

Reentry Guidance with Extended Range Capability for Low L/D Spacecraft

by

Sarah Hendrickson Bairstow
B.S. Engineering and Applied Science (Aeronautics)
California Institute of Technology, 2004

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2006

© 2006 Sarah Hendrickson Bairstow. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and
electronic copies of this thesis document in whole or in part, in any medium now known or
hereafter created.

Signature of Author: _____
Department of Aeronautics and Astronautics
February 3, 2006

Approved by: _____
Gregg H. Barton
Group Leader, Mission Design and Analysis
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by: _____
Professor John J. Deyst, Jr.
Professor of Aeronautics and Astronautics
Thesis Advisor

Accepted by: _____
Professor Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

[This Page Intentionally Left Blank]

Reentry Guidance with Extended Range Capability for Low L/D Spacecraft

by

Sarah Hendrickson Bairstow

Submitted to the Department of Aeronautics and Astronautics on
February 3, 2006 in partial fulfillment of the requirements
for the Degree of Master of Science in Aeronautics and Astronautics

Abstract

A generalized reentry/precision landing algorithm using bank angle modulation control was designed for a low lift-to-drag ratio (L/D) spacecraft that enables precision landing for target locations between 2,400 km and 10,000 km downrange of Entry Interface (EI). The Crew Exploration Vehicle (CEV) design concept was used as the main test case for this algorithm, but the algorithm is general enough to be extensible to similar vehicle concepts with different vehicle characteristics. The algorithm was tested against various reentry scenarios including perturbations in initial entry conditions, vehicle mass and aerodynamic properties, and atmospheric density. The algorithm was shown to be robust to these uncertainties to allow a landing error of less than 3.5 km for the entire 2,400 km – 10,000 km landing footprint.

The guidance algorithm is based on the Apollo entry guidance algorithm. The guidance phases pertaining to short range reentries have remained essentially unaltered. The phases relating to longer range reentries, which rely upon skip trajectories, have been upgraded using PredGuid, a numeric predictor-corrector aerocapture algorithm developed by Draper Laboratory for the Aero-assist Flight Experiment in the late 1980's. In addition, the reference trajectory used for the final phase of reentry was recalculated to apply to the CEV vehicle concept.

These upgrades were sufficient to allow precision landing of skip reentry trajectories for target ranges of up to 10,000 km. In addition, it was shown that the steepness of the skip can be controlled by modulating the time at which the PredGuid guidance phase takes over; starting earlier results in a steeper, higher altitude skip whereas starting later results in a shallower, lower altitude skip.

Technical Supervisor: Gregg H. Barton

Title: Group Leader, Mission Design and Analysis
The Charles Stark Draper Laboratory, Inc.

Thesis Supervisor: Professor John J. Deyst, Jr.

Title: Professor of Aeronautics and Astronautics

[This Page Intentionally Left Blank]

Acknowledgements

February 3, 2006

I would first like to thank the Charles Stark Draper Laboratory not just for the opportunity to pursue my graduate studies at MIT, but also for the tools to make my time here a success. The most valuable of these tools is the chance to work alongside some of the most knowledgeable, helpful people I have ever met. I truly feel that I learned just as much from my co-workers at Draper as I did in my classes at MIT. In particular, my technical supervisor, Gregg Barton, deserves many thanks for all that he has taught me. Not just my 'supervisor' or an 'advisor,' he has in fact been a mentor to me, and I know that his guidance and support will serve me well in the years to come. Thanks also to Steve Paschall, who provided the simulation I used for much of my analysis.

My thanks are much deserved by the professors in the MIT Department of Aeronautics and Astronautics for their excellent classes. Thanks especially to my faculty advisor, Professor John Deyst, for his time, effort, and support.

I'd like to thank all my classmates who helped me through those tough problem sets. My long-lost officemate, Cristin, made coming in to work every day extra fun, until she was banished to the other building. Thanks also to my friends from Caltech who followed me to Boston (or did I follow them?) for keeping me sane these long months. Thursday night dinners on Tuesday on Thursday have been a blast.

I would like to thank my family for all their support and, in the case of my brothers, teasing. Dad, I appreciate our discussions of science-related topics, even if it doesn't seem like it. Despite my initial objections, your review of my thesis turned out to be quite useful; I mostly just like to argue for the sake of arguing. Mom, our frequent phone conversations have meant a lot to me. Thanks for consoling me when things just weren't going my way. Jason and Seth, you guys helped shape me into the nerd I am today. Don't stop making those geeky jokes!

Finally, I would like to thank my wonderful husband, Brian. You are always there to support me in everything I do, and when I'm upset you always know how to cheer me up. I can only hope that I'll be able to help you half as much with your thesis as you have helped me with mine.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc. under Exploration Integrated Development Environment, Internal Research and Development project #20283.

Publication of this thesis does not constitute approval by Draper of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Permission is hereby granted by the Author to the Massachusetts Institute of Technology to reproduce any or all of this thesis.

Sarah Hendrickson Bairstow

TABLE OF CONTENTS

| | |
|---|-----------|
| 1 INTRODUCTION | 15 |
| 1.1 A VISION FOR SPACE EXPLORATION | 15 |
| 1.2 BACKGROUND..... | 16 |
| 1.2.1 BANK ANGLE MODULATION | 16 |
| 1.2.2 GUIDANCE METHODS..... | 17 |
| 1.3 PROBLEM DEFINITION..... | 19 |
| 1.4 THESIS OBJECTIVE..... | 20 |
| 1.5 THESIS OVERVIEW | 21 |
| 2 SIMULATION DESCRIPTION | 23 |
| 2.1 OVERVIEW | 23 |
| 2.2 REFERENCE COORDINATE FRAMES | 23 |
| 2.3 COORDINATE TRANSFORMATIONS | 24 |
| 2.4 ENVIRONMENT MODELS | 27 |
| 2.4.1 EARTH ATMOSPHERE MODEL | 27 |
| 2.4.2 EARTH GRAVITY MODEL | 27 |
| 2.5 VEHICLE MODEL..... | 27 |
| 2.6 EQUATIONS OF MOTION..... | 29 |
| 2.6.1 TRANSLATIONAL MOTION | 29 |
| 2.6.2 ROTATIONAL MOTION | 31 |
| 2.7 AERODYNAMIC HEATING..... | 32 |
| 2.8 SIMULATION INITIALIZATION..... | 33 |
| 2.8.1 INITIAL POSITION | 33 |
| 2.8.2 INITIAL VELOCITY | 34 |
| 2.8.3 TARGET LOCATION SPECIFICATION | 35 |
| 2.8.4 VEHICLE PROPERTIES | 36 |
| 2.9 SIMULATION TERMINATION CONDITIONS | 38 |

| | | |
|----------|--|-----------|
| 3 | GUIDANCE ALGORITHM METRICS..... | 39 |
| 3.1 | OVERVIEW | 39 |
| 3.2 | PERFORMANCE-BASED METRICS | 39 |
| 3.2.1 | PRECISION LANDING FOOTPRINT | 40 |
| 3.2.2 | LANDING PRECISION | 41 |
| 3.2.3 | G-LOADS..... | 43 |
| 3.2.4 | AEROHEATING | 45 |
| 3.3 | MARGIN-BASED METRICS | 46 |
| 3.3.1 | VERTICAL L/D FRACTION | 47 |
| 3.3.2 | FINAL PHASE ENERGY BUCKET AND TRANSITION | 49 |
| 4 | BASELINE GUIDANCE ALGORITHM | 53 |
| 4.1 | OVERVIEW | 53 |
| 4.1.1 | DOWNRANGE MANAGEMENT..... | 54 |
| 4.1.2 | CROSSRANGE MANAGEMENT..... | 55 |
| 4.2 | INITIAL RESULTS | 57 |
| 4.3 | ANALYSIS OF FINAL PHASE ENERGY BUCKET | 58 |
| 4.4 | ANALYSIS OF UPCONTROL PHASE TARGETING | 59 |
| 4.5 | ANALYSIS OF EXIT CONDITIONS SELECTED BY THE HUNTEST PHASE..... | 62 |
| 5 | ENHANCED GUIDANCE DESIGN | 63 |
| 5.1 | OVERVIEW | 63 |
| 5.2 | PREDICTOR-CORRECTOR TO REPLACE UPCONTROL PHASE | 63 |
| 5.3 | THE ORIGINAL PREDGUID NUMERIC PREDICTOR-CORRECTOR | 65 |
| 5.4 | PREDGUID IN THE UPCONTROL PHASE..... | 66 |
| 5.4.1 | PREDGUID IMPLEMENTATION IN THE UPCONTROL PHASE | 66 |
| 5.4.2 | RESULTS WITH PREDGUID IN THE UPCONTROL PHASE..... | 68 |
| 5.5 | FINAL PHASE REFERENCE TRAJECTORY AND GAIN DESIGN | 70 |
| 5.5.1 | TRAJECTORY AND GAIN DESIGN PROCEDURE..... | 71 |

| | | |
|----------|---|------------|
| 5.5.2 | CHOOSING THE REFERENCE TRAJECTORY PARAMETERS | 76 |
| 5.5.3 | UPDATED REFERENCE TRAJECTORY AND BUCKET TARGETING METHOD..... | 77 |
| 5.5.4 | RESULTS WITH NEW REFERENCE AND TARGETING METHOD | 79 |
| 5.6 | PREDGUID IN THE BALLISTIC PHASE | 80 |
| 5.6.1 | CONTROL AUTHORITY DURING THE BALLISTIC PHASE | 80 |
| 5.6.2 | RESULTS WITH PREDGUID IN THE BALLISTIC PHASE | 82 |
| 5.7 | PREDGUID IN THE DOWNCONTROL PHASE..... | 83 |
| 6 | RESULTS | 87 |
| 6.1 | OVERVIEW | 87 |
| 6.2 | TEST CASE SUMMARY | 87 |
| 6.2.1 | PRECISION LANDING FOOTPRINT | 88 |
| 6.2.2 | G-LOADS..... | 88 |
| 6.2.3 | AEROHEATING | 93 |
| 6.2.4 | CONTROL AUTHORITY | 96 |
| 6.2.5 | FINAL PHASE ENERGY BUCKET | 99 |
| 6.3 | ROBUSTNESS TO ATMOSPHERIC UNCERTAINTY | 102 |
| 7 | CONCLUSIONS..... | 105 |
| 7.1 | SUMMARY AND CONCLUSIONS | 105 |
| 7.2 | FUTURE WORK | 105 |
| | APPENDIX A APOLLO GUIDANCE ALGORITHM | 107 |
| A.1 | BASELINE ALGORITHM | 107 |
| A.2 | GUIDANCE INTERFACES | 108 |
| A.3 | VERIFICATION..... | 109 |
| | APPENDIX B ENHANCED ALGORITHM WITH PREDGUID..... | 113 |
| B.1 | OVERVIEW | 113 |
| B.2 | REENTRY GUIDANCE EXECUTIVE | 113 |
| B.2.1 | GUIDANCE FUNCTIONAL FLOW..... | 113 |

| | | |
|---|---|------------|
| B.2.2 | INPUTS | 114 |
| B.2.3 | OUTPUTS..... | 115 |
| B.2.4 | STATE VARIABLES | 115 |
| B.2.5 | CONSTANTS | 117 |
| B.3 | TARGETING SUBROUTINE | 122 |
| B.4 | PRE-ENTRY ATTITUDE HOLD AND INITIAL ROLL PHASES | 124 |
| B.5 | HUNTEST/CONSTANT DRAG PHASE | 126 |
| B.6 | DOWNCONTROL AND UPCONTROL PHASES | 131 |
| B.7 | PREDGUID SUBROUTINE | 133 |
| B.7.1 | PREDICTOR | 136 |
| B.7.2 | CORRECTOR..... | 137 |
| B.8 | BALLISTIC PHASE | 139 |
| B.9 | FINAL PHASE | 140 |
| B.10 | LATERAL LOGIC SUBROUTINE | 141 |
| APPENDIX C MONTE CARLO METHOD AND PARAMETERS . | | 145 |
| C.1 | MONTE CARLO METHOD | 145 |
| C.2 | PARAMETER VALUES AND UNCERTAINTIES | 147 |
| APPENDIX D METRIC CALCULATIONS | | 149 |
| D.1 | LANDING ERROR..... | 149 |
| D.2 | DURATION-BASED G-LOADS | 151 |
| D.3 | FINAL PHASE ENERGY BUCKET | 152 |
| APPENDIX E RESULTS – METRICS..... | | 155 |
| E.1 | ORIGINAL APOLLO ALGORITHM | 155 |
| E.2 | LOW LOFT ENHANCED ALGORITHM | 164 |
| E.3 | HIGH LOFT ENHANCED ALGORITHM..... | 172 |
| REFERENCES..... | | 181 |

List of Figures

| | |
|---|----|
| Figure 1.1: Bank Angle Modulation..... | 16 |
| Figure 1.2: Direct Reentry vs. Skip Reentry | 19 |
| Figure 2.1: Coordinate Reference Frames..... | 24 |
| Figure 2.2: Vehicle Orientation with Respect to Stability Axes | 29 |
| Figure 2.3: Target Location Specification..... | 36 |
| Figure 3.1: Sample Landing Error Distribution..... | 42 |
| Figure 3.2: Sample G-Load Histogram | 44 |
| Figure 3.3: NASA STD-3000 G-Load Guidelines and Sample Duration-Based Monte Carlo Results..... | 45 |
| Figure 3.4: Sample Peak Heat Rate Histogram | 46 |
| Figure 3.5: Sample Total Heat Load Histogram..... | 46 |
| Figure 3.6: Sample Trajectory L/D Fraction vs. Time | 48 |
| Figure 3.7: Final Phase Energy Bucket | 49 |
| Figure 3.8: Poor Bucket Performance | 51 |
| Figure 3.9: Good Bucket Performance | 52 |
| Figure 4.1: Reentry Guidance Phases [11] | 54 |
| Figure 4.2: Bank Angle Reversal | 56 |
| Figure 4.3: Typical Trial Groundtrack with Bank Reversals | 56 |
| Figure 4.4: Landing Error Scatter Plots for Various Target Ranges. | 57 |
| Figure 4.5: Transition into the Final Phase, Apollo Algorithm, 7,300 km..... | 58 |
| Figure 4.6: Transition into the Final Phase, Apollo Algorithm, 10,000 km..... | 59 |
| Figure 4.7: Commanded L/D Fraction during the Upcontrol Phase..... | 60 |
| Figure 4.8: Upcontrol Velocity Errors..... | 60 |
| Figure 4.9: Upcontrol Altitude Rate Errors | 61 |
| Figure 4.10: Ballistic Range Calculation Errors..... | 62 |
| Figure 5.1: The PredGuid Numeric Predictor-Corrector..... | 65 |
| Figure 5.2: Guidance Targeting Flow during Upcontrol Phase..... | 67 |
| Figure 5.3: Improved Precision Landing with PredGuid Replacing Upcontrol | 69 |
| Figure 5.4: Transition into the Final Phase, PredGuid Replacing Upcontrol Phase..... | 70 |
| Figure 5.5: Reference Trajectory Generation and Gain Design Procedure | 71 |
| Figure 5.6: Final Phase Reference L/D | 77 |
| Figure 5.7: Final Phase Range Estimation: Original vs. New Method..... | 78 |
| Figure 5.8: New Reference Results, 10,000 km Target..... | 79 |

| | |
|---|-----|
| Figure 5.9: New Reference Results, 2,400 km Target..... | 80 |
| Figure 5.10: Ballistic Phase Control Authority [22]..... | 81 |
| Figure 5.11: Results with PredGuid Replacing Ballistic Phase..... | 82 |
| Figure 5.12: Comparison of Trajectory Shaping..... | 83 |
| Figure 6.1: G-loading vs. Time, Low Loft Algorithm, Direct Reentry | 89 |
| Figure 6.2: G-loading vs. Time, Low Loft Algorithm, Skip Reentry..... | 90 |
| Figure 6.3: Final Phase Transition, Original Apollo Algorithm, 3,500 km..... | 91 |
| Figure 6.4: G-loading vs. Time, Original Apollo Algorithm, 3,500 km Case..... | 91 |
| Figure 6.5: Duration-Based G-loads, Original Apollo Algorithm, 2,400 km Case..... | 92 |
| Figure 6.6: Duration-Based G-loads, Original Apollo Algorithm, 10,000 km Case..... | 92 |
| Figure 6.7: Landing Error Scatter Plot, Original Apollo Algorithm, 10,000 km Case..... | 93 |
| Figure 6.8: Heat Rate vs. Time, Low Loft Algorithm, Direct Reentry | 94 |
| Figure 6.9: Heat Rate vs. Time, Low Loft Algorithm, Skip Reentry | 95 |
| Figure 6.10: Control Authority Saturation, Long Target Ranges, Original Apollo Algorithm.... | 96 |
| Figure 6.11: Control Authority Saturation, Medium Target Range, Original Apollo Algorithm . | 97 |
| Figure 6.12: Control Authority Saturation, Long Target Ranges, Low Loft Algorithm | 98 |
| Figure 6.13: Control Authority Saturation, Long Target Ranges, High Loft Algorithm | 98 |
| Figure 6.14: Apollo Algorithm Trajectories in Energy Bucket, 10,000 km Case | 99 |
| Figure 6.15: Low Loft Trajectories in Energy Bucket, 10,000 km Case..... | 100 |
| Figure 6.16: High Loft Trajectories in Energy Bucket, 10,000 km Case..... | 100 |
| Figure 6.17: Apollo Algorithm Trajectories in Energy Bucket, 2,400 km Case | 101 |
| Figure 6.18: Low Loft Trajectories in Energy Bucket, 2,400 km Case..... | 101 |
| Figure 6.19: High Loft Trajectories in Energy Bucket, 2,400 km Case | 102 |
| Figure 6.20: Upcontrol and Ballistic Phase Error Analysis..... | 103 |
| Figure 6.21: Atmosphere Model Mismatch: Thick-to-Thin Shear | 104 |
| Figure A.1: Comparison to Apollo Test Case; Acceleration, Altitude, and Inertial Downrange [9] | 110 |
| Figure A.2: Comparison to Apollo Test Case; Roll Angle, Lateral Angle, and Lateral Switch [9] | 111 |
| Figure B.1: Reentry Guidance Functional Flow..... | 114 |
| Figure B.2: Targeting Logical Flow | 124 |
| Figure B.3: Initial Roll Logical Flow | 125 |
| Figure B.4: Hunttest Logical Flow | 130 |
| Figure B.5: Constant Drag Logical Flow | 131 |
| Figure B.6: Upcontrol Logical Flow (Low Loft) | 132 |
| Figure B.7: Upcontrol Logical Flow (High Loft)..... | 133 |

| | |
|---|-----|
| Figure B.8: PredGuid Subroutine Logical Flow..... | 135 |
| Figure B.9: Ballistic Logical Flow | 139 |
| Figure B.10: Final Logical Flow | 141 |
| Figure B.11: Lateral Logic - Logical Flow | 143 |
| Figure C.1: Multiplier for Coefficients of Drag and Lift | 146 |
| Figure D.1: Projection of Target Vector through Time of Flight [9] | 149 |
| Figure E.1: Landing Precision Scatter Plots, Original Apollo Algorithm, All Ranges | 156 |
| Figure E.2: Max G-load Histograms, Original Apollo Algorithm, All Ranges | 157 |
| Figure E.3: Duration-based G-loads, Original Apollo Algorithm, All Ranges | 158 |
| Figure E.4: Max Heat Rate Histograms, Original Apollo Algorithm, All Ranges..... | 159 |
| Figure E.5: Total Heat Load Histograms, Original Apollo Algorithm, All Ranges..... | 160 |
| Figure E.6: Commanded L/D Fraction, Original Apollo Algorithm, All Ranges | 161 |
| Figure E.7: Energy Bucket Traces, Original Apollo Algorithm, All Ranges..... | 162 |
| Figure E.8: Energy Bucket Transitions, Original Apollo Algorithm, All Ranges | 163 |
| Figure E.9: Landing Precision Scatter Plots, Low Loft Enhanced Algorithm, All Ranges..... | 164 |
| Figure E.10: Max G-load Histograms, Low Loft Enhanced Algorithm, All Ranges | 165 |
| Figure E.11: Duration-based G-loads, Low Loft Enhanced Algorithm, All Ranges | 166 |
| Figure E.12: Max Heat Rate Histograms, Low Loft Enhanced Algorithm, All Ranges | 167 |
| Figure E.13: Total Heat Load Histograms, Low Loft Enhanced Algorithm, All Ranges | 168 |
| Figure E.14: Commanded L/D Fraction, Low Loft Enhanced Algorithm, All Ranges..... | 169 |
| Figure E.15: Energy Bucket Traces, Low Loft Enhanced Algorithm, All Ranges | 170 |
| Figure E.16: Energy Bucket Transitions, Low Loft Enhanced Algorithm, All Ranges | 171 |
| Figure E.17: Landing Precision Scatter Plots, High Loft Enhanced Algorithm, All Ranges..... | 172 |
| Figure E.18: Max G-load Histograms, High Loft Enhanced Algorithm, All Ranges | 173 |
| Figure E.19: Duration-based G-loads, High Loft Enhanced Algorithm, All Ranges..... | 174 |
| Figure E.20: Max Heat Rate Histograms, High Loft Enhanced Algorithm, All Ranges..... | 175 |
| Figure E.21: Total Heat Load Histograms, High Loft Enhanced Algorithm, All Ranges..... | 176 |
| Figure E.22: Commanded L/D Fraction, High Loft Enhanced Algorithm, All Ranges | 177 |
| Figure E.23: Energy Bucket Traces, High Loft Enhanced Algorithm, All Ranges..... | 178 |
| Figure E.24: Energy Bucket Transitions, High Loft Enhanced Algorithm, All Ranges | 179 |

List of Tables

| | |
|---|-----|
| Table 2.1: Variation of Aerodynamic Properties with Mach Number | 37 |
| Table 5.1: Linear Expansion Example | 78 |
| Table 5.2: Pros and Cons of Low Altitude Skip vs. High Altitude Skip..... | 85 |
| Table 6.1: Landing Precision (CEP) by Algorithm Version and Target Range | 88 |
| Table 6.2: Maximum G-loading by Algorithm Version and Target Range | 89 |
| Table 6.3: Maximum Heat Rate by Algorithm Version and Target Range..... | 94 |
| Table 6.4: Total Heat Load by Algorithm Version and Target Range | 95 |
| Table A.1: Units Abbreviations..... | 109 |
| Table A.2: Apollo Guidance Inputs..... | 109 |
| Table A.3: Apollo Guidance Outputs | 109 |
| Table B.1: Reentry Guidance Inputs | 114 |
| Table B.2: Reentry Guidance Outputs..... | 115 |
| Table B.3: Reentry Guidance State Variables | 115 |
| Table B.4: Reentry Guidance Outputs..... | 117 |
| Table B.5: Original Apollo Final Phase Reference Trajectory Table | 118 |
| Table B.6: New Final Phase Reference Trajectory Table for Enhanced Reentry Guidance | 120 |
| Table B.7: PredGuid Constants | 121 |
| Table B.8: Inputs to pc_sequencer | 135 |
| Table B.9: Outputs from pc_sequencer | 135 |
| Table B.10: New Integration Stop Conditions for PredGuid | 137 |
| Table B.11: PredGuid Corrector Comparison | 138 |
| Table C.1: Monte Carlo Parameter Values..... | 147 |
| Table D.1: Maximum G-load Limits for Automated Crew Abort/Escape | 151 |
| Table D.2: Nominal G-load Limits for Conditioned Crew | 151 |
| Table D.3: G-load Limits for Deconditioned, Ill, or Injured Crew | 152 |

Chapter 1

Introduction

1.1 A Vision for Space Exploration

In January 2004, President George W. Bush set forth a new vision for space exploration to reinvigorate and focus the United States manned space program [1]. This vision calls for a return to the Moon by 2020, but with the Space Shuttle fleet aging and its maintenance costs increasing, a new space vehicle becomes necessary – the Crew Exploration Vehicle (CEV). The CEV is intended to be the primary vehicle for ferrying astronauts to and from Low Earth Orbit and eventually to the Moon and beyond. President Bush’s plan calls for the CEV to be designed by 2008, which leaves the National Aeronautics and Space Administration (NASA) a very short timeline for vehicle design. After intensive study, taking into account performance, cost, and timeline considerations, NASA has decided that the CEV will be a capsule-type vehicle very similar to the Command Module used in the Apollo space program.

Yet, NASA does not want just to repeat a previous space program – the people at NASA are committed to forwarding the field of space exploration. The vision calls for an eventual permanent moonbase, so they are thinking large-scale and long-term. For that reason, they have decided that the CEV will carry more crew members than the Apollo Command Module, and for a longer duration. In addition, it will be reusable, which means it must be recovered after landing. In order to minimize the resources needed for recovery, the CEV will touch down at a small, designated landing site on dry land. Thus, precision landing capability for the reentry guidance algorithm will be required for the CEV.

One important aspect of manned space exploration is, of course, the safety of the astronauts. It is necessary to provide safety considerations at all stages of flight; during launch, in flight, and during reentry. This is why a robust reentry guidance algorithm is not only desirable, it is essential.

1.2 Background

There are a variety of guidance methods which have been investigated for use in spacecraft atmospheric reentry. Of course, the applicability of the algorithm will partially depend on the methods of flight control being used. The CEV, as with the Apollo Command Module, utilizes only bank angle modulation to steer during reentry.

1.2.1 Bank Angle Modulation

An object moving in an airstream always experiences a drag force in the direction opposite its velocity relative to the airflow. If the lift-to-drag ratio (L/D) is non-zero, the object will also experience a lift force acting in a direction perpendicular to the velocity vector. The method of bank angle modulation is based on the condition that the vehicle has a non-zero L/D.

By changing the orientation of the vehicle with respect to the oncoming airflow, the vehicle can change the direction of its lift vector. If the lift vector points downward, the vehicle will accelerate downward and reach the ground sooner, whereas the lift vector pointing upward will allow the vehicle to stay in the air longer and increase its range. If the lift vector points to the side, the vehicle will steer out-of-plane. It turns out that the direction of the lift vector has a surprisingly large effect on the trajectory of the vehicle, even with a relatively small L/D typical of a capsule.

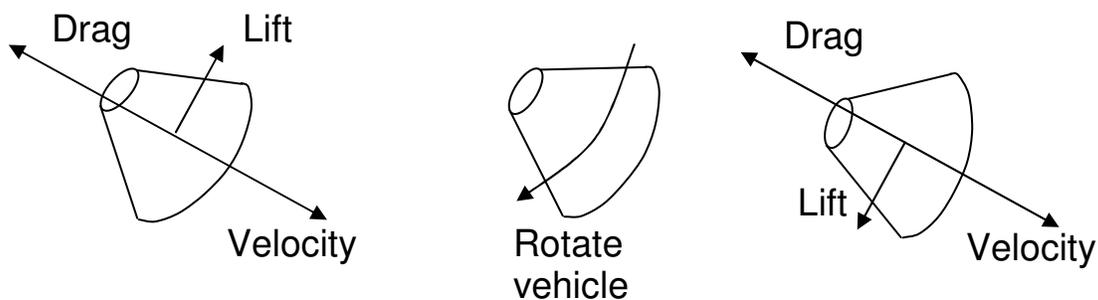


Figure 1.1: Bank Angle Modulation

The vehicle's orientation is controlled using the reaction control system (RCS) thrusters. Certain rotations are not practical, since deviations from the equilibrium orientation cause large restoring moments. However, it is possible to cause a bank angle rotation (a coupled roll and yaw rotation) without any restoring moments. This means that bank

angle can be varied in a limited manner to control the direction of the lift vector. Hence, the method is called ‘bank angle modulation,’ and a conceptual drawing of this method is presented in Figure 1.1. Through the method of bank angle modulation, the vehicle can manage energy and range to target, in order to achieve precision landing.

1.2.2 Guidance Methods

Reentry is simply a specific application of general guidance principles, so guidance methods used for similar purposes may easily be pertinent to reentry guidance. Aerocapture, for example, is in some ways similar to reentry. A spacecraft performing aerocapture dives into the atmosphere as it is passing by a planet in order to change its trajectory into an orbit about that planet. Just like a reentry algorithm, an aerocapture algorithm must:

1. Capture into the atmosphere,
2. Manage energy by removing excess velocity through drag management, and
3. Steer to a target. For reentry the target is a landing site, and for aerocapture the target is a set of atmospheric exit conditions.

In addition, most aerocapture algorithms have been devised with bank angle modulation as the sole method of flight control. Thus, reentry guidance design for a capsule may benefit a great deal from recent aerocapture research.

Some particular methods of guidance are described below, namely the numeric predictor-corrector, the analytic predictor-corrector, and the reference-following controller. These methods are all expressed in the context of bank angle modulation as the method of flight control. It should be noted that it is not necessary to use a single guidance method for the entire reentry; these guidance methods, and others, can be sequenced into various guidance ‘phases,’ which can be used in succession to accomplish reentry.

A **numeric predictor-corrector** is an iterative algorithm which assumes a certain bank angle policy – in this case, a constant bank angle – and then predicts a final state, such as landing location, by numerically integrating the equations of motion. Based on the results of the integration, the corrector chooses a new bank angle policy – in this case, a *different* constant bank angle – and then the predictor runs again based on this new

policy. The algorithm iterates between the predictor and the corrector until a bank angle policy is found which allows the vehicle to reach the desired final state, within some specified tolerance. This iteration is completed at every guidance cycle, based on current actual conditions. Some such algorithms for reentry are described in References [2] and [3]. References [2] and [4] give examples of numeric predictor-corrector algorithms for aerocapture.

Similar to the numeric predictor-corrector, an **analytic predictor-corrector** uses the same iterative process between a predictor and corrector, but instead of numerically integrating the equations of motion, the predictor uses a closed form expression to determine the vehicle's final state. Reference [5] describes such an algorithm for reentry and References [6] and [7] describe ones for aerocapture.

A **reference-following controller** attempts to follow a reference trajectory which is typically computed pre-flight. In flight, guidance determines a bank angle command to correct the deviation of actual variables (e.g. velocity, altitude, altitude rate) from the reference variables. Typically this is accomplished by multiplying the amount of the deviations (the 'control errors') by some pre-determined, optimized control gains. This correction is then applied to the reference bank angle profile to determine what the actual bank angle profile should be. Reference [8] describes a reference-following controller for entry at Mars. This algorithm is based on part of the Apollo guidance algorithm, described in Reference [9] and later in this thesis.

Each of these methods has certain advantages and disadvantages. A reference-following controller, for example, is much simpler to implement and requires less computation to execute than the other two methods described above. However, since the reference trajectory is defined beforehand, it is less flexible to changes in flight conditions. In this respect, an analytic predictor-corrector is more flexible, since it can constantly update knowledge of vehicle characteristics and atmospheric parameters in its closed form expression. However, derivation of an appropriate closed form expression typically requires some simplification of the equations of motion. Thus, any results will be, by their underlying nature, limited in accuracy. A numeric predictor-corrector can integrate unsimplified, nonlinear equations of motion, allowing improved accuracy over an analytic predictor-corrector. The disadvantage is that code involving numeric integration is far more difficult to verify as flight-ready than code involving only closed form expressions. In addition, numeric integration takes much more computational power than

the other two methods described above. These considerations must be taken into account when choosing a guidance method. Reference [10] provides a comparison study including these three guidance methods as applied to aerocapture.

1.3 Problem Definition

Although the early U.S. space programs demonstrated near-precision landing of capsule-type vehicles in water, NASA has not yet demonstrated the capability for precision landing of a capsule on dry land. From a guidance standpoint, the main differences between these two cases are that there are fewer acceptable target sites on land, and the landing sites would typically be smaller in size. Almost anywhere in the ocean would be satisfactory for a splashdown, but only a few target sites on land would be adequate. With only a few landing sites to choose from, the guidance algorithm for a land-targeting capsule must be capable of reaching as large a region as possible, in order to assure accessibility of at least one target site. A capsule with limited lift-to-drag ratio can use a ‘skip’ trajectory to increase its range capability.

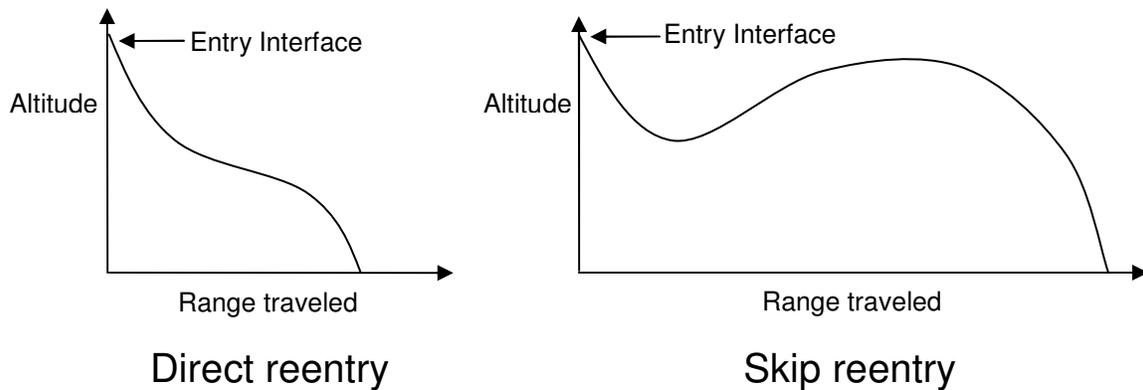


Figure 1.2: Direct Reentry vs. Skip Reentry

Traditionally, reentry vehicles follow a trajectory which is monotonically decreasing in altitude – a ‘direct’ reentry. A ‘skip’ trajectory, however, descends during the first leg of the trajectory, ascends for some duration in the middle of the trajectory, and then finally descends again to reach the ground, as shown in Figure 1.2. This type of trajectory would allow the vehicle to travel much farther from Entry Interface (EI) – the starting position of reentry – than a traditional direct reentry.

The reentry algorithm used by the Apollo Command Module was designed to target splashdown sites with downrange distances of 2,400 km to 4,600 km from EI [11], a range variation which was designed to allow for bad-weather avoidance. The Apollo algorithm included logic to carry out direct trajectories for short-range targets and skip trajectories for longer range targets. The maximum reentry range ever actually flown by a manned Apollo capsule was 3,400 km [12], so it appears that the skip capability was never used.

1.4 Thesis Objective

The objective of this thesis is to show the feasibility of a generalized reentry/precision landing algorithm for a low L/D spacecraft that enables precision landing for target locations between 2,400 and 10,000 km from EI. The 2,400 km limit matches the short-range design limit for the Apollo algorithm, and the 10,000 km limit is consistent with the maximum range required to achieve a land-landing within the continental United States on a lunar-return trajectory assuming worst-case phasing (antipode in southern latitudes). In addition, the algorithm must be robust to uncertainties and dispersions in various reentry scenarios. The Crew Exploration Vehicle (CEV) design concept will be used as the main test case for this algorithm, but the algorithm should be general enough to be extensible to similar vehicle concepts with different vehicle characteristics.

This algorithm will be based on the Apollo reentry guidance algorithm. The Apollo algorithm works on the principle of bank angle modulation to manage energy and range to target. Having already been rigorously tested, verified, and flown, the Apollo algorithm provides a fairly robust basis from which to derive the new algorithm even though the skip capability was never used. It has been found that direct application of the algorithm to the CEV concept is not sufficient to enable precision landing within the CEV's entire corridor. In particular, extended range targets (those beyond the original Apollo algorithm's specified capability) are not achievable using the original skip guidance, which was based on a reference-following controller to shape the ascending portion of the skip. The skip guidance reference trajectory, calculated in-flight using approximate closed form expressions, was not updated once the skip was initiated.

In order to enable precision landing for extended ranges, replacement of Apollo's reference-following controller skip guidance with a predictor-corrector algorithm will be

investigated. In addition, the reference trajectory used for the final descent guidance – another reference-following controller – will be updated to improve the robustness of the skip guidance.

1.5 Thesis Overview

The present chapter provides a broad overview of the subject of research for this thesis. Chapter 2 provides a description of the computer simulation environment used to test the guidance algorithm, and Chapter 3 explains the evaluation criteria used to assess how well the guidance performs in terms of results from the simulation. Chapter 4 highlights the main features of the original Apollo guidance algorithm and evaluates the algorithm's performance with application to the CEV. Chapter 5 describes the enhancements made to the algorithm to enable precision landing for skip trajectories, and Chapter 6 presents in detail the performance and robustness of the improved algorithm. Finally, Chapter 7 discusses the merits of the enhanced algorithm and provides recommendations for future research.

[This Page Intentionally Left Blank]

Chapter 2

Simulation Description

2.1 Overview

In order to evaluate the effectiveness of various guidance algorithms, it is desirable to create computer-based simulations of the vehicle during reentry. This can be done by numerically integrating the equations of motion to propagate the vehicle's trajectory forward in time. The following sections describe the coordinate frames, assumptions, and equations used in the simulation environment, which was provided by Steve Paschall.

These equations were implemented in MATLAB version 7.0.4 in conjunction with Simulink version 6.2. The integration of the equations of motion was implemented using the Bogacki-Shampine ODE (ode3) integrator [13] built into MATLAB using a fixed time step of 0.25 seconds.

2.2 Reference Coordinate Frames

There are a number of standard ways to define a coordinate frame, and the ones that pertain to this thesis are described below and illustrated below in Figure 2.1. They are all right-handed, rectangular Cartesian axes.

Inertial Reference Frame ($\hat{i}_i, \hat{j}_i, \hat{k}_i$): a non-rotating, Earth-centered frame. The \hat{k}_i axis points toward the North Pole. The \hat{i}_i axis points toward the point on the Earth's surface with zero latitude and zero longitude at time $t = 0$ (time at start of reentry). The \hat{j}_i axis completes the right-handed set.

Relative-Velocity Reference Frame ($\hat{i}_v, \hat{j}_v, \hat{k}_v$): a frame centered at the vehicle's center of gravity (CG), which references the vehicle's motion to the motion of the surrounding air. The \hat{i}_v axis points along the wind-relative velocity vector, \bar{v}_{rel} . The \hat{j}_v axis is in the 'local horizontal' plane (i.e. perpendicular to the inertial radius vector, \bar{r}_i). The \hat{k}_v axis completes the right-handed set such that it is pointing 'upward' (i.e. away from the Earth.)

Stability Reference Frame ($\hat{i}_s, \hat{j}_s, \hat{k}_s$): a coordinate frame centered at the vehicle's CG. If it is assumed that there are no side forces, the stability frame describes orientation in terms of the lift and drag forces on the vehicle. The \hat{i}_s axis coincides with the \hat{i}_v axis. The \hat{k}_s axis points along the vehicle's lift vector, and the \hat{j}_s axis completes the right-handed set.

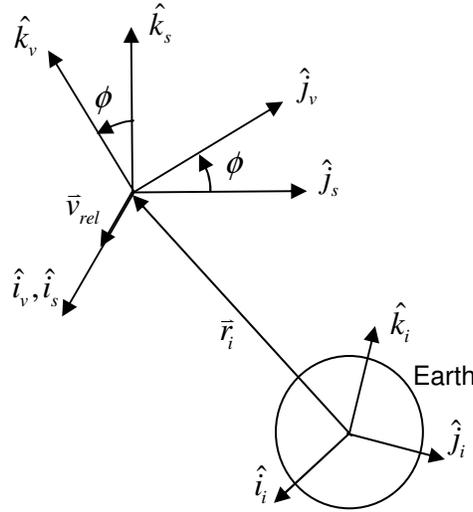


Figure 2.1: Coordinate Reference Frames

2.3 Coordinate Transformations

Converting a vector from one coordinate frame to another can be accomplished by some combination of rotation, translation, and scaling in order to match the orientation, origin of the new coordinate frame, and magnitude of the vector. Only rotational transformations must be considered here, for the following two reasons:

1. The non-inertial frames (which are accelerating with respect to the inertial frame) are centered at the vehicle's CG, which is where the accelerations are applied.
2. The only vectors for which transformations are made are acceleration vectors, which deal with changes in velocity rather than with velocities themselves.

A rotational transformation is accomplished by taking the matrix product of the original vector (\vec{v}_1) with the basis vectors for the old frame ($\hat{i}_1, \hat{j}_1, \hat{k}_1$) as expressed in the new frame. This yields the components for the vector in the new frame (\vec{v}_2). This is often

represented effectively using matrix notation, where the basis vectors for the new frame are combined into a transformation matrix ($T_{2\leftarrow 1}$), as in Equation 2.1. In this notation, vectors are considered to be column vectors unless otherwise stated.

$$\bar{v}_2 = \begin{bmatrix} \uparrow & \uparrow & \uparrow \\ (\hat{i}_1)_2 & (\hat{j}_1)_2 & (\hat{k}_1)_2 \\ \downarrow & \downarrow & \downarrow \end{bmatrix} \bar{v}_1 = T_{2\leftarrow 1} \bar{v}_1 \quad (2.1)$$

The reverse transformation (\bar{v}_2 to \bar{v}_1) can be calculated by using the inverse of the previously determined transformation matrix. Since the matrix is composed of mutually orthogonal unit vectors, the matrix is orthonormal, and thus the inverse is the same as the transpose:

$$\bar{v}_1 = T_{2\leftarrow 1}^{-1} \bar{v}_2 = T_{2\leftarrow 1}^T \bar{v}_2 \quad (2.2)$$

The two transformations used to develop the equations of motion are described below.

Relative-Velocity to Inertial Transformation: The vectors which form the transformation matrix are the basis vectors of the relative velocity frame, as expressed in the inertial frame.

$$T_{i\leftarrow v} = \begin{bmatrix} \uparrow & \uparrow & \uparrow \\ (\hat{i}_v)_i & (\hat{j}_v)_i & (\hat{k}_v)_i \\ \downarrow & \downarrow & \downarrow \end{bmatrix} \quad (2.3)$$

where:

$$(\hat{i}_v)_i = \frac{(\bar{v}_{rel})_i}{v_{rel}} \quad (2.4)$$

$$(\hat{j}_v)_i = \frac{\bar{r}_i \times (\bar{v}_{rel})_i}{|\bar{r}_i \times (\bar{v}_{rel})_i|} \quad (2.5)$$

$$(\hat{k}_v)_i = (\hat{i}_v)_i \times (\hat{j}_v)_i \quad (2.6)$$

and:

$$(\vec{v}_{rel})_i = \vec{v}_i - \vec{v}_{wind} \quad (2.7)$$

$$v_{rel} = |(\vec{v}_{rel})_i| \quad (2.8)$$

where:

\vec{r}_i : vehicle inertial position vector

\vec{v}_i : vehicle inertial velocity vector

\vec{v}_{wind} : inertial velocity of atmosphere/wind at vehicle's position

Stability to Relative-Velocity Transformation: The vectors which form the transformation matrix are the basis vectors of the stability frame, as expressed in the relative velocity frame.

$$T_{v \leftarrow s} = \begin{bmatrix} \uparrow & \uparrow & \uparrow \\ (\hat{i}_s)_v & (\hat{j}_s)_v & (\hat{k}_s)_v \\ \downarrow & \downarrow & \downarrow \end{bmatrix} \quad (2.9)$$

This is a simple rotation of the bank angle, ϕ , about the shared i axis, as shown above in Figure 2.1, so the basis vectors are as follows:

$$(\hat{i}_s)_v = (\hat{i}_v)_v = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (2.10)$$

$$(\hat{j}_s)_v = \cos \phi \cdot (\hat{j}_v)_v - \sin \phi \cdot (\hat{k}_v)_v = \begin{bmatrix} 0 \\ \cos \phi \\ -\sin \phi \end{bmatrix} \quad (2.11)$$

$$(\hat{k}_s)_v = \sin \phi \cdot (\hat{j}_v)_v + \cos \phi \cdot (\hat{k}_v)_v = \begin{bmatrix} 0 \\ \sin \phi \\ \cos \phi \end{bmatrix} \quad (2.12)$$

Note that a bank angle of zero means the vehicle is in a lift-up configuration, and the stability axes will then coincide with the relative-velocity axes.

2.4 Environment Models

The atmospheric model and gravity model defined below are empirical descriptions of the Earth environment, from which forces on the vehicle can be derived. This allows the simulation to yield results which accurately represent the real dynamics of the vehicle.

2.4.1 Earth Atmosphere Model

The atmospheric model used in the simulation environment is the Standard U.S. Atmosphere, 1962, as described in Reference [14]. The model outputs temperature, pressure, density, and Mach number as a function of the input position and velocity vectors in the inertial reference frame. Winds were not considered in this study, and were assumed to be absent. The atmosphere, as described above, is assumed to rotate with the Earth, so the atmosphere is not stationary with respect to the inertial frame.

2.4.2 Earth Gravity Model

The simulation environment uses a simple inverse-square relation for acceleration from gravity, and gravity acts in the direction opposite the radius vector:

$$\bar{a}_g = \frac{\mu_{\oplus}}{r^2} \cdot \left(-\frac{\bar{r}_i}{r} \right) = -\frac{\mu_{\oplus}}{r^3} \bar{r}_i \quad (2.13)$$

where μ_{\oplus} is an empirically determined constant property of the Earth, and r is the magnitude of \bar{r}_i . Note that this formulation of gravitational acceleration does not take into account the distortional effects due to the fact that the Earth is not a perfectly uniform sphere. A more detailed gravity model should be included in follow-on work.

2.5 Vehicle Model

In order to simplify the vehicle description, the following assumptions are made:

1. The vehicle's outer mold line is rotationally symmetrical about an axis of symmetry.
2. The vehicle's mass distribution is mirror symmetrical across the $\hat{i}_s - \hat{k}_s$ plane, but the vehicle's CG does not lie along the axis of rotational symmetry.
3. The vehicle maintains constant mass throughout reentry (i.e. any effects of fuel usage and ablation are ignored).
4. The vehicle maintains its statically trimmed orientation with respect to the relative-velocity vector (i.e. there is no motion about the \hat{j}_s and \hat{k}_s axes).
5. The vehicle produces no thrust.
6. The vehicle experiences no side forces (i.e. there is no sideslip).

These assumptions enable the vehicle to be sufficiently described by two parameters: ballistic coefficient (B_n) and lift-to-drag ratio (L/D). The first parameter, ballistic coefficient, is calculated from mass (m), coefficient of drag (C_D), and reference area (S), as shown in Equation 2.14.

$$B_n = \frac{m}{SC_D} \quad (2.14)$$

The second parameter, lift-to-drag ratio, is calculated by dividing coefficient of lift (C_L) by coefficient of drag:

$$\frac{L}{D} = \frac{C_L}{C_D} \quad (2.15)$$

The coefficients of drag and lift (C_D , C_L) depend on the orientation of the vehicle, which is shown in Figure 2.2 with respect to the stability axes.

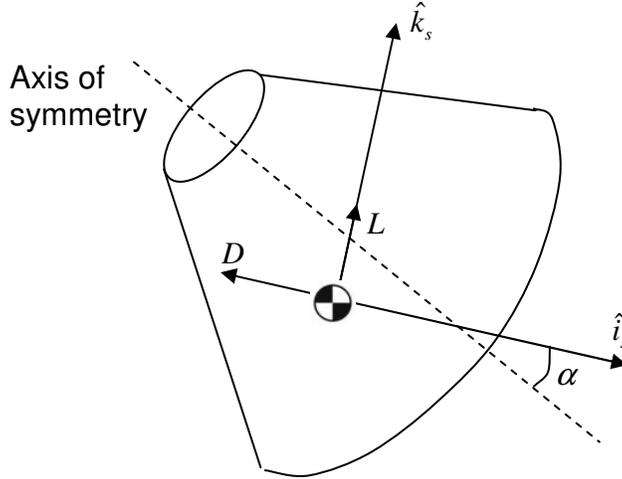


Figure 2.2: Vehicle Orientation with Respect to Stability Axes

Due to the offset of the CG from the axis of rotational symmetry, the axis of symmetry makes an angle with the \hat{i}_s axis, α , which is called the angle of attack. As stated above, it is assumed that the orientation remains statically trimmed. However, the trim angle of attack may vary depending on the character of the airflow. As angle of attack varies, the coefficients of drag and lift (C_D , C_L) will vary as well.

2.6 Equations of Motion

The simulation used in this thesis uses four degrees of freedom. All three spatial dimensions are allowed to describe the translational motion of the vehicle. The fourth degree of freedom is the rotational motion of the vehicle about the \hat{i}_s axis, described by the bank angle.

2.6.1 Translational Motion

Only gravitational and aerodynamic accelerations are assumed to contribute to the total acceleration of the vehicle:

$$\vec{a}_{tot} = \vec{a}_g + \vec{a}_a \quad (2.16)$$

Gravitational acceleration (\vec{a}_g) is described above by the Earth Gravity Model.

Aerodynamic acceleration can be calculated as a sum of drag acceleration and lift acceleration:

$$\bar{a}_a = \bar{a}_D + \bar{a}_L \quad (2.17)$$

The magnitude of the drag vector is described by:

$$a_D = \frac{\bar{q}}{B_n} \quad (2.18)$$

Dynamic pressure, \bar{q} , is dependent on current flight conditions, and is calculated from atmospheric density (ρ) and relative velocity (v_{rel}) as in Equation 2.19.

$$\bar{q} = \frac{1}{2} \rho v_{rel}^2 \quad (2.19)$$

Note that atmospheric density is determined by the Earth Atmosphere Model, described above, as a function of altitude.

The magnitude of the lift vector is related simply to drag by the vehicle's lift-to-drag ratio:

$$a_L = a_D \cdot \frac{L}{D} \quad (2.20)$$

Since the drag vector acts in the direction opposite the relative velocity vector ($-\hat{i}_s$), and lift acts perpendicular to the velocity vector (in the \hat{k}_s direction), the aerodynamic acceleration vector in the stability coordinate frame sums to:

$$(\bar{a}_a)_s = -\frac{\bar{q}}{B_n} \hat{i}_s + \frac{\bar{q}}{B_n} \cdot \frac{L}{D} \hat{k}_s \quad (2.21)$$

The aerodynamic acceleration vector must be transformed into the inertial frame for integration, as follows:

$$(\bar{a}_a)_i = T_{i \leftarrow v} T_{v \leftarrow s} (\bar{a}_a)_s \quad (2.22)$$

Since gravitational acceleration is already in the inertial frame, both gravitational and aerodynamic accelerations can be summed to yield the total inertial acceleration (\bar{a}_i):

$$\bar{a}_i = \begin{bmatrix} -\frac{\bar{q}}{B_n} i_{v_1} + \frac{\bar{q}}{B_n} \cdot \frac{L}{D} \sin(\phi) i_{v_2} + \frac{\bar{q}}{B_n} \cdot \frac{L}{D} \cos(\phi) i_{v_3} - \frac{\mu_{\oplus}}{r^3} r_{i_1} \\ -\frac{\bar{q}}{B_n} j_{v_1} + \frac{\bar{q}}{B_n} \cdot \frac{L}{D} \sin(\phi) j_{v_2} + \frac{\bar{q}}{B_n} \cdot \frac{L}{D} \cos(\phi) j_{v_3} - \frac{\mu_{\oplus}}{r^3} r_{i_2} \\ -\frac{\bar{q}}{B_n} k_{v_1} + \frac{\bar{q}}{B_n} \cdot \frac{L}{D} \sin(\phi) k_{v_2} + \frac{\bar{q}}{B_n} \cdot \frac{L}{D} \cos(\phi) k_{v_3} - \frac{\mu_{\oplus}}{r^3} r_{i_3} \end{bmatrix} \quad (2.23)$$

where:

$$\begin{bmatrix} i_{v_1} \\ i_{v_2} \\ i_{v_3} \end{bmatrix} = \hat{i}_v, \quad \begin{bmatrix} j_{v_1} \\ j_{v_2} \\ j_{v_3} \end{bmatrix} = \hat{j}_v, \quad \begin{bmatrix} k_{v_1} \\ k_{v_2} \\ k_{v_3} \end{bmatrix} = \hat{k}_v, \quad \begin{bmatrix} r_{i_1} \\ r_{i_2} \\ r_{i_3} \end{bmatrix} = \bar{r}_i \quad (2.24)$$

This equation for acceleration can be integrated once to determine inertial velocity (\bar{v}_i), and twice to determine inertial position (\bar{r}_i). Constants of integration are dependent on the initial position and velocity for the trajectory.

2.6.2 Rotational Motion

Reaction control system (RCS) jets are not modeled in this simulation. Therefore, rotational torques from which to derive bank rate dynamics are not available. Instead, the bank angle of the vehicle is restricted by a 20 deg/sec rate limit, and bank rate changes are assumed to be instantaneous. The same rate limit was used in Reference [9], and was thus deemed appropriate for this simulation.

2.7 Aerodynamic Heating

The simulation includes estimates of heat rate and heat load caused by reentry. Empirical relationships are employed for both convective and radiative heat rates. For convective aeroheating, the Chapman Equation [15] is used:

$$q_c = 17600 * \left(\frac{1}{r_n} \right)^{0.5} \left(\frac{\rho}{\rho_0} \right)^{0.5} \left(\frac{v_{rel}}{v_{cir}} \right)^{3.15} \quad (2.25)$$

where:

$$q_c: \text{convective heat rate} \left[\frac{\text{BTU}}{\text{ft}^2 \cdot \text{s}} \right]$$

$$r_n: \text{vehicle effective nose radius [ft]}$$

This calculation is only valid for relative velocities of 9,000 to 16,000 m/s, and thus is disabled for velocities below 9,000 m/s.

Radiative aeroheating is based on the Tauber-Sutton Relation:

$$q_r = C r_n^a \rho^b f(v_{rel}) \quad (2.26)$$

where:

$$q_r: \text{radiative heat rate} \left[\frac{\text{W}}{\text{cm}^2} \right]$$

$$r_n: \text{vehicle effective nose radius [m]}$$

$$C = 4.736 \times 10^4$$

$$a = \begin{cases} 0.6 & 0.3 \leq r_n < 2 \\ 0.5 & 2 \leq r_n \leq 3 \end{cases}$$

$$b = 1.22$$

and the function $f(v_{rel})$ is tabulated in Reference [16] for the Earth atmosphere.

The total heat rate is simply the sum of the convective and radiative heat rates. Heat load is obtained by integrating the combined heat rate over time. Note that these expressions are intended to approximate stagnation point heating only; they do not address the heating distribution over the entire heat shield. In addition, the models do not include the effects of heat soak or irradiation during the skip portion of the trajectory.

2.8 Simulation Initialization

The trajectory generated by the equations of motion depends only on the initial vehicle state at Entry Interface (EI) and the bank angle commands. Bank angle commands are provided to the simulation by the guidance subroutine, but the vehicle state at EI is specified by the simulation initialization routine. Since the CEV is being used as the test case for the reentry guidance algorithm, vehicle characteristics and EI conditions are taken from the CEV design and mission profile.

2.8.1 Initial Position

By the definition of the inertial reference frame (see above), the initial position vector can be defined in terms of latitude, longitude, and altitude. The direction of that vector is calculated from latitude and longitude as in Equation 2.27 and the magnitude can be calculated from the altitude as in Equation 2.28.

$$\hat{r}_i = \begin{bmatrix} \cos(lon) \cdot \cos(lat) \\ \sin(lon) \cdot \cos(lat) \\ \sin(lat) \end{bmatrix} \quad (2.27)$$

$$r_i = h + r_e \quad (2.28)$$

The oblateness of the Earth is taken into account for the position calculation by allowing r_e to vary with latitude, as shown in Equation 2.29:

$$r_e = \frac{r_{e-polar} \cdot r_{e-equatorial}}{\sqrt{r_{e-polar}^2 \cdot \cos^2(lat) + r_{e-equatorial}^2 \cdot \sin^2(lat)}} \quad (2.29)$$

where:

$$r_{e-polar} = 6,356,750 \text{ meters}$$

$$r_{e-equatorial} = 6,378,140 \text{ meters}$$

Entry Interface (EI) is defined to occur at an altitude of 400,000 ft., which corresponds to an altitude of 121.92 km. For simplicity, an initial position of zero latitude and zero longitude has been chosen. Thus, the initial position vector is:

$$\vec{r}_i(t_0) = \begin{bmatrix} 6500060 \\ 0 \\ 0 \end{bmatrix} \text{ meters}$$

2.8.2 Initial Velocity

The CEV is expected to travel both to Low Earth Orbit (LEO) and the moon. NASA has required that the CEV be capable of making all its landings at sites within the continental United States [17]. With careful planning of the deorbit phasing, this can be achieved for LEO return cases using short range reentry trajectories. Because of the declination of the Moon's orbit about the Earth, a worst-case lunar return phasing could place the antipode and the resulting EI location in the southern hemisphere. This means that the landing in the United States will require a long-range skip trajectory. This, in combination with water-only restrictions on the approach flight, could require the guidance algorithm to support target ranges of up to 10,000 km [17]. For that reason, a lunar return trajectory has been chosen as the nominal case for use with the simulation. This corresponds to a velocity magnitude in inertial space of:

$$v_i(t_0) = |\vec{v}_i(t_0)| = 11,032 \text{ m/s}$$

The actual direction of the velocity can be varied, and is described in terms of an azimuth and flight path angle. Azimuth (A_z) is the angle of the velocity heading from local north, with positive angles going eastward. Flight path angle (γ) is the angle between local horizontal and the velocity vector, with positive angles going in the direction of the position vector.

The direction of the velocity vector can then be calculated based on these parameters and the initial position, as in Equation 2.30.

$$\hat{v}_i = \left(\hat{r}_i \times (\hat{k}_i \times \hat{r}_i) \cdot \cos(Az) + (\hat{k}_i \times \hat{r}_i) \cdot \sin(Az) \right) \cdot \cos(\gamma) + \hat{r}_i \cdot \sin(\gamma) \quad (2.30)$$

For simplicity, an initial azimuth of 90 degrees was chosen; this corresponds to an equatorial orbit with the vehicle traveling eastward. In contrast, the choice of flight path angle is very important and cannot be chosen so easily. If the flight path angle is too steep, the vehicle will experience excessive acceleration forces. On the other hand, if the flight path angle is too shallow, the vehicle will ‘bounce’ off the atmosphere. The window of acceptable flight path angles is actually very small, on the order of 1.5 degrees wide. For the CEV, this window, or ‘corridor,’ ranges from an angle of -4.9 deg. to -6.3 deg. After much study and consideration, a nominal reentry flight path angle of -5.9 deg. was chosen.

Applying the choice of azimuth, flight path angle, and position to Equation 2.30 yields the following initial velocity vector:

$$\bar{v}_i(t_0) = \begin{bmatrix} -1154 \\ 10974 \\ 0 \end{bmatrix} \text{ m/s}$$

2.8.3 Target Location Specification

In order to evaluate guidance performance, it was necessary to specify a target location in terms of the downrange distance to the target (d_{target}) from EI. This is referred to in this thesis as ‘target range’ and is defined as the downrange distance along the surface of the Earth between the initial (EI) position and the position of the target at the start of reentry. Since the initial orbit is equatorial and guidance performance is being analyzed in terms of downrange capability only – as described in Chapter 3 – the target location is therefore assumed to be on the equator. Guidance requires the target location in terms of the target’s initial inertial position vector. The magnitude of this vector will be the radius of the Earth (r_E), and the direction of the vector can be calculated by rotating $\hat{r}_i(t_0)$ toward $\hat{v}_i(t_0)$ by an angle θ , where θ is defined in Equation 2.31 and is shown in Figure 2.3.

$$\theta = \frac{d_{\text{target}}}{r_E} \text{ [rad]} \quad (2.31)$$

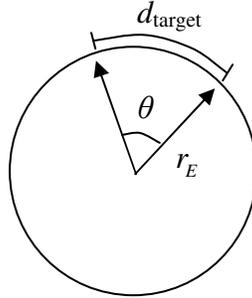


Figure 2.3: Target Location Specification

2.8.4 Vehicle Properties

The following values were used for mass and reference area:

$$m = 9500 \text{ kg}$$

$$S = 23.8 \text{ m}^2$$

The mass was rounded off from NASA's air bag landing solicitation, which cites a CEV of 'approximately 21,000 pounds' (9,545 kg) [18]. The reference area was calculated by assuming a circular heat shield with a diameter of 5.5 meters, which was specified by NASA's System Requirements Document (SRD) [17]. Thus, the CEV is larger and heavier than the Apollo Command Module (CM), which measured approximately 4 meters in diameter and weighed in at about 5,500 kg to 5,900 kg, depending on the mission.

The set of trimmed aerodynamic properties used for this thesis is derived from the Apollo command module's aerodynamic properties. This is also in accordance with NASA's SRD. However, the location of the vehicle's center of gravity (CG), which drives vehicle L/D, is not specified by NASA. For this thesis, the nominal CG location was chosen in order to obtain a trim L/D of approximately 0.35 in the hypersonic region, a value which falls halfway in the range of typical L/D values for a capsule (0.3 to 0.4). The vehicle's aerodynamic data, including coefficients of drag, lift, and moments, was

available as a function of angle of attack and Mach number. The trim angle of attack – the angle of attack for which the vehicle would experience no restoring moments about the CG chosen – was tabulated as a function of Mach number, as shown in Table 2.1. Thus, the coefficients of lift and drag also vary as a function of Mach number. For Mach numbers which fall between data points, a linear interpolation is performed to determine angle of attack and aerodynamic coefficients.

Table 2.1: Variation of Aerodynamic Properties with Mach Number

| Mach # | Angle of Attack (deg.) | C_D | C_L | $\frac{L}{D}$ |
|--------|------------------------|---------|---------|---------------|
| 0.5 | 17.43 | 0.86879 | 0.32106 | 0.37069 |
| 0.7 | 17.54 | 0.92765 | 0.26019 | 0.28153 |
| 0.9 | 22.36 | 1.0116 | 0.35029 | 0.34744 |
| 1.1 | 27.97 | 1.1366 | 0.51974 | 0.4587 |
| 1.2 | 29.22 | 1.171 | 0.54683 | 0.4675 |
| 1.5 | 29.67 | 1.2414 | 0.57913 | 0.46673 |
| 2 | 29.45 | 1.2282 | 0.53645 | 0.43718 |
| 2.4 | 29.2 | 1.1808 | 0.5186 | 0.43994 |
| 3 | 28.65 | 1.1525 | 0.51709 | 0.44983 |
| 4 | 27.2 | 1.1444 | 0.50069 | 0.439 |
| 6 | 26.68 | 1.1651 | 0.47066 | 0.40544 |
| 10 | 25.94 | 1.1886 | 0.46326 | 0.39078 |
| 18 | 24.18 | 1.2307 | 0.44825 | 0.36513 |
| 25 | 23.68 | 1.2446 | 0.4376 | 0.35288 |
| 32.2 | 23.22 | 1.2507 | 0.43513 | 0.34943 |

Due to its CG location, the CEV experiences a slightly higher angle of attack than the Apollo CM, whose trimmed aerodynamic properties are described in Reference [11]. The CEV high-mach L/D of 0.35 is thus higher than the Apollo L/D of 0.30.

The guidance algorithm requires a single value for C_D , C_L , and L/D. For the purposes of guidance, the following approximate values of C_D and C_L are used:

$$C_D \approx 1.25$$

$$C_L \approx 0.44$$

These are the values taken from the hypersonic region, which is the situation for most of reentry, so guidance assumes them to be constant over the entire trajectory. These

values, combined with the mass and reference area, lead to the following approximate values for ballistic coefficient and lift-to-drag ratio:

$$B_n \approx 320 \frac{\text{kg}}{\text{m}^2}$$
$$\frac{L}{D} \approx 0.35$$

This ballistic coefficient is actually about the same as the Apollo CM ballistic coefficient. Of course, these values will vary slightly with Mach number in the simulation environment since the angle of attack is allowed to vary.

2.9 Simulation Termination Conditions

There are two conditions for which the simulation terminates: if the vehicle skips away from the Earth, or if the vehicle reaches the ground. Of course, the skip-away situation indicates atmospheric reentry has failed, and this would be a very serious failure mode indeed. None of the simulation trials in this thesis experience a skip-away situation; they all reach the ground successfully.

Originally, the ‘skip-away’ condition was indicated by the vehicle altitude rising above that of EI. However, with very long skip reentries it may be necessary to loft to a very high altitude. Thus, this simulation termination condition was amended to be triggered if altitude rises above EI altitude and if the velocity is above the circular satellite velocity at that altitude (7,850 m/s).

The other termination condition is if the vehicle reaches the ground (i.e. altitude = 0), in which case a comparison can be made with the target location to see how close to the target the vehicle landed. However, a real-life reentry sequence would include parachutes being deployed prior to the vehicle’s landing, in order to further slow the descent. Drogue chute deploy occurs at roughly 24,000 ft, or when the vehicle’s velocity becomes subsonic. It is not the objective of this thesis to examine the effects of parachute drift, so parachute dynamics have been omitted from this simulation. Thus, the results of the vehicle’s guided reentry are referred to as being ‘at chute deploy and projected to the surface.’

Chapter 3

Guidance Algorithm Metrics

3.1 Overview

Several metrics are employed to assess the baseline algorithm's performance, identify upgrades to the algorithm, and evaluate the enhanced algorithm's improvement over the baseline version. A robust guidance algorithm has two objectives: to perform well in simulation under assumed uncertainties, and to carry as much margin as possible for unforeseen events. Metrics addressing both objectives have been employed in this study. The details of the calculations used to generate these metrics are included in Appendix D.

3.2 Performance-Based Metrics

All of the performance-based metrics in this study rely upon the method of Monte Carlo analysis. In such an analysis, a guided reentry is simulated many times, but each time certain parameters are assigned a slightly different value. The perturbation of the chosen parameters is designed to mimic the uncertainties that would be encountered in real life. By running a large number of these trials, the typical behavior of the guidance algorithm, in the presence of uncertainties, can be determined. The results of each trial individually can give insight into how well the algorithm performs under certain, specific conditions, but the conglomeration of all trials can provide a statistical statement of how well guidance might be expected to perform overall, in real life situations.

In the Monte Carlo simulations performed for this study, the parameters perturbed were velocity, azimuth, and flight path angle at EI; atmospheric density; and mass, CG location, and aerodynamic coefficients (C_D, C_L) for the vehicle. The details of how these perturbations are assigned are described in Appendix C. Characteristics of interest include statistics describing landing precision, g-loading, and aeroheating. Landing precision for a variety of different downrange target distances can be accumulated to determine a precision landing footprint.

3.2.1 Precision Landing Footprint

In usual terminology, a vehicle's footprint is a description of which downrange and crossrange landing points the vehicle is capable of reaching in the presence of assumed uncertainties. For the purpose of this thesis, the 'precision landing footprint' will be the portion of the vehicle's footprint for which precision landing is possible. This study is primarily concerned with extending downrange capability, so the landing locations are defined entirely in terms of downrange distance from EI, and target crossrange is assumed to be zero. Thus, the precision landing footprint may be specified in terms of the shortest and longest downranges for which the vehicle can achieve precision landing. The larger the spread of target ranges that show acceptable landing precision (as defined below), the more comprehensive the algorithm is. It is the objective of this thesis, as stated in Chapter 1, to show that the 2,400 km to 10,000 km spread is contained within the CEV's precision landing footprint.

In order to determine the precision landing footprint, a Monte Carlo analysis is performed for each of a variety of target downranges. It is necessary to test intermediate ranges, not just the minimum and maximum ranges, in order to achieve a high level of confidence that all branches of the algorithm logic work, and that there are no 'holes' in the algorithm – that is, to show that the algorithm works for the entire spread of ranges.

In this thesis, five representative ranges were chosen: 2,400 km, 3,500 km, 4,600 km, 7,300 km, and 10,000 km. The 2,400 km and 4,600 km ranges were chosen because they were also the approximate minimum and maximum ranges, respectively, required for the Apollo vehicles [11]. The 2,400 km range is representative of a direct reentry, which does not require a skip. The 4,600 km range requires a substantial skip, and is just at the edge of needing to exit the atmosphere, as defined in Chapter 4; that is, depending on the perturbed Monte Carlo parameters, some reentry situations do exit, and some do not. It was decided to insert a range halfway between these two ranges – 3,500 km – to represent those trajectories which require just a small skip. The longest range, 10,000 km, was mandated by NASA for the CEV [17], and it requires a very large skip, spending up to 13 minutes out of the atmosphere (this is about half of the total reentry time.) The remaining range, 7,300 km, falls halfway between 4,600 km and 10,000 km, and represents those trajectories which spend some time outside the atmosphere, but this time does not dominate the duration of reentry.

At each target range, results are analyzed to determine whether that range has acceptable landing precision. A definition of ‘acceptable’ landing precision is presented below.

3.2.2 Landing Precision

After a set of trials is run, the landing error is determined for each trial. This is accomplished by differencing the final position vector (the reentry trajectory projected to the surface) with the target location vector at the time of landing, accounting for the Earth’s rotation. The magnitude of the resulting miss vector is the miss distance. The vector can also be decomposed into downrange error, which is the component of the miss vector in the plane of the original orbit, and crossrange error, which is the out-of-plane component.

The quality of the landing precision can be characterized by the probability distribution of the landing error. It is common to approximate such a probability distribution as a Gaussian, or ‘normal,’ distribution, whose center is described by the mean value (μ), and the width is characterized by the standard deviation (σ). It is also common in engineering design to place requirements on the three-sigma (3σ) value of a parameter, which, for a truly Gaussian distribution, will be met 99.7% of the time. Thus, for the purposes of this thesis, the downrange and crossrange errors are reported separately as $\mu \pm 3\sigma$. The magnitude of μ is a measure of accuracy and is not very important; as long as μ is fairly small (i.e. on the order of a few km) the landing distribution can be centered on the landing site by biasing the target by an equal amount in the other direction. The value for 3σ , on the other hand, is a measure of precision; a smaller value of 3σ indicates that the trials are clustered closer together, and thus the landing precision is better.

In some cases it is necessary to describe the landing precision with a single value. This can be done using the concept of a Circular Error Probability (CEP). The CEP is the radius of the circle within which a certain percentage (usually 50%) of trials would fall. For reasons described above, this thesis will use a 99.7% CEP; for a 250-trial population size, the 99.7% CEP is the maximum landing error from all the trials. In general CEP values are usually calculated based on landing errors with respect to the target point. However, this thesis is concerned more with precision and repeatability. Since the landing distribution can be re-centered by biasing the commanded target location, this thesis uses a CEP calculation taken with respect to the mean landing location.

The distribution of the landing errors can also be visualized using a scatter plot of crossrange errors vs. downrange errors, as in Figure 3.1.

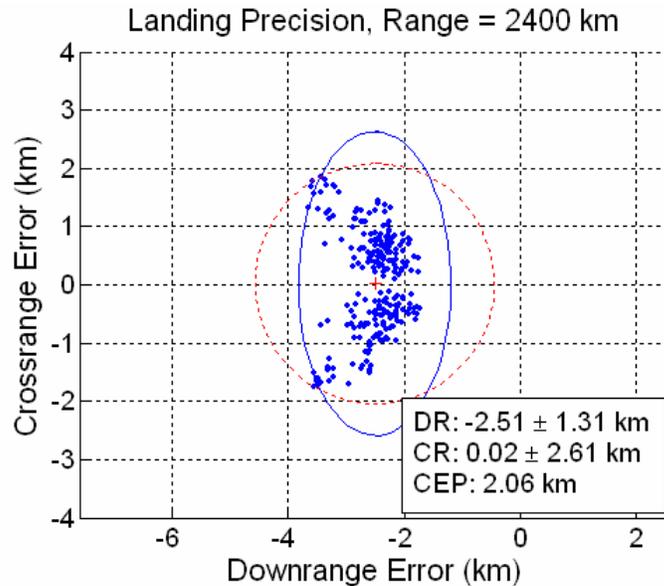


Figure 3.1: Sample Landing Error Distribution

In this plot, each trial's landing position is represented by a dot. A positive downrange error indicates that the vehicle overshot the target, whereas a negative error indicates undershoot. The ellipse in this plot is the 'landing ellipse' and is representative of the 3σ values for downrange and crossrange error. The center of the ellipse is located at the mean values for downrange and crossrange landing error. The radius of the ellipse in the horizontal direction is the 3σ value for downrange error and the radius of the ellipse in the vertical direction is the 3σ value for crossrange error. Thus, 99.7% of the trials would be expected to fall within this ellipse if the landing error were truly a Gaussian distribution. The dotted circle in Figure 3.1 represents the 99.7% CEP, and it is also centered on the mean landing location. This circle would be expected to encompass 99.7% of the trials for a Monte Carlo of infinite population size. For the 2,400 km case shown by Figure 3.1, CEP of 2.06 km was achieved. The mean downrange error of 2.51 km undershoot can be removed by asking guidance to aim for a target range of 2,402.5 km, rather than 2,400 km.

With this notation, it is possible to answer the question of what is 'acceptable' landing precision. NASA has mandated that the total landing error may not be more than 5 km total, including errors from guidance and from parachute drift [17]. As an initial starting

point, it would seem reasonable to assign a preliminary error allocation which allows for equal error between guidance and parachute drift. Assuming that guidance errors and parachute drift are independent, total error would be calculated as the RSS of the two CEP values:

$$CEP_{tot} = \sqrt{(CEP_{guid})^2 + (CEP_{para})^2} < 5 \text{ km} \quad (3.1)$$

Assuming the two errors are equal and solving:

$$CEP_{guid} = \sqrt{\frac{(CEP_{tot})^2}{2}} < \sqrt{\frac{(5 \text{ km})^2}{2}} = 3.5 \text{ km} \quad (3.2)$$

Substituting a value of 5 km for $3\sigma_{tot}$ yields a value of 3.5 for $3\sigma_{guid}$. Accordingly, the following definition of ‘acceptable’ landing precision has been adopted by this thesis: a target range for which the Monte Carlo results in a CEP of less than 3.5 km is said to have acceptable landing precision. Thus, the example in Figure 3.1 does show acceptable landing precision.

3.2.3 G-loads

G-loads are caused by aerodynamic acceleration on the vehicle, and they are an important consideration with respect to astronaut safety. The g-load at any given point in time is described by the magnitude of the aerodynamic acceleration – the sum of lift and drag forces – reported in units of g’s. One g is equivalent to the acceleration of gravity at the surface of the Earth; this is about 9.8 m/s^2 or $32.2 \text{ feet per second squared}$.

The maximum g-load experienced during reentry is of interest, because g-loads which are too high can cause damage to the human body or the vehicle structure. The maximum g-load from each trial can be accumulated into statistics for the whole Monte Carlo. This data can be represented visually by a histogram, and mathematically by $\mu \pm 3\sigma$, as described above and as shown in Figure 3.2.

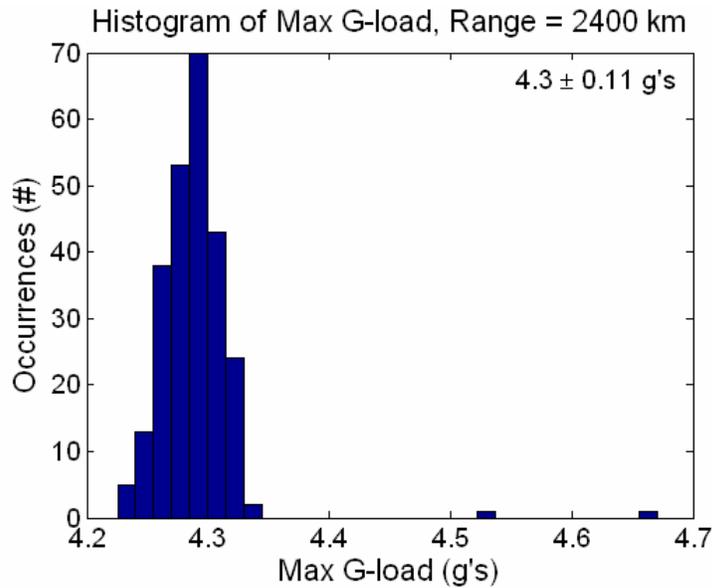


Figure 3.2: Sample G-Load Histogram

Another concern is the duration-based g-load analysis. The human body is not designed to endure elevated g-loads (i.e. above 1 g) for long periods of time. For that reason, it is necessary to examine the duration for which elevated g-loads are experienced. NASA has developed a guideline for this matter, and it is reported as a plot of acceptable g-loads as a function of duration. There are actually multiple guidelines: one for deconditioned, ill, or injured crew; one for nominal ascent and entry operations; and one for the maximum allowable conditions. These guidelines are described in the NASA-STD-3000 document [19] and are shown in Figure 3.3.

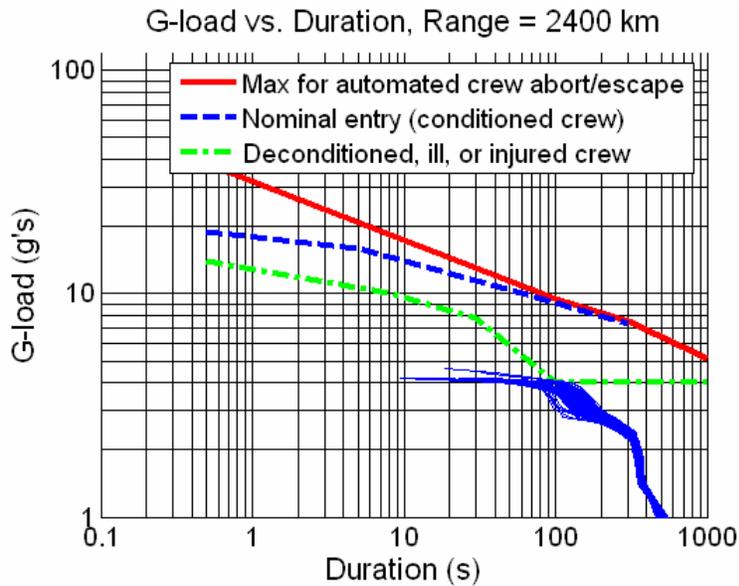


Figure 3.3: NASA STD-3000 G-Load Guidelines and Sample Duration-Based Monte Carlo Results

Also shown in Figure 3.3 is a cluster of blue lines representing sample results from a Monte Carlo analysis. Each line represents one trial, and the line is created by connecting a series of dots, one for each g-load level (i.e. 1 g, 2 g's, 3 g's, etc.) up to the maximum g-load level. Each g-load level is matched with its duration by adding up the amount of time that aerodynamic acceleration is above that particular g-level for that particular trial. The lower the g-loads are overall, the lower on the plot the cluster of blue lines will be. If part of the cluster crosses above one of the guidelines, that guideline has been violated. In this example, none of the guidelines has been violated, although some trials just barely meet the 'injured crew' constraint.

3.2.4 Aeroheating

Aeroheating is an important consideration with respect to the heat shield on the blunt side of the capsule. The kinds of material that can be used are restricted by the peak heating rate experienced by the vehicle, and the thickness of the heat shield follows from the total heat load. In general, it is desirable to have a low heat rate and also a low heat load. In reality, however, there is often a tradeoff between heat rate and heat load. For example, heat load may be reduced by decreasing the time to reach the surface, but that makes it necessary to increase the descent rate, thereby increasing the heat rate.

Peak heat rate and total heat load are reported in the same manner as maximum g-loads, in terms of $\mu \pm 3\sigma$ and a histogram, as in Figure 3.4 and Figure 3.5.

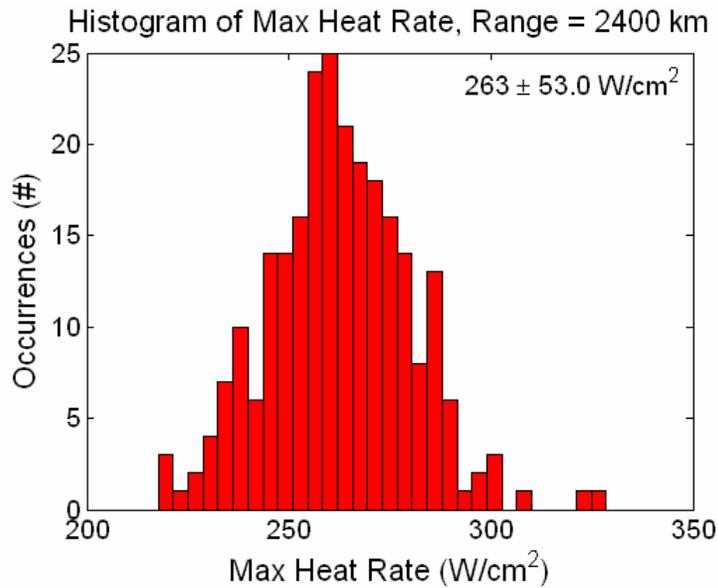


Figure 3.4: Sample Peak Heat Rate Histogram

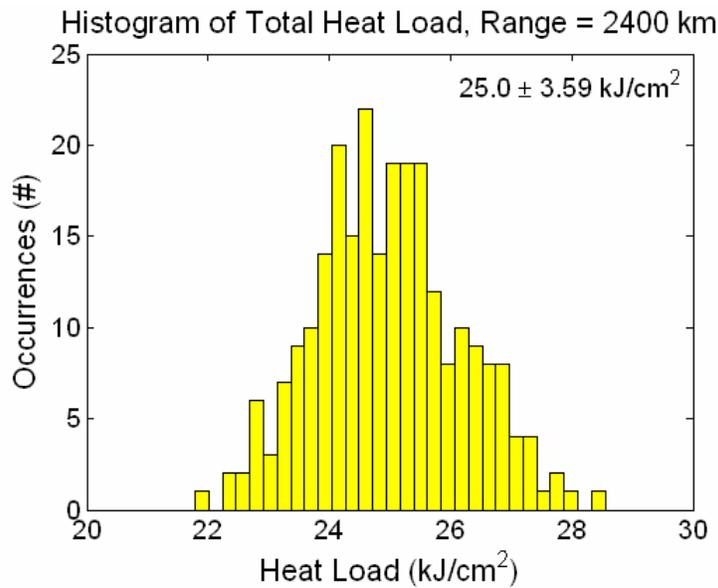


Figure 3.5: Sample Total Heat Load Histogram

3.3 Margin-Based Metrics

These metrics were developed to evaluate the amount of margin available during a reentry trajectory to deal with unforeseen uncertainties. Each of these metrics can be

applied to an individual trial, or evaluated collectively for an entire Monte Carlo ensemble.

3.3.1 Vertical L/D Fraction

The shape of the trajectory, and thus the resulting landing location is a function of the bank angle command history – the direction of the lift vector as a function of time. Downrange is controlled by the amount of lift in the vertical direction, and crossrange is controlled by the amount in the horizontal direction. Thus, the vehicle’s capability to reach the target downrange can be evaluated by examining the amount of lift in the vertical direction. One metric based on this idea can be calculated as shown in Equation 3.3 and is referred to in this thesis as the ‘vertical L/D fraction’ or simply ‘L/D fraction.’

$$Frac_{L/D} = \frac{L/D_{vertical}}{L/D_{vehicle}} = \cos(\phi) \quad (3.3)$$

where:

ϕ : bank angle

As its name suggests, the vertical L/D fraction describes the fraction of the total vehicle L/D which is pointed in the vertical direction. For example, a vehicle with a total L/D of 0.4 and an effective L/D of 0.2 in the vertical direction would have an L/D fraction of 0.5 and a bank angle of 60 degrees as shown in Equations 3.4 and 3.5.

$$Frac_{L/D} = \frac{0.2}{0.4} = 0.5 \quad (3.4)$$

$$\phi = \cos^{-1}(0.5) = 60 \text{ deg.} \quad (3.5)$$

This metric can be useful because the results are fairly intuitive. An increasing L/D fraction means that the amount of lift in the vertical direction is increasing. Since an L/D fraction of 0 corresponds to neutral lift (lift vector entirely in the horizontal direction), a positive value for L/D fraction indicates the lift vector is pointing somewhat upward whereas a negative value indicates the lift vector is pointing downward. At the extremes,

an L/D fraction of 1 corresponds to a bank angle of 0 deg. (full lift up), whereas an L/D fraction of -1 indicates a bank angle of 180 deg. (full lift down).

The L/D fraction can be calculated from either the bank angle command, or the actual bank angle, which includes the effects of crossrange control. The more useful one turns out to be the one derived from the commanded bank angle, because it gives a little more insight into the internal workings of guidance, without being cluttered up by bank angle reversals, which are a method of crossrange control which is discussed in Chapter 4. Figure 3.6 shows an example of the commanded L/D fraction as a function of time during a single reentry trajectory.

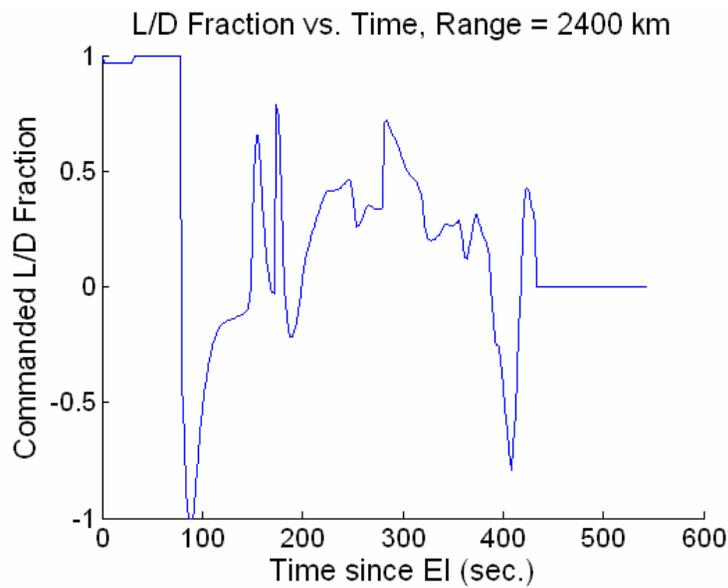


Figure 3.6: Sample Trajectory L/D Fraction vs. Time

From this plot one can ascertain, for example, that the vehicle was commanded to maintain a full lift up attitude for the first 70 seconds of reentry and spent the last 100 seconds or so with neutral lift.

L/D fraction is a useful metric because it indicates when control authority has been saturated. When the value is 1, the vehicle is in a full lift up orientation and cannot achieve any more lift in the vertical direction. This typically occurs when range error corrections are beyond the vehicle's capability; the result is that the vehicle will fall short of its current target. Similarly, an L/D fraction of -1 indicates that the vehicle will overshoot its current target.

3.3.2 Final Phase Energy Bucket and Transition

The ‘Final’ phase of the guidance algorithm is, as its name suggests, the last phase of the algorithm, which guides the vehicle in the last leg of its trajectory to the ground. It is based on a reference-following controller and is described in more detail in Chapters 4 and 5. The ‘Final phase energy bucket’ is a set of conditions during the Final phase for which the vehicle has enough control authority to reach the target landing site. This set of conditions is expressed in terms of a plot of energy/weight vs. range to go, an idea which was also used in Space Shuttle guidance. Energy/weight is calculated as in Equation 3.6:

$$E/W = h + \frac{1}{2}v^2 / g \tag{3.6}$$

where:

- E/W : energy/weight (m)
- h : altitude (m)
- v : inertial velocity (m/s)
- g : acceleration of gravity near Earth's surface: 9.81 m/s

An example of the energy bucket for various target ranges is shown in Figure 3.7.

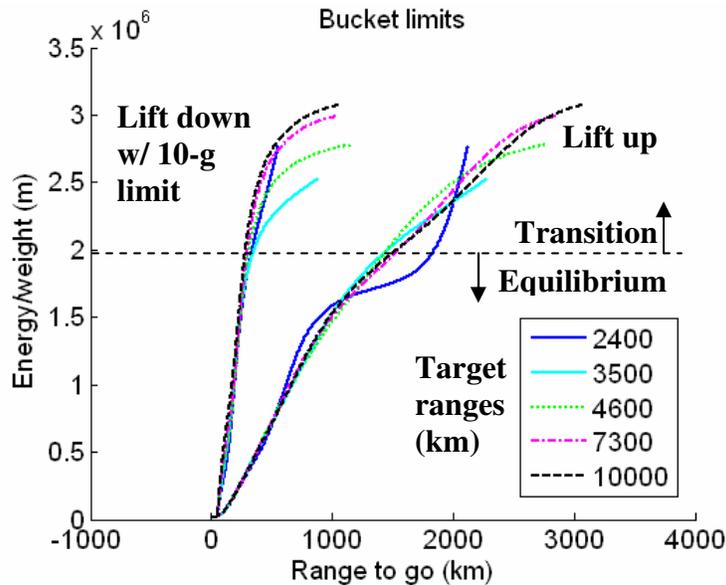


Figure 3.7: Final Phase Energy Bucket

The bounds of the energy bucket are determined by the vehicle's aerodynamic capability. The steepest descent is produced by a full-lift-down orientation with a g-limit, and is the left set of energy/weight contours in the figure. Similarly, the maximum range is produced by full lift up, and is the right set of energy/weight contours in the figure. Together, these bounds are referred to as the 'bucket.'

The high energy portion of this bucket corresponds to the first part of the Final phase, with high velocity and at high altitudes. For this part of the bucket, the bucket's bounds depend on the initial conditions of the phase, which vary according to the target range. This portion of the bucket transitions into equilibrium flight at lower altitudes and lower velocities, where the bucket's bounds are invariant of the target range. Note that the waviness of the 2,400 km lift-up bound is an artifact of the way the bucket is generated; in reality that line should follow the same path as seen with the other target ranges.

A vehicle within these bounds has sufficient control authority to reach the target, whereas a vehicle outside these bounds does not. A robust algorithm should not place any additional limits on the vehicle capability, so it should be able to guide to the target from any starting point within the bucket limits. The farther from either edge the vehicle is, the more control authority margin the vehicle has. The most margin can be obtained by following the center of the bucket.

It is the responsibility of the guidance phases prior to the Final phase to guide the vehicle into the bucket, so the concept of the energy bucket can be used to evaluate the performance of those phases. This can be an indicator of not only whether the phases were successful (did the vehicle transition into the bucket or not?) but also how much margin is available at the start of the Final phase (how close to the centerline of the bucket is the vehicle?) Those phases should be designed to maximize this margin, and thus should target the center of the bucket.

The energy bucket can also be used to evaluate the performance of the Final phase. Margin can be tracked throughout the Final phase by assessing how far from the center of the bucket the trajectory deviates. The Final phase should be designed to follow the center of the bucket as closely as possible, in order to maximize margin.

Figure 3.8 and Figure 3.9 show examples of two sets of Monte Carlo simulations. The top plot in each set shows both the points within the bucket where each trajectory

transitioned into the Final phase, and the points which were being targeted. How well the actual transitions match up with the targeted points indicates the quality of the performance of the previous phases. The dotted line in this plot is the reference trajectory that is used during the Final phase; the more closely it follows the center of the bucket, the more margin it reserves. The bottom plot shows how each trajectory evolves with respect to the bucket; as time progresses, each trial inscribes a curve from the top right of the plot to the bottom left.

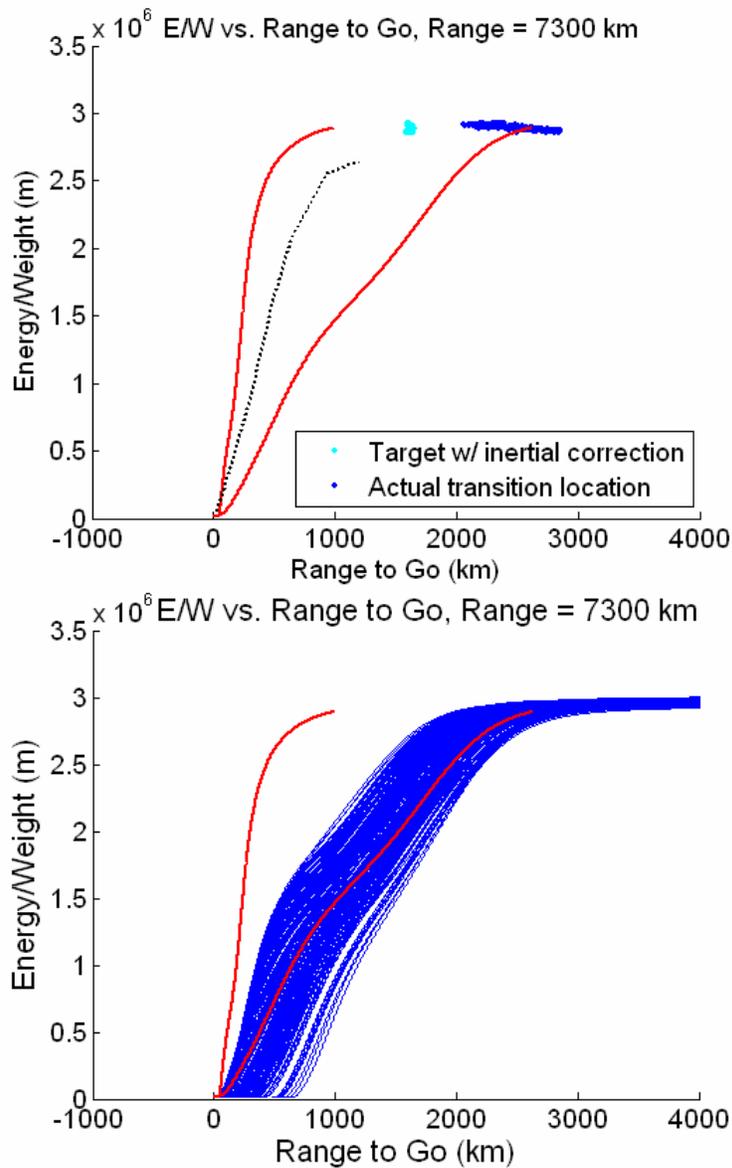


Figure 3.8: Poor Bucket Performance

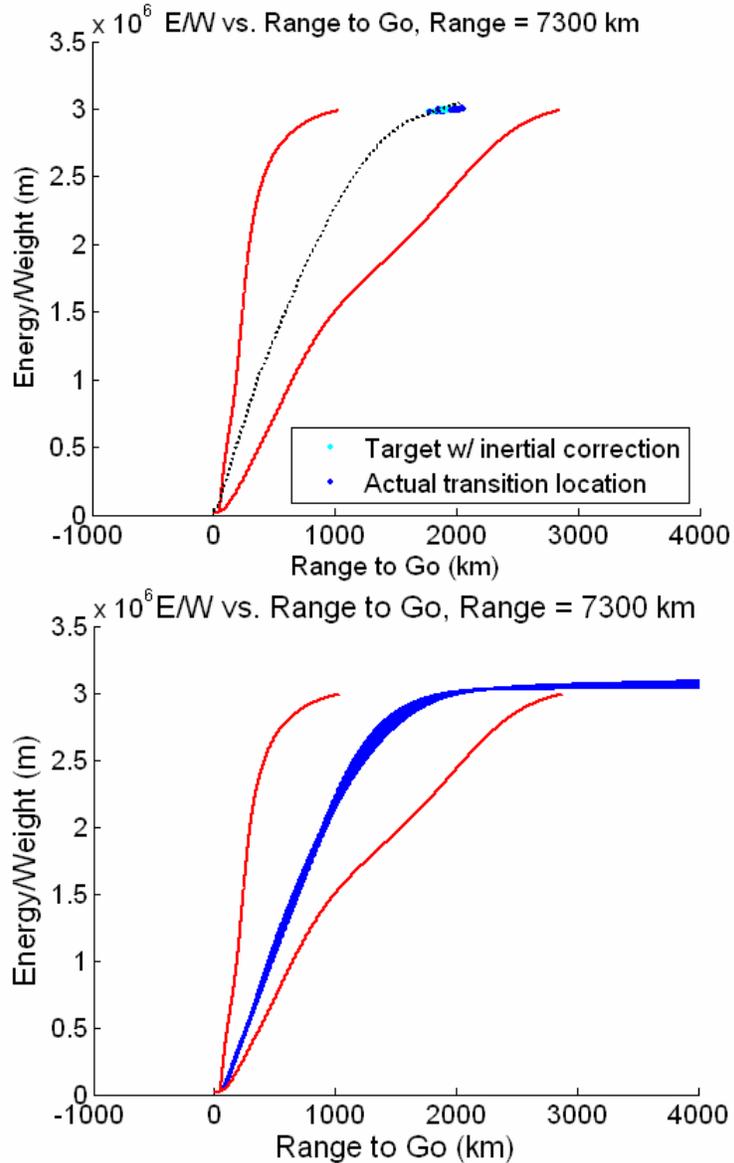


Figure 3.9: Good Bucket Performance

The guidance algorithm margin in Figure 3.8 is not very good; the transition into the Final phase for some of the trials wasn't even within the bucket, and the rest are biased toward the edge. Few of the trajectories actually follow the center of the bucket, so those that do stay within the bucket don't show much margin. The margin in Figure 3.9 is much better; the transition between phases occurs near the center of the bucket, and the Monte Carlo trials all do a fairly good job of following the reference trajectory.

Chapter 4

Baseline Guidance Algorithm

4.1 Overview

The primary function of the reentry guidance algorithm is to manage energy as the spacecraft descends to the parachute deploy interface. There are two channels in the guidance strategy: vertical and lateral. However, there is only one control: bank angle modulation. The guidance algorithm updates the bank angle command once every guidance cycle. For the original Apollo algorithm, which is described in this chapter, a guidance cycle occurs once every two seconds.

Energy, and thus downrange to the target, is managed in the vertical channel by orienting the lift vector. Full lift-up provides maximum range while full lift-down provides the steepest descent. Lift-down may be constrained by the maximum allowed g-loads that can be experienced by the crew and vehicle. Any bank orientation other than full lift-up or full lift-down will place a component of lift in the lateral channel. Guidance's primary goal is to manage lift in the vertical channel so that the vehicle enters into the wind-corrected parachute deploy box (defined by values of dynamic pressure and Mach number) at the appropriate downrange position. Crossrange position is maintained by the lateral channel by reversing the lift command into the mirror quadrant (e.g. $+30^\circ$ from vertical to -30°) once the lateral range errors to the target cross a threshold. The vehicle continues this bank command reversal strategy as it descends to the target. As the energy (velocity) decreases, the lateral threshold shrinks so that the vehicle maintains control authority to minimize the lateral errors prior to chute deploy. The guidance phases and phase-transition logic are discussed fully in Reference [9].

4.1.1 Downrange Management

Downrange management is achieved by commanding the amount of lift in the vertical channel. The baseline Apollo algorithm consists of seven phases designed to control the downrange position of the vehicle, as shown in Figure 4.1.

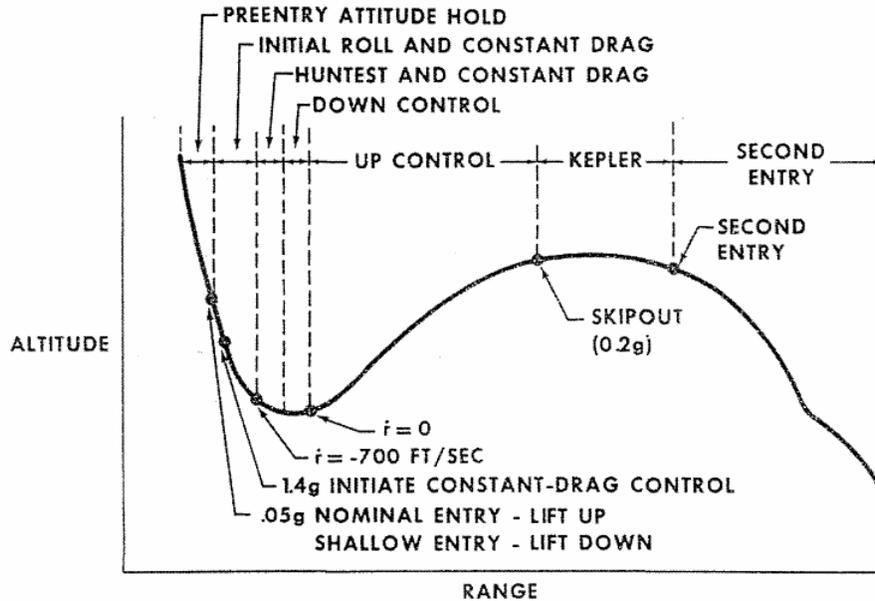


Figure 4.1: Reentry Guidance Phases [11]

The first phase is the Pre-entry Attitude Hold phase, which begins as soon as the guidance algorithm gains control of the vehicle. This phase maintains current attitude until a sensible atmosphere has been detected, at which point the algorithm begins to control the lift vector of the vehicle as part of the Initial Roll phase. This phase seeks to guide the vehicle toward the center of the reentry corridor. It does so by commanding the lift vector upward to make a steep reentry path shallower, or commanding the lift vector downward to steepen a shallow reentry.

Once atmospheric capture is assured, the Hunttest and Constant Drag phase begins. This phase maintains constant drag while “hunting” for an “estimate” of the appropriate trajectory to reach the target. Here, the algorithm determines whether the vehicle will need to perform an upward “skip” in order to extend the vehicle’s range, decides which of the possible phases to use, and calculates the conditions which will trigger those phases. For short target ranges, the Hunttest phase will determine that no skip is necessary, and the algorithm will transition directly into the Final (“Second Entry”)

phase. Otherwise, if a skip is necessary, guidance will transition into the Downcontrol phase.

Downcontrol guides the vehicle to the trigger conditions, using a constant drag policy, until a velocity and drag trigger previously determined by Hunttest activates the Upcontrol phase. Upcontrol guides the vehicle along a self-generated reference trajectory, previously generated by the Hunttest phase, using a reference-following controller. It is important to note that this trajectory is not updated during the Upcontrol phase. The reference trajectory is indicated in terms of velocity and altitude rate reference variables, which are functions of the independent variable, aerodynamic drag. If the vehicle does not exit the atmosphere, the algorithm will enter the Final phase at the peak of the skip. Otherwise, the Ballistic (“Kepler”) phase will take over. For the purposes of the Apollo algorithm, atmospheric exit is defined to take place when aerodynamic acceleration drops below 0.2 g’s.

The Ballistic phase is assumed to be simply a ballistic trajectory, and since atmospheric density is very low, it is assumed that there is not enough dynamic pressure for the vehicle to have enough control authority to make steering effective. Thus, bank angle commands are no longer updated and remain at the previously commanded value until aerodynamic acceleration rises above 0.2 g’s again, at which point the Final phase begins. Like the Upcontrol phase, the Final phase is based on a reference-following controller. Unlike the Upcontrol phase, however, the reference is based on a stored nominal trajectory which was calculated preflight, and is indicated in terms of drag and altitude rate as functions of velocity. Once the velocity drops below a threshold value, the algorithm stops updating bank commands until chute deploy, when the guidance algorithm would be disabled. Since chute deploy is not modeled in the simulation used for this thesis, a neutral bank command is adopted after steering shutoff for projection to the ground.

4.1.2 Crossrange Management

The component of lift in the vertical channel, and thus the component of lift in the lateral channel, is dictated by the downrange management phases. Lateral control can still be achieved, however, by periodically rotating to the mirror quadrant, as in Figure 4.2. Such a ‘bank angle reversal’ allows the same amount of lift to remain in the vertical channel

while reversing the direction of the lift in the lateral channel. In this manner, the vehicle can ‘zigzag’ its way to the target.

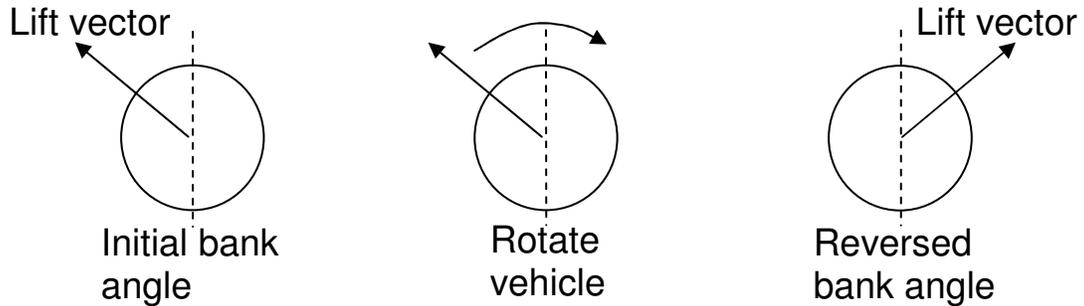


Figure 4.2: Bank Angle Reversal

In order to limit the number of reversals required during the entire trajectory, the amount of crossrange error allowed before a reversal is commanded is large at first and then decreases as the vehicle’s remaining crossrange capability diminishes. This can be seen in Figure 4.3.

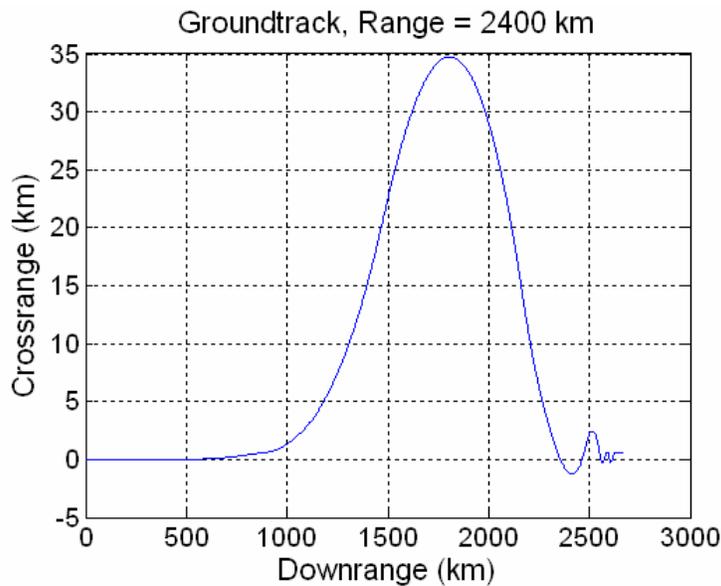


Figure 4.3: Typical Trial Groundtrack with Bank Reversals

The allowable crossrange error does not go to zero as the target approaches; otherwise, the vehicle would be making constant bank reversals toward the end of the trajectory. Instead, there is a bias term which allows about 2 km of crossrange error.

4.2 Initial Results

A Monte Carlo performance analysis of the baseline Apollo guidance algorithm was implemented. The algorithm was tested as-is; control gains and pre-defined reference trajectories were not optimized for the CEV vehicle configuration. The Monte Carlo analyses were performed for various target ranges: 2,400 km, 3,500 km, 4,600 km, 7,300 km, and 10,000 km. Figure 4.4 shows the landing error scatter plots for all target ranges.

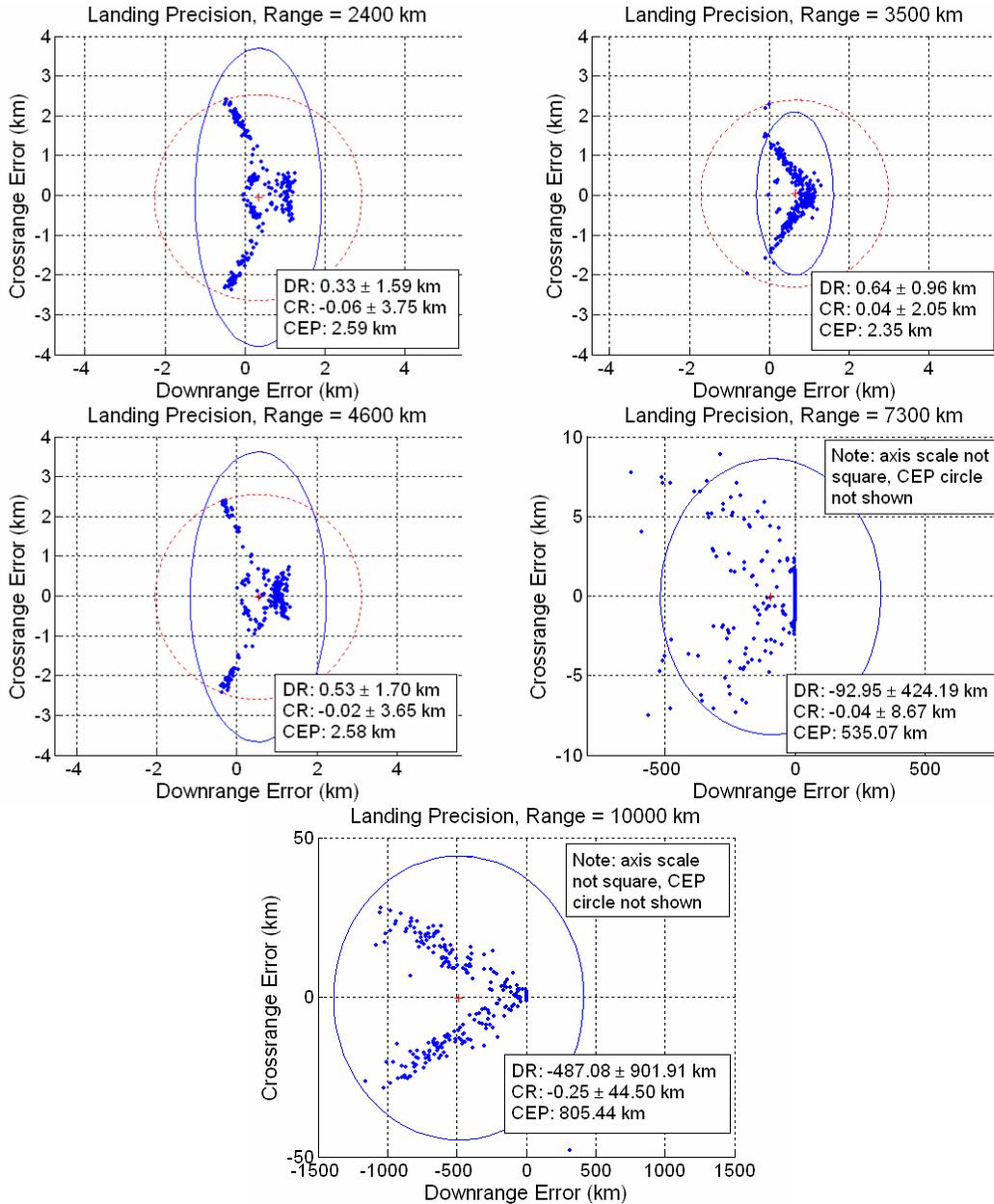


Figure 4.4: Landing Error Scatter Plots for Various Target Ranges.

For short target ranges (i.e. 2,400 km, requiring no skip) and medium target ranges (i.e. 3,500 km and 4,600 km, requiring small to medium skips) the Apollo reentry guidance performed with acceptable precision landing, achieving 99.7% Circular Error Probabilities (CEPs) of 2.59 km, 2.35 km, and 2.58 km respectively. All of these are less than the 3.5 km error allocation. However, long target ranges (7,300 km and 10,000 km, requiring a substantial skip and extended periods of time in the Ballistic phase) did not achieve acceptable precision landing, with 99.7% CEPs of 535 km and 805 km, respectively.

4.3 Analysis of Final Phase Energy Bucket

In order to determine what caused the poor performance for the long target ranges, it is logical to work backwards in evaluating each phase’s performance. We start with the Final phase; if the transition to the Final phase occurred within the energy bucket, the vehicle should have been able to reach the landing target and thus the fault would lie with the algorithm’s Final phase. Otherwise, if the transition to the Final phase occurred outside of the energy bucket, the fault lies with one of the previous phases. Figure 4.5 and Figure 4.6 show the transition locations for the 7,300 km and 10,000 km cases with respect to their energy buckets.

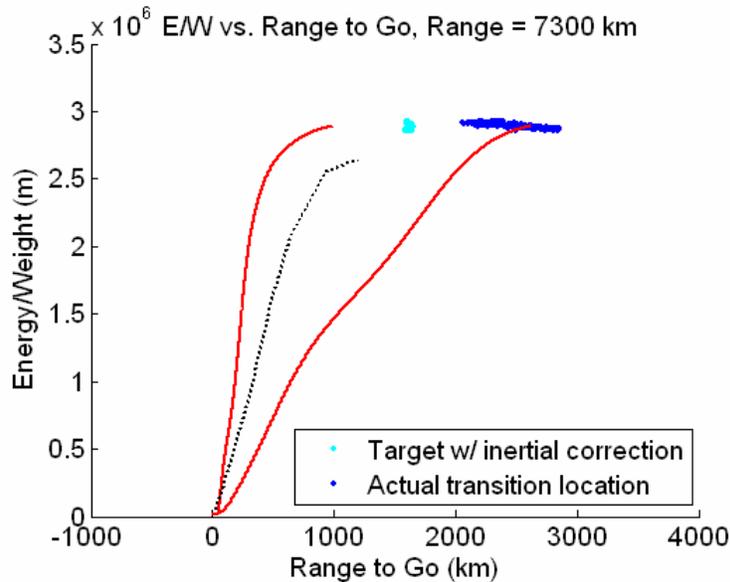


Figure 4.5: Transition into the Final Phase, Apollo Algorithm, 7,300 km

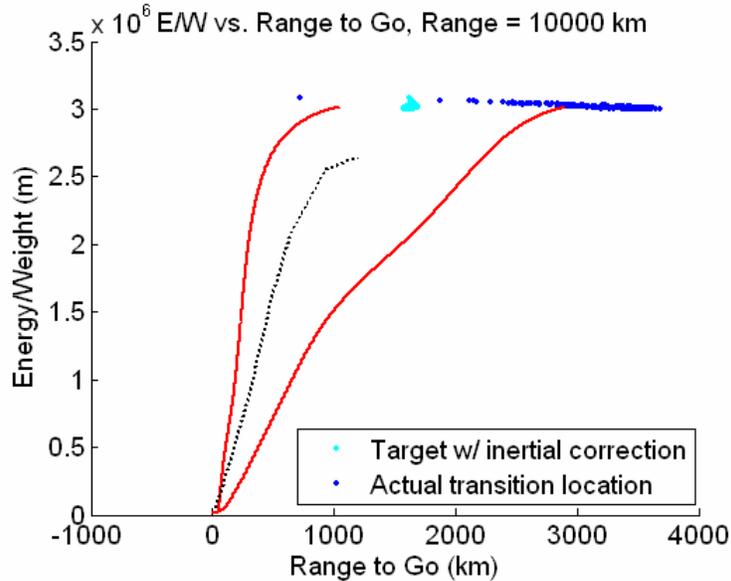


Figure 4.6: Transition into the Final Phase, Apollo Algorithm, 10,000 km

Although both cases show targets within the bucket, most of the cases undershoot the target severely. Some of the cases even fall outside the bucket, which is what causes the poor landing errors. This analysis leads to the conclusion that the previous phases do not sufficiently lead the trajectories into the Final phase energy bucket. Since no calculations or steering occur during the Ballistic phase, the next order of business would be to investigate the performance of the Upcontrol phase.

4.4 Analysis of Upcontrol Phase Targeting

In order to investigate the effectiveness of the Upcontrol phase, the bank angle history and control errors during the Upcontrol phase should be investigated. Figure 4.7 shows the L/D fraction, during the Upcontrol phase only, for all trials in each of the 7,300 km and the 10,000 km Monte Carlos.

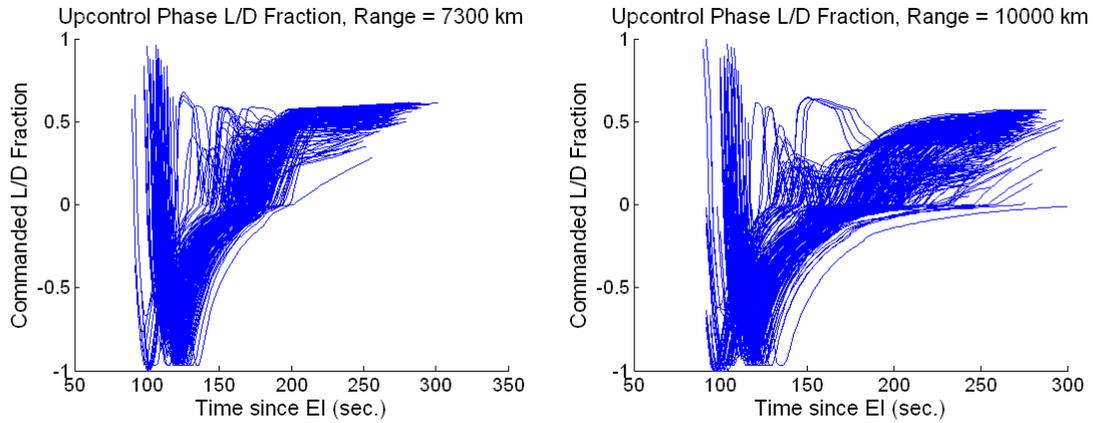


Figure 4.7: Commanded L/D Fraction during the Upcontrol Phase

Many trials fell short of the energy bucket, yet according to Figure 4.7, there was plenty of margin left in the Upcontrol phase for increasing the L/D fraction, and yet it was not used. A look at the control errors with respect to the Upcontrol reference trajectory might prove instructive.

As stated above, the reference trajectory is defined in terms of velocity and altitude rate, and thus errors are taken with respect to those variables. The control errors as a function of time are shown in Figure 4.8 and Figure 4.9.

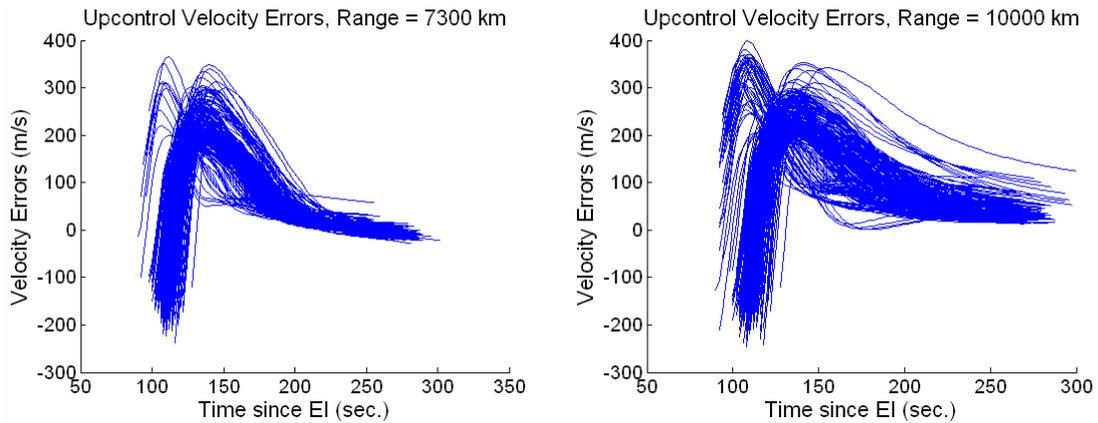


Figure 4.8: Upcontrol Velocity Errors

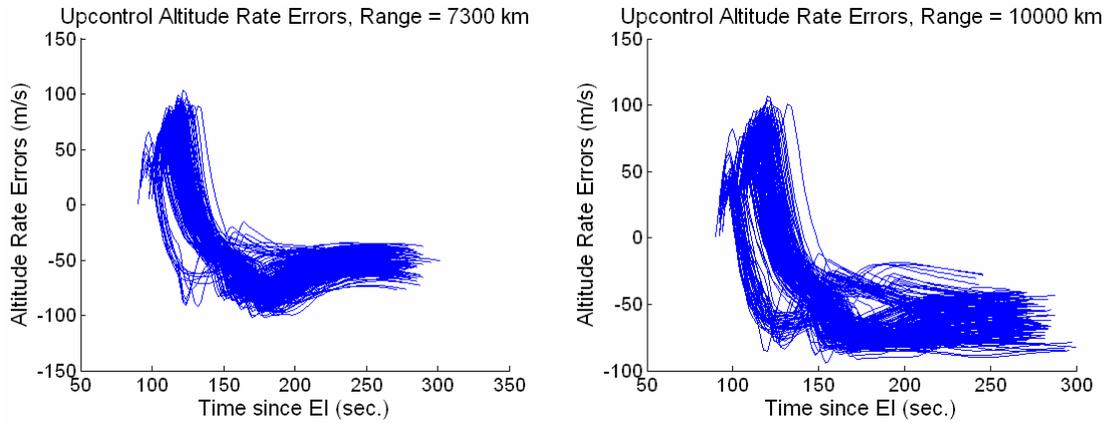


Figure 4.9: Upcontrol Altitude Rate Errors

Figure 4.8 shows that the transition to the Upcontrol phase occurs when the actual velocity is significantly lower than the reference velocity. As time progresses, the velocity then increases and overshoots the reference velocity, and then settles down to near the reference value. To give a sense of magnitude of these errors, the reference velocity of the vehicle in this time period is approximately 7,600 m/s. This means that the errors of up to 200 m/s seen at the end of the Upcontrol phase amount to less than 3% of the reference value.

Figure 4.9 shows a slightly different characteristic for the altitude rate errors. The altitude rate starts too high, crosses over the reference value and ends up too low by 50 – 100 m/s. Although these errors may seem small compared to the velocity errors, it is important to take into account the magnitude of the altitude rate reference. By the end of the Upcontrol phase, the reference altitude rate is approximately 200 m/s, meaning that the altitude rate errors at the end of the phase are 25% – 50% of the reference value. With an error of less than 3% for the exit velocity, but an error of up to 50% for the exit altitude rate, this means there is an error of up to 50% for flight path angle. Undershooting the flight path angle by so much would tend to significantly shorten the range covered during the Ballistic phase; undershoot is consistent with the results for longer target ranges (see Figure 4.4, above.)

The question remains whether this failure to hit exit conditions at the end of the Upcontrol phase is the sole reason for the failure to hit the Final phase energy bucket. The next step is to evaluate whether the exit conditions, as determined by the Hunttest phase, are appropriate.

4.5 Analysis of Exit Conditions Selected by the Hunttest Phase

The Apollo algorithm chooses exit conditions such that the downrange covered during the Ballistic phase will be sufficient to hit the energy bucket. To predict this range, Hunttest uses a purely Keplerian calculation, ignoring the presence of drag in the upper atmosphere. Depending on the exit conditions and the magnitude of the skip, this may or may not be a reasonable assumption. In order to evaluate the validity of this assumption, numerical simulations were run to determine what downrange *would have been* covered during the ballistic phase if the Upcontrol phase had accurately steered to the exit conditions indicated by the Hunttest phase. Figure 4.10 shows the range calculation errors vs. the accumulated drag calculated in the numerical simulation.

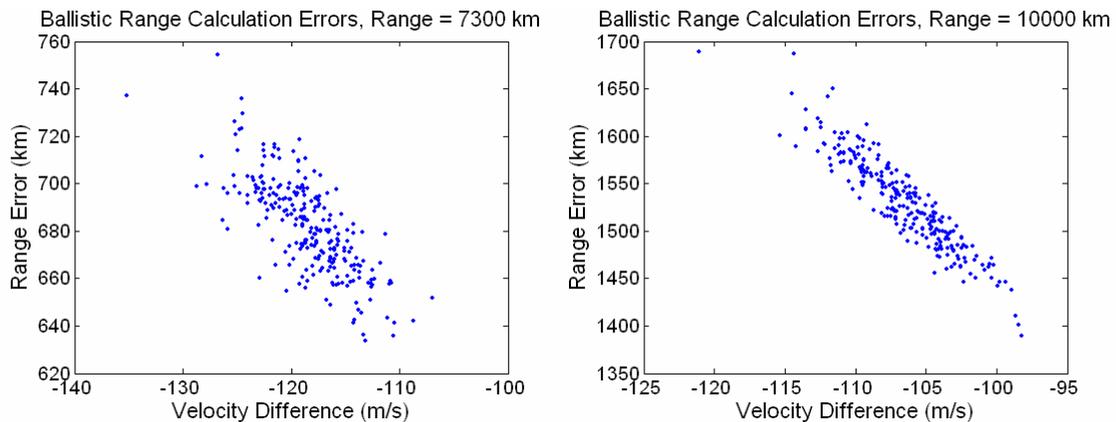


Figure 4.10: Ballistic Range Calculation Errors

This plot shows that a typical trial in the 7,300 km Monte Carlo will actually have an accumulated drag of 100 – 140 m/s, which leads to an undershoot of 600 – 750 km in downrange compared to what the Keplerian estimate (assuming no drag) would predict. The 10,000 km Monte Carlo shows even worse errors, with 95 – 125 m/s of accumulated drag leading to 1400 – 1700 km undershoot. These undershoot values are significant when compared to the width of the Final phase energy bucket. Even if the exit conditions commanded by Hunttest had been achieved, not all of the trajectories would have made it into the bucket. Thus, without taking into account the drag encountered during the Ballistic phase, Hunttest’s current method of defining target exit conditions is not sufficient to allow precision landing for such long target ranges.

Chapter 5

Enhanced Guidance Design

5.1 Overview

Analysis of the long target ranges showed that the degradation of precision landing performance was due to two issues with the Upcontrol phase. First, the vehicle was not being guided to the desired exit conditions calculated by the Hunttest phase. Second, the exit conditions calculated by Hunttest were inaccurate. These two issues were resolved by replacing the Upcontrol phase with a numeric predictor-corrector algorithm (PredGuid). An analytic predictor-corrector option was investigated but rejected for reasons described below.

In addition, the Final phase reference trajectory was redefined and extended to re-center it for the CEV, which has different vehicle characteristics than the original Apollo Command Module. The Final phase range estimation method was updated to allow the new Final phase reference trajectory to support all target ranges.

5.2 Predictor-Corrector to Replace Upcontrol Phase

Between the miscalculation of exit conditions and inability to steer to the targeted conditions, the original Apollo algorithm undershoots the desired conditions for entering into the Final phase for the 7,300 km and 10,000 km target ranges. For target ranges up to 4,600 km, there is enough margin in control authority for the Final phase to remove any trajectory errors caused by these two problems. However, the two longer target ranges experience enough undershoot that it is impossible for the Final phase to remove all the trajectory errors.

It was determined that a predictor-corrector (PC) algorithm would be a good replacement for the Upcontrol phase in order to resolve the two problems the original Apollo algorithm was experiencing: miscalculation and mis-steering. A PC algorithm could

avoid the problem of miscalculation by targeting the start of the Final phase so that the effects of drag during the Ballistic phase would be included in the range estimation process. The other problem, mis-steering, would be avoided by updating the reference trajectory at every guidance cycle based on actual conditions. In addition, the algorithm could collect and filter data about actual atmospheric conditions, in order to obtain better atmospheric knowledge and improve performance in the presence of uncertainties. Both analytic predictor-corrector (APC) and numeric predictor-corrector (NPC) options were investigated.

It was difficult to find a suitable closed form expression for use in an APC algorithm. Such an expression would need to estimate downrange traveled during the Upcontrol and Ballistic phases, taking into account the effects of drag during the Ballistic phase. Several references were found which calculated various expressions for skip trajectory state variables, including downrange, using the method of matched asymptotic expansions [20],[21]. However, these expressions included some constants of integration which are determined by trajectory conditions such as velocity, flight path angle, and altitude. The constants of integration would need to be recalculated each guidance cycle from current conditions. However, due to the nonlinear nature of the equations, the constants of integration could not be solved for in closed form – they would have to be solved numerically.

This necessity removed the primary advantage of an APC algorithm over an NPC algorithm; now the APC would experience the same computation and convergence issues as an NPC. An NPC, however, would be more accurate since it could utilize more detailed environmental models than an APC algorithm. For that reason, it was decided to use an NPC algorithm to replace the Upcontrol phase.

Rather than designing an NPC algorithm from scratch, it was decided to utilize an existing NPC. Since targeting skip trajectory conditions is in many ways similar to aerocapture targeting, the algorithm chosen for this testing phase was PredGuid, a numeric predictor-corrector aerocapture algorithm developed for the Aero-assist Flight Experiment (AFE). The algorithm is described fully in Reference [4]. It should be noted that there are many other suitable NPCs that could have been chosen – PredGuid was merely the most convenient.

5.3 The Original PredGuid Numeric Predictor-Corrector

Figure 5.1 shows a functional diagram of the tasks included in the original PredGuid algorithm, as applied to Earth aerocapture.

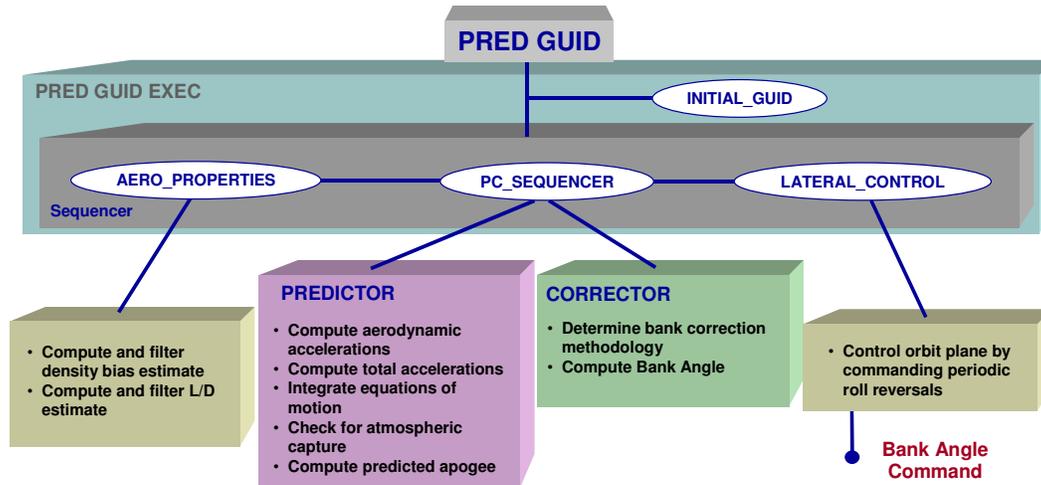


Figure 5.1: The PredGuid Numeric Predictor-Corrector

During the main phase of aerocapture (i.e. in the atmosphere), PredGuid runs three processes during every guidance cycle: AERO_PROPERTIES, PC_SEQUENCER, and LATERAL_CONTROL.

AERO_PROPERTIES: PredGuid first updates its estimate of relevant aerodynamic properties: vehicle L/D and an atmospheric density multiplier (which is applied to the internal atmosphere model to better estimate actual conditions).

PC_SEQUENCER: Next the predictor-corrector sequencer is run, which iterates between the predictor and corrector until a constant bank angle is found which will produce the desired apogee for the target aerocapture orbit. The predictor in PredGuid uses an inverse square gravity field with adjustments for J2 effects, and the 1976 U.S. Standard Atmosphere model. The full, nonlinear equations of motion are used and numerically integrated from current conditions until the trajectory reaches the 'edge of the atmosphere,' defined by PredGuid to be the same altitude as EI: 400,000 ft (122 km). From that point, a Keplerian calculation is made to estimate the apogee of the new orbit.

LATERAL_CONTROL: In order to target the desired orbital inclination, PredGuid commands a bank angle reversal if the orbital inclination predicted by the predictor is outside a certain tolerance. The allowed deviation from the desired inclination decreases ('necks down') as the algorithm gets closer to completion.

5.4 PredGuid in the Upcontrol Phase

Only portions of PredGuid were implemented in the guidance algorithm. Some modifications were required to account for differences between aerocapture and skip trajectory applications. The method used to determine PredGuid's target relies heavily on the characteristics of the Final phase reference trajectory.

5.4.1 PredGuid Implementation in the Upcontrol Phase

Since PredGuid is being substituted here as part of an algorithm, rather than being a self-contained algorithm, it is necessary only to use the main phase sequencer, so the INITIAL_GUID subroutine was discarded. In addition, the original Apollo algorithm already includes a lateral control subroutine nearly identical to that of PredGuid, except that Apollo's lateral control targets a particular crossrange rather than an orbital inclination. Thus, Apollo's lateral targeting was retained and PredGuid's became unnecessary.

It was decided to calculate the atmospheric density bias estimate from the beginning of reentry and continue throughout the entire reentry trajectory. This allows a best estimate already to be in place when PredGuid is first brought online in the Upcontrol phase. For that reason, the density bias estimation calculation was moved to the targeting step which occurs at the beginning of every guidance cycle, rather than being kept attached to PredGuid. The other part of the AERO_PROPERTIES subroutine, L/D estimation, was not implemented in this phase of development, in order to simplify the inputs required by the guidance algorithm.

The only remaining portion of PredGuid is the PC_SEQUNCER subroutine, and this is what was used to replace the Upcontrol phase. However, some changes were necessary in order to make it more suitable to skip guidance. PredGuid was altered to target a particular downrange rather than an orbital apogee. The stop conditions for the integrator

were changed so that the trajectory is now integrated out to the start of the Final phase – not just to atmospheric exit. Rather than calculating apogee, the downrange from the start of the skip is calculated, and this is the value handed off to the corrector. Some of the corrector’s logic had to be changed to reflect the fact that atmospheric capture was essential (rather than undesired, as with aerocapture).

Figure 5.2 shows how PredGuid targeting algorithm fits into the larger guidance algorithm during the Upcontrol phase.

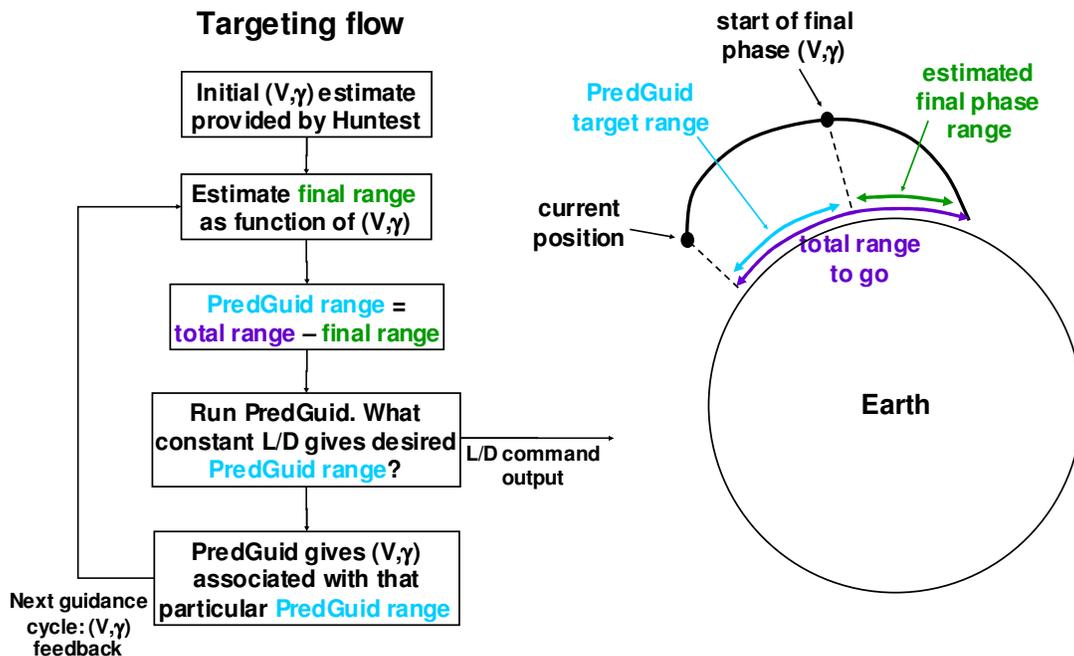


Figure 5.2: Guidance Targeting Flow during Upcontrol Phase

The inputs to PredGuid are current conditions (e.g. inertial position and velocity) and a desired range to cover before the start of the Final phase. This desired range, shown in the figure as ‘PredGuid range’, is calculated by subtracting the estimated Final phase range from the total range to go. Thus, the PredGuid’s ability to target the Final phase energy bucket is dependent on the Final phase range estimation. The Final phase range estimation is identical to the one used in the Hunttest phase of the original Apollo algorithm. The range is estimated as a linear perturbation of the velocity (V) and flight path angle (γ) at the transition into the Final phase around a reference target point. This is the same point which is used as the initial condition for the Final phase reference

trajectory, and it should correspond to the expected conditions at the start of the final phase [11]. For the original Apollo algorithm, this point corresponds to:

$$V = 23,500 \text{ fps}$$

$$\gamma = -2 \text{ deg.}$$

The linear perturbation is essentially a Taylor expansion around the reference point, as in Equation 5.1. The gains $(\partial X/\partial V, \partial X/\partial \gamma)$ are determined as part of the Final phase gain design.

$$X = X_{ref} + \frac{\partial X}{\partial V}(V - V_{ref}) + \frac{\partial X}{\partial \gamma}(\gamma - \gamma_{ref}) \quad (5.1)$$

where:

X : range to go

It is thus evident that this method of targeting the energy bucket is very closely related to the design of the Final phase reference trajectory, and the targeting capability may later be improved by updating the Final phase reference trajectory to more closely match a given vehicle's characteristics, rather than using one tuned specifically for the Apollo capsule.

The Final phase range estimation is a function of the V and γ predicted for the start of the Final phase. As shown above in Figure 5.2, the Hunttest phase provides an initial estimate of these values, but they are updated each guidance cycle as an output from PredGuid. PredGuid also outputs a bank angle command each guidance cycle, which is put through the original Apollo lateral logic routine before being passed on to flight control.

5.4.2 Results with PredGuid in the Upcontrol Phase

Figure 5.3 shows the improved precision landing performance achieved for the 7,300 km and 10,000 km target ranges when the Upcontrol phase was replaced by PredGuid. The other target ranges are not shown because they show the same excellent landing precision performance as before. Of course, it would be expected that performance would not change for the 2,400 km target range, since short ranges do not require a skip, and thus do

not use PredGuid – the parts of the algorithm that are used have remained exactly the same.

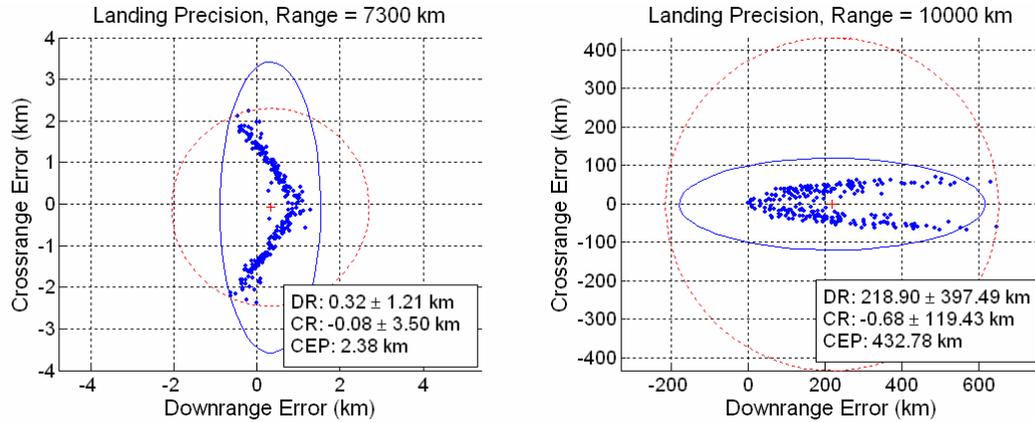


Figure 5.3: Improved Precision Landing with PredGuid Replacing Upcontrol

Using PredGuid enabled precision landing for the 7,300 km case, which was not achievable using the original Apollo algorithm given the assumed atmospheric and aerodynamic uncertainties. Also, the 10,000 km target range appears to have improved – the CEP is down from 800 km with the original algorithm to 430 km with PredGuid – but it still does not show acceptable landing precision. It is interesting to note that the characteristic of the landing pattern has changed; whereas the original Apollo algorithm missed by undershooting the target in almost every trial, the algorithm with PredGuid misses the target by overshooting. The fact that there is still such a large miss distance raises the question of how well PredGuid guided the trajectory into the Final phase energy bucket, and where the algorithm is targeting with respect to the bucket. These questions are answered by Figure 5.4, which depicts the targeted point and actual state for each trial’s transition into the Final phase.

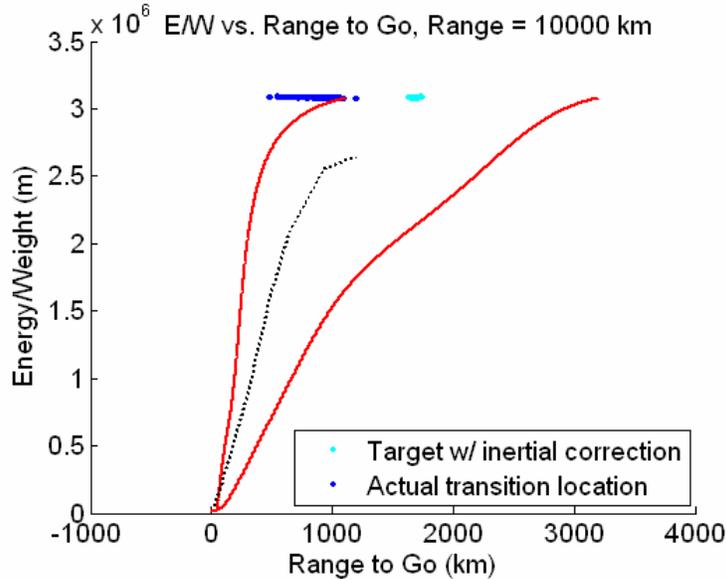


Figure 5.4: Transition into the Final Phase, PredGuid Replacing Upcontrol Phase

It appears that for each trial, the target is biased toward the left side of the bucket, and that the actual transition overshoot the targeted transition point. The fact that the trajectories overshoot the target can be mitigated somewhat by centering the target within the bucket. Since the target location is very closely related to the Final phase reference trajectory, this can be done by updating the original Apollo reference trajectory to a trajectory specifically designed for the CEV, as described in the next section. This will also extend and re-center the reference trajectory in the energy bucket.

5.5 Final Phase Reference Trajectory and Gain Design

The reference trajectory generation and gain design process is a two-step procedure. Figure 5.5 gives a conceptual view of this process. First, the simplified equations of motion are integrated from the chosen initial conditions to the ground, to yield the reference trajectory. Then, the adjoint to the linearized perturbation equations are integrated from a desired final condition to the start of the reference trajectory. This yields the sensitivities of range capability to various state variables, which are used in the Final phase as control gains and also used by PredGuid for targeting the energy bucket.

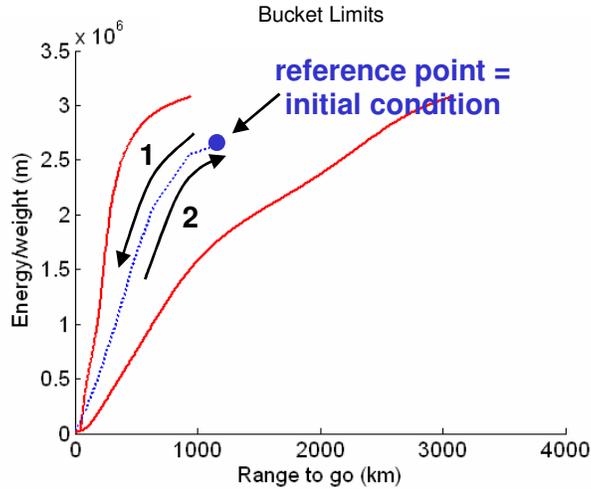


Figure 5.5: Reference Trajectory Generation and Gain Design Procedure

Following Apollo's example, the initial condition was chosen to be the expected Final phase initial conditions for the maximum range (in this case, 10,000 km). The Final phase range estimation method was also enhanced to vary the transition point about which a linear expansion is taken. This allows the new Final phase reference trajectory to support a much larger spread in target ranges.

5.5.1 Trajectory and Gain Design Procedure

The state vector for the equations of motion is shown in Equation 5.2.

$$\bar{x}(t) = \begin{bmatrix} X(t) : \text{Downrange} \\ V(t) : \text{Velocity} \\ \gamma(t) : \text{Flight path angle} \\ H(t) : \text{Altitude} \\ \frac{L}{D}(t) : \text{Lift-to-drag ratio} \end{bmatrix} \quad (5.2)$$

Four of the states, downrange, velocity, flight path angle, and altitude are fairly standard state variables. The fifth state is L/D: the portion of the lift-to-drag ratio which is in the vertical channel. Since the reference trajectory is generated using the assumption of constant bank angle, it is not necessary to include L/D as a state variable for the reference generation process. It is used later, however, in the gain design process.

The equations of motion are described by Equations 5.3 – 5.5. These state derivatives represent flight over a non-rotating, spherical planet, and Equation 5.3 is a fairly standard representation.

$$\dot{\vec{x}} = \begin{bmatrix} V \cos(\gamma) \\ -D - g \sin(\gamma) \\ \frac{L}{V} - \left(\frac{g}{V} - \frac{V}{R_E + H} \right) \cos(\gamma) \\ V \sin(\gamma) \\ 0 \end{bmatrix} \quad (5.3)$$

where:

$$D = \frac{SC_D}{m} \cdot \frac{1}{2} \rho(H) \cdot V^2 = \frac{\bar{q}}{B_n} \quad (5.4)$$

$$L = \left(\frac{L}{D} \right) \cdot D \quad (5.5)$$

and:

$$\begin{aligned} R_E &: \text{Earth's radius (m)} \\ H &: \text{Altitude (m)} \end{aligned}$$

An atmospheric model must be used to determine density (ρ) as a function of altitude. The original Apollo reference trajectory was generated using the 1959 ARDC Standard Atmosphere. For this thesis, however, it is appropriate to use the 1976 U.S. Standard Atmosphere since this is the atmospheric model used in other portions of the guidance algorithm.

The equations of motion are then numerically integrated subject to the following initial conditions:

$$\bar{x}(t_0) = \begin{bmatrix} X(t_0) \\ V(t_0) \\ \gamma(t_0) \\ H(t_0) \\ \frac{L}{D}(t_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 7,163 \text{ m/s} \\ -2 \text{ deg.} \\ H(0.2g's) \\ 0.18 \end{bmatrix} \quad \text{(Apollo)} \quad (5.6)$$

Integration is terminated once the altitude has reached zero, representing the vehicle hitting the ground.

In order to generate the range sensitivities, the adjoint to the linearized perturbation equations must be determined. The general equations of motion can be represented in the format shown by Equation 5.7, and linearized about the reference trajectory as shown in Equations 5.8 and 5.9.

$$\dot{\bar{x}}(t) = \bar{f}(\bar{x}(t)) + \bar{u}(t) \quad (5.7)$$

$$\delta \dot{\bar{x}}(t) = A(t) \cdot \delta \bar{x}(t) + \bar{u}(t) \quad (5.8)$$

where:

$$A(t) = \left[\frac{\partial \bar{f}(\bar{x}(t))}{\partial \bar{x}(t)} \right]_{\text{Ref. traj.}} \quad (5.9)$$

In general, the adjoint equations to a general linearized system are as shown in Equation 5.10. Equation 5.11 shows the adjoint equations for the particular system of equations of motion described above by Equation 5.3.

$$\dot{\bar{\lambda}}(t) = -A^T(t) \cdot \bar{\lambda}(t) \quad (5.10)$$

$$\dot{\bar{\lambda}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\cos(\gamma) & \rho V \frac{SC_D}{m} & \left(-\frac{1}{R_E} - \frac{g}{V^2}\right) \cos(\gamma) - \frac{1}{2} \rho \frac{SC_L}{m} & -\sin(\gamma) & 0 \\ V \sin(\gamma) & g \cos(\gamma) & \left(\frac{V}{R_E} - \frac{g}{V}\right) \sin(\gamma) & -V \cos(\gamma) & 0 \\ 0 & \frac{1}{2} V^2 \frac{SC_D}{m} \frac{d\rho}{dH} & -\frac{1}{2} V^2 \frac{SC_L}{m} \frac{d\rho}{dH} & 0 & 0 \\ 0 & 0 & -\frac{1}{2} \rho V \frac{SC_D}{m} & 0 & 0 \end{bmatrix} \bar{\lambda} \quad (5.11)$$

When these adjoint equations are applied as integrating factors to the equations of motion, the problem reduces to Equation 5.12. This equation relates the final state's perturbations from the reference trajectory as a function of the initial perturbations.

$$\bar{\lambda}^T(t_f) \Delta \bar{x}(t_f) = \bar{\lambda}^T(t_0) \Delta \bar{x}(t_0) \quad (5.12)$$

It is desirable to choose the boundary conditions such that $\lambda(t_f)$ represents only range error at the final time. In such a case, Equation 5.12 simplifies to Equation 5.13, which can be re-written as in Equation 5.14.

$$\lambda_1(t_f) \Delta x_1(t_f) = \bar{\lambda}^T(t_0) \Delta \bar{x}(t_0) \quad (5.13)$$

$$\Delta X(t_f) = \left[\lambda_1 \Delta X + \lambda_2 \Delta V + \lambda_3 \Delta \gamma + \lambda_4 \Delta H + \lambda_5 \Delta \frac{L}{D} \right] \Bigg|_{t=t_0} \quad (5.14)$$

Equation 5.14 can be interpreted as a Taylor expansion of the range at t_f , where the λ_i variables represent the sensitivities of the final range to the initial conditions, as in Equation 5.15. In such a case, the λ vector can be defined as in Equation 5.16.

$$\Delta X(t_f) = \left[\frac{\partial X}{\partial X} \Delta X + \frac{\partial X}{\partial V} \Delta V + \frac{\partial X}{\partial \gamma} \Delta \gamma + \frac{\partial X}{\partial H} \Delta H + \frac{\partial X}{\partial(L/D)} \Delta \frac{L}{D} \right] \Bigg|_{t=t_0} \quad (5.15)$$

$$\bar{\lambda}(t) = \begin{bmatrix} \frac{\partial X}{\partial X} \\ \frac{\partial X}{\partial V} \\ \frac{\partial X}{\partial \gamma} \\ \frac{\partial X}{\partial H} \\ \frac{\partial X}{\partial(L/D)} \end{bmatrix} \quad (5.16)$$

The λ vector should represent sensitivities of range to the various state variables, as in Equation 5.12. Thus, the range error, which is the difference between the actual range to go and the reference trajectory range to go, can be calculated simply as in Equation 5.13.

The sensitivities which comprise the λ vector are known at the end of the reference trajectory (i.e. at the ground) and can be used as boundary conditions as shown in Equation 5.14.

$$\bar{\lambda}(t_f) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \cot(\gamma(t_f)) \\ 0 \end{bmatrix} \quad (5.14)$$

The system of equations can then be numerically integrated backward in time along the reference trajectory. This results in a reference trajectory and gains which correspond to particular points in time; time is the independent variable. Since velocity is continuously decreasing, however, velocity can be used as the independent variable for the purposes of the Final phase controller and Final phase range estimation, in order to avoid time-dependence of the reference trajectory and gains.

5.5.2 Choosing the Reference Trajectory Parameters

Some parameters used for the reference trajectory generation and gain design are clearly defined, such as vehicle mass, lift and drag coefficients, and so on. Others are up to the discretion of the trajectory designer, in particular, reference L/D and trajectory initial conditions. There are many possible values for these parameters, and this section describes how the parameters were chosen.

First, the initial conditions for the reference trajectory are chosen as the expected initial conditions for the maximum target range. This is the most difficult target range to reach, and thus it is most important to choose these conditions so that the bucket targeting may be as effective as possible [11]. From the set of Monte Carlos described in the previous section, it was determined that the typical initial conditions at the start of the final phase were a velocity of about 7,700 m/s and a flight path angle of approximately -1 deg.

Second, it is desirable to create a reference trajectory which follows the center of the Final phase energy bucket. The Apollo guidance design document states that a reference L/D of 0.18 was chosen to achieve the center of the vehicle's range capability for a vehicle with an overall L/D of 0.30 [9]. Note that the L/D chosen was 60% of the overall vehicle L/D capability.

In order to choose the appropriate L/D for the CEV design, simulations were run from the expected initial conditions using values in increments of 0.035 between -0.35 (full lift down) and +0.35 (full lift up) for a reference L/D. The simulations then ran each trajectory to the ground with the L/D command constant at the reference value, but included the 10-g G-limiter from the Apollo algorithm. The results of these trajectories are shown with respect to the energy bucket in Figure 5.6.

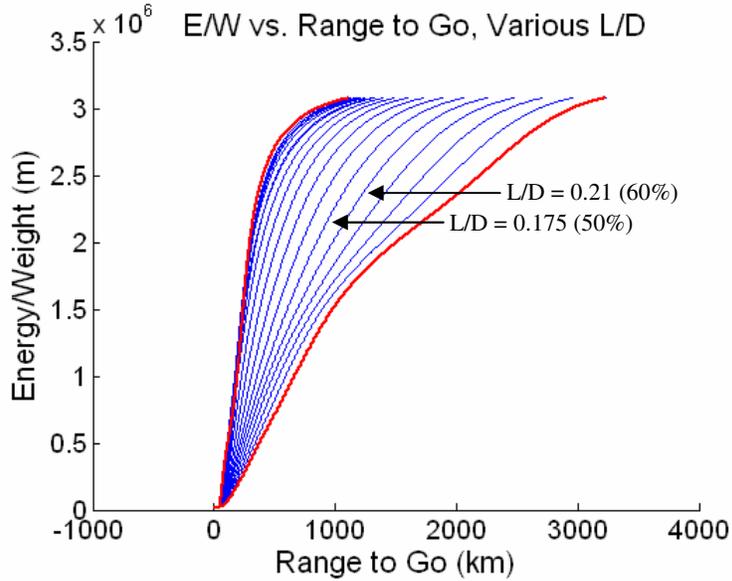


Figure 5.6: Final Phase Reference L/D

It appears that the L/D contour which most closely follows the center of the energy bucket is somewhere between an L/D of 0.175 and 0.21. These L/D values are 50-60% of the overall CEV L/D capability of 0.35. Thus, in order to remain consistent with both this analysis and the original Apollo algorithm, a reference L/D of approximately 60% of the total vehicle L/D was chosen for the Final phase reference trajectory.

5.5.3 Updated Reference Trajectory and Bucket Targeting Method

Using this new reference seems to improve the precision landing performance for the 10,000 km target range, but it seriously impairs the performance of the 2,400 km case. This turns is because the Hunttest phase now estimates that the Final phase will travel a *negative* distance, and thus infers a skip phase is necessary to reach the target range.

This is quite understandable. With linear expansions, the accuracy of the approximation decreases the farther one deviates from the reference point. For example, a linear expansion of the function $f(x) = \sin(x)$ about the point $x = 0$ is $g(x) = x$. As Table 5.1 shows, the linear expansion $g(x)$ is a fairly good approximation to $f(x)$ for values of x near the expansion point. For points farther away, however, $g(x)$ continues to become increasingly worse as an approximation.

Table 5.1: Linear Expansion Example

| Independent variable, x | Original function, f(x) | Linear expansion, g(x) | % error |
|-------------------------|-------------------------|------------------------|---------|
| 0.1 | 0.0998 | 0.1 | 0.17 % |
| 0.2 | 0.198 | 0.2 | 0.67 % |
| 0.4 | 0.389 | 0.4 | 2.7 % |
| 0.8 | 0.717 | 0.8 | 12 % |
| 1.6 | 0.9996 | 1.6 | 60 % |

The Final phase range estimation calculation is also a linear expansion, and is subject to the same limitations. Previously, with a reference point designed for the 4,600 km expected initial conditions, the actual conditions of the 2,400 km case were close enough to make a reasonable range estimation, and the Final phase could account for any errors in targeting. Now that the reference point is based on the 10,000 km expected initial conditions, with higher velocity and shallower flight path angle, the actual 2,400 km conditions are so far off that the linear expansion for the range estimation is highly inaccurate. The Final phase simply does not have enough control authority to account for the errors in the range estimation.

With a much larger spread of possible Final phase initial conditions than the ones for which Apollo algorithm was designed, it appears that it is not possible for the single-point linear expansion method will work for all target ranges. Thus, an additional interpolation method was introduced in order to enable the single reference trajectory to work acceptably for all target ranges. This method is described graphically in Figure 5.7.

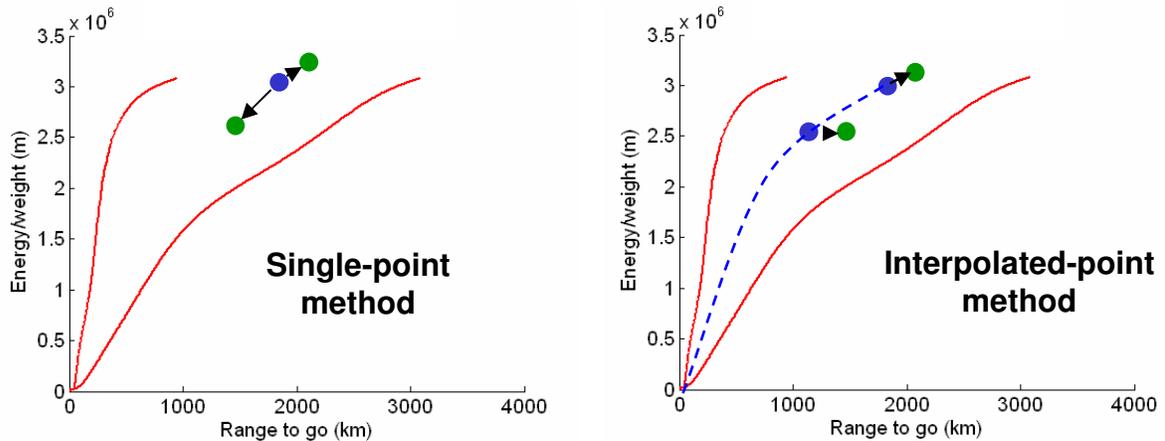


Figure 5.7: Final Phase Range Estimation: Original vs. New Method

The original ‘single-point’ method used a single data point from which a linear expansion was taken. The new method of range estimation still uses a linear expansion, but the point about which the expansion occurs may now vary according to energy level. If the projected energy of the particular trajectory is higher than the energy of the highest point in the reference trajectory, the ‘reference point’, then the linear expansion occurs from that point exactly as with the original Apollo method. However, if the projected energy is lower than that of the reference point, a linear expansion is taken from the point along the reference trajectory with the corresponding energy level, an ‘interpolated point.’ This decreases the deviation of the initial conditions for projected point from the point about which the linear expansion is taken, thus increasing the fidelity of the estimation.

5.5.4 Results with New Reference and Targeting Method

Figure 5.8 shows the landing precision and bucket targeting for the 10,000 km target range when the new reference and interpolated-point method for range estimation are implemented. The 2,400 km case is shown in Figure 5.9 in order to verify that the new reference trajectory and range estimation method works for the entire extent of the target ranges.

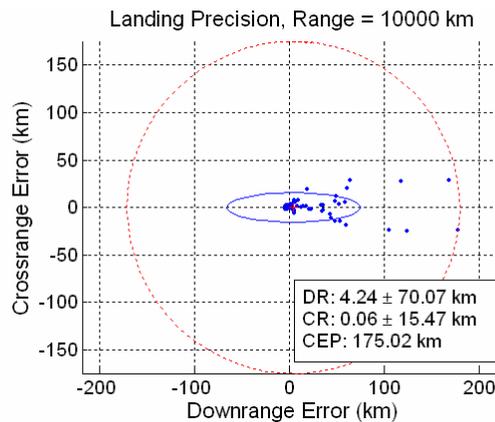


Figure 5.8: New Reference Results, 10,000 km Target

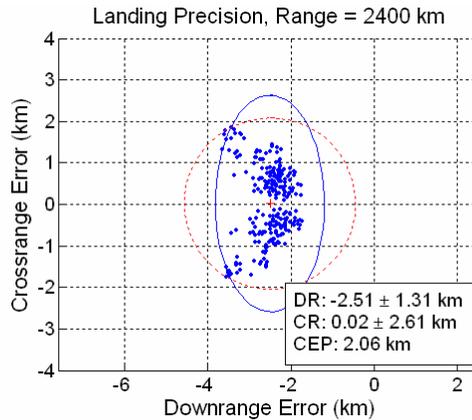


Figure 5.9: New Reference Results, 2,400 km Target

The new Final phase reference trajectory results in improved precision landing for the 10,000 km case; the CEP is down from 430 km to 175 km. This is still not enough for acceptable landing precision, an issue which will be addressed in the next section. Landing precision for the 2,400 km case has also improved slightly from a CEP of 2.5 km with the original reference trajectory to 2.1 km with the new reference.

5.6 PredGuid in the Ballistic Phase

Further analysis of the long target ranges showed that there was control authority during the Ballistic phase that was not being utilized. Performance was improved by allowing PredGuid to control steering during the Ballistic phase.

5.6.1 Control Authority during the Ballistic Phase

Since the 10,000 km target range was the only target range not showing acceptable landing precision in the Monte Carlo landing footprints, an extensive analysis was performed on this case. It was found that both exit conditions (leaving the atmosphere) and reentry conditions (into the Final phase) had much shallower flight path angles than expected by the original Apollo algorithm; angles of less than a degree were typical, rather than the expected 2 to 2.5 degrees. In such a shallow case, the vehicle would be skimming the atmosphere, encountering much higher drag levels than previously expected. For that reason, it was decided to re-examine the original Apollo algorithm's assumption that the vehicle has no control authority during the ballistic phase.

An analysis of the validity of this assumption was carried out in Reference [22]. A series of trajectories was started with varying exit conditions (both velocity and flight path angle), and each trajectory was propagated from the beginning of the Ballistic phase (corresponding to the exit conditions) until the end of the Ballistic phase. For each case, one trial was run with the vehicle’s lift vector directed upward, and one was run with the lift vector downward, and the range covered for each trial was recorded. If the vehicle truly had no control authority during the ballistic phase, the direction of the lift vector wouldn’t matter and the two cases would end up with the same downrange traveled. If, however, the vehicle did have some control authority, the trials would show a difference in range traveled.

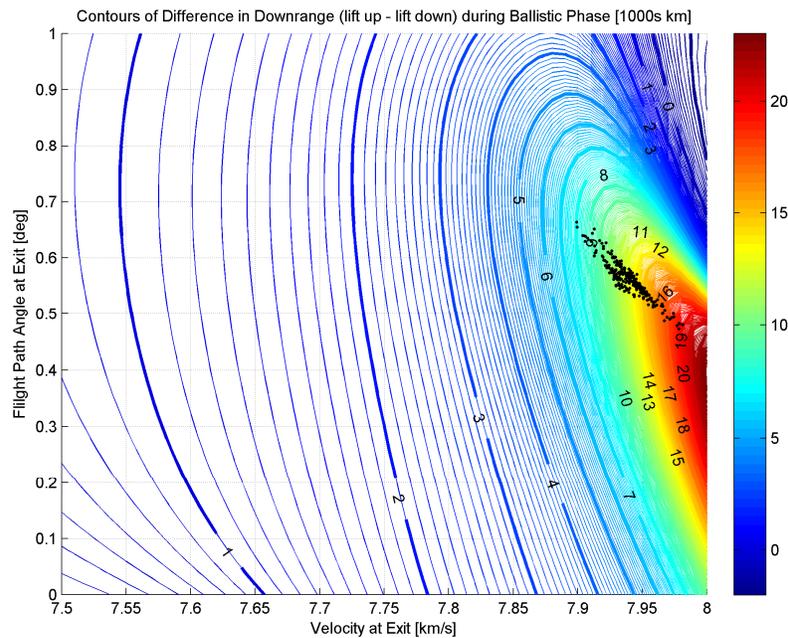


Figure 5.10: Ballistic Phase Control Authority [22]

Figure 5.10 shows the results of this analysis. Along both axes are the varying exit conditions corresponding to the start of the Ballistic phase, and the contours show the difference in downrange traveled between the lift up case and the lift down case. Larger flight path angles show a smaller range difference, indicating less sensitivity to the direction of the lift vector. This is to be expected because a larger angle would lead the vehicle into a higher skip, farther out of the atmosphere, where there would be less

dynamic pressure. Similarly, lower velocities show less sensitivity to the lift vector direction.

The smattering of black dots indicates the exit conditions taken from an 8,000 km target range Monte Carlo. These dots fall within a region of high sensitivity, indicating that the Upcontrol phase (including PredGuid) is guiding to exit conditions such that significant control authority is available during the Ballistic phase. The exit conditions from the 10,000 km target range Monte Carlo fall within an even higher-sensitivity region of the plot.

Since it is apparent that control authority is available during the Ballistic phase, and the 10,000 km target range still needed improvement in its landing precision, it only makes sense to allow PredGuid to continue steering during the Ballistic phase. This change has been implemented essentially identically to the implementation described for the Upcontrol phase.

5.6.2 Results with PredGuid in the Ballistic Phase

Figure 5.11 shows the improved precision landing performance and bucket targeting achieved for the 10,000 km target range – the only target range still not showing acceptable landing precision – when the Ballistic phase was replaced by PredGuid.

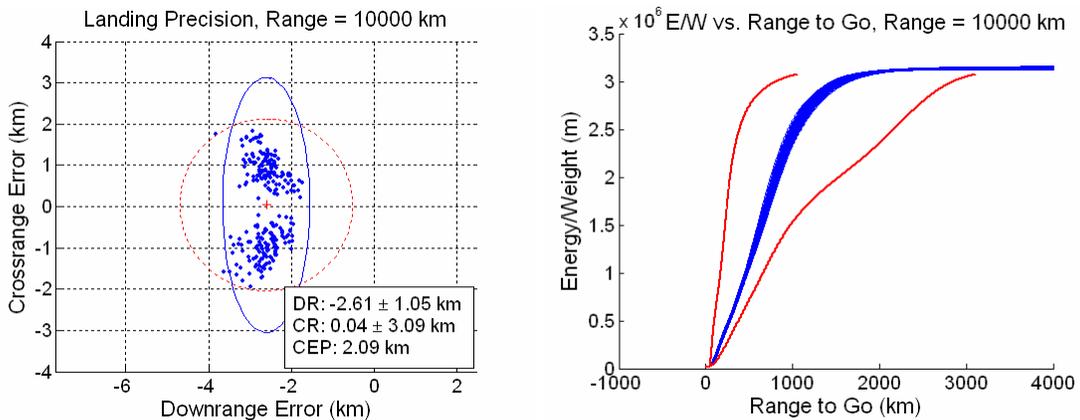


Figure 5.11: Results with PredGuid Replacing Ballistic Phase

The 10,000 km range now shows acceptable landing precision. The trajectories also show margin with respect to the Final phase energy bucket, which indicates that the algorithm should be fairly robust to unexpected uncertainties.

5.7 PredGuid in the Downcontrol Phase

Replacement of the Upcontrol and Ballistic phases of the original Apollo algorithm proved sufficient to achieve acceptable precision landing. It was of interest to investigate the effects of starting PredGuid sooner. It was convenient to start PredGuid by replacing the Downcontrol phase with PredGuid in the same manner as the implementation for the Upcontrol and Ballistic phases.

In most respects, the effects of starting PredGuid earlier – during the Downcontrol phase rather than the Upcontrol phase – are minimal; landing precision is unaffected by this change. The chief difference between the two versions is the shape of the skip trajectory, which can be seen in Figure 5.12. Starting PredGuid earlier results in a steeper, higher altitude skip whereas starting later results in a shallower, lower altitude skip. Each of these options has its advantages and disadvantages.

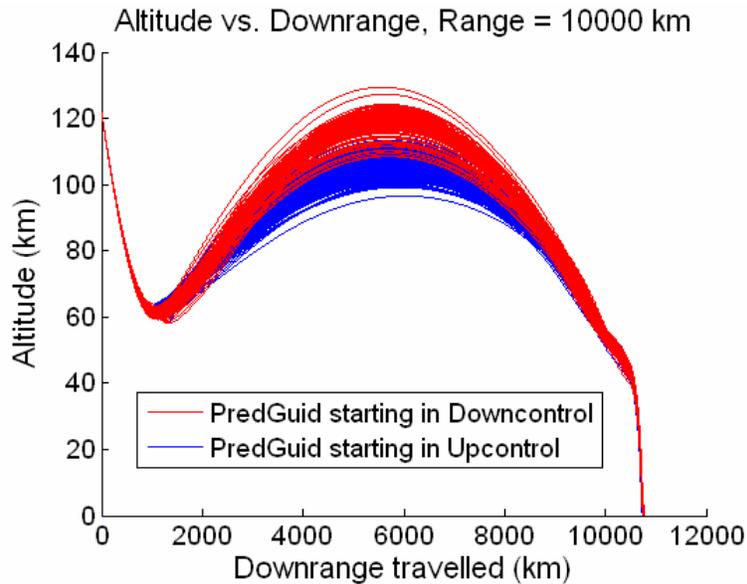


Figure 5.12: Comparison of Trajectory Shaping

A lower altitude skip keeps the vehicle in a region of the atmosphere with higher density, so there is more aerodynamic control authority. However, the atmospheric characteristics of this region of the atmosphere are highly variable, and the associated uncertainties could result in sudden, unexpected loss of aerodynamic control.

A suggestion has been made to use the thrusters during the Ballistic phase for a ‘cleanup maneuver,’ in order to retarget the Final phase energy bucket and make up for any inaccuracies in targeting the exit conditions at the end of the Upcontrol phase. This possibility is the subject of another thesis [22]. A cleanup maneuver would require re-orientation of the vehicle in a potentially aerodynamically unstable orientation, so that the thrusters would be pointed in the right direction. In some respects, a high altitude loft would be more desirable for use with a cleanup maneuver. With lower densities, there would be less aerodynamic torque to overcome in reorienting the vehicle, and it would be easier to maintain an unstable orientation. In addition, this type of trajectory results in a slightly lower heat load since there is less time spent in the atmosphere. However, this type of trajectory results in a lower-energy orbit, which means that a cleanup maneuver would have less control authority for a given ΔV allocation. A high altitude loft also takes a slightly longer time to complete, and with the lack of aerodynamic forces, there is a lack of aerodynamic control authority.

Table 5.2 summarizes the advantages and disadvantages of the low altitude skip vs. the high altitude skip.

Table 5.2: Pros and Cons of Low Altitude Skip vs. High Altitude Skip

| | Pros | Cons |
|-------------------------------------|--|--|
| Low altitude skip (shallow FPA) | <ul style="list-style-type: none"> • Higher density → more aero control authority • Shorter skip time in emergency scenario • Higher energy orbit → cleanup maneuver has more control authority | <ul style="list-style-type: none"> • Upper atmosphere highly uncertain and variable → potential for sudden loss of control authority • Larger aero moments → potential difficulty in changing attitude for cleanup maneuvers • Limited time for any cleanup maneuvers (if required) |
| High altitude skip (steeper FPA) | <ul style="list-style-type: none"> • Additional time to navigate • Additional time to reorient & perform cleanup maneuver • Less atmospheric uncertainty • Smaller disturbing aero forces • Greater heat dissipation, lower total heat load | <ul style="list-style-type: none"> • Places the CEV in flight regime with no aero control authority • Lower energy orbit → cleanup maneuver has less control authority |

[This Page Intentionally Left Blank]

Chapter 6

Results

6.1 Overview

An algorithm has been found with a precision landing footprint that satisfies the CEV precision landing requirements set forth by NASA. In addition, it has been found that the trajectory can be shaped by modulating the start time of the PredGuid portion of the algorithm. This chapter investigates how the algorithm metrics respond to this particular algorithm, how they vary with target range, and how they vary with PredGuid start time. The final section of this chapter evaluates the robustness of the algorithm with respect to atmospheric uncertainty.

6.2 Test Case Summary

Each of the algorithm metrics described in Chapter 3 are discussed with respect to the standard target ranges: 2,400 km, 3,500 km, 4,600 km, 7,300 km, and 10,000 km. In addition, comparisons are made between the results for three different guidance algorithm versions:

1. Original Apollo Algorithm: the original, unaltered Apollo algorithm.
2. Low Loft Enhanced Algorithm: the version with PredGuid replacing only the Upcontrol and Ballistic phases which results in a lower-altitude loft.
3. High Loft Enhanced Algorithm: the version with PredGuid replacing all three of the Downcontrol, Upcontrol, and Ballistic phases, resulting in a high-altitude loft.

The High Loft and Low Loft algorithms also include the updated Final phase reference trajectory and range estimation method. Note that the full set of results is presented in Appendix E.

6.2.1 Precision Landing Footprint

The original Apollo algorithm does not have a precision landing footprint meeting NASA’s requirements for the CEV. The enhanced algorithm does meet those requirements, regardless of the start time for the PredGuid portion of the algorithm (for the range of start times investigated by this thesis). Table 6.1 gives a summary of the CEP values by algorithm version and target range. The landing error scatter plots for each of these cases are provided in Appendix E.

Table 6.1: Landing Precision (CEP) by Algorithm Version and Target Range

| (units are all km) | 2,400 km | 3,500 km | 4,600 km | 7,300 km | 10,000 km |
|---------------------------|----------|----------|----------|----------|-----------|
| Original Apollo Algorithm | 2.59 | 2.35 | 2.58 | 535 | 805 |
| Low Loft Algorithm | 2.06 | 2.11 | 2.21 | 2.07 | 2.09 |
| High Loft Algorithm | 2.06 | 2.07 | 2.07 | 2.01 | 1.98 |

The squares shaded gray in Table 6.1 are those that do not meet the 3.5 km CEP requirement for precision landing. This shows that the original Apollo algorithm does not achieve acceptable landing precision for the 7,300 km and 10,000 km target ranges. Both the low-loft and high-loft versions of the algorithm show acceptable landing precision for all target ranges. It is fair to say that both version show comparable precision landing capability for these target ranges.

6.2.2 G-loads

Table 6.2 gives a summary of the maximum g-loading; in this table, the gray shading is merely used as a visual aid to differentiate between columns. The histograms of maximum g-load for each of these cases are provided in Appendix E.

Table 6.2: Maximum G-loading by Algorithm Version and Target Range

| (units are all g's) | 2,400 km | | 3,500 km | | 4,600 km | | 7,300 km | | 10,000 km | |
|---------------------------|----------|-------------|----------|-------------|----------|-------------|----------|-------------|-----------|-------------|
| | μ | 3- σ | μ | 3- σ |
| Original Apollo Algorithm | 4.3 | 0.18 | 5.7 | 1.01 | 4.2 | 0.90 | 4.0 | 0.98 | 3.8 | 1.56 |
| Low Loft Algorithm | 4.3 | 0.11 | 4.0 | 0.21 | 4.0 | 0.39 | 3.8 | 0.34 | 3.8 | 0.39 |
| High Loft Algorithm | 4.3 | 0.11 | 4.0 | 0.51 | 4.0 | 0.25 | 3.8 | 0.33 | 3.8 | 0.51 |

The results in Table 6.2 show that, in general, as the target range increases, the maximum g-loading tends to decrease. This is to be expected, since the shorter target ranges require more downward acceleration (i.e. a more negative L/D fraction) to reach the shorter range; downward acceleration causes higher g-loads than upward acceleration. There is one exception to this trend: the 3,500 km case of the original Apollo algorithm.

Typically, a direct reentry will experience only one g-load peak, as in Figure 6.1, whereas a skip reentry will split into two peaks, as in Figure 6.2.

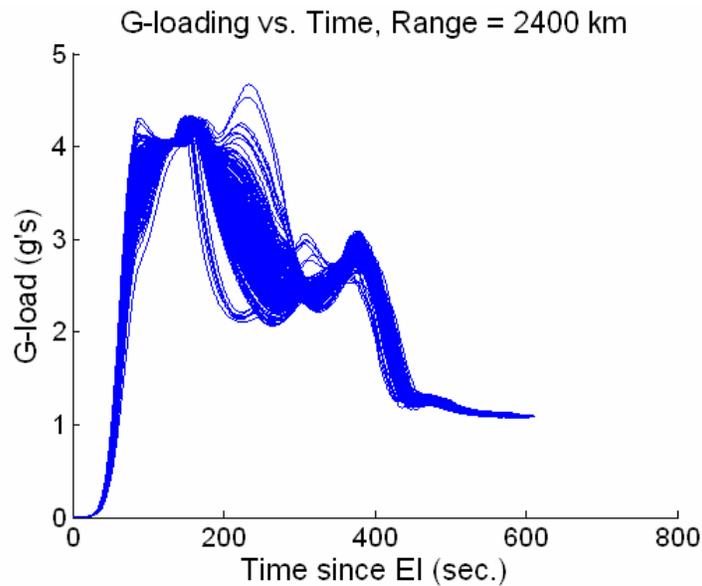


Figure 6.1: G-loading vs. Time, Low Loft Algorithm, Direct Reentry

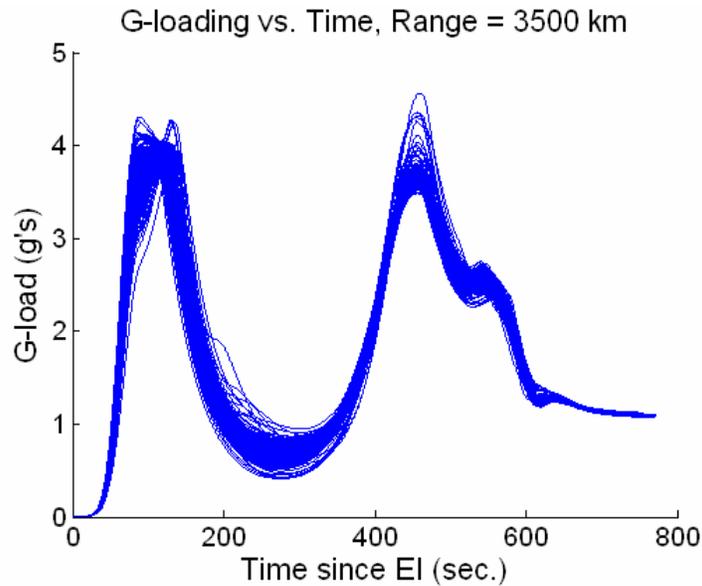


Figure 6.2: G-loading vs. Time, Low Loft Algorithm, Skip Reentry

For skip reentries, the first peak corresponds to the initial entry, and the second peak corresponds to the ‘second entry’ after the skip. Since the initial bank angle command is the same (lift up) for all of the trials in each of the Monte Carlo sets in this thesis, the magnitude of the first peak is mainly dependent on the steepness of the initial entry. Thus, all of the trials show roughly the same magnitude for the first peak. The magnitude of the second peak ends up being dependent on how successful the targeting of the Final phase energy bucket is. If the guidance hits the target perfectly, the vehicle should be able to transition into the reference trajectory seamlessly with the reference L/D. If the target has been undershot, the vehicle will have to command a higher vertical L/D in order to reach the reference trajectory, leading to lower g-loads; conversely, overshooting the target leads to more of a lift down command, leading to higher g-loads. For trials which reach the bucket target perfectly and for those that undershoot, the magnitude of the second peak will typically be of equal or lesser magnitude than the first peak. However, trials that overshoot the bucket target will show a larger second peak. This is the case for the trials in the 3,500 km Monte Carlo for the original Apollo algorithm, as shown in Figure 6.3 and Figure 6.4.

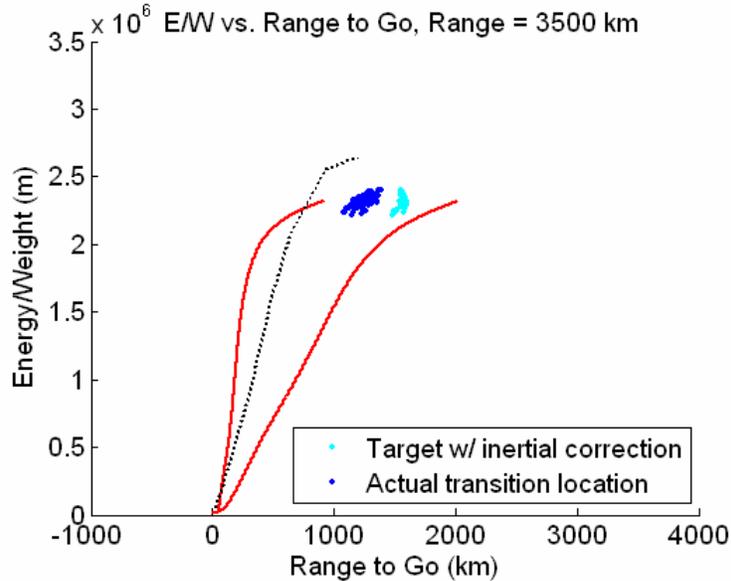


Figure 6.3: Final Phase Transition, Original Apollo Algorithm, 3,500 km

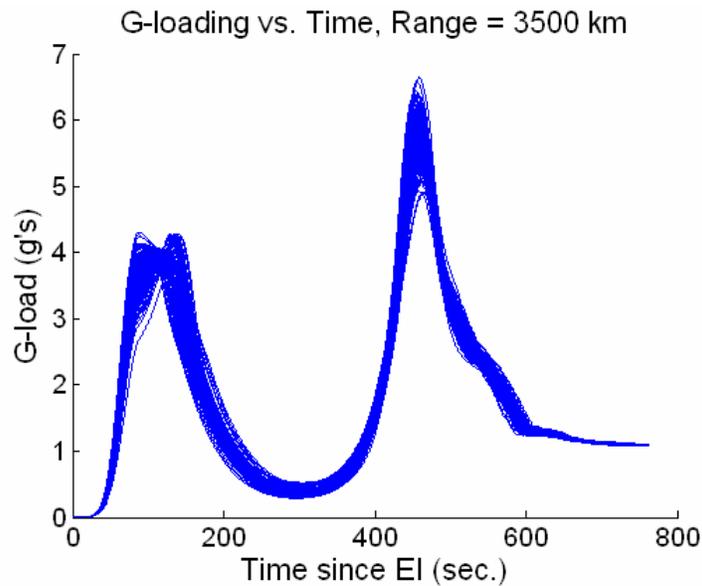


Figure 6.4: G-loading vs. Time, Original Apollo Algorithm, 3,500 km Case

The reason that this Monte Carlo shows higher g-loads than any other case is that all the other cases either reached the bucket target or undershot, and thus g-loads were driven by the first peak; this is the single case that consistently overshot the energy bucket target, causing g-loads higher than the first peak.

As for the duration-based g-loads, the constraints are very rarely violated. The 2,400 km cases for all three algorithm version each skirt the edge of the ‘sick or injured limit,’ as in the example in Figure 6.5.

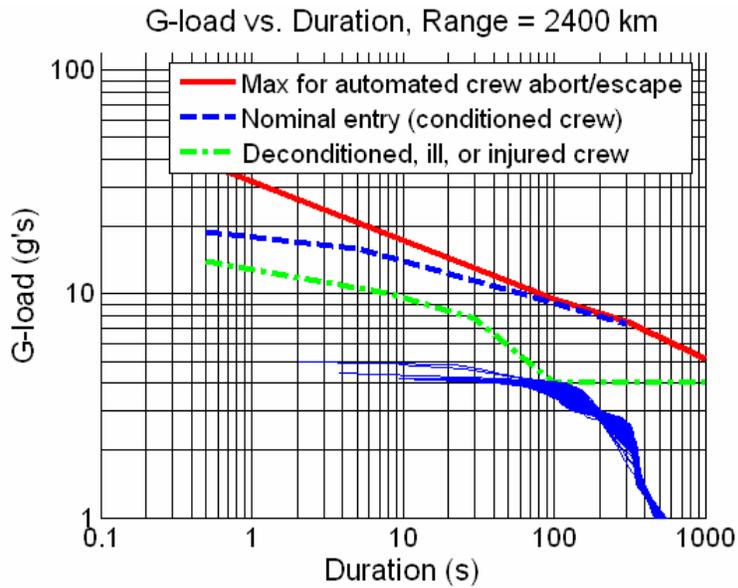


Figure 6.5: Duration-Based G-loads, Original Apollo Algorithm, 2,400 km Case

The g-loads decrease from there as the target range increases, so the trials back off from the constraint. However, there is one trial in the 10,000 km Monte Carlo using the original Apollo algorithm which violates the ‘sick or injured limit.’ This is shown in Figure 6.6.

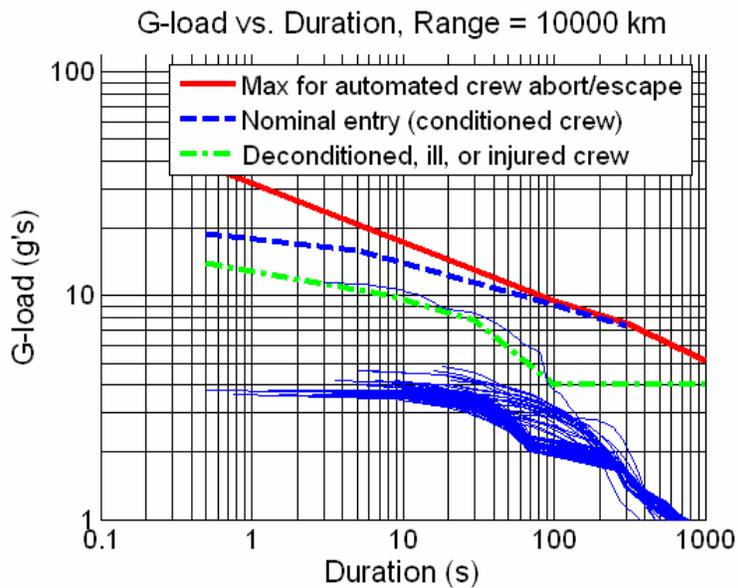


Figure 6.6: Duration-Based G-loads, Original Apollo Algorithm, 10,000 km Case

This single violation comes from the single trial that overshoot the target in this Monte Carlo, as shown in the landing error scatter plot in Figure 6.7. The dot that is circled at the bottom of the plot represents the trial under study.

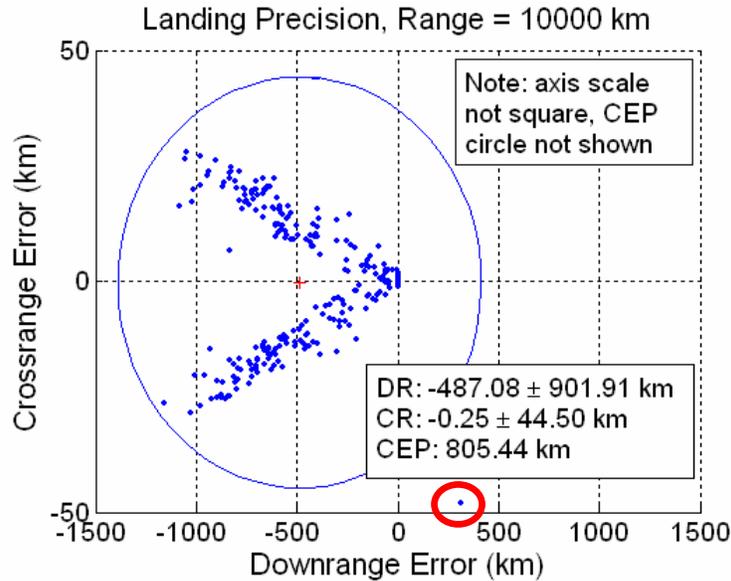


Figure 6.7: Landing Error Scatter Plot, Original Apollo Algorithm, 10,000 km Case

It makes sense that this trial should experience high g-loads; since it overshoot the target the guidance would command full lift down in order to decrease the overshoot by as much as possible, thus inducing high downward accelerations. The reason that this trial did not violate more than one g-load constraint is the 10-g limiter in the guidance logic kicked in, commanding full lift up, thus limiting the acceleration to a tolerable level.

None of the constraints are violated for either the low loft or high loft versions of the algorithm. A nominal design for guidance should only require adherence to the ‘nominal entry limit,’ so this is exceptionally good behavior.

6.2.3 Aeroheating

Table 6.3 gives a summary of the maximum heat rate values. The histogram of maximum heat rate for each of these cases is provided in Appendix E.

Table 6.3: Maximum Heat Rate by Algorithm Version and Target Range

| (units are all W/cm ²) | 2,400 km | | 3,500 km | | 4,600 km | | 7,300 km | | 10,000 km | |
|------------------------------------|----------|-------------|----------|-------------|----------|-------------|----------|-------------|-----------|-------------|
| | μ | 3- σ | μ | 3- σ |
| Original Apollo Algorithm | 263 | 53.0 | 262 | 53.0 | 262 | 53.1 | 262 | 53.1 | 262 | 53.0 |
| Low Loft Algorithm | 263 | 53.0 | 262 | 53.0 | 262 | 53.1 | 262 | 53.1 | 262 | 53.0 |
| High Loft Algorithm | 263 | 53.0 | 262 | 53.0 | 262 | 53.1 | 262 | 53.1 | 262 | 53.0 |

For these cases shown, the maximum heat rate is essentially invariant of target range and algorithm version. As with g-loading, direct reentries experience one heat rate peak, and skip reentries experience two, as seen in Figure 6.8 and Figure 6.9.

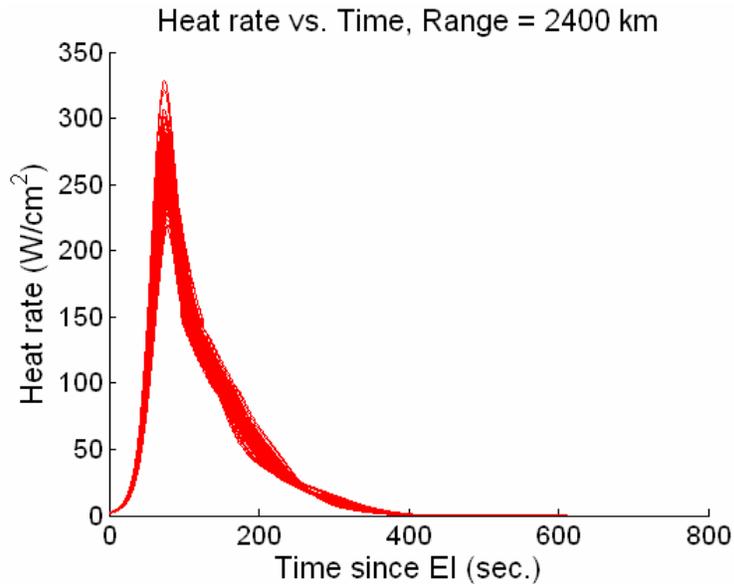


Figure 6.8: Heat Rate vs. Time, Low Loft Algorithm, Direct Reentry

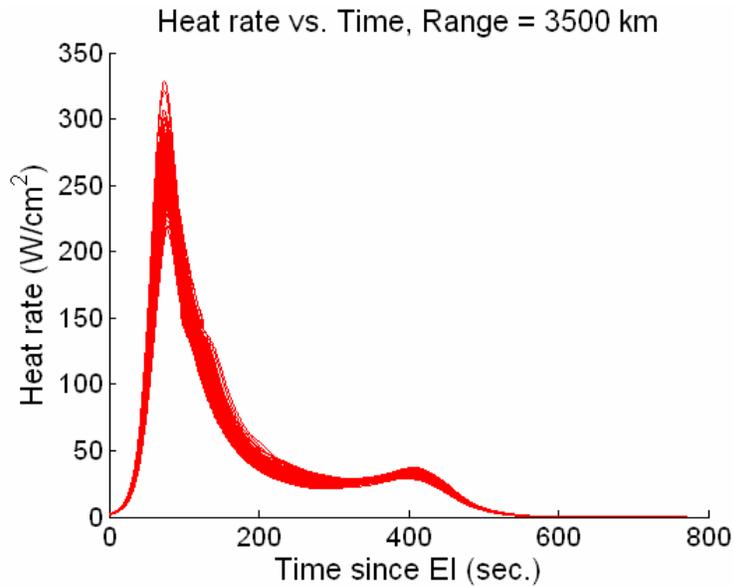


Figure 6.9: Heat Rate vs. Time, Low Loft Algorithm, Skip Reentry

The magnitude of the first peak is much larger than the second because the initial entry is both steeper and faster (higher velocity) than the second entry. Thus, heat rate is driven by the initial flight path angle and velocity and is not affected by target range or algorithm.

Total heat load, however, is affected significantly by target range and, to a small extent, by algorithm version. Table 6.4 gives a summary of the maximum heat rate values by algorithm and target range. The histogram of maximum heat rate for each of these cases is provided in Appendix E.

Table 6.4: Total Heat Load by Algorithm Version and Target Range

| (units are all kJ/cm ²) | 2,400 km | | 3,500 km | | 4,600 km | | 7,300 km | | 10,000 km | |
|-------------------------------------|----------|------|----------|------|----------|------|----------|------|-----------|------|
| | μ | 3-σ | μ | 3-σ | μ | 3-σ | μ | 3-σ | μ | 3-σ |
| Original Apollo Algorithm | 25.0 | 3.56 | 28.7 | 3.74 | 31.1 | 3.83 | 34.9 | 4.46 | 36.2 | 5.36 |
| Low Loft Algorithm | 25.0 | 3.59 | 29.1 | 3.74 | 31.6 | 4.09 | 34.8 | 5.14 | 35.0 | 5.18 |
| High Loft Algorithm | 25.0 | 3.60 | 29.0 | 3.78 | 31.2 | 4.06 | 33.1 | 4.32 | 32.8 | 4.06 |

Since heat load is simply the heat rate integrated over time, longer target ranges, which spend more time in the atmosphere, would be expected to have larger heat loads. Table 6.4 shows that this is indeed the case. There is one exception, however. The 10,000 km high loft case shows a lower heat load than the 7,300 km high loft case. This is because the 10,000 km trajectory shoots up outside the atmosphere higher and sooner than the 7,300 km case. With less time in the atmosphere, the trajectories result in lower average heat loads.

A smaller effect which is really only noticeable with longer target ranges is that the high loft algorithm shows slightly lower heat loads than the low loft algorithm. This makes sense since the high loft algorithm causes the vehicle to spend more time in the upper atmosphere, where the density is lower and the heat rate is smaller. Conversely, the low loft algorithm spends more time in the lower atmosphere with higher densities and thus higher heat rates. According to the aeroheating models used for this thesis, the total heat load can be reduced up to 6% for the 10,000 km target range by using the high loft version rather than the low loft version of the algorithm.

6.2.4 Control Authority

The primary cases which show significant control authority saturation are the long target ranges: 7,300 km and 10,000 km. It is interesting, however, that the saturation manifests itself in different ways according to which algorithm is being used.

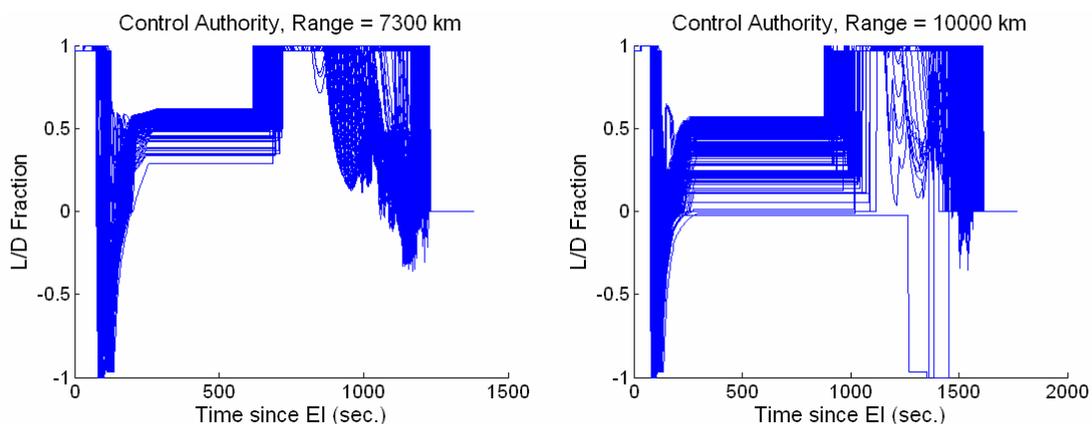


Figure 6.10: Control Authority Saturation, Long Target Ranges, Original Apollo Algorithm

Figure 6.10 shows the L/D fraction history for the 7,300 km and 10,000 km target ranges using the original Apollo algorithm. For these two cases, control authority saturation

occurs very late in the trajectory; this is because the algorithm suspends steering during the ballistic phase. The vehicle undershoots the Final phase energy bucket due to the Keplerian assumption in the Hunttest phase's Ballistic phase range estimation, thus once steering is resumed during the Final phase, the vehicle turns to a full lift up orientation to extend the vehicle's range as much as possible. The single trial which overshoot the 10,000 km target can be seen in the right plot saturating at lift down.

It is interesting to note that control authority saturation is seen at the beginning of the Final phase for the 4,600 km case as well, as in Figure 6.11. This shows that the Keplerian assumption for Ballistic range estimation is imperfect even for the 4,600 km target range, but that the Final phase has more than enough control authority to compensate. This is probably why the Keplerian assumption was deemed acceptable for the Apollo mission; it was never intended to achieve longer target ranges than 4,600 km.

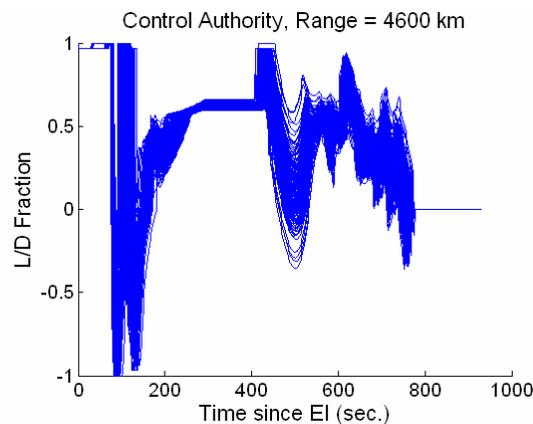


Figure 6.11: Control Authority Saturation, Medium Target Range, Original Apollo Algorithm

For the Low Loft Enhanced Algorithm, the saturation shows a different characteristic, as seen in Figure 6.12.

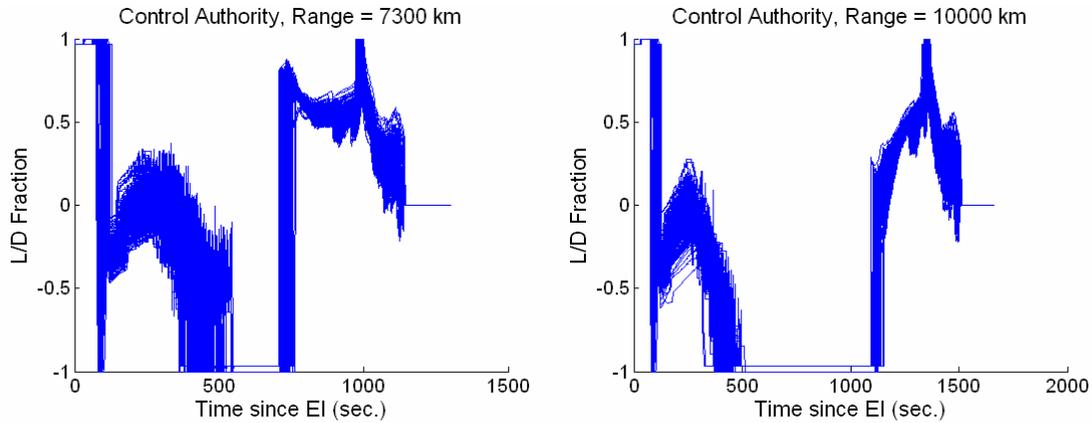


Figure 6.12: Control Authority Saturation, Long Target Ranges, Low Loft Algorithm

The saturation occurs during the Ballistic phase, rather than the Final phase. The consistent lift down saturation indicates that the vehicle overshoots its Final phase transition target in every case. This is explained later in this chapter under the heading ‘Robustness to Atmospheric Uncertainty.’ When saturation occurs consistently to one side, as in the above case, it is possible to bias the Final phase target to reduce the effects of saturation.

Once the Final phase begins, control authority is no longer saturated, since the trajectory has been guided to the reference trajectory within acceptable tolerances. The fact that the L/D command transitions so sharply from a saturated lift down command to a somewhat lift up command might indicate a mismatch between the Final phase transition target and the Final phase reference-following controller.

The High Loft Algorithm shows yet another characteristic, as in Figure 6.13.

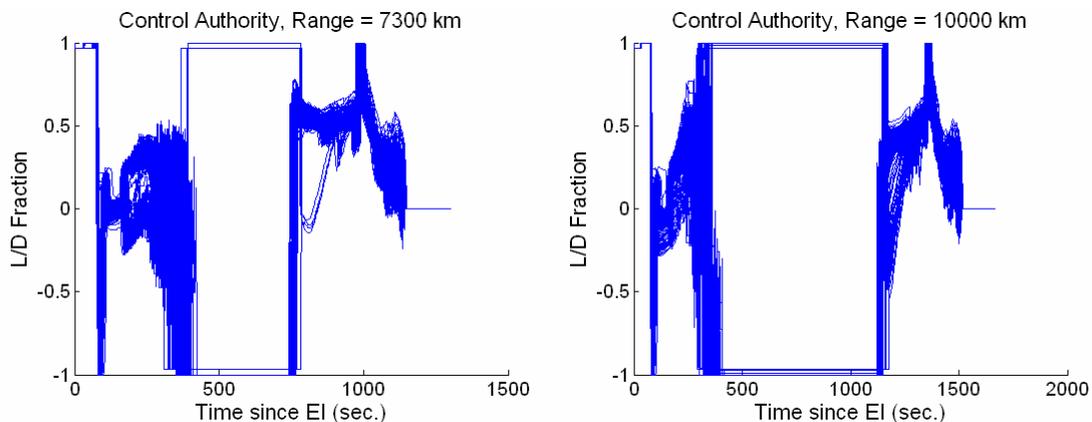


Figure 6.13: Control Authority Saturation, Long Target Ranges, High Loft Algorithm

Like the Low Loft Algorithm, the High Loft Algorithm shows saturation during the Ballistic phase and not during the Final phase. However, some of the trials saturate at lift up whereas others saturate at lift down. This indicates that some trials undershoot the Final phase transition target whereas others overshoot. In fact, only a few actually overshoot their target, but this is a good indication that the dispersions of the actual transition are closer to being centered on the target transition location. However, the fact that saturation can occur both at lift up and at lift down means that the vehicle can never reach the target within the tolerance currently specified by PredGuid – 10 nautical miles (18 km). This may mean that the tolerances need to be loosened, in order to better reflect the vehicle’s capability.

6.2.5 Final Phase Energy Bucket

For long target ranges, those trials which transition to the Final phase within the energy bucket are able to reach the target. Otherwise, they can’t. For the 7,300 km and 10,000 km cases, the original Apollo algorithm was unable to guide all trajectories into the bucket under the assumed uncertainties, whereas the enhanced algorithms, both Low Loft and High Loft, were able to do so. Thus, the Apollo Algorithm did not achieve acceptable landing precision for these target ranges, and the others did. In addition, the Low Loft Algorithm trajectories remain more tightly clustered. This can be seen for the 10,000 km target range in Figure 6.14 through Figure 6.16.

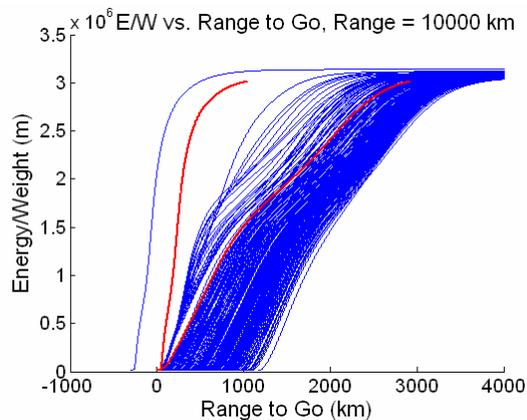


Figure 6.14: Apollo Algorithm Trajectories in Energy Bucket, 10,000 km Case

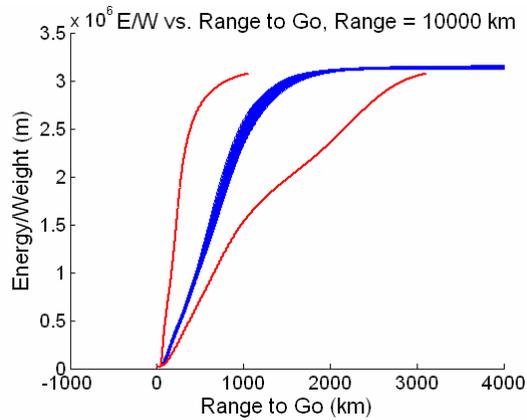


Figure 6.15: Low Loft Trajectories in Energy Bucket, 10,000 km Case

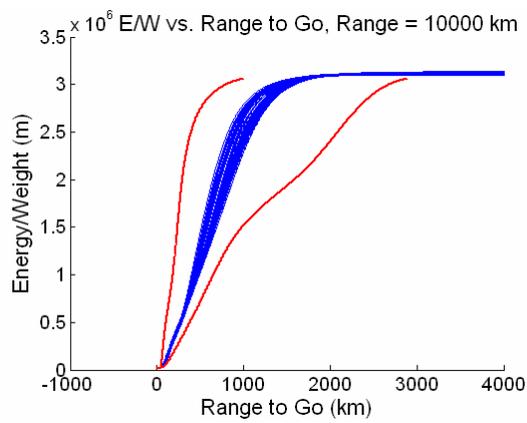


Figure 6.16: High Loft Trajectories in Energy Bucket, 10,000 km Case

However, examination of the energy bucket for the shortest ranges – 2,400 km and 3,500 km – reveals some issues with the phase transitioning logic in the Hunttest phase. This is most evident in the energy bucket plots for the 2,400 km case, seen in Figure 6.17 through Figure 6.19.

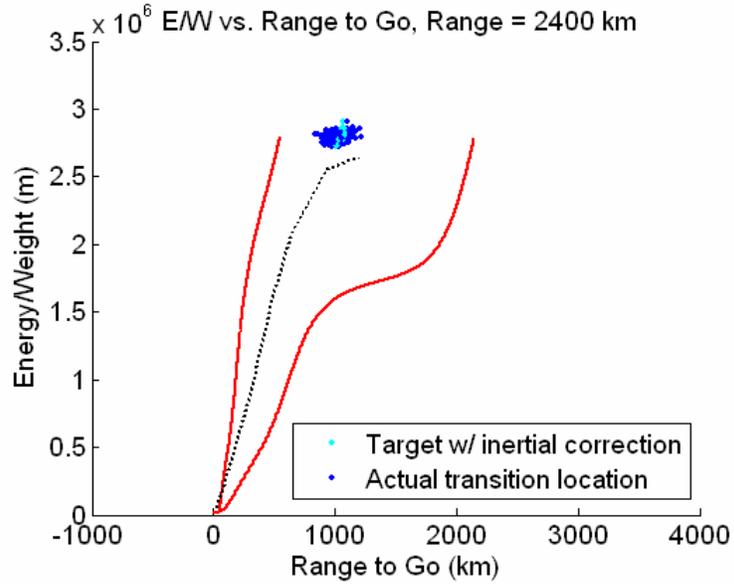


Figure 6.17: Apollo Algorithm Trajectories in Energy Bucket, 2,400 km Case

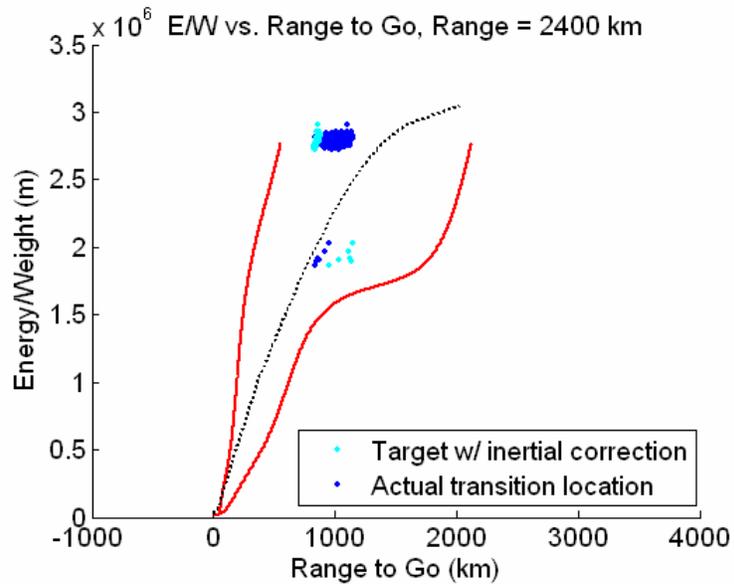


Figure 6.18: Low Loft Trajectories in Energy Bucket, 2,400 km Case

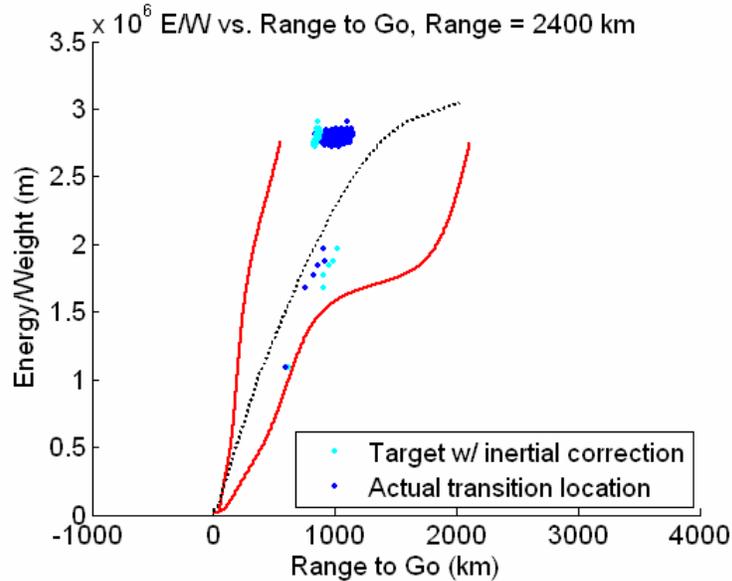


Figure 6.19: High Loft Trajectories in Energy Bucket, 2,400 km Case

For the original Apollo algorithm, all of the trials transition in a fairly tight cluster. Both the Low Loft and High Loft algorithms have this same cluster, but there are a few transitions which occur at a significantly lower energy level, at a much later time. If delayed much longer, these transitions might start to happen outside the bucket, causing unacceptable landing precision. The underlying reason for the lateness of these few transitions is that Hunttest triggers the lofting phases for those trials, even though no loft is necessary for the 2,400 km case. Thus, the transitioning logic of the Hunttest phase should be revisited.

6.3 Robustness to Atmospheric Uncertainty

The fact that some trajectories still overshoot the target for the start of the Final phase was mentioned in the section in this chapter on control authority. Figure 6.20 shows the commanded L/D fraction vs. time for a typical trial from the 10,000 km target range Monte Carlo, during the Upcontrol and Ballistic phases only. A vertical black line indicates the time at which the algorithm transitioned from the Upcontrol phase to the Ballistic phase.

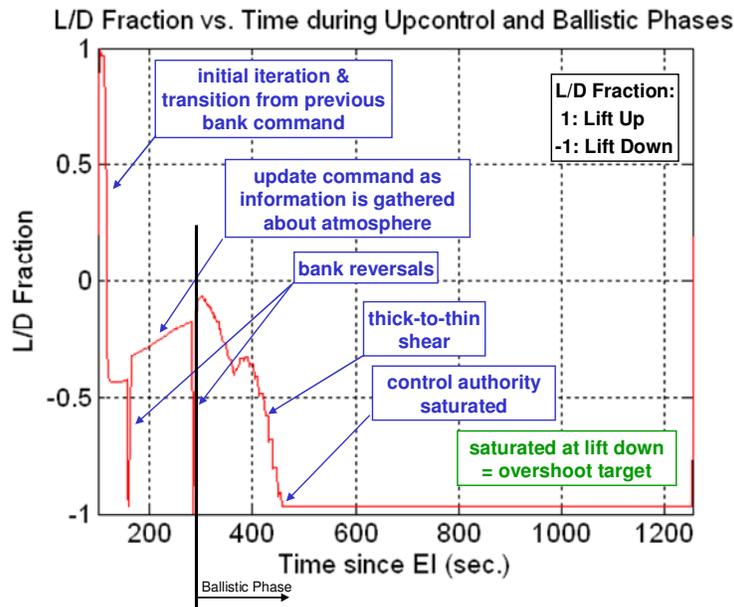


Figure 6.20: Upcontrol and Ballistic Phase Error Analysis

For this particular trial, without steering during the Ballistic phase, the bank angle command would have been left at the value calculated at the end of the Upcontrol phase, which is nearly neutral. On the other hand, allowing steering during the Ballistic phase eventually ended up with a saturated command of lift down. Such a difference in bank angle commands was sufficient to reduce a target overshoot of 2,300 km to only 15 km. However, overshoot of the Final phase transition target still occurs, and this can be seen by the control authority saturation during the Ballistic phase.

This kind of saturation is seen in every single Monte Carlo trial for the 10,000 km range. Although they all reach the landing target, they all overshoot the target for the end of the Ballistic phase, but fortunately, control authority is sufficient in the Final phase to remove the trajectory errors. This saturation occurs in every trial due to a problem of atmospheric estimation and uncertainty, mentioned earlier. At the heart of this problem is a mismatch between the atmosphere model used for Monte Carlo cases in the simulation environment and the atmosphere model used in guidance.

Figure 6.21 shows the ratio of the environment density to guidance-predicted density as a function of altitude. The Upcontrol phase occurs in the 60 – 80 km altitude regime, where the ‘actual’ environment atmospheric density is higher than the density predicted by guidance. At this point, the vehicle is early in the skip and has the most control authority, and guidance takes the thicker atmosphere into account when determining its

bank command. Once the vehicle enters the Ballistic phase, which occurs in the 80-110 km altitude regime, the environment's atmosphere is suddenly thinner than expected by guidance. This thinner atmosphere not only causes the vehicle to travel farther than expected because there is less drag, there is not as much dynamic pressure and thus the vehicle has lost the control authority necessary to correct this overshoot.

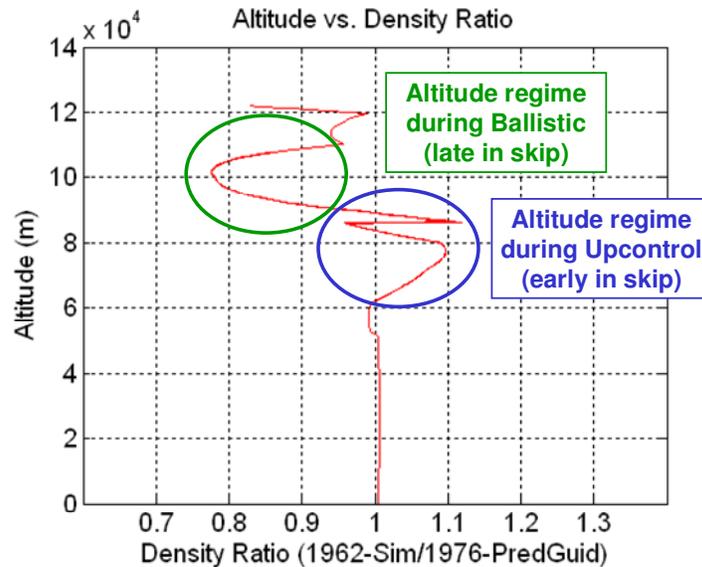


Figure 6.21: Atmosphere Model Mismatch: Thick-to-Thin Shear.

This phenomenon is essentially a thick-to-thin density shear, which is known in the aerocapture world to be a driving or stress case. Thus, all the Monte Carlo analyses as described in this document have been applied to a driving case, and yet good landing precision has still been obtained. This supports the argument for the robustness of this algorithm.

Chapter 7

Conclusions

7.1 Summary and Conclusions

The guidance algorithm for CEV Earth reentry and precision landing is based on the Apollo reentry guidance algorithm. The guidance phases pertaining to direct reentry have remained essentially unaltered, but the phases relating to skip trajectories have been upgraded using PredGuid, a numeric predictor-corrector aerocapture algorithm developed by Draper Laboratory for the Aero-assist Flight Experiment (AFE – circa 1988). These upgrades have been sufficient to allow precision landing of skip reentry trajectories for target ranges of up to 10,000 km. The algorithm has been shown to be quite robust given day of flight uncertainties and has been successfully tested against certain driving or stress cases.

In addition, it has been shown that the steepness of the skip can be controlled by modulating the time at which PredGuid takes over; starting earlier results in a steeper, higher altitude skip whereas starting later results in a shallower, lower altitude skip. Each of these options has its advantages and disadvantages. Determining which option is preferable and what is the optimal start time may be the topic of future work.

7.2 Future Work

Although this thesis has shown the feasibility of using skip trajectories to extend precision landing, there remains a significant amount of work to generate flight-ready code.

First of all, additional work needs to be done to ensure the flight readiness of PredGuid. For this thesis, the heuristics for the corrector were based on the certain assumptions involving an aerocapture situation. A skip trajectory requires different assumptions which must be reflected in the corrector to ensure PredGuid's convergence to an acceptable bank angle command. PredGuid's convergence can also be improved by

updating the final phase range estimate at each predictor cycle, rather than at each guidance cycle. Additional accuracy in PredGuid's predictions can be achieved by modeling bank reversals in the predictor and by implementing PredGuid's lift-to-drag ratio estimator. Computational efficiency may be improved by increasing PredGuid's integration step size and then interpolating estimated range to the 0.2-g Ballistic phase cutoff, rather than just taking the results of the last integration step.

The robustness of the phases preceding PredGuid should be investigated; they were not designed for such long target ranges and thus certain assumptions in their design may not hold true for the CEV application. For example, the first controlled phase of the algorithm (Initial Roll) makes a bank angle command based on the steepness of the reentry flight path angle; the logic used to make this decision seems to be tuned for the Apollo Command Module but it should be generalized to work with other vehicles. Another example is that the energy management phase (Huntest) may enter an infinite loop if it appears that the vehicle does not have sufficient energy to reach the desired target range. In fact, these assumptions may no longer be accurate, and the logic needs to be updated to be able to handle the long target ranges.

The fidelity of the Monte Carlo process could be improved by updating atmospheric models used by the simulation environment. The simulations used for this thesis used a fairly simple method of modeling atmospheric uncertainties, yet atmospheric estimation capability was shown to be important to the success of the algorithm; a more complex, realistic method of modeling atmospheric uncertainties, such as the GRAM-99 atmospheric model, should be applied to the algorithm to verify its robustness. In addition, the high loft version of the algorithm causes the vehicle to spend a significant amount of time at high altitudes where continuum flow descriptions of the airflow may no longer be accurate; particle flow models should be investigated and implemented.

After these improvements to the algorithm have been made, certain steps should be taken to verify the robustness of the algorithm. Iterative processes – such as PredGuid's predictor-corrector sequencer and the energy management phase's phase-transition logic – must be verified to converge. The effects of flight control on the algorithm should be investigated in order to confirm that a more sluggish response to the bank angle command will still result in acceptable landing precision.

Appendix A

Apollo Guidance Algorithm

A.1 Baseline Algorithm

The documentation for the original Apollo algorithm can be found in “Reentry Guidance for Apollo” [9]. The implementation used for this thesis is exactly as described in that document, with the following exceptions:

1. The Hunttest and Upcontrol phases were implemented as described in appendix J, which is an update to the versions in the main text of the document.
2. This thesis assumed perfect navigation. Thus, the Navigation subroutine was not used; instead, actual state vectors for position and velocity were provided directly to the Targeting subroutine.
3. During the Ballistic phase (the KEP2 subroutine), the algorithm instructs the vehicle to “maintain attitude control.” This was interpreted to mean that the bank angle command is maintained at the previous value until the phase is over.
4. The Final phase (the PREDICT3 subroutine) shuts off steering below a certain velocity. When this occurs, the bank command is set to neutral ($L/D = 0$) for projection to the ground.
5. After calculating THETA (the angle between the current position vector and the target position vector) in the Targeting subroutine, the algorithm should calculate THETNM (the range between the current position vector and the target position vector.) This calculation was omitted in the document but was implemented in the code used for this thesis. THETNM is calculated as follows:

$$THETNM = THETA \cdot ATK \tag{A.1}$$

where:

ATK : Earth radius [nm]

6. In the lateral logic subroutine (named “310” in the document) the calculation for Y (the lateral angle deadband allowance) is as follows:

$$Y = KLAT \cdot VSQ + LATBIAS \quad (A.2)$$

and:

$$VSQ = V^2 / VSAT^2 \quad (A.3)$$

where:

Y : Lateral deadband allowance [rad]
 $KLAT$: Lateral switch gain [rad]
 $LATBIAS$: Lateral deadband bias [nm]
 V : Current velocity [fps]
 $VSAT$: Satellite velocity at Earth's surface [fps]

However, this formulation does not make sense as the units for $KLAT \cdot VSQ$ [rad] and $LATBIAS$ [nm] do not agree. In order to bring the units into agreement with each other, and with further calculations in the subroutine, the calculation was modified as follows:

$$Y = KLAT \cdot VSQ + LATBIAS / ATK \quad (A.4)$$

A.2 Guidance Interfaces

The guidance interfaces are slightly different from those specified by “Reentry Guidance for Apollo” due to the assumptions and implementation interpretations described above. Most of the variables and constants used in guidance are in English units. Abbreviations for these units are given in Table A.1.

Table A.1: Units Abbreviations

| <u>Abbreviation</u> | <u>Description</u> |
|---------------------|--------------------------|
| deg | degrees |
| ft | feet |
| fps | feet per second |
| fpss | feet per second, squared |
| nd | non-dimensional |
| nm | nautical miles |
| psf | pounds per square foot |
| rad | radians |
| sec | seconds |

Table A.2: Apollo Guidance Inputs

| <u>Name</u> | <u>Description</u> | <u>Units</u> | <u>Values</u> |
|-------------|---|--------------|--------------------------|
| Rbar | Inertial position vector | ft | Real number [3] |
| Vlbar | Inertial velocity vector | fps | Real number [3] |
| Alt | Altitude | ft | Real number |
| Time | Time since start of reentry | sec | Non-negative real number |
| Dbar | Inertial aerodynamic drag (magnitude of lift vector plus drag vector) | fpss | Real number [3] |
| R_ini | Initial position vector | ft | Real number [3] |
| V_ini | Initial velocity vector | fps | Real number [3] |
| RT0bar | Target position vector at time of EI | ft | Real number [3] |

Table A.3: Apollo Guidance Outputs

| <u>Name</u> | <u>Description</u> | <u>Units</u> | <u>Values</u> |
|-------------|-------------------------------------|--------------|---|
| Phi | Bank angle command | rad | Real number: $2n\pi$: lift up; $(2n+1)\pi$: lift down |
| QIND | Quit indicator (i.e. stop steering) | integer | 0: continue steering; 1: quit steering |

A.3 Verification

In order to verify that the algorithm had been correctly implemented, results using this implementation were compared to a test case from “Reentry Guidance for Apollo.” This test case, from chapter 5 of “Reentry Guidance for Apollo,” uses the following assumptions:

1. Initial altitude: 400,000 ft
2. Initial velocity: 36,200 fps
3. Initial flight path angle: -6 deg
4. Target range: 2000 nm
5. Initial latitude: -12.7 deg
6. Initial longitude: 122.9 deg east
7. Initial azimuth: 61 deg
8. Vehicle L/D: 0.34
9. Vehicle ballistic coefficient: 66 psf

Figure A.1 shows the superimposed acceleration, altitude, and velocity histories for the two sets of results, and Figure A.2 shows roll angle, lateral angle (the angle between the current velocity heading and the direction to the target location), and the lateral switch indicator. Inertial range and roll fuel data were provided by “Reentry Guidance for Apollo,” but those data were not compared.

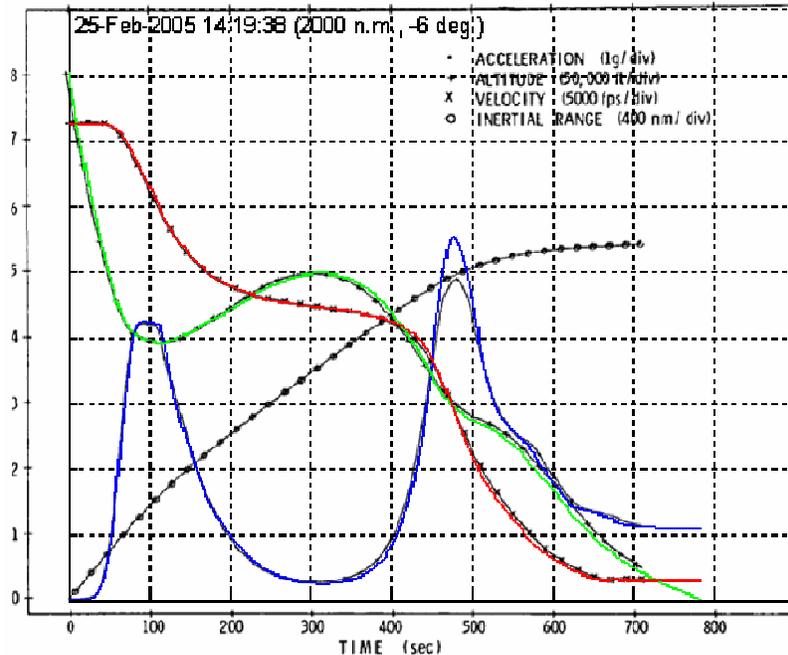


Figure A.1: Comparison to Apollo Test Case; Acceleration, Altitude, and Inertial Downrange [9]

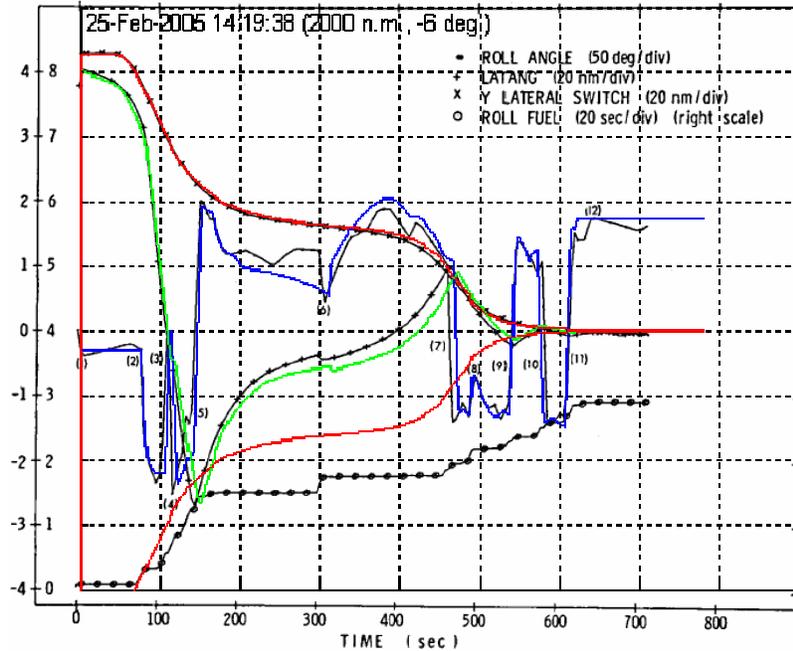


Figure A.2: Comparison to Apollo Test Case; Roll Angle, Lateral Angle, and Lateral Switch [9]

The two cases compare quite well. In Figure A.1, the only noticeable difference is that the second acceleration peak is higher for this thesis’s implementation than for the test case represented in “Reentry Guidance for Apollo.” In Figure A.2, the bank angle histories show some slight differences but they are qualitatively the same. The remaining differences can probably be attributed to the fact that the original simulation used a different atmospheric model than the simulation used for this thesis, and the original simulation also appears to have modeled some additional roll dynamics not modeled by this thesis.

[This Page Intentionally Left Blank]

Appendix B

Enhanced Algorithm with PredGuid

B.1 Overview

This algorithm is derived from the original Apollo algorithm. Some of the subroutines have not changed, and some have changed only minimally. Thus, frequent reference will be made in the following sections to the subroutine descriptions in chapter 4 of “Reentry Guidance for Apollo” [9].

The portions of the PredGuid algorithm used for this thesis were based on the PredGuid algorithm from “Aerocapture Guidance Methods for High Energy Trajectories” [4]. Thus, reference will also be made to the subroutine descriptions in Appendix A of that document.

B.2 Reentry Guidance Executive

The functional flow of the Reentry Guidance Executive is illustrated in Figure B.1. The inputs and outputs used for this version of the algorithm are similar to those used in the original Apollo algorithm outlined in Appendix A, and are described in Table B.1 and Table B.2, respectively. The state variables and guidance constants are described in Table B.3 and Table B.4, respectively.

B.2.1 Guidance Functional Flow

1. If this is the first guidance cycle, initialize the guidance variables. Otherwise, proceed directly to step 2.
2. Execute the targeting subroutine to calculate the remaining range to go and lateral angle to target.

3. Based on the current phase setting (stored in the variable, SELECTOR) execute the appropriate phase subroutine to determine the appropriate bank command. Follow any 'goto' commands within that subroutine.
4. Execute the lateral logic subroutine to determine whether a bank reversal is necessary.

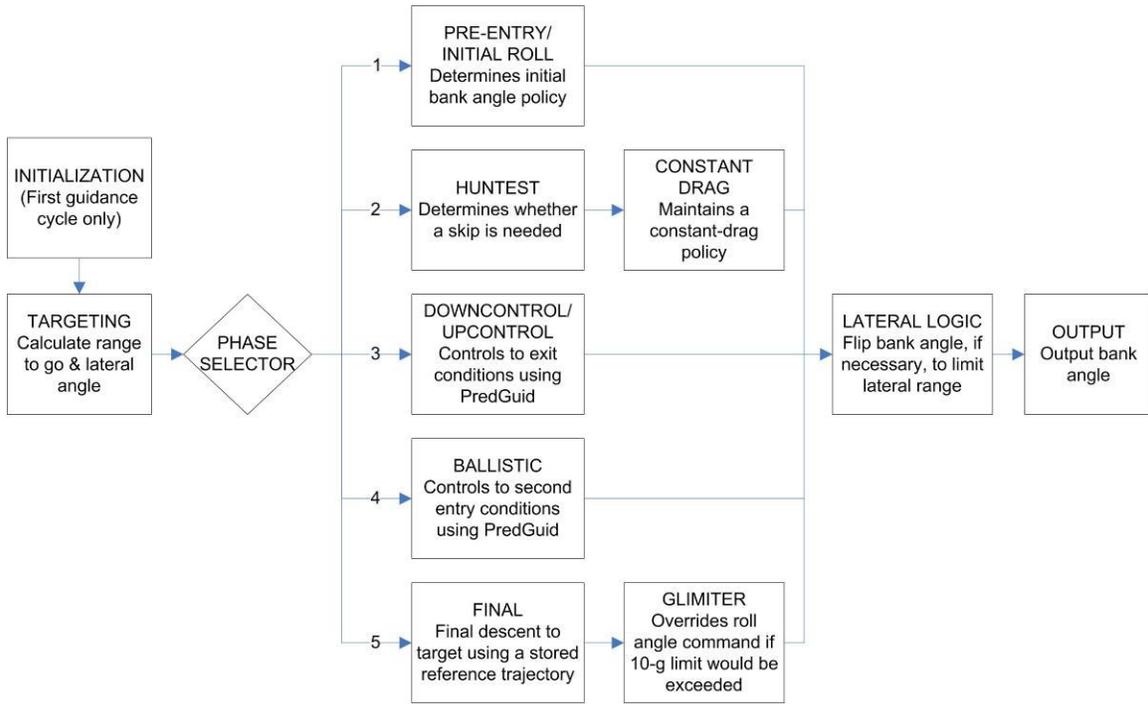


Figure B.1: Reentry Guidance Functional Flow

B.2.2 Inputs

Table B.1: Reentry Guidance Inputs

| Name | Description | Units | Values |
|-----------|---|-------|--------------------------|
| Rbar | Inertial position vector | ft | Real number [3] |
| Vlbar | Inertial velocity vector | fps | Real number [3] |
| Alt | Altitude | ft | Real number |
| Time | Time since start of reentry | sec | Non-negative real number |
| Dbar | Inertial aerodynamic drag (magnitude of lift vector plus drag vector) | fpss | Real number [3] |
| V_rel_mag | Wind-relative velocity magnitude | fps | Real number |
| R_ini | Initial position vector | ft | Real number [3] |
| V_ini | Initial velocity vector | fps | Real number [3] |
| RT0bar | Target position vector at time of EI | ft | Real number [3] |

B.2.3 Outputs

Table B.2: Reentry Guidance Outputs

| Name | Description | Units | Values |
|------|-------------------------------------|---------|---|
| Phi | Bank angle command | rad | Real number; $\pi \cdot 2n$: lift up; $\pi \cdot (2n+1)$: lift down |
| QIND | Quit indicator (i.e. stop steering) | integer | 0: continue steering; 1: quit steering |

B.2.4 State Variables

Table B.3: Reentry Guidance State Variables

| Name | Description | Units | Initialized value |
|----------|--|-------------|---------------------------------|
| Rbar | Inertial position vector | ft | R_ini: input |
| Vlbar | Inertial velocity vector | fps | V_ini: input |
| Dbar | Aerodynamic acceleration vector | fpss | [0,0,0] |
| RTEbar | Local east unit vector at initial target position | nd | UZbar \times URT0bar |
| UNlbar | Unit vector normal to trajectory plane | nd | unit(Vlbar) \times unit(Rbar) |
| URT0bar | Initial target unit vector | nd | unit(RT0bar: input) |
| URTbar | Rotated target unit vector | nd | URT0bar |
| UTRbar | Unit normal to RTE and UZ | nd | RTEbar \times UZbar |
| UZbar | Unit vector through north pole | nd | (0, 0, 1) |
| Vbar | Velocity vector | fps | Vlbar |
| GONEPAST | Indicates whether current position has overshoot target | nd | 0 |
| RELVELSW | Indicates whether to use inertial or relative velocity | nd | 0 |
| EGSW | Indicates whether final phase has started | nd | 0 |
| HUNTIND | Indicates whether an initial pass through HUNTEST has occurred | nd | 0 |
| HIND | Indicates whether an iteration in HUNTEST has occurred | nd | 0 |
| INRLSW | Indicates whether the initial roll attitude has been set | nd | 0 |
| A0 | Drag level at which to start UPCTRL | fpss | 0 |
| AHOOK | Correction for Upcontrol | nd | 0 |
| ALP | const for UPCONTROL | nd | 0 |
| ASKEP | Kepler range | nm | 0 |
| ASP | predicted range | nm | 0 |
| ASPF | final phase range | nm | 0 |
| ASPDWN | range down to pull-up | nm | 0 |
| ASPUP | up-range | nm | 0 |
| COSG | cosine(GAMMAL) | nd | 0 |
| D | Current total aerodynamic acceleration | fpss | 0 |
| DHOOK | Term in GAMMAL computation | fpss | 0 |
| DIFF | Range difference (THETNM - ASP) | nm | 0 |
| DIFFOLD | Previous guidance cycle's value of DIFF | nm | 0 |
| DR | Reference drag for DOWNCONTROL | fpss | 0 |
| DREFR | Reference drag | fpss | 0 |
| DVL | VS1 - VL | fps | 0 |
| E | Eccentricity of Ballistic phase conic section | nd | 0 |
| F1 | Final phase range sensitivity to drag level | nm/ fpss | 0 |

| | | | |
|-----------------|--|------------|---|
| F2 | Final phase range sensitivity to altitude rate | nm/ fps | 0 |
| F3 | Final phase range sensitivity to L/D | nm | 0 |
| FACT1 | const for UPCONTRL | nd | 0 |
| FACT2 | const for UPCONTRL | nd | 0 |
| GAMMAL | Exit flight path angle | rad | 0 |
| GAMMAL1 | Simple form of GAMMAL | rad | 0 |
| K1ROLL | Indicator for roll switch | nd | 0 |
| K2ROLL | Indicator for roll switch | nd | 0 |
| LATANG | Lateral angle to target | rad | 0 |
| LD | Commanded lift to drag ratio in the vertical plane | nd | 0 |
| LEQ | Excess C.F. over GRAV=(VSQ - V)GS | fpss | 0 |
| LEWD | UPCONTROL L/D | nd | 0 |
| PREDANGL | Predicted range to go during Final phase | nm | 0 |
| Q7 | Drag for end of UPCONCONTROL phase | fpss | 0 |
| RDOT | Current altitude rate | fps | 0 |
| RDOTREF | Reference altitude rate for Final phase | fps | 0 |
| RDTR | Reference altitude rate for DOWNCONTROL | fps | 0 |
| ROLLC | Roll command | rad | 0 |
| RTOGO | Range to go during Final phase | nm | 0 |
| SELECTOR | Indicates current phase | nd | 1 |
| SL | Sine of latitude | nd | 0 |
| T | Time since EI | sec | 0 |
| THETA | Current downrange angle to target | rad | 0 |
| THETNM | Current downrange distance to target | nm | 0 |
| V | Current velocity magnitude | fps | 0 |
| V1 | Velocity at which to start Upcontrol | fps | 0 |
| V1OLD | Previous guidance cycle's value for V1 | fps | 0 |
| VBARS | Normalized exit velocity, squared | nd | 0 |
| VCORR | Velocity correction for V at which to start Upcontrol | fps | 0 |
| VL | Exit velocity | fps | 0 |
| VS1 | Smaller of VSAT or V1 | fps | 0 |
| VSQ | Normalized current velocity, squared | nd | 0 |
| WT | Angle from current position to predicted target location | rad | 0 |
| X | Limit altitude rate in G-LIMITER | fps | 0 |
| Y | Lateral miss limit | rad | 0 |
| NEXTSTEP | Numerical indicator of which subroutine should be carried out next | nd | 1 |
| A1 | Starting drag for Upcontrol | fpss | 0 |
| Krho_est | Cumulative estimated density bias factor | nd | 1 |
| Krho_curr | Current density bias estimate | nd | 1 |
| QUITIND | Indicates whether to stop steering | nd | 0 |
| PG_target_range | Downrange distance which PredGuid targets | nm | 0 |
| PIND | Indicates whether the PredGuid subroutine has been called before | nd | 0 |

Note that after initializing the state variables to the above values, the targeting subroutine is run twice to start convergence of the LATANG value to the correct sign. After this, K2ROLL is set to:

$$K2ROLL = -\text{sign}(LATANG)$$

(B.1)

B.2.5 Constants

Table B.4: Reentry Guidance Outputs

| <u>Name</u> | <u>Description</u> | <u>Original Value</u> | <u>New Value</u> | <u>Units</u> |
|-------------|--|-----------------------|------------------|--------------|
| C1 | Factor in ALP computation | 1.25 | | nd |
| C16 | CONSTD gain on drag | 0.01 | | nd |
| C17 | CONSTD gain on RDOT | 0.001 | | nd |
| C18 | Bias vel. for final phase start | 500 | | fps |
| C19 | Minimum constant drag | 130 | | fpss |
| C20 | Max drag for Downcontrol | 175 | | fpss |
| CHOOK | Factor in AHOOK computation | 0.25 | | nd |
| CH1 | Factor in GAMMAL computation | 0.75 | | nd |
| GMAX | Maximum acceleration | 322 | | fpss |
| KA | Drag to roll up if down initially (=KAT) | 64.4 | | fpss |
| KDMIN | Increment on Q7 to detect end of Kepler phase | 0.5 | | fpss |
| KLAT | Lateral switch gain | 0.0125 | | |
| KTETA | Time of flight constant | 1000 | | sec/rad |
| K44 | Gain used in initial roll section | 44389312 | | fps |
| LAD | Max L/D (min actual vehicle L/D) | 0.3 | 0.35 | nd |
| LATBIAS | Lateral switch bias term | 0.4 | | nm |
| LDCMINR | LAD*cos(15 deg) | 0.2895 | 0.3381 | nd |
| LOD | Final phase L/D | 0.18 | 0.21 | nd |
| Q2 | Final phase range for zero vel., according to linear expansion | -1002 | -4276.5 | nm |
| Q3 | Final phase range sensitivity to velocity | 0.07 | 0.21245 | nm/fps |
| Q5 | Final phase range sensitivity to flight path angle | 7050 | 18932 | nm/rad |
| Q6 | Final phase initial flight path angle | 0.0349 | 0.017453 | rad |
| Q7F | Min drag for UPCONTROL | 6 | | fpss |
| VFINAL | Velocity to start final phase on initial entry | 25000 | | fps |
| VLMIN | Minimum VL | 18000 | | fps |
| VMIN | Velocity to switch to relative vel | VSAT/2 | | fps |
| VRCONTRL | RDOT to start into HUNTEST | 700 | | fps |
| VCORLIM | Max value of VCORR | 1000 | | fps |
| TOL | Tolerance to stop range iteration | 25 | | nm |
| VQUIT | Velocity to stop steering | 1000 | | fps |
| ATK | Angle in rad to NM | 3437.7468 | | nm/rad |
| GS | Nominal G value for scaling | 32.2 | | fpss |

| | | | | |
|-----------|---|-----------------|---------------|---------|
| HS | Atmosphere scale height | 28500 | | ft |
| KWE | Equatorial Earth rate | 1546.70168 | | fps |
| RE | Earth radius | 21202900 | | ft |
| VSAT | Satellite velocity at RE | 25766.1973 | | fps |
| WIE | Earth rate | 0.0000729211505 | | rad/sec |
| LEWD1 | Initial UPCONTROL ref. L/D | 0.1 | | nd |
| LEWD2 | Later UPCONTROL ref. L/D | 0.2 | | nd |
| D2 | Changeover drag from LEWD1 to LEWD2 | 175 | | fpss |
| D0 | Controlled constant drag | 130 | | fpss |
| Q7MIN | Max value for Q7 | 35 | | fpss |
| ASENSIBLE | Aero acceleration at which 'sensible atmosphere' has been reached | 1.6 | | fpss |
| FTABLE | Table for final phase; data structure | See Table B.5 | See Table B.6 | n/a |
| Q19 | Weighting for GAMMAL1 calc. | 0.2 | | nd |
| c | Data structure of constants used primarily for PredGuid | n/a | See Table B.7 | n/a |

Table B.5: Original Apollo Final Phase Reference Trajectory Table

| VREF | RDOTREF | DREFR | F2 | F1 | RTOGO | F3 |
|-------|---------|--------|----------|-----------|-------|--------|
| (fps) | (fps) | (fpss) | (nm/fps) | (nm/fpss) | (nm) | (nm) |
| 0 | -331 | 34.1 | 0 | -0.02695 | 0 | 1 |
| 337 | -331 | 34.1 | 0 | -0.02695 | 0 | 1 |
| 1080 | -693 | 42.6 | 0.002591 | -0.03629 | 2.7 | 12.88 |
| 2103 | -719 | 60 | 0.003582 | -0.05551 | 8.9 | 21.82 |
| 3922 | -694 | 81.5 | 0.007039 | -0.09034 | 22.1 | 43.28 |
| 6295 | -609 | 93.9 | 0.01446 | -0.141 | 46.3 | 96.7 |
| 8531 | -493 | 98.5 | 0.02478 | -0.1978 | 75.4 | 187.44 |
| 10101 | -416 | 102.3 | 0.03391 | -0.2372 | 99.9 | 282.2 |
| 14014 | -352 | 118.7 | 0.06139 | -0.3305 | 170.9 | 329.4 |
| 15951 | -416 | 125.2 | 0.07683 | -0.3605 | 210.3 | 465.5 |
| 18357 | -566 | 120.4 | 0.09982 | -0.4956 | 266.8 | 682.7 |
| 20829 | -781 | 95.4 | 0.1335 | -0.6483 | 344.3 | 980.5 |
| 23090 | -927 | 28.1 | 0.2175 | -2.021 | 504.8 | 1385 |
| 23500 | -820 | 6.4 | 0.3046 | -3.354 | 643 | 1508 |
| 35000 | -820 | 6.4 | 0.3046 | -3.354 | 794.3 | 1508 |

The new Final phase reference trajectory is calculated as described in Chapter 5, assuming the vehicle characteristics described in Chapter 2. After the $\vec{\lambda}(t)$ vector is calculated, the following variable transformations are applied to arrive at the sensitivities F1, F2, and F3, to the variables drag, altitude rate, and L/D, respectively, used in the reference trajectory lookup table:

$$F1(t) = -HS*\lambda_4(t)/D(t) = \frac{\partial X}{\partial D}(t) \quad (B.2)$$

$$F2(t) = \lambda_3(t)/V(t) = \frac{\partial X}{\partial (RDOT)}(t) \quad (B.3)$$

$$F3(t) = \lambda_5(t) = \frac{\partial X}{\partial (L/D)}(t) \quad (B.4)$$

Constants Q2, Q3, Q5, and Q6 must be recalculated to agree with the new Final phase reference trajectory, as follows:

$$Q2 = X(t_0) - \lambda_2(t_0) \cdot V(t_0) \quad (B.5)$$

$$Q3 = \lambda_2(t_0) \quad (B.6)$$

$$Q5 = \lambda_3(t_0) \quad (B.7)$$

$$Q6 = -\gamma(t_0) \quad (B.8)$$

Table B.6: New Final Phase Reference Trajectory Table for Enhanced Reentry Guidance

| VREF | RDOTREF | DREFR | F2 | F1 | RTOGO | F3 |
|-------|---------|--------|----------|-----------|----------|--------|
| (fps) | (fps) | (fpss) | (nm/fps) | (nm/fpss) | (nm) | (nm) |
| 0 | -230.68 | 34.219 | 7.54E-05 | -0.03546 | 0.003927 | 1 |
| 236 | -230.68 | 34.219 | 7.54E-05 | -0.03546 | 0.003927 | 1 |
| 862 | -660.22 | 42.245 | 0.002553 | -0.03072 | 3.4691 | 18.295 |
| 1487 | -702.79 | 50.763 | 0.003253 | -0.05057 | 7.4404 | 24.982 |
| 2113 | -679.37 | 60.767 | 0.004175 | -0.06528 | 11.696 | 31.059 |
| 2739 | -650.97 | 70.409 | 0.005342 | -0.07676 | 16.218 | 37.987 |
| 3364 | -627.14 | 79.309 | 0.006714 | -0.08658 | 21.014 | 46.297 |
| 3990 | -608.56 | 86.488 | 0.008289 | -0.09333 | 26.128 | 56.361 |
| 4616 | -593.43 | 92.557 | 0.01008 | -0.10427 | 31.608 | 68.483 |
| 5241 | -581.75 | 97.704 | 0.012064 | -0.11493 | 37.458 | 82.923 |
| 5867 | -573.14 | 101.93 | 0.014238 | -0.12565 | 43.715 | 99.974 |
| 6493 | -567.11 | 105.21 | 0.016599 | -0.13665 | 50.396 | 119.86 |
| 7118 | -563 | 107.52 | 0.019151 | -0.1482 | 57.529 | 142.8 |
| 7744 | -560.07 | 108.85 | 0.021914 | -0.1606 | 65.181 | 169.11 |
| 8370 | -557.53 | 109.22 | 0.024903 | -0.17412 | 73.399 | 199.04 |
| 8995 | -554.57 | 108.65 | 0.028139 | -0.18906 | 82.237 | 232.84 |
| 9621 | -550.32 | 107.19 | 0.031664 | -0.20582 | 91.811 | 270.97 |
| 10246 | -543.93 | 104.94 | 0.035505 | -0.22474 | 102.18 | 156.84 |
| 10872 | -534.58 | 101.99 | 0.039719 | -0.24629 | 113.49 | 180.76 |
| 11498 | -521.47 | 98.491 | 0.044358 | -0.27089 | 125.86 | 207.48 |
| 12123 | -504.59 | 95.413 | 0.04944 | -0.31708 | 139.34 | 237.23 |
| 12749 | -484.8 | 92.244 | 0.05497 | -0.34261 | 154 | 270.36 |
| 13375 | -461.93 | 88.864 | 0.061012 | -0.37169 | 169.96 | 307.15 |
| 14000 | -436.05 | 85.421 | 0.067631 | -0.4206 | 187.29 | 347.83 |
| 14626 | -408.27 | 82.869 | 0.074858 | -0.46954 | 206.08 | 392.94 |
| 15252 | -380.15 | 80.383 | 0.082694 | -0.50208 | 226.28 | 442.78 |
| 15877 | -352.5 | 78.015 | 0.091211 | -0.53858 | 247.92 | 497.64 |
| 16503 | -326.17 | 75.797 | 0.10053 | -0.57955 | 271.12 | 558.18 |
| 17129 | -302.25 | 73.742 | 0.11075 | -0.62545 | 295.88 | 624.83 |
| 17754 | -281.91 | 71.832 | 0.12197 | -0.67694 | 322.21 | 698.07 |
| 18380 | -266.14 | 69.999 | 0.13437 | -0.73528 | 350.22 | 778.82 |
| 19006 | -255.98 | 68.151 | 0.14812 | -0.80222 | 379.96 | 867.78 |
| 19631 | -252.19 | 66.157 | 0.16342 | -0.88035 | 411.52 | 965.7 |
| 20257 | -255.29 | 63.845 | 0.18066 | -0.9742 | 445.22 | 1074 |
| 20882 | -265.54 | 61.031 | 0.20026 | -1.0903 | 481.35 | 1193.7 |
| 21508 | -282.96 | 57.488 | 0.22299 | -1.2403 | 520.65 | 1327 |
| 22134 | -307.2 | 52.985 | 0.24995 | -1.4438 | 564.09 | 1476 |
| 22759 | -337.54 | 47.281 | 0.28297 | -1.738 | 613.32 | 1643.8 |
| 23385 | -372.79 | 40.091 | 0.32568 | -2.2064 | 671.64 | 1836 |
| 24011 | -410.49 | 31.14 | 0.3856 | -3.069 | 745.54 | 2061 |
| 24636 | -444.44 | 20.128 | 0.48396 | -5.1679 | 852.47 | 2336.2 |
| 25262 | -440.95 | 6.3697 | 0.74897 | -18.237 | 1090.1 | 2722.7 |
| 35000 | -440.95 | 6.3697 | 0.74897 | -18.237 | 1090.1 | 2722.7 |

Table B.7: PredGuid Constants

| <u>Name</u> | <u>Original Value</u> | <u>New Value</u> | <u>Units</u> |
|------------------------|---|------------------------|---------------------------------|
| APOGEE_EPSILON_TIGHTER | 40 | | nm |
| APOGEE_EPSILON_TIGHT | 3 | 25 | nm |
| APOGEE_TARGET | 216 | overwritten each cycle | nm |
| ATMOS_ALT_MAX | 400000 | 600000 | ft |
| BANK_MAX | 165 | 180 | deg |
| BANK_MIN | 15 | 0 | deg |
| CD_ALPHA | 0 | | nd |
| CD_ALPHA_SQ | 0 | | nd |
| CD_EST_INITIAL | 1.4286 | 1.25 | nd |
| DEG_TO_RAD | $\pi/180$ | | rad/deg |
| DELTA_PHI_MIN | 0 | | deg |
| DELTA_T_PRED | 10 | | sec |
| EARTH_FLAT | 1/298.3 | | nd |
| EARTH_J2 | 1082.63e-6 | | nd |
| EARTH_MU | $3.986005e14/0.3048^3$ | | ft ³ /s ² |
| EARTH_POLE | [0.0042772340; -0.0000901672; 0.9999908485] | UZbar | nd |
| EARTH_R | 6378140/0.3048 | | ft |
| EARTH_RATE | 7.292114883223324e-5 | | rad/ sec |
| INFINITY | 99999999 | | nd |
| K_RHO_EST_INITIAL | 1.0 | | nd |
| K_RHO_FILTER_GAIN | 0.05 | | nd |
| L_OVER_D_EST_INITIAL | L_D | 0.35 | nd |
| LIFT_INC_CAPTURE | 0.25 | -3 | deg |
| LIFT_PERCENT_CAPTURE | 0.5 | | nd |
| MASS_EI | 15.4783 | 650.9524 | slugs |
| MAX_NUMBER_RUNS | 10 | | nd |
| RAD_TO_DEG | $180/\pi$ | | deg/rad |
| S_REF | 12.163 | 256.18 | ft ² |
| VI_LOOSE_APOGEE | 26600 | | fps |

B.3 Targeting Subroutine

This corresponds to the TARGETING subroutine described in “Reentry Guidance for Apollo”, and has remained unchanged except for an additional function to estimate the atmospheric density bias factor. This calculation is based roughly on the “aero_properties” subroutine described in “Aerocapture Guidance Methods for High Energy Trajectories.” The logical flow for this subroutine is illustrated in Figure B.2.

1. Calculate the velocity vector, Vbar.
 - a. If RELVELSW is zero, calculate the current velocity vector in inertial space:

$$\mathbf{Vbar} = \mathbf{VIbar} \quad (\text{B.9})$$

- b. Otherwise, calculate the current velocity vector in an Earth-fixed, rotating frame:

$$\mathbf{Vbar} = \mathbf{VIbar} - \mathbf{KWE} \cdot \mathbf{UZbar} \cdot \text{unit}(\mathbf{Rbar}) \quad (\text{B.10})$$

2. Calculate various control parameters from current conditions:

$$\begin{aligned} V &= |\mathbf{Vbar}| \\ VSQ &= V^2/VSAT^2 \\ LEQ &= (VSQ - 1) \cdot GS \\ RDOT &= \mathbf{Vbar} \cdot \text{unit}(\mathbf{Rbar}) \\ \text{UNIbar} &= \text{unit} [\mathbf{Vbar} \times \text{unit}(\mathbf{Rbar})] \\ D &= |\mathbf{Dbar}| \end{aligned} \quad (\text{B.11})$$

3. Estimate the density bias factor:

$$\begin{aligned} \rho &= \frac{2 \cdot c.MASS_EI \cdot D}{V_rel_mag^2 \cdot c.S_REF \cdot c.CD_EST_INITIAL \cdot \sqrt{1 + LAD^2}} \\ \rho_std &= f(\text{alt}): \text{atmosphere model table lookup by altitude} \\ K\rho_now &= \rho/\rho_std \\ K\rho_est &= c.K_RHO_FILTER_GAIN \cdot K\rho_now \\ &\quad + (1 - c.K_RHO_FILTER_GAIN) \cdot \text{DATA.K}\rho_est \end{aligned} \quad (\text{B.12})$$

4. Estimate the angle by which the target will have rotated due to the rotation of the Earth between the start of entry and the time the vehicle reaches the target.
 - a. If RELVELSW is zero, the inertial velocity calculation is being used.
 - i. If EGSW is zero, then the Final phase has been reached. Use Method 1 to estimate the rotation of the Earth:

$$WT = WIE \cdot (KTETA \cdot THETA + T) \quad (B.13)$$

Move on to step 5.

- ii. Otherwise, use Method 2 to estimate the rotation of the Earth:

$$WT = WIE \cdot \left(\frac{RE \cdot THETA}{V} + T \right) \quad (B.14)$$

Also, check whether the current velocity, V , is low enough to switch to the relative velocity calculation.

1. If $V < VMIN$, then indicate a switch to the relative velocity calculation:

$$RELVELSW=1 \quad (B.15)$$

Move on to step 5.

- b. Otherwise, if RELVELSW is not zero, the relative velocity calculation is being used. Estimate the rotation of the Earth using Method 3:

$$WT = WIE \cdot T \quad (B.16)$$

Move on to step 5.

5. Calculate the rotated target vector, crossrange angle, and downrange to go:

$$\begin{aligned} URTbar &= URT0bar + UTRbar \cdot [\cos(WT) - 1] \\ &\quad + RTEbar \cdot \sin(WT) \\ LATANG &= URTbar \cdot UNIbar \\ THETA &= \cos^{-1}[URTbar \cdot unit(R)] \\ THETNM &= THETA \cdot ATK \end{aligned} \quad (B.17)$$

6. Go to the appropriate phase's subroutine based on the SELECTOR variable:
- If SELECTOR = 1, then go to the INITIAL ROLL subroutine.
 - If SELECTOR = 2, then go to the HUNTEST subroutine.
 - If SELECTOR = 3, then go to the UPCONTROL subroutine.
 - If SELECTOR = 4, then go to the BALLISTIC subroutine.
 - If SELECTOR = 5, then go to the FINAL subroutine.

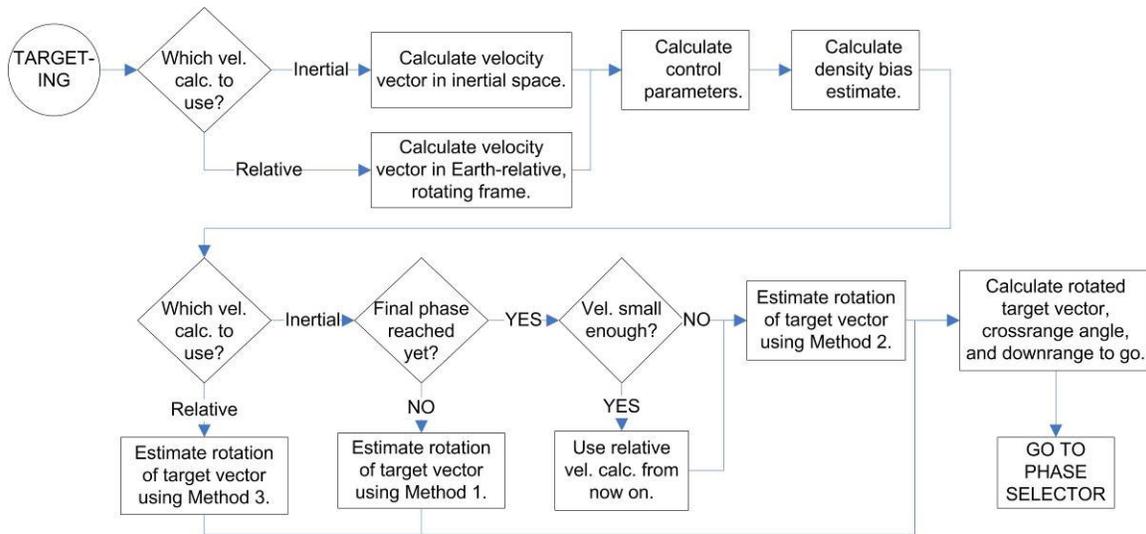


Figure B.2: Targeting Logical Flow

B.4 Pre-entry Attitude Hold and Initial Roll Phases

These two phases are contained within the INTROLL subroutine described in “Reentry Guidance for Apollo”. This subroutine has not been altered from the original Apollo algorithm. This subroutine is illustrated in Figure B.3.

1. Check whether the atmosphere has been entered yet.
 - a. If $INRLSW = 0$, drag has not yet increased above $0.5 g$'s. The atmosphere has not yet been entered, so carry through with the Pre-entry Attitude Hold Phase (steps 2-5).
 - b. Otherwise, the atmosphere has been reached. Skip over the Pre-entry Attitude Hold and skip to the Initial Roll Phase (steps 6-7).
2. Check whether the atmosphere has been entered during this guidance cycle.
 - a. If $D < ASENSIBLE$, then the atmosphere has not yet been reached. Skip to Step 5.
3. The atmosphere has just been reached in this iteration. Check whether the current velocity is high enough to carry through with a normal reentry.
 - a. If $V < VFINAL$, then the current velocity is too low. Indicate a switch to the Ballistic phase and set the L/D command to full lift up:

$$\begin{aligned} \text{SELECTOR} &= 4 \text{ (Ballistic Phase)} \\ \text{LD} &= \text{LAD} \end{aligned} \tag{B.18}$$

Skip to Step 5.

4. Determine the appropriate initial attitude for entering into the atmosphere.
 - a. If $V > V_{FINAL} - K44 \cdot (RDOT/V)^3$, then the entry is too shallow. Command full lift down to steepen the trajectory:

$$LD = -LAD \quad (B.19)$$

- b. Otherwise, the trajectory is steep enough. Command full lift up:

$$LD = LAD \quad (B.20)$$

5. Go to the Lateral Logic subroutine.
6. The current phase is the Initial Roll Phase. If drag is too high ($D > KA$), command full lift up:

$$LD = LAD \quad (B.21)$$

7. Check whether the altitude rate is shallow enough to move to the Hunttest phase.
 - a. If $RDOT < -VRCONTRL$, then the altitude rate is too steep to move to the Hunttest phase. Skip the rest of this subroutine and go to the Lateral Logic subroutine.
 - b. Otherwise, the altitude rate is small enough. Indicate a switch to the Hunttest phase:

$$SELECTOR = 2 \text{ (Hunttest Phase)} \quad (B.22)$$

Go to the Hunttest subroutine.

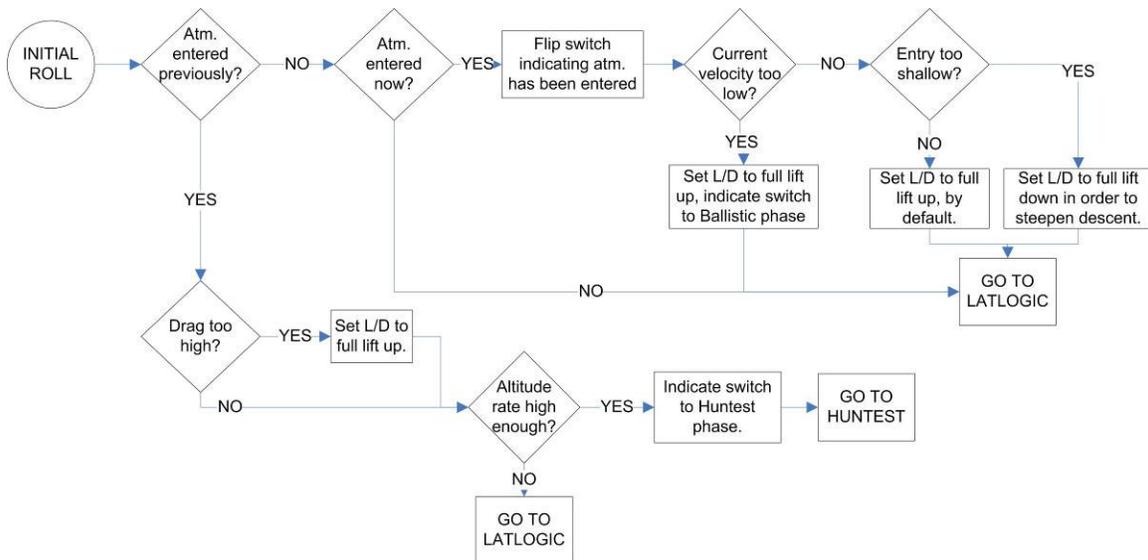


Figure B.3: Initial Roll Logical Flow

B.5 Hunttest/Constant Drag Phase

This phase corresponds to the HUNTEST, HUNTEST1, RANGE PREDICTION, and CONSTD subroutines described in “Reentry Guidance for Apollo”. These subroutines have remained unchanged except for a new Final phase range estimation technique which replaces the old technique in the RANGE PREDICTION subroutine. The logical flow for the HUNTEST/HUNTEST1/RANGE PREDICTION combined subroutine is illustrated in Figure B.4, and the logical flow for the constant drag controller is illustrated in Figure B.5.

1. Decide which value to use for the reference L/D.

- c. If $D > D_2$, use set LEWD1 and the reference value:

$$LEWD = LEWD1 \quad (B.23)$$

- d. Otherwise, set LEWD2 as the reference value:

$$LEWD = LEWD2 \quad (B.24)$$

2. Estimate the conditions for the start of the Upcontrol phase. Check whether altitude is decreasing.

- a. If altitude rate is currently decreasing ($RDOT < 0$), project the trajectory conditions to pullout using the reference L/D:

$$\begin{aligned} V1 &= V + RDOT/LEWD \\ A0 &= (V1/V)^2 (D + RDOT^2/(2 \cdot HS \cdot LEWD)) \\ A1 &= D \end{aligned} \quad (B.25)$$

- b. Otherwise, project forward assuming full lift up:

$$\begin{aligned} V1 &= V + RDOT/LAD \\ A0 &= (V1/V)^2 (D + RDOT^2/(2 \cdot HS \cdot LAD)) \\ A1 &= A0 \end{aligned} \quad (B.26)$$

3. If this is the first time through the Hunttest subroutine ($HUNTIND = 0$), then initialize some Hunttest variables:

$$\begin{aligned} HUNTIND &= 1 \\ DIFFOLD &= 0 \\ V1OLD &= V1 + C18 \\ Q7 &= Q7F \end{aligned} \quad (B.27)$$

4. Calculate the exit velocity, VL, for the end of the Upcontrol phase.

$$\begin{aligned}
ALP &= 2 \cdot C1 \cdot A0 \cdot HS / (LEWD \cdot V1^2) \\
FACT1 &= V1 / (1 - ALP) \\
FACT2 &= ALP \cdot (ALP - 1) / A0 \\
VL &= FACT1 \cdot [1 - \sqrt{FACT2 \cdot Q7 + ALP}]
\end{aligned}
\tag{B.28}$$

5. Check whether the exit velocity is sufficient to carry through a skip trajectory.
- a. If $VL < VL_{MIN}$, the velocity is too low to carry through a skip. Indicate a switch to the Final phase:

$$\begin{aligned}
SELECTOR &= 5 \text{ (Final Phase)} \\
EGSW &= 1
\end{aligned}
\tag{B.29}$$

Skip the rest of this subroutine and go directly to the Final phase subroutine.

- b. Otherwise, Check whether exit velocity is greater than satellite velocity ($VL > VSAT$). If so, the skip energy is too excessive and the vehicle would skip away. Indicate that the vehicle is to remain in the Hunttest phase:

$$SELECTOR = 2 \text{ (Hunttest Phase)} \tag{B.30}$$

Skip the rest of this subroutine and go directly to the Constant Drag subroutine.

6. Set $VS1$ to the smaller of $V1$ or $VSAT$.
7. Calculate the flight path angle, $GAMMAL$, at exit velocity VL :

$$\begin{aligned}
DVL &= VS1 - VL \\
DHOOK &= [(1 - VS1/FACT1)^2 - ALP] / FACT2 \\
AHOOK &= CHOOK (DHOOK/Q7 - 1) / DVL \\
GAMMAL1 &= LEWD (V1 - VL) / VL \\
GAMMAL &= GAMMAL1 - [CH1 \cdot GS \cdot DVL^2 \cdot \\
&\quad (1 + AHOOK DVL)] / [DHOOK VL^2]
\end{aligned}
\tag{B.31}$$

8. If $GAMMAL < 0$, the skip cannot achieve atmospheric exit. Adjust VL and $Q7$ to represent the conditions at the zero flight path angle condition:

$$\begin{aligned}
VL &= VL + \frac{GAMMAL \cdot VL}{LEWD - (3 \cdot AHOOK \cdot DVL^2 + 2 \cdot DVL) \cdot [(CH1 \cdot GS) / (DHOOK \cdot VL)]} \\
Q7 &= [(1 - VL/FACT1)^2 - ALP] / FACT2 \\
GAMMAL &= 0
\end{aligned}
\tag{B.32}$$

9. Calculate the simple form of $GAMMAL$, $GAMMAL1$:

$$GAMMAL1 = GAMMAL1 \cdot (1 - Q19) + Q19 \cdot GAMMAL \tag{B.33}$$

10. Calculate range to touchdown by adding up estimates for individual phases:

- a. Ballistic phase range:

$$\begin{aligned}
VBARS &= VL^2/VSAT^2 \\
COSG &= 1-GAMMAL^2/2 \\
E &= \sqrt{1 + (VBARS - 2) \cdot COSG^2 \cdot VBARS} \\
ASKEP &= 2 \cdot ATK \cdot \sin^{-1}(VBARS \cdot COSG \cdot GAMMAL/E)
\end{aligned}
\tag{B.34}$$

b. Final phase range:

- i. If $V < 25262$ (initial condition for Final phase reference trajectory), then use interpolated-point linear expansion, interpolating values of RTOGO, F2, RDOTREF, F1, and DREFR from the reference trajectory table, FTABLE:

$$\begin{aligned}
ASPF &= RTOGO(VL) + F2(VL) \cdot (-GAMMAL \cdot VL - RDOTREF(VL)) \\
&\quad + F1(VL) \cdot (Q7 - DREFR(VL))
\end{aligned}
\tag{B.35}$$

- ii. If $V > 25262$, then use linear expansion about that point:

$$ASP = Q2 + Q3 \cdot VL + Q5 \cdot (Q6 - GAMMAL)
\tag{B.36}$$

c. Upcontrol phase range:

$$ASPUP = ATK/RE \cdot (HS/GAMMAL1) \cdot \log[A0 \cdot VL^2 / (Q7 \cdot V1^2)]
\tag{B.37}$$

d. Downcontrol phase range:

$$ASPDWN = -RDOT \cdot V \cdot ATK / (A0 \cdot LAD \cdot RE)
\tag{B.38}$$

e. Total range:

$$ASP = ASKEP + ASPF + ASPUP + ASPDWN
\tag{B.39}$$

f. Difference between predicted downrange and desired downrange:

$$DIFF = THETNM - ASP
\tag{B.40}$$

11. If $\text{magnitude}(DIFF) < TOL$, the skip trajectory will land within the tolerance of 25 nautical miles from the target, assuming the vehicle maintains the nominal reference L/D. Indicate a switch to the Upcontrol phase:

$$SELECTOR = 3 \text{ (Downcontrol/Upcontrol Phase)}
\tag{B.41}$$

Skip the rest of this subroutine and go directly to the Upcontrol subroutine.

12. If $HIND = 0$, a previous iteration the algorithm through the section of code which would attempt to change the Upcontrol starting velocity, $V1$, has not occurred. Check whether the predicted range is too far or too close.

- a. If $DIFF < 0$, then the predicted range, ASP , is too far. Store the old values of $DIFF$ and $V1$:

$$\begin{aligned} \text{DIFFOLD} &= \text{DIFF} \\ \text{V1OLD} &= \text{V1} \end{aligned} \tag{B.42}$$

Skip the rest of this subroutine and go directly to the constant drag subroutine in order to bleed off some energy

- b. Otherwise, the predicted range is too close and the vehicle has bled off slightly too much energy. The starting velocity for Upcontrol, V1, should be tweaked in order to bring the predicted range within the tolerance of the desired range. Initialize the velocity correction to apply to V1:

$$\text{VCORR} = \text{V1} - \text{V1OLD} \tag{B.43}$$

13. Calculate the velocity correction, VCORR:

$$\text{VCORR} = \frac{\text{VCORR} \cdot \text{DIFF}}{\text{DIFFOLD} - \text{DIFF}} \tag{B.44}$$

14. Limit the velocity correction to a maximum value:

$$\text{VCORR} = \max(\text{VCORR}, \text{VCORLIM}) \tag{B.45}$$

15. If $(\text{VSAT} - \text{VL}) < \text{VCORR}$, then applying the velocity correction to V1 will cause the exit velocity to exceed the satellite velocity. If this is the case, cut VCORR in half:

$$\text{VCORR} = \text{VCORR}/2 \tag{B.46}$$

16. Apply the velocity correction to the Upcontrol starting velocity. Indicate an iteration through this section of code and record the old value of DIFF:

$$\begin{aligned} \text{V1} &= \text{V1} + \text{VCORR} \\ \text{HIND} &= 1 \\ \text{DIFFOLD} &= \text{DIFF} \end{aligned} \tag{B.47}$$

17. Go back to Step 4.

Constant Drag Controller:

1. Implement the constant drag controller:

$$L/D = -LEQ/D0 + C16(D - D0) - C17(RDOT + 2 \cdot HS \cdot D0/V) \quad (B.48)$$

2. Perform a test to determine whether a negative command for L/D is causing too much drag: $L/D < 0$ and $D > C20$. If this is the case, set the L/D command to a neutral L/D of zero to limit the drag acceleration:

$$LD = 0 \quad (B.49)$$

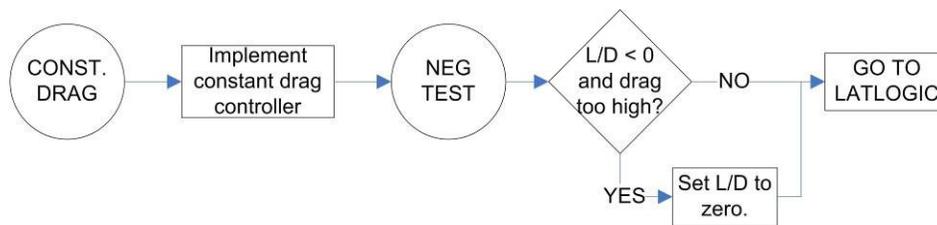


Figure B.5: Constant Drag Logical Flow

B.6 Downcontrol and Upcontrol Phases

These two phases are contained within the UPCTRL subroutine described in “Reentry Guidance for Apollo.” Depending on which version of the algorithm (Low Loft vs. High Loft) is being used, one or both of the original phases, described in “Reentry Guidance for Apollo” has been replaced by the PredGuid Targeting subroutine, described in Aerocapture Guidance Methods for High Energy Trajectories.” This subroutine is illustrated for the Low Loft version of the algorithm in Figure B.6, and for the High Loft version in Figure B.7.

Low Loft Enhanced Version of Upcontrol:

1. If the velocity is still above the starting velocity for Upcontrol ($V > V1$), the Downcontrol phase is still in effect. Calculate the reference altitude rate and drag, and then calculate the L/D command based on trajectory errors:

$$\begin{aligned} RDTR &= LAD \cdot (V1 - V) \\ DR &= (V/V1)^2 \cdot AO - RDTR^2 / (2 \cdot HS \cdot LAD) \\ L/D &= LAD + C16 \cdot (D - DR) - C17 \cdot (RDOT - RDTR) \end{aligned} \quad (B.50)$$

Skip the rest of this subroutine and go directly to the Lateral Logic subroutine.

- If the velocity has reached the exit velocity within some tolerance ($V < VL + C18$) and altitude is decreasing ($RDOT < 0$), then it is time to move to the Final phase. Indicate a switch to the Final phase:

$$\begin{aligned} \text{SELECTOR} &= 5 \text{ (Final Phase)} \\ \text{EGSW} &= 1 \end{aligned} \tag{B.51}$$

Skip the rest of this subroutine and go directly to the Final subroutine.

- If the drag level has gone below 0.2 g's ($D < Q7$), then it is time to move to the Ballistic phase. Indicate a switch to the Ballistic phase:

$$\text{SELECTOR} = 4 \text{ (Ballistic Phase)} \tag{B.52}$$

Skip the rest of this subroutine and go directly to the Ballistic subroutine.

- If the drag level is still higher than the Upcontrol starting drag ($D > A0$), then the vehicle needs to decrease the drag level. Command full lift up:

$$\text{LD} = \text{LAD} \tag{B.53}$$

Skip the rest of this subroutine and go directly to the Lateral Logic subroutine.

- If the algorithm has bypassed all the previous tests, run the PredGuid subroutine, and then proceed to the NEGTEST subroutine which is described as part of the Constant Drag subroutine.

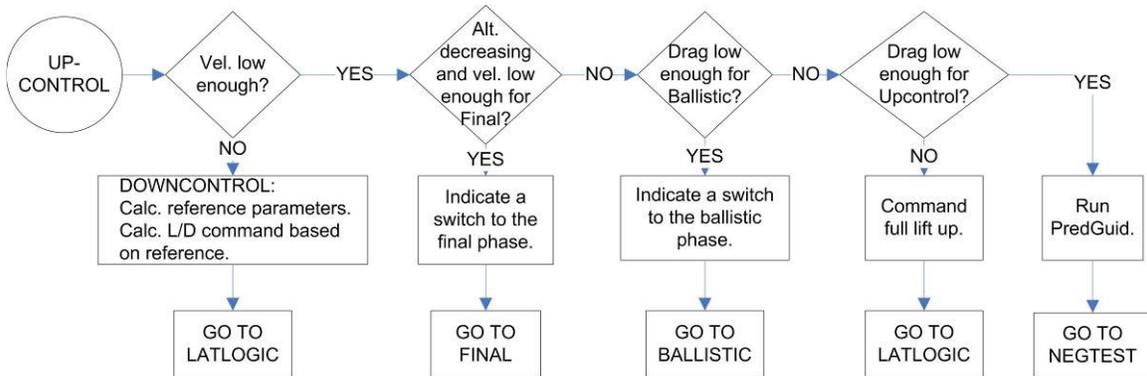


Figure B.6: Upcontrol Logical Flow (Low Loft)

High Loft Enhanced Version of Upcontrol:

- If the velocity has reached the exit velocity within some tolerance ($V < VL + C18$) and altitude is decreasing ($RDOT < 0$), then it is time to move to the Final phase. Indicate a switch to the Final phase:

$$\begin{aligned} \text{SELECTOR} &= 5 \text{ (Final Phase)} \\ \text{EGSW} &= 1 \end{aligned} \tag{B.54}$$

Skip the rest of this subroutine and go directly to the Final subroutine.

2. If the drag level has gone below 0.2 g's ($D < Q7$), then it is time to move to the Ballistic phase. Indicate a switch to the Ballistic phase:

$$\text{SELECTOR} = 4 \text{ (Ballistic Phase)} \tag{B.55}$$

Skip the rest of this subroutine and go directly to the Ballistic subroutine.

3. If the algorithm has bypassed all the previous tests, run the PredGuid subroutine, and then proceed to the NEGTEST subroutine which is described as part of the Constant Drag subroutine.

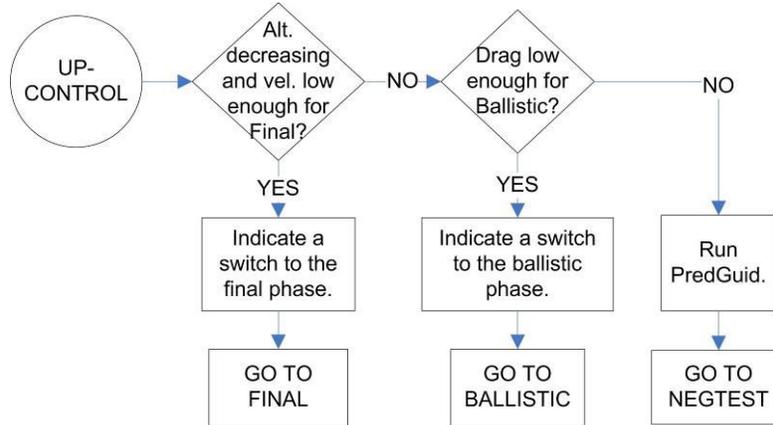


Figure B.7: Upcontrol Logical Flow (High Loft)

B.7 PredGuid Subroutine

This subroutine is based primarily on the “pc_sequencer” subroutine described in “Aerocapture Guidance Methods for High Energy Trajectories.” The functionality of the sequencer itself has not been changed; however, the “predictor” and “corrector” subroutines have been altered to reflect the requirements for a skip reentry, rather than aerocapture. The changes that have been made to the predictor and the corrector are described in the following two sections.

The logical flow for the PredGuid subroutine is illustrated in Figure B.8.

The inputs to pc_sequencer are described in Table B.8.

The outputs from pc_sequencer are described in Table B.9.

1. Calculate the range that needs to be covered during the PredGuid phases (from current time until the start of the Final phase).

- a. Calculate the Final phase range:

- i. If $V < 25262$ (initial condition for Final phase reference trajectory), then use interpolated-point linear expansion, interpolating values of RTOGO, F2, RDOTREF, F1, and DREFR from the reference trajectory table, FTABLE:

$$\text{ASPF} = \text{RTOGO}(\text{VL}) + \text{F2}(\text{VL}) \cdot (-\text{GAMMAL} \cdot \text{VL} - \text{RDOTREF}(\text{VL})) + \text{F1}(\text{VL}) \cdot (\text{Q7} - \text{DREFR}(\text{VL})) \quad (\text{B.56})$$

- ii. If $V > 25262$, then use linear expansion about that point:

$$\text{ASPF} = \text{Q2} + \text{Q3} \cdot \text{VL} + \text{Q5} \cdot (\text{Q6} - \text{GAMMAL}) \quad (\text{B.57})$$

- b. Calculate PredGuid range:

$$\text{PG_target_range} = \text{THETNM} - \text{ASPF} \quad (\text{B.58})$$

2. Decide whether to run pc_sequencer. Since the range calculation used in the predictor experiences precision errors for small ranges, PredGuid is not run if $\text{PG_target_range} < 100$ nm. However, pc_sequencer must be run in spite of the possible precision errors if PredGuid has not yet been run once. Otherwise, target ranges which use only very short skips will begin to experience large precision landing errors. Thus:

- a. If $\text{PG_target_range} > 100$ nm or $\text{PIND} = 0$, then continue to Step 3.
 - b. Otherwise, end the PredGuid subroutine here.

3. Initialize the inputs to pc_sequencer, as in Table B.8.

- a. If this is the first time through PredGuid ($\text{PIND} = 0$), indicate that PredGuid has been run. Also, override the default values for several inputs as follows:

$$\begin{aligned} \text{PIND} &= 1 \\ \text{PG.CPhi_Desired} &= 0; \\ \text{c.LIFT_INC_CAPTURE} &= -10; \\ \text{c.MAX_NUMBER_RUNS} &= 20; \end{aligned} \quad (\text{B.59})$$

Overriding the default values in this manner allows improved convergence of the predictor-corrector iteration to an acceptable bank angle, since we have no previous guess.

4. Run pc_sequencer.
5. Assign outputs from pc_sequencer to state variables used in the rest of the guidance algorithm, as in Table B.9.

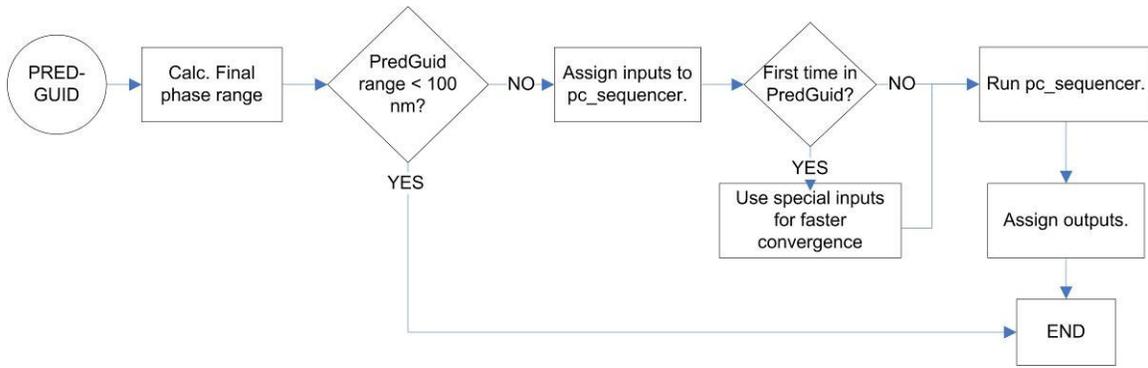


Figure B.8: PredGuid Subroutine Logical Flow

Table B.8: Inputs to pc_sequencer

| <u>Name</u> | <u>Description</u> | <u>Units</u> | <u>Assignment</u> |
|--------------|---|--------------|---|
| CD_est | Estimated vehicle coefficient of drag | nd | c.CD_EST_INITIAL |
| LD_est | Estimated vehicle lift-to-drag ratio | nd | LAD |
| PG_Krho_est | Estimated density bias factor | nd | Krho_est |
| CPhi_Desired | Initial guess for cos(bank angle) | nd | LD/LAD |
| Sign_Of_Bank | Quadrant of bank angle | nd | 1 |
| Position | Initial position vector to propagate | ft [3] | Rbar |
| Velocity | Initial velocity vector | fps [3] | VIbar |
| Acceleration | Initial acceleration vector | fpss [3] | Dbar |
| Altitude | Initial altitude | ft | Alt |
| Velocity_Mag | Initial velocity magnitude | fps | abs(VIbar) |
| PG_Q7 | Drag level at which Ballistic phase ends | fpss | Q7 + KDMIN |
| IND_ini | Indicator of whether current phase is Downcontrol | nd | 0 if (RDOT < 0) && (D > PG_Q7), otherwise 1 |

Table B.9: Outputs from pc_sequencer

| <u>Name</u> | <u>Description</u> | <u>Units</u> | <u>Assignment</u> |
|-------------|---|--------------|--|
| CPHI | cos(bank angle) which should result in reaching target within some tolerance | nd | LD = CPHI*LAD |
| TARGET_MISS | Tolerance within which target is reached | nm | |
| VELMAG | Velocity magnitude at end of propagated trajectory using bank angle specified by CPHI | fps | If abs(TARGET_MISS) < 1000, VL = VELMAG, otherwise no assignment |
| RDOT | Altitude rate at end of propagated trajectory using bank angle specified by CPHI | fps | If abs(TARGET_MISS) < 1000, GAMMAL = -RDOT/VELMAG, otherwise no assignment |

B.7.1 Predictor

See “Aerocapture Guidance Methods for High Energy Trajectories” for a full description of the original predictor’s logical flow and calculations. This section describes how the original ‘aerocapture predictor’ has been modified into a ‘skip predictor’ to be suitable for skip guidance applications.

The aerocapture predictor propagated forward the vehicle’s trajectory by numerically integrating the fully nonlinear equations of motion. Since it was being used for an aerocapture application, the integration would stop either when the vehicle exited the atmosphere, or it became clear that the vehicle would not exit the atmosphere. After integrating the equations of motion to atmospheric exit, the predictor would then calculate the apogee of the new orbit by propagating the exit conditions assuming a Keplerian orbit (i.e. a conic section). If the trajectory resulted in atmospheric capture, the resulting apogee was said to be ‘negative infinity’ (in practice, simply a very large negative number) in order to facilitate the corrector logic.

For the skip predictor, the integration stop conditions have been altered, as summarized below in Table B.10. An additional indicator, `IND_pred`, was added to enable trajectory prediction starting in the Downcontrol phase. This is because the trajectory conditions experienced during the Downcontrol phase would normally trigger the integration stop conditions immediately, resulting in no trajectory propagation. To avoid this, `IND_pred` is initialized to a value of zero during the Downcontrol phase, and is switched to a value of one once the trajectory has finished propagating the Downcontrol portion of the trajectory. This indicates that the normal stop conditions for a captured skip trajectory may now be applied. During the Upcontrol and Ballistic phases, `IND_pred` is initialized to a value of one, indicating that the stop conditions may be applied at any time.

In addition, the predictor has been changed to calculate a downrange distance rather than an apogee altitude. If the trajectory does not capture into the atmosphere, the resulting downrange distance is said to be ‘infinity,’ whereas the downrange for a trajectory which intersects the Earth’s surface is said to be ‘negative’ infinity. This creates a simple situation for the corrector: a too-low downrange indicates that the lift vector should be increased, and a too-high downrange indicates that the lift vector should be decreased.

Table B.10: New Integration Stop Conditions for PredGuid

| <u>Integration stop condition</u> | <u>Range calculation</u> | <u>Type of solution</u> |
|--|--|-------------------------|
| AltPred < 0 | Range = - c.INFINITY (nm) | 'Crash' |
| IND_pred = 1 AND AMag > PG_Q7 AND RdotPred < 0 | Range = $\cos^{-1}(\text{dot}(r_i, r_f)) * 3437.7468$ (nm) | 'Capture' |
| AltPred > c.ALT_MAX | Range = c.INFINITY (nm) | 'Escape' |

B.7.2 Corrector

See “Aerocapture Guidance Methods for High Energy Trajectories” [4] for a full description of the original PredGuid corrector’s logical flow and calculations. This section describes how the original corrector has been updated to work for skip guidance.

The original corrector chose from six different methods to convert a previous bank angle guess to a new bank angle guess, based on the results of the predictor for previous guesses. Table B.11 summarizes the six possible situations and how they were altered for use with skip guidance.

Table B.11: PredGuid Corrector Comparison

| Method # | | Original Method | New Method |
|----------|----------------|--|---|
| 1 | Situation | First iteration during this guidance cycle | Same |
| | Implementation | Use last guess from previous guidance cycle | Same |
| 2 | Situation | Have at least one each of a high solution and a low solution | Same |
| | Implementation | Interpolate between high and low solutions | Same |
| 3 | Situation | Have one good <u>high</u> solution and one <u>capture</u> solution (target is 'bracketed') | Have one good <u>low</u> solution and one <u>escape</u> solution (target is 'bracketed') |
| | Implementation | Interpolate between <u>high</u> and <u>capture</u> solutions | Interpolate between <u>low</u> and <u>escape</u> solutions |
| 4 | Situation | No good solutions (only have <u>capture</u> solutions) | No good solutions |
| | Implementation | March out of capture region by <u>increasing</u> L/D by a specified amount | If there are only <u>escape</u> solutions, march out of capture region by <u>decreasing</u> L/D by a specified amount. If there are only <u>crash</u> solutions, march into capture region by <u>increasing</u> L/D by a specified amount. |
| 5 | Situation | Only one good (<u>escape</u>) solution | Only one good (<u>capture</u>) solution |
| | Implementation | Extrapolate using stored sensitivity of apogee to change in L/D | Use Method 4 implementation; an appropriate stored sensitivity value has not been determined |
| 6 | Situation | At least 2 good (<u>escape</u>) solutions, but target has not been 'bracketed.' | At least 2 good (<u>capture</u>) solutions, but target has not been 'bracketed.' |
| | Implementation | Extrapolate from 2 good high guesses or 2 good low guesses | Same |

For the lowest L/D values, the aerocapture predictor will start with ‘bad’ (capture) solutions, and as L/D increases, the predictor will move into a ‘good’ (escape) region. Because of this, the aerocapture corrector can use fairly simple heuristics. The skip predictor, on the other hand, goes from ‘bad’ (‘crash’) solutions at the lowest L/D values, moves through a region of ‘good’ (capture) solutions, and then back into a region of ‘bad’ (escape) solutions as L/D increases. This movement from bad to good and back to bad again increases the complexity of the required heuristics, a complexity which is not captured well within this simple adaptation of the corrector. It would be desirable to design a new corrector with more specialized heuristics in order to better describe the skip guidance situation.

B.8 Ballistic Phase

This corresponds to the KEP3 subroutine described in “Reentry Guidance for Apollo”. The logical flow of this phase is illustrated in Figure B.9.

1. If the drag has risen above 0.2 g’s ($D < Q7 + KDMIN$), then it is time to move to the Final phase. Indicate a switch to the Final phase:

$$\begin{aligned} \text{SELECTOR} &= 5 \text{ (Final Phase)} \\ \text{EGSW} &= 1 \end{aligned} \tag{B.53}$$

Skip the rest of this subroutine and go directly to the Final subroutine.

2. Otherwise, continue steering by running the PredGuid subroutine, then proceed to the Lateral Logic subroutine.

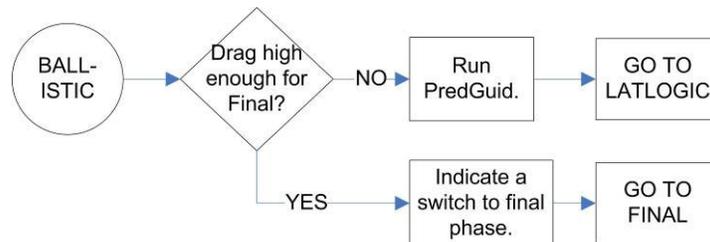


Figure B.9: Ballistic Logical Flow

B.9 Final Phase

This corresponds to the PREDICT3 subroutine described in “Reentry Guidance for Apollo”. The logical flow of this subroutine, along with the 10-g limiter, is illustrated in Figure B.10.

1. If the velocity has dropped low enough ($V < V_{QUIT}$), it is time to stop steering. Set the L/D to a neutral command of zero, and flip the switch indicating that there will be no more bank angle updates:

$$\begin{aligned} LD &= 0 \\ ROLLC &= K2ROLL \cdot \cos^{-1}(LD/LAD) + 2\pi \cdot K1ROLL \\ QUITIND &= 1 \end{aligned} \quad (B.54)$$

Skip the rest of this subroutine and send the bank angle command (ROLLC) and quit indicator (QUITIND) directly to output.

2. If the target has been overshoot previously ($GONEPAST = 1$), then command full lift down in order to minimize the overshoot:

$$LD = -LAD \quad (B.55)$$

Skip to Step 5.

3. If the target has just been overshoot, indicate so and command full lift down:

$$\begin{aligned} GONEPAST &= 1 \\ LD &= -LAD \end{aligned} \quad (B.56)$$

Skip to Step 5.

4. Calculate the L/D command based on trajectory errors, interpolating values of RTOGO, F2, RDOTREF, F1, DREFR, and F3 from the reference trajectory table, FTABLE:

$$\begin{aligned} PREDANGL &= RTOGO(V) + F2(V) \cdot [RDOT - RDOTREF(V)] \\ &\quad + F1(V) \cdot [D - DREFR(V)] \\ L/D &= LOD + 4 \cdot (THETNM - PREDANGL)/F3(V) \end{aligned} \quad (B.57)$$

5. If the current drag is lower than half the 10-g limit ($D < GMAX/2$), then there is no need to limit the aerodynamic acceleration at this time. Skip the rest of this subroutine and go directly to the Lateral Logic subroutine.
6. If the current drag is greater than the 10-g limit ($D > GMAX$), then command full lift up in order to reduce the g-loads:

$$LD = LAD \quad (B.58)$$

Skip the rest of this subroutine and go directly to the Lateral Logic subroutine.

7. Calculate the limiting altitude rate:

$$X = \sqrt{2 \cdot HS \cdot (GMAX - D) \cdot (LEQ/GMAX + LAD) + (2 \cdot HS \cdot GMAX/V)^2} \quad (B.59)$$

8. Check whether the current altitude rate exceeds the limit ($RDOT < -X$). If so, command full lift up to reduce the g-loads:

$$LD = LAD \quad (B.60)$$

9. Continue on to the Lateral Logic subroutine.

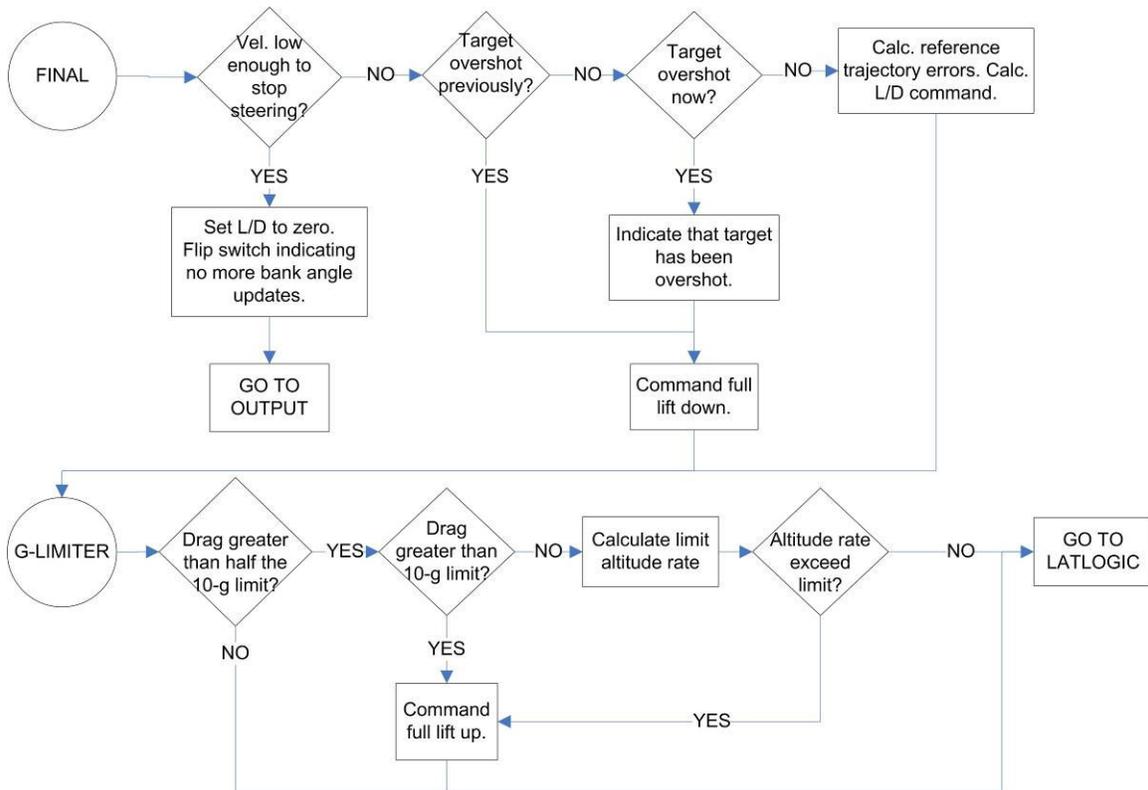


Figure B.10: Final Logical Flow

B.10 Lateral Logic Subroutine

This corresponds to the 310 subroutine described in “Reentry Guidance for Apollo”.

This subroutine has not been altered from the original Apollo algorithm. The logical flow is illustrated in Figure B.11.

1. Check whether the target has been overshoot ($GONEPAST = 1$).

- a. If so, no bank reversal logic needs to be performed. This is because the vehicle will be at full lift down and there will be no lateral control available. Skip to Step 5.
2. Calculate the lateral switch limit:

$$Y = K_{LAT} \cdot VSQ + LATBIAS/ATK \quad (B.61)$$

3. Check whether the L/D command is within 15 deg. of full lift up or full lift down ($abs(LD) > LDCMINR$). If so:
 - a. The vehicle has reduced control authority in the lateral channel. Reduce the lateral switch limit by half:

$$Y = Y/2 \quad (B.62)$$

- b. Check whether the lift vector being in this quadrant is causing the lateral angle to decrease ($K2ROLL \cdot LATANG > 0$). If so:
 1. Increase the amount of control in the lateral channel by changing the L/D command to 15 deg. from full lift up or full lift down, as appropriate:

$$LD = LDCMINR \cdot sign(LD) \quad (B.63)$$

2. Skip to Step 5.

4. Check whether the lateral angle limit is exceeded ($K2ROLL \cdot LATANG > Y$). If so:
 - a. Command a bank angle reversal by switching the quadrant of the bank angle:

$$K2ROLL = -K2ROLL \quad (B.64)$$

- b. Check whether the vehicle is in a lift down orientation ($LD < 0$). If so, Command a reversal through lift down since this is shorter than going through lift up. This can be done by incrementing the revolution counter:

$$K1ROLL = K1ROLL - K2ROLL \quad (B.65)$$

5. Check whether the commanded L/D is larger than the maximum vehicle L/D ($abs(LD) > LAD$). If so, reduce the L/D command to the maximum vehicle L/D:

$$LD = sign(LD) \cdot LAD \quad (B.66)$$

6. Calculate the bank command from the desired L/D:

$$ROLLC = K2ROLL \cdot \cos^{-1}(LD/LAD) + 2\pi \cdot K1ROLL \quad (B.67)$$

7. Output the bank command (ROLLC) and the steering quit indicator (QUITIND).

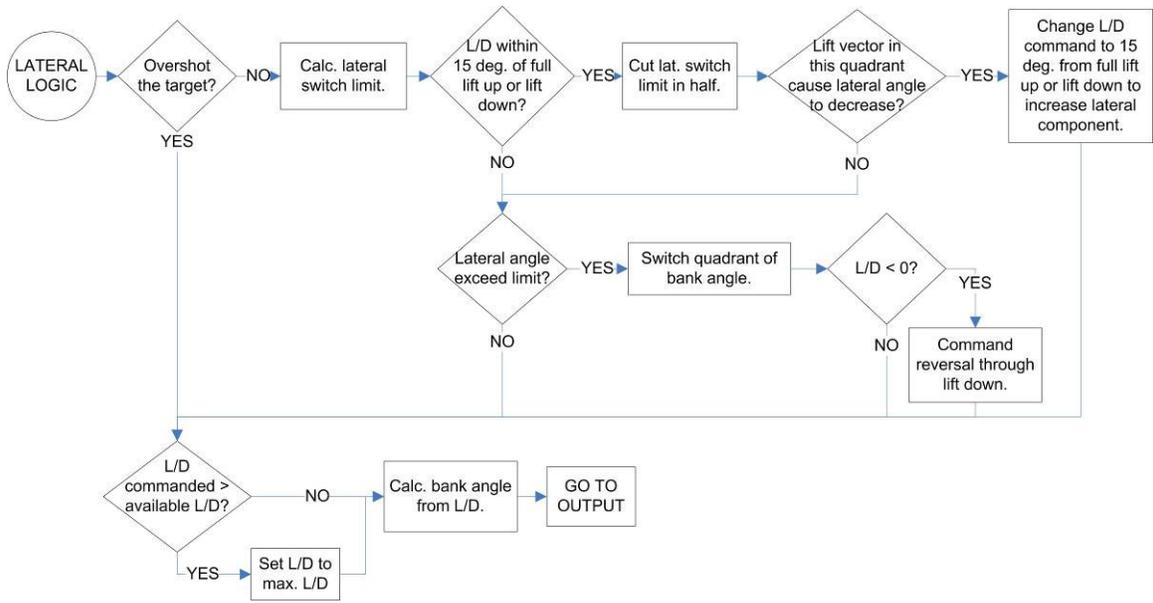


Figure B.11: Lateral Logic - Logical Flow

[This Page Intentionally Left Blank]

Appendix C

Monte Carlo Method and Parameters

C.1 Monte Carlo Method

As described in Chapter 3, Monte Carlo analyses were performed on the various versions of the guidance algorithm. Each Monte Carlo consists of 250 guided reentry trials with various perturbed simulation parameters. Each of the perturbations is described by a scalar value assumed to conform to a Gaussian distribution and specified by a mean value (μ) and a maximum expected perturbation ($3\text{-}\sigma$). The values used are described in the next section.

The following parameters are perturbed for each trial:

Initial velocity, azimuth, and flight path angle. These perturbations characterize the uncertainties in the vehicle's capability to precisely target Entry Interface.

Vehicle mass. This uncertainty stems from the inability to know exactly how much of the consumables will have been expended by the time of reentry.

CG location. Although CG location uncertainty primarily affects the aerodynamic characteristics of the vehicle, it is due to the uncertainty in distribution of the mass within the vehicle. The CG location along the x-axis of the vehicle primarily affects vehicle stability, which is not relevant in this 4-DOF simulation. In addition, the vehicle axes are defined such that the y-axis will always have zero CG offset; any y-direction offset will just change the definition of the z-axis. Thus, it is only necessary to perturb the z-component of the CG location.

Multiplier for atmospheric density. Atmospheric density variation is an environmental effect. Atmospheric density is a function of altitude, it is reasonable to implement uncertainties by generating a perturbation function (also a function of altitude) against

which the density function is multiplied. For simplicity, it was decided to define the perturbation function as a single scalar multiplier, which can then be randomized according to a Gaussian distribution. By definition, the mean value of such a multiplier must be 1, so that applying the mean value would return the nominal atmosphere. A value of greater than 1 would represent a thicker atmosphere, and a value of less than 1 would represent a thinner atmosphere.

Multipliers for coefficients of drag and lift. In contrast to CG location, uncertainties in coefficients of drag and lift are intended to model uncertainties about the precise shape of the outer shell of the vehicle. Since C_D and C_L are functions of mach number, a perturbation function is also defined in terms of mach number. Since hypersonic vehicle aerodynamic characteristics are more predictable than supersonic and subsonic vehicle characteristics, the multiplier should have a smaller deviation for high mach numbers than for low mach numbers. For simplicity, the perturbation function used here is defined in terms of two numbers: a hypersonic value which is constant above Mach 6, and a super/sub-sonic value which is constant below Mach 4. Between Mach 4 and 6, there is a linear interpolation by mach number so that the aerodynamic characteristics transition smoothly between the high-mach and low-mach regimes, as shown in Figure C.1.

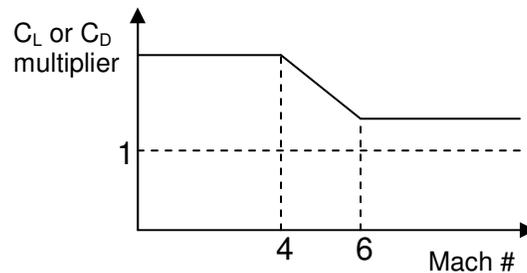


Figure C.1: Multiplier for Coefficients of Drag and Lift

It makes sense that the hypersonic and subsonic values should remain correlated, so the two values are calculated from the same random number to have the same amount of relative perturbation. For example, a multiplier which has a $2\text{-}\sigma$ deviation on the high side (above 1) for its hypersonic value will also have a $2\text{-}\sigma$ super/sub-sonic perturbation on the high side. Note that these multipliers are applied after the trim values for angle of attack, C_D , and C_L are calculated based on a perturbed CG location.

C.2 Parameter Values and Uncertainties

Table C.1 describes the mean and 3- σ values used to represent the Gaussian distribution for each parameter's perturbation.

Table C.1: Monte Carlo Parameter Values

| Parameter | | Mean value (μ) | Reason for mean | 3- σ deviation | Reason for 3- σ |
|----------------------------------|-----------------|----------------------|---------------------------------|-----------------------|---|
| EI Velocity | | 11,032 m/s | Lunar return velocity | 300 m/s | Based on various lunar departure declinations and Earth approach inclinations |
| EI Azimuth | | 90 deg. (East) | Equatorial orbit for simplicity | 0.1 deg. | Equal to orbital inclination error [23] |
| EI Flight Path Angle | | -5.9 deg. | Analysis by Draper Laboratory | 0.1 deg. | Based on 0.5 nm vacuum periapsis altitude error [23] |
| Vehicle Mass | | 9,500 kg | Approximately 21,000 lbs [18] | 5% (475 kg) | Design uncertainty |
| Vehicle CG Offset (along z-axis) | | 0.255 m | Yields L/D of approx. 0.35 | 0.05 m (~2 in.) | Analysis by Draper Laboratory |
| Atmospheric Density Multiplier | | 1 | Definition of multiplier | 30% (0.30) | Day of flight uncertainty |
| C _D multiplier | Super/sub-sonic | 1 | Definition of multiplier | 15% (0.15) | Based on CFD analysis and test flight calibration [24] |
| | Hypersonic | 1 | Definition of multiplier | 5% (0.05) | |
| C _L multiplier | Super/sub-sonic | 1 | Definition of multiplier | 15% (0.15) | |
| | Hypersonic | 1 | Definition of multiplier | 5% (0.05) | |

[This Page Intentionally Left Blank]

Appendix D

Metric Calculations

D.1 Landing Error

Each Monte Carlo trial is initiated at time $t = 0$ and position vector $\bar{r}_i(0)$. The simulation is terminated when the trajectory reaches an altitude of less than zero. This final integration step occurs at time t_f and corresponds to an inertial position vector of $\bar{r}_i(t_f)$.

Landing error is a comparison of the actual landing site to the target landing site. Since the Earth rotates, it is necessary to determine where the landing target is at the time the vehicle touches down. This is accomplished by rotating the initial target position vector by the angle the Earth has rotated between the initial time and the final time, as illustrated in Figure D.1 and calculated in Equation D.1. Equation D.1 shows the general equation for rotating any target vector; the calculation could be simplified for an equatorial orbit, but that simplified equation is not shown here.

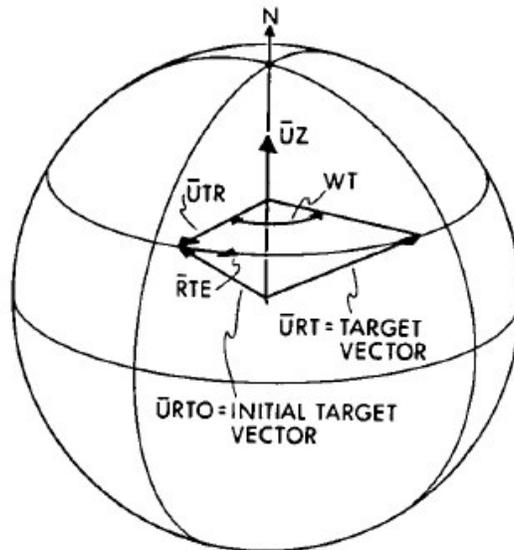


Figure D.1: Projection of Target Vector through Time of Flight [9]

$$\text{URT} = \text{URT0} + \text{UTR} \cdot (\cos(\omega \cdot t_f) - 1) + \text{RTE} \cdot \sin(\omega \cdot t_f); \quad (\text{D.1})$$

where:

$$\begin{aligned} \text{RTE} &= \text{UZ} \times \text{URT0} \\ \text{UTR} &= \text{RTE} \times \text{UZ} \end{aligned} \quad (\text{D.2})$$

and:

URT: unit rotated target vector in inertial coordinates
 URT0: initial target vector (input to guidance) in inertial coordinates
 RTE: unit local east vector at URT0
 UTR: unit vector perpendicular to both UZ and RTE
 UZ: unit vector north from center of Earth
 ω : Earth rotation rate: 7.29211505 e-5 rad/sec

Assuming the miss distance is on the order of a few kilometers, which is small compared to the radius of the Earth, the miss distance is simply the difference between the actual final position vector and the target position vector, as in Equation D.3:

$$\vec{r}_{miss} = r_e \cdot (\hat{r}_i(t_f) - \text{URT}) \quad (\text{D.3})$$

where:

\vec{r}_{miss} : miss vector (m)
 r_e : Earth radius corresponding to URT (m) (see Equation 2.27)
 $\hat{r}_i(t_f)$: unit final position vector

This vector is then decomposed into the eastward and northward components. For an equatorial orbit such as is assumed for this thesis, this corresponds to the downrange and crossrange miss components, respectively.

$$\begin{aligned} r_{miss,dr} &= \vec{r}_{miss} \cdot \hat{E} \\ r_{miss,cr} &= \vec{r}_{miss} \cdot \hat{N} \end{aligned} \quad (\text{D.4})$$

where:

$$\begin{aligned} \hat{E} &= \text{UZ} \times \text{URT}: \text{local unit vector east at URT} \\ \hat{N} &= \text{URT} \times \hat{E}: \text{local unit vector north at URT} \end{aligned} \quad (\text{D.5})$$

Using these values, it is possible to plot each trial’s crossrange vs. downrange miss distance to create the landing error scatter plot. After the downrange and crossrange miss distance has been calculated for each of the trials, it is possible to calculate the Monte Carlo landing error statistics: namely, the landing ellipse errors and the 95% circular error probability (CEP).

The landing ellipse errors are calculated by considering downrange miss ($r_{miss,dr}$) and crossrange miss ($r_{miss,cr}$) independently when calculating the statistics (mean and standard deviation). The resulting mean downrange error ($r_{miss,dr,mean}$) and mean crossrange error ($r_{miss,cr,mean}$) together specify the ‘mean landing location’ for the Monte Carlo, with respect to the target at location (0, 0). The distance from the mean landing location for any given Monte Carlo trial is calculated as follows:

$$d_{mean} = \sqrt{(r_{miss,dr} - r_{miss,dr,mean})^2 + (r_{miss,cr} - r_{miss,cr,mean})^2} \quad (D.6)$$

The CEP is calculated by taking the 99.7th percentile value for distance from mean landing location (d_{mean}). For a 250-trial population, such as used in this thesis, this corresponds to the trial with the largest distance from the mean landing location.

D.2 Duration-based G-loads

The constraints for the duration-based g-load plots are based on the NASA-STD-3000 guidelines [19]. By visual inspection, several data points were taken for each constraint so that they could be plotted in MATLAB, as in Table D.1 through Table D.3. These constraints were plotted on a log-log plot of g-load vs. duration.

Table D.1: Maximum G-load Limits for Automated Crew Abort/Escape

| | | | | |
|----------------|-----|-----|-----|------|
| Duration (sec) | 0.5 | 100 | 300 | 1000 |
| G-load (g’s) | 38 | 9.5 | 7.5 | 5.1 |

Table D.2: Nominal G-load Limits for Conditioned Crew

| | | | | |
|----------------|-----|----|----|-----|
| Duration (sec) | 0.5 | 5 | 60 | 300 |
| G-load (g’s) | 19 | 16 | 10 | 7.3 |

Table D.3: G-load Limits for Deconditioned, Ill, or Injured Crew

| | | | | | |
|----------------|-----|----|-----|-----|------|
| Duration (sec) | 0.5 | 8 | 30 | 100 | 1000 |
| G-load (g's) | 14 | 10 | 7.8 | 4 | 4 |

In order to determine the duration-based g-loads for each trial, it was necessary to keep a record of the g-load level at each integration step of the trajectory. First, the maximum g-load for that trial was determined. This g-load was rounded down to the nearest 0.2 g increment, and the number of timesteps for which the g-load level was above this 0.2 g increment were counted. Then, since the integration timesteps for the trajectory were spaced at 0.25 seconds, the number of timesteps was multiplied by 0.25 seconds to yield the total amount of time for which the trajectory was above this g-level. This point was plotted on the log-log plot mentioned above. This procedure was repeated for every 0.2 g increment from the maximum down to 1 g, and the dots were connected together with a line. This represents the duration-based g-load for a single trial. Then, this whole procedure was repeated for each trial, until all of the trials were plotted on a single graph with the constraints.

D.3 Final Phase Energy Bucket

The Final phase energy bucket limits depend on the initial conditions for the Final phase. The conditions vary depending on both target range and the algorithm used, and thus the limits of the bucket will vary with both of these parameters. For this reason, the lift-up and lift-down limits for the Final phase energy bucket must be calculated separately for each Monte Carlo set.

For a given Monte Carlo set, the initial conditions for the Final phase must be determined. These can be specified in terms of altitude, flight path angle, and velocity. These values vary somewhat from trial to trial, but a statistical analysis of the variations shows that the variation is typically small with respect to the mean values. Thus, a representative energy bucket can be defined from the mean values of altitude, flight path angle, and velocity at the beginning of the Final phase as averaged over all the trials in the Monte Carlo.

Given these representative initial conditions, two vehicle reentry simulations are run using these as the initial conditions. One of the simulations is run with the vehicle bank command held constant at full lift up, without any bank modulation or guidance logic. The other is run at full lift down without any guidance logic except for the 10-g limiter, which is described above in Appendix B.

This simulation yields two sets of bucket limits which are specified as energy/weight and range to go as functions of time and are calculated from other state variables. For the bucket limits, the calculation of range to go can be described as range-to-touchdown. The calculation of energy/weight is shown in Equation D.7 and the range to go calculation is shown in Equation D.8.

$$E/W = h + \frac{1}{2}v^2 / g \quad (D.7)$$

where:

E/W : energy/weight (m)
 h : altitude (m)
 v : inertial velocity (m/s)
 g : acceleration of gravity near Earth's surface: 9.81 m/s

$$RTG = DR(r_f) - DR(r) \quad (D.8)$$

where:

$$DR(r) = r_e \cdot \cos^{-1}(R0 \bullet (U0 \times (r \times U0))) \quad (D.9)$$

$U0 = \text{unit}(R0 \times V0)$: unit normal to plane of initial orbit

and:

r_f : vehicle position at touchdown
 r_e : Earth radius (m)
 $R0$: initial, inertial position of orbit
 $V0$: initial, inertial velocity of orbit

These limits are then plotted in red as energy/weight vs. range to go in both types of energy bucket plots used in this thesis.

The first type of energy bucket plot includes traces of energy/weight vs. range to go for each of the trials in the Monte Carlo. For this, the energy/weight calculation is the same

as in equation D.7. The range to go calculation, shown in Equation D.10, reflects a range-to-target rather than a range-to-touchdown as calculated in Equation D.8.

$$RTG = DR(URT) - DR(r) \quad (D.10)$$

The second type of energy bucket plot includes actual transition locations and targeted transition locations for each of the trials in the Monte Carlo. The actual transition locations are described in terms of energy/weight and range to go at the beginning of the Final phase. These points are calculated as in Equation D.7 and Equation D.10 above.

Since the guidance algorithm targets only a range to go, and not energy/weight, for the start of the Final phase, the energy/weight value for the target can be taken as the actual energy/weight at the time of the Final phase transition. The targeted range to go value is recorded in the guidance state variable ‘ASPF,’ and can be based on the value of that variable at the time of the Final phase transition. However, this range to go value is based on an estimate of the time to touchdown, not the actual time, and its adjustment for the rotation of the Earth is not exact in the inertial frame. Since the energy bucket is plotted in terms of ‘inertial downrange,’ the targeted point must be correct for this epoch error, so that an actual transition overlaying the target point on the plot will mean that the vehicle has hit the target. The epoch correction is shown in Equation D.11.

$$RTG_{\text{target}} = \text{ASPF}(t_t) + (\omega \cdot t_f - \text{WT}(t_t)) \cdot r_e \quad (D.11)$$

where:

ASPF, WT: state variables
 t_t : time of transition to final phase

Appendix E

Results – Metrics

E.1 Original Apollo Algorithm

Figure E.1 through Figure E.8 show the metrics for the Monte Carlo series investigating the performance of the original, unaltered Apollo algorithm as applied to the 2,400 km, 3,500 km, 4,600 km, 7,300 km, and 10,000 km target ranges.

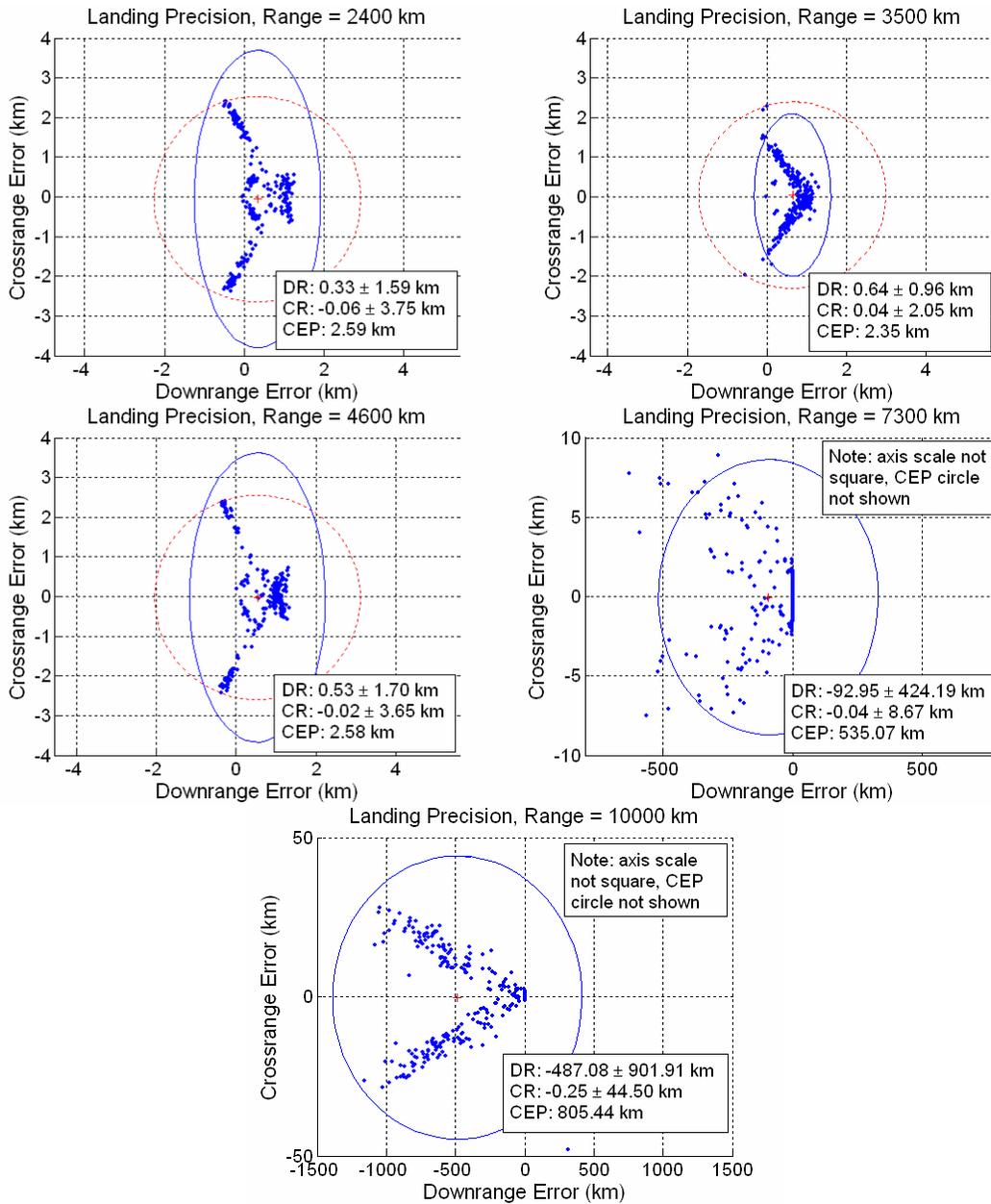


Figure E.1: Landing Precision Scatter Plots, Original Apollo Algorithm, All Ranges

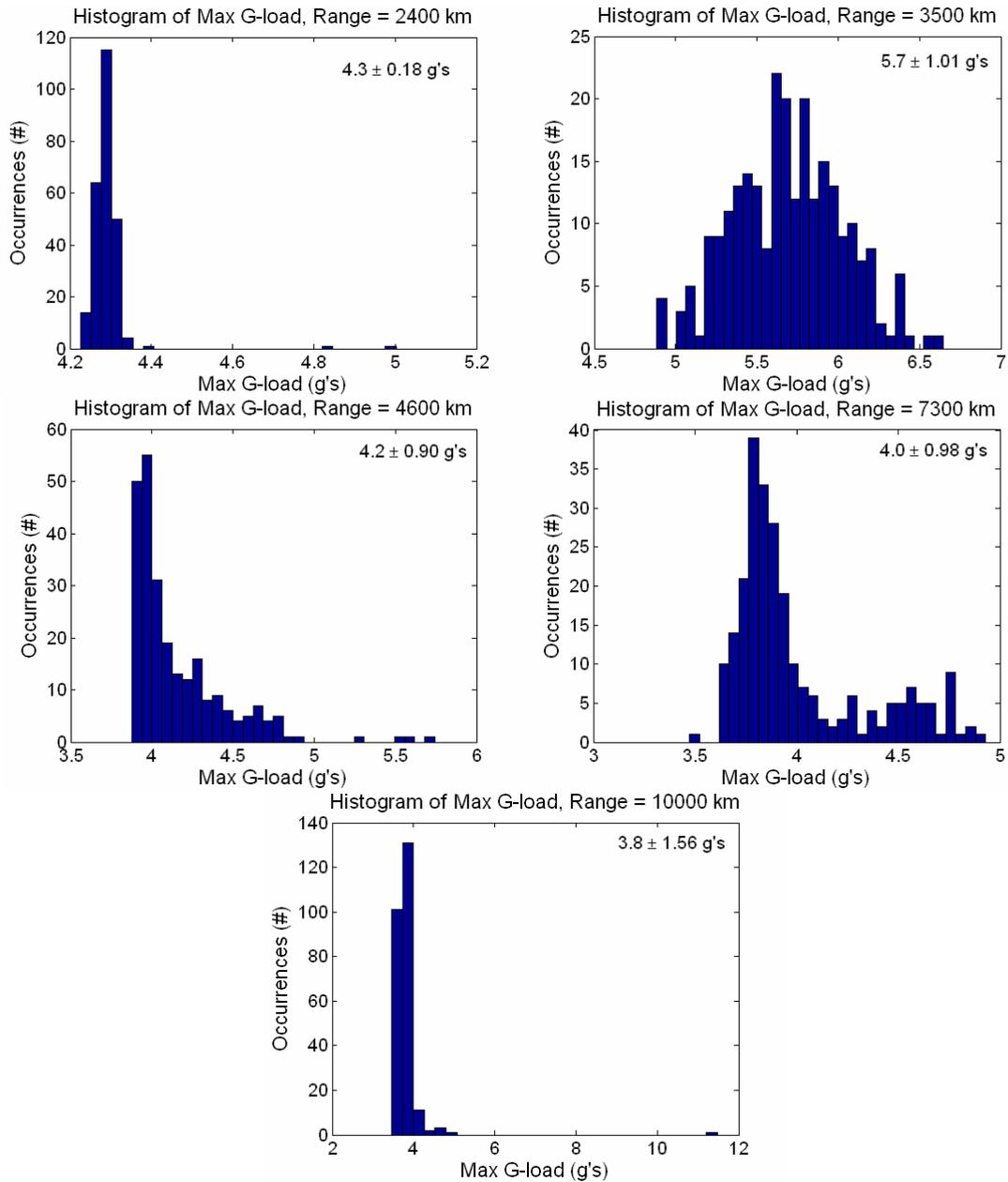


Figure E.2: Max G-load Histograms, Original Apollo Algorithm, All Ranges

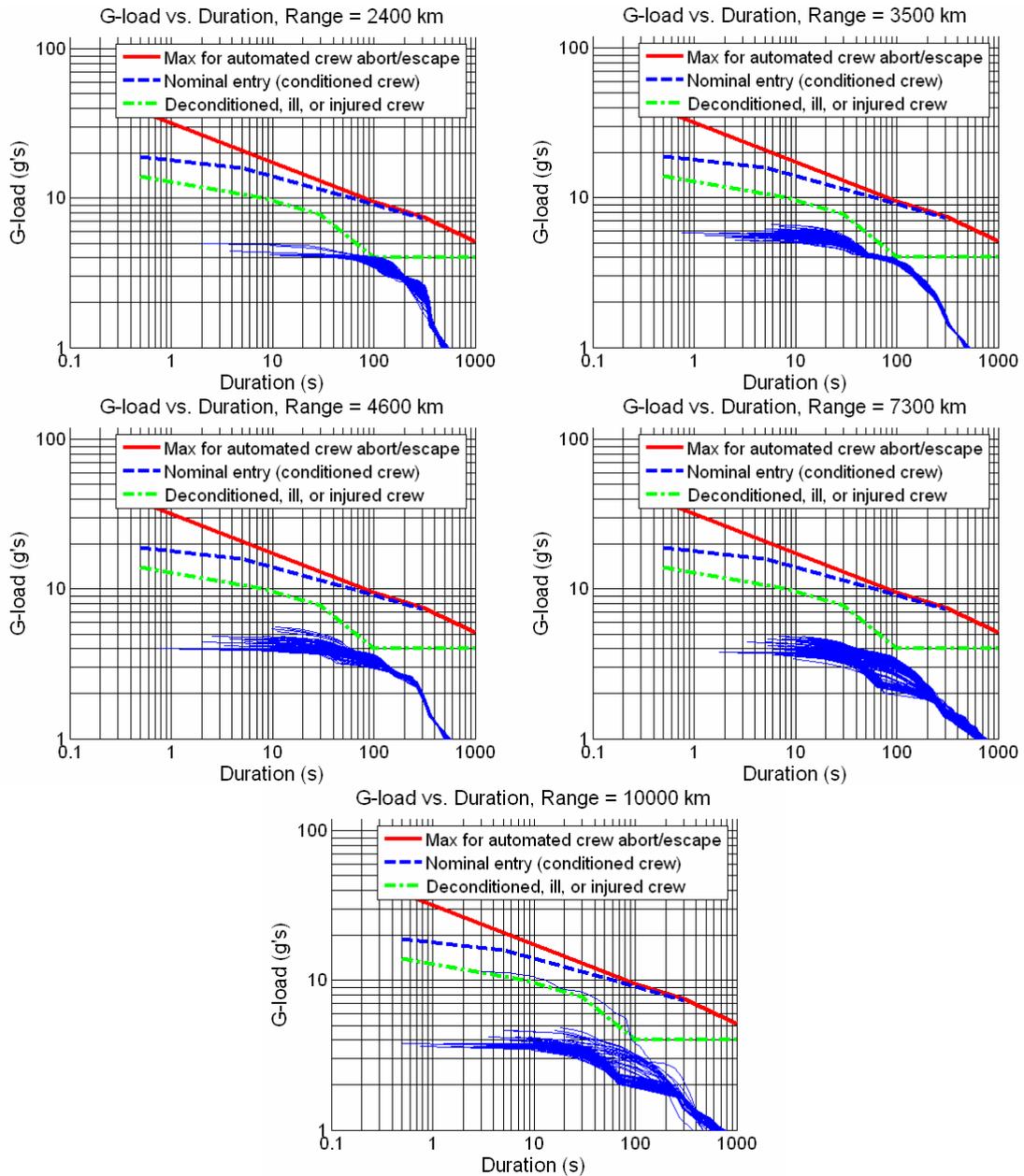


Figure E.3: Duration-based G-loads, Original Apollo Algorithm, All Ranges

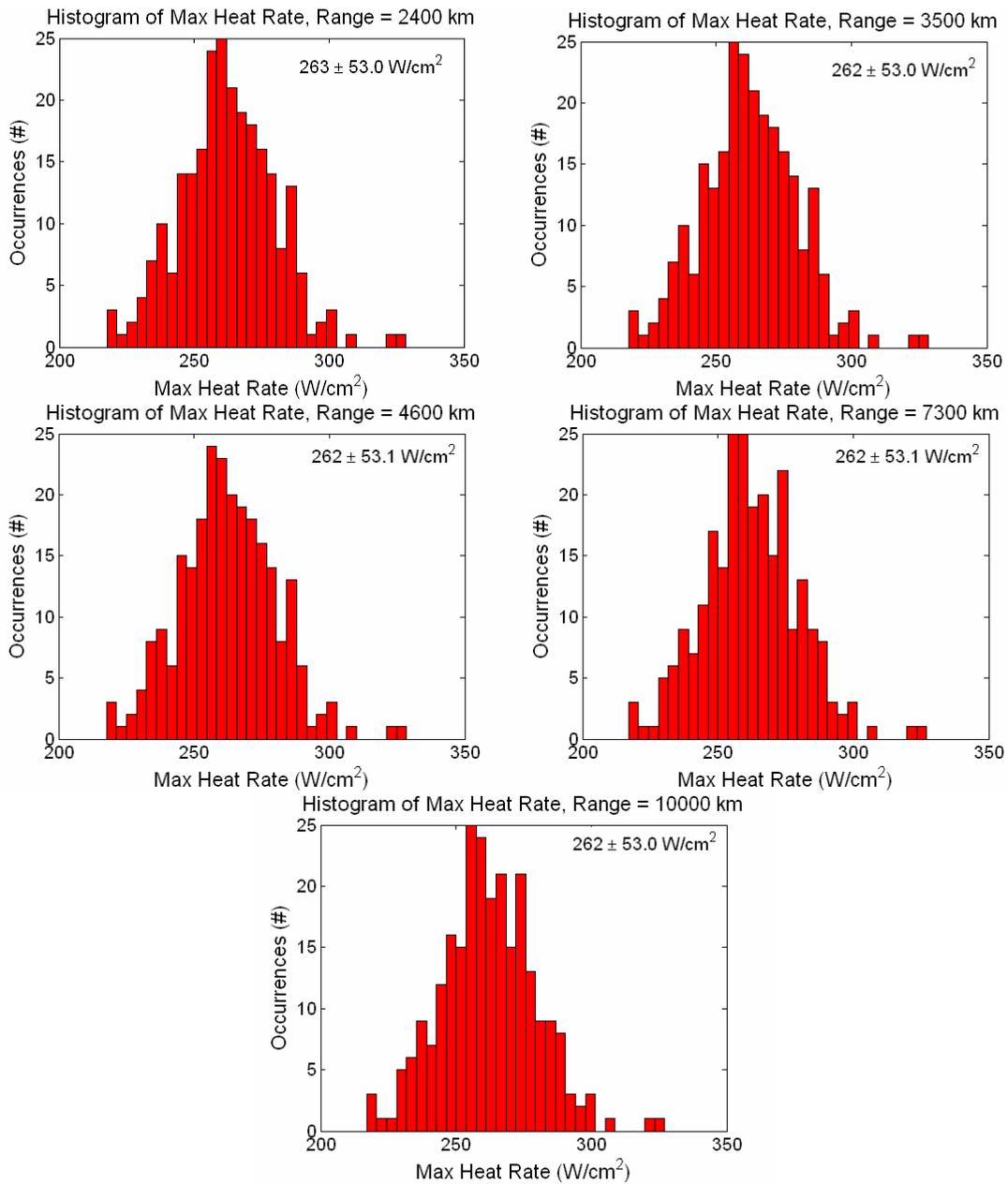


Figure E.4: Max Heat Rate Histograms, Original Apollo Algorithm, All Ranges

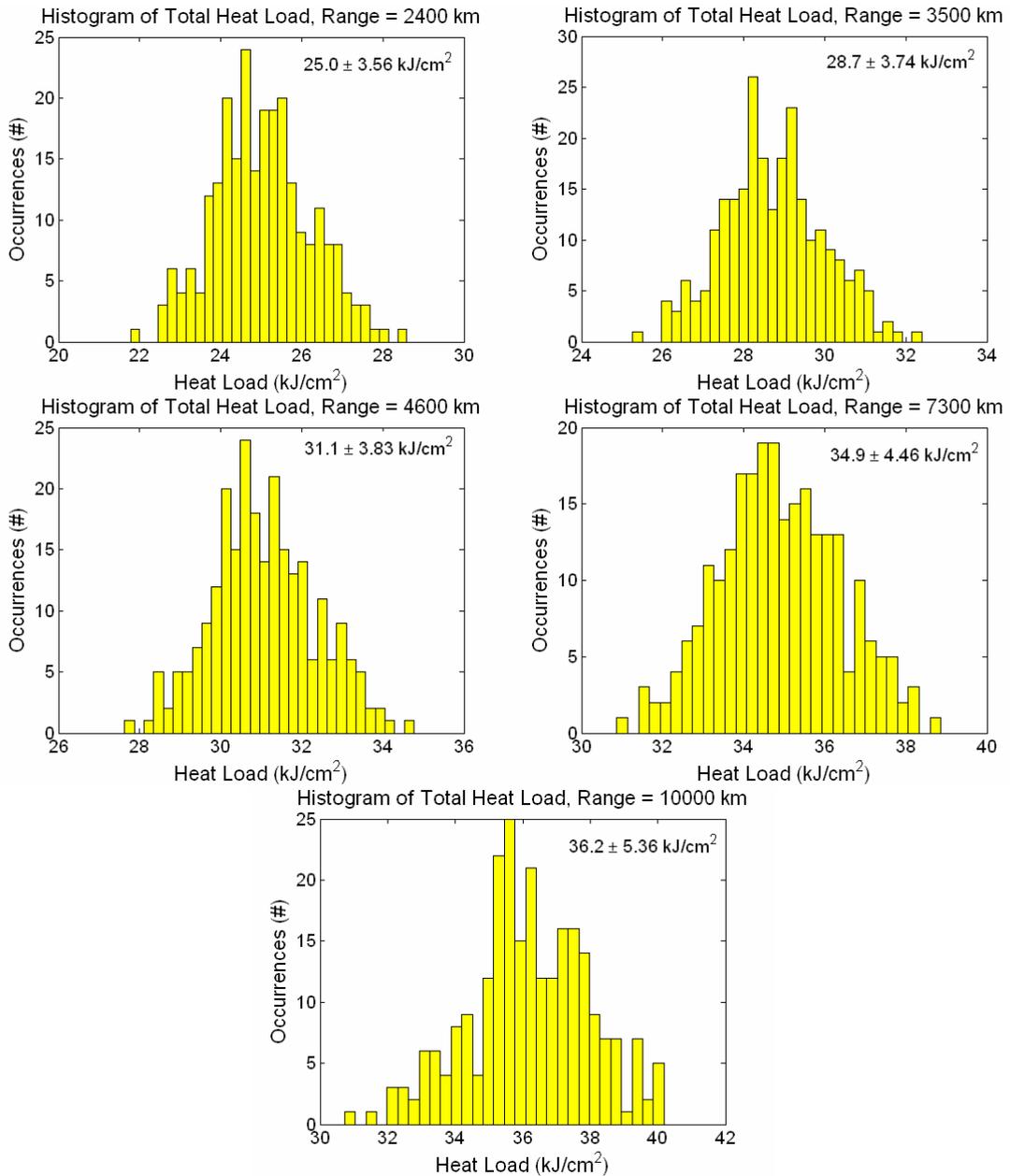


Figure E.5: Total Heat Load Histograms, Original Apollo Algorithm, All Ranges

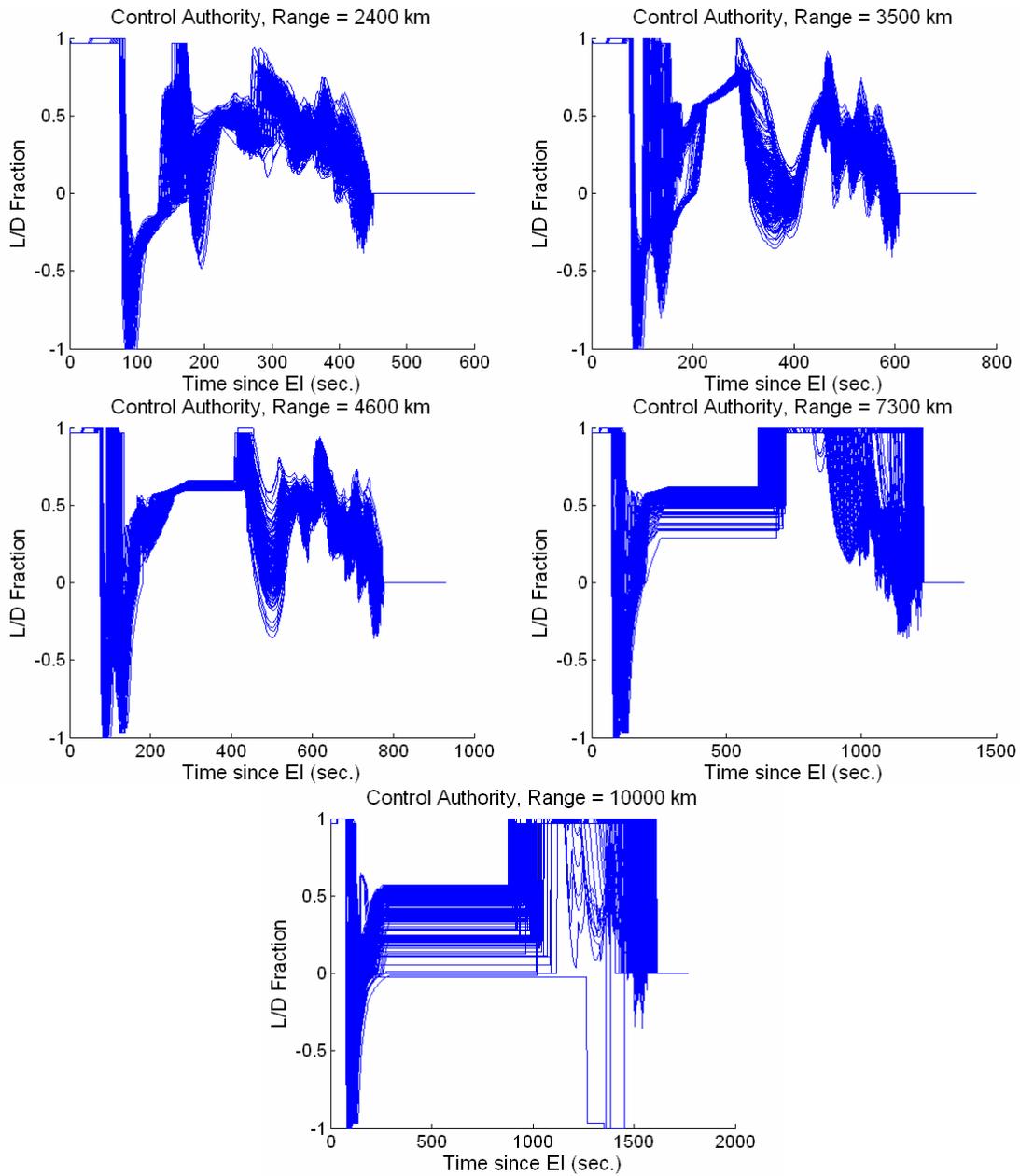


Figure E.6: Commanded L/D Fraction, Original Apollo Algorithm, All Ranges

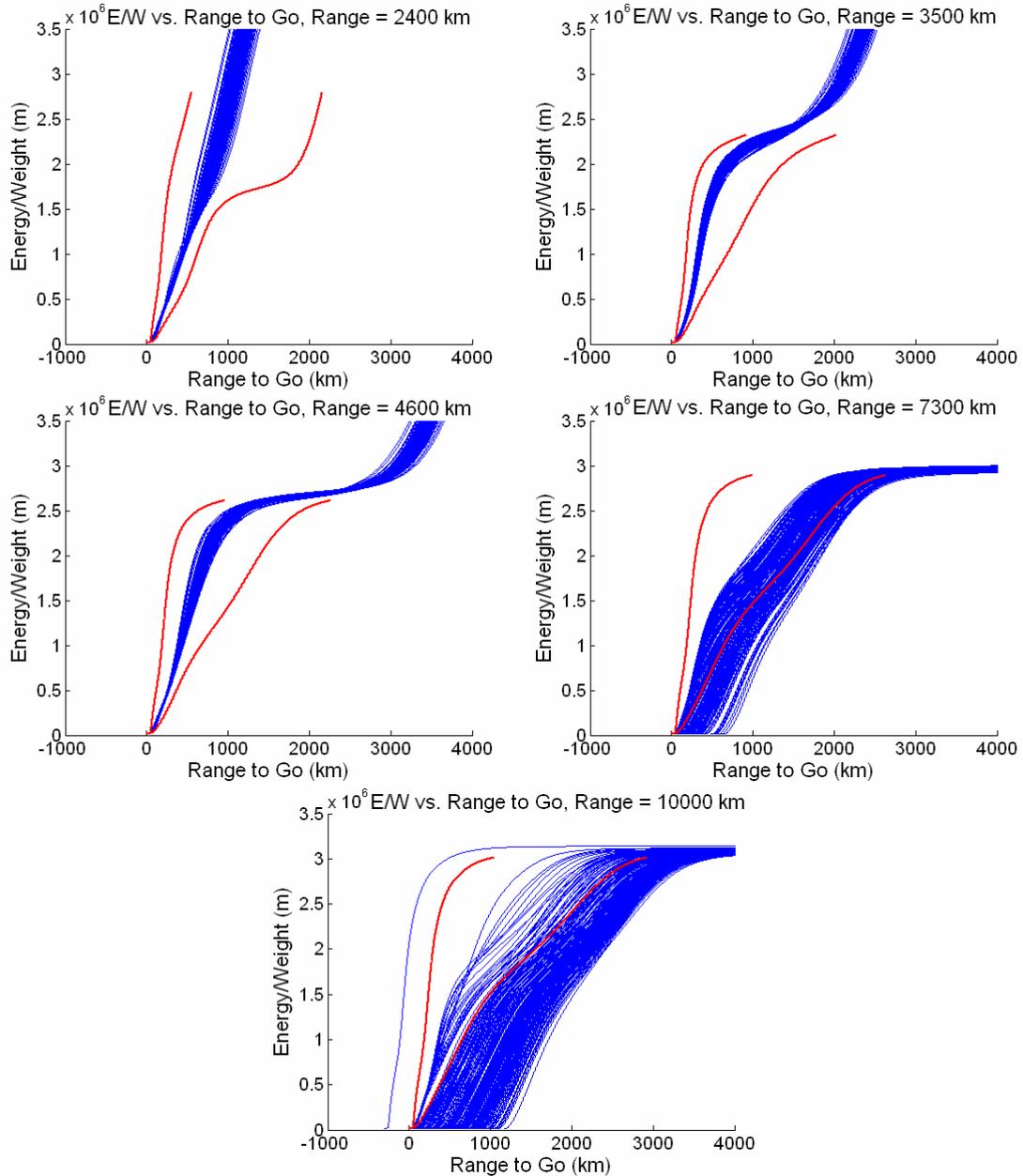


Figure E.7: Energy Bucket Traces, Original Apollo Algorithm, All Ranges

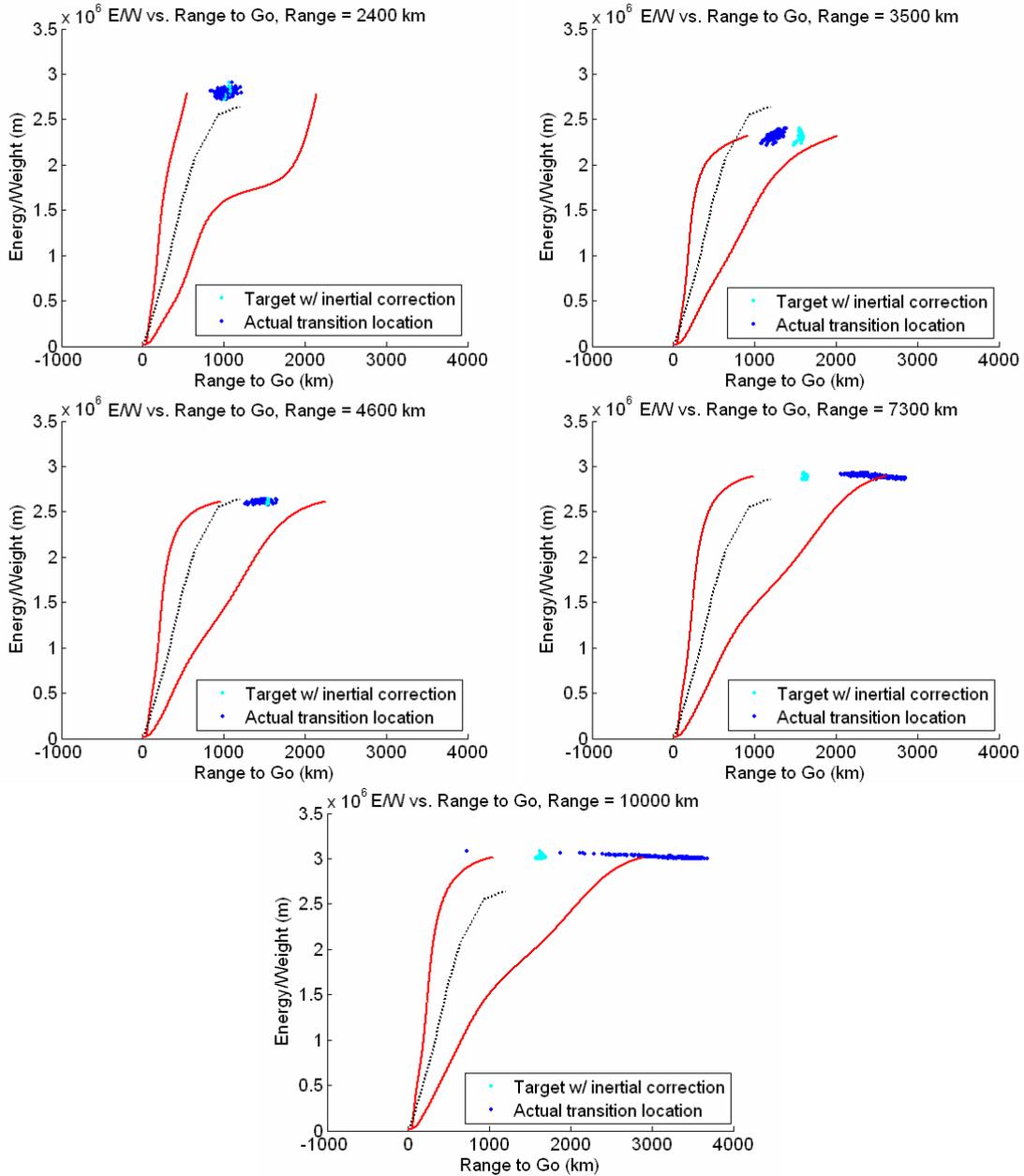


Figure E.8: Energy Bucket Transitions, Original Apollo Algorithm, All Ranges

E.2 Low Loft Enhanced Algorithm

Figure E.9 through Figure E.16 show the metrics for the Monte Carlo series investigating the performance of the Low Loft Enhanced version of the algorithm. In this version, the Upcontrol and Ballistic phases of the original Apollo algorithm are replaced by PredGuid, resulting in a lower-altitude loft. In addition, this version includes the new Final phase reference trajectory and range estimation method.

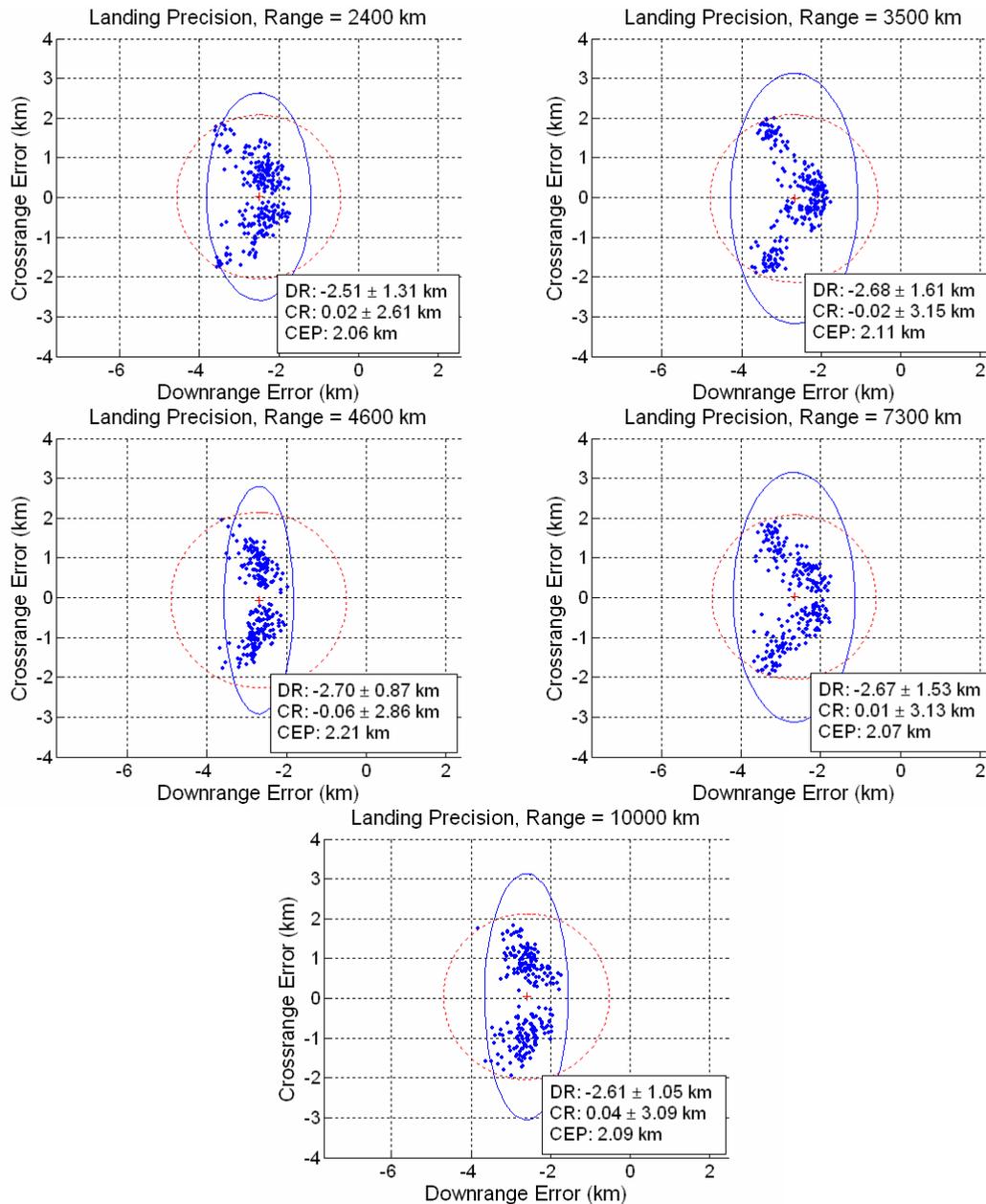


Figure E.9: Landing Precision Scatter Plots, Low Loft Enhanced Algorithm, All Ranges

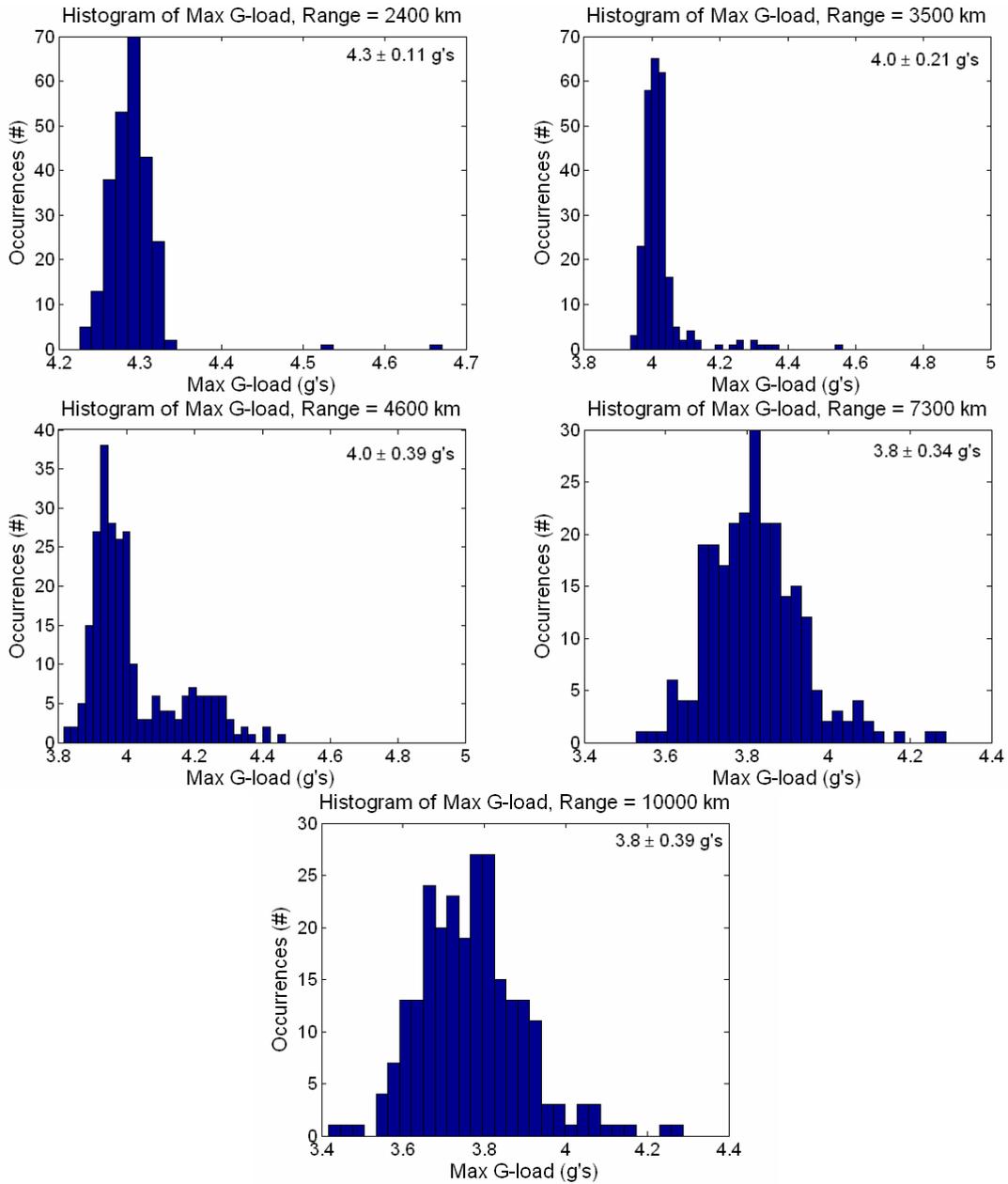


Figure E.10: Max G-load Histograms, Low Loft Enhanced Algorithm, All Ranges

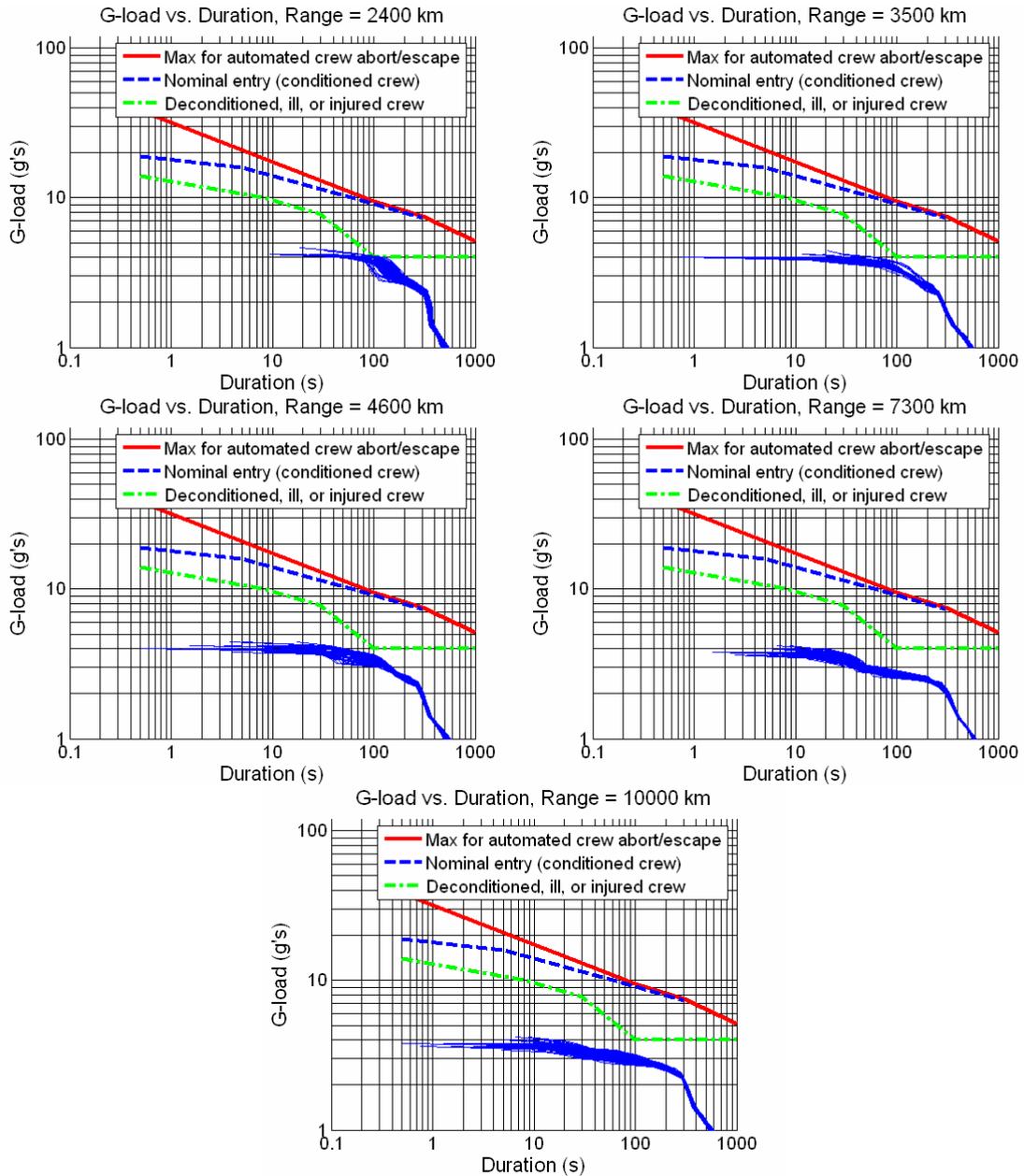


Figure E.11: Duration-based G-loads, Low Loft Enhanced Algorithm, All Ranges

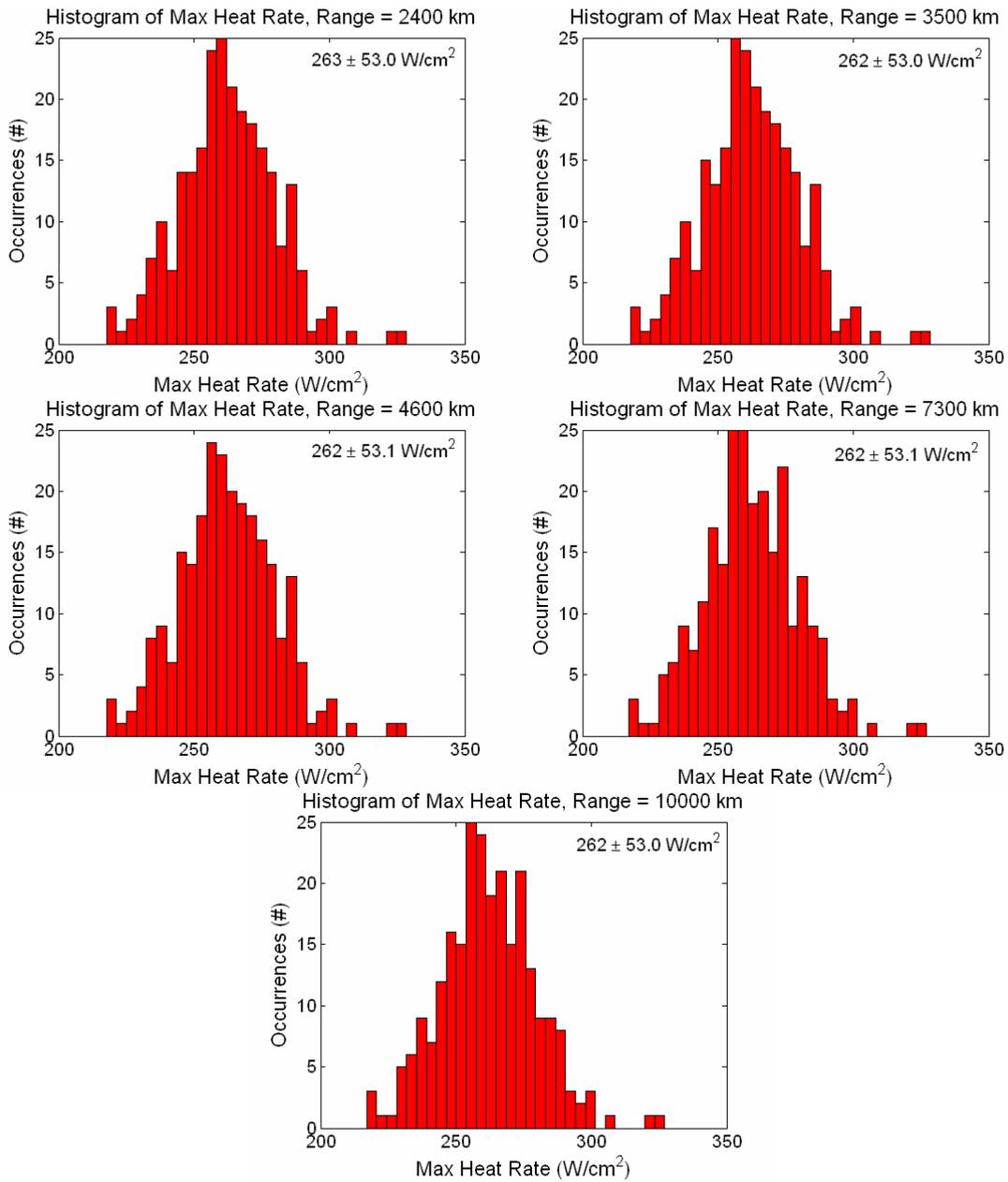


Figure E.12: Max Heat Rate Histograms, Low Loft Enhanced Algorithm, All Ranges

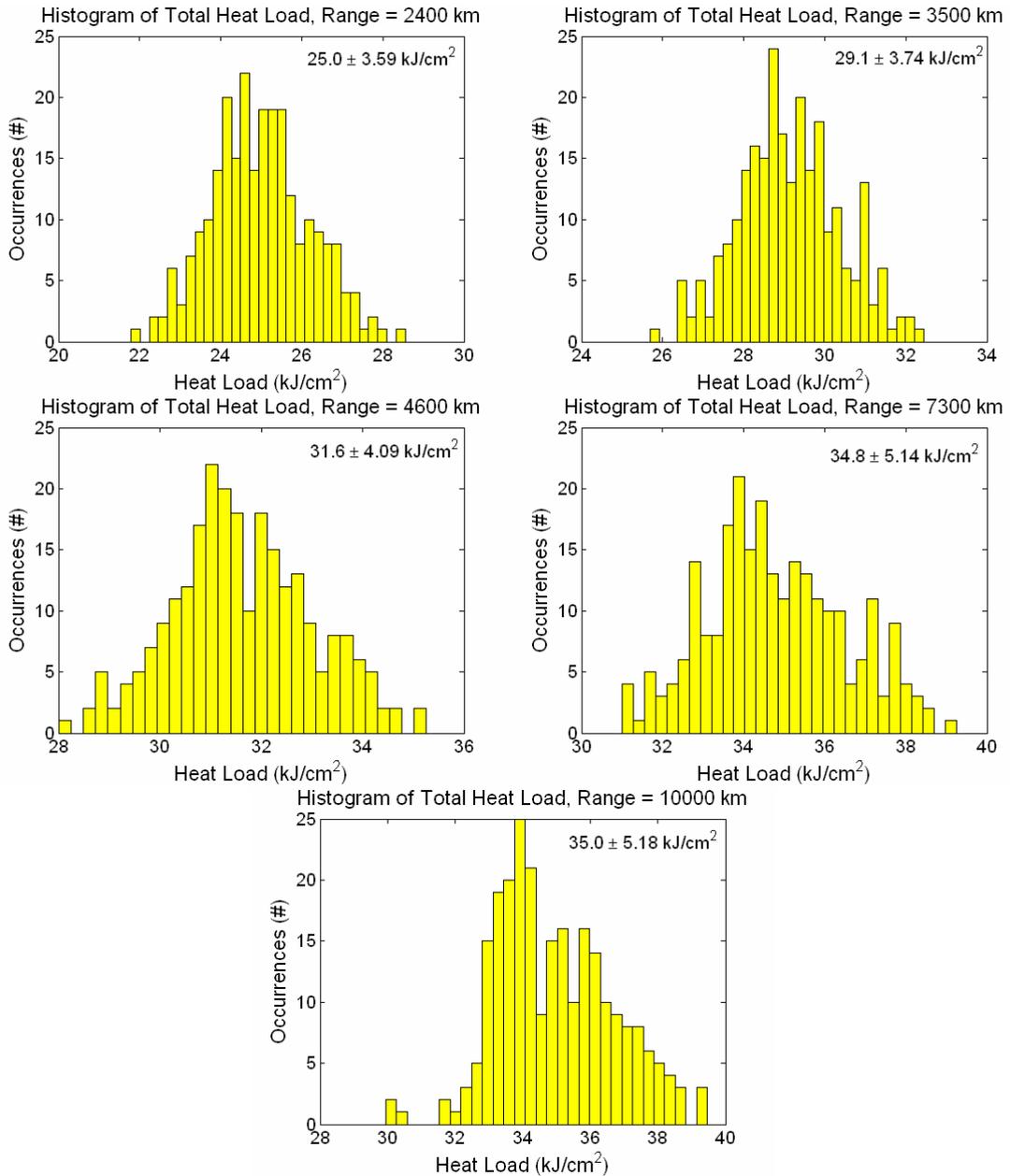


Figure E.13: Total Heat Load Histograms, Low Loft Enhanced Algorithm, All Ranges

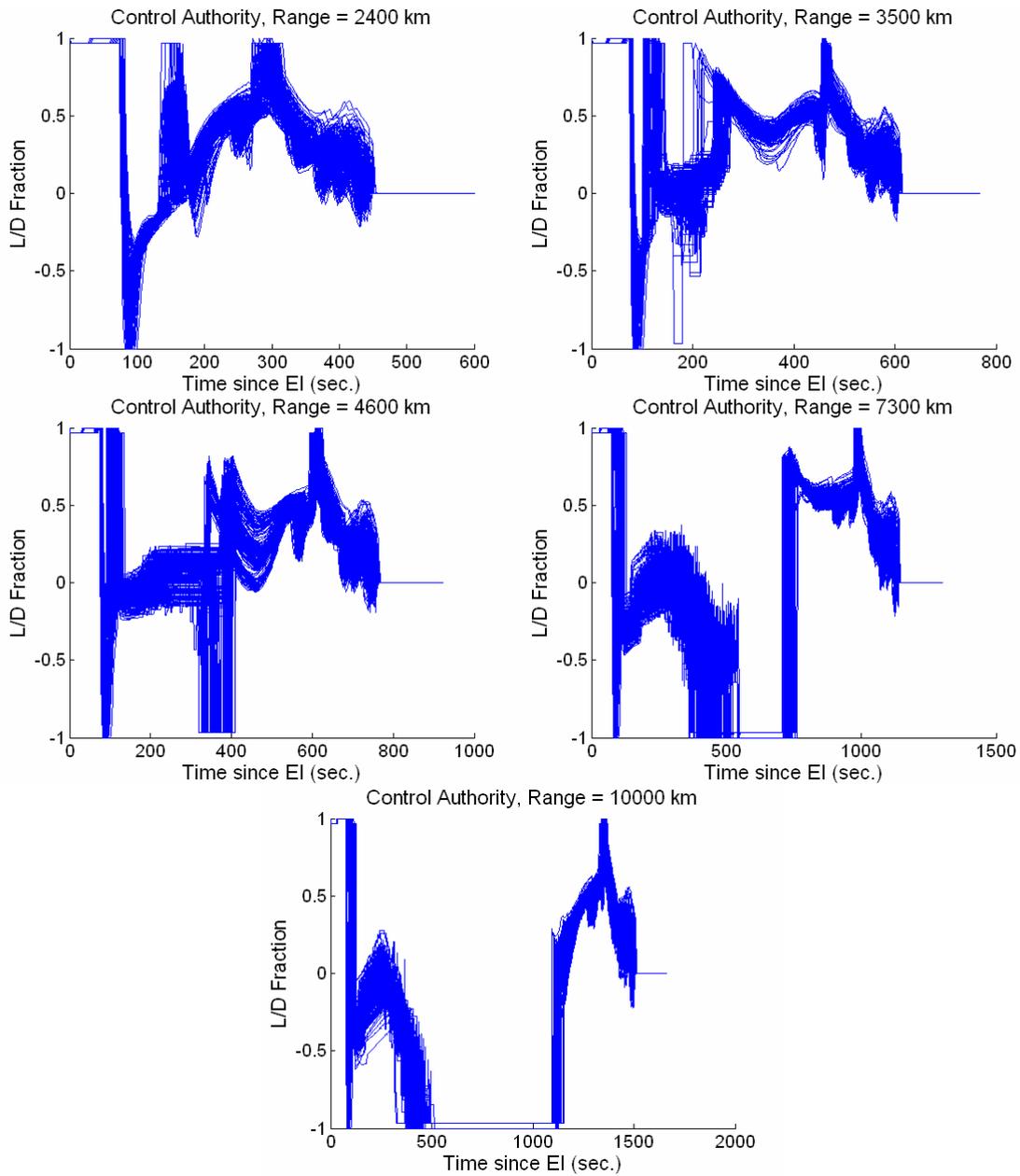


Figure E.14: Commanded L/D Fraction, Low Loft Enhanced Algorithm, All Ranges

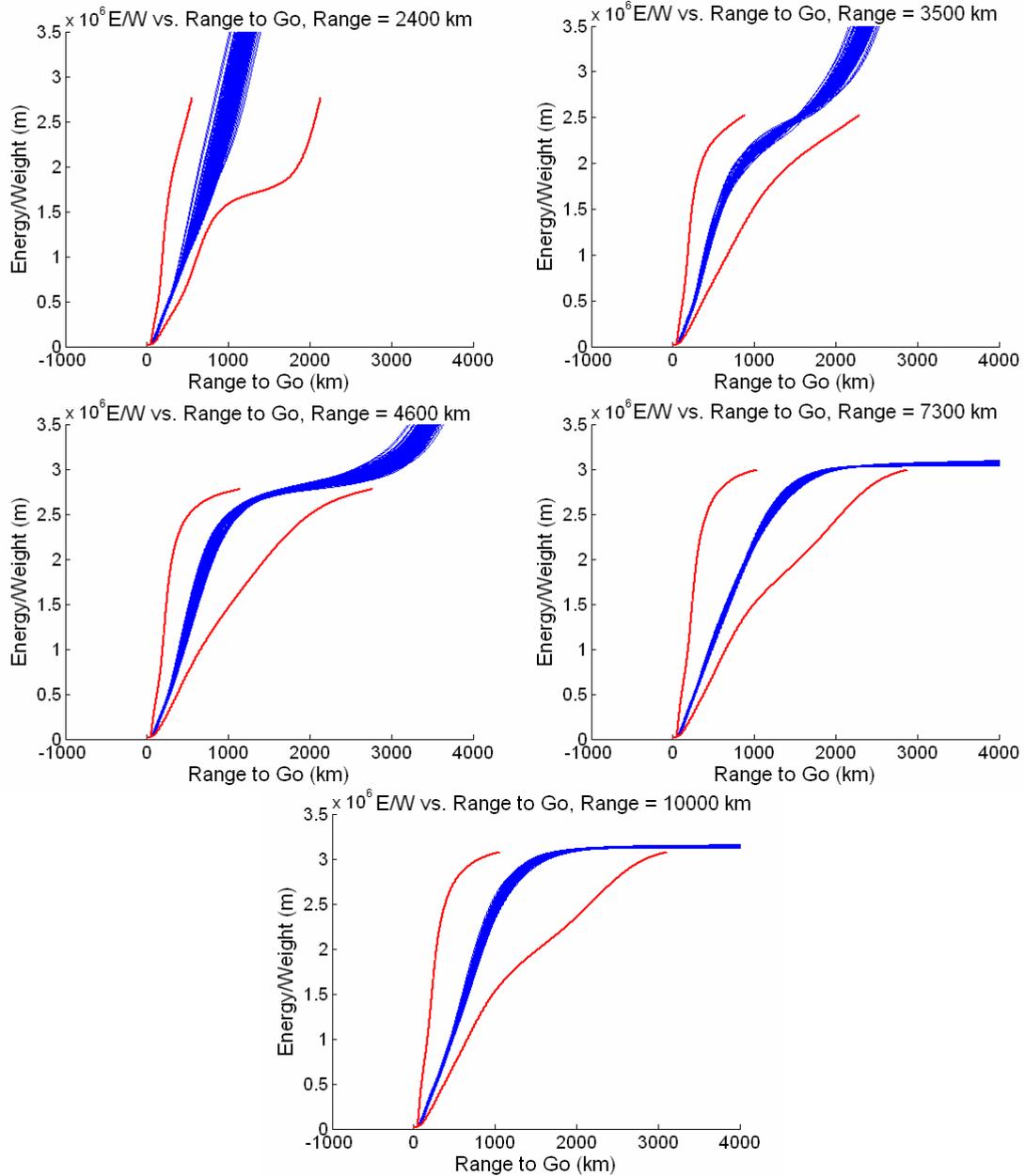


Figure E.15: Energy Bucket Traces, Low Loft Enhanced Algorithm, All Ranges

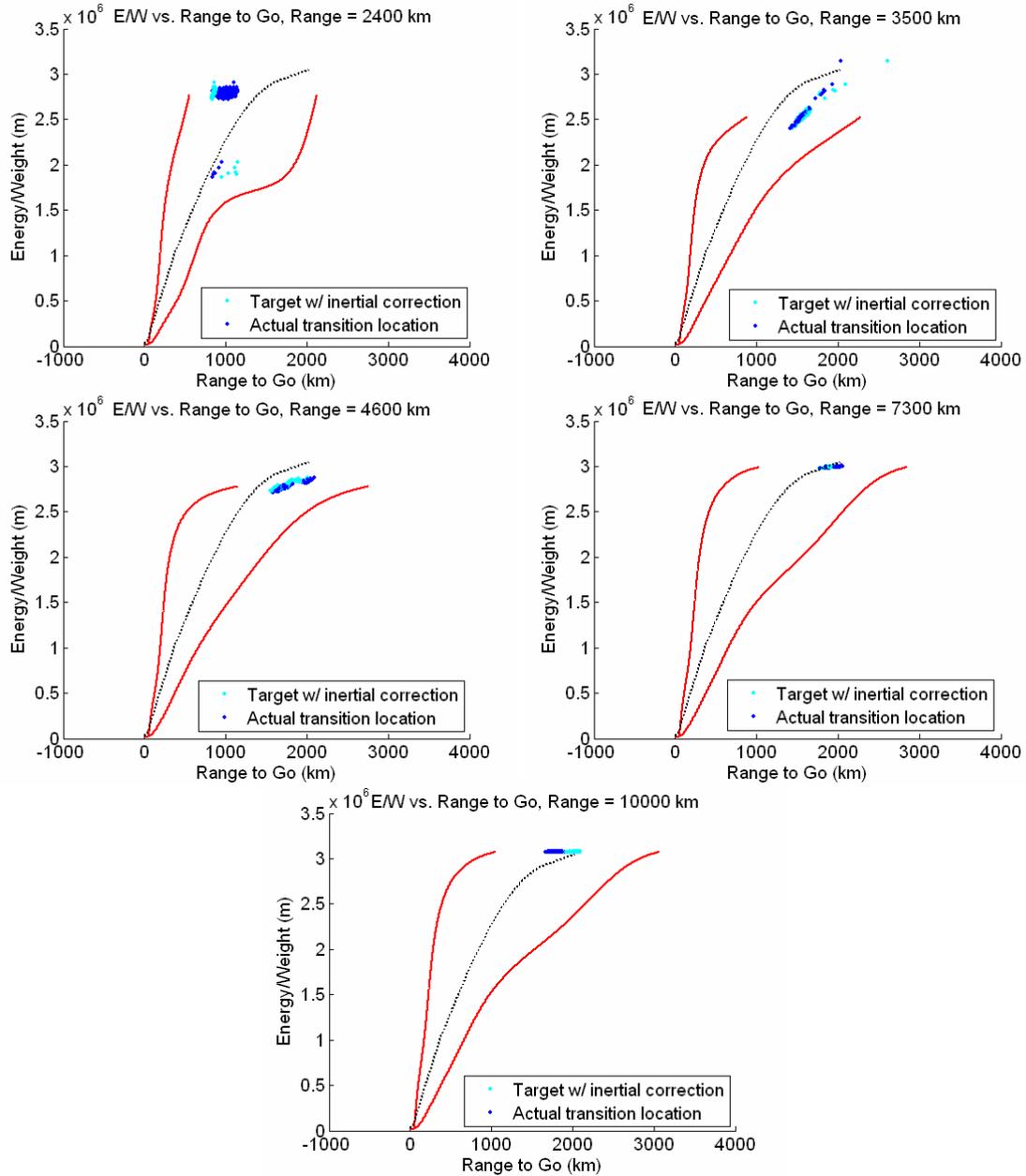


Figure E.16: Energy Bucket Transitions, Low Loft Enhanced Algorithm, All Ranges

E.3 High Loft Enhanced Algorithm

Figure E.17 through Figure E.24 show the metrics for the Monte Carlo series investigating the performance of the High Loft Enhanced version of the algorithm. In this version, the Downcontrol, Upcontrol, and Ballistic phases of the original Apollo algorithm are replaced by PredGuid, resulting in a high-altitude loft. In addition, this version includes the new Final phase reference trajectory and range estimation method.

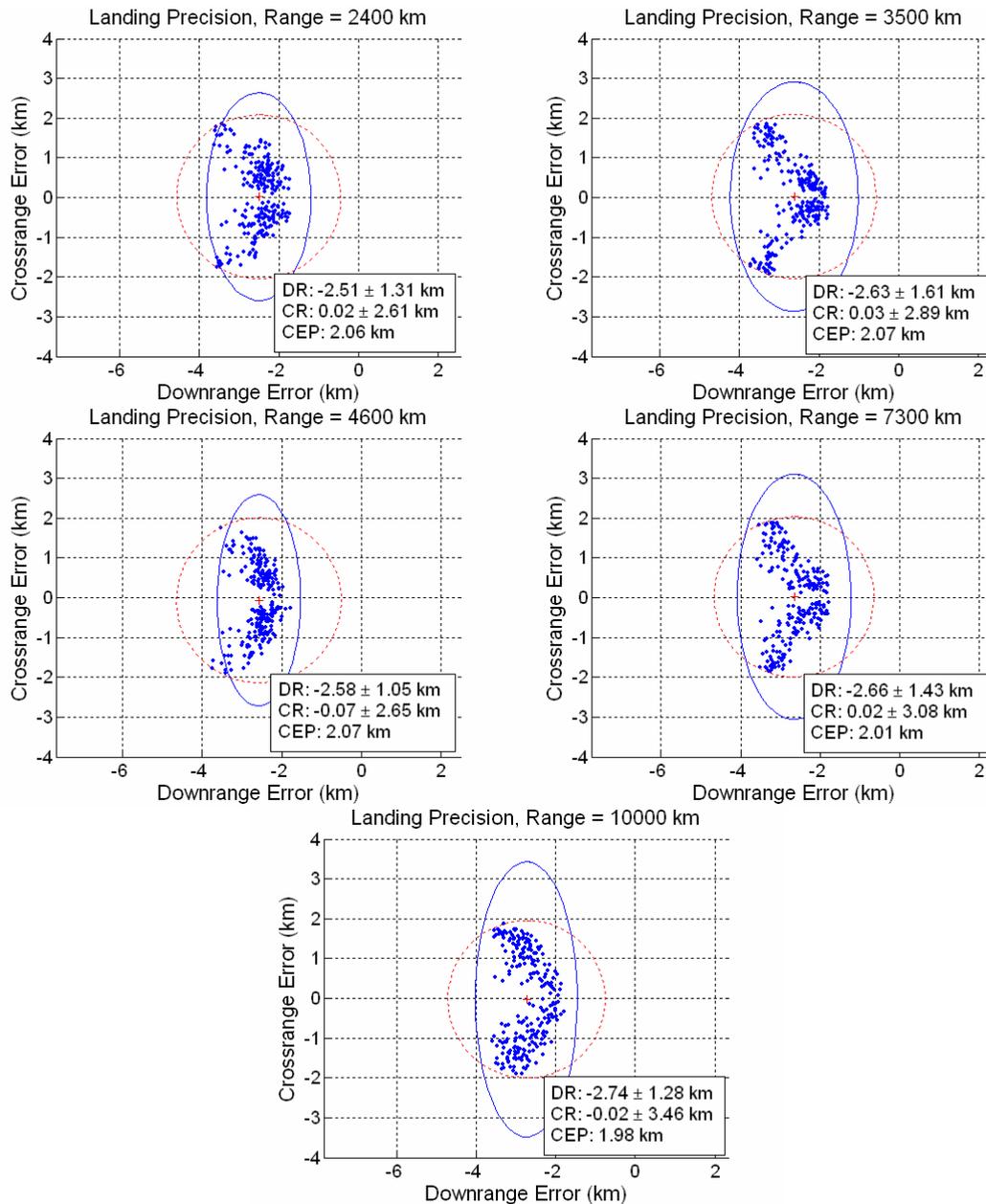


Figure E.17: Landing Precision Scatter Plots, High Loft Enhanced Algorithm, All Ranges

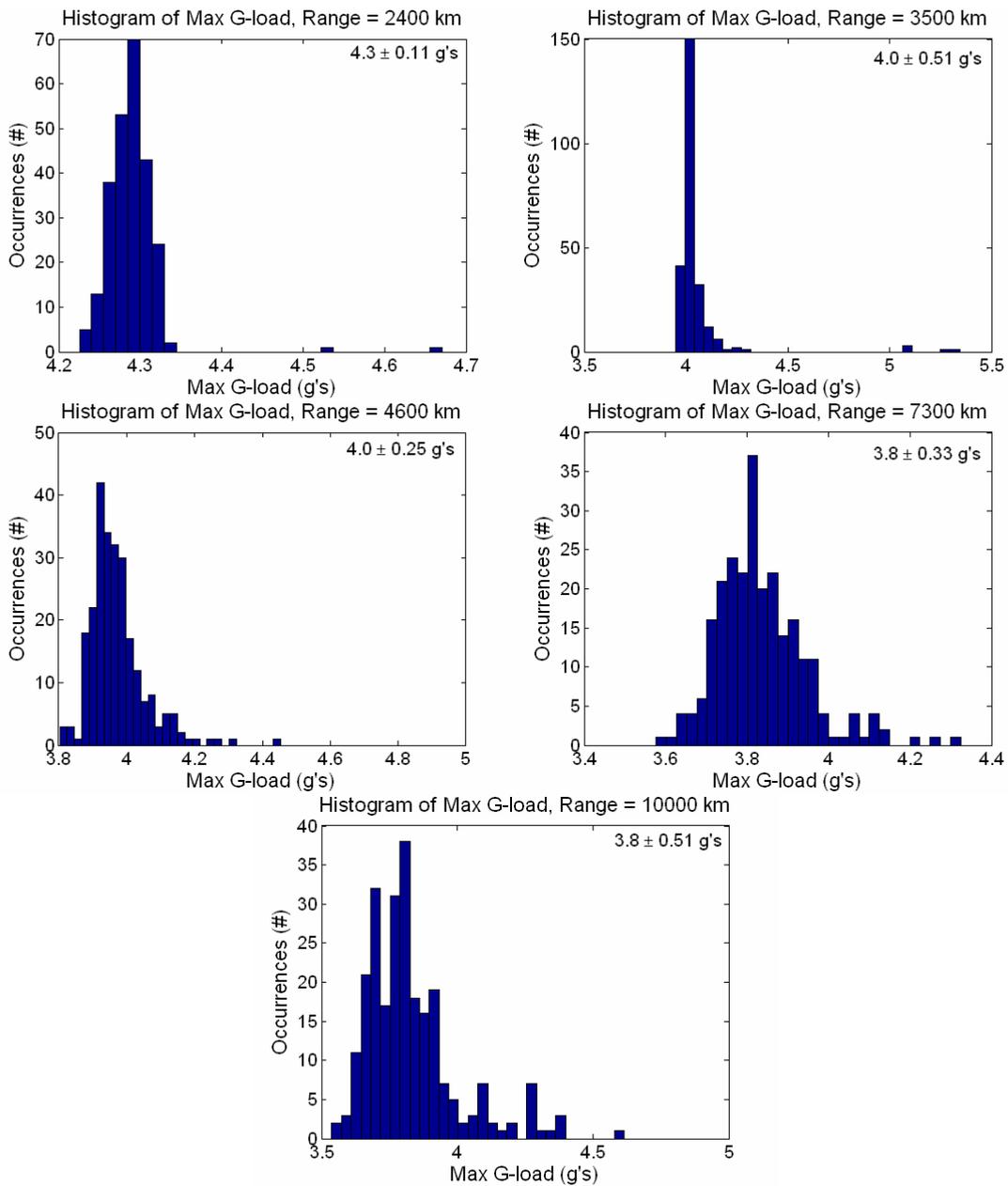


Figure E.18: Max G-load Histograms, High Loft Enhanced Algorithm, All Ranges

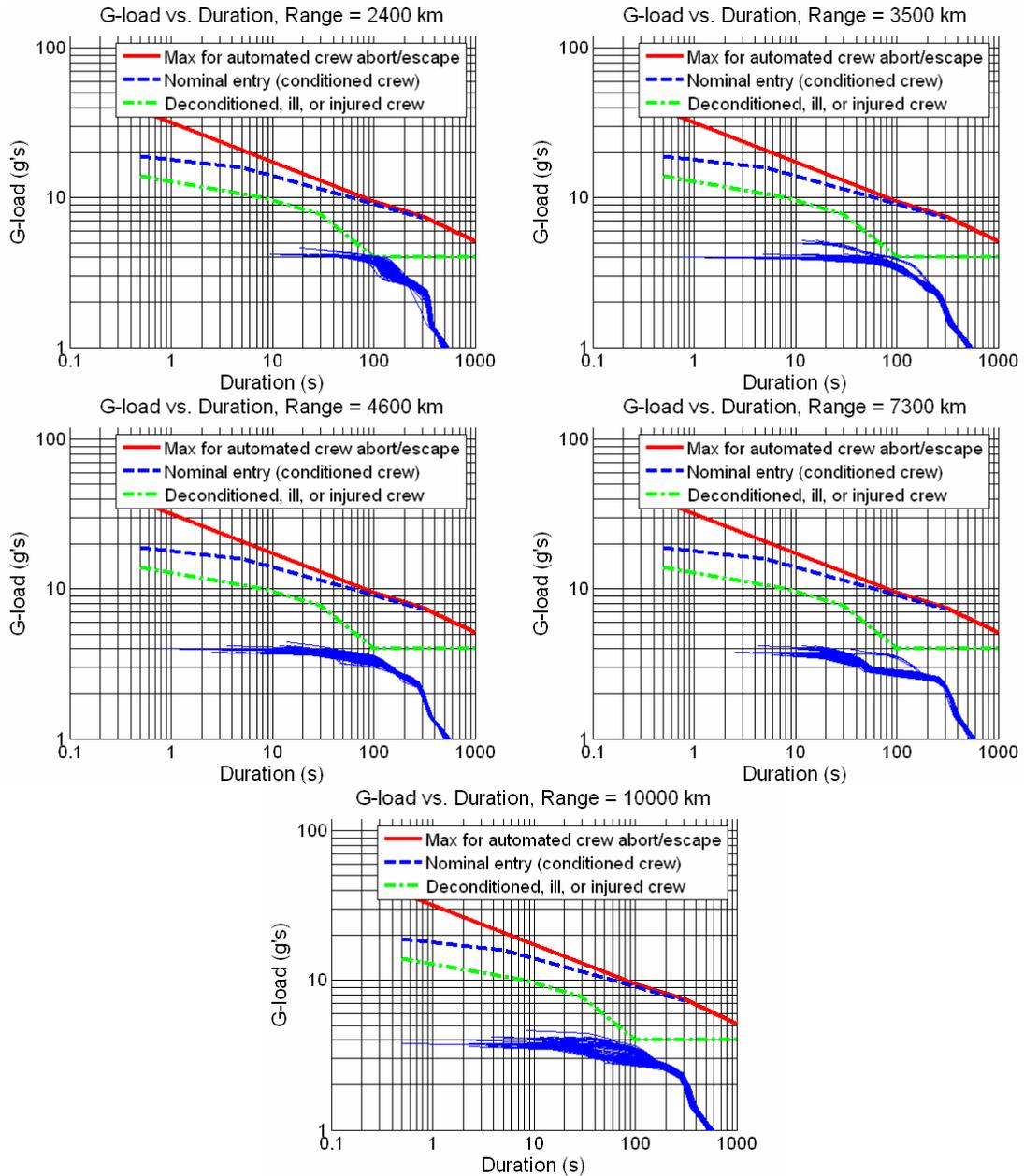


Figure E.19: Duration-based G-loads, High Loft Enhanced Algorithm, All Ranges

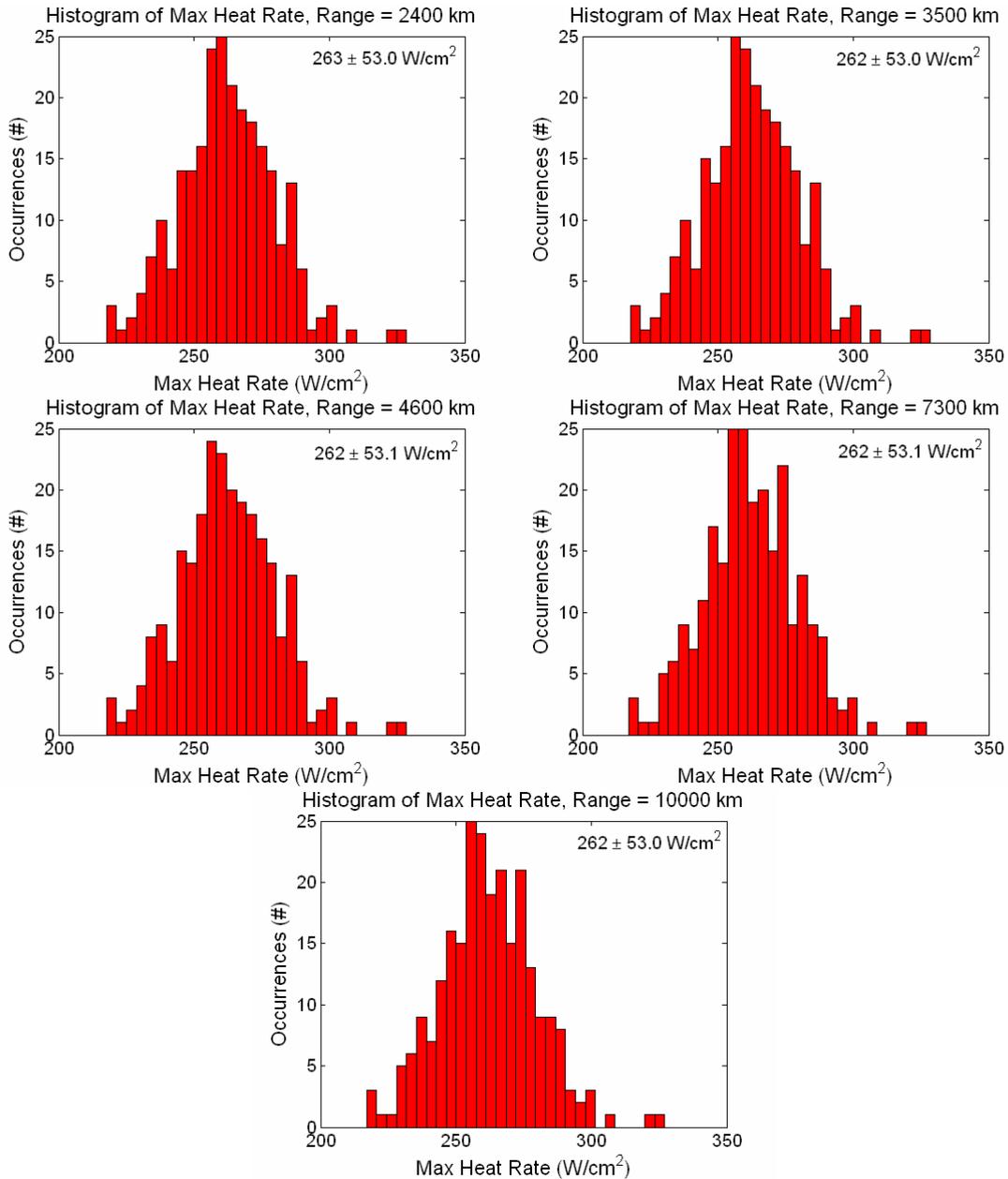


Figure E.20: Max Heat Rate Histograms, High Loft Enhanced Algorithm, All Ranges

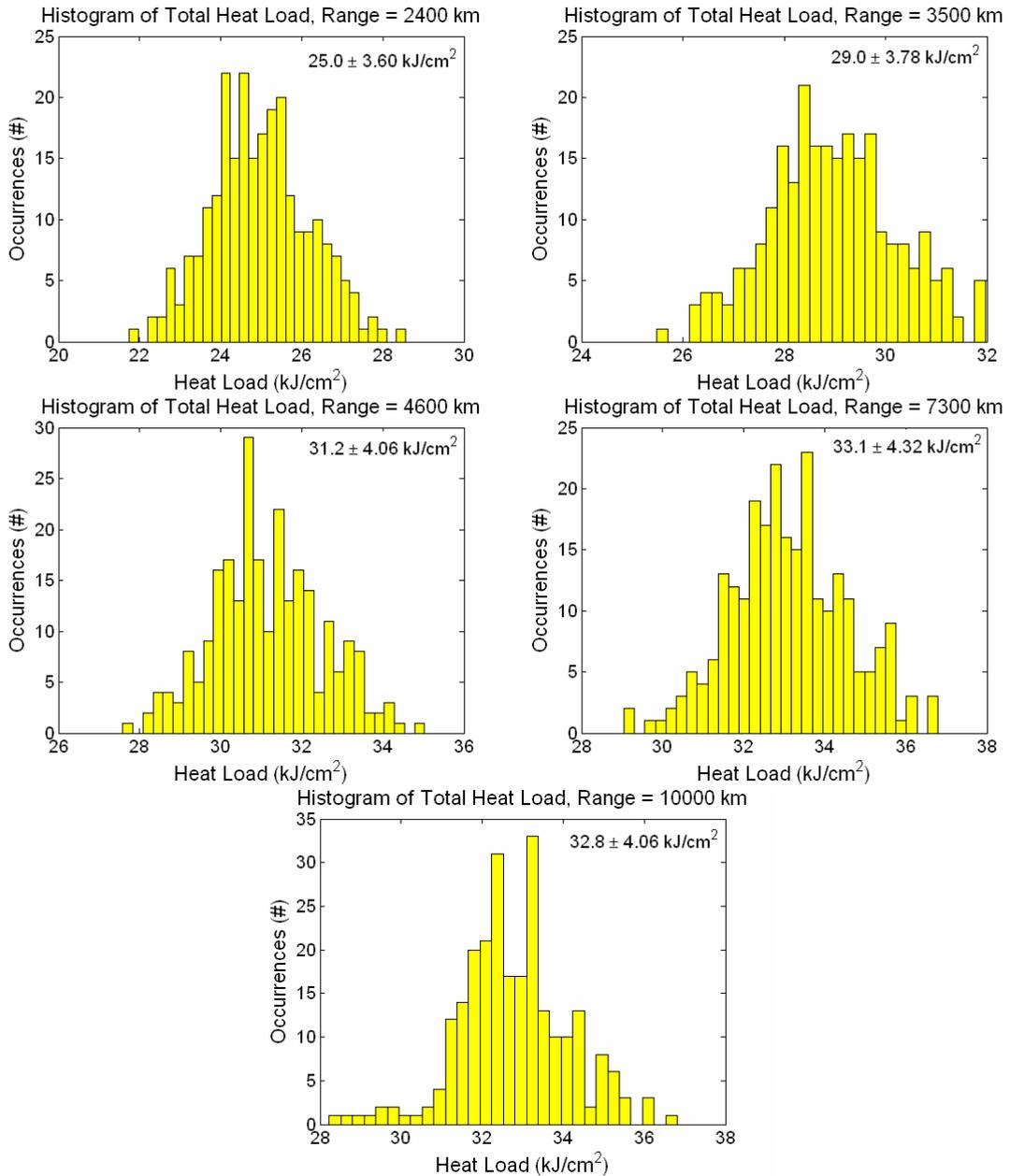


Figure E.21: Total Heat Load Histograms, High Loft Enhanced Algorithm, All Ranges

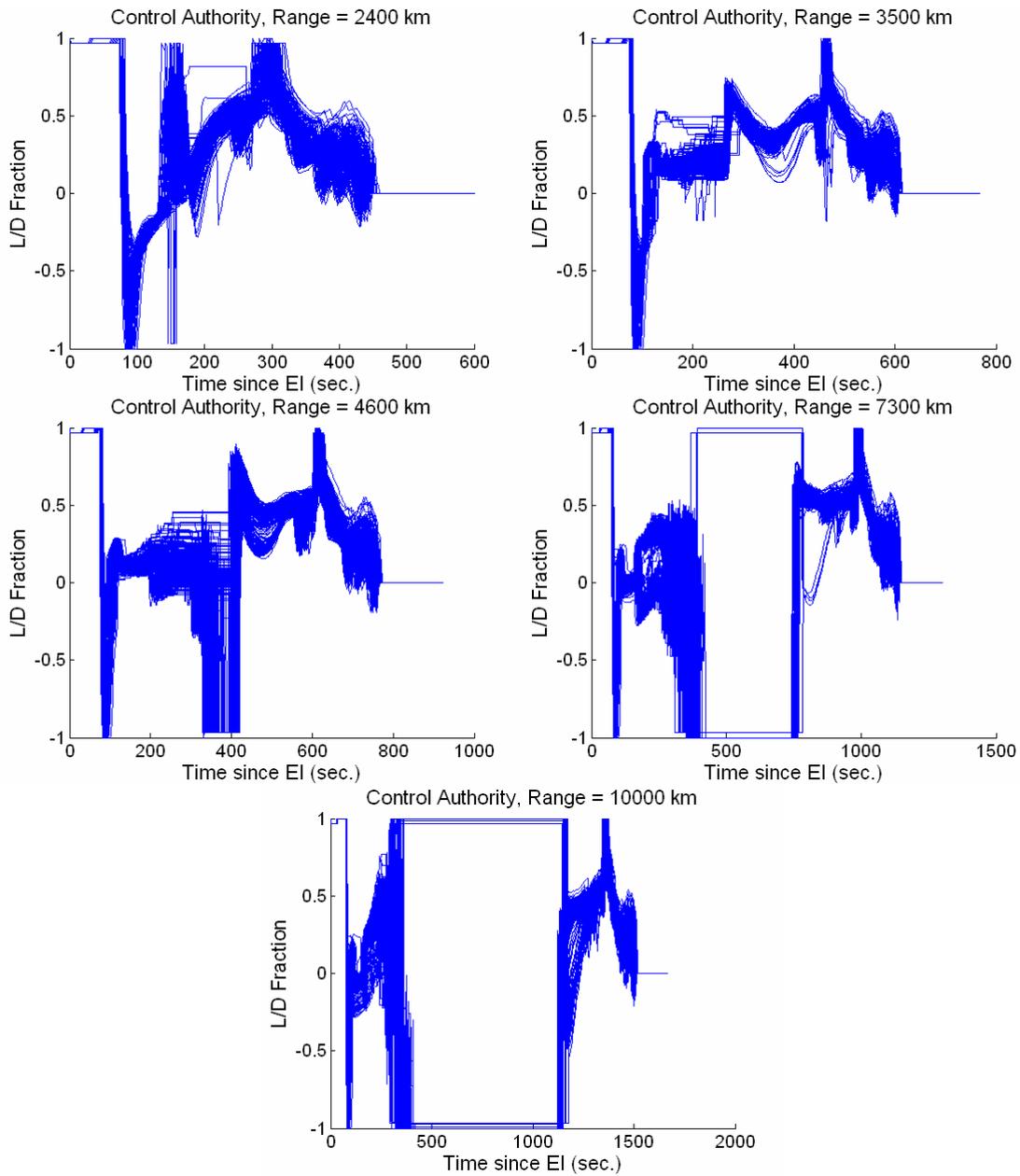


Figure E.22: Commanded L/D Fraction, High Loft Enhanced Algorithm, All Ranges

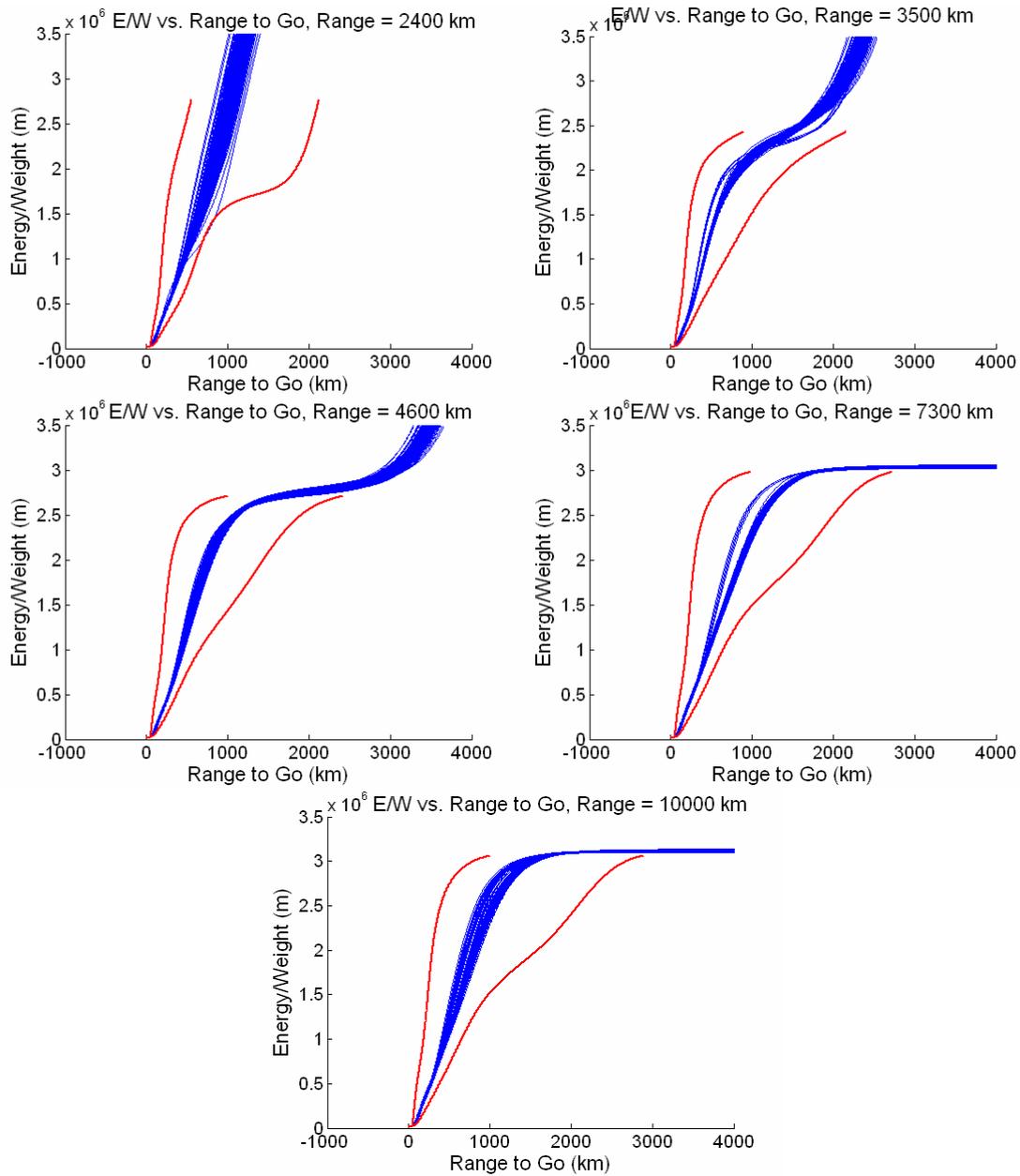


Figure E.23: Energy Bucket Traces, High Loft Enhanced Algorithm, All Ranges

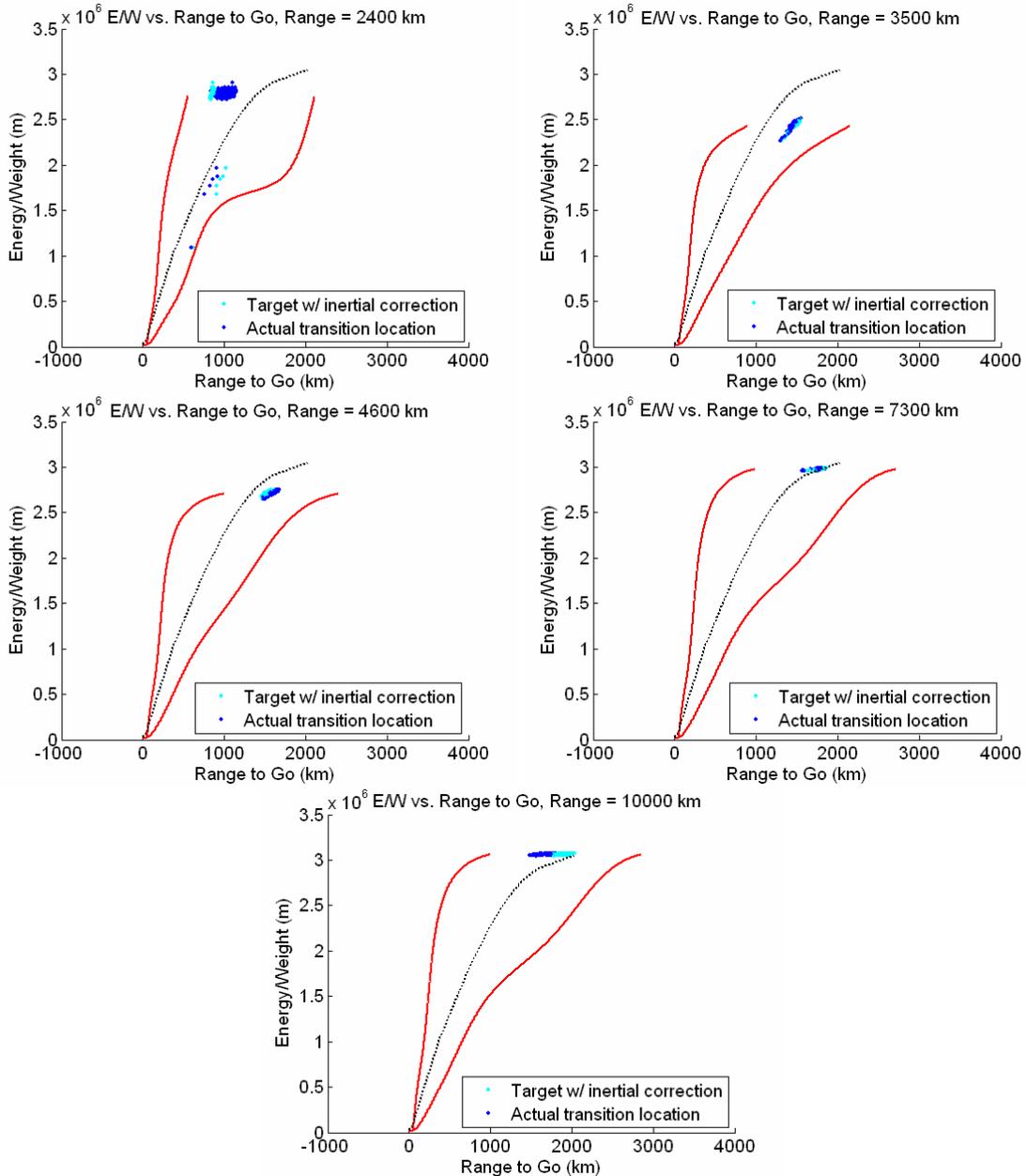


Figure E.24: Energy Bucket Transitions, High Loft Enhanced Algorithm, All Ranges

[This Page Intentionally Left Blank]

References

- [1] The White House Website Presidential News and Speeches, “President Bush Announces New Vision for Space Exploration Program,” URL: www.whitehouse.gov/news/releases/2004/01/20040114-3.html [cited 26 October 2005]
- [2] Powell, R., “Numerical Roll Reversal Predictor Corrector Aerocapture and Precision Landing Guidance Algorithms for the Mars Surveyor Program 2001 Missions,” AIAA Paper 1998-4574, 1998.
- [3] Haroz, C.S., *A Predictor-Corrector Guidance Algorithm Design for a Low L/D Autonomous Re-entry Vehicle*. S.M. Thesis, Department of Aeronautics and Astronautics, MIT, 1999.
- [4] DiCarlo, J.L., *Aerocapture Guidance Methods for High Energy Trajectories*. S.M. Thesis, Department of Aeronautics and Astronautics, MIT, June 2003.
- [5] Mease, K. and McCreary, F., “Atmospheric Guidance Law for Planar Skip Trajectories,” AIAA Paper 85-1818, 1985.
- [6] Kuo, Z. and Liu, K., “Explicit Guidance of Aeroassisted Orbital Transfer Using Matched Asymptotic Expansions,” *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, Jan.-Feb. 2002, pp. 80-87.
- [7] Cerimele, C.J. and Gamble, J.D., “A Simplified Guidance Algorithm for Lifting Aeroassist Orbital Transfer Vehicles,” AIAA Paper 85-0348, 1985.
- [8] Carman, G., Ives, D., Geller, D., “Apollo-Derived Mars Precision Lander Guidance,” AIAA Paper 98-4570, 1998.
- [9] Morth, R., “Reentry Guidance for Apollo,” MIT/IL R-532 Vol. I, 1966.
- [10] Rousseau, S., Perot, E., Graves, C., Masciarelli, J, and Queen, E., “Aerocapture Guidance Algorithm Comparison Campaign,” AIAA Paper 2002-4822, 2002.
- [11] Graves, C. and Harpold, J., “Re-Entry Targeting Philosophy and Flight Results from Apollo 10 and 11,” AIAA Paper 70-28, 1970.
- [12] Orloff, R.W., *Apollo by the Numbers: A Statistical Reference*, NASA SP-2000-4029, NASA History Division, Washington, D.C., 2000.

- [13] Shampine, L.F., and Reichelt, M.W., "The Matlab ODE Suite," URL: www.mathworks.com/access/helpdesk/help/pdf_doc/otherdocs/ode_suite.pdf [cited 23 January 2006]
- [14] *U.S. Standard Atmosphere, 1962*, United States Committee on Extension to the Standard Atmosphere, Washington, D.C., 1962.
- [15] Chapman, D., "An Approximate Analytical Method for Studying Entry into Planetary Atmospheres," NACA Technical Note 4276, 1958.
- [16] Tauber, M. and Sutton, K., "Stagnation-Point Radiative Heating Relations for Earth and Mars Entries," *Journal of Spacecraft and Rockets*, Vol. 28, No. 1, 1991, pp. 40-42.
- [17] "System Requirements for the Crew Exploration Vehicle Element." NASA Document CXP-10001, 2005.
- [18] "NASA Solicitation: Conceptual Design of an Air Bag Landing Attenuation System for the Crew Exploration Vehicle." Langley Research Center Press Release, Dec. 14, 2005.
- [19] "International Space Station Flight Crew Integration Standard," NASA-STD-3000/T, August 1995.
- [20] Naidu, D., "Three-Dimensional Atmospheric Entry Problem Using Method of Matched Asymptotic Expansions," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 25, No. 5, 1989, pp. 660-667.
- [21] Kuo, Z. and Vinh, N., "Improved Matched Asymptotic Solutions for Three-Dimensional Atmospheric Skip Trajectories," *Journal of Spacecraft and Rockets*, Vol. 34, No. 4, 1997, pp. 496-502.
- [22] Teahan, G., *Propulsive Guidance Analysis for Atmospheric Entry Skip Trajectories*. S.M. Thesis, Department of Aeronautics and Astronautics, MIT, June 2006.
- [23] *Guidance and Navigation for Entry Vehicles*, NASA SP-8015, Nov. 1968.
- [24] Striepe S.A., Queen E.M., Powell R.W., Braun R.D., Cheatwood F.M., Aguirre J.T., Sachi L.A. and Lyons D.T., "An Atmospheric Guidance Algorithm Testbed for Mars Surveyor Program 2001 Orbiter and Lander," AIAA Paper 98-4569, Aug 1998.