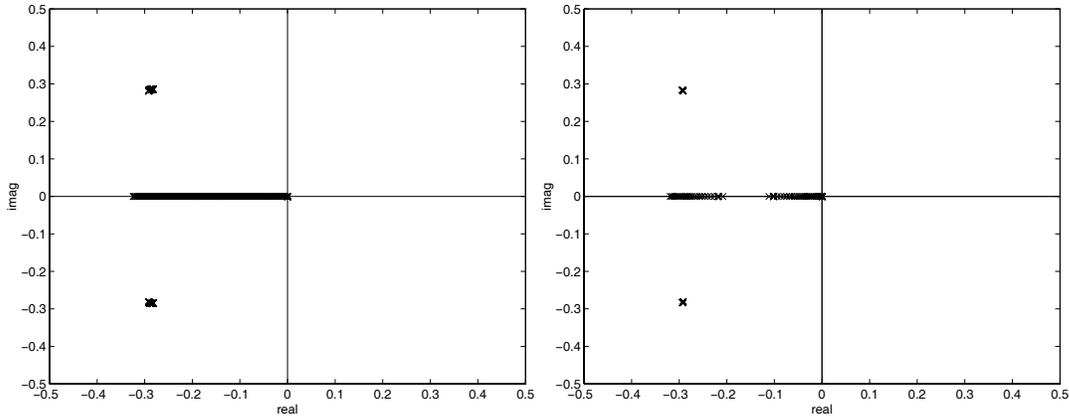


Figure 4.19. Root loci for $k_{p_{22}}$ (left plot) and $k_{i_{22}}$ (right plot)

We also see that the controller slowly increases the lift coefficient to maintain the rate of climb as the aircraft slows down. If the controller is left to continue, it will stall the aircraft. Adding feedback to thrust corrects the problem. Feedback to thrust is initiated using proportional control. We increase the gain $k_{p_{22}}$ to 1.56×10^6 as shown on the left side of Figure 4.19. The Phugoid is virtually unaffected by the proportional feedback to throttle. The biggest influence of the proportional feedback to the thrust is to move the integrator pole for the lift coefficient feedback farther negative. The lift coefficient's integrator pole is moved to -0.3222. This is actually desirable because it reduces the steady state error in altitude rate. The final loop closure, integrated thrust is closed next. As shown on the right hand side of Figure 4.19, the integrated feedback $k_{i_{22}}$ to thrust has virtually no effect on the Phugoid dynamics. Rather, it tends to draw the two integrator poles together. If the gain is increased further, the integrator poles are drawn together and become complex conjugates. This in turn creates another modal oscillation which is slower than the Phugoid. This extra mode is undesirable, so we stop the integrated feedback at $k_{i_{22}} = 1.1 \times 10^5$. The final system is described as:

- Phugoid poles: $-0.2927 \pm 0.2836i$
- Thrust integrator pole : -0.1020
- Lift coefficient integrator pole: -0.2179
- Modal properties: $\zeta_p = 0.7182$ $\omega_p = 0.407 \text{ rad/sec}$

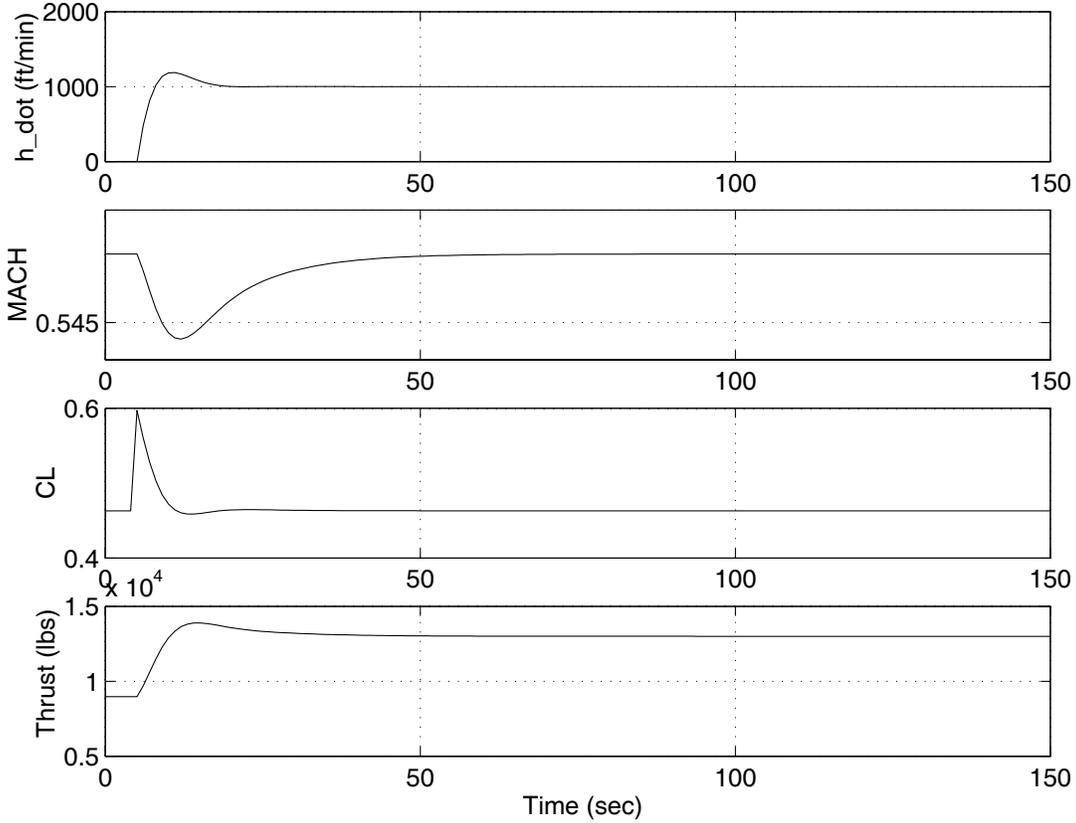


Figure 4.20. Simulation of a commanded 1000ft/min climb rate using the final controller for region 7. The final system is shown simulating a commanded altitude rate of 1000 ft/min in Figure 4.20. Note that with the addition of thrust feedback, the throttle is adjusted simultaneously to maintain the desired speed.

As with the other regions, the exercise of successive loop closures is useful but not practical for the task of scheduling many different conditions. Therefore, an automated approach is developed. With the addition of throttle feedback, the problem becomes much more complex. Again, we will substitute place holders into the LTD system to make the algebra more manageable.

$$\begin{bmatrix} \dot{V}_a \\ \dot{\gamma}_a \\ \dot{I}_{C_L} \\ \dot{I}_T \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_a \\ \gamma_a \\ I_{C_L} \\ I_T \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & 0 & 0 \\ b_{21} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{p_{C_L}} \\ u_{p_T} \\ u_{i_{C_L}} \\ u_{i_T} \end{bmatrix} \tag{4.26}$$

$$y = \begin{bmatrix} M \\ \dot{h} \end{bmatrix} = \begin{bmatrix} c_{21} & 0 & 0 & 0 \\ 0 & c_{42} & 0 & 0 \end{bmatrix} \begin{bmatrix} V_a \\ \gamma_a \\ I_{C_L} \\ I_T \end{bmatrix}$$

We need to close the loops around the throttle and the lift coefficient. The control law is summarized as

$$C_{L_p} = k_{p_{14}} (\dot{h}_d - \dot{h})$$

$$C_{L_i} = k_{i_{14}} (\dot{h}_d - \dot{h})$$

$$T_p = k_{p_{22}} (M_d - M)$$

$$T_i = k_{i_{22}} (M_d - M)$$

The closed loop A-matrix, $\mathbf{A}_{cl} = \mathbf{A} - \mathbf{BKC}$ is,

$$\mathbf{A}_{cl} = \begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} & 0 & 0 \\ b_{21} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & k_{p_{14}} \\ k_{p_{21}} & 0 \\ 0 & k_{i_{14}} \\ k_{i_{21}} & 0 \end{bmatrix} \begin{bmatrix} c_{11} & 0 & 0 & 0 \\ 0 & c_{42} & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_{cl} = \begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} & 0 & 0 \\ b_{21} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & c_{42}k_{p_{14}} & 0 & 0 \\ c_{11}k_{p_{21}} & 0 & 0 & 0 \\ 0 & c_{42}k_{i_{14}} & 0 & 0 \\ c_{11}k_{i_{21}} & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_{cl} = \begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{12}c_{11}k_{p_{21}} & b_{11}c_{42}k_{p_{14}} & 0 & 0 \\ 0 & b_{21}c_{42}k_{p_{14}} & 0 & 0 \\ 0 & c_{42}k_{i_{14}} & 0 & 0 \\ c_{11}k_{i_{21}} & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_{cl} = \begin{bmatrix} a_{11} - b_{12}c_{11}k_{p_{21}} & a_{12} - b_{11}c_{42}k_{p_{14}} & b_{11} & b_{12} \\ a_{21} & -b_{21}c_{42}k_{p_{14}} & b_{21} & 0 \\ 0 & -c_{42}k_{i_{14}} & 0 & 0 \\ c_{11}k_{i_{21}} & 0 & 0 & 0 \end{bmatrix}$$

The characteristic equation of the closed-loop A-matrix is,

$$\det[s\mathbf{I} - \mathbf{A}_{cl}] = \det \begin{bmatrix} s - a_{11} + b_{12}c_{11}k_{p_{21}} & -a_{12} + b_{11}c_{42}k_{p_{14}} & -b_{11} & -b_{12} \\ -a_{21} & s + b_{21}c_{42}k_{p_{14}} & -b_{21} & 0 \\ 0 & c_{42}k_{i_{14}} & s & 0 \\ c_{11}k_{i_{21}} & 0 & 0 & s \end{bmatrix} \quad (4.27)$$

Simplifying Equation (4.27),

$$\det[s\mathbf{I} - \mathbf{A}_{cl}] = +(c_{11}k_{i_{21}}) \det \begin{bmatrix} -a_{12} + b_{11}c_{42}k_{p_{14}} & -b_{11} & -b_{12} \\ s + b_{21}c_{42}k_{p_{14}} & -b_{21} & 0 \\ c_{42}k_{i_{14}} & s & 0 \end{bmatrix} \\ - (s) \det \begin{bmatrix} s - a_{11} + b_{12}c_{11}k_{p_{21}} & -a_{12} + b_{11}c_{42}k_{p_{14}} & -b_{11} \\ -a_{21} & s + b_{21}c_{42}k_{p_{14}} & -b_{21} \\ 0 & c_{42}k_{i_{14}} & s \end{bmatrix}$$

$$\det[s\mathbf{I} - \mathbf{A}_{cl}] = (c_{11}k_{i_{21}}) \left[(-b_{12}s^2 - b_{12}b_{21}c_{42}k_{p_{14}}s) - (b_{12}b_{21}c_{42}k_{i_{14}}) \right] \\ - (s) \left[(s - a_{11} + b_{12}c_{11}k_{p_{21}})(s + b_{21}c_{42}k_{p_{14}})(s) \right. \\ \left. + (-a_{21})(c_{42}k_{i_{14}})(-b_{11}) \right. \\ \left. - (-b_{21})(c_{42}k_{i_{14}})(s - a_{11} + b_{12}c_{11}k_{p_{21}}) \right. \\ \left. - (-a_{12} + b_{11}c_{42}k_{p_{14}})(-a_{21})(s) \right]$$

$$\det[s\mathbf{I} - \mathbf{A}_{cl}] = -b_{12}c_{11}k_{i_{21}}s^2 - b_{12}b_{21}c_{42}k_{p_{14}}c_{11}k_{i_{21}}s - c_{11}k_{i_{21}}b_{12}b_{21}c_{42}k_{i_{14}} \\ - (s^2)(s - a_{11} + b_{12}c_{11}k_{p_{21}})(s + b_{21}c_{42}k_{p_{14}}) \\ - a_{21}b_{11}c_{42}k_{i_{14}}s - b_{21}c_{42}k_{i_{14}}(s^2 - a_{11}s + b_{12}c_{11}k_{p_{21}}s) \\ - a_{21}(-a_{12} + b_{11}c_{42}k_{p_{14}})s^2$$

$$\det[s\mathbf{I} - \mathbf{A}_{cl}] = \\ -b_{12}c_{11}k_{i_{21}}s^2 - b_{12}b_{21}c_{42}k_{p_{14}}c_{11}k_{i_{21}}s - c_{11}k_{i_{21}}b_{12}b_{21}c_{42}k_{i_{14}} \\ - s^4 + a_{11}s^3 - b_{12}c_{11}k_{p_{21}}s^3 - b_{21}c_{42}k_{p_{14}}s^3 + a_{11}b_{21}c_{42}k_{p_{14}}s^2 - b_{12}c_{11}k_{p_{21}}b_{21}c_{42}k_{p_{14}}s^2 \\ - a_{21}b_{11}c_{42}k_{i_{14}}s - b_{21}c_{42}k_{i_{14}}s^2 + b_{21}c_{42}k_{i_{14}}a_{11}s - b_{21}c_{42}k_{i_{14}}b_{12}c_{11}k_{p_{21}}s \\ + a_{21}a_{12}s^2 - a_{21}b_{11}c_{42}k_{p_{14}}s^2$$

$$\begin{aligned}
\det[s\mathbf{I} - \mathbf{A}_d] = & \\
& + (1)s^4 \\
& - (a_{11} - b_{12}c_{11}k_{p_{21}} - b_{21}c_{42}k_{p_{14}})s^3 \\
& - (a_{11}b_{21}c_{42}k_{p_{14}} - b_{12}c_{11}k_{p_{21}}b_{21}c_{42}k_{p_{14}} - b_{21}c_{42}k_{i_{14}} + a_{21}a_{12} - a_{21}b_{11}c_{42}k_{p_{14}} - b_{12}c_{11}k_{i_{21}})s^2 \\
& - (-b_{12}b_{21}c_{42}k_{p_{14}}c_{11}k_{i_{21}} - a_{21}b_{11}c_{42}k_{i_{14}} + b_{21}c_{42}k_{i_{14}}a_{11} - b_{21}c_{42}k_{i_{14}}b_{12}c_{11}k_{p_{21}})s \\
& - (-c_{11}k_{i_{21}}b_{12}b_{21}c_{42}k_{i_{14}})
\end{aligned}$$

The equation is of the form,

$$\det[s\mathbf{I} - \mathbf{A}_d] = C_1s^4 + C_2s^3 + C_3s^2 + C_4s + C_5$$

The coefficients of the characteristic equation are shown below.

$$\begin{aligned}
C_1 &= 1 \\
C_2 &= -(a_{11} - b_{12}c_{11}k_{p_{21}} - b_{21}c_{42}k_{p_{14}}) \\
C_3 &= -(a_{11}b_{21}c_{42}k_{p_{14}} - b_{12}c_{11}k_{p_{21}}b_{21}c_{42}k_{p_{14}} - b_{21}c_{42}k_{i_{14}} + a_{21}a_{12} - a_{21}b_{11}c_{42}k_{p_{14}} - b_{12}c_{11}k_{i_{21}}) \\
C_4 &= -(-b_{12}b_{21}c_{42}k_{p_{14}}c_{11}k_{i_{21}} - a_{21}b_{11}c_{42}k_{i_{14}} + b_{21}c_{42}k_{i_{14}}a_{11} - b_{21}c_{42}k_{i_{14}}b_{12}c_{11}k_{p_{21}}) \\
C_5 &= -(-c_{11}k_{i_{21}}b_{12}b_{21}c_{42}k_{i_{14}})
\end{aligned}$$

Since we have four gains and four coefficients, we are able to arbitrarily place the poles. Because the algebra is complex in the solution of these equations, we simplify the equations with following substitutions.

$$\begin{array}{lll}
C'_2 = C_2 + a_{11} & x_1 = +b_{12}c_{11} & x_6 = +a_{21}b_{11}c_{42} \\
C'_3 = C_3 + a_{21}a_{12} & x_2 = +b_{21}c_{42} & x_7 = +b_{12}c_{11} \\
C'_4 = C_4 & x_3 = -a_{11}b_{21}c_{42} & x_8 = +b_{12}b_{21}c_{42}c_{11} \\
C'_5 = C_5 & x_4 = b_{12}c_{11}b_{21}c_{42} & x_9 = +a_{21}b_{11}c_{42} - b_{21}c_{42}a_{11} \\
& x_5 = +b_{21}c_{42} & x_{10} = +b_{21}c_{42}b_{12}c_{11} \\
& & x_{11} = c_{11}b_{12}b_{21}c_{42}
\end{array}$$

Finally, we have the following system of equations.

$$\begin{aligned}
C_1 &= 1 \\
C_2' &= (x_1 k_{p_{21}} + x_2 k_{p_{14}}) \\
C_3' &= (x_3 k_{p_{14}} + x_4 k_{p_{21}} k_{p_{14}} + x_5 k_{i_4} + x_6 k_{p_{14}} + x_7 k_{i_{21}}) \\
C_4' &= (x_8 k_{p_{14}} k_{i_{21}} + x_9 k_{i_4} + x_{10} k_{i_4} k_{p_{21}}) \\
C_5' &= (x_{11} k_{i_{21}} k_{i_4})
\end{aligned}$$

Several techniques were tried to determine the best way to solve these equations. One method, the Newton-Raphson method was originally employed but was finally discarded because convergence was not guaranteed and it was computationally intensive. Other more sophisticated routines were also tried. Furthermore, optimal control techniques were explored where the Bass-Guru approach was discarded completely and replaced with a linear quadratic design using output feedback with cost function minimized using the Simplex algorithm. In the end, algebra was used to extract as much information from the system of equations as possible and then a simple iterative routine was created to determine a solution. This method proved to be satisfactory and is now outlined.

First, we can simplify the C_3' equation.

$$C_3' = ((x_3 + x_6)k_{p_{14}} + x_4 k_{p_{21}} k_{p_{14}} + x_5 k_{i_4} + x_7 k_{i_{21}})$$

adding C_2' to C_3' ,

$$C_3' = \left((x_3 + x_6)k_{p_{14}} + x_4 \frac{C_2' - x_2 k_{p_{14}}}{x_1} k_{p_{14}} + x_5 k_{i_4} + x_7 k_{i_{21}} \right)$$

$$C_3' = \left((x_3 + x_6)k_{p_{14}} + \left(\frac{x_4}{x_1} \right) (C_2' k_{p_{14}} - x_2 k_{p_{14}}^2) + x_5 \frac{C_5'}{x_{11} k_{i_{21}}} + x_7 k_{i_{21}} \right)$$

$$C_3' = \left((x_3 + x_6)k_{p_{14}} + \left(\frac{x_4}{x_1} \right) (C_2' k_{p_{14}} - x_2 k_{p_{14}}^2) + x_5 \frac{C_5'}{x_{11} k_{i_{21}}} + x_7 k_{i_{21}} \right)$$

We end up with a quadratic equation in terms of $k_{p_{14}}$ where $k_{i_{21}}$ is a term in the coefficients as shown in Equation (4.28).

$$x_2 k_{p_{14}}^2 - \left((x_3 + x_6) + C_2' \left(\frac{x_4}{x_1} \right) \right) k_{p_{14}} + \left(C_3' - x_5 \frac{C_5'}{x_{11} k_{i_{21}}} + x_7 k_{i_{21}} \right) = 0 \quad (4.28)$$

Using C_5' , C_4' , and C_2' we can get $k_{i_{21}}$ in terms of $k_{p_{14}}$ as shown in Equation (4.29).

$$\begin{aligned}
C'_4 &= \left(x_8 k_{p_{14}} \left(\frac{C'_5}{x_{11} k_{i_{14}}} \right) + x_9 k_{i_{14}} + x_{10} k_{i_{14}} \left(\frac{C'_2 - x_2 k_{p_{14}}}{x_1} \right) \right) \\
C'_4 k_{i_{14}} &= \frac{C'_5 x_8 k_{p_{14}}}{x_{11}} + x_9 k_{i_{14}}^2 + \frac{x_{10}}{x_1} k_{i_{14}}^2 (C'_2 - x_2 k_{p_{14}}) \\
C'_4 k_{i_{14}} &= \frac{C'_5 x_8 k_{p_{14}}}{x_{11}} + x_9 k_{i_{14}}^2 + \frac{x_{10} C'_2}{x_1} k_{i_{14}}^2 - x_2 k_{p_{14}} k_{i_{14}}^2 \\
\frac{C'_5 x_8 k_{p_{14}}}{x_{11}} + x_9 k_{i_{14}}^2 + \frac{x_{10} C'_2}{x_1} k_{i_{14}}^2 - x_2 k_{p_{14}} k_{i_{14}}^2 - C'_4 k_{i_{14}} &= 0 \\
\left(x_9 + \frac{x_{10} C'_2}{x_1} - x_2 k_{p_{14}} \right) k_{i_{14}}^2 - C'_4 k_{i_{14}} + \frac{C'_5 x_8 k_{p_{14}}}{x_{11}} &= 0 \\
k_{i_{14}} &= \frac{C'_4 \pm \sqrt{C_4'^2 - 4 \left(x_9 + \frac{x_{10} C'_2}{x_1} - x_2 k_{p_{14}} \right) \frac{C'_5 x_8 k_{p_{14}}}{x_{11}}}}{2 \left(x_9 + \frac{x_{10} C'_2}{x_1} - x_2 k_{p_{14}} \right)} \\
k_{i_{21}} &= \frac{C'_5}{x_{11}} \frac{C'_4 \pm \sqrt{C_4'^2 - 4 \left(x_9 + \frac{x_{10} C'_2}{x_1} - x_2 k_{p_{14}} \right) \frac{C'_5 x_8 k_{p_{14}}}{x_{11}}}}{2 \left(x_9 + \frac{x_{10} C'_2}{x_1} - x_2 k_{p_{14}} \right)} \tag{4.29}
\end{aligned}$$

Since we know that $k_{p_{14}}$ is always positive and on the order of 10^{-4} , a simple iteration algorithm was developed that begins with an initial $k_{p_{14}}$, calculates a $k_{i_{21}}$ using Equation (4.29), and then tests the solution using Equation (4.28). Generally, the method yields two sets of workable gains. The solution chosen of the two workable sets is the solution with the lowest throttle gains. This way the throttle is modulated the least.

5 The Supporting Functional Logic of the Longitudinal Control System

In the previous chapter, we developed the feedback control algorithms that stabilize the aircraft and drive it to the desired condition. The result was our basic, linear control model, equation (4.2), which can be expressed in matrix/operator notation as,

$$\mathbf{u} = \mathbf{K}\mathbf{e} \quad (5.1)$$

where \mathbf{u} is the control vector (in our case, lift coefficient and thrust), \mathbf{e} is the output error vector (which contains the indicated airspeed, Mach number, altitude, and altitude rate), and \mathbf{K} is the constant matrix operator of proportional, integral, and feedback gains. Our goal is to squeeze standard piloting strategies into that basic, linear model and use it to control our non-linear dynamics.

But a basic, linear control model is too simple to capture the complexities of piloting an aircraft throughout its entire flight envelope. In general, a pilot will provide different control inputs for a given error vector based on the flight regime. As an illustration of the different piloting strategies used in different flight regimes, consider the following.

- During take-off, the pilot sets take-off thrust and a nominal stick location and holds them constant until the aircraft reaches its rotation speed.
- In climb, the pilot fixes the thrust at climb thrust and uses the control stick to capture either airspeed or altitude rate.
- In descent, the pilot fixes the thrust at descent thrust and uses the control stick to capture either airspeed or altitude rate. Our linear controller may be able to use the same lift coefficient controller as for climbs, but the fixed thrust value is different.
- In steady, level flight, the pilot uses the throttle to hold the airspeed at the desired value and the control stick to hold the altitude.

So there are cases in which there is no modulation of control inputs at all, cases in which only one control parameter is modulated, and cases in which they are both modulated. Since different flight regimes require different control strategies, we will need to develop a basic, linear controller for each flight regime. The purpose of this chapter is to define these different flight regimes, or **regions**, and the conditions that bound them, and to develop the algorithms for determining the desired condition within those regions.

In the previous chapter, we introduced the concept of the desired output. Because our system uses multiple regions and multiple controllers, we need to differentiate between the desired output that each mathematical controller sees and the commanded output that the system (in the real world this would be the pilot) is trying to attain. The commanded output, \mathbf{y}_c , typically comes from the user interface. Each region will use supporting functional logic to select a time-varying desired output, $\mathbf{y}_d(t)$, that defines a “path” to the

commanded output, y_c . The desired output is then passed to the region's controller and the controller determines the control inputs needed to follow that path.

5.1 Control Strategies

In this section, we briefly revisit the discussions of the previous chapter on the different control strategies used for different phases of flight and explicitly state the control law used for each.

5.1.1 Feeding Back Altitude Rate Only

A pilot will accomplish speed changes in level flight typically by fixing the throttle appropriately (advanced for speed increase, reduced for speed decrease) and using the control stick to maintain level flight. To insure that the aircraft stays in level flight, the pilot will rely primarily on the altimeter and the attitude indicator. In a sense, he is using feedback control on his altitude rate (with a desired rate of zero) and allowing the aircraft to accelerate (or decelerate).

The pilot can also use the control stick to gain a balance of airspeed acceleration and vertical speed. This amounts to dividing the changing total energy between changes in potential and kinetic energy. It is also possible to exchange potential and kinetic energy without affecting the aircraft's total energy much at all (e.g., descending and accelerating). By adjusting the control stick, the pilot can control how much energy goes to changing airspeed and how much goes to changing altitude.

The NextGen simulator can use altitude-rate-only feedback to accomplish speed and/or altitude changes in any combination using this control strategy. The only thing left to do is determine the desired altitude rate, \dot{h}_d . For altitude-rate-only feedback, equation (5.1) becomes

$$C_L(t) = k_{p_{h_4}} (\dot{h}_d(t) - \dot{h}(t)) + \int k_{i_{h_4}} (\dot{h}_d(t) - \dot{h}(t)) dt - k_{b_{h_4}} \dot{h}(t) \quad (5.2)$$

5.1.2 Feeding Back Speed Only

Climbs and descents at constant airspeed are also typically accomplished without much modulation of the throttle. For climbs the throttle is advanced to the desired climb power, and for descents the power is reduced to idle. The control stick is then adjusted to maintain the proper airspeed while the altitude is allowed to change. The pilot uses information from the airspeed (or Mach) indicator to adjust the control stick to maintain speed. This type of control is fundamentally different from the speed change in level flight because the control stick is now controlling speed instead of altitude rate. Likewise, the throttle is controlling the climb rate of the aircraft. Notice that this amounts to a reversal in the roles of the two controls.

The NextGen simulator can use speed-only feedback to accomplish altitude changes using this control strategy. The only thing left to do is determine the desired speed, V_{IAS} , or Mach number, M . For indicated airspeed feedback, equation (5.1) becomes

$$C_L(t) = k_{p_{11}} \left(V_{IAS_d}(t) - V_{IAS}(t) \right) + \int k_{i_{11}} \left(V_{IAS_d}(t) - V_{IAS}(t) \right) dt - k_{b_{11}} V_{IAS}(t) \quad (5.3)$$

and for Mach feedback, equation (5.1) becomes

$$C_L(t) = k_{p_{12}} \left(M_d(t) - M(t) \right) + \int k_{i_{12}} \left(M_d(t) - M(t) \right) dt - k_{b_{12}} M(t) \quad (5.4)$$

5.1.3 Feeding Back Speed and Altitude Rate

A pilot will maintain steady, level flight typically by using the control stick to fly level and adjusting the throttle to hold the desired speed. To insure that the aircraft stays in level flight, the pilot will monitor the altimeter and the attitude indicator. To insure that the aircraft holds speed, the pilot will monitor the airspeed (or Mach) indicator.

The NextGen simulator can use this strategy to maintain steady, level flight, to fly a constant vertical speed at a specified airspeed, or to follow a specific speed-altitude profile, as on approach and landing. The only thing left to do is determine the desired speed and altitude rate. For IAS-based control, equation (5.1) becomes

$$\begin{aligned} C_L(t) &= k_{p_{14}} \left(\dot{h}_d(t) - \dot{h}(t) \right) + \int k_{i_{14}} \left(\dot{h}_d(t) - \dot{h}(t) \right) dt - k_{b_{14}} \dot{h}(t) \\ T(t) &= k_{p_{21}} \left(V_{IAS_d}(t) - V_{IAS}(t) \right) + \int k_{i_{21}} \left(V_{IAS_d}(t) - V_{IAS}(t) \right) dt - k_{b_{21}} V_{IAS}(t) \end{aligned} \quad (5.5)$$

and for Mach feedback, equation (5.1) becomes

$$\begin{aligned} C_L(t) &= k_{p_{14}} \left(\dot{h}_d(t) - \dot{h}(t) \right) + \int k_{i_{14}} \left(\dot{h}_d(t) - \dot{h}(t) \right) dt - k_{b_{14}} \dot{h}(t) \\ T(t) &= k_{p_{22}} \left(M_d(t) - M(t) \right) + \int k_{i_{22}} \left(M_d(t) - M(t) \right) dt - k_{b_{22}} M(t) \end{aligned} \quad (5.6)$$

5.1.4 Altitude Capture

Capturing altitude is a fundamental function that the longitudinal dynamics must perform; however, altitude is never a feedback parameter directly. Instead altitude rate is commanded in a manner such that altitude capture is obtained. Initially it was not clear that this was the best solution to the problem. In fact many direct altitude feedback strategies were tried, and the state space model still allows for altitude feedback. However, instead of direct altitude feedback, the altitude error is used to determine an appropriate value for \dot{h}_d , the desired altitude rate. The reason for this decision is based primarily on the need for a smooth transition between a control region that does not control altitude rate, and one that does.

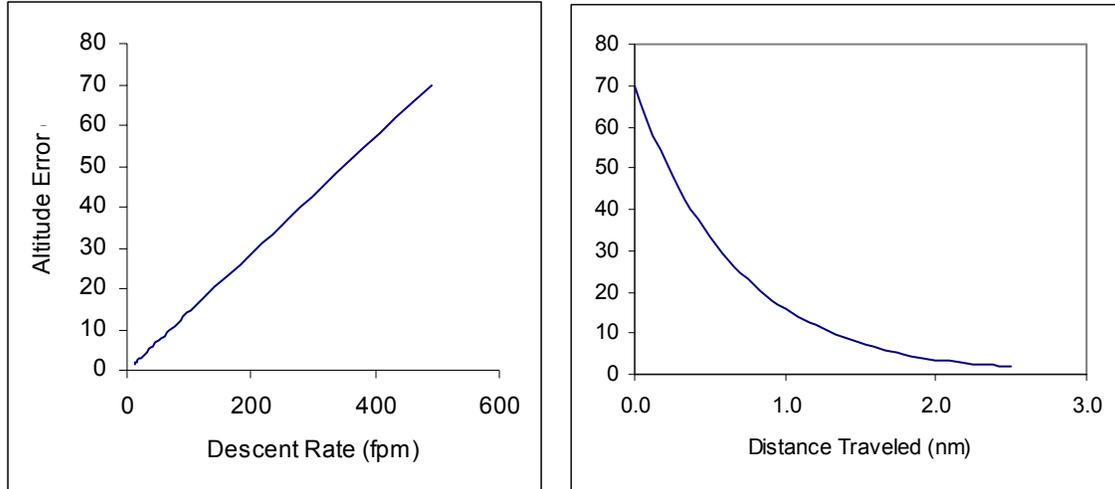


Figure 5.1: Illustration of an aircraft capturing an altitude

For regions in which the altitude rate is captured, the desired altitude rate is calculated by applying a gain to the difference between the desired altitude and the actual altitude.

$$\dot{h}_d = K_{\dot{h}} (h_d - h) \quad (5.7)$$

We set a nominal value of $K_{\dot{h}} = 7 \text{ } 1/\text{min}$. Figure 5.1 shows how the descent rate varies over the final 70 feet and the smooth capture of the altitude when $K_{\dot{h}} = 7$.

There is still a problem to contend with at region transition. Consider an aircraft in, descent at idle throttle at a specified airspeed as shown in Figure 5.2. Since there is no direct control over the rate of descent, the aircraft descends at whatever rate is required to maintain the commanded airspeed with an idle throttle. There is likely to be an undesired transient when the aircraft makes the transition to a region with controlled altitude rate. The cause of the undesirable transient is the fact that the aircraft's altitude rate upon entering the new region and the desired altitude rate derived from Equation (5.7) do not necessarily match. The mismatch causes the control law to drive this initial 'error' to zero with excessive control inputs. Figure 5.2 shows the mismatch and the sudden increase in the descent rate.

To solve this problem, the initial desired altitude rate is set equal to the aircraft's current altitude rate. The measured (current) altitude and altitude rate are used to calculate the value of $K_{\dot{h}}$ by rearranging Equation (5.7).

$$K_{\dot{h}_{\text{initial}}} = \frac{\dot{h}_{\text{current}}}{(h_d - h)_{\text{current}}} \quad (5.8)$$

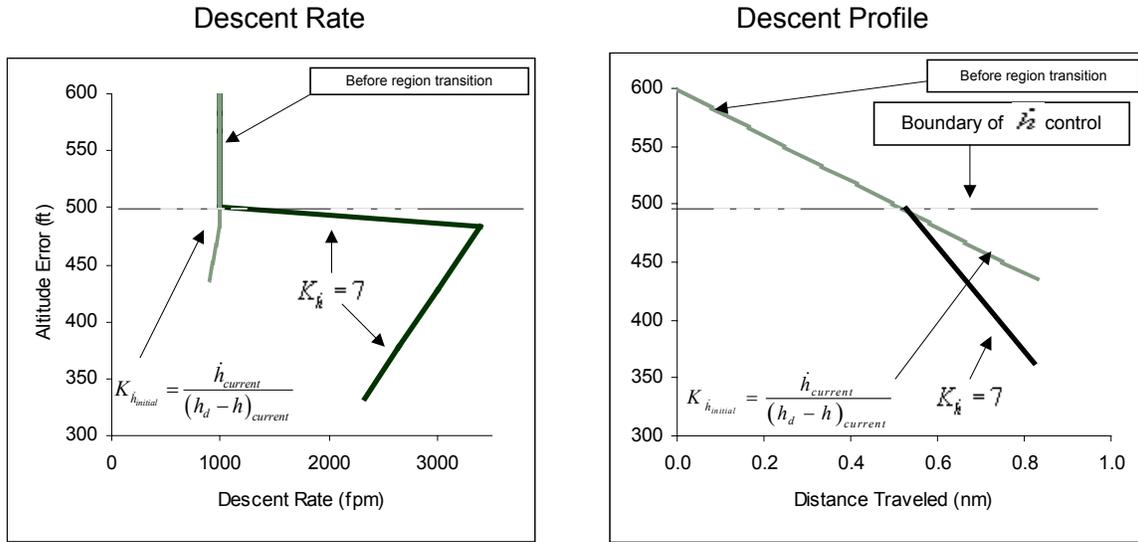


Figure 5.2: Altitude Rate During Region Transition

When the aircraft is within 70 feet of the target altitude, K_h is returned to its nominal value of seven regardless of the initial calculated value. Seventy feet is chosen because it yields an altitude rate near 500 fpm, a standard value within the simulator. This change in K_h does yield an unrealistic jump in the altitude rate just before altitude capture but it is much smaller than at region transition and it ensures that the altitude is captured in a timely manner. It also prevents the previous descent or climb from affecting the continuing cruise performance of the aircraft that would come from preserving either an unusually low or high K_h . Figure 5.3 shows the logic for calculating K_h .

5.1.5 Speed Capture

The feedback control laws were derived for small errors. In the steady-state control region, which uses speed control, the speed error is never more than 10 knots. This is a suitable speed to keep the control inputs from being excessive. But in the climb and descent regions, which also use speed control, it is possible for the speed error to be 100 knots or more. Our controller would produce excessive control inputs to correct this error.

To avoid these excessive control inputs, we must devise a varying desired speed profile that will produce acceptable control inputs and speed rates. A constant desired acceleration of one knot per second was chosen for its simplicity and suitability in producing an acceptable transient response. The initial desired speed upon entering a speed-controlled region is the actual speed.

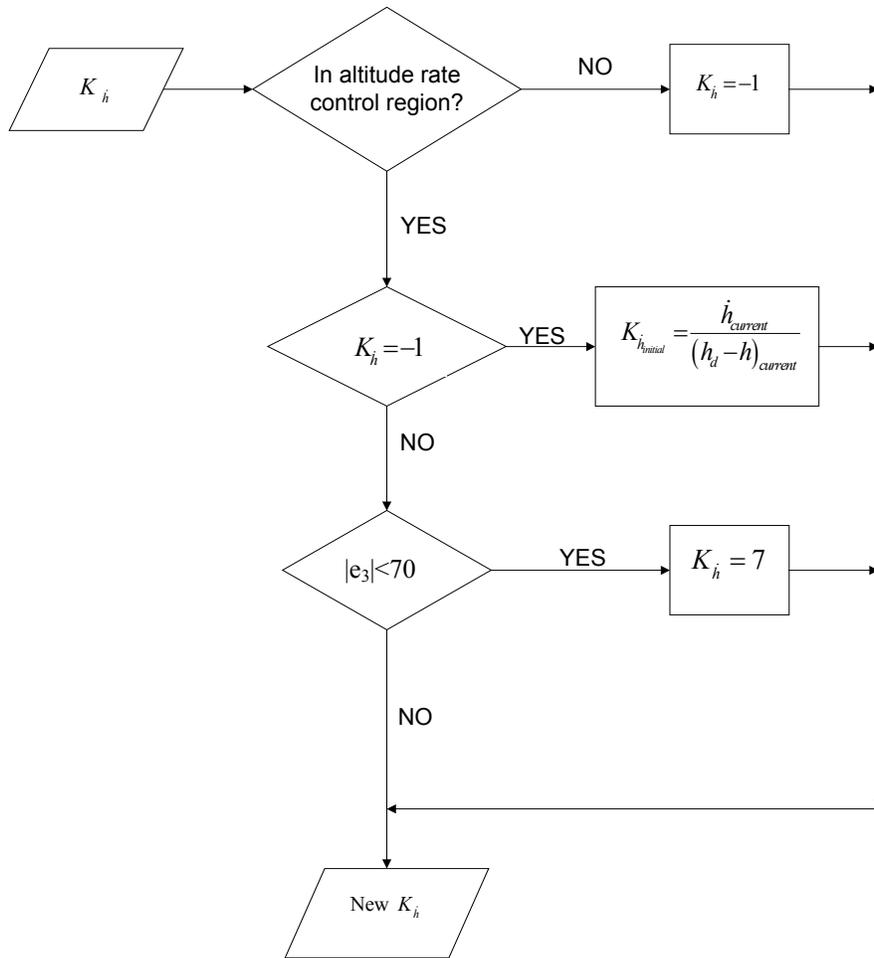


Figure 5.3. Flow diagram for calculating K_h

$$\begin{aligned}
 V_{IAS_{d0}} &= V_{IAS} \\
 V_{IAS_d} &= V_{IAS_d} + \dot{V}_{IAS_d} \Delta t
 \end{aligned}
 \tag{5.9}$$

where

$$\dot{V}_{IAS_d} = \left(1 \frac{kt}{sec}\right) * \text{sign}(V_{IAS_c} - V_{IAS})$$

The function $sign(x)$ returns $+1$ if $x \geq 0$ and -1 otherwise. Once the indicated airspeed is within five knots of the commanded value, the desired speed is simply the commanded speed.

For Mach control, the equations for desired Mach are similar. A constant desired acceleration of 0.02 Mach per second is selected. It is equivalent to an acceleration of one knot per second at 25,000 feet altitude and 300 knots.

$$\begin{aligned}
 M_{d_0} &= M \\
 M_d &= M_d + \dot{M}_d \Delta t
 \end{aligned}
 \tag{5.10}$$

where

$$\dot{M}_d = \left(0.02 \frac{1}{sec} \right) * \text{sign}(M_c - M)$$

5.2 Dividing the Flight Envelope into Regions

Because of the need for different controllers and different supporting logic in different regimes of the flight envelope, we need to divide the flight envelope into regions. Each region's control law, supporting logic, and desired output reflect the pilot's decision logic in bringing the aircraft from its actual state to its desired state.

All of the supporting functional logic for the longitudinal control system centers on the concept of the speed-altitude plane. The speed-altitude plane was used by Mukai [Mu92] during the development of *Pseudocontrol*, the original control system developed for the Pseudo Aircraft Simulation (PAS) system developed for NASA Ames. The speed-altitude plane has been revisited and adapted for the TGF project and has undergone extensive modification since it was used in PAS.

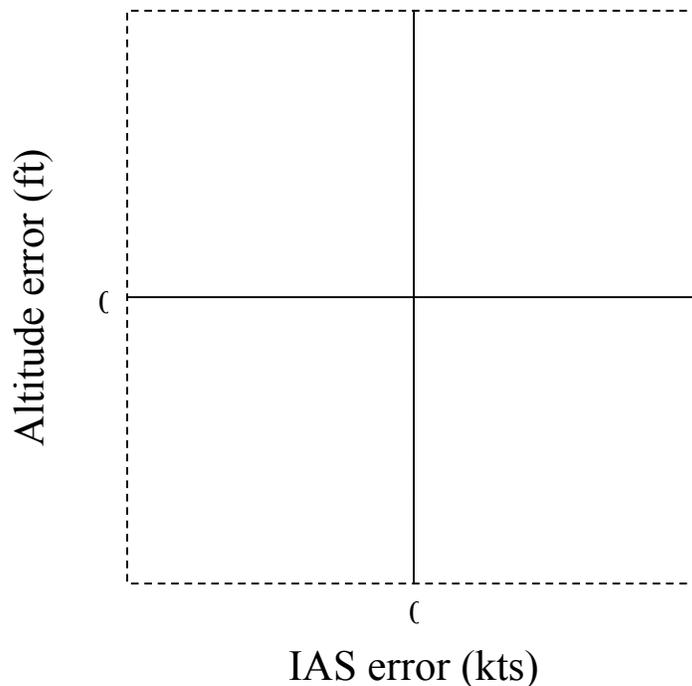


Figure 5.4. The Speed-Altitude Plane

5.2.1 The Speed-Altitude Plane

The speed-altitude plane is used to represent the aircraft's actual position relative to its commanded position, with altitude error plotted on the vertical axis and speed error plotted on the horizontal axis. It allows us to split the aircraft's en route flight envelope into different control regions. A diagram of the speed-altitude plane is shown in Figure 5.4.

The basic purpose of the speed-altitude plane is to emulate the way a pilot makes decisions about flying an aircraft. We define an area in the immediate vicinity of the commanded condition in which the aircraft is in steady, level flight. We set up speed error and altitude error bounds to define this steady, level flight region. We then need to define what to do outside of these bounds in order to bring the aircraft within them.

The speed-altitude plane provides a means of generalizing these relationships and also provides insight into what the throttle setting should be based on the aircraft's current energy level.

5.2.1.1 The Line of Constant Energy

The line of constant energy illustrates how an aircraft can have the same energy level at different speeds and altitudes. It is a means of determining whether the aircraft is low on energy or high on energy with respect to its commanded condition; that is, does the aircraft need to add thrust or reduce thrust to attain the commanded condition. Originally, the line of constant energy was based on true energy calculations for the aircraft but as it evolved it became simply a gross representation of the aircraft's energy. This section discusses the evolution of the line of constant energy.

The total energy of an aircraft is the sum of its potential and kinetic energies.

$$E = K.E. + P.E. = mgh + \frac{1}{2} mV_a^2$$

The terms are defined as follows:

- E : The total aircraft energy
- m : The aircraft mass
- g : The gravitational acceleration
- h : The altitude
- V_a : The true airspeed

The equation can also be written in terms of the energy per unit mass.

$$e = gh + \frac{1}{2} V_a^2 \tag{5.11}$$

The energy that the aircraft would have at some commanded state can be written similarly,

$$e_c = gh_c + \frac{1}{2}V_{a_c}^2 \quad (5.12)$$

where e_c is the total energy at the commanded state, h_c is the commanded altitude, and V_{a_c} is the commanded true airspeed. Notice that it is quite possible for $e_c = e$ without the aircraft actually being at the commanded state. That is to say, the aircraft could have the right amount of energy but be either fast and low or high and slow. This is illustrated in Figure 5.5 which shows the line of constant energy. The x - and y -axes on Figure 5.5 are the speed error and altitude error from some commanded state shown at the point $(0,0)$.

If the aircraft's current state lies on the constant energy line, the aircraft already has enough energy to attain the commanded state. Therefore, the amount of throttle adjustment needed is minimal. However, if the aircraft lies below the constant energy line, the aircraft needs energy to attain the commanded state. Likewise, if the aircraft is above the energy line, the aircraft has excess energy and must lose energy to achieve the commanded state.

The constant energy curve is a parabola, as illustrated by the quadratic relationship between altitude and speed in equation (5.11). For practical control implementation, linear approximation to the constant energy curve in the speed-altitude plane is used, as shown in Figure 5.5. However, even this approximation is simplified because of transition problems between regions. The final form of the line of constant energy approximates the actual energy curve by forming a diagonal across the steady, level flight region, as shown in Figure 5.6.

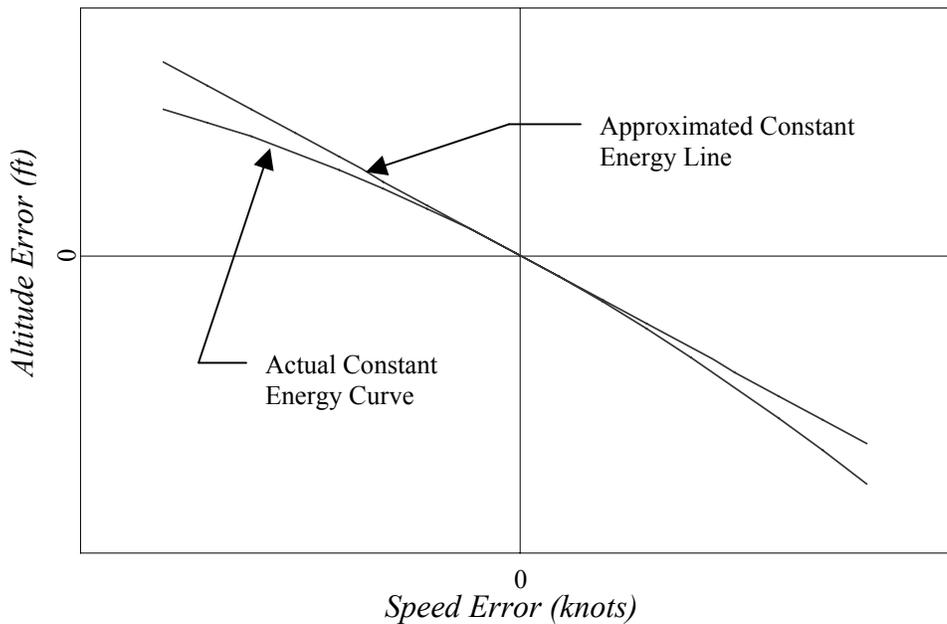


Figure 5.5. The constant energy line on the speed-altitude plane

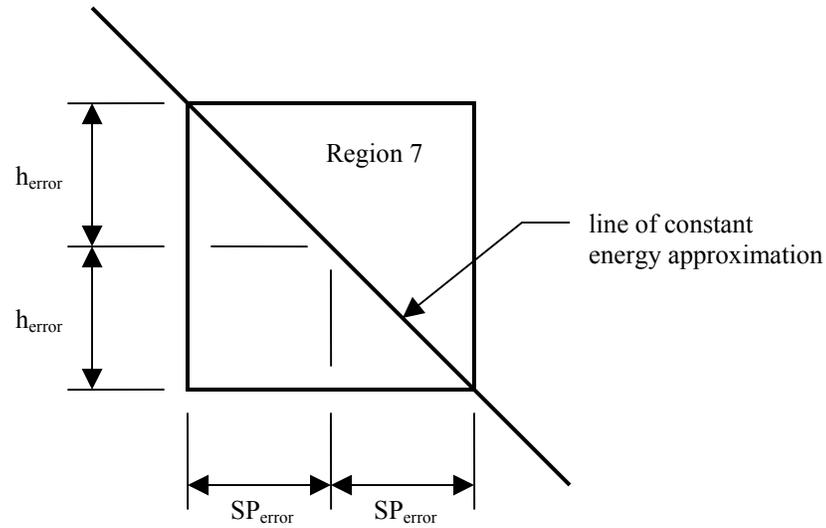


Figure 5.6. Illustration of the approximation for the constant energy line using the diagonal cut across the steady, level flight region (region 7) of the speed-altitude plane

5.2.1.2 Bounding the Speed-Altitude Plane

The speed error bound on the steady, level flight region is 10 knots. The simulation engineers chose this value because it is a bound typically used in real flight; an aircraft within ten knots of a controller-assigned speed is considered to be in compliance with that speed. The simulation engineers chose the altitude error bounds so that the slope of the diagonal of the steady, level flight region is the same as the slope of the constant energy line in the vicinity of a flight condition that is representative of en route flight: 300 knots IAS @ 25,000 ft.

At 25,000 feet altitude in the standard atmosphere, 300 knots IAS corresponds to 728 fps true airspeed (TAS) and a ten knot IAS speed difference corresponds to about a 22 fps speed difference in TAS. The slope of the constant energy curve at this point is given by the derivative of equation (5.11). Assuming that gravity is constant,

$$d(e) = d(gh) + d\left(\frac{1}{2}V_a^2\right)$$

$$de = g dh + V_a dV_a$$

Using discrete notation, and substituting in the values corresponding to our speed change and flight condition, we can calculate the corresponding altitude change that would keep the total energy constant.

$$\Delta e = g \Delta h + V_a \Delta V_a$$

$$(0) = \left(32.2 \frac{ft}{s^2}\right) \Delta h + \left(728 \frac{ft}{s}\right) \left(22 \frac{ft}{s}\right)$$

$$\Delta h = 497 \frac{ft}{s}$$

We choose to round to 500 feet for the altitude error.

The parameters needed to define the regions of the speed-altitude plane are the speed error, altitude error, and slope of the constant energy line. These parameters nominally are set as follows:

- h_{error} : This is the altitude error used to bound the steady, level flight region and, consequently, the slope of the constant energy line of the speed-altitude plane. The nominal value is 500 feet.
- SP_{error} : This is the speed error used to bound the steady, level flight region. It needs to be defined in terms of knots for IAS-based control and in terms of Mach number for Mach-based control.
 - IAS-based: $SP_{error} = IAS_{error} = 10 \text{ knots}$. Aircraft are typically expected to hold their speeds within 10 knots.
 - Mach-based: $SP_{error} = M_{error} = 0.022$. This is the Mach error used to bound the steady, level flight region in the Mach speed-altitude plane. The value corresponds to a 10 knot IAS speed change at 300 knots IAS and 25,000 feet in the standard atmosphere.

For complete definition, the speed-altitude plane also requires the slope of the diagonal line that roughly approximates the aircraft's energy relative to its commanded state. There are two slopes: one is for Mach-based control, and the other is for IAS-based control.

$$m_{IAS} = \frac{-h_{error}}{IAS_{error}} \quad (5.13)$$

$$m_{Mach} = \frac{-h_{error}}{M_{error}} \quad (5.14)$$

5.2.1.3 The Regions of the Speed-Altitude Plane

The speed-altitude plane is divided into nine different regions. Each region has a different combination of control law and supporting functional logic. The speed error on the speed-altitude plane is either represented in knots of indicated airspeed or in Mach number. Figure 5.7 shows the regions of the speed-altitude plane in terms of indicated airspeed. The regions are enumerated below with their various functions.

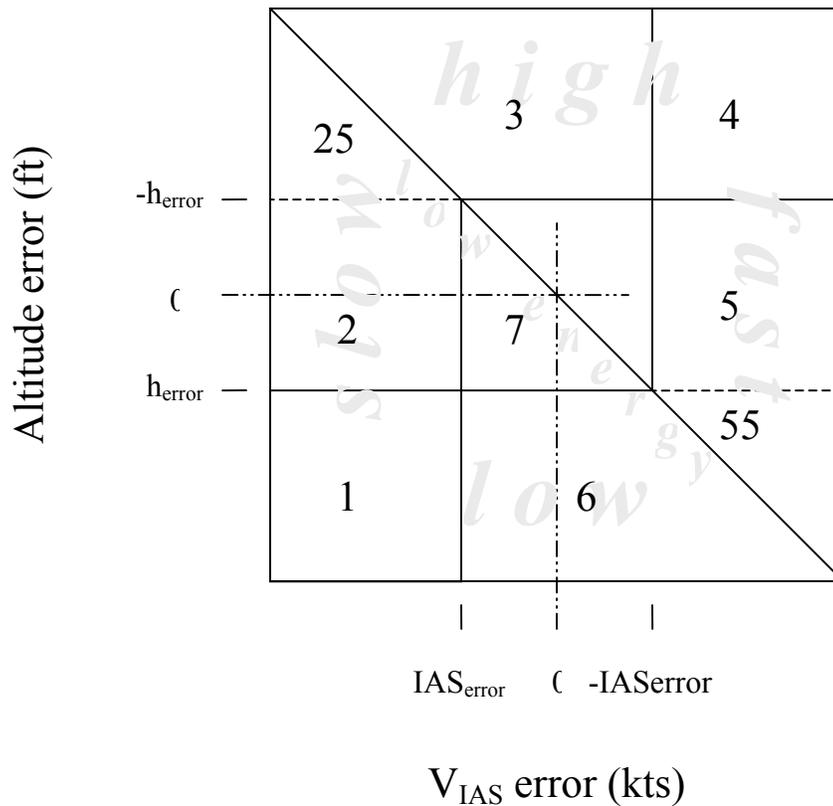


Figure 5.7. The speed-altitude plane in terms of indicated airspeed

- **Steady, Level Flight (Region 7):** In Region 7, the aircraft is sufficiently close to converging on a desired state. The pilot modulates the control stick and the throttle simultaneously to capture the desired state.
- **Climbing & Accelerating (Region 1):** In Region 1, the aircraft is low and slow and is, therefore, low on energy. The throttle is set to full and the pilot climbs and accelerates the aircraft, using the control stick to maintain a balance of airspeed acceleration and climb rate.
- **Level Acceleration (Region 2):** In Region 2, the aircraft is slow enough to be low on energy; therefore, the throttle is set to full. The pilot uses the control stick to capture and maintain the desired altitude while accelerating into Region 7. Region 25 (Descending & Accelerating) is a subset of Region 2 that uses a constant desired altitude rate.
- **Steady Descent (Region 3):** In Region 3, the aircraft is high enough to have excess energy; therefore, the throttle is set to idle. The pilot uses the control stick to capture and maintain the desired speed while descending into Region 7.
- **Descending & Decelerating (Region 4):** In Region 4, the aircraft is high and fast and is, therefore, high on energy. The throttle is set to idle and the pilot descends and decelerates the aircraft, using the control stick to maintain a balance of airspeed deceleration and descent rate that is weighted towards descending.

- **Level Deceleration (Region 5):** In Region 5, the aircraft is fast enough to have excess energy; therefore, the throttle is set to idle. The pilot uses the control stick to capture and maintain the desired altitude while decelerating into Region 7. Region 55 (Climbing & Decelerating) is a subset of Region 5 that uses a constant desired altitude rate.
- **Steady Climb (Region 6):** In Region 6, the aircraft is low enough to be low on energy; therefore, the throttle is set to full. The pilot uses the control stick to capture and maintain the desired speed while climbing into Region 7.

The reader should notice a similarity between the basic strategy in each of the regions and the discussion of Section 4.1.1, where the basic piloting strategies for different types of maneuvers are outlined. The speed-altitude plane is a means of mechanizing the control strategy that a pilot would use depending on the aircraft's state relative to the commanded state.

5.2.1.4 Region Management in the Speed-Altitude Plane

The error between the aircraft's commanded and actual states is compared to the error bounds of the speed-altitude plane. This determines the aircraft's control region in the speed-altitude plane. The absolute errors between the aircraft's actual and commanded states, e_1 through e_4 , are defined below.

$$e_1 = V_{IASc} - V_{IAS} \quad (5.15)$$

$$e_2 = M_c - M \quad (5.16)$$

$$e_3 = h_c - h \quad (5.17)$$

$$e_4 = \dot{h}_c - \dot{h} \quad (5.18)$$

The terms are defined as follows:

- V_{IASc} : The commanded indicated airspeed (kts)
- V_{IAS} : The actual indicated airspeed (kts)
- M_c : The commanded Mach number
- M : The actual Mach number
- h_c : The commanded altitude (ft)
- h : The actual altitude (ft)
- \dot{h}_c : The commanded altitude rate (ft/min)
- \dot{h} : The actual altitude rate (ft/min)

We can use Boolean expressions to define four true-false parameters that coincide with the bounds and definitions of the speed-altitude plane. These are used to aid the flow algorithm (illustrated below) that determines which region logic to use. For example, if the commanded speed is more than ten knots below the actual speed, the aircraft is defined to be *fast*. These *speed-altitude plane Booleans* are defined in Table 5.1.

<i>low</i>	$e_3 > h_{error}$
<i>high</i>	$e_3 < -h_{error}$
<i>slow</i>	$e_s > SP_{error}$
<i>fast</i>	$e_s < -SP_{error}$
<i>lowenergy</i>	$e_3 > (m * e_s)$

Table 5.1: Speed - Altitude Plane Booleans

This section describes the control strategies and supporting functional logic used in each of the regions of the speed-altitude plane.

5.2.1.4.1 Steady, Level Flight (Region 7)

Region 7 is the most highly used region because it represents the aircraft at or near steady, level flight. The aircraft is close enough to a desired state that our linear controller can capture that state without excessive control inputs. Both the lift coefficient and the thrust are controlled. The control law is given by equations (5.5) and (5.6) for IAS-based and Mach-based control, respectively. The desired altitude rate is given by equation (5.7) and the desired speed is given by equations (5.9) and (5.10) for IAS-based and Mach-based control, respectively.

5.2.1.4.2 Accelerating (Region 2)

In Region 2, the aircraft is slow enough to be low on energy, but it may be above or below its commanded altitude. Because the aircraft is low on energy, the throttle is advanced to full throttle. The system uses the lift coefficient to capture and maintain the desired altitude rate while accelerating into Region 7. The control law is given by equation (5.2) with the desired altitude rate given by equation (5.7) (the same as for Region 7).

5.2.1.4.3 Descending & Accelerating (Region 25)

Region 25 is a subset of Region 2 that uses a constant descent rate; that is, the greater of its current descent rate and a 500 fpm descent rate. (Note the use of the **min** function because we are working with negative numbers.)

$$\dot{h}_d = \min(\dot{h}, -500 \text{ fpm})$$

5.2.1.4.4 Decelerating (Region 5)

In Region 5, the aircraft is fast enough to be high on energy, but it may be above or below its commanded altitude. It has more energy than it needs so the throttle is reduced to idle. It is the same as Region 2 except for the throttle setting. The system uses the lift coefficient to capture and maintain the desired altitude rate while decelerating into

Region 7. The control law is given by equation (5.2) with the desired altitude rate given by equation (5.7) (the same as for Region 7).

5.2.1.4.5 Climbing & Decelerating (Region 55)

Region 55 is a subset of Region 5 that uses a constant descent rate; that is, the greater of its current altitude rate and 500 fpm.

$$\dot{h}_d = \max(\dot{h}, 500 \text{ fpm})$$

5.2.1.4.6 Descending (Region 3)

In Region 3, the aircraft is high enough to be high on energy. It may be above or below its desired speed. Because the aircraft is high on energy, the thrust is set to idle. The aircraft uses lift coefficient control to capture its desired speed while descending into Region 7. The control law is given by equation (5.3) and equation (5.4) for IAS-based and Mach-based control, respectively. The desired speed is given by equations (5.9) and (5.10) for IAS-based and Mach-based control, respectively.

5.2.1.4.7 Climbing (Region 6)

In Region 6, the aircraft is low enough to be low on energy. It may be above or below its desired speed. Because the aircraft is low on energy, the thrust is set to full. The aircraft uses lift coefficient control to capture its desired speed while climbing into Region 7. The control law is given by equation (5.3) and equation (5.4) for IAS-based and Mach-based control, respectively. The desired speed is given by equations (5.9) and (5.10) for IAS-based and Mach-based control, respectively.

5.2.1.4.8 Climbing & Accelerating (Region 1)

In Region 1, the aircraft is low and slow and is, therefore, low on energy. The throttle is set to full and the system uses altitude rate feedback to maintain a balance of airspeed acceleration and climb rate that is weighted towards acceleration. The control law is given by equation (5.2).

Since the system uses altitude rate feedback, we need to determine the desired altitude rate that will yield the balance of airspeed acceleration and climb rate mentioned above. With the throttle set to full, the aircraft's thrust is typically greater than its drag. This means that the aircraft's total energy is increasing. We need to determine how much of this energy increase goes towards accelerating and how much goes towards climbing. An equation relating acceleration and altitude rate is given by equation (2.59),

$$\dot{V}_a = \frac{T - D}{m} - g \sin \gamma = \frac{T - D}{m} - g \frac{\dot{h}}{V_a} \quad (2.59)$$

which can be written in the form of the changing energy.

$$\frac{(T-D)V_a}{m} = g\dot{h} + V_a\dot{V}_a \quad (5.19)$$

Note that this equation is presented in BADA as equation (3.1-1). Note also that this equation is the time derivative of equation (5.11). We can rewrite this equation to express an aircraft's desired altitude rate in terms of an **energy ratio**, ER .

$$\dot{h}_d = \frac{(T-D)V_a}{mg \left(1 + \frac{V_a\dot{V}_a}{g\dot{h}} \right)} = \frac{(T-D)V_a}{mg(1+ER)} \quad (5.20)$$

The energy ratio, ER , is an expression of the ratio of changing kinetic energy to changing potential energy. We can use it to express how much of the changing energy (i.e., the thrust energy being added to the system) goes towards changing speed and how much goes towards changing altitude. BADA uses the same approach, but expresses equation (5.20) in terms of an **energy share factor** as a function of Mach number. For acceleration in climb, BADA recommends an energy share factor that corresponds to an energy ratio of $ER = 2.3$, and that is what we use for our controller. With the desired altitude rate determined by equation (5.20) we can use an altitude rate controller to capture that altitude rate.

5.2.1.4.9 Descending & Decelerating (Region 4)

In Region 4, the aircraft is high and fast and is, therefore, high on energy. The throttle is set to idle and the system uses altitude rate feedback to maintain a balance of airspeed deceleration and descent rate that is weighted towards the descent. The control law is given by equation (5.2).

Just as in Region 1, we need to determine the desired altitude rate that will yield the balance of changing airspeed and altitude. For deceleration in descent, BADA recommends an energy share factor that corresponds to an energy ratio of $ER = 2.3$; however, our tests have shown that an energy ratio of $ER = 1.0$ is more suitable. To determine the desired altitude rate for Region 4, we use equation (5.20) with the throttle set to idle thrust and the energy ratio set to $ER = 1.0$.

5.2.1.5 Constant Vertical Speed Mode – Region 8

The aircraft operates in Region 8 when a vertical speed is commanded. For constant vertical speed, the pilot uses throttle to capture the desired airspeed and control stick to capture the desired altitude rate. The control law is given by equation (5.5) and equation (5.6) for IAS-based and Mach-based control, respectively. The aircraft is bumped into speed-altitude plane management if it cannot maintain the desired airspeed within SP_{error} .

The desired speed is given by equations (5.9) and (5.10) for IAS-based and Mach-based control, respectively.

5.2.2 Taking off

If an aircraft is initiated in the simulation as a take-off, the guidance module generates a sequence of legs that will take the aircraft to its low-altitude cruising speed and 6000 feet altitude. That sequence is the take-off ground run, rotation, lift-off, initial climb, and cruise at 6000 feet altitude. The take-off ground run, rotation, and lift-off legs are the “take-off” legs. An aircraft cannot enter either of these legs unless it is initiated as a take-off and it must successively satisfy the boundary conditions of each leg to progress into the initial climb. Additionally, these legs use their own controllers and are, therefore, given their own regions. The initial climb and level cruise legs use the region management of the speed-altitude plane.

5.2.2.1 Take-Off Ground Run (Region 12)

The take-off ground run region uses an open-loop controller to accelerate the aircraft until it is within ΔV_R of its rotation speed while keeping on the ground. The region specifies constant control inputs. A lift coefficient of zero is specified to keep the aircraft from lifting off during the ground run and take-off thrust is specified to give the aircraft its maximum acceleration.

$$\begin{aligned} C_L &= 0 \\ T &= C_{T_{TO}} T_{\max} \end{aligned} \tag{5.21}$$

Boundary Condition: $V_{IAS} < V_R - \Delta V_R$

The constant ΔV_R is defined the same for all aircraft. It represents the speed at which the simulated aircraft enters the rotation region, region 13. Conceptually, it is intended to represent the increment below rotations speed at which the pilot begins to pull back on the control stick to lift off the runway. Its value in NextGen is selected so as to provide ample time for the controller of region 13 to ramp up to the aircraft’s rotation lift coefficient.

$$\Delta V_R = 20 \text{ knots}$$

5.2.2.2 Rotation (Region 13)

The sole purpose of the rotation region is to ramp the lift coefficient up from zero to the aircraft’s rotation lift coefficient as a function of speed. At first look, this requires an open-loop controller that ramps the lift coefficient between these two points as a function of time, but this would require a unique controller form. NextGen programmers and engineers decided that it would be easiest to implement a controller of the same form as all the other controllers, i.e., of the form of equation (5.1). This requires the development of gains that cause the feedback controller to mimic our desired open-loop behavior.

Equation (5.1) is a shortened form of equation (4.2), which is rewritten here.

$$\mathbf{u}(t) = \mathbf{K}_p \mathbf{e}(t) + \int_0^t \mathbf{K}_i \mathbf{e}(t) dt - \mathbf{K}_b \mathbf{y}(t) \quad (4.2)$$

We desire lift coefficient behavior that increases steadily. Since indicated airspeed increases with time during the ground run, we simplify our controller by selecting that output only. We also specify that the desired speed is the rotation speed throughout the region. The single input, single output (SISO) system is then,

$$C_L(V_{IAS}(t)) = k_{p1} (V_R - V_{IAS}(t)) + \int_0^t k_{i1} (V_R - V_{IAS}(t)) dt - k_{b1} V_{IAS}(t) \quad (5.22)$$

The lift coefficient needed to lift the aircraft off the runway at rotation speed is dubbed the rotation lift coefficient.

$$C_{L_r} = \frac{W}{\frac{1}{2} \rho V_R^2 S}$$

We would like the lift coefficient to change from zero at the beginning of region 13 to the rotation lift coefficient when the aircraft reaches its rotation speed. At rotation speed, we want the rate of change of the lift coefficient with speed to flatten out so that we don't overshoot it and impose undo drag on the aircraft. We, therefore, specify the following conditions for our controller.

$$\begin{aligned} C_L(V_R - \Delta V_R) &= 0 \\ C_L(V_R) &= C_{L_r} \\ \frac{d}{dV_{IAS}}(C_L(V_R)) &= 0 \end{aligned}$$

The bound of region 13 is defined by aircraft speed only. Once the aircraft speed is greater than the rotation speed, control is passed to region 14. A lower bound is not specified; once control is passed from region 12, the aircraft stays within region 13 until the boundary condition is satisfied.

Boundary Condition: $V_{IAS} \leq V_R$

To simplify integration, we assume the indicated airspeed in region 13 has constant acceleration; i.e., it is of the following form.

$$V_{IAS} = \dot{V}_{IAS} t \quad (\dot{V}_{IAS} = constant) \quad (5.23)$$

We transform equations (5.22) and (5.23) into functions of the indicated airspeed.

$$dt = \frac{1}{\dot{V}_{IAS}} dV_{IAS}$$

$$C_L(V_{IAS}) = k_{p_{11}}(V_R - V_{IAS}) + \int_{V_R - \Delta V_R}^{V_{IAS}} k_{i_{11}}(V_R - V_{IAS}) \left[\frac{1}{\dot{V}_{IAS}} dV_{IAS} \right] - k_{b_{11}} V_{IAS}$$

$$C_L(V_{IAS}) = k_{p_{11}}(V_R - V_{IAS}) + \frac{k_{i_{11}}}{\dot{V}_{IAS}} \int_{V_R - \Delta V_R}^{V_{IAS}} (V_R - V_{IAS}) dV_{IAS} - k_{b_{11}} V_{IAS}$$

$$C_L(V_{IAS}) = k_{p_{11}}(V_R - V_{IAS}) + \frac{k_{i_{11}}}{\dot{V}_{IAS}} \left\{ [V_R((V_{IAS}) - (V_R - \Delta V_R))] - \left[\frac{(V_{IAS})^2 - (V_R - \Delta V_R)^2}{2} \right] \right\} - k_{b_{11}} V_{IAS}$$

We now have an equation relating the lift coefficient to the indicated airspeed.

$$C_L(V_{IAS}) = k_{p_{11}}(V_R - V_{IAS}) - \frac{k_{i_{11}}}{\dot{V}_{IAS}} \frac{(V_{IAS} - V_R)^2 - \Delta V_R^2}{2} - k_{b_{11}} V_{IAS} \quad (5.24)$$

The derivative of equation (5.24) is,

$$\frac{d}{dV_{IAS}}(C_L(V_{IAS})) = -k_{p_{11}} - \frac{k_{i_{11}}}{\dot{V}_{IAS}}(V_{IAS} - V_R) - k_{b_{11}} \quad (5.25)$$

The initial conditions plugged into equations (5.24) and (5.25) yield the following equations.

$$C_L(V_R - \Delta V_R) = k_{p_{11}}(V_R - [V_R - \Delta V_R]) - \frac{k_{i_{11}}}{\dot{V}_{IAS}} \frac{([V_R - \Delta V_R] - V_R)^2 - \Delta V_R^2}{2} - k_{b_{11}} [V_R - \Delta V_R] = 0$$

$$C_L(V_R) = k_{p_{11}}(V_R - [V_R]) - \frac{k_{i_{11}}}{\dot{V}_{IAS}} \frac{([V_R] - V_R)^2 - \Delta V_R^2}{2} - k_{b_{11}} [V_R] = C_{L_R}$$

$$\frac{d}{dV_{IAS}}(C_L(V_R)) = -k_{p_{11}} - \frac{k_{i_{11}}}{\dot{V}_{IAS}}([V_R] - V_R) - k_{b_{11}} = 0$$

$$k_{p_{11}} \Delta V_R - k_{b_{11}} [V_R - \Delta V_R] = 0$$

$$\frac{k_{i_{11}}}{\dot{V}_{IAS}} \frac{\Delta V_R^2}{2} - k_{b_{11}} V_R = C_{L_R}$$

$$-k_{p_{11}} - k_{b_{11}} = 0$$

This system yields the following solution for our region 13 controller.

$$\begin{aligned}
 k_{p_{11}} &= 0 \\
 k_{i_{11}} &= \frac{2\dot{V}_{IAS} C_{L_R}}{\Delta V_R^2} \\
 k_{b_{11}} &= 0
 \end{aligned} \tag{5.26}$$

The region 13 controller is given by equation (5.22).

5.2.2.3 Lift-Off (Region 14)

When the aircraft reaches its rotation speed, the pilot pulls back on the control stick to lift off the runway, but maintains take-off thrust and take-off flaps. He uses this configuration until he reaches the maneuver altitude. NextGen operates similarly: control is passed to region 14 at rotation speed and the aircraft remains in region 14 control until it reaches its maneuver altitude, set to 400 feet AGL in NextGen. Therefore, we need a controller that returns take-off thrust and uses the lift coefficient to control speed. In this sense, region 14 is similar to region 6 (constant speed climbs) except that it uses take-off thrust.

$$\begin{aligned}
 T &= C_{T_{TO}} T_{\max} \\
 V_d &= V_R
 \end{aligned}$$

Boundary Condition: $h \leq 400 \text{ ft AGL}$

The controller is given by equation (5.3). Additionally, as soon as we enter this region, we can raise the landing gear, which amounts to removing the gear drag coefficient $C_{D_{gear}}$ from the drag equation.

5.2.3 Landing Region Management

An aircraft that has received the command to begin the landing sequence will remain in the speed-altitude plane regions until it reaches its glide slope. Once the aircraft reaches the glide slope, it begins the landing sequence and progresses through regions 9, 10, and 11. A new commanded altitude will move control back to the speed-altitude plane manager (i.e., landing is aborted). It is assumed that Mach-based control is not used for any landing maneuver, so Mach-based controllers are not considered.

5.2.3.1 Approach (Region 9)

Because the aircraft is following the glide slope (a linear altitude profile) and a preferred speed profile, it needs a controller that feeds back speed and altitude rate. The control law is given by equation (5.5).

The only difference is in the development of the desired output. Region 9 specifies the aircraft's speed profile as a function of distance from the runway threshold. Alternatively, the aircraft can be speed restricted with a commanded speed. Either way, region 9 specifies a desired altitude rate that will have the aircraft descend along the glide slope.

The guidance module defines the altitude profile based on the aircraft's distance from the glide slope antenna of its assigned runway. The guidance module defines a distance, d_{GS} , as the distance parallel to the runway from the glide slope antenna to the aircraft. The aircraft's desired altitude is a function of that distance and the angle of the glide slope antenna's signal, γ_{GS} .

$$h_d = h_{GS} = d_{GS} \tan \gamma_{GS} \quad (5.27)$$

The aircraft's desired altitude rate must consider the local glide slope altitude as well as the aircraft's altitude relative to the glide slope. As in Region 7, equation (5.7) is used to determine the altitude that would correct the aircraft's altitude error, but we must also consider that the desired altitude is changing per equation (5.27). We add the derivative of equation (5.27) to equation (5.7) to get the desired altitude rate for Region 9.

$$\dot{h}_d = K_h (h_d - h) + \frac{d}{dt} (d_{GS}) \tan \gamma_{GS}$$

The time derivative of the aircraft's distance from the runway threshold is the aircraft's groundspeed projected into the plane of the localizer. If we assume that the aircraft captures the localizer at the same time as the glide slope, we needn't concern ourselves with the angle between the aircraft's ground path and the localizer. This is a reasonable assumption that makes the equation much simpler.

$$\dot{h}_d = K_h (h_d - h) + V_G \tan \gamma_{GS} \quad (5.28)$$

The speed profile is defined as a function of the aircraft's distance from the outer marker of its assigned runway. It was developed to be consistent with typical airport approach speed profiles. The intent is to have the aircraft at its preferred speed for descents below 10,000 feet altitude by the time it is 20 nautical miles from the outer marker of its assigned runway and at 10 knots above its landing speed at the outer marker. At 10 nautical miles from the outer marker, the aircraft should have lost 2/3 of the difference. The final 10 knots is bled off by the time the aircraft reaches the runway.

The guidance module defines a distance, d_{OM} , as the distance parallel to the runway from the outer marker to the aircraft. The guidance module then establishes an upper limit of the desired speed as a function of d_{OM} . The desired speed schedule is illustrated in Figure 5.8. The upper limit is defined by points at 20 nm out, 10 nm out, at the outer marker, and at the runway threshold, with linear slopes between them. The speeds are defined as follows:

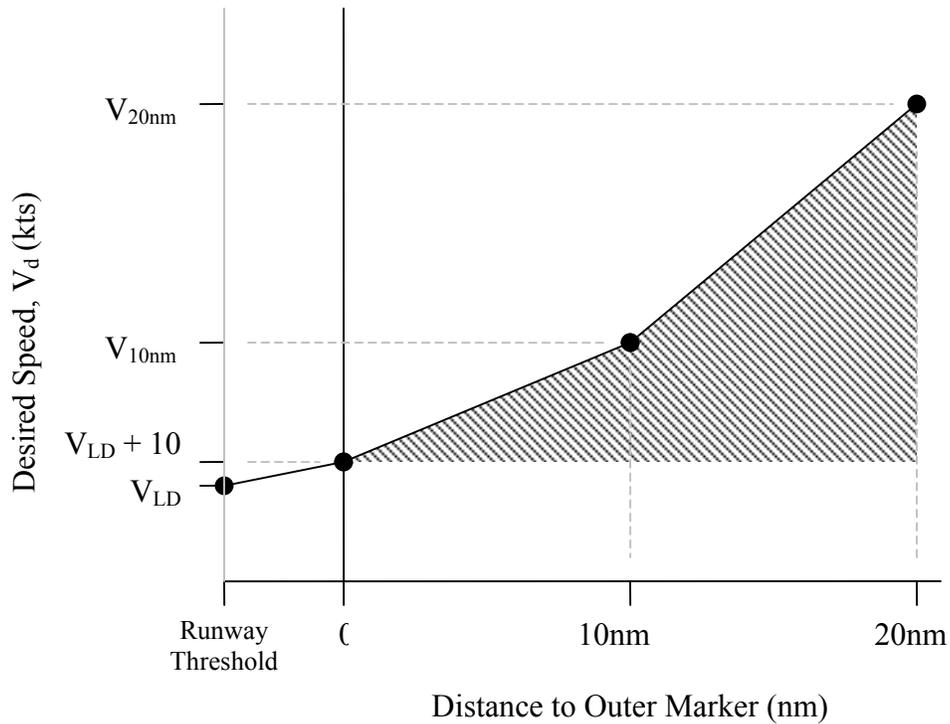


Figure 5.8: Desired Landing Speed Profile

$$V_{20nm} = \min(250 \text{ kts}, V_{des,1})$$

$$V_{LD} = 1.3V_{stall_{LD}}$$

$$V_{10nm} = \frac{V_{20nm} + 2(V_{LD} + 10 \text{ kts})}{3}$$

where $V_{des,1}$ is defined for each aircraft in the aircraft input file. If the aircraft is within the shaded area of Figure 5.8, it maintains its speed until it reaches the upper limit. If it is outside the shaded area, its desired speed is the limit speed.

5.2.4 Landing Flare (Region 10)

When the aircraft is within 100 ft above the runway, it begins the landing flare. The landing flare region (region 10) is a region of heightened control sensitivity to match the desired flare profile. The desired profile is a quadratic relationship between height above the runway and desired altitude rate. It was designed to be tangent to a 3° glide slope at 100 ft above the runway and to touch down on the runway at 1 ft/s.

$$\dot{h}_d = 0.0011(h - h_{rwy})^2 - 0.22(h - h_{rwy}) - 1$$

In this equation, height above the runway, $(h-h_{rwy})$, is in feet and the desired altitude rate, \dot{h}_d , is in ft/s. The desired speed is the aircraft's landing speed, V_{LD} . The control law is given by equation (5.5).

5.2.5 Landing Ground Run (Region 11)

Once the aircraft touches down, it enters the landing ground run region (region 11). In the landing ground run, the thrust is throttled back to idle, the lift coefficient is set to zero, and the drag coefficient is increased by the spoiler coefficient (to simulate the deployment of the spoiler). There is no feedback control in this region. When the speed is 20 knots below the landing speed, the aircraft is terminated from the simulation.

$$\begin{aligned} C_L &= 0 \\ T &= C_{T_{idle}} T_{max} \end{aligned} \tag{5.29}$$

5.3 Throttling in Regions 1 Through 6

When the thrust controller returns a thrust that differs from the current thrust, we must consider that the thrust difference may not be available in one time step; particularly when the difference is the full range from idle to max thrust, and particularly when the engine is a turbine engine. To account for this, a simple spooling lag has been added to the controller to limit the maximum amount that the thrust can change in a time step.

$$\Delta T_{max} = \frac{T_{max}}{k_{lag}} \Delta t \tag{5.30}$$

The term k_{lag} is the lag factor. Conceptually, it is the time it takes the engine to spool from zero thrust to maximum thrust. While spooling is not applicable to piston engines, k_{lag} is still used, just at a much smaller value. The current values for the different engines are presented below.

turbofans:	$k_{lag} = 20 \text{ sec}$
turboprops:	$k_{lag} = 5 \text{ sec}$
pistons:	$k_{lag} = 2 \text{ sec}$

We have chosen to model engine spooling in the longitudinal control logic rather than in the engine. Certainly, the real turbofan aircraft has spooling in the engine; however, spooling in the engine would introduce a troublesome nonlinearity into the open loop dynamics. Such nonlinearities would undoubtedly expand the control logic and increase the number of gains needed, and they would require extensive engineering effort to develop a sufficient control law. Furthermore, the location of the spooling has no bearing on the perceived motion of the aircraft.

5.4 Aircraft Device Deployment

5.4.1 Flaps

For all aircraft, the flaps (or, more appropriately, high-lift devices) are modeled, as they are in BADA, with lift and drag coefficient values corresponding to the flap and slat deployment for the different wing configurations for the aircraft. The five configurations are cruise, initial climb, take-off, approach, and landing. The algorithm for the deployment of flaps is presented in Figure 5.9. Basically, takeoff flaps are deployed for takeoff and retracted as the aircraft climbs, and flaps are sequentially extended during the landing sequence.

5.4.2 Speed Brakes

Speed brakes are modeled in the simulator as an increment, $C_{D_{brakes}}$, to the profile drag coefficient. Speed brakes are deployed if an aircraft is asked to expedite its descent. Algorithms within the simulator guidance module will deploy speed brakes in the case that the aircraft is predicted to be unable to reach an altitude-fix restriction. Speed brakes may also be deployed in the landing sequence if the aircraft is well above the glide slope.

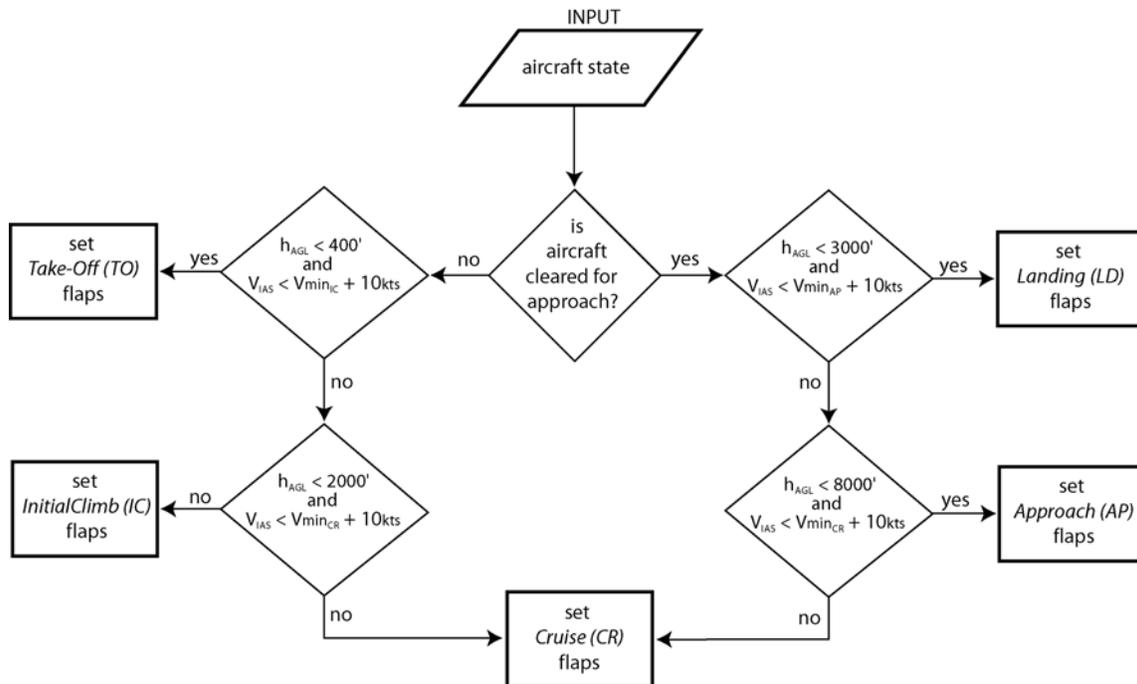


Figure 5.9: Flap Deployment Algorithm

5.4.3 Spoiler

The spoiler is modeled in the simulator as an increment, $C_{D_{spoiler}}$, to the profile drag coefficient and a complete loss of lift. It is used in the Touchdown region (Region 10) only and is deployed immediately when the aircraft's altitude is the same as the runway

altitude. Region 10 uses an open-loop controller with the thrust set to touchdown thrust and the lift coefficient set to zero (a consequence of spoiler deployment).

5.4.4 Landing Gear

Landing gear are modeled in the simulator as an increment, $C_{D_{gear}}$, to the profile drag coefficient. They are extended for the takeoff sequence and retracted once the aircraft reaches 400 feet AGL. They are also extended during the landing sequence once the aircraft has passed the outer marker of its assigned runway.

5.4.5 Ground Braking

When the braking function was initially conceived, it was thought that some number from a data file would be read into the airframe model, and an increase in drag would result from some static braking force. However, the aircraft data files did not have any information regarding braking force. Equation (5.31) is a proposed ground braking model that assumes that the braking force is 30% of the aircraft's weight. This approximation is convenient because it does not rely on an independent ground braking parameter in the aircraft data files. However, ground braking is not currently implemented in the TGF simulator.

$$D_{brake} = 0.3W_{ac} \quad (5.31)$$

THIS PAGE INTENTIONALLY LEFT BLANK

6. The Selection of Gains

In Section 3, we determined feedback control strategies for each of the regions in the speed-altitude plane. Furthermore, much effort was expended to develop a means to calculate acceptable gains for the different regions. The purpose for these computational methods was two-fold. First, there were many aircraft models to develop. Manually determining gains using root locus or bode techniques would be time consuming and would require a skilled controls engineer. Secondly, it was expected that each aircraft would need to have a schedule of gains to provide sufficient performance throughout the aircraft's flight envelope. Therefore, each aircraft would require gains to be calculated at many different reference conditions.

However, by carefully choosing the reference flight condition, it is possible to choose one set of gains that will work for the aircraft's entire flight envelope. This section documents the decision process that led to the final conclusion that gain scheduling would not be necessary.

6.1 The Aircraft's Flight Envelope

For a given aircraft weight, there are generally two parameters that define the aircraft's flight envelope: altitude and airspeed. The flight envelope, shown in Figure 6.1, illustrates how fast and slow the aircraft can fly and how high the aircraft can fly.

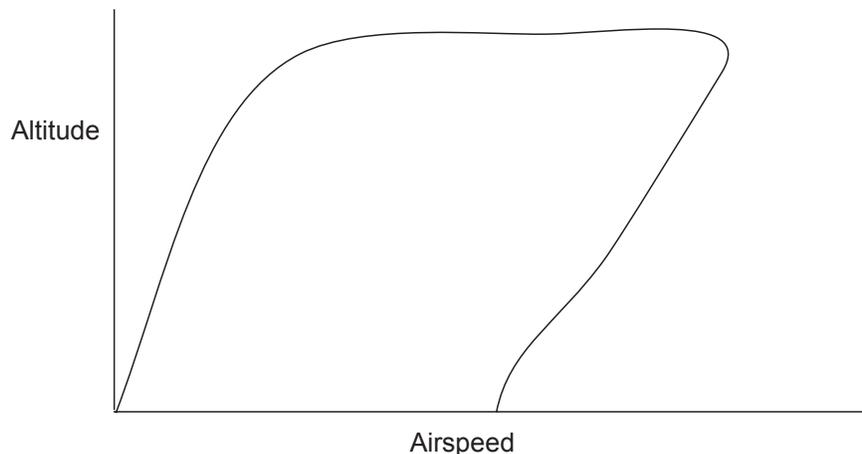


Figure 6.1. An example flight envelope

Of course, as the aircraft's weight changes, so do portions of the flight envelope of the aircraft; therefore, there are three parameters that affect the aircraft's dynamic condition.

In an effort to remove one of these parameters, Section 3 demonstrated that the flight envelope could be represented with true airspeed and lift coefficient rather than true airspeed, altitude and weight. The phugoid dynamics could be represented exclusively in terms of lift coefficient and true airspeed if a trimmed aircraft is assumed. The only term

that varied with weight was the thrust control derivative, which contributes to the forcing function of the system.

Therefore, the flight envelope can be modeled as shown in Figure 6.2. Only reasonable trim conditions are considered. Figure 6.2 shows the high-weight/high-altitude and low-weight/low-altitude stall condition boundaries, which encompass an area labeled as the useable range. This useable range represents all possible trim conditions for the aircraft. Outside of that range, the aircraft either must fly too fast, (i.e. faster than Mach 0.9) or must have air denser than sea level. Note that while the Region 7 controller may use all of this envelope, in the other regions, the envelope is bounded on the left by the max L/D lift coefficient. The interesting observation here is that the useable area is rather small when compared to the total range of viable lift coefficients and true airspeeds. One must ask the question, how much modal property variation can there be within this range?

To answer this question, the aircraft's flight envelope was represented in yet another way. The locus of all possible phugoid poles for the entire flight envelope was plotted on a single graph. The aircraft's speed, altitude, and weight were varied encompassing the entire range of reasonable trim conditions. The results are shown in Figure 6.3. From the locus we can see three extremes. Roughly, the points are:

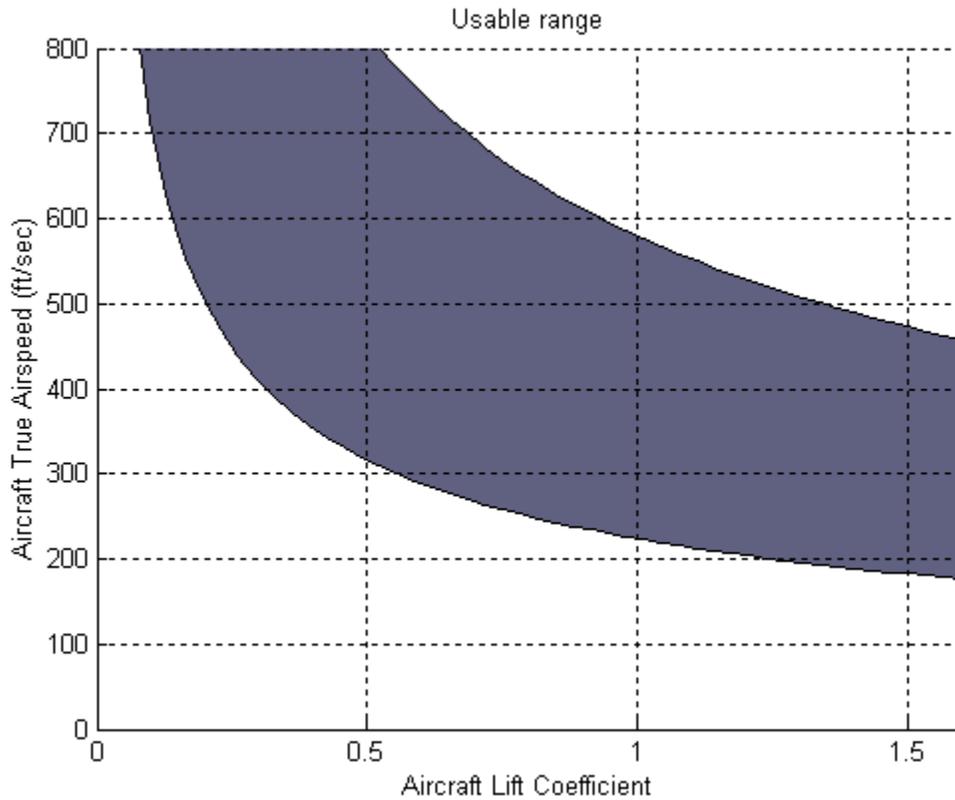


Figure 6.2. The flight envelope for a DC-9 in terms of CL and True Airspeed

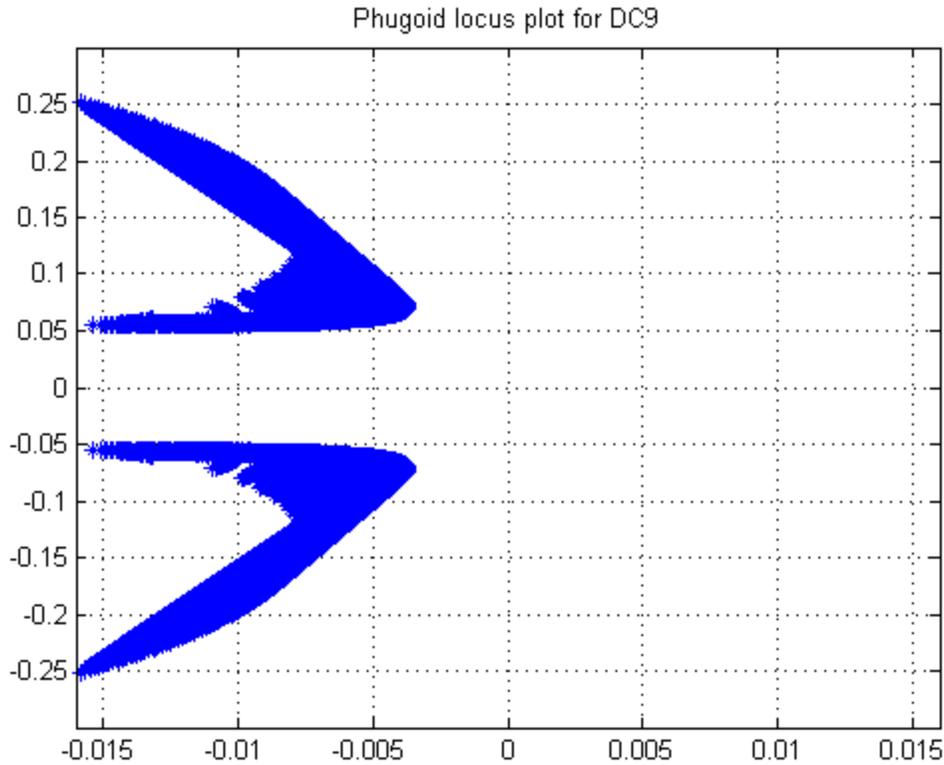


Figure 6.3. The locus of Phugoid poles for the entire flight envelope of a DC-9 in the clean configuration

- $-0.0011 \pm 0.056i$ $\omega_{n_p} = 0.057 \text{ rad/sec}$ $\xi_p = 0.19$ High lift coefficient at highest trimmable airspeed
- $-0.0074 \pm 0.042i$ $\omega_{n_p} = 0.043 \text{ rad/sec}$ $\xi_p = 0.17$ Low lift coefficient at highest trimmable airspeed
- $-0.0062 \pm 0.099i$ $\omega_{n_p} = 0.099 \text{ rad/sec}$ $\xi_p = 0.063$ High lift coefficient at lowest trimmable airspeed.

While there is considerable variation in frequency and damping, it is plain to see that the aircraft needs an increase in damping to have acceptable modal properties.

6.2 Determining Acceptable Modal Properties

In previous sections of the document, a considerable effort was made to determine gains that would yield desirable transient responses for the aircraft's phugoid dynamics. Generally, determining what is desirable is easy. Most dynamic systems are considered to be well behaved if they have a damping ratio of 0.7 and a frequency sufficiently high enough to remove transients quickly. A harder question to answer however is what

dynamic properties are sufficient. That is to say while it may be obvious what is desirable, it may not at all be obvious how to determine the range of acceptable values. Different types of operations may be more or less sensitive to poor uncontrolled dynamics. This section addresses the question of just how precisely the phugoid dynamics must be held to a specified set of modal properties.

There is no precise answer to this question. However, from observation of the aircraft flying with varied modal properties, one can conclude that a very wide range of properties is acceptable. First, consider the most important state variables in the longitudinal dynamics from the pilot's point of view: speed and altitude. The control system must be able to drive the aircraft to different speeds and altitudes. Consider changes in altitude. The feedback control systems don't use altitude explicitly, but rather feedback its derivative, altitude rate. Because of the integral relationship between the two, altitude tends to be insensitive to small transients in altitude rate. Furthermore, the careful design of the desired output vector minimizes large errors that would produce undesirable transients. In addition, the steady state error between the desired output vector and the actual output vector is of little concern until the desired output reaches the commanded output. Zero steady state error is important only when the commanded output values have been reached. This is in contrast, of course, to a mission such as precise terrain following where the error in following a time-varying output vector would be critical.

While it is difficult to put a range on acceptable modal properties, we can state some general guidelines that are based purely on observation. These are:

- The damping ratio of the mode is more important than the frequency
- The damping ratio can vary from roughly 0.5 – 1.0 and achieve satisfactory performance
- The frequency can vary from 0.1 rad/sec to 1.0 rad/sec and still yield acceptable results.

This is a rather large range which suggests that a control system with even meager performance is likely to be acceptable. Most importantly, however, such a wide range suggests that a single set of gains, if chosen carefully, could accommodate the entire flight envelope of the aircraft.

6.3 Choosing a Single Reference Condition for Gain Calculation

To choose a single reference condition that would serve as representative of the whole flight envelope, several conclusions from Section 3 must be revisited. These conclusions are:

- $\zeta_p = \frac{1}{\sqrt{2}(\frac{L}{D})}$; The damping of the mode is inversely proportional to the L/D ratio

- $\omega_p = \sqrt{\left(\frac{2g^2}{V_a^2}\right)}$; The frequency of the phugoid is inversely proportional to the true airspeed.

The lift coefficient that yields the highest L/D ratio, $C_{L_{(L/D)_{max}}}$, has the lowest damping.

This occurs at the bottom of the thrust curve, i.e., at the bucket speed. Any variation in the lift coefficient on either side of the thrust curve will yield a decrease in the L/D ratio and, therefore, an increase in phugoid damping. Furthermore, the lowest true airspeed will have the highest phugoid frequency. Using this information the following reference condition was chosen.

- Choose the trim condition for the maximum L/D ratio
- Using the lift coefficient for maximum L/D, trim the aircraft with the lowest possible true airspeed. Generally this is done by choosing a low altitude and a low weight.

The rationale for gain selection is as follows:

- Since it is natural for the phugoid damping to increase, select gains at the reference condition that puts the damping near the lower bound of acceptable. As the lift coefficient varies, the damping will increase and fall within the acceptable range.
- Since the phugoid frequency decreases with increasing speed, select gains to put the frequency and the reference condition near the top of the acceptable frequency range. As the velocity increases, the frequency will come down and stay within the acceptable range.

Of course, in practice there is no guarantee that a system augmented with feedback control will maintain any of its open loop tendencies so the rationale as stated is merely a vague guideline. In reality different properties work better, however the stated reference condition did turn out to be a good choice.

6.4 Evaluating System Performance with Scheduled Gains

In practice, the aircraft dynamics did not vary as predicted when feedback control was applied; however, through some experimentation, the following modal properties were found to yield favorable results.

For jet (turbofan) and turboprop aircraft:

- Integrator pole locations: $-0.20, -0.25$
- $\omega_p = 0.25$ rad/sec, $\zeta_p = 0.9$

For piston aircraft:

- Integrator pole locations: $-0.20, -0.25$
- $\omega_p = 0.40$ rad/sec, $\zeta_p = 0.9$

Gains were calculated for the DC-9 aircraft in all control regions. Using these desired modal properties the loci of poles over the entire flight envelope are plotted in Figure 6.4 - Figure 6.7. A key to the figures is presented in Table 6.1. The poles are plotted only for the corners of the flight envelope as defined by the lift coefficient, altitude, and weight.

Table 6.1 Marker key to Figure 6.4 - Figure 6.7

Outer marker - Lift Coefficient		
	box	front-side
	circle	bucket
	diamond	back-side
Middle marker - Altitude		
	box	sea level
	circle	mid-range altitude
	diamond	service ceiling
Inner marker - Weight		
	+	empty weight
	x	mid-range weight
	dot	max take-off weight

Consider altitude-rate-only feedback. The locus of closed loop poles produced for the aircraft is in Figure 6.4. Notice that the lowest phugoid damping occurs on the back-side of the thrust curve, instead of at the bucket speed as predicted earlier. The highest phugoid frequencies occur at sea level, as predicted. One should note the location of the integral poles since they stray into the right-half-plane. This means that our analysis of the linear system is predicting instabilities near stall speed, particularly at low weights and low altitudes. We will have to validate this flight condition by analyzing the transient response of the non-linear system. It is useful to note that we should expect a significant difference between the non-linear system and our linear approximation outside of the steady, level flight region because we stray more from the reference condition used for the linearization.

Figure 6.5 shows a similar locus for speed-only feedback. The trend is similar to that of Figure 6.4, except that the integral poles do not extend into the right-half-plane. While this region demonstrates greater stability, we should continue with an analysis of the non-linear transient response.

Figure 6.6 is a root locus plot for the steady, level flight region. We note similar trends as in Figure 6.4 and Figure 6.5: the lowest damping occurs on the back-side of the power curve and the highest phugoid frequencies occur at low weight and low altitude. However, inspection of the locus does show that the modal properties for the entire flight envelope do fall within the general guidelines set forth in Section 6.2.

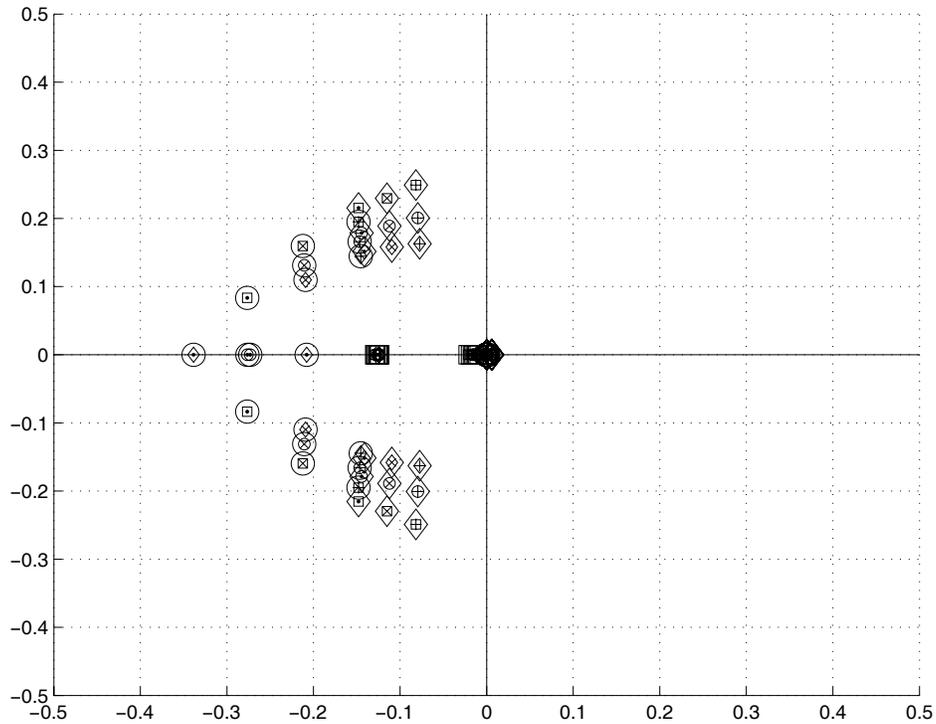


Figure 6.4. The locus of closed loop phugoid poles in for altitude-rate-only feedback for the entire flight envelope of a DC-9 in the clean configuration

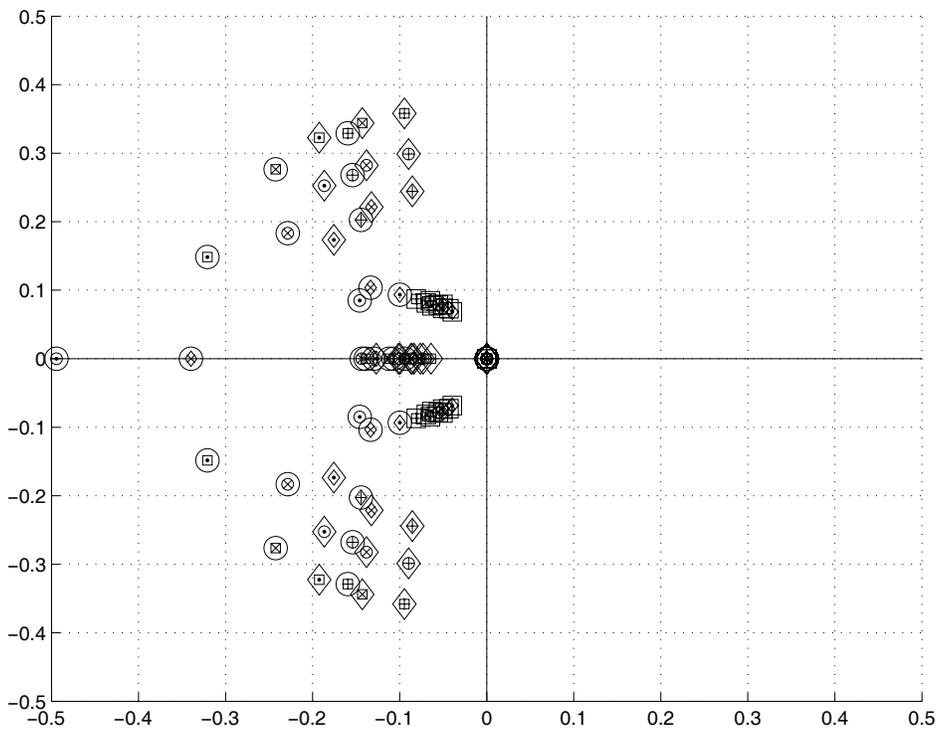


Figure 6.5. The locus of closed loop Phugoid poles for speed-only feedback for the entire flight envelope of a DC-9 in the clean configuration

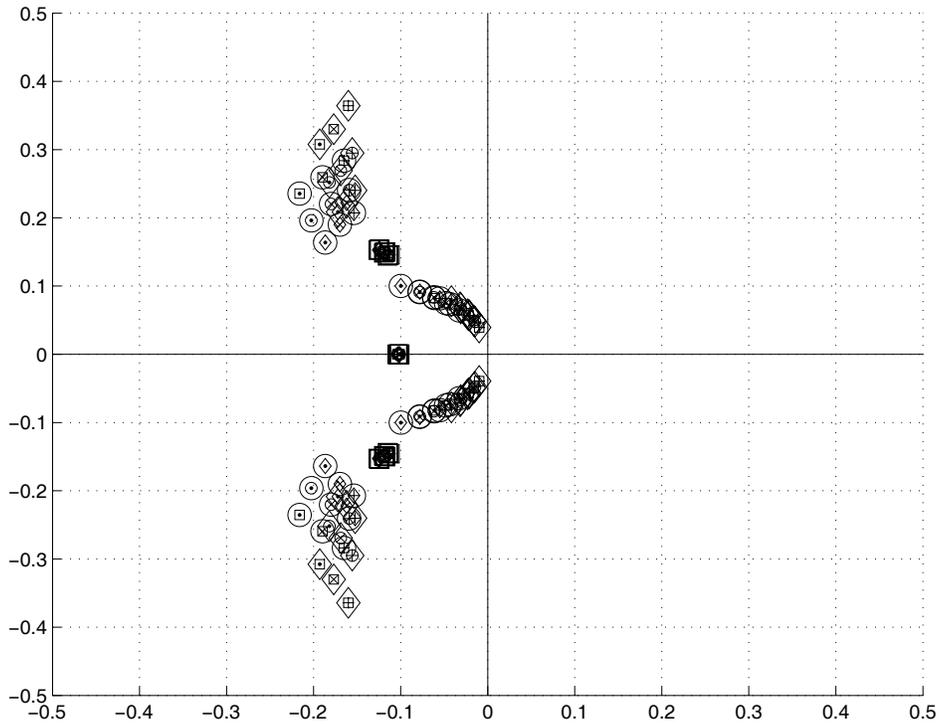


Figure 6.7. The locus of closed loop Phugoid poles in the steady, level flight region for the entire flight envelope of a DC-9 with full flaps deployed

A more sophisticated control law will fly the airplane more precisely; however, it would also require extra code, and more gains. For right now, since performance is still reasonably good in the low speed range, the decision has been made to capitalize on this unexpected result to simplify the control system. However, future requirements for other aircraft types may require a more complex control system to address the needs of low speed flight.

THIS PAGE INTENTIONALLY LEFT BLANK

7. The Lateral Directional Control Laws

The lateral directional control laws are considerably simpler than the control laws required to fly the aircraft longitudinally. The reason for the added simplicity is that one of the governing differential equations, the roll equation, can be modeled with a linear approximation. This approximation is used for two reasons. The first reason is that an accurate model of the full nonlinear roll dynamics is not essential to the modeling of accurate trajectories. The second is that the roll mode is so heavily augmented by the pilot or autopilot that the dynamics of turn rate capture is much more dependent on pilot response than the actual roll dynamics. The decision to approximate the speedy roll mode permits us to ease computational effort by selecting a half-second time step. Such a long time step is not adequate to capture the roll dynamics, though it is more than adequate to capture the slower phugoid dynamics.

Because our roll mode is linear, we can close feedback loops analytically without requiring the same detail that is done with the longitudinal dynamics. The main loop closures for the turning dynamics were already closed analytically in Section 2 and imbedded directly into the open loop dynamics making the desired bank angle, ϕ_{des} , the primary input to the roll equation. Although the loop closures are done analytically, we discuss them in this section as we outline the complete lateral directional control laws.

The four topics for discussion are:

1. The bank angle capture algorithm
2. The heading capture algorithm
3. Using the Bank Angle Capture and Heading Capture Algorithms to execute a turn
4. Deciding which way to turn

7.1 The Bank Angle Capture Algorithm

The bank angle capture algorithm is the major kernel of the lateral directional control law. Consider the governing lateral directional equations of motion shown in equations (7.1) through (7.3). The first two equations characterize the aircraft's response in roll to the aileron deflection. Equation (7.3) characterizes the aircraft's turn rate with respect to a given bank (or roll) angle.

$$\dot{p} = L_p p + L_{\delta_a} \delta_a \quad (7.1)$$

$$\dot{\phi} = p \quad (7.2)$$

$$\dot{\psi} = \frac{L S_\phi}{m V_a C_{\gamma_a}} \quad (7.3)$$

The only input to the system is the aileron deflection and the state variables of immediate interest are the roll rate p and the bank angle, ϕ . The equations of motion can be written in state space as shown in equation (7.4).

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} L_p & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} + \begin{bmatrix} L_{\delta_a} \\ 0 \end{bmatrix} \delta_a \quad (7.4)$$

For feedback control we choose to feed back both p and ϕ so that we can control both the frequency and the damping of the roll mode. The closed loop block diagram is shown in Figure 7.1.

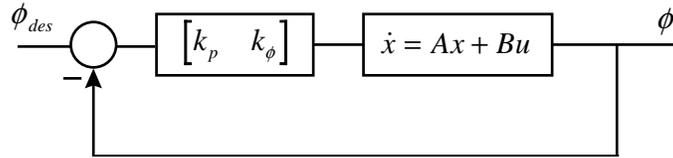


Figure 7.1. Block diagram for the rolling dynamics

The closed loop state space equations are:

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} L_p - L_{\delta_a} k_p & -L_{\delta_a} k_\phi \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} + \begin{bmatrix} L_{\delta_a} k_\phi \\ 0 \end{bmatrix} \phi_{des} \quad (7.5)$$

Section 2.14 explains that the natural integral relationship between δ_a and ϕ guarantees a zero steady state error without the use of integral control. We can verify this by examining the ϕ/ϕ_{des} transfer function shown in equation (7.6). From inspection, we see that the DC gain of the transfer function is unity.

$$\frac{\phi}{\phi_{des}} = \frac{L_{\delta_a} k_\phi}{s^2 + (L_{\delta_a} k_p - L_p)s + L_{\delta_a} k_\phi} \quad (7.6)$$

We can see that with the closed loop dynamics already incorporated into the open loop dynamics, there is very little left to do. The feedback gains are chosen to give the desired time response and can be tailored for each individual flight. Nominally, the gains are set to $k_p = 22$ and $k_\phi = 50$.

Sometimes it is desirable to command a specific turn rate. Since the turn rate equation is very nearly a linear function of the roll angle, we simply choose to adjust our commanded bank angle rather than creating another feedback control loop to drive the system to a commanded turn rate. Calculating the required bank angle is done by rearranging equation (7.3).

$$\phi_d = \sin^{-1}\left(\frac{m\dot{\psi}V_a C_{\gamma_a}}{L}\right) \quad (7.7)$$

7.2 The Heading Capture Algorithm

The heading capture algorithm is designed to capture a specified heading. To capture a given heading, we feed back the desired heading to the bank angle using the control law shown in equation (7.8).

$$\phi_d = k_\psi(\psi_d - \psi) \quad (7.8)$$

To predict the effect of this feedback control law, we must first add the heading equation to our state space model. Consider the linearized version of the turn rate equation which finds its way into our state matrix.

$$\frac{d\psi}{d\phi} = \frac{d}{d\phi}\left(\frac{LS_\phi}{mV_a C_{\gamma_a}}\right) = \frac{LC_\phi}{mV_a C_{\gamma_a}|_0} \Delta\phi \quad (7.9)$$

$$\frac{d\psi}{dp} = 0 \quad (7.10)$$

If we assign our reference condition for the linearization to be $\phi = 0.0$, then $\Delta\phi = \phi$. Furthermore, if we note that for the bulk of the flight the lift equals the weight and the flight path angle is near zero, we can simplify equation (7.9) to equation (7.11).

$$\frac{d\psi}{d\phi} \cong \frac{g}{V_a} \phi \quad (7.11)$$

Arranging the system of equations in state space we have equation (7.12).

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} L_p - L_{\delta_a} k_p & -L_{\delta_a} k_\phi & 0 \\ 1 & 0 & 0 \\ 0 & \frac{g}{V_a} & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} L_{\delta_a} k_\phi \\ 0 \\ 0 \end{bmatrix} \phi_{des} \quad (7.12)$$

When we close a proportional loop around the system with k_ψ as our feedback gain, as shown in Figure 7.2, we see that the new closed loop system is equation (7.13). There is an integral relationship between the heading and the roll angle so zero steady state error is achieved without the use of integral control.

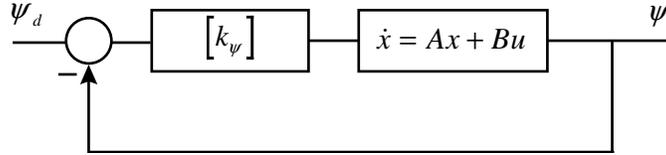


Figure 7.2. Block diagram for heading feedback

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} L_p - L_{\delta_a} k_p & -L_{\delta_a} k_\phi & -L_{\delta_a} k_\phi k_\psi \\ 1 & 0 & 0 \\ 0 & \frac{g}{V_a} & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} L_{\delta_a} k_\phi k_\psi \\ 0 \\ 0 \end{bmatrix} \psi_d \quad (7.13)$$

Further verifying that integral control is unnecessary, we see that the transfer function that characterizes the relationship between ψ and ψ_d , equation (7.14), has a DC gain of 1.

$$\frac{\psi}{\psi_d} = \frac{\frac{g}{V_a} L_{\delta_a} k_\phi k_\psi}{s^3 - (L_p - L_{\delta_a} k_p) s^2 + L_{\delta_a} k_\phi s + \frac{g}{V_a} L_{\delta_a} k_\phi k_\psi} \quad (7.14)$$

7.3 Using the Bank Angle Capture and Heading Capture Algorithms to Execute a Turn

When turning, the heading capture algorithm cannot be used for large heading errors. The reason is that the heading capture algorithm will command a bank angle proportional to the heading error. If the heading error is large, the control law will command an unreasonably large bank angle such as 180 degrees. This bank angle would correspond to an inverted aircraft and certainly does not make the aircraft turn any faster. Therefore, the heading capture algorithm is used only when the heading error is less than 15 degrees. For errors greater than 15 degrees, the bank angle control law is used to command a constant turn rate in the direction of minimizing the heading error. Nominally, a bank angle of 30 degrees is used. Consider the following simulation example. The simulation parameters are as follows:

- $V_a = 300 \text{ ft/sec}$
- $L_p = -0.475$
- $L_{\delta_a} = 0.185$
- $k_p = 2.836$
- $k_\phi = 2.756$

Although the ADM uses $k_\psi = 1$, this simulation uses the following feedback gain, which accounts for variations in airspeed.

$$k_\psi = 0.005 \frac{V_a}{g} \quad (7.15)$$

We simulate a turn to the right from a heading of 0 degrees to a heading of 100 degrees as shown in the simulation results in Figure 7.3. Initially, the aircraft rolls to the right to achieve a bank angle of 30 degrees. The aircraft holds the bank angle and steadily turns toward a heading of 100 degrees. When the aircraft is within 15 degrees of the desired heading, the heading capture algorithm takes over and drives the remaining heading error to zero.

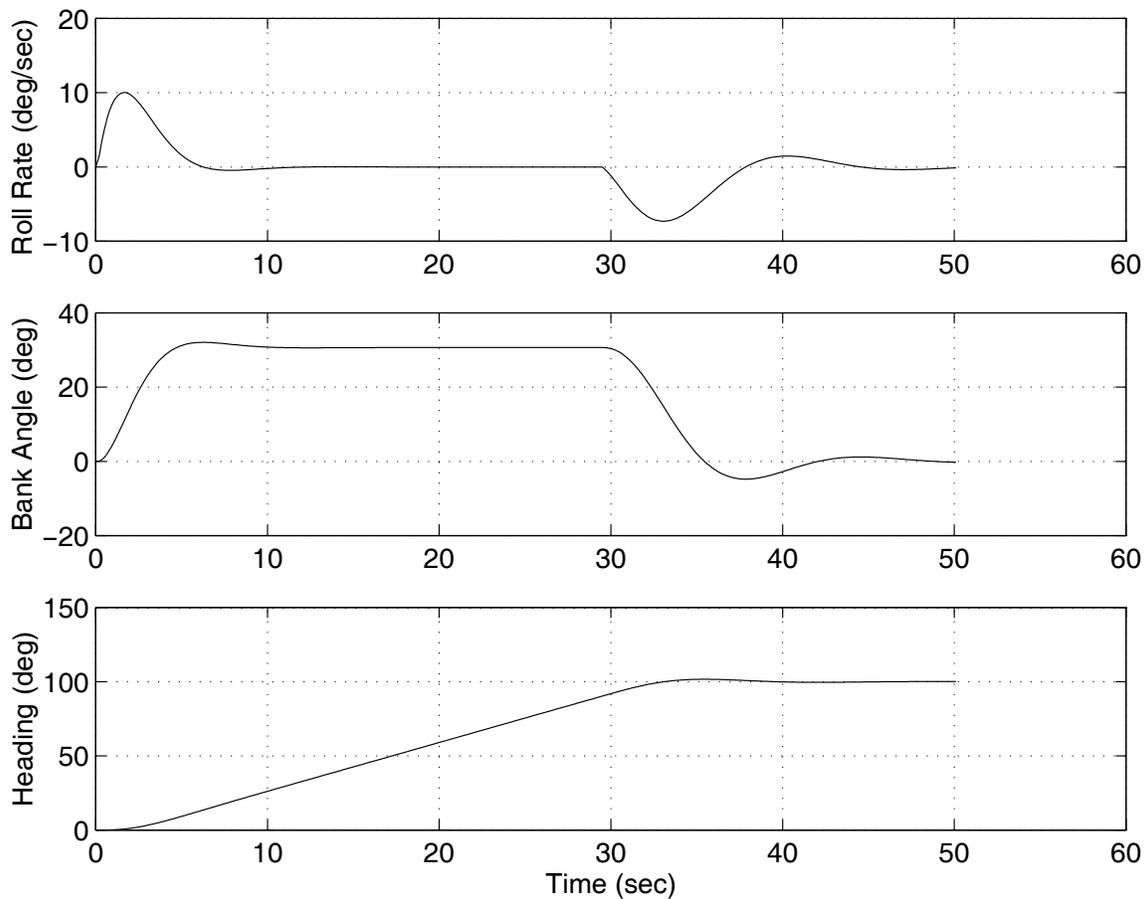


Figure 7.3. Simulation of Aircraft Executing a Turn

7.4 Deciding Which Way to Turn

The TGF simulator's user interface allows for a turn left or right to the heading command. Often, however the aircraft is left to make that decision for itself. In this case, the aircraft must choose which direction of turn is the shortest. Either a left turn or a right turn will work, but one turn is shorter. The dilemma is illustrated in Figure 7.4. To the

human, it is obvious that a right turn is appropriate for the situation presented in Figure 7.4; however, the logic required to make the autopilot come to the same conclusion is not trivial. The following logic determines the magnitude and the sign of the heading error, referred to as e_5 in the simulation code. The first task is to determine the magnitude of the heading errors to the left and right, symbolically represented as e_{left_turn} and e_{right_turn} .

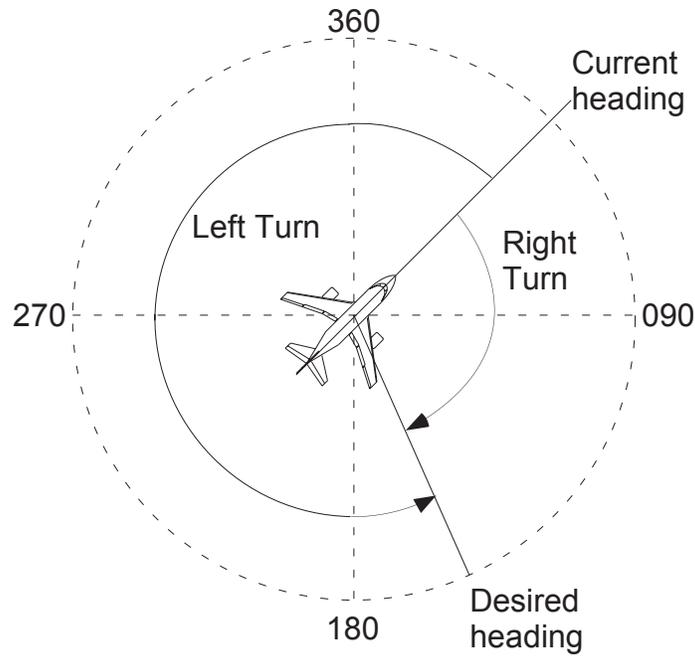


Figure 7.4. An illustration of the dilemma of whether to make a right or left turn to a heading

$$\begin{aligned}
 &\text{if } \psi_d > \psi \\
 &\quad e_{right_turn} = (\psi_d - \psi) \\
 &\quad e_{left_turn} = (\psi_d - \psi) - 360 \\
 \\
 &\text{if } \psi_d < \psi \\
 &\quad e_{right_turn} = (\psi_d - \psi) + 360 \\
 &\quad e_{left_turn} = (\psi_d - \psi)
 \end{aligned}$$

Next, the absolute value of e_{left_turn} and e_{right_turn} are compared to determine which is smaller. The actual heading error, e_5 , is set equal to the smaller of these two errors. It is convenient to use the convention that turning errors to the left are always negative and turning errors to the right are always positive. This corresponds nicely to the bank angle convention where banks to the right are considered positive and banks to the left are negative. Therefore, there is no need to adjust the previously developed control laws to make sure that the aircraft turns in the desired direction when commanded.

7.5 Capturing a Heading when the Direction of Turn is Specified

The introduction of the left or right turn variability adds more complexity to the system than what was previously anticipated. Basically, when the aircraft was constrained to turn in only one direction, it would not always capture the proper heading. This happened because when the aircraft turned in the specified direction towards the target heading, it would overshoot slightly and instead of turning back to the heading, it would turn 360 degrees around. Therefore, there needed to be some distinction made between the initial turn and the capture of the heading. Figure 7.5 shows the algorithm for determining whether or not a heading has been captured. Essentially, the aircraft turns in the specified direction until the heading error is within 5° , at which point it uses the logic of the previous section to determine the turn direction on its own. The term e_5 is the error in heading.

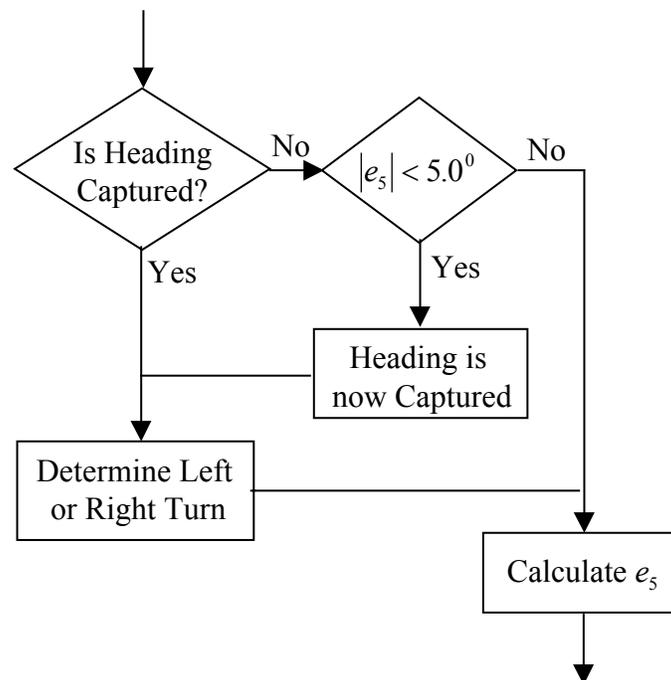


Figure 7.5. Algorithm for capturing a heading

THIS PAGE INTENTIONALLY LEFT BLANK

8. The Lateral Guidance System

The purpose of the lateral guidance system is to steer the aircraft to follow routes or other sim-pilot commands within the horizontal plane. There are four basic maneuvers. These are:

- Ground track guidance
- Fix capture guidance
- Route following
- Route capture

The ground track guidance algorithm steers the aircraft along a specified ground track from the flight plan. In the presence of wind, the algorithm must determine a wind bias to the aircraft heading to maintain the ground track. The fix capture algorithm flies the aircraft to a fix. Route following steers the aircraft along a specified route. Finally, the route capture algorithms are discussed. The route capture algorithms steer the aircraft towards a route and then capture the route. Several different ways to capture a route will be discussed.

8.1 Ground Track Guidance

The ground track azimuth is the angle between the aircraft's ground track and true North. Under a zero wind condition, the ground track azimuth is the same as the aircraft's heading. In the presence of wind, the ground track azimuth will differ from the aircraft heading as illustrated in Figure 8.1. To fly to a fix or capture a route, the aircraft must follow a given ground track azimuth rather than a specific heading; yet the lateral control system is designed only to turn to a desired heading. The lateral guidance must bias its heading commands to the lateral control system with a correction factor that accounts for

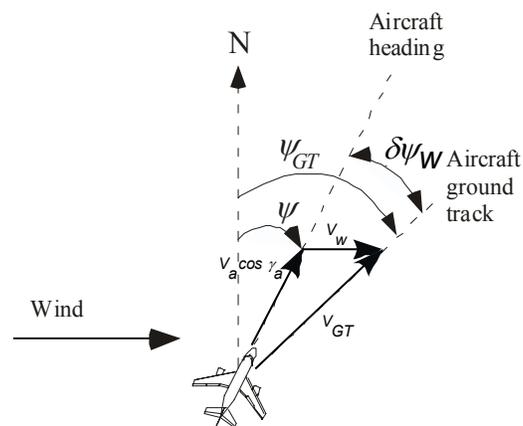


Figure 8.1. Illustration of the difference between ground track and heading

winds. To accommodate this requirement, the lateral guidance measures the difference between the heading and the ground track azimuth, which we denote as $\Delta\psi$. Our tracking algorithm makes uses the following nomenclature:

- ψ : The aircraft's heading, in degrees.¹
- ψ_{GT} : The aircraft's ground track azimuth, in degrees.
- $\delta\psi_w$: The wind bias.
- ψ_d : The desired heading.
- ψ_{GT_d} : The desired ground track azimuth.

We define the wind bias as the difference between the aircraft's ground track azimuth and its heading.

$$\delta\psi_w \equiv \psi_{GT} - \psi \quad (8.1)$$

The aircraft's ground track azimuth and heading are available from the aircraft dynamics. The wind bias, $\delta\psi_w$, is then used to adjust the desired ground track so that the aircraft will track properly. The result, calculated using Equation(8.2), is the desired heading.

$$\psi_d = \psi_{GT_d} - \delta\psi_w \quad (8.2)$$

Equation(8.2) is not intended as an accurate representation of the vector algebra graphically depicted in Figure 8.1; it merely shows the use of the wind bias as a correction factor. Its simplicity does not compromise its intent, which is to capture the desired ground track azimuth.

8.2 Fix Capture Guidance

To fly to a fix, it is necessary to know the range and bearing to the fix. Algorithms that perform these operations are discussed later. Once the bearing to the fix is known, the turn-to-heading logic is used to turn the aircraft to that bearing. This control strategy is effective as long as the aircraft is sufficiently far away from the fix so that the bearing is not changing quickly as seen in Figure 8.2. Bearing changes constantly as the aircraft moves, except when flying directly to the fix.

¹ Analysis of the wind bias requires a comparison of the azimuth of the aircraft's velocity vector, ψ_a , with the ground track azimuth, ψ_{GT} ; but in Chapter 2, we stated the assumption that the aircraft is always trimmed for coordinated flight; i.e., the sideslip is always zero. This means that the azimuth of the aircraft's velocity vector, ψ_a , is coincident with the aircraft's heading, ψ . Therefore, for our purposes, the analysis is equally accurate in comparing ψ and ψ_{GT} .

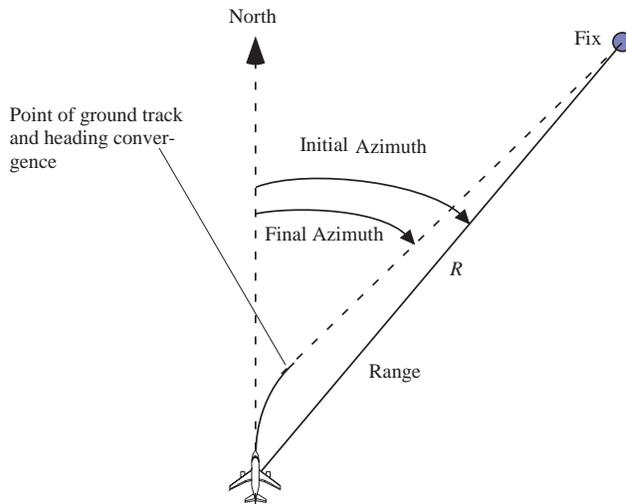


Figure 8.2. An aircraft turning to a fix

If the aircraft is sufficiently close to the fix and the required turn to the fix is large, the rate of bearing change will be equal to or greater than the turn rate of the aircraft. This will prevent the convergence of aircraft ground track and bearing, as shown in Figure 8.2. This particular case was of some concern originally; however, when the algorithm was actually tested, it turned out to be very difficult to set up a case where the aircraft would perpetually orbit a fix. Since the likelihood of such an occurrence is so rare, the current simulation makes no allowance for this limitation.

Ground track guidance is operated from within the fix capture guidance system to steer the aircraft. When the aircraft is 0.1 nm from the fix, the fix is considered captured. Figure 8.3 illustrates the fix capture functionality.

8.3 Route Following

The purpose of the route following algorithm is to guide the aircraft along a route. This includes maintaining the ground track along individual segments and transitioning to new segments at the proper time.

8.3.1 Maintaining Ground Track Along a Segment

When flying a particular segment, the route following algorithm commands the ground track of the aircraft based on the lateral distance that the aircraft is away from the segment, the capture segment's bearing, and the aircraft's radius of turn. The scenario is illustrated in Figure 8.4. The intercept angle for the given segment is a function of how far the aircraft is laterally from the segment.

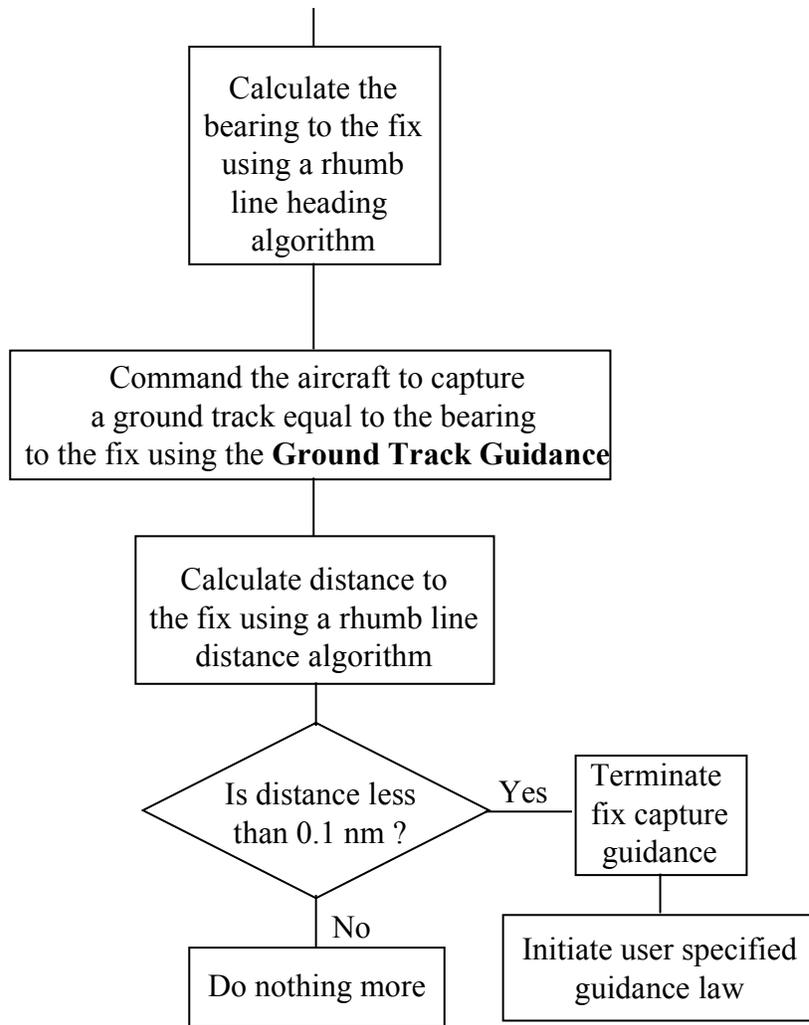


Figure 8.3. Fix capture guidance algorithms

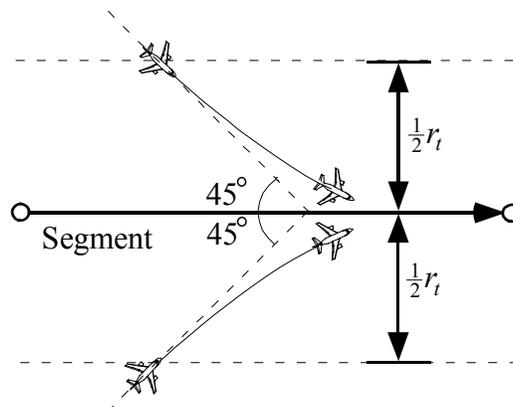


Figure 8.4. Illustration of the aircraft in route following mode

The intercept reaches a maximum of 45 degrees when the aircraft is one-half a turn radius away from the segment. The intercept angle is bounded at 45 degrees. Equations (8.3) and (8.4) are used to determine the aircraft's desired ground track. First, $\Delta\psi$ is calculated using Equation (8.3). If the result has a magnitude greater than 45 degrees, the answer is bounded at 45 degrees using Equation (8.4). The ratio $\delta/|\delta|$ is used to preserve the sign of the original value. Note that the lateral distance term, δ , maintains a sign convention of positive values on the right side of a segment and a negative value on the left side of the segment. This solution is adapted from the original System Segment Specification [TGF93].

$$\Delta\psi = 90 \frac{\delta}{r_t} + \delta\psi_{FTE}, \quad |\Delta\psi| < 45^\circ \quad (8.3)$$

$$\Delta\psi = 45 \frac{\delta}{|\delta|}, \quad |\Delta\psi| \geq 45^\circ \quad (8.4)$$

$$\psi_{GT_d} = \psi_r - \Delta\psi \quad (8.5)$$

The terms are defined as follows:

- δ : The aircraft's lateral distance from the capture segment (nm)
- ψ_r : The capture segment's bearing. (degrees)
- r_t : The aircraft's turn radius. (nm)
- ψ_{GT_d} : The aircraft's desired ground track (degrees)
- $\delta\psi_{FTE}$: The heading bias from flight technical error (degrees)

As with all other heading commands, the term ψ_{GT_d} needs to be adjusted to keep values within the 0 - 360 degree range. The logic for this operation is shown in Figure 8.5.

The flight technical error (Chapter 11) is sent to the heading-based course guidance in the form of a lateral offset, denoted as δr_{FTE} . The lateral error offset is then related to a heading bias using Equation (8.6).

$$\delta\psi_{FTE} = \frac{90}{r_t} \delta r_{FTE} \quad (8.6)$$

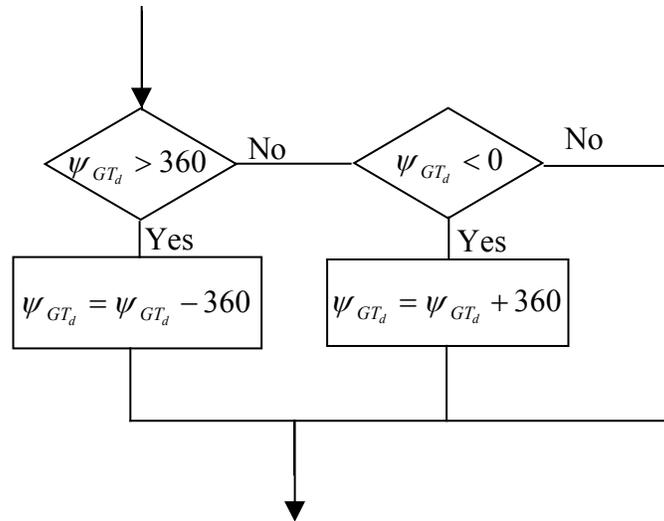


Figure 8.5. Logic for insuring desired ground track is within proper boundaries

The terms in the equation are defined as follows:

- $\delta\psi_{FTE}$: The heading bias created from flight technical error (degrees).
- r_t : The turn radius for the aircraft at the current speed (nm).
- δr_{FTE} : The lateral offset from flight technical error (nm).

The flight technical error conversion from lateral distance to a heading bias mimics Equation (8.3) in form and causes the course guidance algorithm to produce the lateral offset error of δr_{FTE} in the flight path.

8.3.2 Transitioning to the Next Segment

The route following and route capture algorithms both advance the route segment number when the aircraft moves along a segment beyond a point where a turn must be made to affect a smooth transition to the next segment. This distance is termed the segment transition distance and is illustrated by Figure 8.6. The magnitude of this distance is affected by the turning radius of the aircraft and hence the speed of the aircraft.

To determine the segment transition distance, two different drawings of the scenario are presented in Figure 8.7 and Figure 8.8. Figure 8.7 shows the more common case where segments intersect at obtuse angles. Figure 8.8 shows the less common case where segments intersect at acute angles. First, the angle between the segments is calculated using the definition of the dot product as shown in Equation (8.7) and its final form Equation (8.8).

$$\mathbf{R}_{s_1} \cdot \mathbf{R}_{s_2} = \|\mathbf{R}_{s_1}\| \|\mathbf{R}_{s_2}\| \cos a \quad (8.7)$$

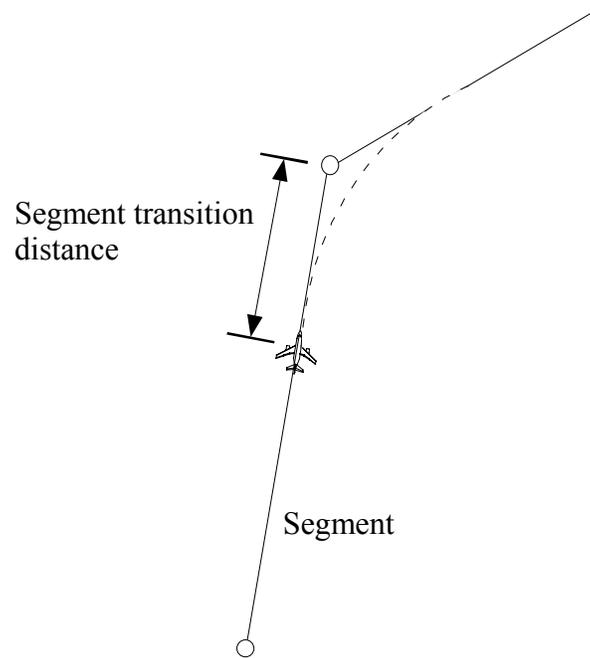


Figure 8.6. Illustration of segment transition distance

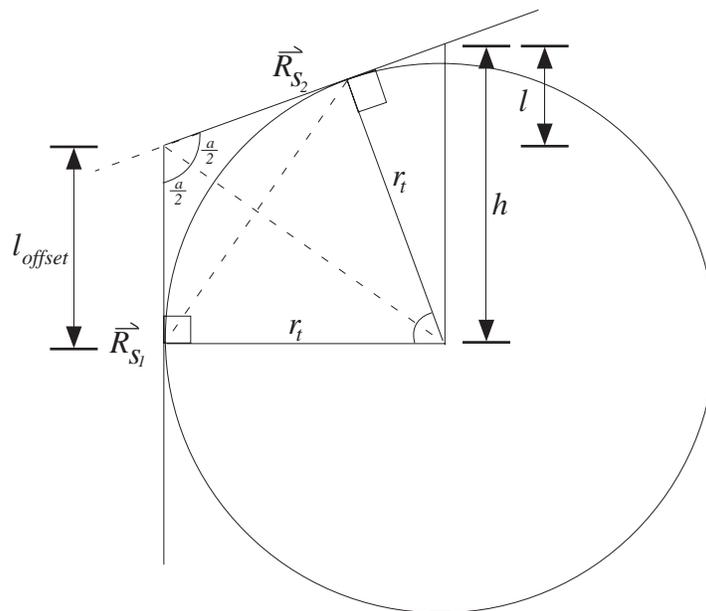


Figure 8.7. The geometry of segments which intersect at obtuse angles

$$a = \cos^{-1} \left(\frac{\mathbf{R}_{s_1} \cdot \mathbf{R}_{s_2}}{\|\mathbf{R}_{s_1}\| \|\mathbf{R}_{s_2}\|} \right) \quad (8.8)$$

The segment transition distance, l_{offset} , is then calculated by observing in Figure 8.7 that two isosceles triangles are formed creating a ‘kite’ like pattern. We can then bisect the angle and form two right triangles. Trigonometry can then be used to calculate the

segment transition distance using Equation (8.9), where r_t is the turn radius of the aircraft. A factor of 1.3 is added to allow for a margin of error since turn dynamics are not instantaneous. The equation is valid for the acute angle case of Figure 8.8 as well.

$$l_{offset} = r_t \tan\left(\frac{a}{2}\right) \quad (8.9)$$

8.4 Capturing a Route

A route in the aircraft simulation consists of a list of fixes. Segments in the simulation are defined by adjacent fixes along a route. There are three ways to used to capture a route. They are:

- Automatic route capture
- Vectored route capture
- Initial fix route capture

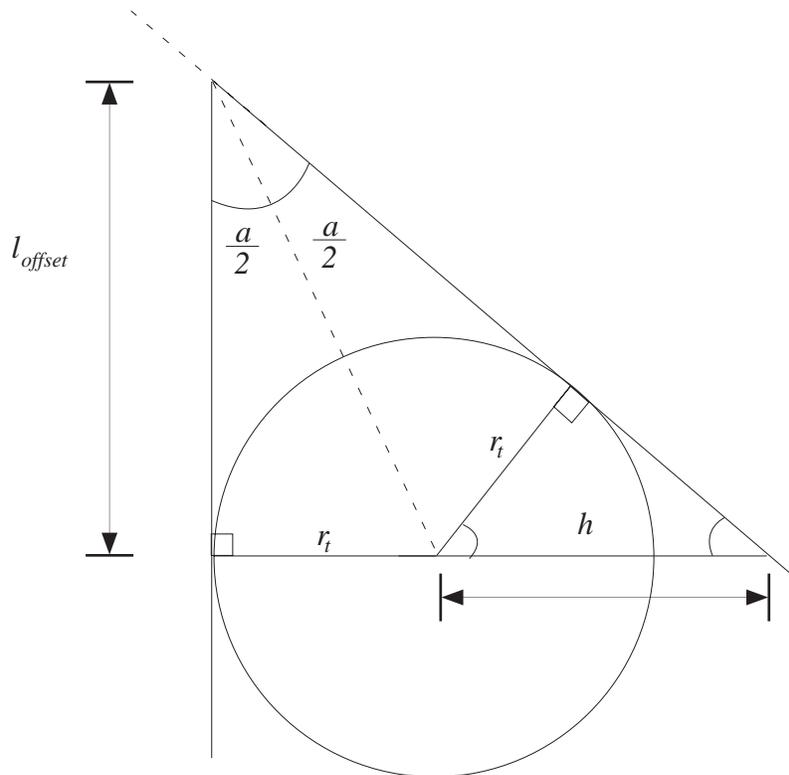


Figure 8.8. Geometric representation of segments adjoined at an acute angle

When using automatic route capture, the aircraft guidance performs all of the necessary operations to determine which segment should be captured first and then steer the aircraft toward the segment. Finally, the capture algorithm merges onto the route. The vectored route capture algorithm requires manual guidance of the aircraft to the route; however, once the aircraft is sufficiently close to the route, the guidance algorithm merges the aircraft with the route. The last route capture algorithm is the initial fix route capture. This algorithm flies the aircraft to the initial fix first and then captures the route.

Some general algorithms are used to make the captures algorithm work. These functions are:

- Determining a capture segment
- Determining if it is time to turn onto the route

8.4.1 Determining a Capture Segment

There are three criteria which are used to determine the appropriate segment to capture. These criteria are:

1. The Dot Products of the aircraft's location relative to the leading and trailing fixes and the segment's vector \mathbf{R}_s .
2. The lateral distance from the segment.
3. The closest trailing fix.

Criterion #1 is summarized as:

If a segment's position vector when dotted with a position vector from the aircraft's location to the leading fix yields a positive value, and if a segment's position vector when dotted with a position vector from the aircraft's location to the trailing fix of the same segment yields a negative value, then the aircraft should capture the segment.

$$\begin{array}{l} \mathbf{r}_1 \cdot \mathbf{R}_{s1} < 0 \quad \mathbf{r}_2 \cdot \mathbf{R}_{s2} < 0 \quad \mathbf{r}_3 \cdot \mathbf{R}_{s3} > 0 \\ \mathbf{r}_2 \cdot \mathbf{R}_{s1} < 0 \quad \mathbf{r}_3 \cdot \mathbf{R}_{s2} > 0 \quad \mathbf{r}_4 \cdot \mathbf{R}_{s3} > 0 \end{array}$$

Consider the following scenario shown in Figure 8.9. There are three segments in the route and the aircraft must determine which segment to capture. To do this the position vectors \mathbf{r}_1 through \mathbf{r}_4 are determined. These vectors are then dotted with the position vectors of each segment. If the dot product between a position vector from the aircraft location to the leading fix of a segment and the segment position vector is positive, then the aircraft tends to be behind the leading fix of a segment. Likewise, if the dot product is negative, the aircraft will be ahead of the fix. For a segment to be a good choice for capture, the aircraft should be behind the leading fix and in front of the trailing fix. In the scenario in Figure 8.9, we see that the dot products for the first segment are both negative. Therefore the aircraft is in front of the first segment. For the second segment, the dot product to the trailing fix is negative while the dot product to the leading fix is positive. The second segment would therefore be an acceptable choice for capture.

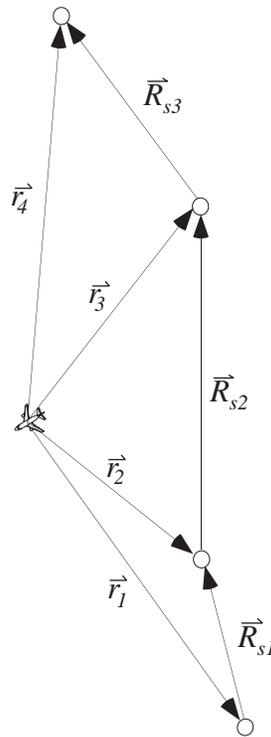


Figure 8.9. Scenario of an aircraft determining which segment to capture

Looking at the third segment, both dot products are positive so the segment is in front of the aircraft. Initially, this test alone was thought to be sufficient to determine which segment should be captured; however, it is not.

Consider the next case shown in Figure 8.10. When the required dot products are calculated, it is shown that both segments meet the requirements for capture. Therefore criterion #2 calculates the lateral distance from every segment deemed acceptable by criterion #1. The closest segment is captured. In Figure 8.10, the first segment is chosen because it is the closest to the aircraft.

Before extensive testing, it was thought that these two criteria would be sufficient to handle all cases. They are not. Consider the cases where no segment is acceptable as defined by criterion #1. These cases are illustrated in Figure 8.11. Criterion #1 will fail to yield any segment for capture if its dot product requirements are not met. This often occurs when the aircraft is sufficiently behind or in front of the route as shown in regions A and C of Figure 8.11. There is also another “dead” region where two segments meet as shown in region B. If an aircraft is in this region, criterion #1 will not find a segment. In this case, criterion #3 is used. Criterion #3 checks the aircraft’s distance from every

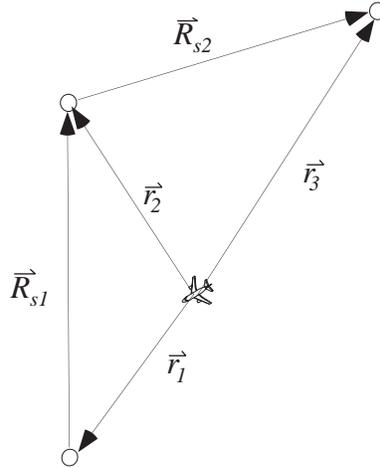


Figure 8.10. A scenario demonstrating the failure of criterion #1

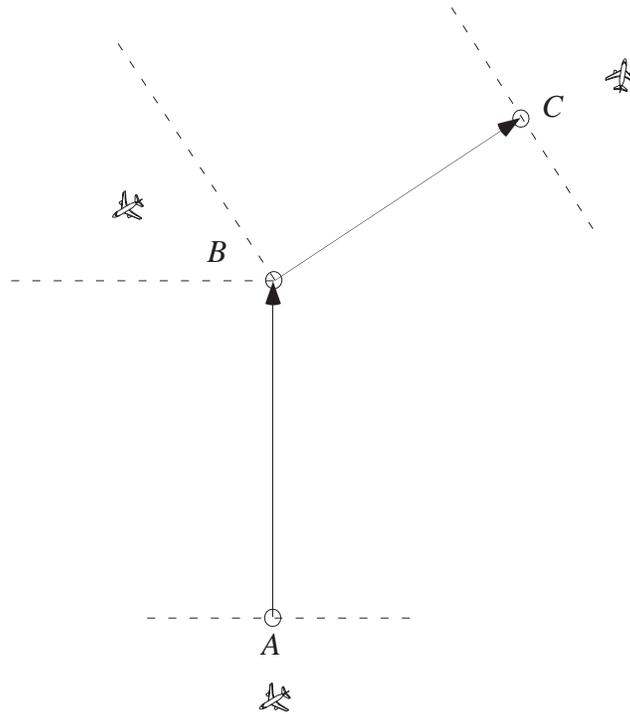


Figure 8.11. Regions where both criterion #1 and criterion #2 fail

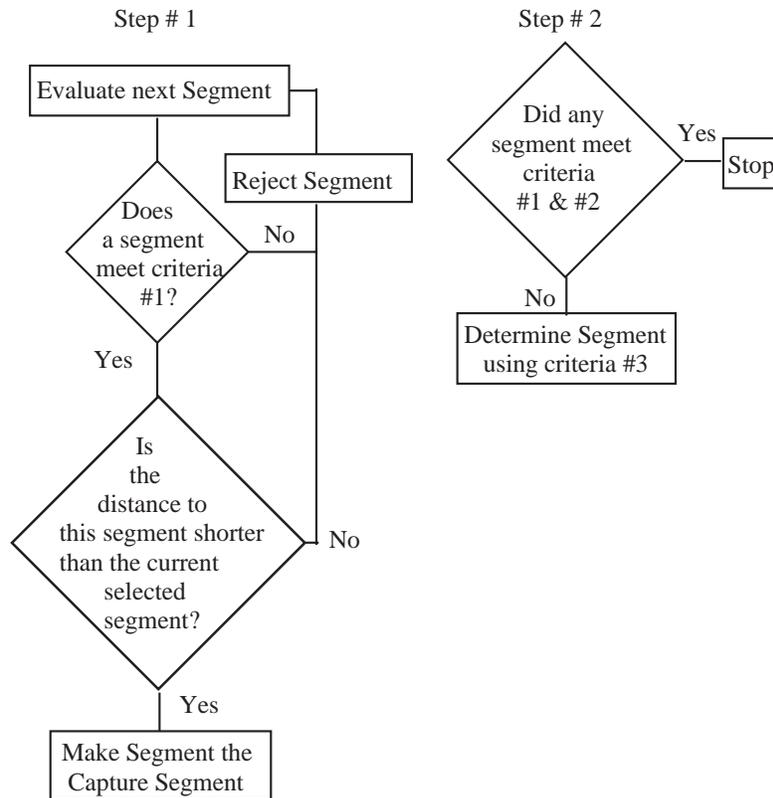


Figure 8.12. Flow chart detailing segment determination logic

segment's trailing fix. It then chooses to capture the segment that is associated with the closest trailing fix. Figure 8.12 illustrates the segment determination logic.

8.4.2 Determining if it is Time to Merge onto the Route

When an aircraft approaches a segment on a route, it must gauge when it should start to turn to merge cleanly onto the route. Generally, the distance that is required is a function of the aircraft's speed and the intercept angle that the aircraft has with the segment. It is a very similar calculation to that which is used for segment transition. Figure 8.13 illustrates the geometry of an aircraft merging onto a segment

The algorithm requires the aircraft's true airspeed and heading and a vector describing the segment.

- V_a : Aircraft's true airspeed. (ft/sec)
- ψ : Aircraft's heading. (deg)
- \mathbf{R}_s : A vector describing a segment.

First, a vector, \mathbf{V} , representing the aircraft's velocity is created from the aircraft's airspeed and heading. Using the definition of the dot product, the angle between the vectors is calculated using Equation (8.10).

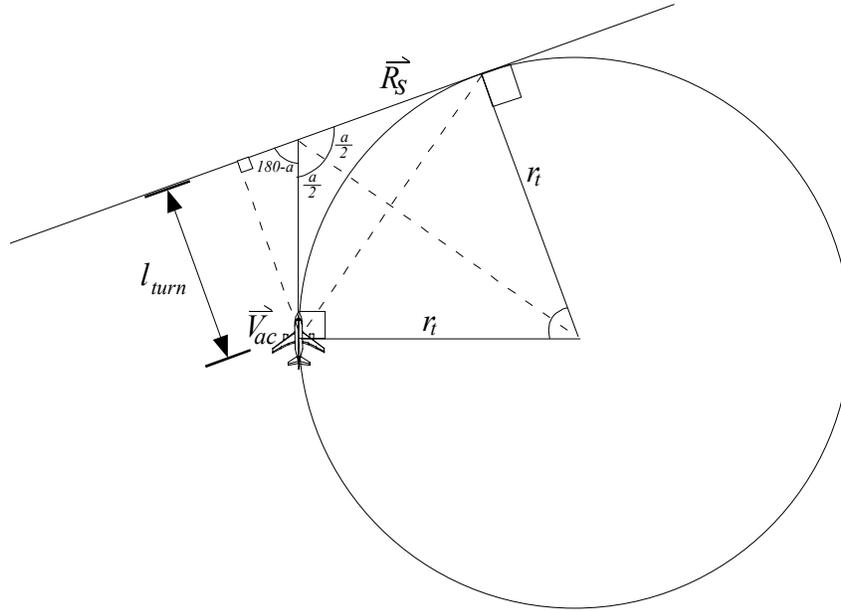


Figure 8.13. Illustration of geometry associated with an aircraft merging onto a segment when aircraft is heading in the direction of the segment

$$a = \cos^{-1} \left(\frac{\mathbf{V} \cdot \mathbf{R}_s}{\|\mathbf{V}\| \|\mathbf{R}_s\|} \right) \quad (8.10)$$

We can see from the geometry in Figure 8.13, that the problem is similar to the segment transition problem. We can see from the figure that the distance at which the aircraft should turn, l_{turn} , is the projection of l_{offset} onto a line normal to the segment. Therefore it can be calculated using Equation (8.11), where r_i is the turn radius of the aircraft in nm.

$$l_{turn} = 1.3r_i \tan \left(\frac{a}{2} \right) \sin(180 - a) \quad (8.11)$$

Using the trigonometry identity $\sin a = \sin(180 - a)$, Equation (8.11) can be simplified to Equation (8.12).

$$l_{turn} = 1.3r_i \tan \left(\frac{a}{2} \right) \sin a \quad (8.12)$$

When the aircraft is tending to head in the direction opposite the direction of the segment, more distance is needed to turn because the aircraft must completely change the direction of flight to fly along the segment. This case is illustrated in Figure 8.14. However, Equation (8.12) is still valid as can be verified from inspection of the geometry in Figure 8.14.

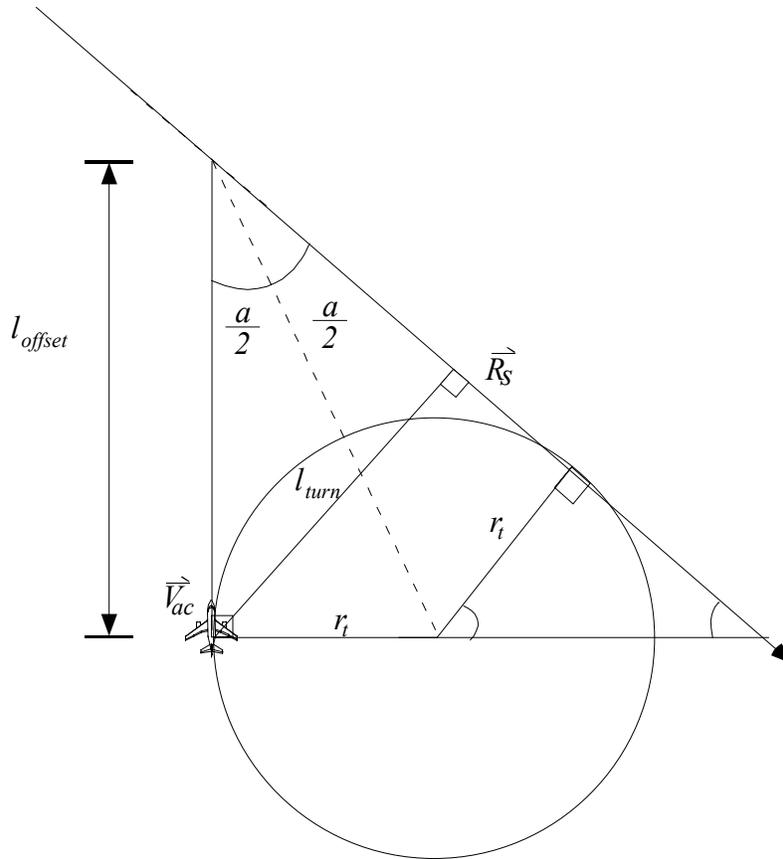


Figure 8.14. An aircraft merging onto a segment which is pointed in a direction opposite of the aircraft's current velocity

8.4.3 Automatic Route Capture

The automatic route capture guidance algorithm automatically steers the aircraft to the nearest segment and captures the segment. The algorithm is described in the flow diagram illustrated in Figure 8.15.

The reader will notice that there are 4 algorithms which are used in automatic route capture guidance. These algorithms are determining a capture segment, determining if it is time to merge onto the route, determining if the segment should be advanced, and determining a dynamic fix. Each algorithm is already covered in a previous section with the exception of the dynamic fix which is unique to automatic route capture.

8.4.3.1 The Dynamic Fix

The dynamic fix is an imaginary fix which is created by the system at some location along a segment and is used as a point of reference for capture. When the automatic route capture algorithm was first conceived, it seemed as though the most obvious method of capturing the route was to fly some intercept heading to the route. For instance, once the capture segment was determined, the aircraft could be given an intercept heading of 45 degrees and intercept the segment. However, this method seemed to have some inherent

limitations. First, the aircraft would always intercept using 45 degrees regardless of how far the aircraft was away from the segment. An aircraft far away from the capture segment might pass the segment before ever capturing it. This situation is illustrated in Figure 8.16.

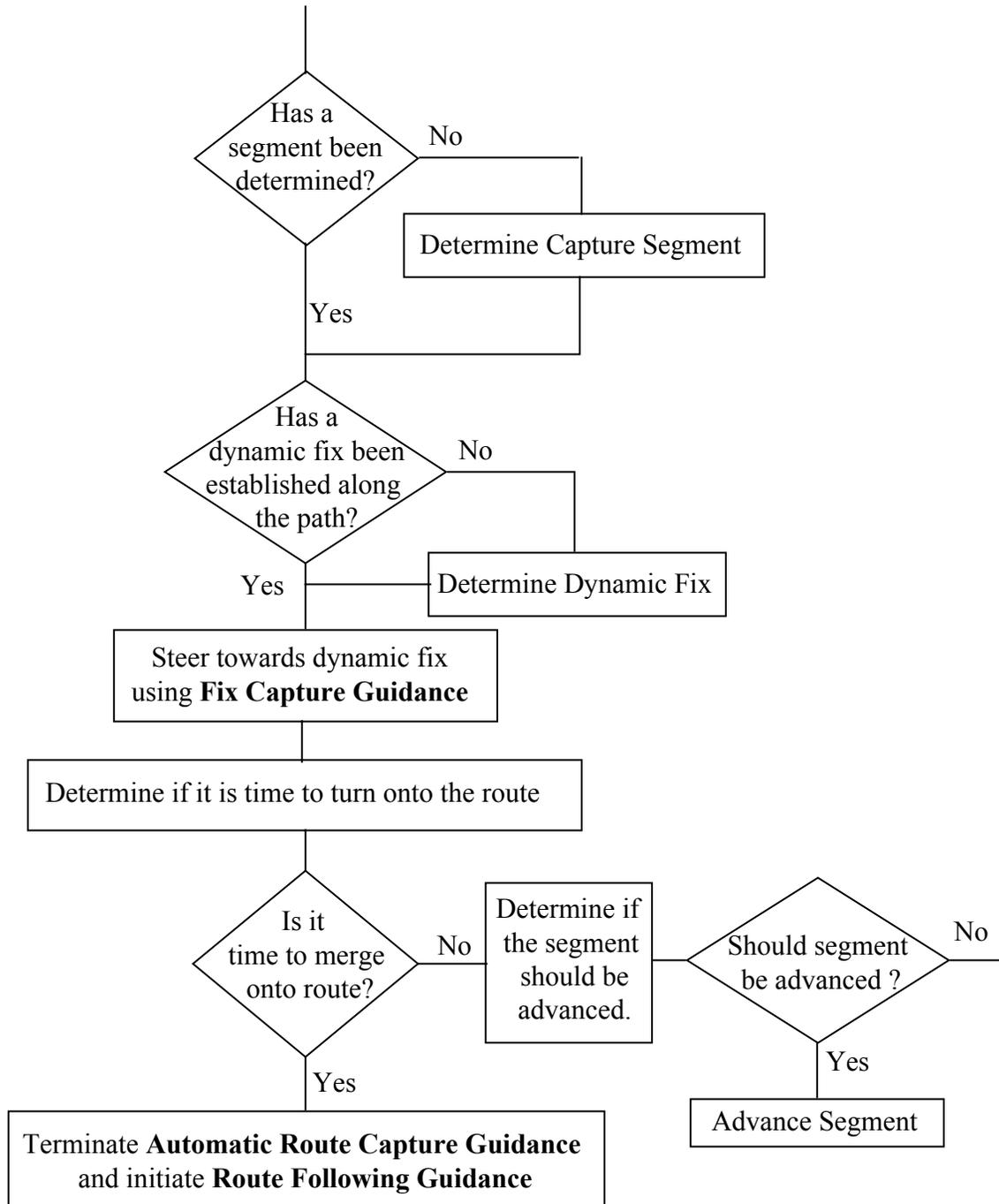


Figure 8.15. Logic for Automatic Route Capture Guidance

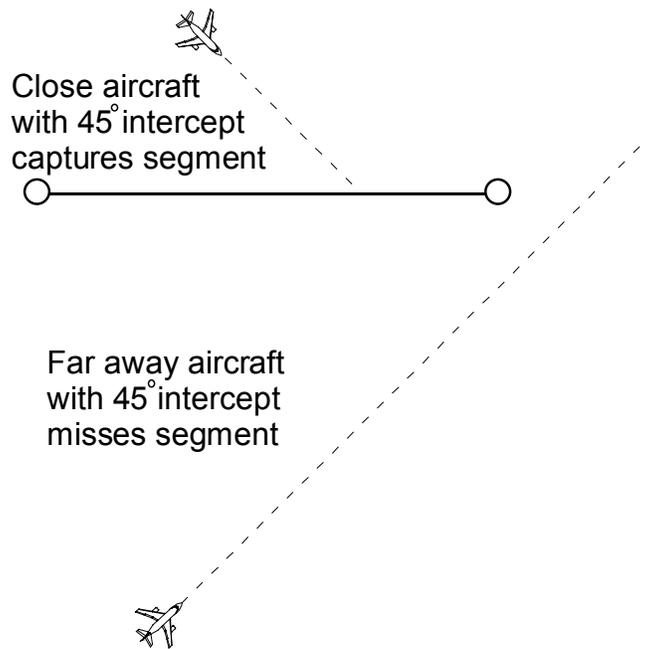


Figure 8.16. Illustration of aircraft using a 45 degree intercept

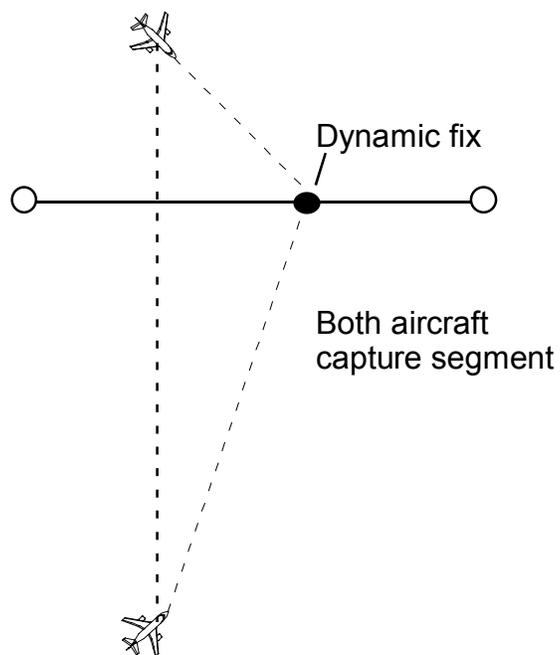


Figure 8.17. Illustration of two aircraft capturing a segment using a dynamic fix

To avoid the problem of aircraft overshooting capture segments, a dynamic fix is placed on the segment to be captured, and the aircraft is commanded to fly toward the dynamic fix. This situation is illustrated in Figure 8.17. In this case the further aircraft naturally uses a larger intercept angle. This system insures that the proper segment is captured and also provides some apparent variety in intercept angles so that all aircraft do not appear to behave the same. To calculate a dynamic fix, first consider the drawing in Figure 8.18.

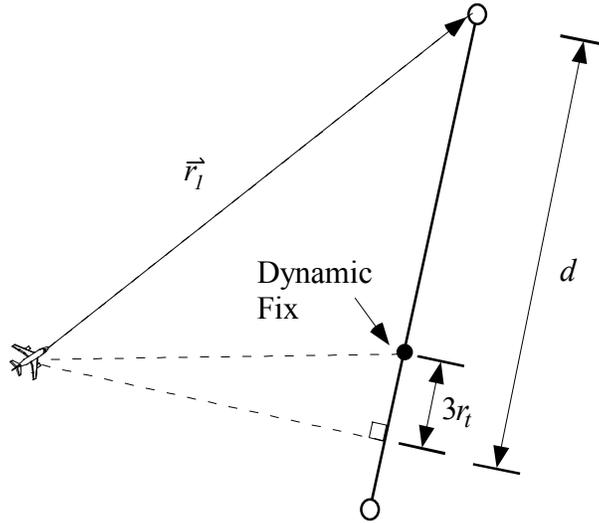


Figure 8.18. Determining an offset fix (dynamic fix) location

The distance d , the distance left to travel on a given segment, is determined by dotting \mathbf{r}_1 , the position vector from the aircraft to the leading fix, with a unit vector in the direction of the segment, \hat{r}_s . The dynamic fix distance from the leading fix, d_{offset} , is somewhat arbitrarily chosen to be three turn radii less than the distance d . The turn radius of the aircraft is notated, r_t .

The three-turn-radii distance was chosen to insure that the aircraft, regardless of its initial orientation or position, can capture the dynamic fix while still maintaining the proper general direction along the route. Generally, two turn radii would be sufficient, providing the aircraft does not speed up during the maneuver. However, three turn radii are used as a factor of safety just in case the aircraft increases its speed and hence its turn radius during the maneuver.

$$d = \mathbf{r}_1 \cdot \hat{r}_s \quad (8.13)$$

$$d_{offset} = d - 3r_t \quad (8.14)$$

We can create a vector, \mathbf{R}_{offset} , describing the location of the dynamic fix where \hat{x}_s is a unit vector pointing true North and \hat{y}_s is a unit vector pointing true East.

$$\mathbf{R}_{offset} = -d_{offset} \cos \psi_r \hat{x}_s - d_{offset} \sin \psi_r \hat{y}_s \quad (8.15)$$

However, to use the standard fix capture algorithm, the fix must be represented in terms of a latitude and a longitude. We can approximate the latitude of the fix by converting the \hat{x}_s component of the \mathbf{R}_{offset} vector to a degree value as done in Equation (8.16). Similarly, the longitude can be calculated in Equation (8.17).

$$\mu_{dyn} = \mu_{leading} - d_{offset} \cos \psi_r \frac{360^\circ}{2\pi r_e} \quad (8.16)$$

$$l_{dyn} = l_{leading} - d_{offset} \sin \psi_r \frac{360^\circ}{2\pi r_e \cos \mu_{dyn}} \quad (8.17)$$

8.4.4 Vectored Route Capture

The vectored route capture algorithm steers the aircraft along a fixed heading until the aircraft intercepts the route. Each time step, the algorithm determines which segment is best to capture and, each time step, the algorithm determines if it is time to merge onto the route. It should be noted that the algorithm has no control over the initial heading. Therefore, if the heading steers the aircraft away from the route, the guidance law is unable to do anything about it although it will provide a warning if the aircraft is unlikely to intercept the route. Figure 8.19 shows the basic algorithm for the guidance law. It should be noted that even though the aircraft is being vectored, it is necessary to determine the capture segment so that the pilot knows when to merge onto the route.

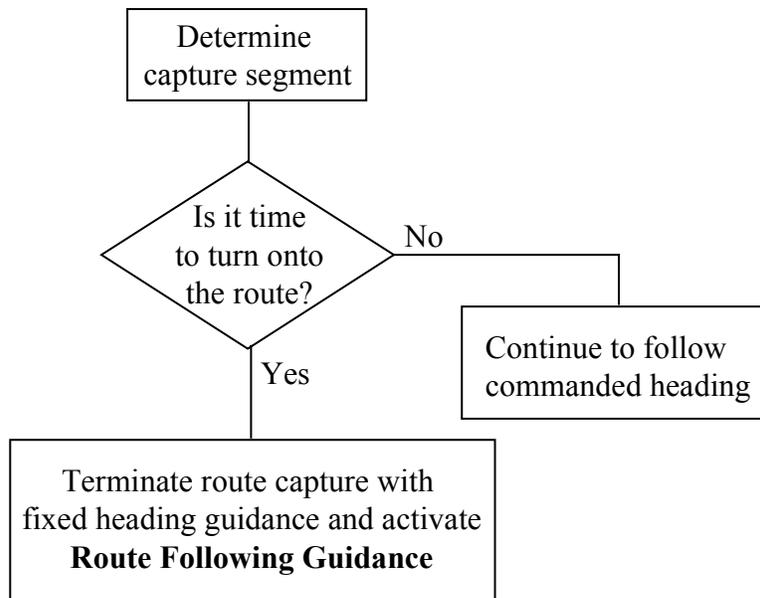


Figure 8.19. Route capture with fixed heading guidance

8.4.4.1 Determining Whether the Heading will Intercept a Route

While the algorithm has no control of the aircraft's initial heading, the algorithm will provide a warning if the heading chosen by the user is unlikely to intercept the given route. Basically, the algorithm measures whether or not the intercept angle crosses a segment and is related to the route following algorithm. The intercept angle ψ_t is calculated using Equation (8.18).

$$\psi_t = \psi_r - \psi_d \quad (8.18)$$

The terms are as follows:

- ψ_t : The intercept angle that the aircraft heading makes with the segment
- ψ_r : The bearing of the segment
- ψ_d : The desired heading

The resulting number is used in the logic presented in Figure 8.20

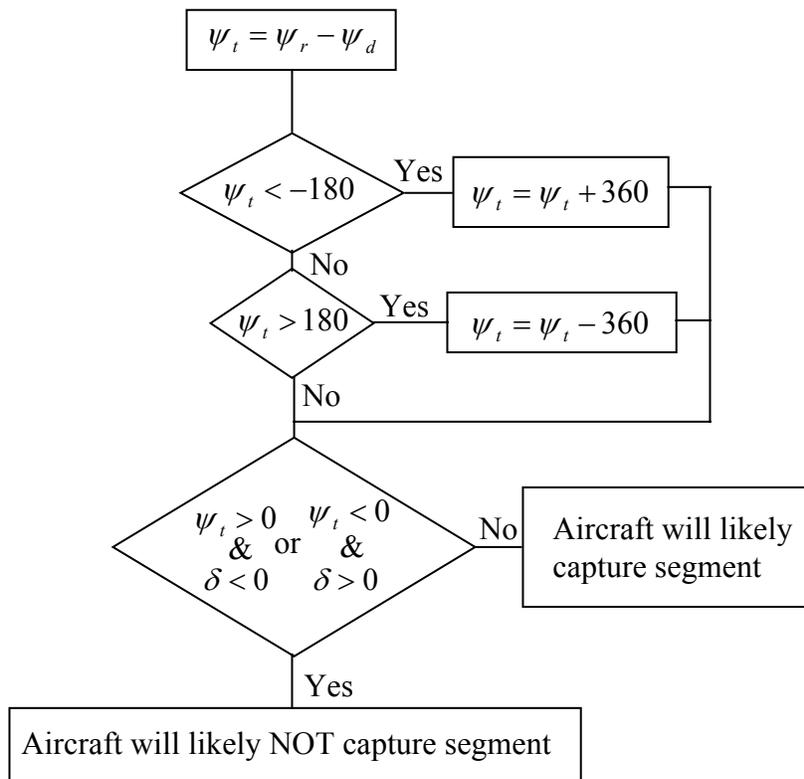


Figure 8.20. Logic for determining whether or not a heading will intercept a segment

8.4.5 Initial Fix Route Capture

The initial fix route capture algorithm is the simplest of all the capture algorithms because it neither needs to determine the appropriate capture segment nor determine when it should merge with the route. The algorithm flies to the first fix on the route. Once the aircraft crosses the fix, the route following algorithm is turned on and the aircraft follows the route. Because the algorithm is constrained to fly through the initial fix before turning to capture the first segment, the algorithm usually overshoots more when merging onto the route.

8.5 Basic Algorithms Required for Complete Functionality

To make the guidance system operate properly, some lower level functions are required. These functions are:

- Calculating the aircraft's turn radius
- Determining the lateral distance to a segment
- Determining the distance to go along a segment
- Determining the Rhumb line bearing and distance to a fix

8.5.1 Calculating the Aircraft Turn Radius

To calculate the aircraft's turn radius, r_t , a standard equation is used from Anderson[A89], Equation (8.19), where V_a is the aircraft's true airspeed, g is the gravitational acceleration, and n is the aircraft load factor.

$$r_t = \frac{V_a^2}{g\sqrt{n^2 - 1}} \quad (8.19)$$

The load factor for the aircraft is calculated by considering Equation (8.20) where L is the lift of the aircraft, ϕ is the bank angle, and W is the weight of the aircraft. In the simulation, we will assume that the aircraft always will provide enough lift to maintain level flight which is implied by the equality of Equation (8.20).

$$L \cos \phi = W \quad (8.20)$$

The load factor of an aircraft is defined as the lift over the weight. Assuming enough lift is provided to maintain level flight, the load factor can be determined as an exclusive function of bank angle.

$$n = \frac{L}{W} = \frac{L}{L \cos \phi} = \frac{1}{\cos \phi} \quad (8.21)$$

8.5.2 Determining the Aircraft's Lateral Distance from a Segment

The aircraft's lateral distance from the segment is calculated using vector operations. The dot product is taken of the position vector from the aircraft's location to the leading fix of the segment and a unit vector normal to the vector describing the segment itself. The expression is best represented mathematically in Equation (8.22) and Figure 8.21. To insure that the desired ground track for segment capture is correct, it is necessary for δ to be negative when on the left side of the segment and positive on the right side of the segment.

$$\delta = \mathbf{r}_{lf} \cdot \hat{\mathbf{n}}_s \quad (8.22)$$

The unit normal is represented in Equation (8.23).

$$\hat{\mathbf{n}}_s = \frac{R_{s_y}}{\|\mathbf{R}_s\|} \hat{\mathbf{x}}_s - \frac{R_{s_x}}{\|\mathbf{R}_s\|} \hat{\mathbf{y}}_s \quad (8.23)$$

The terms in the equations are defined as follows:

- δ : The lateral distance from the segment
- \mathbf{r}_{lf} : A vector from the aircraft's position to the leading fix
- $\hat{\mathbf{n}}_s$: A unit vector normal to the segment
- \mathbf{R}_s : A vector representing the segment

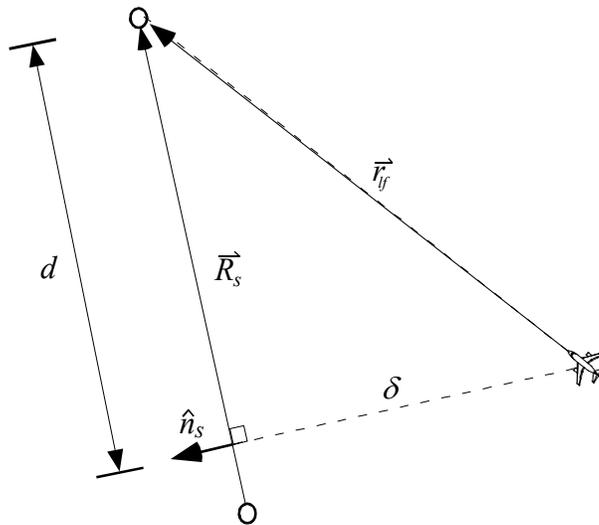


Figure 8.21. Illustration of distance calculation geometry

8.5.3 Determining Distance to go Along a Segment

Using the same nomenclature and geometry presented in Figure 8.21, the distance left to travel along the segment, d , can be calculated using Equation (8.24).

$$d = r_f \cdot \frac{\mathbf{R}_s}{\|\mathbf{R}_s\|} \quad (8.24)$$

8.5.4 Rhumb Line Bearing

The rhumb line is a line of constant course, or heading. The distance between the two fixes using the rhumb line can be much greater than a great circle arc if the fixes are far apart.

For a spherical earth, a rhumb line is a straight line drawn between the two points on a Mercator projection of the earth's surface. The mercator projection separately maps latitude and longitude to a planar surface. The rhumb line is then the hypotenuse of the triangle formed by the projection of the latitude change and longitude change onto that planar surface. Our derived equations should be consistent with the Mercator transformation equations of Clarke [C95].

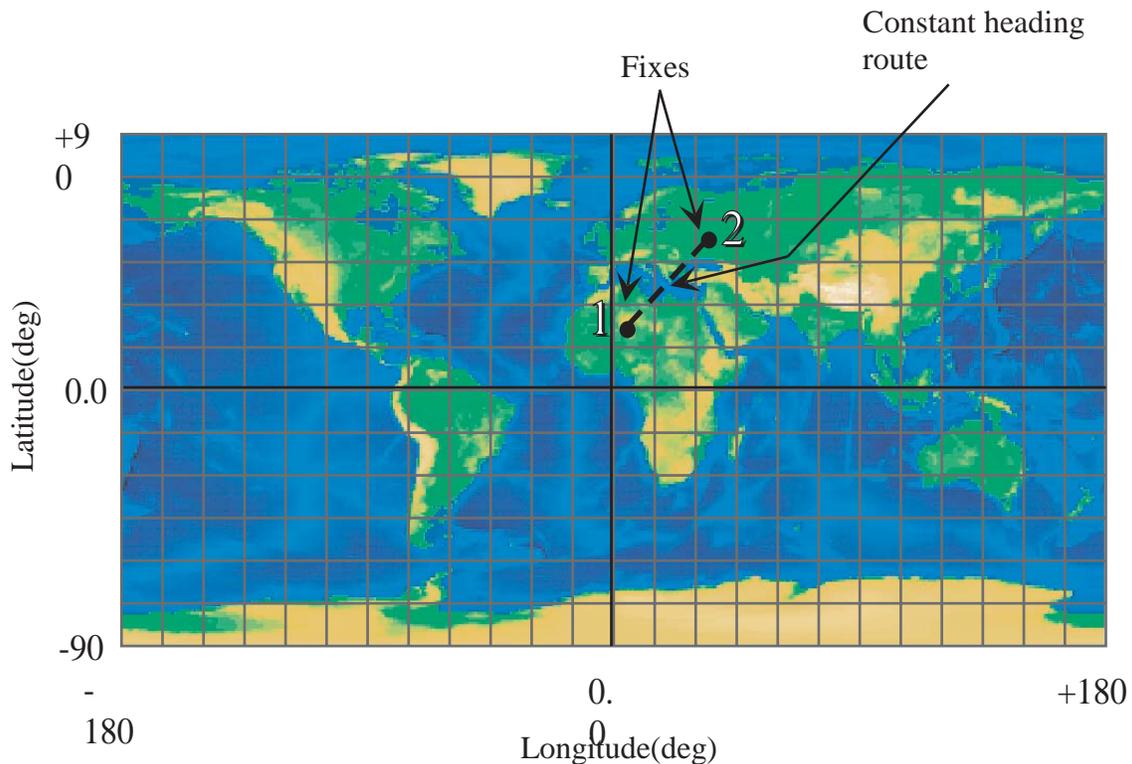


Figure 8.22. Cylindrical mapping of spherical Earth model. Shown are two fixes and the constant heading route between the fixes.

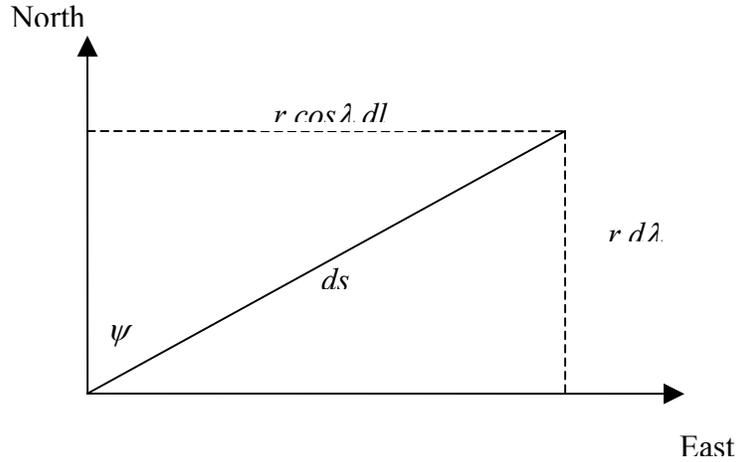


Figure 8.23 Geometry relating latitude, longitude, and bearing on a spherical earth

Referring to Figure 8.23, the changes in latitude and longitude along a rhumb line on a spherical earth are given by,

$$\begin{aligned}\tan \psi &= \frac{r \cos \lambda dl}{r d \lambda} \\ \tan \psi &= \frac{\cos \lambda dl}{d \lambda}\end{aligned}\quad (8.25)$$

which can be rewritten as,

$$dl = \tan \psi \frac{d \lambda}{\cos \lambda}$$

Upon integrating, we get,

$$\begin{aligned}\int_{l_1}^{l_2} dl &= \tan \psi \int_{\lambda_1}^{\lambda_2} \frac{d \lambda}{\cos \lambda} \\ l_2 - l_1 &= \tan \psi \left[\ln \left| \sec \lambda + \tan \lambda \right| \right]_{\lambda_1}^{\lambda_2} \\ l_2 - l_1 &= \tan \psi \left[\ln \left| \tan \left(\frac{\pi}{4} + \frac{\lambda}{2} \right) \right| \right]_{\lambda_1}^{\lambda_2} \\ l_2 - l_1 &= \tan \psi \left[\ln \left| \tan \left(\frac{\pi}{4} + \frac{\lambda_2}{2} \right) \right| - \ln \left| \tan \left(\frac{\pi}{4} + \frac{\lambda_1}{2} \right) \right| \right]\end{aligned}\quad (8.26)$$

Solving for the true heading, we get,

$$\tan \psi = \frac{(l_2 - l_1)}{\left[\ln \left| \tan \left(\frac{\pi}{4} + \frac{\lambda_2}{2} \right) \right| - \ln \left| \tan \left(\frac{\pi}{4} + \frac{\lambda_1}{2} \right) \right| \right]}$$

This is consistent with the equations for the equatorial Mercator projection as presented in Clarke [C95].

The longitudes must be analyzed so that the shorter path around the world is chosen. The following algorithm will normalize the longitude change for our purposes.

$$\begin{aligned} \Delta l &= l_2 - l_1 \\ \text{While } |\Delta l| > 180^\circ & \{ \Delta l = \Delta l + \text{sign}(\Delta l) * 360^\circ \} \end{aligned} \quad (8.27)$$

And the true heading equation becomes,

$$\tan \psi = \frac{\Delta l}{\left[\ln \left| \tan \left(\frac{\pi}{4} + \frac{\lambda_2}{2} \right) \right| - \ln \left| \tan \left(\frac{\pi}{4} + \frac{\lambda_1}{2} \right) \right| \right]} \quad (8.28)$$

Because the arctangent has a range $(-90^\circ, 90^\circ)$ and we want a heading in the range $(0^\circ, 360^\circ)$, we need to be careful about how we solve this equation. We, therefore, develop the 360° arctangent function.

$$\psi = \arctan_{360} \left(\left[\ln \left| \tan \left(\frac{\pi}{4} + \frac{\lambda_2}{2} \right) \right| - \ln \left| \tan \left(\frac{\pi}{4} + \frac{\lambda_1}{2} \right) \right| \right], \Delta l \right) \quad (8.29)$$

8.5.4.1 360° Arctangent Function

In this section, we develop an algorithm for the arctangent of a ratio of Cartesian coordinates in the range $(0^\circ, 360^\circ)$. Because the ratio is of the ordinate to the abscissa, we can adapt the range of the arctangent function per the quadrant of the coordinate pair.

Function $\theta = \arctan_{360}(\delta_{abscissa}, \delta_{ordinate})$

if $\delta_{abscissa} \geq 0$ and $\delta_{ordinate} \geq 0$

$$\theta = \arctan \left(\frac{\delta_{ordinate}}{\delta_{abscissa}} \right)$$

if $\delta_{abscissa} \geq 0$ and $\delta_{ordinate} \leq 0$

$\theta = \arctan\left(\frac{\delta_{ordinate}}{\delta_{abscissa}}\right) + 360^\circ$ <p style="text-align: left; margin-left: 40px;">else</p> $\theta = \arctan\left(\frac{\delta_{ordinate}}{\delta_{abscissa}}\right) + 180^\circ$ <p style="text-align: left; margin-left: 40px;">end</p>
--

8.5.5 Rhumb Line Distance

Once again referring to Figure 8.23, the incremental distance along a rhumb line on a spherical earth is given by,

$$\cos \psi = \frac{r_e d\lambda}{ds}$$

$$ds = \frac{r_e d\lambda}{\cos \psi}$$

which integrates to,

$$s = \left| \frac{r_e (\lambda_2 - \lambda_1)}{\cos \psi} \right| \quad (8.30)$$

Absolute value is used because we want a positive distance. Equation (8.30) does not apply for east-west rhumb lines. For these cases we use an alternate relation, also obtained from the geometry of Figure 8.23.

$$ds_{EW} = r_e \cos \lambda dl \quad (8.31)$$

Since the latitude is constant, this integrates to,

$$s_{EW} = |r_e \cos \lambda \Delta l| \quad (8.32)$$

The question arises as to what earth radius to use in the calculations. A good approximation to the spherical earth radius is to use the local radius of the first point in the WGS-84 earth model, which is given by using equations (2.107) and (2.108).

8.5.6 Creating Vectors Representing Segments

There is a need to represent segments as vectors. To create a vector, a magnitude and bearing are required. Generally, the rhumb line information is used. A segment's bearing

is the rhumb line bearing between the two endpoint fixes that make up the segment, and the segment's length is the rhumb line distance between the two fixes. The vector components are represented in the surface frame as shown in Equation (8.33)

$$\mathbf{R}_s = s_s \cos \psi_s \hat{x}_s + s_s \sin \psi_s \hat{y}_s \quad (8.33)$$

The nomenclature is defined as follows:

- \mathbf{R}_s : The vector representing the segment
- s_s : The rhumb line distance of the segment
- ψ_s : The rhumb line bearing between the trailing and leading fixes of the segment
- \hat{x}_s, \hat{y}_s : Unit vectors representing the x, y components of the surface frame.

9. Navigation Error Modeling

The purpose of navigation error modeling is to model the variances which occur in aircraft flight paths as a result of imperfect information. Three different navigation systems are modeled: These are:

- VOR/DME navigation
- GPS navigation
- ILS navigation

The two navigation types generally used for en route types of operation are VOR/DME and GPS navigation. The ILS model is used only for approach to landing. All of the navigation error models perform similarly in that they create a perturbed estimate of the aircraft's location for the guidance system to use as an input. Therefore, the navigation error models all return a latitude-longitude pair which represents the aircraft's position as determined by imperfect navigation.

9.1 VOR/DME Navigation

Aircraft which use VOR/DME navigation are relying on a network of ground based VOR transmitters for bearing information and DME for distance information. The aircraft use range and bearing information from VOR/DME stations of known position to estimate the position of the aircraft. However, pure VOR/DME navigation puts more constraints on the problem in that aircraft usually always fly either to or from a VOR/DME station along a predetermined radial rather than using some area navigation (RNAV) technique. Therefore, not only does the aircraft's position need to be calculated, but also a technique to determine which VOR/DME is most appropriate for navigation must also be determined.

The process of VOR/DME navigation can be broken into two parts. These are:

- Determining the aircraft position from a given VOR/DME station
- Determining which VOR/DME station is best used for navigation

9.1.1 Determining Aircraft Position from a VOR/DME Station

The VOR transmitters send a line-of-sight RF signal that provides a bearing of the airborne receiver with respect to magnetic north. In the following discussion, it is assumed that the magnetic bearing angle, has been corrected with the magnetic bearing correction, to yield the geodetic bearing angle, B : The VOR/DME navigation error model takes perfect information about the aircraft's position and corrupts it according to the range and bearing biases for a given VOR/DME. This corrupted aircraft position information is sent to the guidance system which guides the aircraft using the corrupted information.

Consider the illustration in Figure 9.1. The estimated aircraft position is in error from the actual aircraft position by a certain range error, $\Delta\rho$, and bearing error, ΔB . Generally, the range and bearing from the station to the aircraft is calculated using the rhumb line bearing and distance algorithms in Section 8.

- ρ : The range to the station (nm)
- B : The bearing from the station (deg)
- $\Delta\rho$: The total range error (nm)
- ΔB : The total bearing error

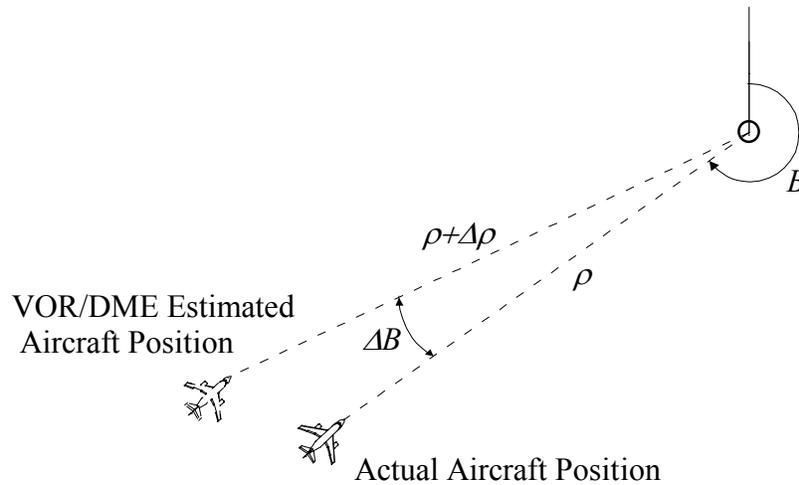


Figure 9.1. An illustration of the range and bearing from the station

The actual aircraft position in the NED or surface frame from the station is represented in (x,y) coordinates as defined in equations (9.1) and (9.2) where the terms x_{act} and y_{act} are the actual (x,y) locations for the aircraft.

$$x_{act} = \rho \cos B \quad (9.1)$$

$$y_{act} = \rho \sin B \quad (9.2)$$

The estimated location of the aircraft as determined from the range and bearing error is defined by equations (9.3) and (9.4).

$$x_{est} = (\rho + \Delta\rho) \cos(B + \Delta B) \quad (9.3)$$

$$y_{est} = (\rho + \Delta\rho) \sin(B + \Delta B) \quad (9.4)$$

The position error can be represented with Δx and Δy . Considering the x equation first, we can write

$$\Delta x = (\rho + \Delta\rho) \cos(B + \Delta B) - \rho \cos B \quad (9.5)$$

$$\Delta x = (\rho + \Delta\rho)(\cos B \cos \Delta B - \sin B \sin \Delta B) - \rho \cos B \quad (9.6)$$

$$\Delta x = \rho \cos B \cos \Delta B - \rho \sin B \sin \Delta B + \Delta\rho \cos B \cos \Delta B - \Delta\rho \sin B \sin \Delta B - \rho \cos B \quad (9.7)$$

Linearizing with respect to the error biases, we have equation (9.8).

$$\begin{aligned} \left. \frac{\partial \Delta x}{\partial \Delta B} \right|_{ref} \Delta B + \left. \frac{\partial \Delta x}{\partial \Delta \rho} \right|_{ref} \Delta \rho = & -\rho \cos B \sin \Delta B \Big|_{ref} \Delta B - \rho \sin B \cos \Delta B \Big|_{ref} \Delta B \\ & - \Delta\rho \cos B \sin \Delta B \Big|_{ref} \Delta B - \Delta\rho \sin B \cos \Delta B \Big|_{ref} \Delta B \\ & + \cos B \cos \Delta B \Big|_{ref} \Delta \rho - \sin B \sin \Delta B \Big|_{ref} \Delta \rho \end{aligned} \quad (9.8)$$

Assuming a reference condition of $\Delta\rho = \Delta B = 0$, the linearized equation reduces to equation (9.9).

$$\Delta x = -\rho \Delta B \sin B + \Delta\rho \cos B \quad (9.9)$$

The y equation can be manipulated similarly.

$$\Delta y = (\rho + \Delta\rho) \sin(B + \Delta B) - \rho \sin B \quad (9.10)$$

$$\Delta y = (\rho + \Delta\rho)(\sin B \cos \Delta B + \cos B \sin \Delta B) - \rho \sin B \quad (9.11)$$

$$\Delta y = \rho \sin B \cos \Delta B + \rho \cos B \sin \Delta B + \Delta\rho \sin B \cos \Delta B + \Delta\rho \cos B \sin \Delta B - \rho \sin B \quad (9.12)$$

Linearizing we have equation (9.13).

$$\begin{aligned} \left. \frac{\partial \Delta y}{\partial \Delta \rho} \right|_{ref} \Delta \rho + \left. \frac{\partial \Delta y}{\partial \Delta B} \right|_{ref} \Delta B = & \sin B \cos \Delta B \Big|_{ref} \Delta \rho + \cos B \sin \Delta B \Big|_{ref} \Delta \rho \\ & - \rho \sin B \sin \Delta B \Big|_{ref} \Delta B + \rho \cos B \cos \Delta B \Big|_{ref} \Delta B \\ & - \Delta\rho \sin B \sin \Delta B \Big|_{ref} \Delta B + \Delta\rho \cos B \cos \Delta B \Big|_{ref} \Delta B \end{aligned} \quad (9.13)$$

Assuming a reference condition of $\Delta\rho = \Delta B = 0$, the linearized equation reduces to equation (9.14).

$$\Delta y = \Delta\rho \sin B + \rho \Delta B \cos B \quad (9.14)$$

Arranging in Vector form, we have equation (9.15)

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \cos B & -\rho \sin B \\ \sin B & \rho \cos B \end{bmatrix} \begin{bmatrix} \Delta \rho \\ \Delta B \end{bmatrix} \quad (9.15)$$

Generally, both the ground station and the airborne receiving equipment contribute to the range error and bias error. The terms $\delta\rho_{VD_A}$ and δB_{T_A} characterize the airborne receiver biases. These terms are randomly generated when the aircraft is initialized. The VOR/DME station has errors, $\delta\rho_{VD_G}$ and δB_{T_G} , which need to be obtained from the VOR/DME station itself. Depending on what quadrant the aircraft is in with respect to the VOR/DME, the bias can be different. A VOR/DME station needs some way of returning the correct bias information when prompted with the bearing from the station, B . Figure 9.2 illustrates the relationship between the VOR/DME and the four quadrants. For now, it may be easier to only have one bias per VOR/DME.

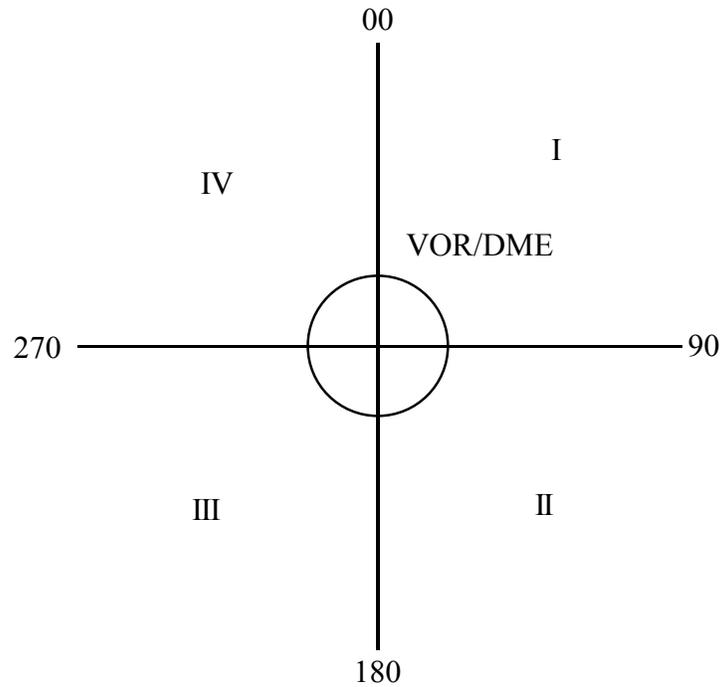


Figure 9.2. Illustration of the quadrants of the compass rose with respect to a VOR/DME station

When the airborne and ground station biases are summed, they can be inserted into equation (9.15) resulting in equation (9.16).

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \cos B & -\rho \sin B \\ \sin B & \rho \cos B \end{bmatrix} \begin{bmatrix} (\delta\rho_{VD_G} + \delta\rho_{VD_A}) \\ (\delta B_{T_G} + \delta B_{T_A}) \end{bmatrix} \quad (9.16)$$

However, the position of the aircraft is represented in terms of longitude and latitude. Therefore a conversion must be made. Two conversion factors are used. These are:

- $\frac{nm}{deg_{\mu}}$: Nautical miles per degree of latitude
- $\frac{nm}{deg_l}$: Nautical miles per degree of longitude

Equations (17) through (19) are used to calculate these conversion factors.

$$\frac{nm}{deg_{\mu}} = \frac{2\pi r_e}{360} \quad (17)$$

$$r_l = r_e \cos \mu_{ac} \quad (18)$$

$$\frac{nm}{deg_l} = \frac{2\pi r_l}{360} \quad (19)$$

The terms in the equations are defined as follows:

- r_e : The radius of the Earth in nautical miles
- r_l : The radius from the polar axis to the surface of the Earth at a given latitude
- μ_{ac} : The aircraft's current latitude

Finally, the aircraft's estimated (corrupted) position can be calculated using equations (9.20) and (9.21).

$$\mu_e = \mu_{ac} + \frac{\Delta x}{\frac{nm}{deg_{\mu}}} \quad (9.20)$$

$$l_e = l_{ac} + \frac{\Delta y}{\frac{nm}{deg_l}} \quad (9.21)$$

The terms in the equations are defined as follows:

- μ_{ac} : The aircraft's current actual latitude
- l_{ac} : The aircraft's current actual longitude
- μ_e : The aircraft's estimated latitude
- l_e : The aircraft's estimated longitude

The estimated values are the final return values.

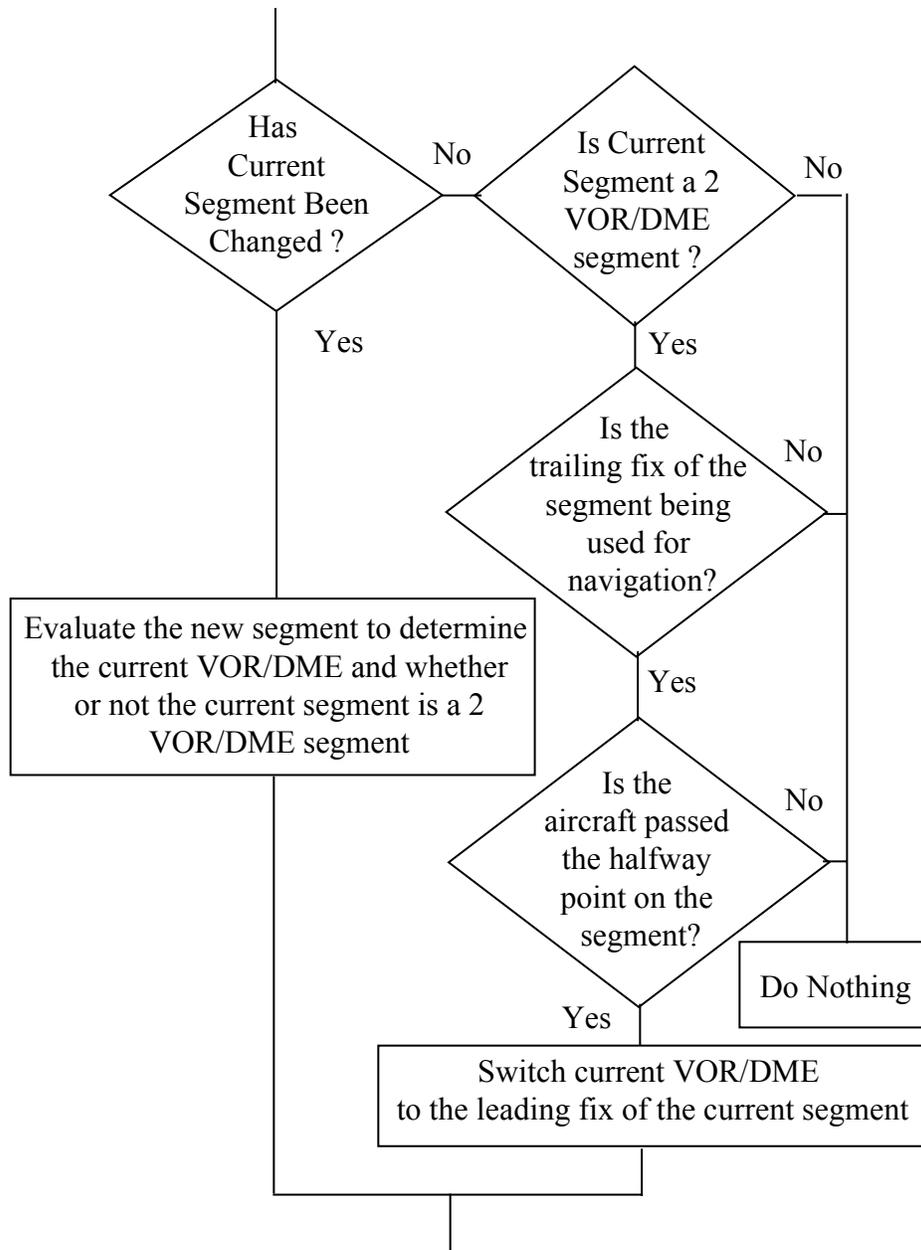


Figure 9.3. Logic for determining if the current VOR/DME used for navigation should be changed

The DME ground equipment accuracy is 0.05nm (1σ) while the airborne equipment accuracy is 0.25nm (1σ) or 1.5% (1σ) of range, whichever is greater (AC90-45A). The accuracy of the VOR ground equipment is 0.95^0 (1σ) while the airborne equipment is 1.5^0 (1σ) (AC90-45A). By far, the greatest contributor to the navigation error is the bearing accuracy. The DME error plays a small role.

Slant range error is not accounted for explicitly in the model. This is because the altitude of each VOR/DME, which would need to be known to make the calculation, is not known. Furthermore, since the slant range error of a given situation can be estimated by

the pilot, the pilot is likely to compensate for it when crossing fixes and capturing radials. Therefore, slant range error is unlikely to contribute greatly to the navigation error.

9.1.2 Determining the Proper VOR/DME Station to use for Navigation

When an aircraft is navigating using VOR/DME navigation, the pilot must tune in the VOR/DME which is associated with the particular segment which he/she is flying. The proper nav-aid information would be retrieved from the chart used for navigation. This level of realism does not exist explicitly in the TGF simulation because capturing every detail and nuance of the low altitude victor routes and high altitude jet routes would be prohibitively expensive to implement. Therefore, the victor and jet routes are not being explicitly followed. Rather, the aircraft only has knowledge of the fixes on the route and whether or not those fixes are VOR/DME stations or intersections. Because of this simplification, the navigation system must look at the available nav-aids along the route and determine which one would be most appropriate to use for navigation.

Figure 9.3 contains the logic which is used to determine whether or not the current VOR/DME should be switched. The logic can also be expressed as a set of the following rules:

- A fix is either a VOR/DME or an intersection
- A segment is defined by two fixes which are located at the endpoints of the segment
- If one of the fixes associated with the segment is a VOR/DME, that VOR/DME is used for navigation.
- If both fixes associated with the segment are VOR/DME's , then the VOR/DME closest to the aircraft is used for navigation.
- If neither fix is a VOR/DME, then a search is done to find the best VOR/DME along the route to use.

It is also worth noting that the navigation algorithms have nothing to do with switching segments. However, VOR/DME navigation needs to be aware of switches when they occur. If the current VOR/DME needs to be switched, the logic in Figure 9.4 must be used.

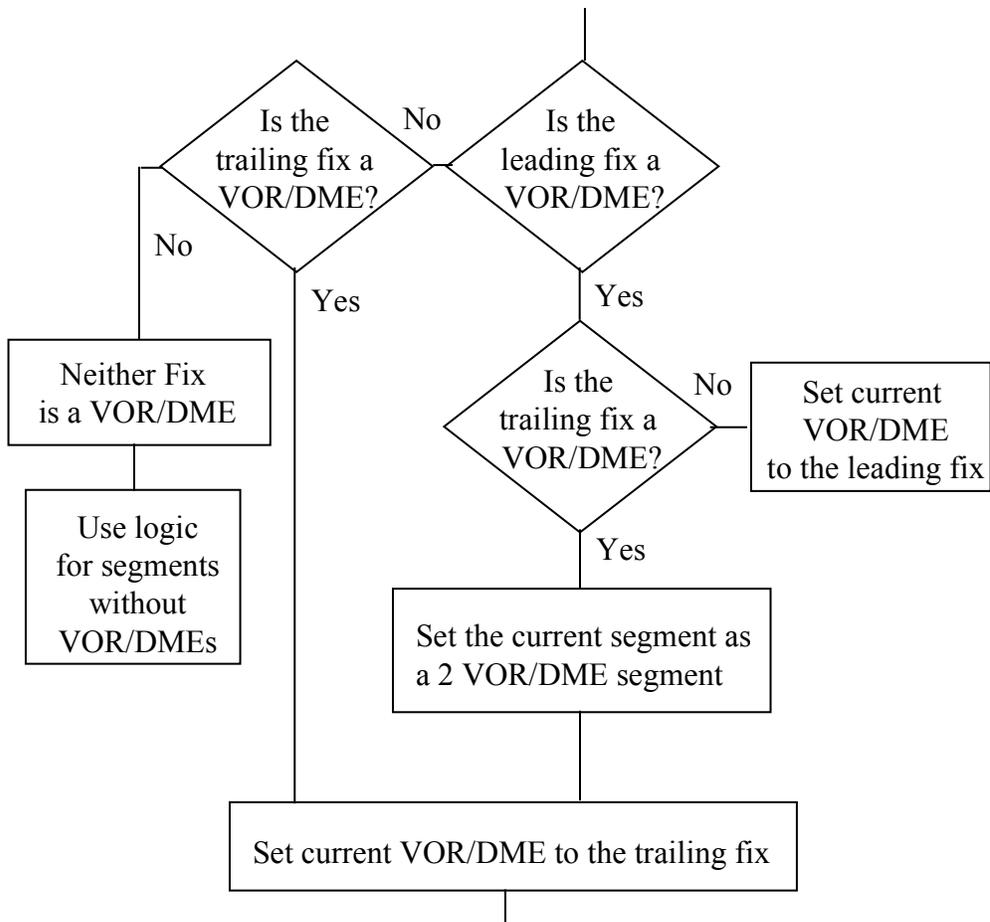


Figure 9.4. Logic for determining the appropriate VOR/DME for the next segment

Generally, there are either one or two VOR/DMEs on the segment. When there is only one VOR/DME, that VOR/DME is used. If there are two VOR/DMEs on the segment, the aircraft must use the closest one. When the segment does not have a VOR/DME associated with it, one must be determined. The algorithm must decide which of two VOR/DMEs is most appropriate. These two VOR/DMEs are:

- The previous VOR/DME which was used for navigation on the last segment
- The next VOR/DME that lies along the route but not on the current segment

Such a scenario is illustrated in Figure 9.5. The aircraft lies on a segment which does not have a VOR/DME but it is in between two segments that do have a VOR/DME. From inspection of the drawing, we can see that the next VOR/DME along the route is a much better choice because the current segment lies along a radial of the next VOR along the route.

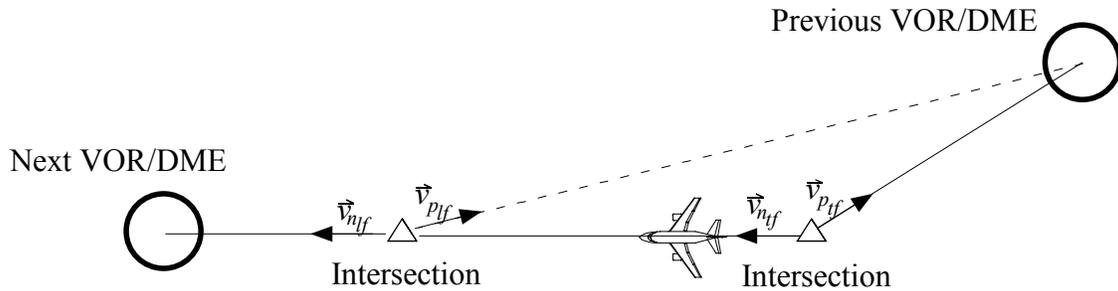


Figure 9.5. An illustration of the geometry used to determine which VOR/DME should be used for segments without a VOR/DME

To algorithmically draw the same conclusion, there are four unit vectors that must be calculated. The calculations can be made with a bearing calculation algorithm along with the vector tool of choice. It is imperative to the function of this algorithm that the vectors be unit vectors and not vectors of unequal magnitudes. These vectors are

- \vec{v}_{n_f} : A unit vector from the trailing fix to the next VOR/DME
- \vec{v}_{p_f} : A unit vector from the leading fix to the next VOR/DME
- \vec{v}_{n_f} : A unit vector from the trailing fix to the previous VOR/DME
- \vec{v}_{p_f} : A unit vector from the leading fix to the previous VOR/DME

The unit vectors associated with each VOR/DME are then dotted with each other. Ideally, the dot product is equal to unity for a perfect match between a VOR/DME and a segment. However, for the purposes of the algorithm, we choose the higher value of equations (9.22) and (9.23) as shown in Figure 9.6.

$$\vec{v}_{n_f} \cdot \vec{v}_{n_f} \tag{9.22}$$

$$\vec{v}_{p_f} \cdot \vec{v}_{p_f} \tag{9.23}$$

The higher value indicates that the vectors are pointing nearly in the same direction. This indicates that the segment lies along a radial to the VOR/DME in question which makes it a good candidate for navigation.

There will be cases when neither VOR/DME is appropriate for navigation. In this case, the algorithm still chooses the highest dot product; however, it can not really be said that the aircraft is following a radial To or From a VOR. The aircraft is area navigating instead. This is not necessarily a realistic procedure for an aircraft that is flying using VOR/DME navigation; however, when such anomalies in the flight plan occur, it is best that the aircraft continue to fly rather than indicate an exception.

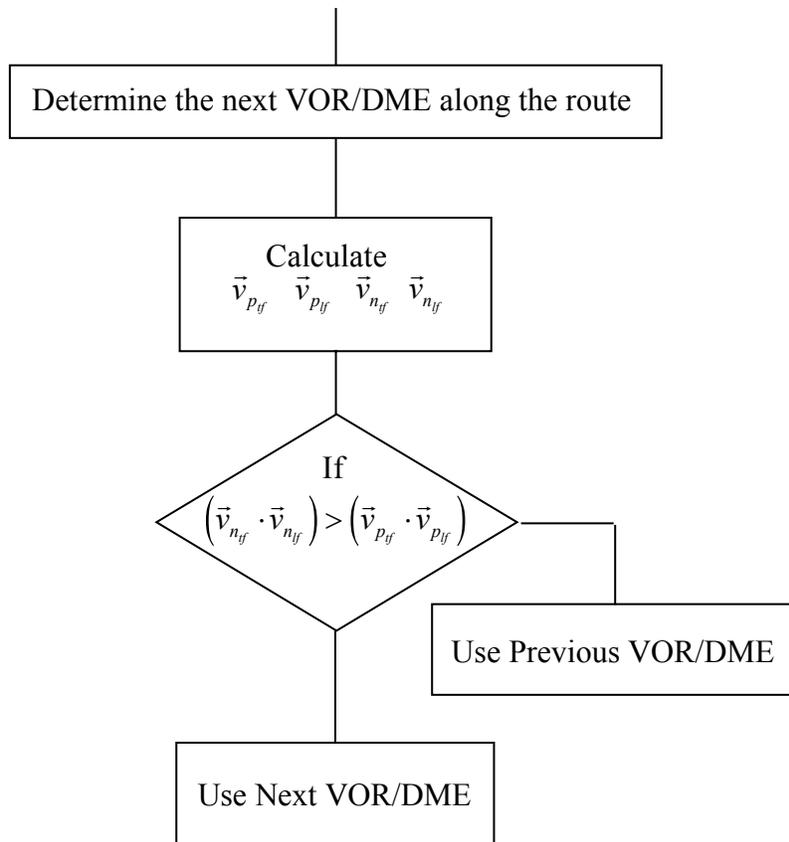


Figure 9.6. Logic For determining which VOR/DME to use when no VOR/DME lies along route

9.2 GPS Navigation

There are a number of error sources that contribute to the aircraft GPS position and velocity error; however, the dominant error is the GPS satellite clock error, $b_{SAT,k}$. Figure 9.7 illustrates the basic satellite geometry.

Since the exact model for GPS satellite Selective Availability (SA) clock error is classified, a number of authors have approximated it using a second order Gauss Markov model in the pseudorange domain [PS96]. These models can be used to determine the GPS receiver position and velocity errors as follows. Starting with the SA clock error model for each visible satellite, the SA clock pseudorange and range rate errors can be obtained. These errors can then be translated into a local coordinate frame, such as a local east-north-up (ENU) frame using the methodology described earlier.

An equivalent local coordinate system SA position and velocity error model can be formulated, to avoid the need to model the GPS satellite orbits which are required to determine the line-of-sight direction cosines. It also avoids the need for a Least Squares filter.

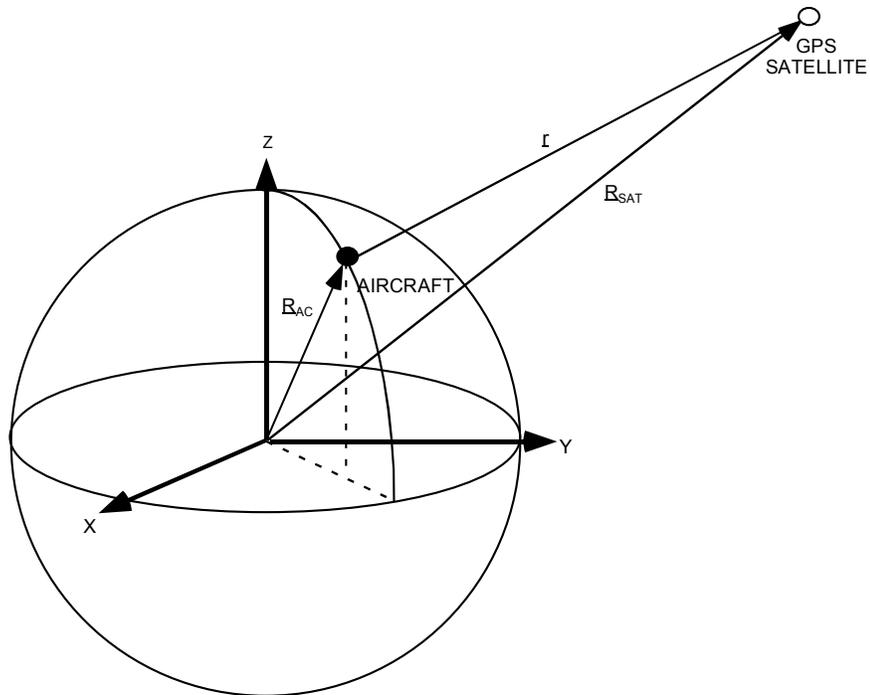


Figure 9.7 GPS Receiver Measurement Geometry

The approach that is used is to start with a second-order Gauss Markov SA pseudorange model. Then, by adjusting the parameters of this model to match observed local coordinate SA position error statistics, it is possible to obtain a simplified SA position and velocity error model.

In general, when the uncorrelated SA pseudorange and range rate errors are translated into SA position and velocity errors, the resulting position and velocity errors are correlated. The fundamental simplifying assumption that will be used is to assume that these correlations are negligible. If the GPS satellites were located directly overhead and exactly on the horizon at the four cardinal directions, the correlations would indeed be zero.

Table 9.1. Observed Local Coordinate Position Root-Mean-Square (rms) Errors

Coordinate	Position Error	Measured Average Daily Variations over 30 Days
East	32 m	15%
North	31 m	14%
Horizontal	41.5* m & 37.5** m	10%
Vertical	67 m	10%

* Based on observed horizontal position errors

** Based on observed steady state horizontal position time difference errors

The available field data, obtained from [TT90] consists of the position error statistics of Table 9.1. In addition to the day-to-day variability, there is also a latitude dependence for the vertical error, particularly for latitudes greater than 60 degrees.

In addition, field data was abstracted from [TT90] to describe the horizontal temporal decorrelation.

A general second-order Gauss Markov model is described by the second-order differential equation [G74]:

$$\ddot{x} + 2\beta\omega_0\dot{x} + \omega_0^2x = cw \quad (9.24)$$

$$\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\beta\omega_0 \end{bmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ c \end{pmatrix} w \quad (9.25)$$

where, x, v = error and error derivative ($v = \dot{x}$)

β = damping factor

ω_0 = natural frequency

w = Gaussian white noise

c = scale factor

equations (9.24) or (9.25) constitutes the simplified SA position and velocity error model. There is one set of these equations for each east, north, and vertical component.

The next step is to select the three unknown parameters, σ_p , σ_v , and β , to match the observed data statistics. The results are summarized in Table 9.2. A one-hour sample history for all three local position and velocity SA error components is illustrated in Figure 9.8 and Figure 9.9.

Table 9.2 Simplified vs. Observed SA Position and Velocity Model Parameters

Parameter	Symbol	Observed	Predicted
North Position Sigma	σ_{pN}	31 m	31 m
East Position Sigma	σ_{pE}	32 m	32 m
Vertical Position Sigma	σ_{pV}	67 m	67 m
North Velocity Sigma	σ_{vN}		0.38 m/s
East Velocity Sigma	σ_{vE}		0.39 m/s
Vertical Velocity Sigma	σ_{vV}		0.82 m/s
Damping Factor	β		0.55

Natural Frequency	ω_0		0.0122
	ω_I		0.0102
	c^2		0.0021
Horizontal Position Sigma	σ_{pH}	41.5* m & 37.5** m	37.3 m
Horizontal Position Correlation	ρ_{NE}	-0.13* & -0.29**	-0.3

* Based on observed horizontal position errors

** Based on observed steady state horizontal position time difference errors in Figure 9.8.

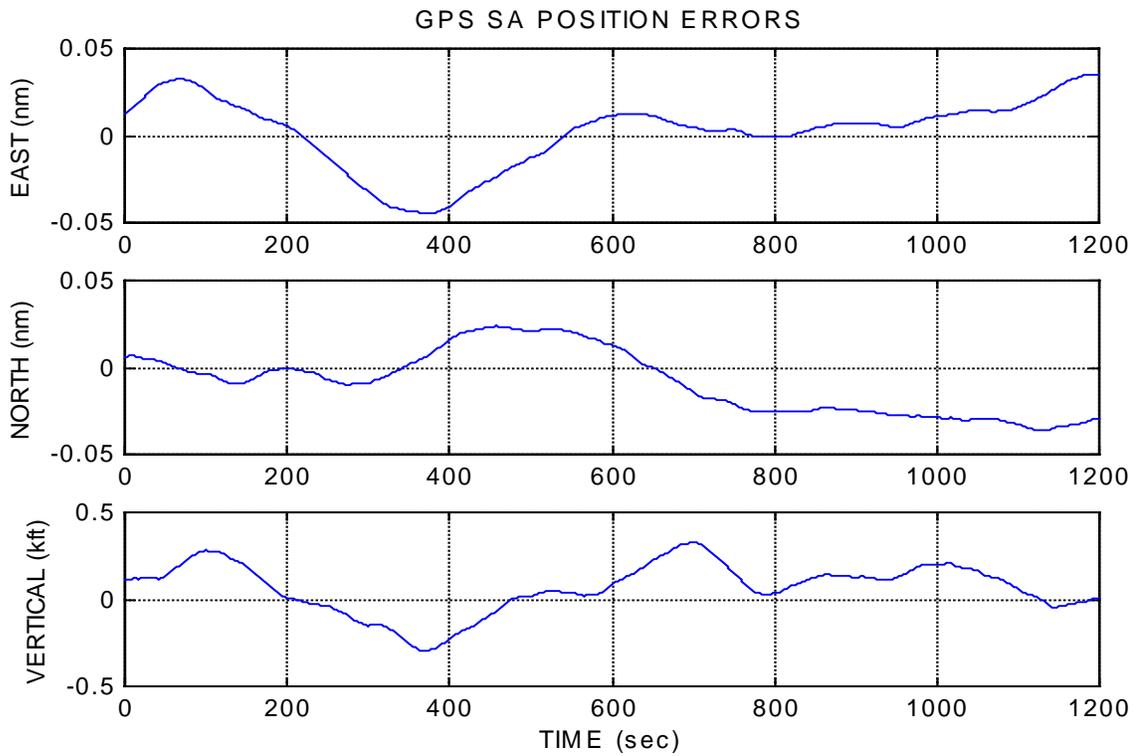


Figure 9.8. Monte Carlo Simulated SA Position Errors

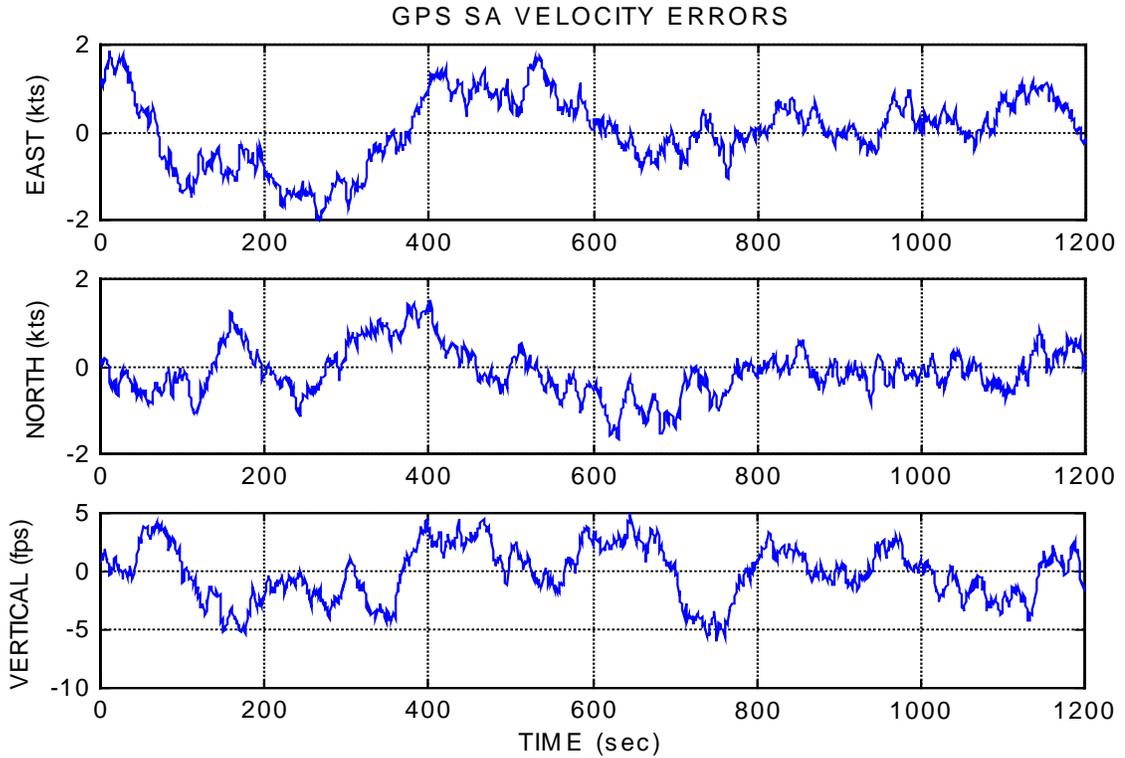


Figure 9.9. Monte Carlo Simulated SA Velocity Errors

9.3 Discretizing the Continuous 2nd Order Gauss Markov Process

To implement a 2nd order Gauss Markov process in code, it must be discretized. The corresponding closed-form 2nd order difference equation is shown in equation (9.26) [PS96]:

$$\begin{pmatrix} x \\ v \end{pmatrix}_{k+1} = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \begin{pmatrix} x \\ v \end{pmatrix}_k + \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ 0 & \gamma_{22} \end{bmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad (9.26)$$

The discrete input matrix has three terms and the state transition matrix has four terms within them that need to be calculated as shown in equations (9.27) and (9.28).

$$\Phi = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \quad (9.27)$$

$$\Gamma = \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ 0 & \gamma_{22} \end{bmatrix} \quad (9.28)$$

To calculate the state transition matrix variables, it is necessary to calculate some preliminary terms. These terms are defined in equations (9.29) and (9.30).

$$\omega_0 = \left(\frac{\sigma_v}{\sigma_p} \right) \quad (9.29)$$

$$\omega_1 = \left(\frac{\sigma_v}{\sigma_p} \right) \sqrt{1 - \beta^2} \quad (9.30)$$

The state transition matrix terms are then defined in equations (9.31) through (9.34).

$$\phi_{11} = e^{-\beta\omega_0\Delta t} \left[\cos(\omega_1\Delta t) + \beta(\omega_0 / \omega_1) \sin(\omega_1\Delta t) \right] \quad (9.31)$$

$$\phi_{12} = (1 / \omega_1) e^{-\beta\omega_0\Delta t} \left[\sin(\omega_1\Delta t) \right] \quad (9.32)$$

$$\phi_{21} = -\omega_0^2 \phi_{12} \quad (9.33)$$

$$\phi_{22} = e^{-\beta\omega_0\Delta t} \left[\cos(\omega_1\Delta t) - \beta(\omega_0 / \omega_1) \sin(\omega_1\Delta t) \right] \quad (9.34)$$

To calculate the discrete input matrix, an additional term is needed which is shown in equation (9.35).

$$c^2 = 4\beta \left(\frac{\sigma_v^3}{\sigma_p} \right) \quad (9.35)$$

The terms for the discrete input matrix are shown in equations (9.36) through (9.38) where the terms Q_{11} , Q_{12} , and Q_{22} , are terms of the white noise error covariance matrix and are defined in equations (9.39) through (9.41).

$$\gamma_{11} = \sqrt{Q_{11} - Q_{12}^2 / Q_{22}} \quad (9.36)$$

$$\gamma_{12} = Q_{12} / \sqrt{Q_{22}} \quad (9.37)$$

$$\gamma_{22} = \sqrt{Q_{22}} \quad (9.38)$$

$$Q_{11} = \frac{c^2}{4\beta\omega_0^3} \left[1 - \left(\frac{\omega_0}{\omega_1} \right)^2 e^{-2\beta\omega_0\Delta t} \left[1 - \beta^2 \cos(2\omega_1\Delta t) + \beta(\omega_1/\omega_0) \sin(2\omega_1\Delta t) \right] \right] \quad (9.39)$$

$$Q_{12} = Q_{21} = \frac{c^2}{4\omega_1^2} \left[e^{-2\beta\omega_0\Delta t} \left[1 - \cos(2\omega_1\Delta t) \right] \right] \quad (9.40)$$

$$Q_{22} = \frac{c^2}{4\beta\omega_0} \left[1 - \left(\frac{\omega_0}{\omega_1} \right)^2 e^{-2\beta\omega_0\Delta t} \left[1 - \beta^2 \cos(2\omega_1\Delta t) - \beta(\omega_1/\omega_0) \sin(2\omega_1\Delta t) \right] \right] \quad (9.41)$$

The state variables of the process, x and v , are initialized using equations (9.42) and (9.43) where the terms w_1 and w_2 are unit variance discrete Gaussian white noise. (Gaussian random numbers).

$$x = \sigma_p w_1 \quad (9.42)$$

$$v = \sigma_v w_2 \quad (9.43)$$

9.4 ILS Localizer Error Model

For an ILS localizer, the measured lateral deviation is the angle, ΔB_l . This can be converted into a lateral position error as follows. The slant range, r_{IT} to the runway, is approximated using the rhumb line distance algorithm. Then the lateral position, r_{CT} is:

$$r_{CT} = r_{IT} \Delta B_l \quad (9.44)$$

The deviation angle is comprised of errors from ground based equipment and airborne equipment as shown in equation (9.45) where $\Delta B_{l,G}$ is the ground based component and $\Delta B_{l,A}$ is the airborne component.

$$\Delta B_l \equiv \Delta B_{l,G} + \Delta B_{l,A} \quad (9.45)$$

A number of references have determined that the ground-based component of the localizer error is not a simple random bias. Instead, it varies with the distance from the runway. A convenient model for this error source is to treat it as a spatial first-order Gauss Markov as shown in equation (9.46). By that is meant that the error does not vary with time but with the location of the receiver from the ILS localizer transmitter.

$$\frac{\partial}{\partial s} \Delta B_{l,G}(s) = -\beta_B(s) \Delta B_{l,G}(s) + n_B(s) \quad (9.46)$$

where,

- $n_B(s)$: Scaled Gaussian white noise
- $\beta_B(s)$: Spatial damping factor

$$ds = \left(\frac{ds}{dt} \right) dt = v_{IT}(t) dt \quad (9.47)$$

$$\beta_B(t) = v_{IT}(t)\beta_B(s) \quad (9.48)$$

$$\Delta\dot{B}_{I,G}(t) = -\beta_B(t)\Delta B_{I,G}(t) + n_B(t) \quad (9.49)$$

The actual ILS localizer beam bending errors for five different airports are illustrated in Figure 9.10. A set of five simulated ILS localizer beam bending errors using the above statistical model is presented in Figure 9.11.

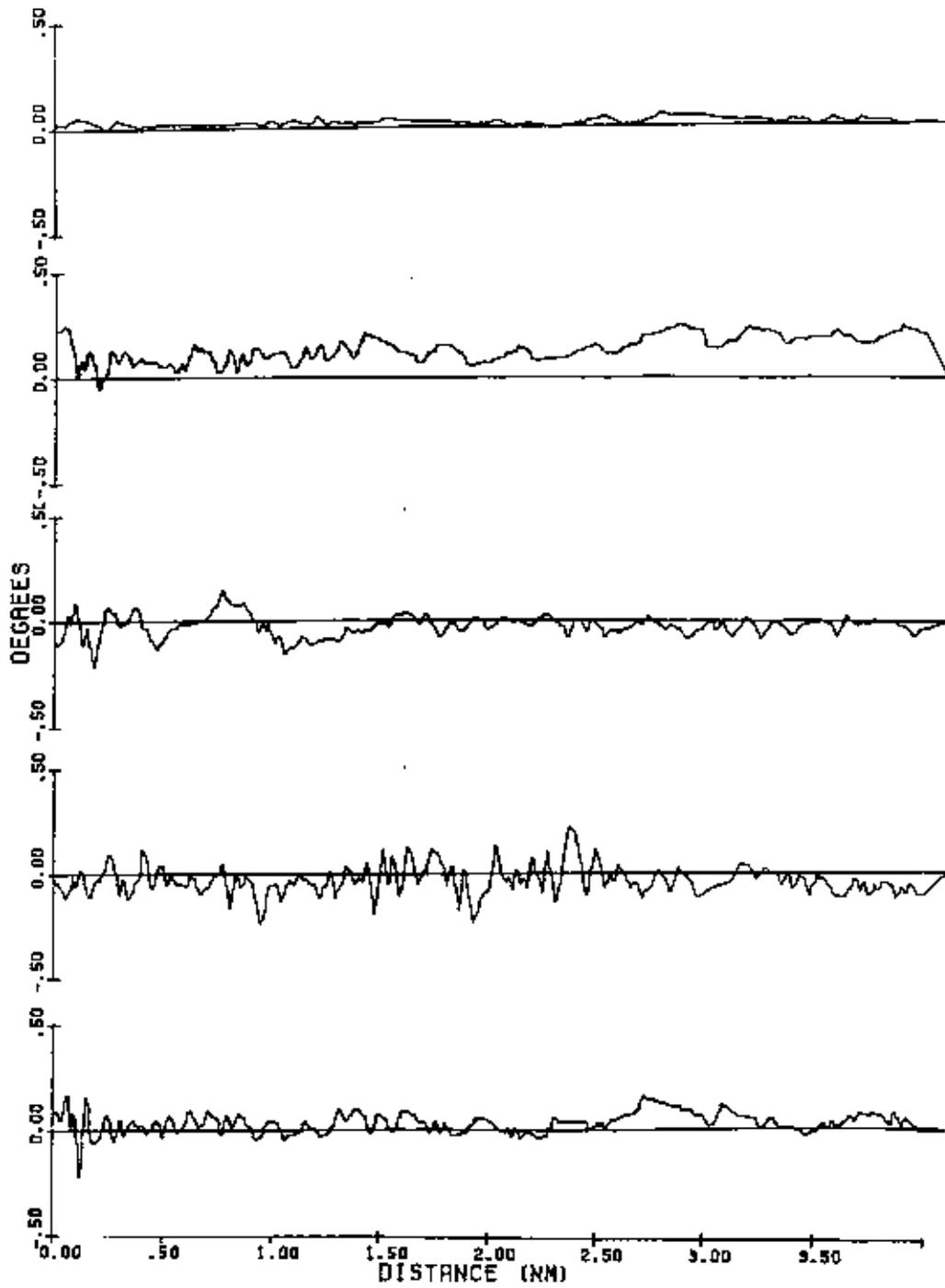


Figure 9.10. Measured ILS Localizer Bearing Deviation Angle Errors

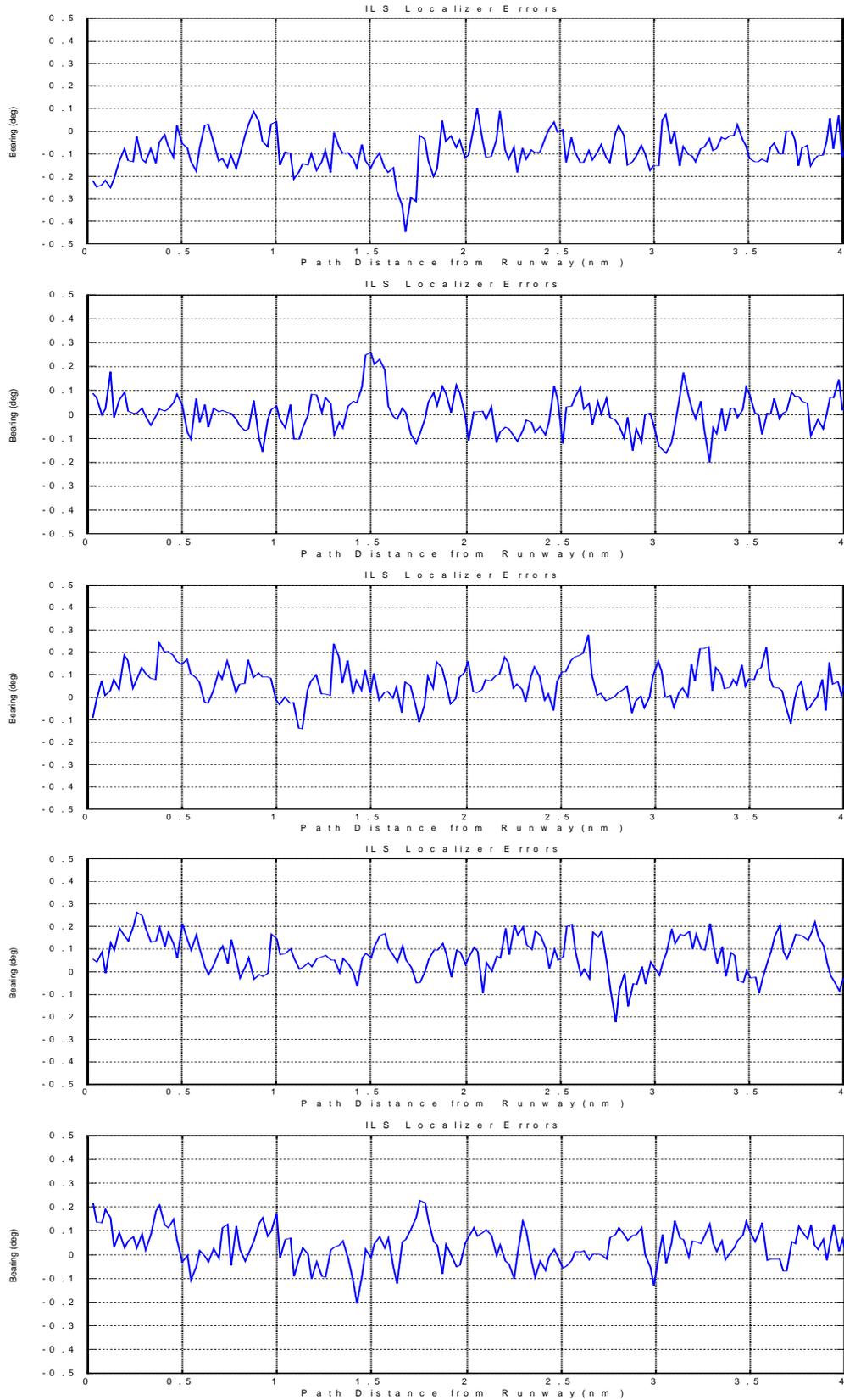


Figure 9.11. Simulated ILS Localizer Bearing Deviation Angle Errors

THIS PAGE INTENTIONALLY LEFT BLANK

10. The Longitudinal Guidance System

When an aircraft flies along a route it is often necessary to have the aircraft automatically meet speed and altitude constraints that are placed on fixes along the route. These are generally termed “crossing restrictions.” One case where crossing restrictions are often used is when modeling flight along a Standard Arrival Route (STAR). When the flight plan contains a STAR for the pilot to follow, it is assumed that the pilot has at least a textual description of the STAR and will make appropriate speed and altitude changes as published.

The TGF is currently developing its longitudinal guidance system. This chapter will be amended once the development is completed.

THIS PAGE INTENTIONALLY LEFT BLANK

11. Flight Technical Error

The flight technical error (FTE) is the inability or inexactness of the pilot or autopilot to steer the aircraft perfectly along the desired course. If the aircraft is steered by an autopilot, it is the error in steering the aircraft perfectly along the intended course. The waypoint and navigation aid errors are independent of the FTE.

Field data indicate that there is a random lateral FTE component that exists along the route segments. For the FMS-guided aircraft, the random en route wander was found to be 0.13 nm. (1σ) while for the non-FMS-guided (piloted) aircraft, it was found to be 0.7 nm (1σ), with a period varying from roughly 4 to 8 minutes during the en route flight segment [Hu96].

A reasonable model for this random lateral position wander, δr_{FTE} , is described by a second order Gauss Markov process:

$$\begin{pmatrix} \delta \dot{r}_{FTE} \\ \delta \dot{v}_{FTE} \end{pmatrix} = \begin{bmatrix} 0, & 1, \\ -\omega_0^2, & -2\beta \omega_0 \end{bmatrix} \begin{pmatrix} \delta r_{FTE} \\ \delta v_{FTE} \end{pmatrix} + c \begin{pmatrix} 0 \\ u_{FTE} \end{pmatrix} \quad (11.1)$$

The terms in the expression are defined as follows:

- δr_{FTE} : The lateral position error (nm)
- δv_{FTE} : The lateral position error velocity (nm/sec)
- c : The scale factor of the forcing function
- ω_0 : The natural frequency of the system
- β : The damping of the system
- u_{FTE} : The zero mean unity variance Gaussian white noise

For terminal flight segment during ILS localizer guidance, it was found that the lateral wander tended to increase linearly with the distance from the runway [T90]. This suggests that a FTE based on bearing deviation angle wander is more appropriate during the terminal flight phase. Therefore, a random wander of 0.24 degrees (1σ) with an approximate time constant of 90 seconds is appropriate. In the case of the ILS, the second order Gauss Markov process is written in terms of bearing deviation angle wander as shown in equation (11.2).

$$\begin{pmatrix} \delta \dot{B}_{ILS,FTE} \\ \delta \dot{\Omega}_{ILS,FTE} \end{pmatrix} = \begin{bmatrix} 0, & 1, \\ -\omega_0^2, & -2\beta \omega_0 \end{bmatrix} \begin{pmatrix} \delta B_{ILS,FTE} \\ \delta \Omega_{ILS,FTE} \end{pmatrix} + c \begin{pmatrix} 0 \\ u_{FTE} \end{pmatrix} \quad (11.2)$$

The new terms in equation (11.2) are as follows:

- $\delta B_{ILS,FTE}$: The bearing deviation angle (deg)
- $\delta \Omega_{ILS,FTE}$: The bearing deviation rate (deg/sec)

11.1 Operational Details

The flight technical error is quite simple to implement using the Gauss Markov processes with valid error parameters. The three types of flight technical error only operate when the aircraft is operating under the route following guidance system. This guidance algorithm must prompt the particular flight technical error model being used for an update to the lateral position deviation, δr_{FTE} .

11.1.1 Piloted Flight Technical Error

The piloted flight technical error proceeds once the Gauss Markov process has been initialized. For each time step that the piloted flight technical error is used, the Gauss Markov process is advanced one time step and a value for δr_{FTE} is returned. The piloted flight technical error uses a Gauss Markov process with the following parameters:

- $\beta = 0.50$: The damping term.
- $\sigma_p = 0.7$ nm: The standard deviation of the ‘position’
- $\sigma_v = 0.011944$ nm/sec: The standard deviation of the ‘velocity’
- $\Delta t = 0.5$: The time step of the process (sec)

11.1.2 FMS Flight Technical Error

The FMS flight technical error proceeds once the Gauss Markov process has been initialized. For each time step that the FMS flight technical error is to be used, the Gauss Markov process is advanced one time step and a value for δr_{FTE} is returned. The FMS flight technical error uses a Gauss Markov process with the following parameters:

- $\beta = 0.50$: The damping term.
- $\sigma_p = 0.13$ nm: The standard deviation of the ‘position’
- $\sigma_v = 1.444 \times 10^{-3}$ nm/sec: The standard deviation of the ‘velocity’
- $\Delta t = 0.5$: The time step of the process (sec)

11.1.3 ILS Flight Technical Error

ILS flight technical error is more complex because the Gauss Markov process is set up to return an angular deviation from the path rather than a linear distance. Therefore the linear distance must be calculated from the angular deviation $\delta B_{ILS,FTE}$ which is returned in degrees. To get the lateral offset, equation (11.3) is used where d_s is the distance to go to the localizer. Generally, the ILS is modeled as a two segment route, where the first capture segment is from some arbitrary initial approach fix to the final approach fix, and

the second segment is from the final approach fix (the beginning of the glide slope for ILS approaches) to the localizer. Therefore, the distance that the aircraft is to the localizer can be calculated by using the rhumb line distance to a fix algorithm.

$$\delta r_{ILS,FTE} = d_s \sin\left(\frac{\pi}{180} \delta B_{ILS,FTE}\right) \quad (11.3)$$

The ILS flight technical error uses a Gauss Markov process with the following parameters:

- $\beta = 0.50$: The damping term.
- $\sigma_p = 0.3^\circ$: The standard deviation of the 'position'
- $\sigma_v = 0.06 \text{ deg/sec}$: The standard deviation of the 'velocity'
- $\Delta t = 0.5$: The time step of the process (sec)

THIS PAGE INTENTIONALLY LEFT BLANK

12. Model Verification and Validation

Verification and validation of the algorithms used to develop the TGF simulation was accomplished primarily by using a small JAVA tool that served as a testing platform for the algorithmic development. This tool, which is named TGF-test, allowed for real time manipulation of aircraft trajectories on the screen and also monitored many of the aircraft's state variables on the screen in the form of stripcharts. All algorithms that are coded in the main TGF simulation were first tested and evaluated in the TGF-test simulation. The main screen of the TGF-test algorithm is shown in Figure 12.1 where aircraft trajectories are superimposed over an electronic map of fixes and routes. The aircraft icon, which represents the flying aircraft, shows the heading orientation so the difference between ground track and heading can also be viewed visually.

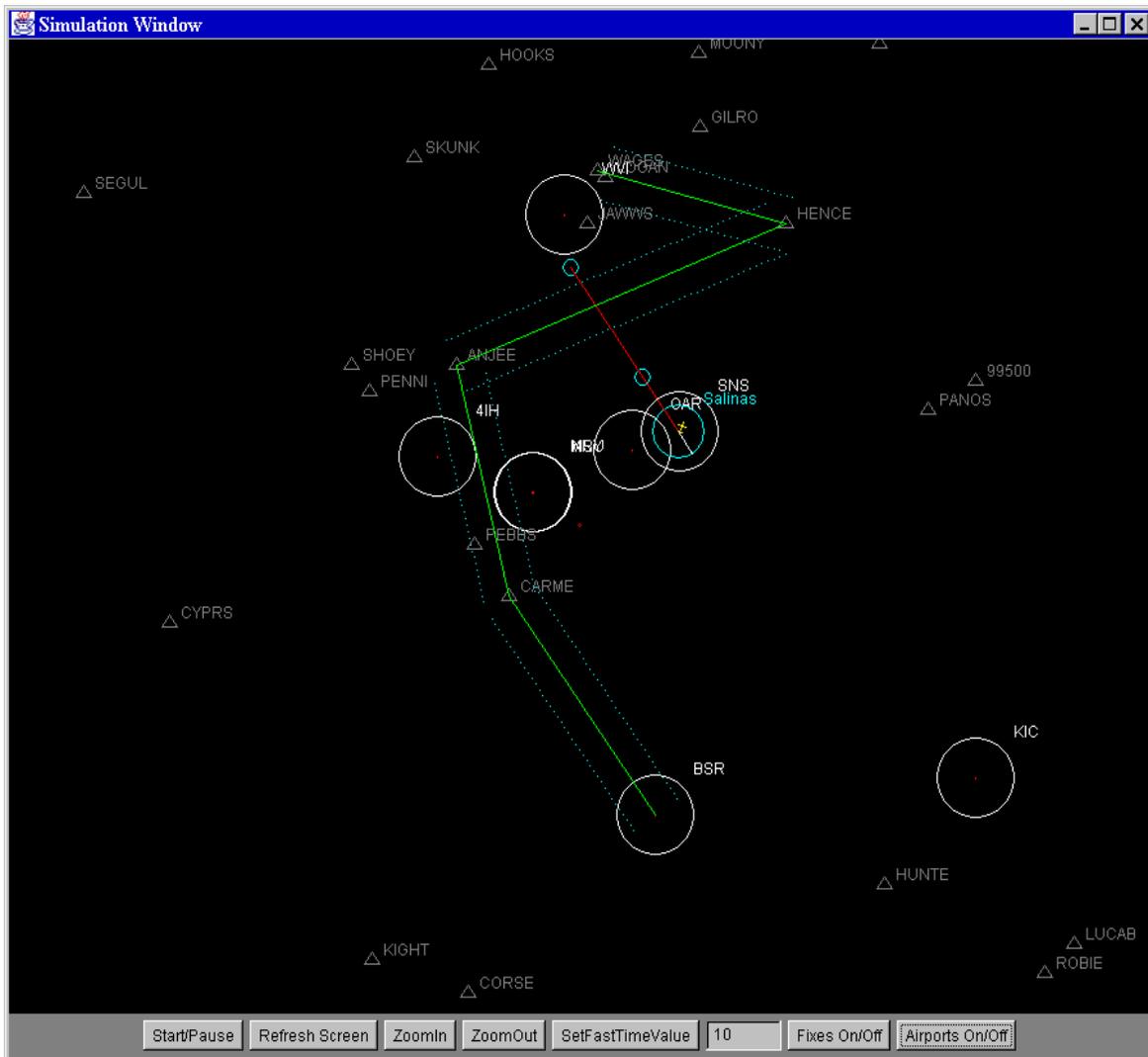


Figure 12.1. Simulation Window for TGF-test

The aspects of the TGF simulation that needed verification are as follows:

- Constant airspeed climbs and descents
- Mach/CAS descents and CAS/Mach climbs
- Speed changes during climbs and descents
- Automatic route capture
- Vectored route capture
- Initial fix route capture
- Segment transition
- Flight technical errors
- Navigation errors
- Take-off and landing

When appropriate, the TGF simulation model was compared to Pseudocontrol, the aircraft dynamics kernel of PAS. PAS, the NASA tool for trajectory generation, has been considered as an acceptable baseline for aircraft performance. Such cases include the verification of climb and descent performance as well as speed changes. For other operations, such as route capture and route following, visually inspecting the maneuvers is sufficient to insure proper operation.

12.1 Constant Airspeed Climbs and Descents

The PAS model in Pseudocontrol uses much higher fidelity aircraft and engine models than what the TGF model uses, so it is expected that there would be some variation in performance. Generally, however, the difference in the actual trajectories generated by the simulations is negligible. While the trajectories are nearly identical, the TGF model does not produce fuel burn estimates which are as accurate as the PAS model because PAS uses many more coefficients in the model. Two comparisons of Pseudocontrol and TGF-test are presented in this section.

The first comparison between Pseudocontrol and TGF-test is shown in Figure 12.2. Figure 12.2 illustrates an MD-80 at 10,000 ft and 280kts as it initiates a constant indicated airspeed climb to 30,000ft. Four stripcharts are shown in the plot, each representing a different aircraft state variable. These are Mach, indicated airspeed, altitude, and the lift coefficient. The Pseudocontrol plots are represented with the dark line and the TGF-test plots are shown in gray. The simulation shows a good match between the two models. Initially, there is a small fluctuation in the indicated airspeed of both models while the climb is established. Once the climb is established, both models hold the appropriate 280kt airspeed. The aircraft climb nearly identically in terms of altitude tracking. This is very important since the air traffic controllers are sensitive to the changing rate at which altitude increases. The Mach plot shows that both models track

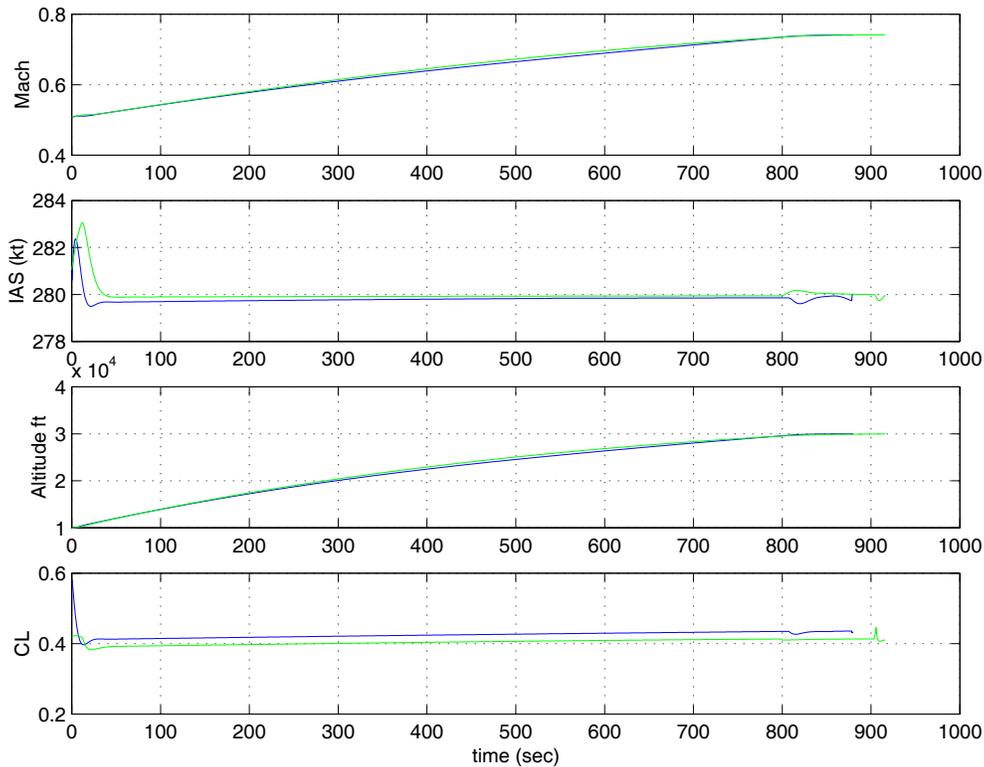


Figure 12.2. Comparison of Pseudocontrol (black) and TGF-test (gray) in a constant indicated airspeed climb and 280kt

Mach number identically as well. Considering the lower fidelity model represented in the TGF-test system, over Pseudocontrol, the data match is quite good.

A similar comparison is made in a descent. An MD80 weighing 130,000lbs is commanded to descend from 30,000ft to 10,000ft at an airspeed of 300kts. The descent is shown in Figure 12.3. When the aircraft initiate the descent, there is some fluctuation in the indicated airspeed. In this example, the TGF-test model has a larger fluctuation than the Pseudocontrol simulation, but the fluctuation is still only 1.5kts. This small fluctuation is acceptable. Once the descent is established, both aircraft hold the commanded airspeed well. The altitude profile of the TGF-test aircraft matches the Pseudocontrol aircraft well and the Mach number varies properly also.

12.2 Mach/CAS descents and CAS/Mach Climbs

The idle thrust Mach/CAS descent and full thrust CAS/Mach climb are important features of the aircraft simulation because jet airliners are most likely to use these types of maneuvers for climbs and descents. Because the maneuvers are similar, this section will consider as its only test case, the idle thrust descent. Full thrust CAS/Mach climbs are simply reverses of the descents.

When an idle descent is initiated, the throttle is pulled idle and the pilot descends at a rate so that the aircraft maintains the desired Mach of the desired Mach/CAS pair. At low speed and low altitude, convention dictates that the speed of aircraft be measured in terms of indicated airspeed. Therefore at some point during the descent, the pilot will capture the desired CAS of the Mach/CAS pair. As the pilot descends at a constant Mach, the indicated airspeed meter will show an ‘increase’ in speed. At some point during the descent, the indicated airspeed meter will read the desired indicated airspeed for the

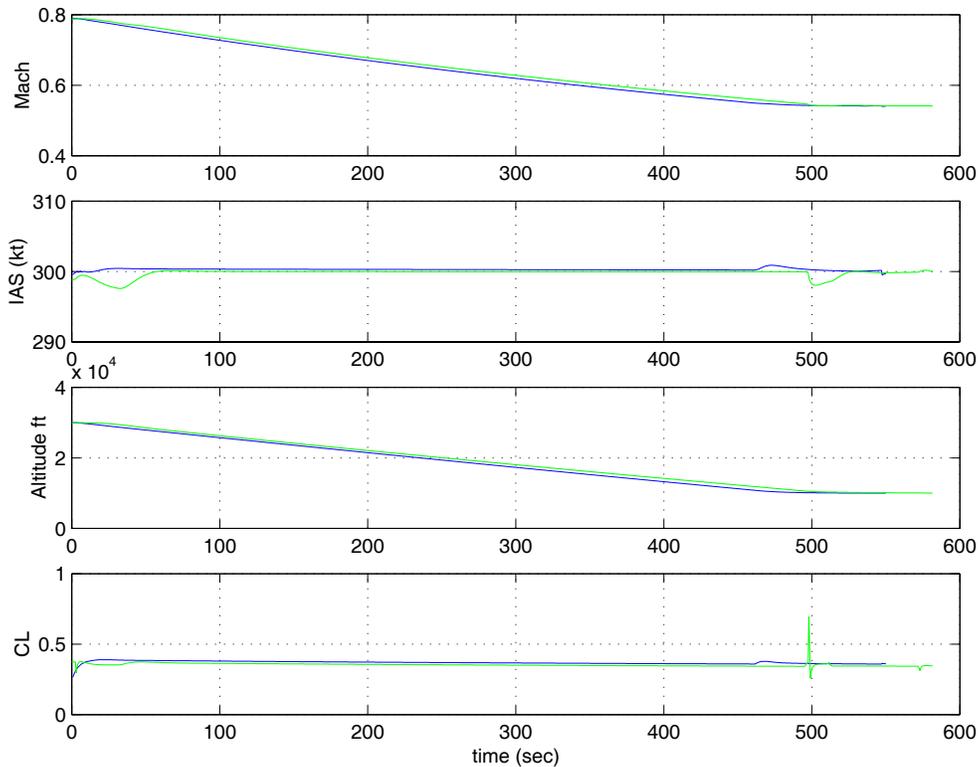


Figure 12.3. A comparison of Pseudocontrol (black) and TGF-test (gray) in a descent at a constant indicated airspeed of 300 kts

descent. At this point, the pilot tracks the desired CAS instead of the Mach. Typically there are four stages to a Mach/CAS descent. These stages are:

1. Change speed from the cruising Mach, M_1 , to the descent Mach, M_2 .
2. Descend at M_2 . The aircraft descends at M_2 until reaching a predetermined CAS.
3. Descend at constant CAS. The aircraft descends at its constant descent CAS until it reaches the metering fix crossing altitude, where it levels off.
4. Decelerate to the metering fix crossing speed. Finally, the aircraft decelerates to 250 kt, the metering fix crossing speed of typically 250kt.

For this section's comparison between Pseudocontrol and TGF-test, we start with an MD80 at 30,000ft in cruise at M0.76. The aircraft initiates a Mach/CAS descent with the following speeds: (M0.76/320kt). The aircraft then levels out at 10,000 ft maintaining 320kts. Figure 12.4 shows the maneuver. The match between the Mach and indicated airspeeds is good and the transition between Mach and indicated airspeed is smooth without any undesirable transients. Similarly, the level off at 10,000 ft is smooth without any overshoot. Most importantly, the altitude profiles for both simulations match very well.

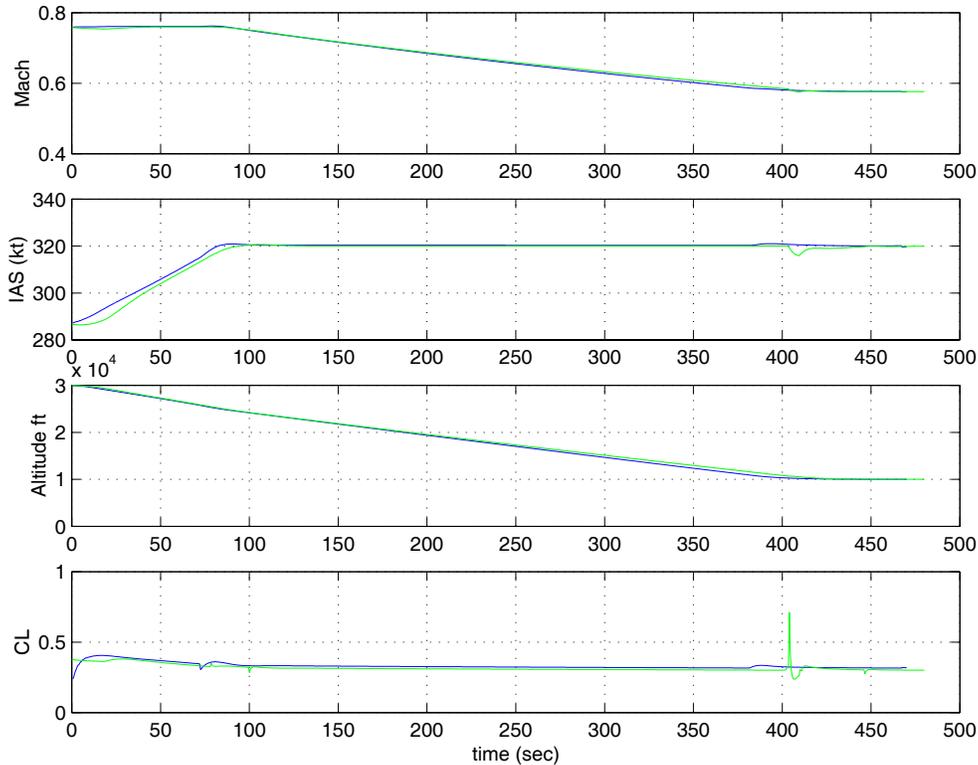


Figure 12.4. Comparison of Pseudocontrol (black) and TGF-test (gray) performing a Mach/CAS descent from 30,000 ft to 10,000ft using an MD80 at 130,000lb

12.3 Speed Changes

Speed changes, while they do not take much time in the course of a flight, do give a good indication of the model's fidelity in terms of drag and thrust. Consider an acceleration. For the aircraft to have the proper acceleration, in a speed up maneuver, the excess thrust must be correct. This excess thrust is a function of the total available thrust and the total drag. If either is off, the acceleration will not be right. However, errors in either could cancel each other out. For instance, a high drag number could be canceled by a high thrust value. Decelerations, because they are performed at idle thrust, tend to remove the thrust from the system so the primary deceleration factor is the aircraft's drag. If

accelerations and decelerations are both studied, generally conclusions about both the drag and thrust can be made. The deceleration gives insight into the fidelity of the drag

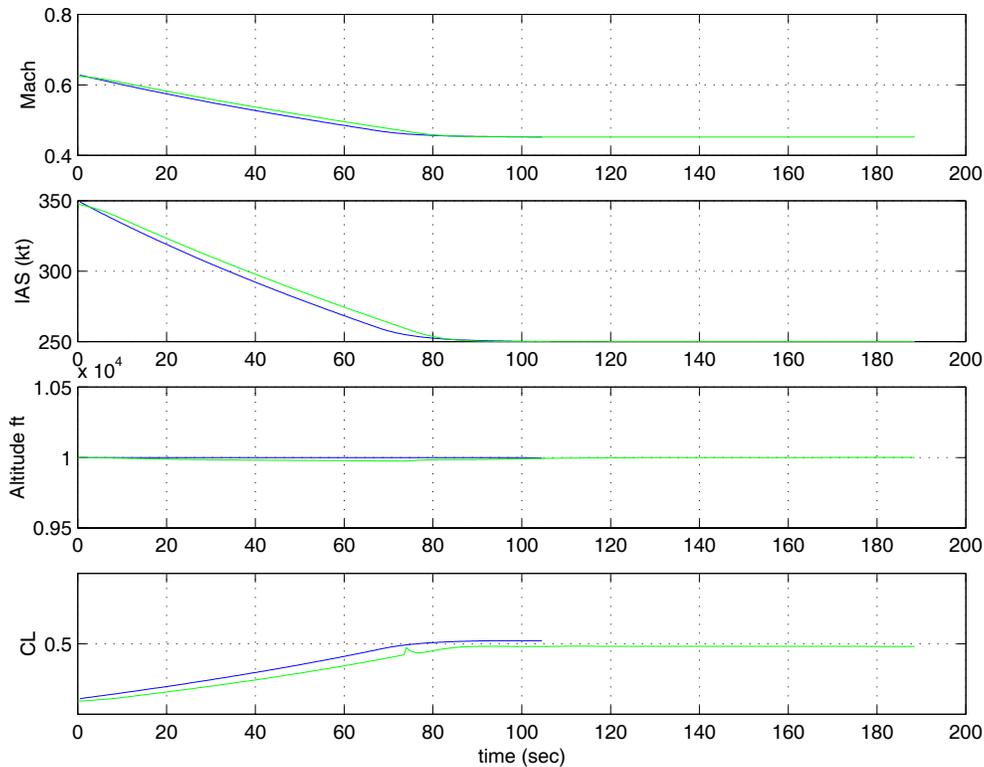


Figure 12.5. A deceleration of an MD80 from 350kts to 250kts while at 10,000ft using Pseudocontrol (black) and TGF-test (gray) simulation tools

model, and the acceleration gives insight into the thrust to drag ratio. Assuming drag information is known from the deceleration, thrust information can be derived from the acceleration.

First, a deceleration is considered. An MD80 at 130,000lbs cruising at 10,000 ft is slowed down from 350kts to 250kts. Figure 12.5 shows the deceleration maneuver. The two simulations show good agreement during the slowdown with no undesirable transients in either simulation. Furthermore, altitude is held constant at 10,000ft. From this plot we can assume that the drag information in the MD80 model is accurate.

Next, an acceleration is considered. The same MD80 is accelerated from a cruise condition of 250kts and 10,000ft to 350kts while maintaining altitude. The maneuver is shown in Figure 12.6. The slope of the speed curves for both simulations match very well, suggesting that the thrust model for the aircraft is working well. The altitude is held reasonably well; however, the TGF-test model does show a slight tendency to let the

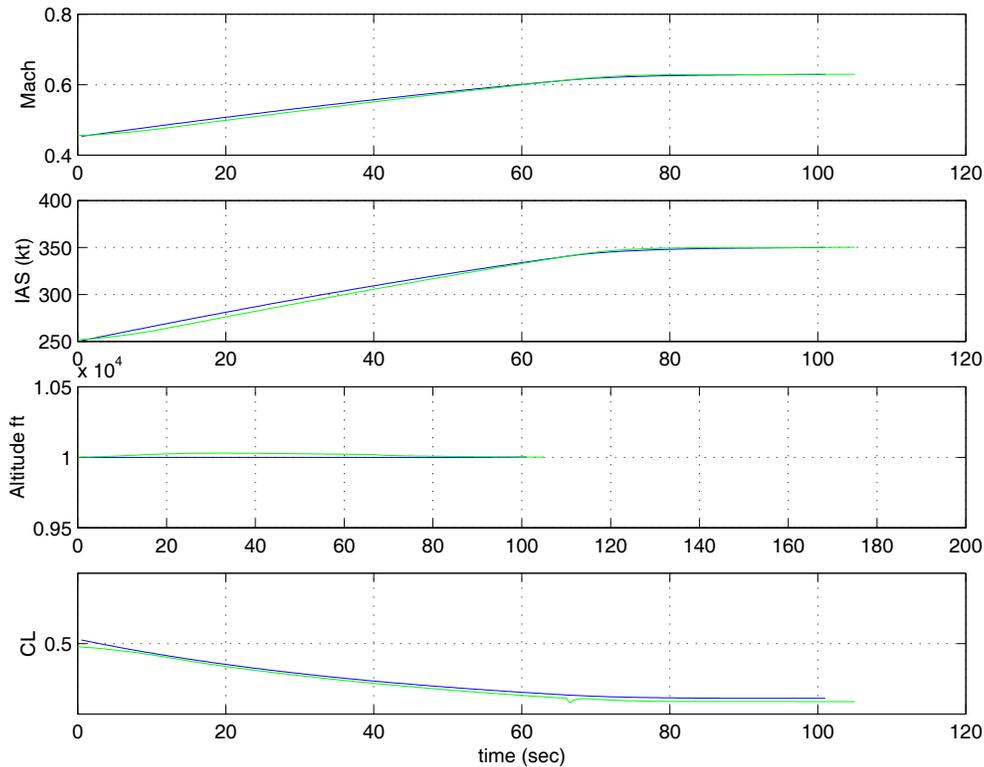


Figure 12.6. An MD80 accelerating from 250kts to 350kts while maintaining 10,000ft using Pseudocontrol (black) and TGF-test (gray) simulation tools

altitude drift slightly. The altitude variance shown on the plot is on the order of 10ft, so it is not a major concern. From the results in Figure 12.5 and Figure 12.6 we conclude that the thrust and drag models for this aircraft are good.

Speed changes are also performed at higher speeds to test the compressibility drag model. First a deceleration is considered as shown in Figure 12.7. An MD80 at 30,000ft and Mach 0.8 is decelerated to Mach 0.6. There is a slight discrepancy here between the Mach numbers of the two models. The TGF-test simulation takes longer to enter the deceleration whereas the Pseudocontrol model enters the deceleration immediately. One reason for this gentle initiation of the maneuver is that the TGF-test simulation models engine spooling whereas the Pseudocontrol model does not. Once the deceleration is established, notice that the two Mach lines are nearly parallel. This suggests that the rate of acceleration is very close but the slow initiation time on the part of TGF-test offsets the maneuver. The result is that the maneuver takes 7-10 seconds longer with TGF-test than it does with Pseudocontrol. Since the rate of acceleration is nearly the same, we know that the drag models are very close. The difference in initiation is explained by the differences in the spooling lags of the engines and some differences in control system design. If the longer deceleration is a problem, the effectiveness of the spooling lags could be decreased. Presently this is not a concern.

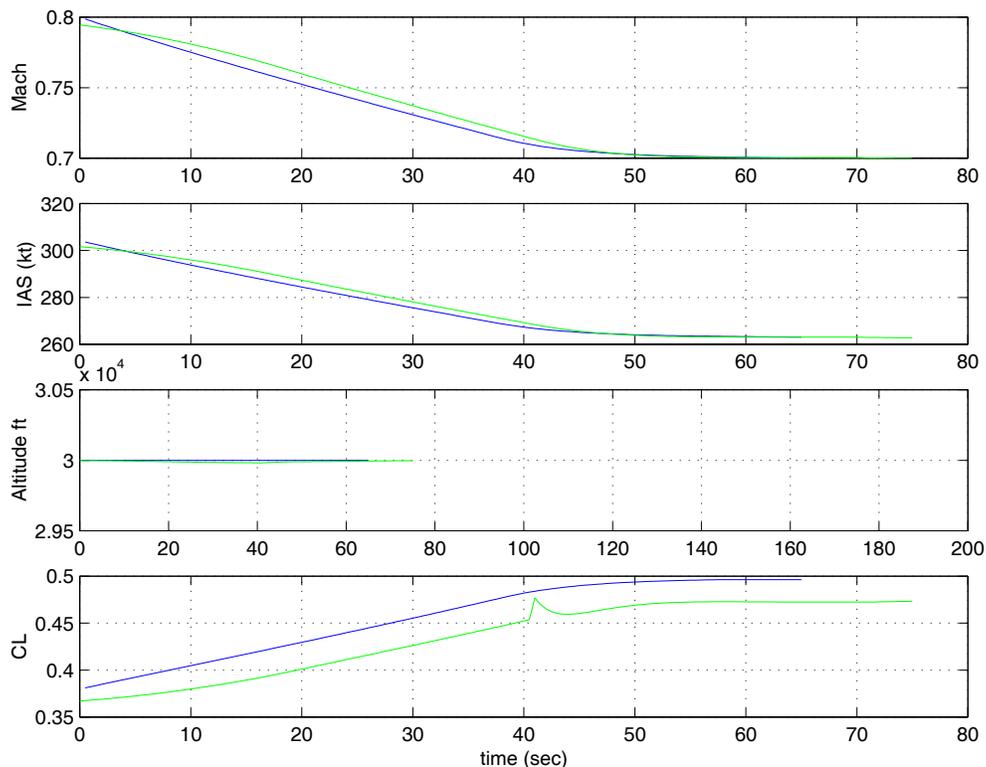


Figure 12.7. An MD80 decelerating from Mach 0.8 to Mach 0.6 while maintaining 30,000ft using Pseudocontrol (black) and TGF-test (gray) simulation tools

The Mach acceleration is shown in Figure 12.8, where the MD80 accelerates from Mach 0.6 to 0.8 while maintaining 25000 ft of altitude. Here we do not see the same spooling lags in the initiation of the maneuver. Since the spooling lags are the same for both increases and decreases in thrust level in the TGF-test model, there is no easy explanation for the discrepancy. At any rate, the two models are virtually identical in the Mach acceleration.

12.4 Speed Changes during Climbs and Descents

One of the more insidious problems encountered during the design of the longitudinal control system was the problem of changing speeds during climbs and descents. Section 4.2 of the longitudinal control system discusses the problem in depth and explains how the ultimate solution to the problem was the ramping of inputs. Generally, the problem centered around the fact that large speed changes while climbing or descending was not anticipated, and the feedback control system was only set up to handle small changes. This meant that the control system had rather high gains to keep the speed errors small. When the high gains were applied to the large errors, the maneuvers became

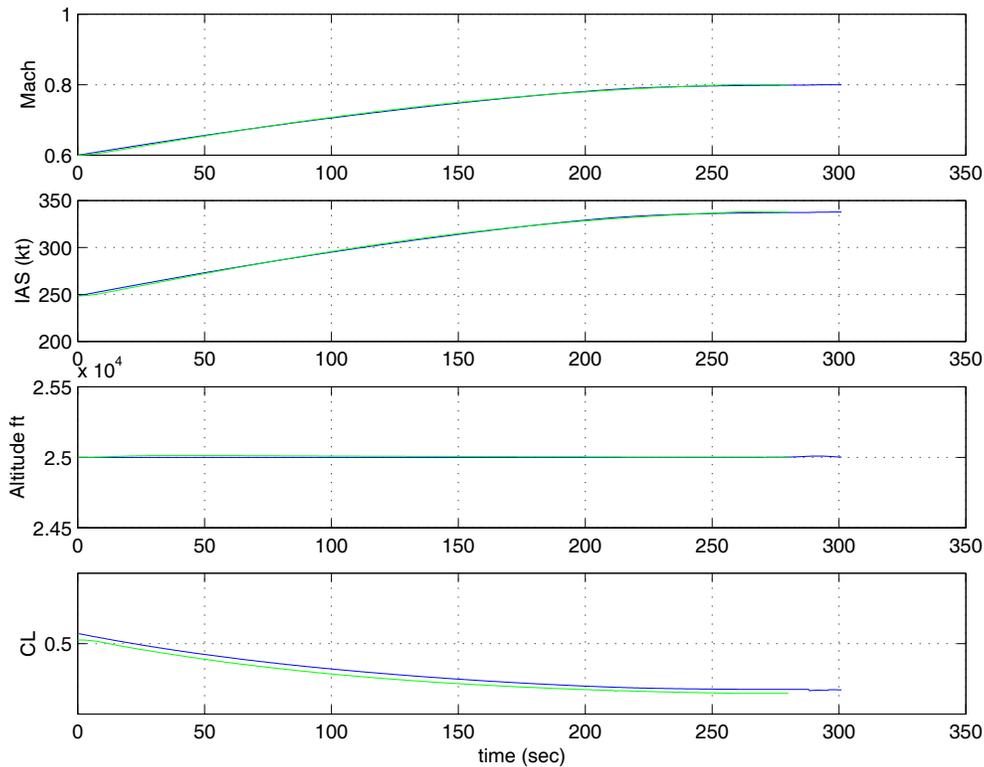


Figure 12.8. An MD80 accelerating from Mach 0.6 to Mach 0.8 while maintaining 25,000ft using Pseudocontrol (black) and TGF-test (gray) simulation tools

violent. It was easy to see what was happening. During climb and descent, the control stick rather than the throttle is used to control speed. Therefore, when the feedback control system saw large errors from a user, the control stick was moved violently to correct the situation. This section uses TGF-test to demonstrate the speed changes during climbs to illustrate the stability of the maneuvers using the ramped inputs.

Figure 12.9 illustrates an MD80 which is initially at 10,000ft and 250kts. A climb is initiated to 30,000ft. During the climb, the speed of the aircraft is first increased to 320kts and then reduced to 280kts. Finally the speed is increased to 300kts where it is held until the aircraft completes the climb. The point of the plot is that speed changes are made gracefully and do not cause any unusual patterns in the altitude profile other than the normal affects associated with the change of speed during a climb. Furthermore, attention to the lift coefficient curve shows that the control system is well behaved and is not commanding unreasonable control inputs. There also isn't any chatter in the system.

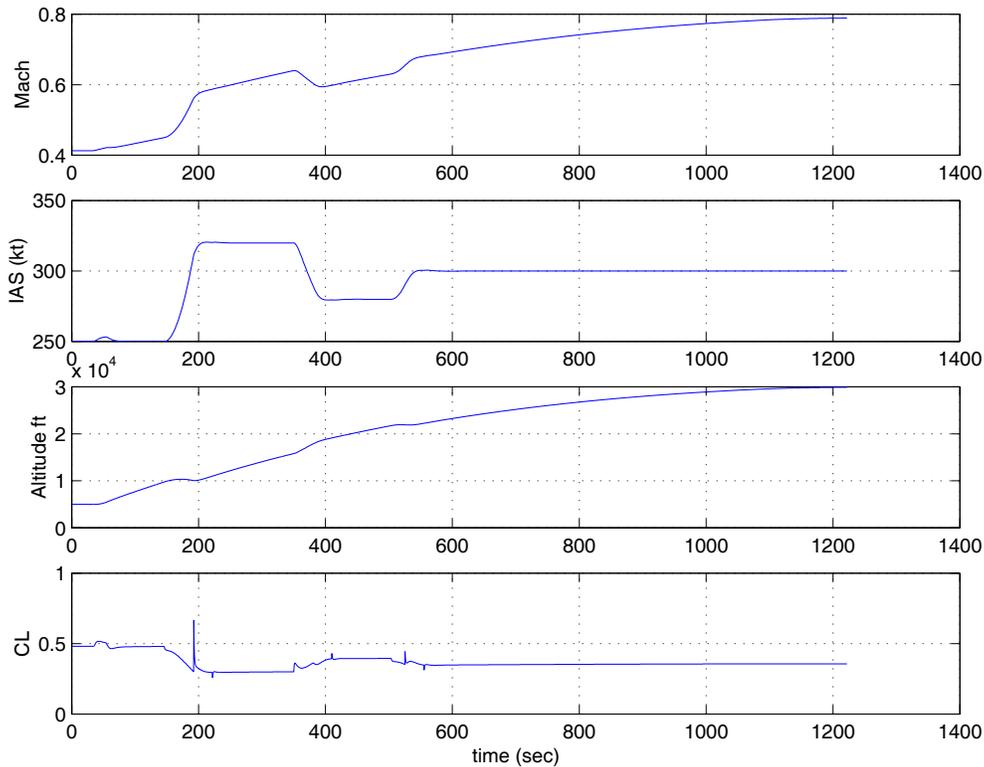


Figure 12.9. An MD80 in a climb with various speed changes using the TGF-test simulation

This is a direct result of the ramped inputs preventing large errors from occurring in Regions 3 and 6.

12.5 Automatic Route Capture

The TGF-test algorithm uses several route capture algorithms to capture routes. The first of these is the automatic route capture algorithm which captures a route regardless of the aircraft's location with respect to the route or its orientation. To verify this always works, many test cases were run. Generally, the key aspects which determine a good capture are whether or not the aircraft chooses a reasonable segment, and whether or not overshoot is excessive. Some of the more difficult test cases are presented in this section.

The first capture, shown in Figure 12.10, illustrates the case where the aircraft is very close to a segment along a route but headed in the wrong direction. When the route capture algorithm is initiated, the automatic route capture algorithm rightly chooses the segment which is closest to the aircraft. Then the algorithm calculates a dynamic fix for intercept along the route. However, because the aircraft is already so close to the segment, the aircraft penetrates the 1 turn radii boundary defining the route following algorithm

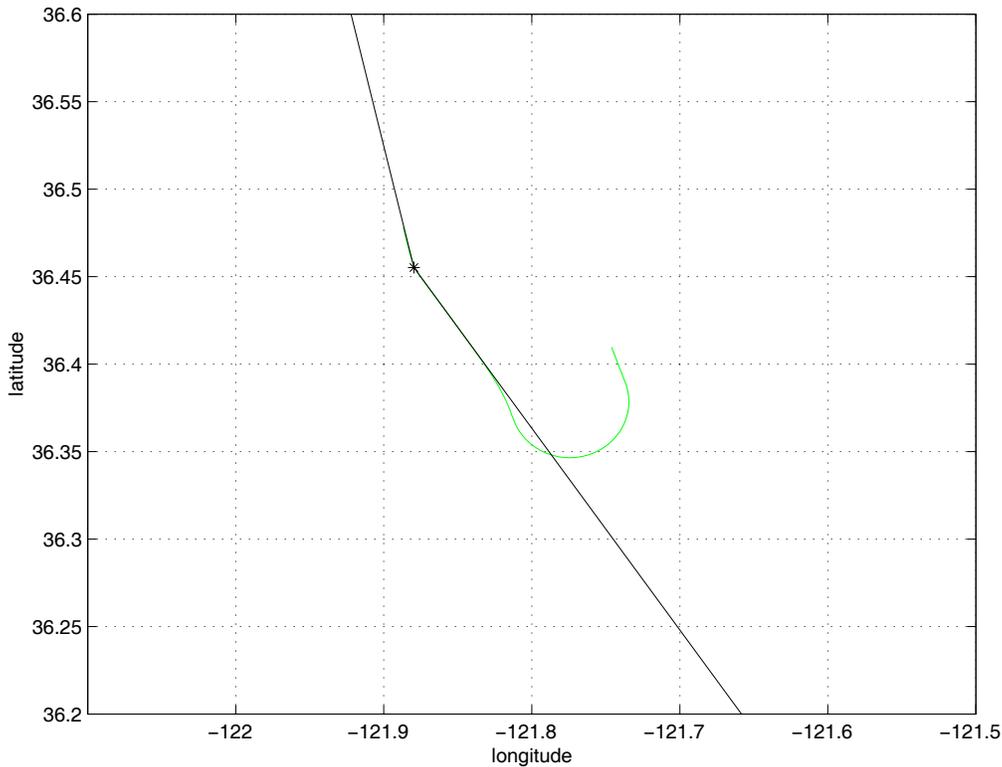


Figure 12.10. Automatic route capture with the aircraft close to the route but headed in the wrong direction

long before the aircraft is able to converge on a heading to the dynamic fix. Therefore, this particular scenario is a test of the route following algorithm more than it is a test of the automatic route capture algorithm. It demonstrates that the route following algorithm can handle large discrepancies in aircraft heading relative to the bearing of the segment. The aircraft makes a right turn towards the segment and then slightly overshoots the segment. This overshoot is unavoidable because the aircraft is well within 2 turn radii of the segment before the maneuver is initiated. The performance shown in Figure 12.10 is exactly what is desired.

The next route capture maneuver, shown in Figure 12.11, demonstrates the case where the aircraft is at a considerable distance from the capture segment. This route capture demonstrates the use of the dynamic fix. As stated in Section 8.4.3.1, the dynamic fix is an imaginary fix which is created by the system at some location along a segment and is used as a point of reference for capture. When the automatic route capture algorithm was first conceived, it seemed as though the most obvious method of capturing the route was to fly some intercept heading to the route. For instance, once the capture segment was determined, the aircraft could be given an intercept heading of 45 degrees and intercept the segment. However, this method seemed to have some inherent limitations. First, the aircraft would always intercept using 45

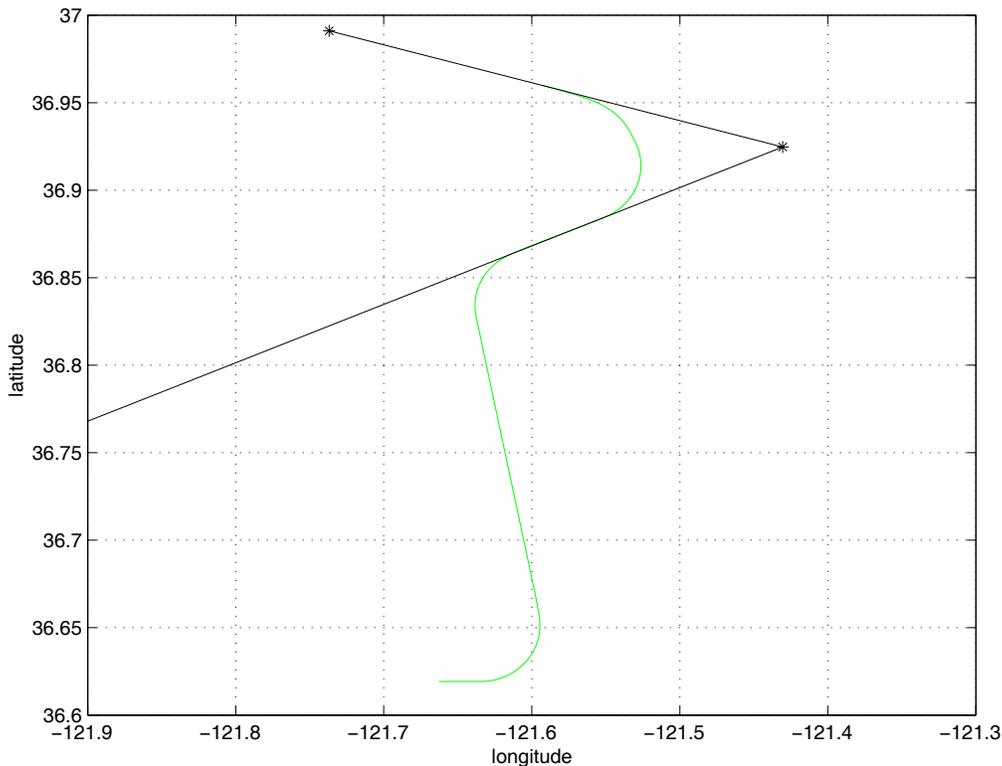


Figure 12.11. Automatic route capture with an aircraft far from the capture segment

degrees regardless of how far the aircraft was away from the segment. An aircraft far away from the capture segment might pass the segment before ever capturing it. Figure 12.11 illustrates an example of this type of capture situation.

To avoid the problem of aircraft overshooting capture segments, a dynamic fix is placed on the segment to be captured, and the aircraft is commanded to fly toward the dynamic fix. In this case the further aircraft naturally uses a larger intercept angle as seen in Figure 12.11. This system insures that the proper segment is captured and also provides some apparent variety in intercept angles so that all aircraft do not appear to behave the same. The capture algorithm does exactly what is expected. The automatic route capture guidance converges on a heading that leads directly to the dynamic fix, and then captures the route when the aircraft is within 1 turn radii of the segment. There is no overshoot because the aircraft is given enough space to maneuver in this example.

The track in Figure 12.11 also demonstrates one other important performance trait of the algorithm. The algorithm switches segments and captures the next segment along the route without overshoot in spite of the acute angle which joins the segments. This example demonstrates the effectiveness of the segment transition algorithms at assuring

smooth segment transition regardless of segment geometry. The case shown in Figure 12.11 is a more difficult segment transition maneuver.

Figure 12.12 is the least sensational of the capture examples in that it demonstrates a rather easy and likely scenario. The aircraft is relatively close to the capture segment and heading in the general direction of the segment. Therefore, the aircraft captures the segment with ease. This example does show how the dynamic fix does make the capture angles vary as a function of the distance that the aircraft is from the segment. Being relatively close in this example, the aircraft takes a smaller intercept angle which is apparent in the difference between the original heading and the intercept heading. Note that in this case, the original heading would have intercepted the segment as well. When the aircraft is well within 1 turn radii, the aircraft turns to intercept the segment without any overshoot. This is exactly what is desired.

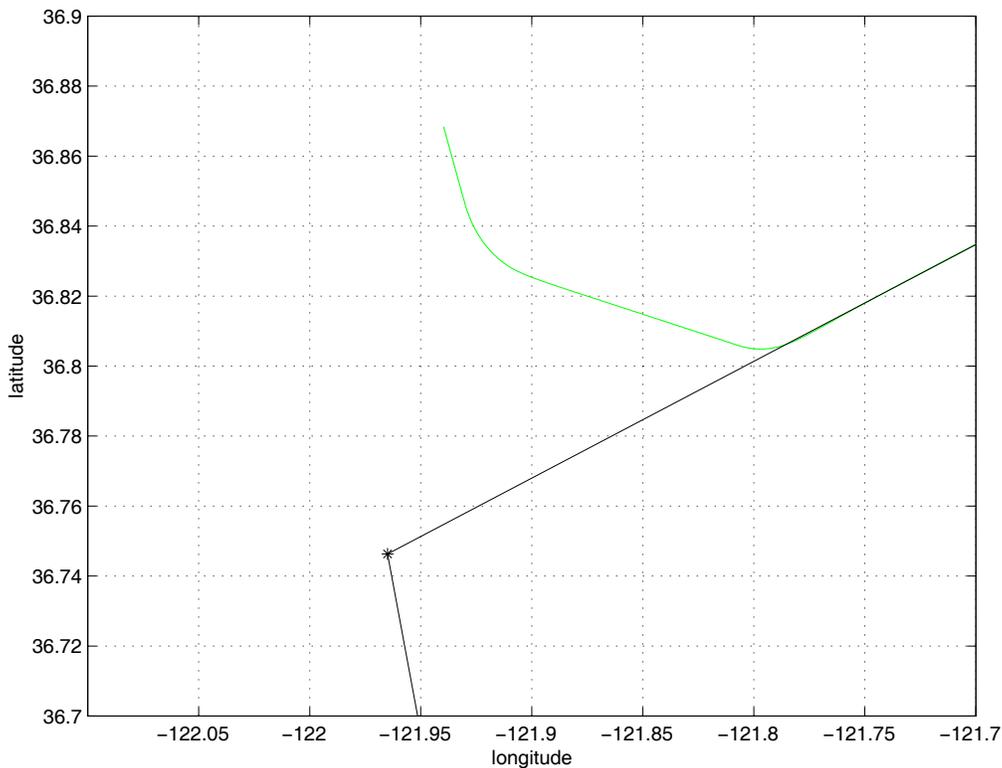


Figure 12.12. Automatic route capture with the aircraft headed perpendicular to route

The final automatic route capture example, shown in Figure 12.13, demonstrates the capture of a route from an ambiguous area relative to the route. As discussed in Section 8.4.1, there is a “dead” region where two segments meet as shown in Figure 12.13. If an aircraft is in this region, the normal segment determination algorithms will find that the aircraft is in front of the trailing segment, behind the leading segment, and will not return a capture segment. In this case, another criteria is used which checks the aircraft’s

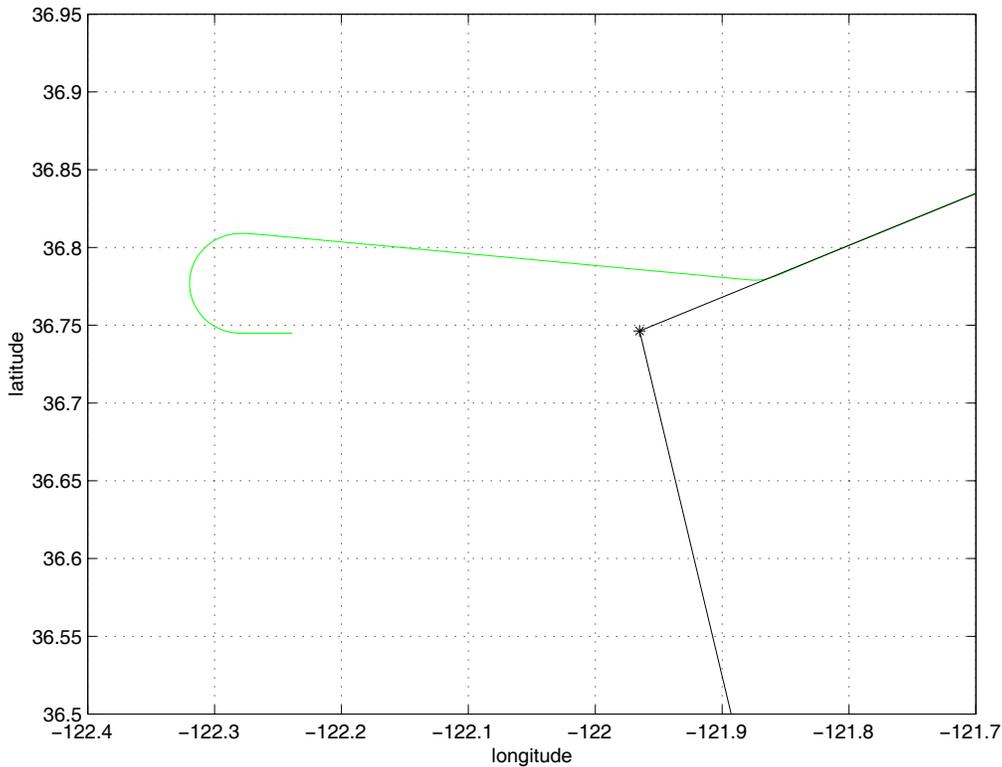


Figure 12.13. Automatic route capture with aircraft in an ambiguous region between segments

distance from every segment’s trailing fix. It then chooses to capture the segment that is associated with the closest trailing fix. In Figure 12.13, the latter logic is used to determine the appropriate capture segment. Once the appropriate segment is determined, the aircraft is flown towards a dynamic fix and ultimately captures the segment with no overshoot.

12.6 Vectored Route Capture

The vectored route capture algorithm steers the aircraft along a user specified heading until the aircraft intercepts the route. Each time step, the algorithm determines which segment is best to capture and, each time step, the algorithm determines if it is time to merge onto the route. It should be noted that the algorithm has no control over the initial heading. Therefore, if the user supplied heading steers the aircraft away from the route, the guidance law is not able to do anything about it although it will provide a warning if the aircraft is unlikely to intercept the route. It is important that this algorithm work properly because it is the most heavily used capture algorithm.

Because the vectored route capture algorithm does not choose the intercept heading, it must be prepared to work with all intercept headings, including those which may be poor

choices. To test the vectored route capture algorithm several poor choices are provided. The first scenario, shown in Figure 12.14, demonstrates a vectored heading where the choice of capture segments is ambiguous. Because the algorithm can not control the heading of the aircraft, it is possible that the best capture segment could change depending on the aircraft's flight path. In our scenario, the aircraft is initially pointed so that the flight path will intercept the front end of the trailing segment; however, the best segment for capture is actually the next segment along the route. The aircraft realizes this and turns onto that segment without overshoot when the segment proximity permits.

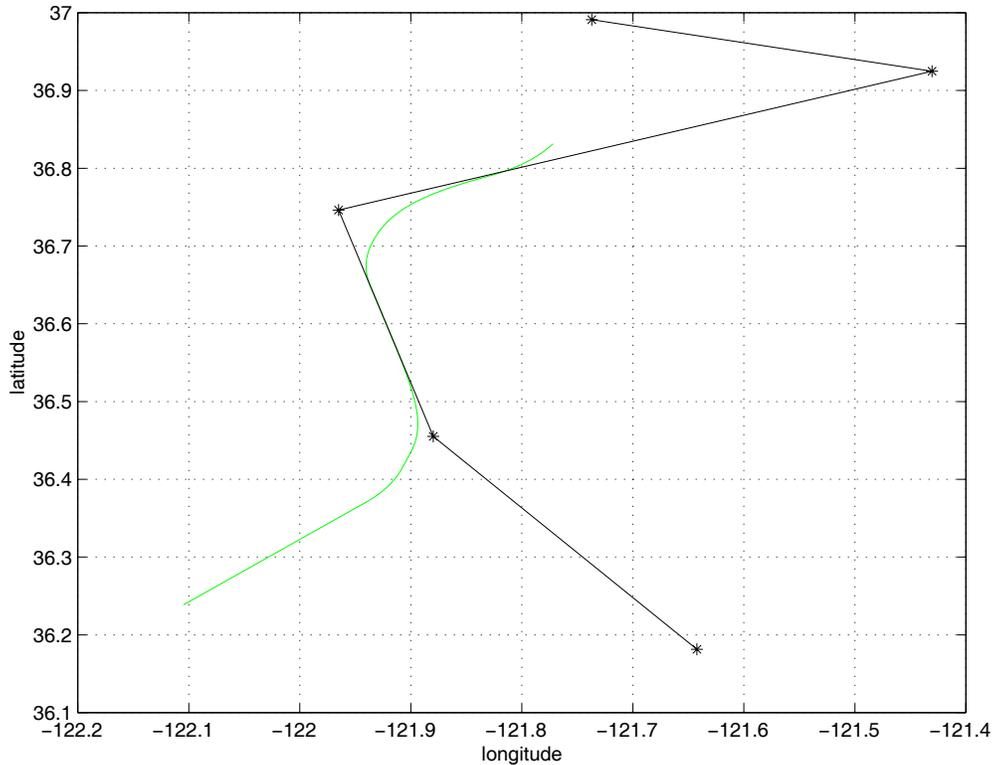


Figure 12.14. Vectored route capture from an ambiguous position

Figure 12.14 also shows the aircraft transitioning to a new segment while on the route. This maneuver provides another example of the segment transition logic at work where the transition between segments occurs without any overshoot. This is good performance. Notice that the aircraft track is stopped just as the aircraft starts to make the turn onto the final segment at the route.

The final example of vectored route capture is shown in Figure 12.15. In this example, the aircraft is commanded to capture a route while being vectored on a heading which tends to be in the opposite direction of the route. This type of capture is particularly challenging because the aircraft must make such a large turn to capture the segment. As can be seen in Figure 12.15, the aircraft maintains the vectored heading until it is time

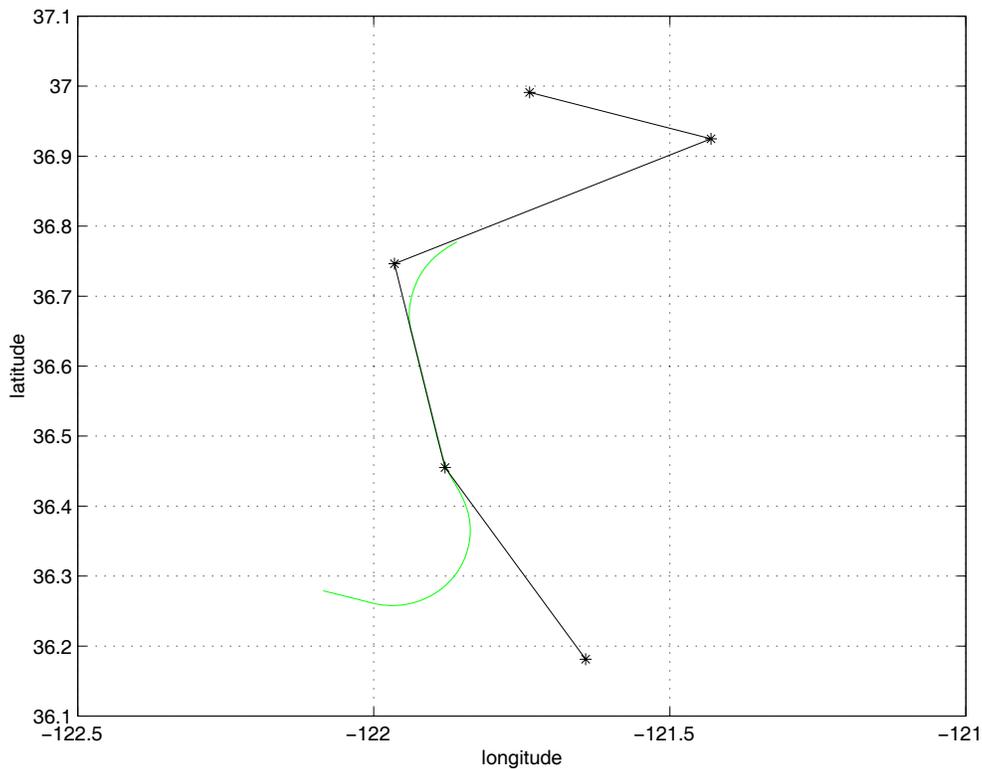


Figure 12.15. Vectored route capture when the vectored heading tends to be in the opposite direction of the route

to merge onto the route. Once the appropriate distance is reached, the aircraft merges with the route with no overshoot.

12.7 Initial Fix Route Capture

One requirement for the simulation was that the aircraft had to be able to start flying along a route by passing through the initial fix. This type of route capture algorithm is actually the least complicated because there is no need to determine a capture segment and there is no need to determine when to initiate route following. The route following is initiated as soon as the aircraft passes through the initial fix. Several examples of this maneuver are shown.

The first example, shown in Figure 12.16, illustrates the unlikely case where the aircraft is commanded to fly back to the beginning of the route from some location in front of the route. This maneuver turned out to be very handy for testing, but it is not likely to be used much in practice. It is interesting to see how this algorithm differs from the other capture algorithms. The algorithm directs the aircraft to fly to the initial fix. However, unlike the automatic route capture algorithm, the initial fix capture flies the aircraft through the initial fix before capturing the route. This method guarantees overshoot when used in this

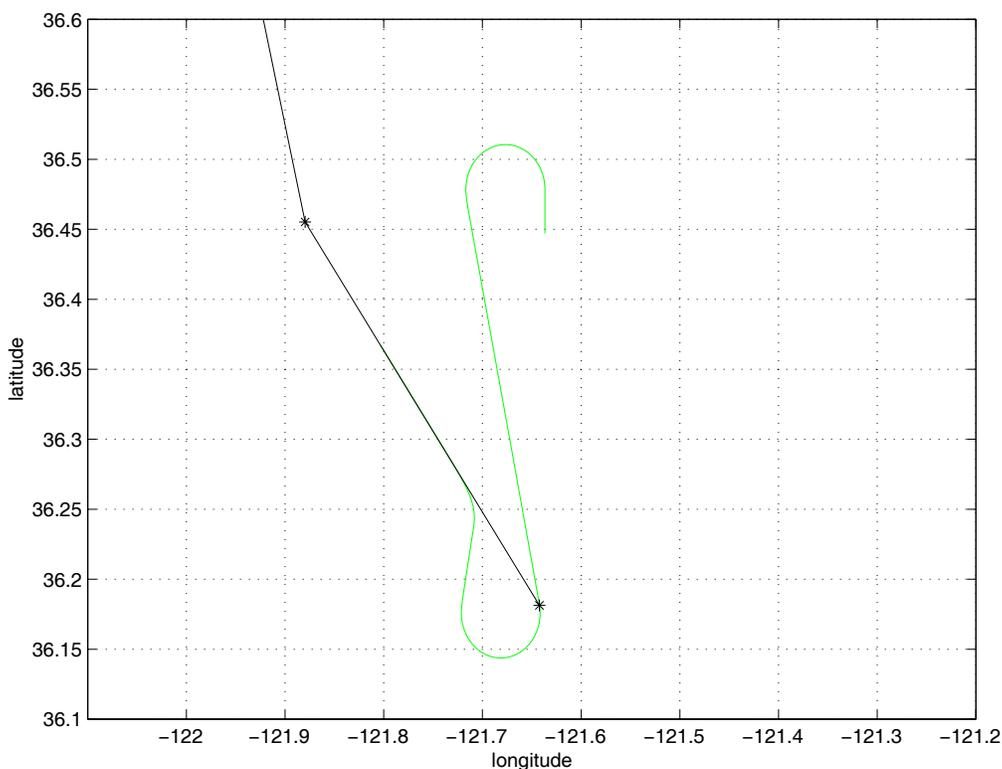


Figure 12.16. The initial fix capture algorithm being used to vector an aircraft back to the beginning of the route

fashion. Naturally, this algorithm is really designed to assure that an aircraft behind a route captures the route by flying through the first fix. This case is illustrated in Figure 12.17. Notice that when the algorithm is used as it is intended, the overshoot is minimized. However, there will usually be some overshoot because the algorithm is constrained to fly through the initial fix.

12.8 Segment Transition

The segment transition algorithm controls when the aircraft initiates a turn from one segment to another. The algorithm must initiate a turn with enough space to smoothly make the transition. Because so many examples of segment transition are contained in the plots dealing with route capture, no additional plots are presented here. The reader should observe the transitions in Figure 12.10, Figure 12.11, Figure 12.14, and Figure 12.15. Figure 12.10 shows a typical segment transition where the difference in bearing angle between segments is small. The aircraft manages these without difficulty. The other examples all deal with transitions between segments with large bearing differences. These are more complex because the amount of space allotted for the transition is

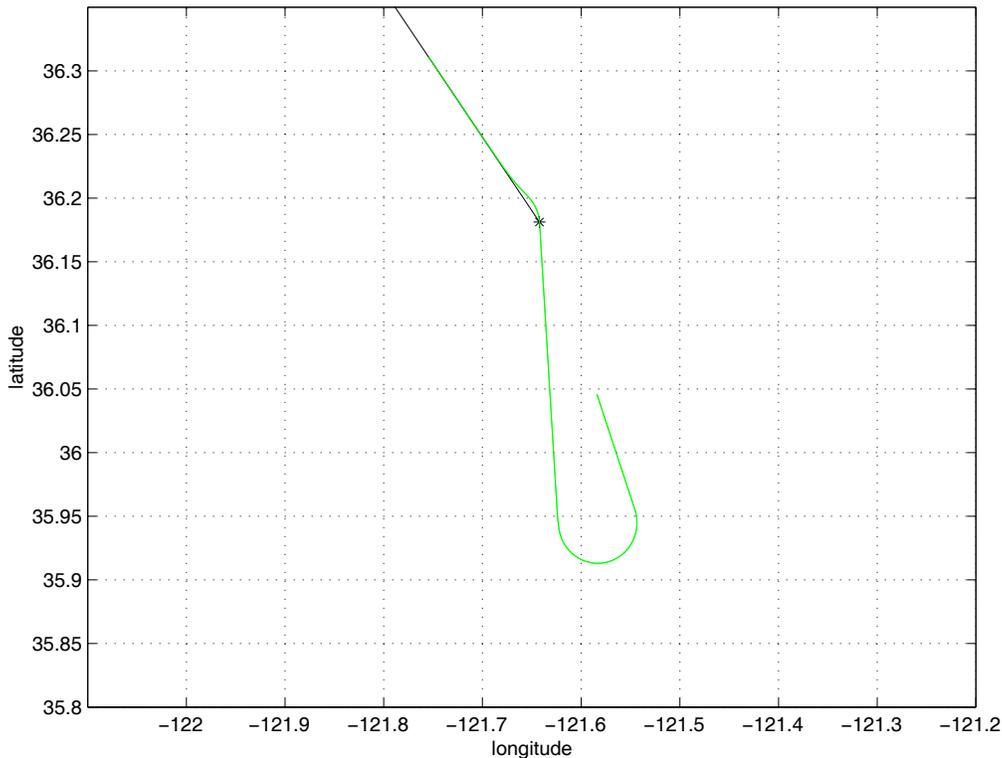


Figure 12.17. An aircraft capturing a route from behind using the initial fix capture algorithm

more constrained. Excessive overshoot occurs if the transition logic isn't designed to handle these cases. As can be seen from the plots, the current segment transition logic has no difficulty with these types of maneuvers.

12.9 Flight Technical Error

The flight technical error (FTE) is the inability or the inattention causing the pilot to steer the aircraft perfectly along the desired course. If the aircraft is steered by an autopilot, it is the error in steering the aircraft perfectly along the intended course. In terms of the previous navigation error sources, the FTE is considered to be the guidance and control error, where only the guidance error is included. The waypoint and navigation aid errors are not included. For the TGF simulation, there are two distinct flight technical errors modeled. The first flight technical error is the piloted flight technical error, the error associated with a human pilot following a route. The second error is the FMS error, the error associated with an FMS driven autopilot guiding the aircraft. Generally, the only difference between these errors is the magnitude of the standard deviation and the frequency of the mode. The statistics summary is reprinted in Table 12.1

Table 12.1 Error Statistics Summary

Error Source	Type	Bias (1σ)	Beta
FMS Enroute FTE	2 nd Order	0.13 nm	0.7
	Gauss Markov	5.2 kts	
Piloted Enroute FTE	2 nd Order	0.7 nm	0.5
	Gauss Markov	42 kts	

Generally, the only means of testing the final algorithms is to make sure that the lateral variation stays within the standard deviation requirements. This is difficult to do using graphical representation of trajectories, so generally tabulated data were used to verify the flight technical error. However, the trajectories provide a richer understanding of what the flight technical data does to the simulation.

Four plots are presented in this section demonstrating the en route flight technical error developed for the simulation. These plots were all created using a closed route so that an aircraft would fly around the same route creating a Monte-Carlo simulation. The route was flown at different speeds using both piloted and FMS flight technical error. Figure 12.18 shows an aircraft flying at 250kts and 5000ft. This is a relatively slow configuration so the trajectories show a considerable amount of ‘wobbling’ back and forth along the route. This is because the frequency of the lateral flight technical error is not a function of the aircraft’s speed. Because the aircraft covers a greater distance per unit of time when moving faster, the flight technical error always appears to be more extreme in slower aircraft. This point is aptly demonstrated when Figure 12.18 is compared to

Figure 12.19 where a piloted aircraft is traveling at 300kts and 30,000ft. Even though the same flight technical error is in effect, the faster aircraft appears to have less ‘wobble’ in its trajectory, although the lateral offset of both plots is about the same.

The final two plots show the same aircraft using FMS. Figure 12.20 shows the aircraft moving at 250kts and 5000ft. Figure 12.9 shows the aircraft moving at 300kts and 30,000ft. Since the FMS has only about 20% of the lateral offset that the piloted flight technical error has, the aircraft trajectories are much tighter. This reduction in lateral offset over the piloted flight technical error is readily apparent; however, it is not impossible to get a sense for the ‘wobble’ of the aircraft along the route in either plot, with this scale of distance.

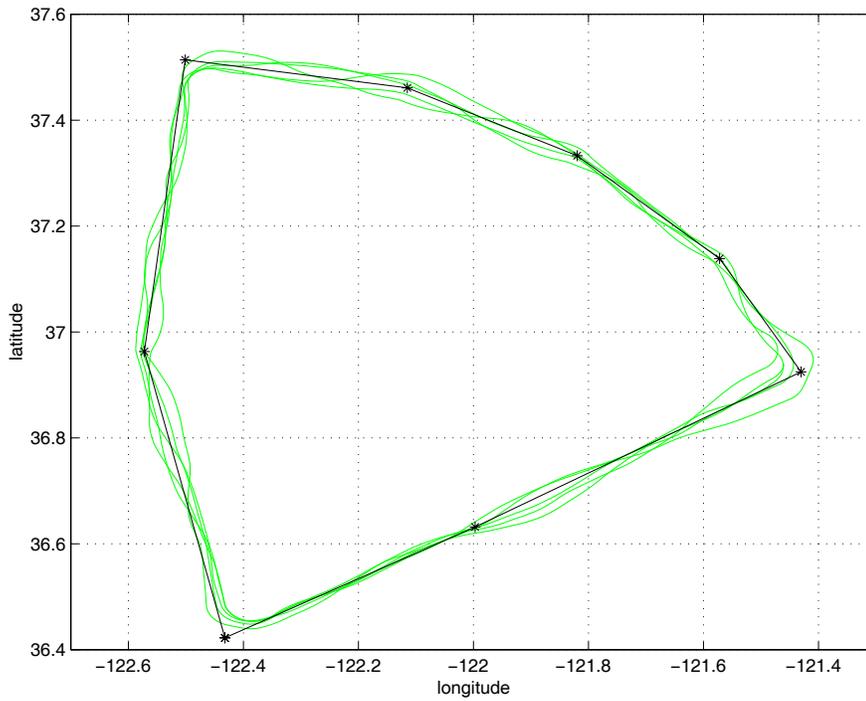


Figure 12.18. Piloted flight technical error of an MD80 traveling at 250kts and 5000ft

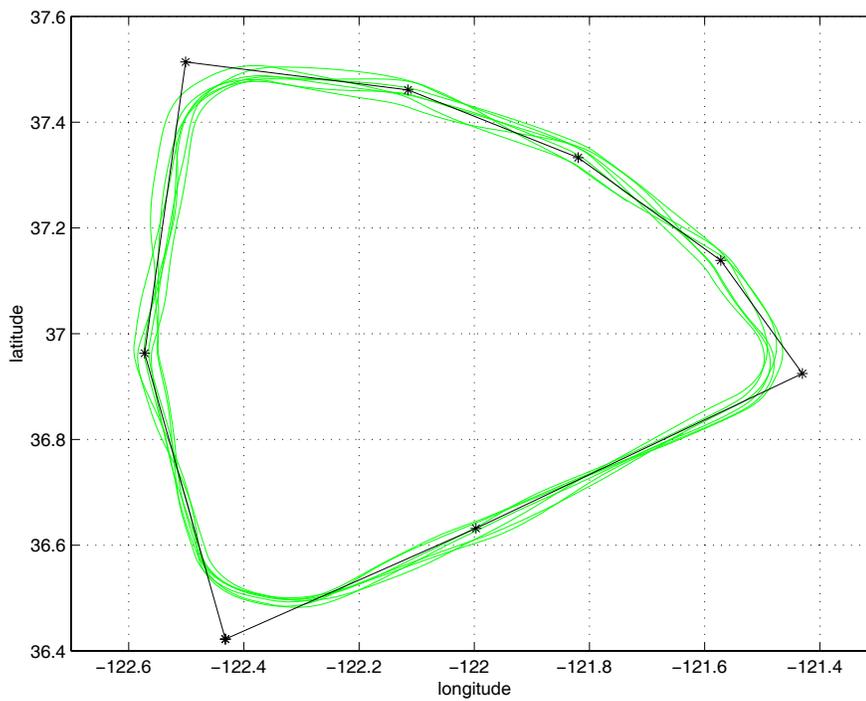


Figure 12.19. Piloted flight technical error of an MD80 traveling at 300kts and 30,000ft

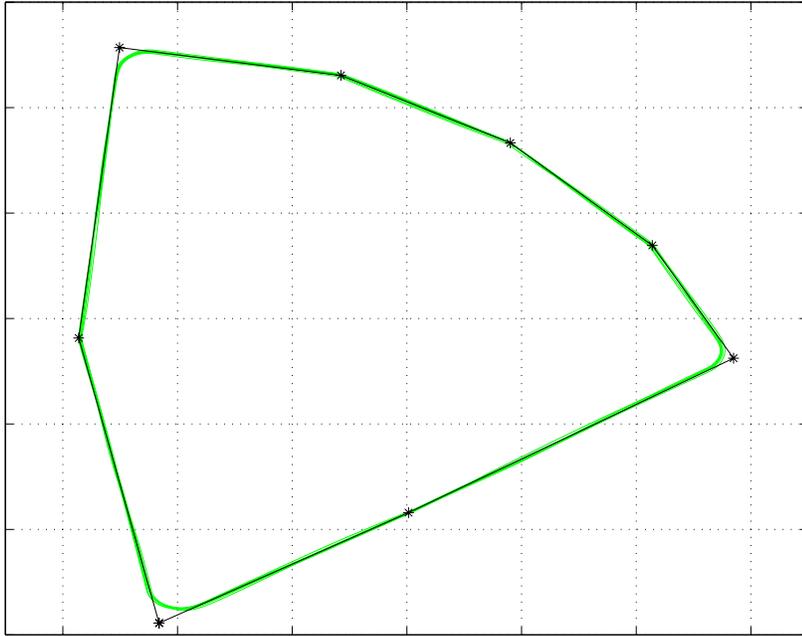


Figure 12.20. FMS flight technical error of an MD80 traveling at 250kts and 5000ft

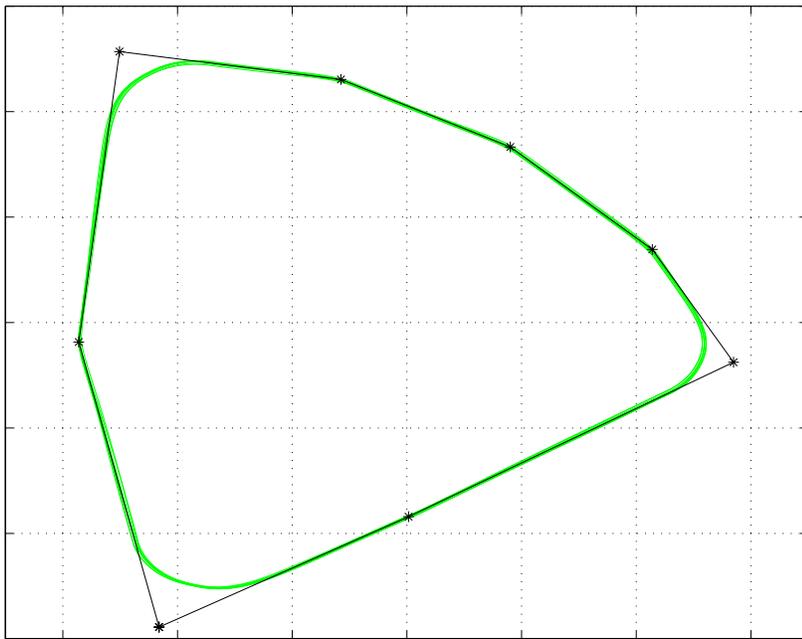


Figure 12.21. FMS flight technical error of an MD80 traveling at 300kts and 30,000ft

12.10 Navigation Errors

The purpose of navigation error modeling is to model the variances which occur in aircraft flight paths as a result of imperfect information. The two navigation types generally used within the simulation at this point for en route types of operation are VOR/DME and GPS navigation. All of the navigation models perform similarly in that they create a perturbed estimate of the aircraft's location for the guidance system to use as an input. Therefore, the navigation error models all return a latitude longitude pair which represents the aircraft's position as determined by imperfect navigation.

12.10.1 GPS Navigation Error

The GPS navigation error is so small that it is generally undetectable, for the typical scales used. For the purposes of completeness, a plot of GPS error over a route is generated, but for en route purposes, GPS error is small enough to warrant ignoring it completely. Figure 12.22 shows an aircraft flying a route. Unfortunately, there is no way of seeing the aircraft track because it is so superbly hidden by the route line itself. Even exploded views of this plot fail to reveal any interesting variation between GPS and the route.

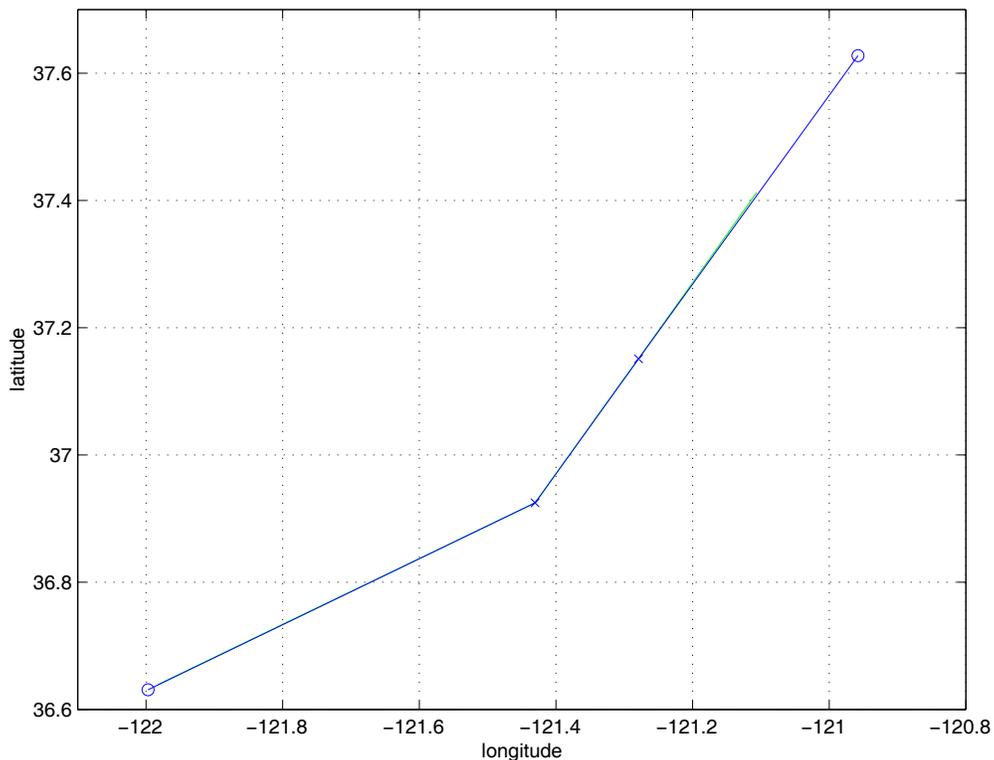


Figure 12.22. An aircraft trajectory using GPS navigation

12.10.2 VOR/DME Error

The navigation errors associated with VOR/DME error is much more interesting than those associated with GPS from a modeling point of view. Generally, VOR/DME navigation systems have biases in their angular measurements which tend to make lateral offset errors grow as a function of distance from the nav-aid. The errors also produce interesting quirks when VOR receivers are switched during navigation. For instance, an aircraft that is following one VOR for a portion of a segment may develop a lateral offset on one side of the segment and upon switching to the next VOR, immediately develop a lateral offset on the other side of the segment.

The first scenario considered is shown in Figure 12.23 where an aircraft is tracking a segment which has a VOR/DME at each endpoint. The northerly most station has a slight easterly bias which tends to make the aircraft track to the east of center. The southerly VOR/DME tends to have a slight westerly bias which makes the aircraft track to the west of center.

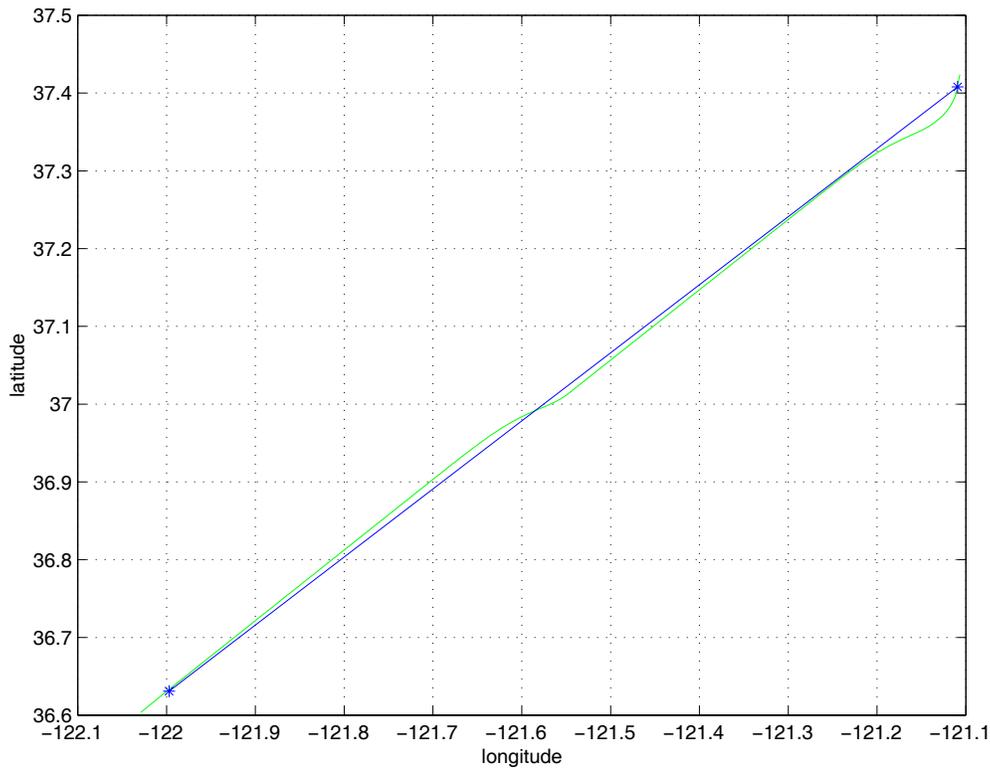


Figure 12.23. An aircraft flying a segment using VOR/DME navigation where both endpoints are VOR/DME stations

The aircraft is heading southwest along the segment, and it first captures the route shortly after passing the northerly VOR/DME. The aircraft then tracks the course using a signal from the northerly VOR/DME. As the aircraft progresses along the route, the aircraft drifts to the east slightly. When the aircraft passes the midpoint of the segment, the

aircraft switches nav-aids and uses the southerly VOR/DME instead. Upon tuning in this nav-aid, the aircraft crosses over the route and tracks on the other side because of the

slight westerly bias associated with the southern VOR/DME station. As the aircraft approaches the station, the error becomes smaller as would be expected.

The second scenario, shown in Figure 12.24, illustrates an aircraft following a route which has a VOR/DME station at each of its endpoints. However, there are two intersections in between the VOR/DME stations making three distinct segments. Because of this arrangement, the center segment does not have a VOR/DME station associated with it. In this situation, the algorithm must choose between the southerly VOR/DME and the northerly VOR/DME to determine which is most appropriate to use for navigation. From visual inspection, the human pilot would automatically choose the northerly most VOR/DME because the center segment is nearly perfectly aligned with a radial from that station. Using the southerly VOR/DME would require some sort of area navigation technique.

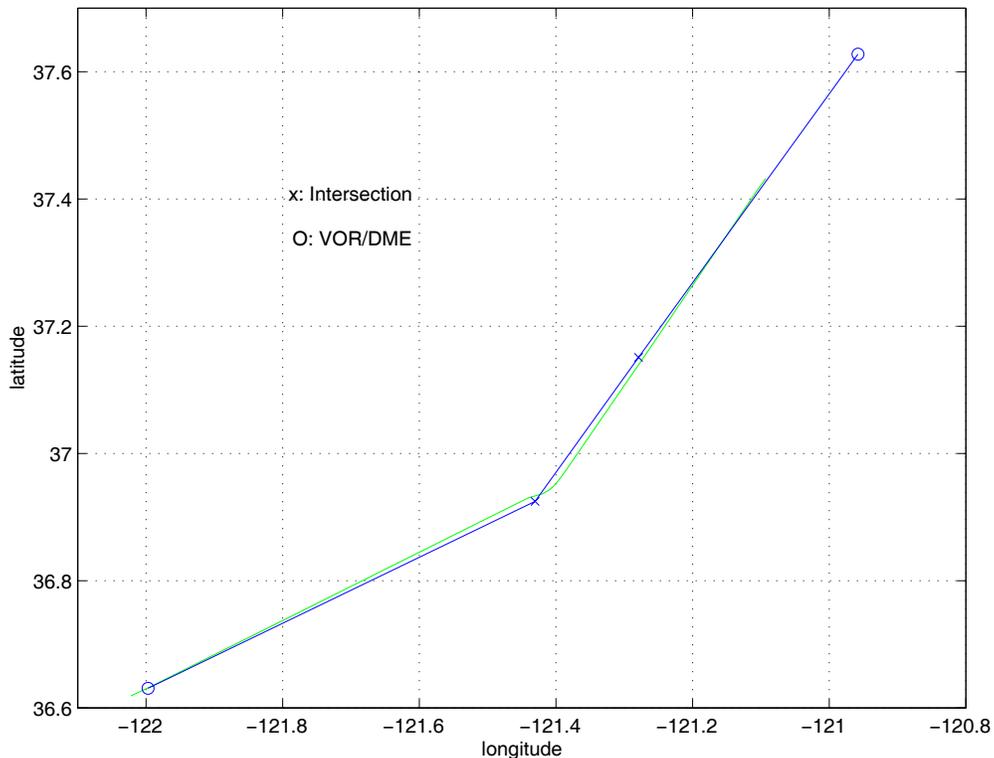


Figure 12.24. An aircraft flying a route comprised of 2 VOR/DME stations with 2 intersections between the VOR/DME stations

The aircraft is initially headed northeast on the first segment using the southerly VOR/DME for navigation. Because this is the only VOR/DME associated with the first

segment, the aircraft does not switch nav-aids. However, when the aircraft switches to the middle segment, it must make a decision about which VOR/DME station to use. As discussed in the last paragraph, the northerly VOR/DME is the best one to use and in fact, we see that the northerly VOR/DME is the one chosen by the algorithm for navigation. We know this by observing the bias that the aircraft takes after switching between the first segment and the middle segment. The aircraft has an easterly bias which is associated with the northerly VOR/DME. This bias is continually decreased as the aircraft flies further towards the VOR/DME and crosses onto the final segment. Eventually, when the aircraft crosses the northern VOR/DME, the bias is reduced to zero.

There is one other characteristic to note regarding the crossing of segment 1 onto segment 2. Notice that the aircraft is using DME to determine the end of the route rather than the intersection of VOR/DME radials. If the intersection of VOR/DME radials was used to determine the fix location, the aircraft would have estimated the fix location to be northeast of the actual fix location. Rather, the aircraft estimates the end of the route with near perfection in spite of the VOR radial biases. This is a trait of an aircraft equipped with DME as opposed to one which only has VOR navigation. This segment transition phenomena is one of the distinguishing characteristics of VOR/DME navigation as opposed to VOR/VOR navigation which is not currently modeled in the system.

12.11 Terminal Flight Phases

The terminal flight phases for the aircraft consist of take off and landing. These maneuvers are different than any which have been considered so far because they require that the aircraft fly slowly and interact with the ground.

12.11.1 Take-Off

During take-off, the aircraft initially accelerates down the runway with the landing gear initially supporting all of the aircraft weight. This phase of take-off is referred to as the ground roll. The lift coefficient is held at zero. Since we assume coordinated flight, we do not concern ourselves with keeping the aircraft on the centerline. When the aircraft reaches rotation speed, the lift coefficient of the aircraft is increased until the aircraft leaves the ground and starts climbing. The landing gear is retracted as soon as the aircraft has climbed several hundred feet. Once the landing gear is retracted, the maximum available throttle is reduced to 90% of maximum possible throttle. Next, the aircraft accelerates to a new speed of 210 kt. While speeding up, the flaps are retracted as the proper speeds are obtained. Retraction of the flaps is the last commands issued by the navigator for take-off.

Figure 12.25 illustrates the take-off of an MD80 aircraft weighing 130,000lbs. The aircraft initiates the takeoff ground roll at 20 seconds into the strip chart recording. Roughly 40 seconds later, the aircraft has achieved sufficient speed for rotation. Looking at the lift coefficient plot, we see the initial lift coefficient spike which is the control system executing the rotation. As the aircraft starts to lift off the ground we see the speed initially stabilize right around the rotation speed. Then, the aircraft starts to accelerate at

a slower rate towards 210kts. Upon reaching 210kts, the aircraft holds a constant speed and

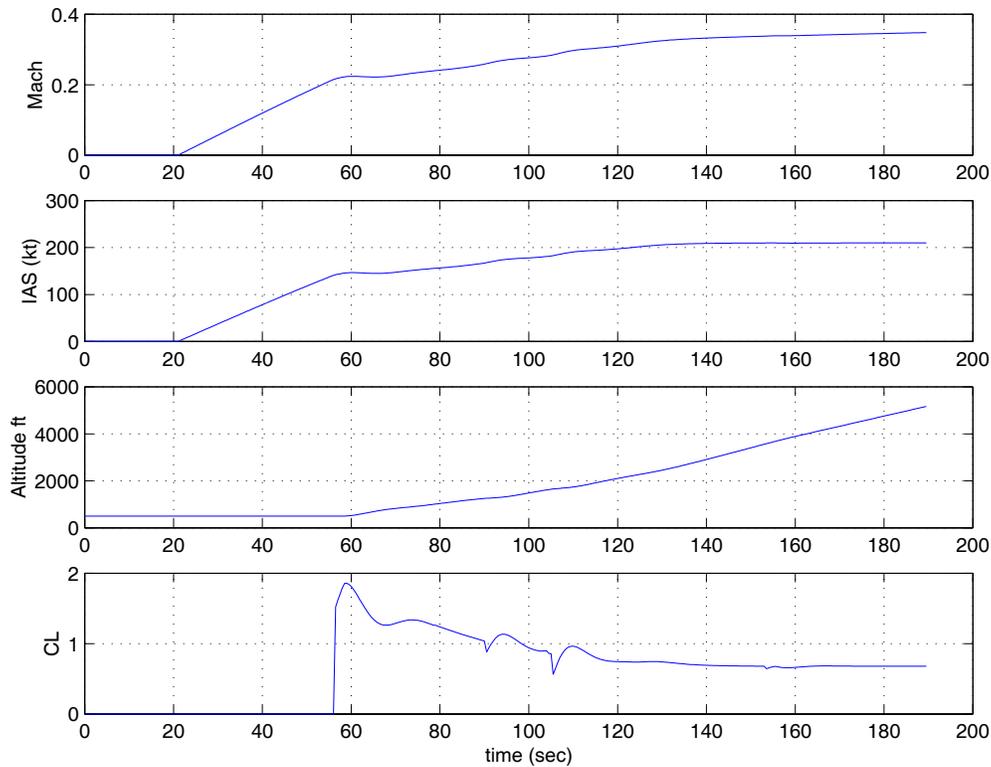


Figure 12.25. An MD80 at 130,000lbs taking off with a rotation speed of 150KIAS

continues to climb. Notice that the rate of climb increases when the aircraft stabilizes at 210kts.

One of the important features of this take-off example to notice is the simultaneous acceleration and climb between 60sec and 120sec. From Figure 12.1, we know that on takeoff the aircraft climbs and accelerates simultaneously. To cause this, special ramping of the speed input is performed as discussed in Section 12.4.3.

12.11.2 Landing

The final approach and landing is quite possibly the most difficult of all maneuvers to simulate. The aircraft must automatically follow an approach to an airport, maintaining the appropriate altitudes all along the path, and then capture the ILS localizer and glide slope for the final vertical descent. Finally, the aircraft must touchdown. All through the maneuver, the control logic must monitor and command the proper aircraft speed and make sure that the appropriate flap settings are deployed. Furthermore, the entire final

approach is flown on the back side of the thrust curve, the most difficult flight regime for the control logic.

The terminal flight phases are also the most difficult to verify. Pseudocontrol does not land aircraft so there is no acceptable baseline for insuring that the longitudinal dynamics are proper. Therefore, verification of the longitudinal performance on landing and take-off consisted of making sure that the aircraft performance conformed to the performance data which were used to create the algorithms. Such data are contained in Figures 12.1 and 12.2.

First, the longitudinal dynamics is considered. Figure 12.26 shows an MD80 on final approach and landing to an airport. By the time Figure 12.26 starts recording the approach, the aircraft has already slowed to 170kts and has captured the localizer. At roughly 55 seconds, the aircraft captures the glide slope. The aircraft simultaneously slows down to its final approach speed of 130kts. At roughly 180 seconds, the aircraft touches down and the brakes are applied. The speed reduces quickly and the aircraft is brought to a standstill by 210 seconds.

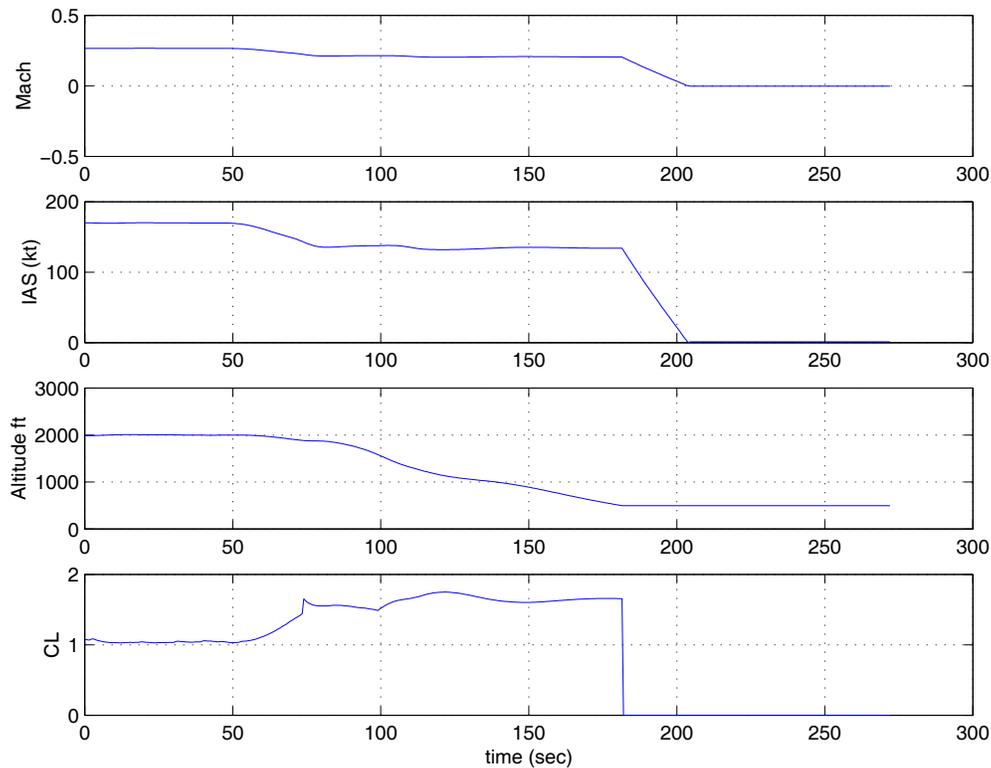


Figure 12.26. Longitudinal view of an MD80 on final approach and landing

Figure 12.27 illustrates the top view of the same MD80 from Figure 12.26. The aircraft has already captured the localizer when the track starts. The aircraft is heading southeast along the ILS. The first circle along the route is an initial approach fix located 20 miles from the airport threshold, which is marked by an 'X.' The second circle, which is 5 miles from the threshold, is the final approach fix for the ILS. For this example, both the flight technical error and the ILS beam bending model are in effect. However, as can be seen from the track in Figure 12.27, little variation is seen. This is expected considering the small standard deviations which are measured from actual data. The aircraft touches down near the threshold and stops moving about 3000 ft later.

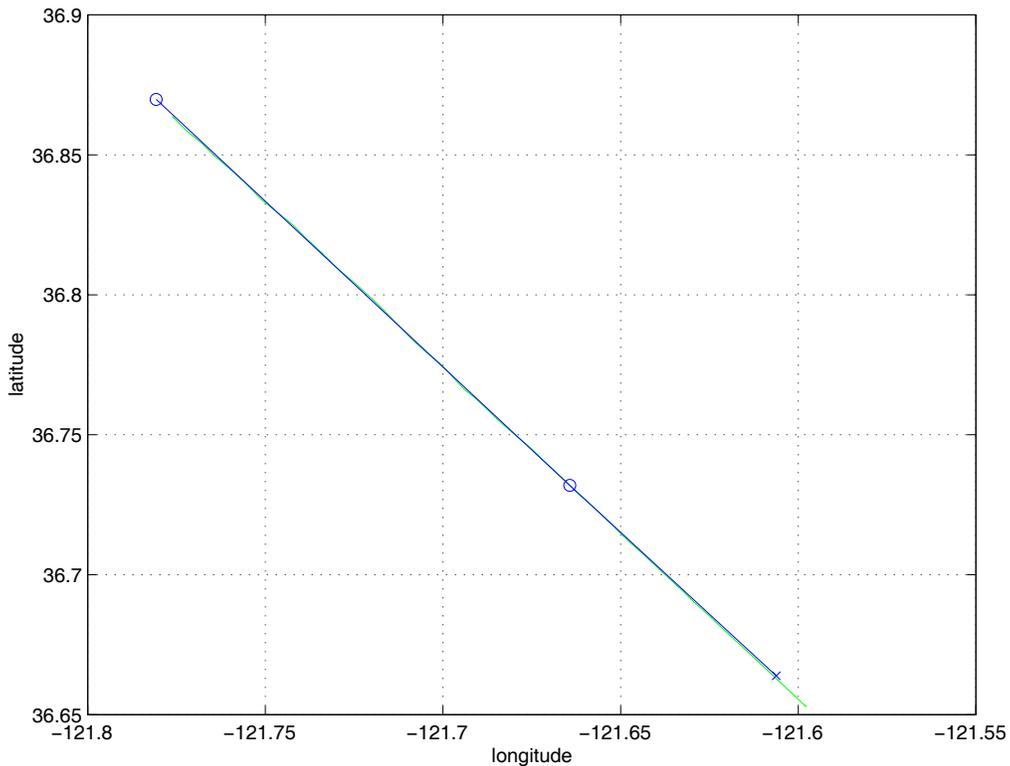


Figure 12.27. A top view of an MD80 on final approach to landing

12.12 Conclusions

The testing that was done to verify and validate the TGF simulation gives us a high degree of confidence that the models contained herein have sufficient fidelity for use as a target generating tool.

Generally, there were two means of verifying and validating the system. For the longitudinal dynamics, a quantitative measuring tool was needed to insure that the aircraft performance was realistic. The tool used was Pseudocontrol, the aircraft

dynamics kernel of PAS. PAS, the NASA tool for trajectory generation, has been accepted as an acceptable baseline for aircraft performance. For the guidance operations, such as route capture and route following, visually inspecting the maneuvers is sufficient to insure proper operation. Repeated testing of algorithms was done to insure that the route capture and route following algorithms would capture the route from all different initial conditions. Examples of the most difficult initial conditions have been discussed in this section.

Flight technical error and navigation error was validated by making sure that the modeled variances conformed to the real flight data statistics which were used to construct the models.

Terminal flight phases were the most difficult to verify because of the lack of information available for actual aircraft descents and landings. Information from flight handbooks and performance manuals was used to verify as best possible the aircraft performance on landing.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Analysis of the Transfer Functions of the Longitudinal Dynamics

In the original version of this document (in section 3.3), an analysis was conducted on the transfer function between a lift coefficient input and an altitude rate output for a single-input, single-output (SISO) system. This was intended to provide insight to the controllability of altitude rate using lift coefficient. There was concern about the right-half-plane zeros in the transfer function when the aircraft is flying on the back-side of the thrust curve. The analysis concluded that a control reversal occurred in this regime making it impossible to maintain stable flight when using lift coefficient to control altitude rate.

The analysis failed to consider that in steady, level flight on the back-side of the thrust curve, the system is not SISO; thrust is used to control speed while lift coefficient is used to control altitude rate. And in level acceleration where the system is SISO, the linearized system of equations (3.27) and (3.28) does not apply. The reader is reminded that the system is linearized about a steady, level reference condition in which thrust equals drag. In level acceleration, the throttle is advanced to full and is greater than drag and energy is being added to the system. A similar argument can be made for level deceleration.

To get an indication of the response of the altitude rate to lift coefficient input in steady, level flight, we can modify the LTD system of equations to absorb the thrust control and then analyze the system as a SISO, LTD system. The purpose of this appendix is to blend the optimized thrust control gains of Chapter 4 into the LTD system and then analyze the $\frac{\Delta \dot{h}}{u_{P_{CL}}}$ transfer function. The analysis shows that, in this dual input, dual output system, thrust is adjusted to counteract the adverse effect of the changing lift coefficient on the speed so that the commanded altitude rate can be captured. Equations (4.26) are restated here.

$$\begin{bmatrix} \dot{V}_a \\ \dot{\gamma}_a \\ \dot{I}_{C_L} \\ \dot{I}_T \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_a \\ \gamma_a \\ I_{C_L} \\ I_T \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & 0 & 0 \\ b_{21} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{P_{CL}} \\ u_{P_T} \\ u_{i_{CL}} \\ u_{i_T} \end{bmatrix} \quad (4.26)$$

$$\mathbf{y} = \begin{bmatrix} \Delta M \\ \Delta \dot{h} \end{bmatrix} = \begin{bmatrix} c_{21} & 0 & 0 & 0 \\ 0 & c_{42} & 0 & 0 \end{bmatrix} \begin{bmatrix} V_a \\ \gamma_a \\ I_{C_L} \\ I_T \end{bmatrix}$$

The control law is reduced from equation (4.14).

$$\mathbf{u} = \begin{bmatrix} u_{P_{CL}} \\ u_{P_T} \\ u_{i_{CL}} \\ u_{i_T} \end{bmatrix} = - \begin{bmatrix} 0 & k_{p_{14}} \\ k_{p_{22}} & 0 \\ 0 & k_{i_{14}} \\ k_{i_{22}} & 0 \end{bmatrix} \begin{bmatrix} \Delta M \\ \Delta \dot{h} \end{bmatrix}$$

In order to incorporate the thrust control into the LTD system, the thrust and lift coefficient controls must first be separated. We also re-introduce the vectors as explicit functions of time.

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} u_{P_{CL}} \\ u_{P_T} \\ u_{i_{CL}} \\ u_{i_T} \end{bmatrix} = \begin{bmatrix} u_{P_{CL}} \\ 0 \\ u_{i_{CL}} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ u_{P_T} \\ 0 \\ u_{i_T} \end{bmatrix} = - \begin{bmatrix} 0 & k_{p_{14}} \\ 0 & 0 \\ 0 & k_{i_{14}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta M \\ \Delta \dot{h} \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ k_{p_{22}} & 0 \\ 0 & 0 \\ k_{i_{22}} & 0 \end{bmatrix} \begin{bmatrix} \Delta M \\ \Delta \dot{h} \end{bmatrix} \\ \mathbf{u}(t) &= \mathbf{u}_{CL}(t) + \mathbf{u}_T(t) = -\mathbf{K}_{CL}\mathbf{y}(t) - \mathbf{K}_T\mathbf{y}(t) \end{aligned}$$

Then, assuming \mathbf{K}_T is known.

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}(\mathbf{u}_{CL}(t) + \mathbf{u}_T(t)) \\ &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}_{CL}(t) + \mathbf{B}\mathbf{u}_T(t) = (\mathbf{A} - \mathbf{B}\mathbf{K}_T\mathbf{C})\mathbf{x}(t) + \mathbf{B}\mathbf{u}_{CL}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}_{CL}(t) \\ \mathbf{u}_{CL}(t) &= -\mathbf{K}_{CL}\mathbf{y}(t) \end{aligned}$$

We arrive at the transfer function by converting to the laplace domain and solving the matrix algebra.

$$\begin{aligned} s\mathbf{x}(s) &= (\mathbf{A} - \mathbf{B}\mathbf{K}_T\mathbf{C})\mathbf{x}(s) + \mathbf{B}\mathbf{u}_{CL}(s) \\ \{s\mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{K}_T\mathbf{C})\}\mathbf{x}(s) &= \mathbf{B}\mathbf{u}_{CL}(s) \\ \mathbf{x}(s) &= \{s\mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{K}_T\mathbf{C})\}^{-1} \mathbf{B}\mathbf{u}_{CL}(s) \\ \mathbf{y}(s) &= \left(\mathbf{C}\{s\mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{K}_T\mathbf{C})\}^{-1} \mathbf{B} + \mathbf{D} \right) \mathbf{u}_{CL}(s) \end{aligned} \tag{A.1}$$

We first solve for the closed-loop A matrix.

$$\begin{aligned}
(\mathbf{A} - \mathbf{BK}_T\mathbf{C}) &= \left(\begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} & 0 & 0 \\ b_{21} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ k_{p22} & 0 \\ 0 & 0 \\ k_{i22} & 0 \end{bmatrix} \begin{bmatrix} c_{21} & 0 & 0 & 0 \\ 0 & c_{42} & 0 & 0 \end{bmatrix} \right) \\
&= \left(\begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} & 0 & 0 \\ b_{21} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ k_{p22}c_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_{i22}c_{21} & 0 & 0 & 0 \end{bmatrix} \right) \\
&= \left(\begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} b_{12}k_{p22}c_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_{i22}c_{21} & 0 & 0 & 0 \end{bmatrix} \right) \\
&= \begin{bmatrix} a_{11} - b_{12}k_{p22}c_{21} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ -k_{i22}c_{21} & 0 & 0 & 0 \end{bmatrix}
\end{aligned}$$

Then,

$$\begin{aligned}
\{s\mathbf{I} - (\mathbf{A} - \mathbf{BK}_T\mathbf{C})\} &= \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & s \end{bmatrix} - \begin{bmatrix} a_{11} - b_{12}k_{p22}c_{21} & a_{12} & b_{11} & b_{12} \\ a_{21} & 0 & b_{21} & 0 \\ 0 & 0 & 0 & 0 \\ -k_{i22}c_{21} & 0 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} s - (a_{11} - b_{12}k_{p22}c_{21}) & -a_{12} & -b_{11} & -b_{12} \\ -a_{21} & s & -b_{21} & 0 \\ 0 & 0 & s & 0 \\ k_{i22}c_{21} & 0 & 0 & s \end{bmatrix}
\end{aligned}$$

Computation of the inverse of this matrix is not trivial. We arrived at the solution using the algebraic matrix inversion functions of MATLAB[®].

$$\left\{ s\mathbf{I} - \begin{pmatrix} \mathbf{A} - \mathbf{BK}_T \mathbf{C} \end{pmatrix} \right\}^{-1} = \frac{1}{s^2 + \begin{pmatrix} -a_{11} + b_{12}k_{12}p_{22} & c_{21} \end{pmatrix} s + \begin{pmatrix} b_{12}k_{12}c_{21} - a_{12}a_{21} \end{pmatrix}} \begin{bmatrix} s & a_{12} & \frac{b_{11}s + a_{12}b_{21}}{s} & b_{12} \\ a_{21} & \frac{s^2 + \begin{pmatrix} -a_{11} + b_{12}k_{12}p_{22} & c_{21} \end{pmatrix} s + b_{12}k_{12}c_{21}}{s} & \frac{b_{21}s^2 + \begin{pmatrix} -a_{11}b_{21} + b_{12}b_{21}k_{12}p_{22} & c_{21} + a_{21}b_{11} \end{pmatrix} s + b_{12}b_{21}k_{12}c_{21}}{s^2} & \frac{a_{21}b_{12}}{s} \\ 0 & 0 & \frac{s^2 + \begin{pmatrix} -a_{11} + b_{12}k_{12}p_{22} & c_{21} \end{pmatrix} s + \begin{pmatrix} b_{12}k_{12}c_{21} - a_{12}a_{21} \end{pmatrix}}{s} & 0 \\ -k_{12}c_{21} & \frac{-a_{12}k_{12}c_{21}}{s} & -k_{12}c_{21} \frac{b_{11}s + a_{12}b_{21}}{s^2} & \frac{s^2 + \begin{pmatrix} -a_{11} + b_{12}k_{12}p_{22} & c_{21} \end{pmatrix} s - a_{12}a_{21}}{s} \end{bmatrix}$$

Because of the form of the \mathbf{C} -matrix, which pre-multiplies the $\left\{s\mathbf{I} - (\mathbf{A} - \mathbf{BK}_T\mathbf{C})\right\}^{-1}$ matrix in the equation (A.1), the $\frac{\Delta\dot{h}}{u_{p_{CL}}}$ transfer function is concerned only with second row of the $\left\{s\mathbf{I} - (\mathbf{A} - \mathbf{BK}_T\mathbf{C})\right\}^{-1}$ matrix. (Note: transposed notation is used to conserve page space.)

$$\left\{s\mathbf{I} - (\mathbf{A} - \mathbf{BK}_T\mathbf{C})\right\}_{(2,:)}^{-1} = \left[\begin{array}{c} a_{21} \\ \frac{s^2 + (-a_{11} + b_{12}k_{p_{22}}c_{21})s + b_{12}k_{i_{22}}c_{21}}{s} \\ \frac{b_{21}s^2 + (-a_{11}b_{21} + b_{12}b_{21}k_{p_{22}}c_{21} + a_{21}b_{11})s + b_{12}b_{21}k_{i_{22}}c_{21}}{s^2} \\ \frac{a_{21}b_{12}}{s} \end{array} \right]^T$$

$$s^2 + (-a_{11} + b_{12}k_{p_{22}}c_{21})s + (b_{12}k_{i_{22}}c_{21} - a_{12}a_{21})$$

Post-multiplying the \mathbf{B} -matrix, we get,

$$\left\{s\mathbf{I} - (\mathbf{A} - \mathbf{BK}_T\mathbf{C})\right\}_{(2,:)}^{-1} \mathbf{B} = \left[\begin{array}{c} a_{21}b_{11} + \frac{s^2 + (-a_{11} + b_{12}k_{p_{22}}c_{21})s + b_{12}k_{i_{22}}c_{21}}{s} b_{21} \\ a_{21}b_{12} \\ \frac{b_{21}s^2 + (-a_{11}b_{21} + b_{12}b_{21}k_{p_{22}}c_{21} + a_{21}b_{11})s + b_{12}b_{21}k_{i_{22}}c_{21}}{s^2} \\ \frac{a_{21}b_{12}}{s} \end{array} \right]^T$$

$$s^2 + (-a_{11} + b_{12}k_{p_{22}}c_{21})s + (b_{12}k_{i_{22}}c_{21} - a_{12}a_{21})$$

and pre-multiplying the \mathbf{C} -matrix, we get,

$$\mathbf{C}\left\{s\mathbf{I} - (\mathbf{A} - \mathbf{BK}_T\mathbf{C})\right\}^{-1} \mathbf{B} = \begin{bmatrix} a_{21}b_{11}c_{42} + \frac{s^2 + (-a_{11} + b_{12}k_{p22}c_{21})s + b_{12}k_{i22}c_{21}}{s} b_{21}c_{42} \\ a_{21}b_{12}c_{42} \\ \frac{b_{21}s^2 + (-a_{11}b_{21} + b_{12}b_{21}k_{p22}c_{21} + a_{21}b_{11})s + b_{12}b_{21}k_{i22}c_{21}}{s^2} c_{42} \\ \frac{a_{21}b_{12}c_{42}}{s} \end{bmatrix}^T$$

$$\frac{\Delta \dot{h}}{u_{p_{CL}}} \text{ (2,1)} = \frac{s^2 + (-a_{11} + b_{12}k_{p22}c_{21})s + (b_{12}k_{i22}c_{21} - a_{12}a_{21})}{s^2 + (-a_{11} + b_{12}k_{p22}c_{21})s + (b_{12}k_{i22}c_{21} - a_{12}a_{21})}$$

The $\frac{\Delta \dot{h}}{u_{p_{CL}}}$ transfer function is element (2,1) of the $\mathbf{C}\left\{s\mathbf{I} - (\mathbf{A} - \mathbf{BK}_T\mathbf{C})\right\}^{-1} \mathbf{B}$ matrix.

$$\frac{\Delta \dot{h}}{u_{p_{CL}}} = \frac{a_{21}b_{11}c_{42} + \frac{s^2 + (-a_{11} + b_{12}k_{p22}c_{21})s + b_{12}k_{i22}c_{21}}{s} b_{21}c_{42}}{s^2 + (-a_{11} + b_{12}k_{p22}c_{21})s + (b_{12}k_{i22}c_{21} - a_{12}a_{21})}$$

$$\frac{\Delta \dot{h}}{u_{p_{CL}}} = c_{42} \frac{b_{21}s^2 + (-a_{11}b_{21} + a_{21}b_{11} + b_{12}b_{21}c_{21}k_{p22})s + b_{12}b_{21}c_{21}k_{i22}}{s(s^2 + (-a_{11} + b_{12}k_{p22}c_{21})s + (b_{12}k_{i22}c_{21} - a_{12}a_{21}))} \quad (\text{A.2})$$

We verify that by setting $k_{p21} = k_{i21} = 0$ and substituting in the matrix partial derivatives, we get equation (3.67).

$$\frac{\Delta \dot{h}}{u_{p_{CL}}} = c_{42} \frac{b_{21}s + (-a_{11}b_{21} + a_{21}b_{11})}{s^2 - a_{11}s - a_{12}a_{21}}$$

$$a_{11} = \frac{\partial f_{V_a}}{\partial V_a}, \quad a_{12} = \frac{\partial f_{V_a}}{\partial \gamma_a}, \quad a_{21} = \frac{\partial f_{\gamma_a}}{\partial V_a}, \quad b_{11} = \frac{\partial f_{V_a}}{\partial C_L}, \quad b_{21} = \frac{\partial f_{\gamma_a}}{\partial C_L}, \quad c_{42} = 60 \frac{\partial f_h}{\partial \gamma_a}$$

$$\frac{\Delta \dot{h}}{u_{p_{CL}}} = 60 \frac{\frac{\partial f_h}{\partial \gamma_a} \frac{\partial f_{\gamma_a}}{\partial C_L} s + \left(-\frac{\partial f_{V_a}}{\partial V_a} \frac{\partial f_{\gamma_a}}{\partial C_L} + \frac{\partial f_{\gamma_a}}{\partial V_a} \frac{\partial f_{V_a}}{\partial C_L} \right)}{s^2 - \frac{\partial f_{V_a}}{\partial V_a} s - \frac{\partial f_{V_a}}{\partial \gamma_a} \frac{\partial f_{\gamma_a}}{\partial V_a}} \quad (\text{3.67})$$

But of course, we are concerned with analyzing the transfer function with the optimized thrust gains left in. For a B763 at stall speed and 30,000 ft. equation (A.2) becomes,

$$\frac{\Delta \dot{h}}{u_{p_{CL}}} = \frac{1640s^2 + 631s + 96.7}{s(s^2 + 0.406s + 0.0758)}$$

which has complex conjugate zeros at $z_{1,2} = -0.19 \pm 0.152i$. In other words, thrust control of speed moves the $\frac{\Delta \dot{h}}{u_{p_{CL}}}$ transfer function zeros into the left-half plane, thereby eradicating the non-minimum phase system. The integral control transfer function, $\frac{\Delta \dot{h}}{u_{i_{CL}}}$, is the same as equation (A.2) except with an added integral pole at $s = 0$. The root loci of proportional and integral $\frac{\Delta \dot{h}}{C_L}$ transfer functions with thrust control of speed are shown in Figure A.1. The figure shows that lift coefficient control of altitude rate is well-behaved as long as speed is controlled by thrust simultaneously. We do, however, have to be mindful of low damping in this area, as indicated by the locus moving up the imaginary axis of the integral control root locus plot.

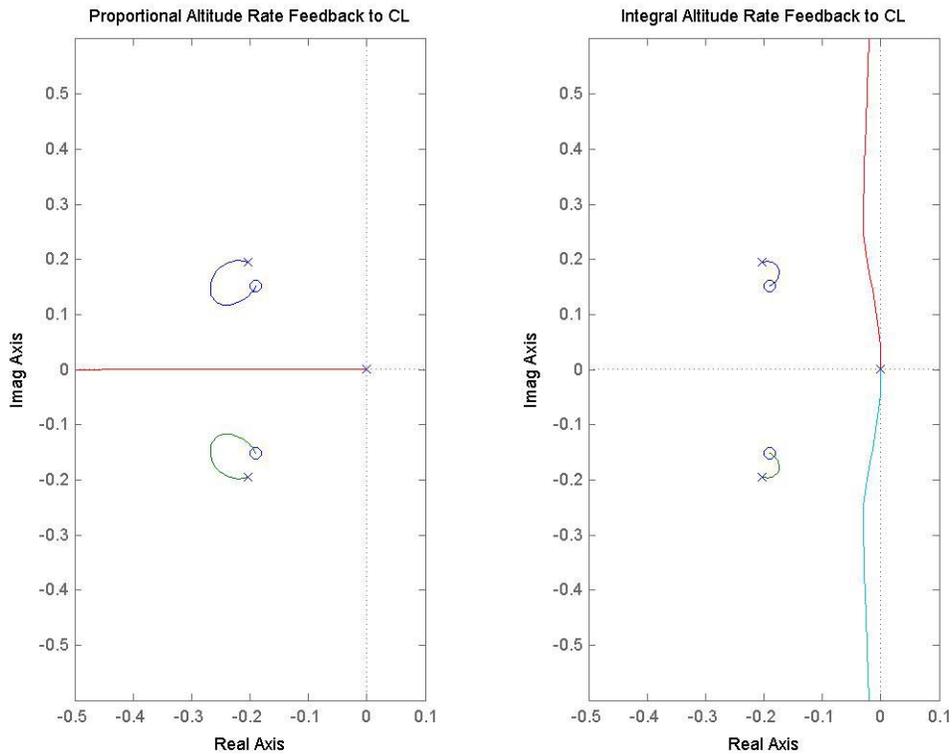


Figure A.1: The root loci of proportional and integral lift coefficient control of altitude rate considering the thrust control of speed

THIS PAGE INTENTIONALLY LEFT BLANK

Glossary

Azimuth	An angle measured relative to the ground-based coordinate system (i.e., true north).
Bearing	The azimuth direction of the position vector from one point to another (e.g., from an aircraft to a fix).
Drag	The component of aerodynamic force acting parallel to the aircraft's longitudinal axis and in a direction opposite the thrust. It is defined positive in the direction of the negative x -axis of the body axis system.
Dutch Roll Mode	A coupled roll and yaw motion that is often insufficiently damped.
Empty Weight	The weight of a fully operational aircraft without fuel or payload.
Flight Path Angle	The angle that the true airspeed vector makes with a horizontal plane.
Fuel Weight	The fuel capacity of the aircraft.
Geocentric Latitude	The angle between a line from center of the earth to the given point and the equatorial plane.
Geodetic Latitude	The angle between a line perpendicular to the surface of the ellipsoidal earth at the given point and the equatorial plane.
Ground Track Heading	The angle that the aircraft's ground speed vector makes with the ground-based coordinate system (i.e., true-north). This is the azimuth of the aircraft's velocity vector. The difference between ground track heading and true heading is due to the wind.
Ground Track Speed	The speed of the aircraft over ground. In other words, it is the magnitude of the aircraft's true airspeed projected to a horizontal plane.
Heading	The azimuth of the aircraft's nose (i.e., longitudinal axis).
Indicated Airspeed	This is the speed shown by an aircraft's airspeed indicator, as calculated from the measured local dynamic pressure. Its difference from true airspeed increases with altitude. (Also known as Calibrated Airspeed.)

Lift	The component of aerodynamic force acting normal to the plane formed by the true airspeed vector and the aircraft's lateral axis.
Mach Number	The ratio of the true airspeed to the local speed of sound.
Magnetic [Azimuth]	An azimuth angle (e.g., heading, bearing) measured relative to magnetic north.
Payload Weight	The payload capacity of the aircraft.
Phugoid Mode	An oscillatory mode of aircraft dynamics in which kinetic and potential energy are exchanged. The angle of attack is mainly unchanged.
Pitch Angle	The angle that the aircraft's longitudinal axis makes with the ground. (Also known as Elevation Angle)
Rhumb Line	A straight line on a Mercator projection of the earth. It is convenient in navigation because it yields the constant bearing to be followed for navigating between the two end points of the rhumb line.
Roll Angle	The angle that the aircraft's lateral axis makes with the ground. (Also known as Bank Angle)
Short Period Mode	An oscillatory motion in the axis of rotation of pitch. The angle of attack is constantly changing. This mode is typically much faster than the phugoid.
Thrust	The thrust force created by the aircraft's engines. Acts along the aircraft's longitudinal axis and is defined positive in the direction of the x -axis of the body axis system.
True [Azimuth]	An azimuth angle (e.g., heading, bearing) measured relative to true north.
True Airspeed	The actual speed of the aircraft relative to the surrounding air mass.

Acronyms

ADM	aircraft dynamics model
AGL	above ground level
AMT	aircraft modeling tool
ATC	air traffic control
ATM	air traffic management
BADA	Base of Aircraft Data
CAS	calibrated airspeed
DIS	Distributed Interactive Simulation (DIS provides a military standard earth coordinate system.)
DME	Distance Measure Equipment
DOF	degree of freedom
ECEF	earth-centered, earth-fixed
FAA	Federal Aviation Administration
FTE	flight technical error
GPS	Global Positioning System
IAS	indicated airspeed
LTD	linear, time-dependent
NAS	National Airspace System
NED	the North-East-Down coordinate system
RNAV	Radio Navigation
SISO	single input, single output
TGF	Target Generation Facility
VOR	VHF Omnidirectional Range navigation system

THIS PAGE INTENTIONALLY LEFT BLANK

References

- [A89] Anderson, J.D. *Introduction To Flight*, McGraw-Hill, New York, 1989.
- [TGF93] Anonymous, "System Segment Specification", TGF-SY01-Rev 1, ", William J. Hughes Technical Center, Atlantic City, NJ, 1993.
- [Bo96] Bos, A., *Equivalences Report for the Base of Aircraft Data (BADA)*, EUROCONTROL Experimental Centre Publications Office, France, 1996.
- [Br91] Brogan, W.L., *Modern Control Theory*, Prentice Hall, Inc, Englewood Cliffs, NJ, 1991.
- [C95] Clark, K. C., *Analytical and Computer Cartography*, Prentice Hall, Englewood Cliffs, NJ, 1995.
- [DIS98] Anonymous, "Unapproved Standards Draft for DIS", William J. Hughes Technical Center, Atlantic City, NJ, 1998.
- [G74] Gelb, A., *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, MA, 1974.
- [H92] Hoffman, J.D., *Numerical Methods for Engineers and Scientists*, McGraw-Hill, New York, 1992.
- [Hu96] Hunter, G., *CTAS Error Sensitivity, Fuel Efficiency, and Throughput Benefits Analysis*, Tech. Report 96150-02, Seagull Technology, Los Gatos, CA, April 1996.
- [KF97] Kayton, M., Fried, W.R., *Avionics Navigation Systems*, John Wiley & Sons, Inc. New York, 1997.
- [M92] Mukai, C., *Design and Analysis of Aircraft Dynamics Models for the ATC Simulation at NASA Ames Research Center*, Tech. Report 92119-02, Seagull Technology, Los Gatos, CA, April 1992.
- [N89] Nelson, R.C., *Flight Stability and Automatic Control*, McGraw-Hill, New York, 1989.
- [O90] Ogata, K., *Modern Control Engineering*, Prentice Hall, Englewood Cliffs, NJ, 1990.
- [PS96] Parkinson, B.W., Spilker, J.J., *Global Positioning System: Theory and Applications*, American Institute of Aeronautics and Astronautics, Washington, DC, 1996.

[SL92] Stevens, B.L., Lewis, F. L. *Aircraft Control and Simulation*, John Wiley & sons, Inc., New York, 1992.

[TT90] Thomas, J., Timoteo, D., *Chicago O'Hare Simultaneous ILS Approach Data Collection and Analysis*, DOT/FAA/CT-TN90/11, William J. Hughes Technical Center, Atlantic City, NJ, 1990.