Managing Energy and Mode Transitions in Skip Entry Guidance for Lunar Return Trajectories

by

Melanie A. Miller

S.B., Aeronautical and Astronautical Engineering
Massachusetts Institute of Technology, 2004

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2007

Author: ______________________________
Department of Aeronautics and Astronautics
May 25, 2007

Certified by: [illegible]
Gregg H. Barton
Group Leader, Mission Design and Analysis
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by: [illegible]
John J. Deyst, Jr., Sc.D.
Professor of Aeronautics and Astronautics
Thesis Advisor

Accepted by: [illegible]
Jaime Peraire, Ph.D.
Professor of Aeronautics & Astronautics
Chair, Committee on Graduate Students

Managing Energy and Mode Transitions in Skip Entry Guidance for Lunar Return Trajectories

by

Melanie A. Miller

Submitted to the Department of Aeronautics and Astronautics
on May 25, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Skip entry trajectories provide a technique for returning an astronaut crew from the Moon to a continental United States landing site at any time during the lunar month. This approach to atmospheric entry requires that the guidance system be capable of precisely targeting a vast array of downrange distances. To meet this objective, Draper Laboratory has developed a baseline skip entry guidance algorithm which is a blend of the original Apollo guidance logic and a numeric targeting algorithm, PredGuid, that is used for the skip portion of the entry. The addition of PredGuid greatly improved the algorithm's performance for skip trajectories, but numerous simplifications and Apollo-based empirical relationships still limit the capability of the algorithm and may be unsuitable for other vehicle configurations. This thesis presents enhancements to the Draper baseline which redesign the energy management system and phase transition logic. A model-based predictor has been developed to determine the type of trajectory that is necessary to cover the target range and the appropriate time to transition to the next flight mode, based on the trajectory type. The direct entry capability has been improved and expanded by incorporating a variable constant drag policy to manage energy, and by designing specific direct entry reference trajectories which are used by the path-following controller in the final descent phase. In addition, an intermediate loft regime is introduced to bridge the range capability between direct and skip entries. These upgrades to the baseline algorithm greatly improve its robustness to uncertainties encountered through the entry, as demonstrated via Monte Carlo simulations, using the Crew Exploration Vehicle capsule concept. This algorithm design also offers the ability to optimize a skip trajectory to meet a particular set of objectives by modifying the reference skip bank angle.

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

Thesis Advisor: John J. Deyst, Jr., Sc.D.
Title: Professor of Aeronautics and Astronautics

Acknowledgments

There are a number of people without which this thesis would not have been possible. First, I would like to thank the Charles Stark Draper Laboratory for allowing me the opportunity to pursue my graduate degree at MIT and the privilege of working with so many knowledgeable and helpful people. I would also like to thank my advisor Gregg Barton, for being an invaluable resource throughout this endeavor and always coming up with just the right advice or insight at critical moments. You have the distinction of being the only person for whom I have performed more than 18,000 Monte Carlo simulations which took over 800 hours, and I hope you will be the last! Several other Draper staff members were instrumental in the development of this work and my understanding of the problem, including Sarah Bairstow, who not only provided the basis for my thesis, but also had answers and advice when I was first getting started; and Matthew Neave and Roberto Pileggi, who have been my CEV resources throughout this project and have never tired (at least outwardly) of answering my often inane questions. Professor Deyst has also been helpful in providing a welcome outside perspective to this work. Chris Stoll and Rebecca Masterson, thanks for looking out for me during my first year and introducing me to missiles. Thanks also to Garrett Teahan and Alison Kremer, for being fantastic officemates and providing entertainment when I needed it most, and to Phil Springmann and Glenn Tournier, the best roommates a girl could ask for. You guys will always know how to make me laugh. I also want to thank Joshua Hardy for endlessly proofreading this manuscript, transcribing movie quotes, and generally being crazy enough to stick with me through these past two years. You have always been the light at the end of this tunnel, and I can't thank you enough for your support.

Finally, I would like to thank my parents and my sisters. These past several months have been challenging for all of us, but your love and encouragement and late night phone calls have meant the world to me. I can't imagine having undertaken this without you.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Mission Design Technology Internal Research and Development Project #21140.

Melanie A. Miller

Contents

1 Introduction **21**

1.1 Key Concepts 22

1.1.1 Direct Entry Trajectory 23

1.1.2 Skip Trajectory 23

1.1.3 Bank Angle Modulation 24

1.2 Motivation 26

1.3 Thesis Objective 27

1.4 Thesis Overview 28

2 Simulation Environment **29**

2.1 Reference Coordinate Frames 29

2.2 Coordinate Transformations 31

2.3 Environment Models 34

2.3.1 Earth Gravity Model 34

2.3.2 Earth Atmosphere 35

2.4 Vehicle Model 35

2.5 Equations of Motion 37

2.5.1 Translational Motion 37

2.5.2 Rotational Motion 39

2.5.3 Bank Angle Definition 41

2.5.4 Planar Equations of Motion for a Point Mass 42

2.6 Aerodynamic Heating 45

2.7 Simulation Initialization . . . 46
2.7.1 Vehicle Initial Conditions . . . 46
2.7.2 Target Location . . . 46
2.8 Simulation Termination Conditions . . . 47
2.9 Monte Carlo Dispersed Parameters . . . 47

3 Guidance Algorithm Metrics **49**
3.1 Performance Based Metrics . . . 49
3.1.1 Landing Accuracy . . . 50
3.1.2 *g* Loads . . . 51
3.1.3 Aeroheating . . . 52
3.2 Margin Based Metrics . . . 53
3.2.1 Control Saturation . . . 54
3.2.2 Final Phase Energy Bucket . . . 55

4 Apollo Guidance Algorithm **59**
4.1 Overview . . . 59
4.2 Targeting . . . 61
4.3 Initial Roll . . . 61
4.4 Huntest . . . 62
4.4.1 Constant Drag . . . 64
4.5 Final . . . 67
4.5.1 Reference Trajectory . . . 68
4.6 Lateral Logic . . . 70

5 Draper PredGuid Algorithm **73**
5.1 Overview . . . 73
5.2 PredGuid . . . 73
5.2.1 PredGuid's Target Range . . . 74
5.2.2 PredGuid Termination Conditions . . . 75
5.3 Upcontrol and Ballistic . . . 76

5.3.1 Upcontrol 78
5.3.2 Ballistic 78
5.4 Final 79

6 Problem Description 83
6.1 Overview 83
6.2 Energy Management and Constant Drag 84
6.3 Final Phase Reference Trajectory 85
6.4 Shallow Entry Test 87
6.5 Determining Whether Skip is Necessary 92
6.6 Transitioning to PredGuid 93

7 EMT Predictor Algorithm 97
7.1 Overview 97
7.2 Design Philosophy and Objectives 99
7.3 Direct Entry 99
7.3.1 Variable Constant Drag Level 100
7.3.2 Initial Roll L/D 103
7.3.3 Gamma Turn Controller 107
7.3.4 Final Phase Reference Trajectory 110
7.4 Skip/Loft 114
7.4.1 PredGuid Changes for Loft 116
7.5 EMT Predictor 119
7.5.1 Trajectory Type Selection 120
7.5.2 Transition Time Determination 120
7.5.3 Shallow Entry Correction 123
7.6 Lateral Steering 125
7.7 Summary of Initial Roll and Huntest 125

8 Results 129
8.1 Overview 129

8.2 Skip/Loft Trajectory Shaping . . . 130
8.2.1 Initial Roll Bank Angle . . . 130
8.2.2 Skip Bank Angle . . . 130
8.3 Precision Landing Downrange Capability . . . 133
8.4 Entry Flight Path Angle . . . 134
8.5 Monte Carlo Results . . . 138
8.5.1 Landing Accuracy . . . 139
8.5.2 *g* Loads . . . 143
8.5.3 Maximum Heat Rate . . . 144
8.5.4 Total Heat Load . . . 145
8.5.5 Control Saturation . . . 146
8.5.6 Energy Bucket . . . 150

9 Conclusion 157
9.1 Summary . . . 157
9.2 Future Work . . . 158

A Additional Results 163
A.1 Predictor algorithm results . . . 163
A.2 Draper Baseline Algorithm Results . . . 178

B Detailed Description of EMT Predictor Enhancements 193
B.1 Overview . . . 193
B.2 Targeting Subroutine . . . 193
B.2.1 L/D Estimation . . . 194
B.2.2 RT Selector . . . 195
B.3 Initial Roll Phase . . . 198
B.3.1 Preentry Attitude Hold . . . 198
B.3.2 EMT Predictor . . . 200
B.3.3 Gamma Turn Controller . . . 217
B.4 Huntest Phase . . . 219

B.4.1 Constant Drag Subroutine 220
B.5 Upcontrol Phase 220
B.6 Ballistic Phase 222
B.7 PredGuid Subroutine 223
B.7.1 Final Range Estimation 226
B.8 Lateral Logic Subroutine 227

List of Figures

1-1 Atmospheric entry [13] 22

1-2 Skip entry trajectory [16] 23

1-3 Direct and skip entry trajectories 24

1-4 Vehicle geometry and aerodynamic forces (Credit: G. Barton & S. H. Bairstow) 25

1-5 Lift vector and bank angle 25

2-1 Inertial coordinate frame at time $t = 0$ 30

2-2 Body reference frame (b) and stability reference frame (s), with $\beta = 0$ 31

2-3 CEV concept's capsule shape [13] 36

2-4 Aerodynamic data at trim angle of attack 37

2-5 Coordinate system for planar, point mass equations 43

3-1 Landing accuracy 50

3-2 Sample landing accuracy comparison plot 51

3-3 Duration-based g loads 52

3-4 Heat rate, integrated to produce total heat load 53

3-5 Bank angle command history 55

3-6 Sample energy bucket 57

4-1 Apollo entry guidance phases 60

4-2 Apollo guidance flowchart 61

4-3 Initial Roll phase logic 63

4-4 Huntest phase logic 65

4-5 Reference L/D ratios for different Constant Drag values 67

4-6 Final phase logic 68
4-7 Final phase reference trajectory for Apollo guidance algorithm 70
4-8 Lateral Logic subroutine 71

5-1 Drag and altitude rate for first and second entries 77
5-2 Upcontrol phase logic 79
5-3 Ballistic phase logic 80
5-4 Draper baseline and Apollo reference trajectories 81

6-1 Drag vs. E/W for 1200 *nm* direct entry Monte Carlo trials 85
6-2 Draper baseline energy bucket for 1300 *nm* target range 88
6-3 Reference trajectory shape change with initial drag 88
6-4 Apollo 11 entry corridor [5] 90
6-5 Orientation of lift vector at 0.05 g's 91
6-6 Trajectory shaping from PredGuid start time 93
6-7 Range to go at PredGuid start time 94

7-1 Predictor algorithm phases 98
7-2 Constant Drag levels for direct entry target ranges 102
7-3 Drag for various Initial Roll L/D fractions 104
7-4 Altitude rate for various Initial Roll L/D fractions 104
7-5 Flight path angle for various Initial Roll L/D fractions 105
7-6 Initial Roll $\dot{h}$ and Constant Drag reference $\dot{h}$ for 0.5 L/D fraction 106
7-7 Initial Roll $\dot{h}$ and Constant Drag reference $\dot{h}$ for all positive L/D fractions . . 106
7-8 Initial Roll flight path angles and Constant Drag references 107
7-9 Without γ_{turn} Controller 110
7-10 With γ_{turn} Controller 110
7-11 Final phase reference trajectories for direct entry and skip/loft 112
7-12 Reference flight path angles for Constant Drag 112
7-13 Reduction in Final phase energy bucket width for direct entry 113
7-14 Reference trajectory selector pseudocode 114

7-15 Apollo algorithm loft during Final phase 115
7-16 Altitude oscillations in loft trajectory due to PredGuid commands 117
7-17 PredGuid command history for loft trajectory 117
7-18 Sequence of PredGuid integration termination conditions for loft (left) and skip (right) trajectories 118
7-19 γ error from Constant Drag reference during direct entry prediction 122
7-20 Predictor algorithm: Initial Roll phase logic 126
7-21 Predictor algorithm: Huntest phase logic 127

8-1 Larger Initial Roll bank angles delay transition to Upcontrol 131
8-2 Variation in aeroheating due to reference skip bank angle 132
8-3 Variation in max g load due to reference skip bank angle 133
8-4 Draper baseline (left) and Predictor (right) downrange footprints 134
8-5 Comparison of trajectory types for different algorithms 134
8-6 Predictor algorithm's increased direct entry capability 135
8-7 Entry flight path angle capability comparison 136
8-8 Bank angle command history, $\gamma_{EI} = -6.6°$, range = 5200 *nm* 137
8-9 Bank angle command history for both algorithms, $\gamma_{EI} = -5°$, range = 1700 *nm* 138
8-10 Summary of landing accuracy for all Monte Carlo target ranges 140
8-11 Monte Carlo landing accuracy, enlarged 140
8-12 Monte Carlo landing accuracy, further enlarged 141
8-13 Large crossrange error in 1300 *nm* target range 141
8-14 Crossrange error case in 1300 *nm* target range 142
8-15 Summary of maximum g loads for all Monte Carlo target ranges 143
8-16 Duration-based g loads for 1050 *nm* target range: Predictor (left) and Draper (right) 144
8-17 Summary of maximum heat rate for all Monte Carlo target ranges 145
8-18 Summary of total heat load for all Monte Carlo target ranges 146
8-19 Summary of lift up saturation for all Monte Carlo target ranges 147

8-20 Atmospheric density scale factor, 2400 nm target range 148

8-21 Bank angle command history, 2400 nm target range 149

8-22 Final phase energy bucket, 2400 nm target range 149

8-23 Summary of lift down saturation for all Monte Carlo target ranges 150

8-24 Predictor algorithm bank command history, range = 4000 nm 151

8-25 Energy buckets for 1050 nm target range 152

8-26 Energy buckets for 1300 nm target range 152

8-27 Energy bucket and landing accuracy for 70° skip bank angle, range = 1300 nm154

8-28 Energy buckets for 1850 nm target range 155

8-29 Energy buckets for 4000 nm target range 155

9-1 Meeting crossrange requirement using one bank reversal 159

A-1 Landing accuracy, Predictor algorithm, target ranges 900 – 2400 nm 164

A-2 Landing accuracy, Predictor algorithm, target ranges 2500 – 5400 nm 165

A-3 Max g load histograms, Predictor algorithm, target ranges 900 – 2400 nm . 166

A-4 Max g load histograms, Predictor algorithm, target ranges 2500 – 5400 nm . 167

A-5 Duration-based g loads, Predictor algorithm, target ranges 900 – 2400 nm . 168

A-6 Duration-based g loads, Predictor algorithm, target ranges 2500 – 5400 nm . 169

A-7 Maximum heat rate histograms, Predictor algorithm, target ranges 900 – 2400 nm 170

A-8 Maximum heat rate histograms, Predictor algorithm, target ranges 2500 – 5400 nm 171

A-9 Total heat load histograms, Predictor algorithm, target ranges 900 – 2400 nm 172

A-10 Total heat load histograms, Predictor algorithm, target ranges 2500 – 5400 nm173

A-11 Bank angle command history, Predictor algorithm, target ranges 900 – 2400 nm 174

A-12 Bank angle command history, Predictor algorithm, target ranges 2500 – 5400 nm 175

A-13 Legend for bank angle command history figures 175

A-14 Final phase energy buckets, Predictor algorithm, target ranges 900 – 2400 nm 176

A-15 Final phase energy buckets, Predictor algorithm, target ranges 2500 – 5400 *nm* 177
A-16 Landing accuracy, Draper baseline algorithm, target ranges 900 – 2400 *nm* . 179
A-17 Landing accuracy, Draper baseline algorithm, target ranges 2500 – 5400 *nm* 180
A-18 Max *g* load histograms, Draper baseline algorithm, target ranges 900 – 2400 *nm* 181
A-19 Max *g* load histograms, Draper baseline algorithm, target ranges 2500 – 5400 *nm* 182
A-20 Duration-based *g* loads, Draper baseline algorithm, target ranges 900 – 2400 *nm* 183
A-21 Duration-based *g* loads, Draper baseline algorithm, target ranges 2500 – 5400 *nm* 184
A-22 Maximum heat rate histograms, Draper baseline algorithm, target ranges 900 – 2400 *nm* 185
A-23 Maximum heat rate histograms, Draper baseline algorithm, target ranges 2500 – 5400 *nm* 186
A-24 Total heat load histograms, Draper baseline algorithm, target ranges 900 – 2400 *nm* 187
A-25 Total heat load histograms, Draper baseline algorithm, target ranges 2500 – 5400 *nm* 188
A-26 Bank angle command history, Draper baseline, target ranges 900 – 2400 *nm* 189
A-27 Bank angle command history, Draper baseline, target ranges 2500 – 5400 *nm* 190
A-28 Legend for bank angle command history figures 190
A-29 Final phase energy buckets, Draper baseline algorithm, target ranges 900 – 2400 *nm* 191
A-30 Final phase energy buckets, Draper baseline algorithm, target ranges 2500 – 5400 *nm* 192

B-1 EMT Predictor algorithm: Initial Roll 198

List of Tables

2.1 3σ Uncertainties and Dispersions for Monte Carlo Analysis 48

5.1 PredGuid Integration Termination Conditions 76

6.1 L/D and ballistic coefficient values for the Apollo and CEV capsules 91

6.2 Pros and cons of high and low altitude skip trajectories [2] 95

7.1 Final phase reference trajectory initial conditions 111

8.1 Monte Carlo target ranges 139

A.1 Landing accuracy, Predictor algorithm 163

A.2 Landing accuracy, Draper baseline algorithm 178

Chapter 1

Introduction

Safely entering the Earth's atmosphere on a return trip from the Moon is no easy task, as aptly described in this quote from the movie *Apollo 13*:

> "In order to enter the atmosphere safely, the crew must aim for a corridor just two and a half degrees wide. If they're too steep, they'll incinerate in the steadily thickening air; if they're too shallow, they'll ricochet off the atmosphere like a rock skipping off a pond. The re-entry corridor is in fact so narrow that if this basketball were the Earth, and this softball were the Moon, and the two were placed 14 feet apart, the crew would have to hit a target no thicker than this piece of paper." [1]

Once the spacecraft enters the atmosphere in the allowable corridor, the crew's perilous journey home is hardly over. The spacecraft must then fly nimbly through the dense air to the ground, carefully steering toward the landing site while making sure not to injure the crew or damage the spacecraft by decelerating too quickly.

The challenge of returning astronauts safely to Earth after a journey to the Moon has been only an academic problem since the end of the Apollo program. But in January 2004, President George W. Bush set forth a new Vision for Space Exploration, committing the National Aeronautics and Space Administration (NASA) to a new goal of returning astronauts to the Moon by 2020 [14]. The Crew Exploration Vehicle (CEV), under the recently-named Orion program, is the spacecraft that will transport astronauts to and from the Moon. It will be a capsule-type vehicle, similar to the Apollo command module. With respect to Earth reentry, one requirement for CEV that is significantly different from Apollo is precision landing of the capsule on dry land, rather than the familiar water landings of Apollo [11]. Since the acceptable area available for landings on land is much smaller than the area available for water landings, this requirement forces the entry guidance system to be capable of very precise targeting to ensure access to at least one, but ideally many, ground landing sites.

1.1 Key Concepts

The atmospheric entry begins at Entry Interface (EI), which is the outer edge of the Earth's atmosphere. Throughout its descent trajectory, the capsule flies through the atmosphere with its blunt end forward, as illustrated in Figure 1-1. From EI, there are two basic types of paths along which the vehicle could travel to take it safely to the ground: a direct entry trajectory or a skip trajectory.

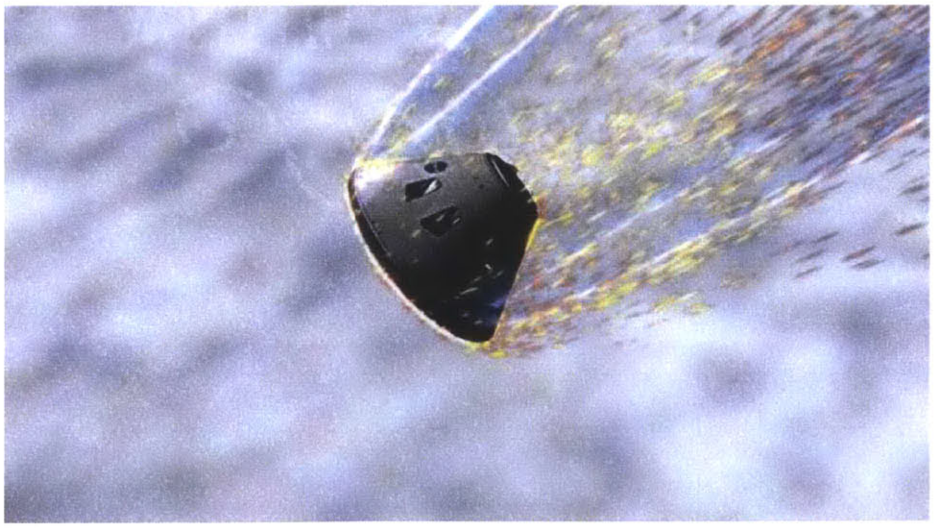


Figure 1-1: Atmospheric entry [13]

1.1.1 Direct Entry Trajectory

A *direct entry* trajectory is one in which the vehicle descends through the atmosphere from EI directly to the ground. It is characterized by continuously decreasing altitude and velocity, and it is typically used in situations where the range between Entry Interface and the desired landing site is relatively short. Due to the short downrange distance over which to reduce the vehicle's speed, direct entry trajectories typically impart higher deceleration loads and maximum heat rates on the vehicle.

1.1.2 Skip Trajectory

A *skip* trajectory is one in which the vehicle descends partway into the atmosphere from EI and decelerates slightly, then changes its vertical direction and starts to gain altitude, eventually leaving the atmosphere again. The vehicle then "skips" out of the atmosphere on a ballistic trajectory, travels some downrange distance, and reenters the atmosphere a second time, now heading directly to the ground on a trajectory with continuously decreasing altitude and velocity. Figure 1-2 shows an example of a skip trajectory, and Figure 1-3 shows the differences in basic trajectory shape between direct and skip entries.

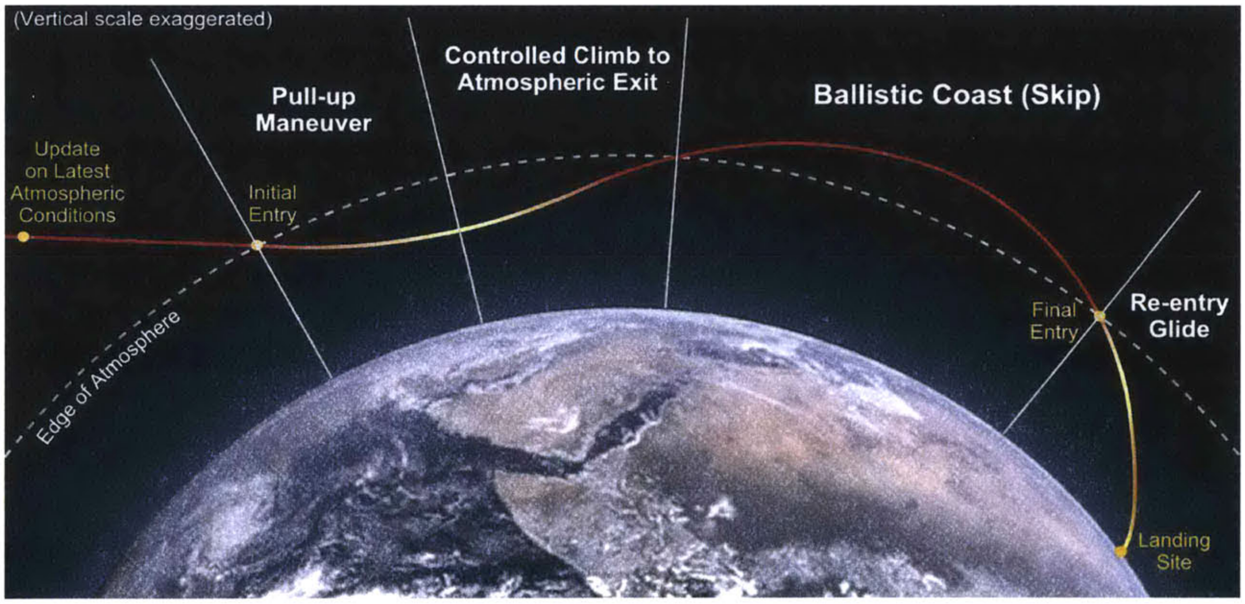


Figure 1-2: Skip entry trajectory [16]

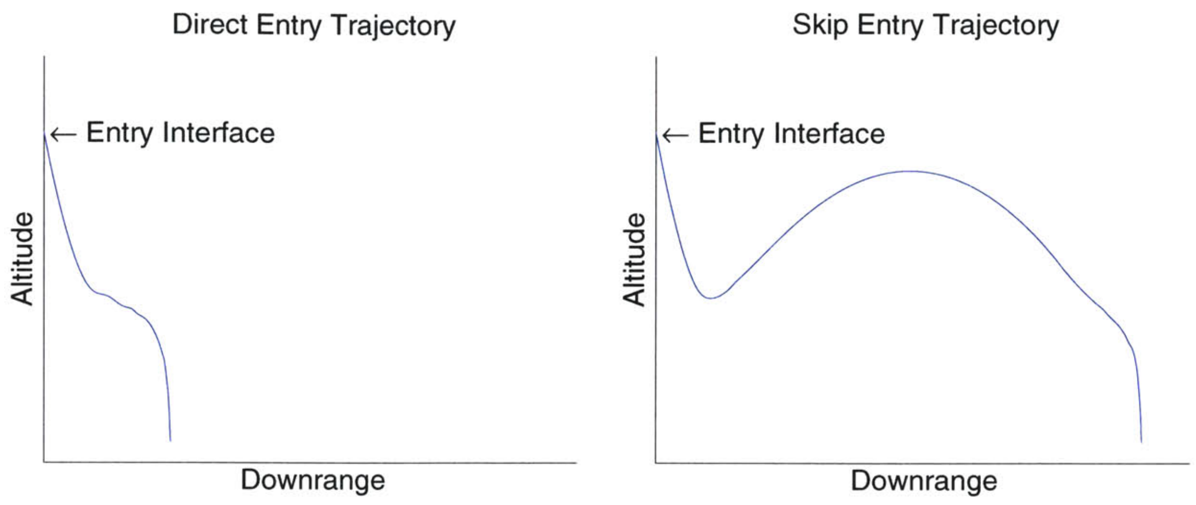


Figure 1-3: Direct and skip entry trajectories

Skip entry is most commonly required for situations in which the downrange distance between EI and the target landing site is relatively long. If the vehicle followed a direct entry trajectory in this case, it would lose too much energy flying through the atmosphere (due to aerodynamic drag) and would land short of the target. Flying a portion of the total distance outside of the atmosphere reduces the amount of energy lost to atmospheric drag, allowing the vehicle to cover a longer distance. Skip entry can also be used to fly over bad weather patterns between the Entry Interface point and the landing site, rather than through them. Skips usually impart lower deceleration loads on the vehicle than direct entries, but due to the longer amount of time required to complete the trajectory, the total heat load is typically higher. The extended entry time can be undesirable if the vehicle's crewmembers are sick or injured.

1.1.3 Bank Angle Modulation

A lifting body spacecraft, like an airplane, has several different aerodynamic surfaces through which to control or change its trajectory—for example: flaps, ailerons, elevator, and rudder. A capsule spacecraft, on the other hand, is much more limited. It typically has no aerodynamic surfaces and can only modify its trajectory by changing the orientation of its lift vector. Any body placed in an airflow experiences a drag force in the direction of the airflow, and if the body has a nonzero lift to drag ratio (L/D), then it also experiences a lift force perpendicular to the direction of the airflow. In this case, the nonzero L/D is a result of the offset center of gravity (CG) of the capsule. Figure 1-4 shows the aerodynamic forces on a capsule, where α is the angle of attack and θ is the angle of the acceleration vector with respect to the airflow.

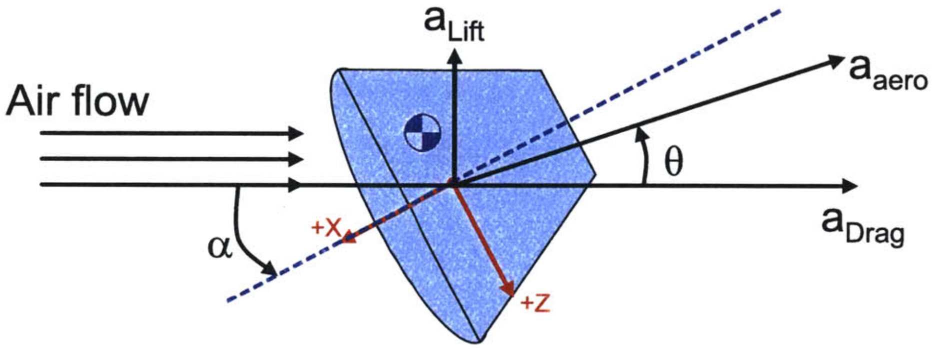


Figure 1-4: Vehicle geometry and aerodynamic forces (Credit: G. Barton & S. H. Bairstow)

While the lift vector will always be perpendicular to the airflow, the lift vector can rotate about the velocity vector, changing the angle it makes with the vertical. This puts a component of the lift vector in the lateral direction. This angle that the lift vector makes with the vertical is the *bank angle* (ϕ), and this is shown in Figure 1-5, looking at the nose of the capsule down to the heat shield. The bank angle is defined as the vehicle's amount of rotation about its velocity vector.

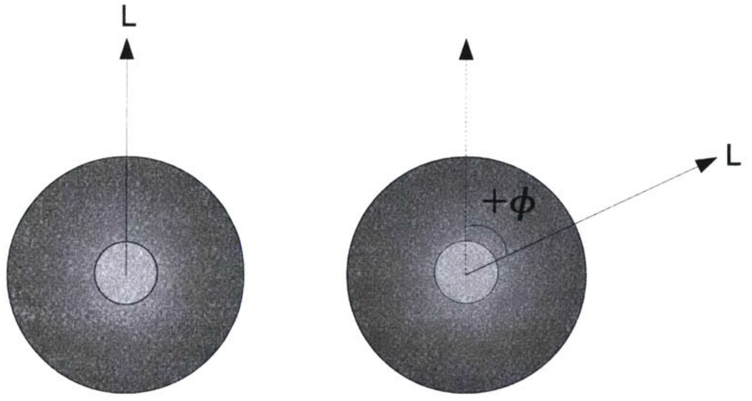


Figure 1-5: Lift vector and bank angle

During atmospheric reentry, the guidance system must steer laterally toward the target and manage energy so that the vehicle can reach the target landing site without overshooting it. These two tasks are accomplished by modulating the vehicle's bank angle. Flying *full lift up*, or bank angle of 0°, increases the vehicle's downrange capability. Flying *full lift down*, or bank angle of 180°, increases the drag on the vehicle, which increases the rate of energy decay and decreases the downrange capability. Flying *lift neutral*, or bank angle of 90°, provides an intermediate downrange capability but now places a component of lift in the lateral channel, allowing the vehicle to control its crossrange. This ability to manage energy while maintaining crossrange control can only be achieved through the orientation of the lift vector.

1.2 Motivation

The Charles Stark Draper Laboratory currently has a guidance algorithm to accomplish this type of atmospheric entry, both by direct and skip trajectories. It uses a bank-to-steer approach based on the original guidance logic for the Apollo program, but has been updated by Sarah Bairstow in several areas to make it more appropriate for the CEV capsule and to improve the landing accuracy for long range skips [2]. Bairstow's most significant change was the addition of a numeric predictor-corrector algorithm (PredGuid) that replaced the skip portion of the original Apollo code. The numeric predictor-corrector is used only during the skip portion of the trajectory to determine the necessary bank angle commands to reach the target landing site.

Although Bairstow's work greatly improved the Apollo algorithm, there are some shortcomings that still exist in the Apollo software which degrade landing accuracy for short-range targets and negatively affect robustness to "day of flight" uncertainties. Additionally, there are several logical transitions in the guidance algorithm that are based on empirical relationships for the Apollo capsule, making them inappropriate for any other vehicle. This underutilizes the full capability of the vehicle and creates logical paths that are never tested with the CEV configuration. The original Apollo code also contains many predictions based on simplified calculations and approximations. This was originally done to reduce the necessary computational power, but since modern spaceflight computers have much more computational capability than computers of the 1960's, these simplifications should be replaced with their more complex and accurate counterparts.

The early phases of the algorithm show some room for improvement in determining whether a skip is necessary or achievable, based on the current vehicle state and the desired range to go. This decision is made relatively late in the entry, and the algorithm would benefit from additional time to set up the desired trajectory if the decision were made earlier. Another concern is the energy management system. It uses a drag policy that is constant with time to deplete energy, but it sets the constant drag value at 130 ft/s^2 for every entry. With this approach, the only way to adjust the amount of energy depleted is by varying the amount of time spent flying the constant drag policy. This does not work well for situations in which a large amount of energy must be bled off over a short range.

Initially there were two versions of the Draper software, and the difference between them was the point at which guidance transferred control from the analytic Apollo algorithm to the numeric PredGuid skip algorithm. These two software versions shape the entry trajectory differently. This is most evident in the height of the skip, but the two versions both have other strengths and weaknesses. This implies that an optimal blend of the two may exist—a "best" point to switch to the numeric predictor-corrector.

1.3 Thesis Objective

The objective of this thesis is to identify shortcomings in the Draper PredGuid algorithm baseline and to propose energy management and phase transition enhancements to improve the algorithm's performance and robustness. This algorithm should fix the observed problems with the current Draper guidance, as well as reduce its dependency on Apollo-based heuristics and empirical constants. Overall, this algorithm should demonstrate improved robustness to uncertainties over its predecessors.

Specifically, this design will implement a variable constant drag policy, thereby allowing the algorithm to select the best constant drag level for the current energy and remaining range. The decision to perform a skip or direct entry will be made early in the algorithm, and the decision should incorporate as much of the current vehicle capability as possible.

Additionally, the reference trajectory used for the final descent phase will be redesigned for direct entry situations. Finally, this algorithm will determine the best time to transition to the numeric predictor-corrector and begin the skip. This decision will be specific to each individual entry.

This algorithm will be developed and tested against the Draper PredGuid baseline algorithm to ensure that the changes improve overall robustness. The CEV design concept will be used as the entry vehicle, but the algorithm should be generic enough to accommodate other vehicle configurations. The resulting algorithm should provide robust precision landing capability for target ranges and dispersions that are within the allowable limits.

1.4 Thesis Overview

This chapter provides an introduction to and motivation for the problem under consideration, as well as key concepts that will be used in the following chapters. Chapter 2 describes the simulation environment in which the guidance algorithms are developed and analyzed, while Chapter 3 summarizes the various metrics for evaluating the performance of the guidance algorithms. The guidance algorithm enhancements developed for this thesis are based on two predecessors: the original Apollo entry guidance (covered in Chapter 4), and the Draper PredGuid improvements to the skip logic (discussed in Chapter 5). Chapter 6 details the current problems in the Draper baseline, and Chapter 7 presents proposed algorithm enhancements developed for this thesis and describes how it addresses the shortcomings in the baseline logic. The resulting combination of the Draper baseline and these enhancements is referred to as the Predictor algorithm, and results from a “fly off” between the Draper baseline and the Predictor algorithm are shown and discussed in Chapter 8. Finally, Chapter 9 contains the conclusions that can be drawn from this work, as well as a brief list of items for future analysis.

Chapter 2

Simulation Environment

Computer simulations are used to thoroughly test a guidance algorithm for Earth atmospheric entry before performing any expensive and time-consuming flight tests. The results of the simulations vary in accuracy, depending on the complexity and fidelity of the simulation environment. Together, Matlab and Simulink provide the simulation environment used for the development and testing of these guidance algorithms. Specifically, Matlab version 7.2.0.232 (R2006a) and Simulink version 6.4 are used. This simulation assumes that vehicle maintains constant mass throughout reentry—i.e. no mass is lost due to fuel use or heat shield ablation. In addition, the simulation assumes perfect and continuous navigation, so the vehicle has accurate knowledge of its state throughout the entry.

This chapter gives an overview of the simulation used for this thesis, which is Draper Laboratory's Entry, Descent, and Landing (EDL) Simulation, version 2. Details of the mathematical equations used in the simulation are provided, as well as the coordinate frames, vehicle and environment models, and other relevant conventions.

2.1 Reference Coordinate Frames

A coordinate frame establishes a basis for defining vector quantities. Depending on the situation, one particular reference frame may be more appropriate than another, often because it gives more insight into the motion of the spacecraft or it simplifies the mathematical equations. This section describes the three major coordinate frames used in the simulation.

Inertial Frame

The inertial reference frame ($\hat{\mathbf{i}}_I$, $\hat{\mathbf{j}}_I$, $\hat{\mathbf{k}}_I$) is a non-rotating, Earth-centered coordinate system. The $\hat{\mathbf{k}}_I$ axis is aligned with the Earth's North Pole. The $\hat{\mathbf{i}}_I$ axis points through the intersection of the equator and the Prime Meridian at time $t = 0$, which is the start of reentry. The $\hat{\mathbf{j}}_I$ axis is perpendicular to both the $\hat{\mathbf{i}}_I$ and the $\hat{\mathbf{k}}_I$ axis, completing the right-handed coordinate system. This frame is shown at $t = 0$ in Figure 2-1.

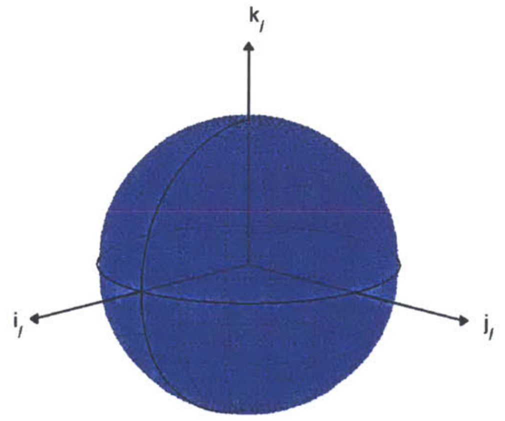


Figure 2-1: Inertial coordinate frame at time $t = 0$

Body Frame

The body reference frame ($\hat{\mathbf{i}}_b$, $\hat{\mathbf{j}}_b$, $\hat{\mathbf{k}}_b$) is a spacecraft-centered coordinate system that remains fixed to the vehicle. The origin of this coordinate frame is the vehicle's center of gravity (CG). The $\hat{\mathbf{i}}_b$ axis runs along the centerline of the capsule and points out the heat shield. Due to the axisymmetric shape of the capsule, the $\hat{\mathbf{j}}_b$ and $\hat{\mathbf{k}}_b$ axes are defined by the internal layout. The $\hat{\mathbf{k}}_b$ axis is positive in the "foot to head" direction of a seated crewmember; the $\hat{\mathbf{j}}_b$ axis completes the right-handed coordinate system.

Stability Frame

The stability reference frame ($\hat{\mathbf{i}}_s$, $\hat{\mathbf{j}}_s$, $\hat{\mathbf{k}}_s$) is also a spacecraft-centered coordinate system that is centered at the capsule's CG; however, this frame is used to measure the aerodynamic forces and torques on the vehicle. The $\hat{\mathbf{i}}_s$ axis points along the air-relative velocity vector (opposite the drag vector), the $\hat{\mathbf{k}}_s$ axis is aligned with the lift vector, and the $\hat{\mathbf{j}}_s$ completes the right-handed coordinate system. Figure 2-2 shows the alignment of body frame and stability frame when the sideslip angle (β) is zero.

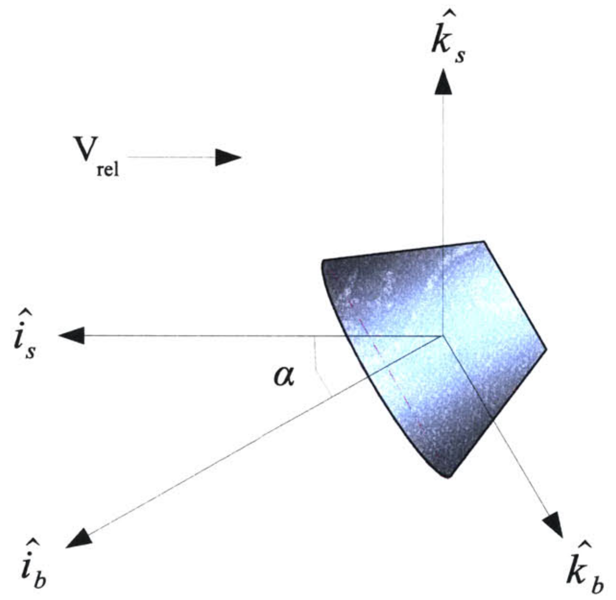


Figure 2-2: Body reference frame (b) and stability reference frame (s), with $\beta = 0$

2.2 Coordinate Transformations

Although multiple coordinate frames often allow for a more intuitive understanding of the spacecraft's motion, the equations describing the laws of physics are often only valid under non-accelerating and non-rotating frames of reference. To convert a mathematical description of the spacecraft dynamics from one frame to another, it is necessary to use coordinate transformations. These transformations, through a combination of translation, rotation, and scaling, convert a vector from one frame to another by multiplying the vector by a transformation matrix, $\mathcal{T}$:

$$\mathbf{z}_b = \mathcal{T}_a^b \mathbf{z}_a$$

where the subscript and superscript denote a transformation from frame a to frame b.

The reverse transformation can also be performed; however, instead of using the inverse of $\mathcal{T}_a^b$ in the matrix equation, the transpose can be used instead. This is because the transformation matrices are orthogonal. Therefore, the following relationship holds:

$$\mathcal{T}_b^a = \left(\mathcal{T}_a^b\right)^{-1} = \left(\mathcal{T}_a^b\right)^T$$

Furthermore, a vector can be converted from one reference frame to another through an intermediate reference frame, as demonstrated by the following equation:

$$\mathbf{z}_b = \mathcal{T}_m^b \mathcal{T}_a^m \mathbf{z}_a$$

However, these matrix multiplications are not commutative: $\mathcal{T}_m^b \mathcal{T}_a^m \neq \mathcal{T}_a^m \mathcal{T}_m^b$.

The coordinate transformations used in this simulation are described in the remainder of this section.

Stability Frame to Body Frame

These two frames are related by the angle of attack (α) and the sideslip angle (β). The transformation from the body frame to the stability frame is given by:

$$\mathcal{T}_b^s = \begin{bmatrix} -\sin\alpha & 0 & \cos\alpha \\ \cos\alpha\cos\beta & -\sin\beta & \sin\alpha\cos\beta \\ \cos\alpha\sin\beta & \cos\beta & \sin\alpha\sin\beta \end{bmatrix} \quad (2.1)$$

Body Frame to Inertial Frame

The body frame can be related to the inertial coordinate frame using a series of transformations and some intermediate reference frames: the Local Vertical, Local Horizontal (LVLH) frame and the relative velocity frame. These two spacecraft-centered frames are used purely as intermediate steps and will not be described here.

The transformation from the inertial frame to the LVLH frame can be made using the inertial position and velocity vectors:

$$\hat{\mathbf{c}}_3 = -\hat{\mathbf{r}}_I \tag{2.2}$$

$$\hat{\mathbf{c}}_2 = -\hat{\mathbf{r}}_I \times \hat{\mathbf{v}}_I \tag{2.3}$$

$$\hat{\mathbf{c}}_1 = (-\hat{\mathbf{r}}_I \times \hat{\mathbf{v}}_I) \times -\hat{\mathbf{r}}_I \tag{2.4}$$

where $\hat{\mathbf{c}}_1$, $\hat{\mathbf{c}}_2$, and $\hat{\mathbf{c}}_3$ are all normalized to be unit vectors. The transformation matrix is then composed of these three vectors as the columns of the matrix:

$$\mathcal{T}^{I}_{LVLH} = \begin{bmatrix} \uparrow & \uparrow & \uparrow \\ \hat{\mathbf{c}}_1 & \hat{\mathbf{c}}_2 & \hat{\mathbf{c}}_3 \\ \downarrow & \downarrow & \downarrow \end{bmatrix} \tag{2.5}$$

The transformation from the LVLH frame to the relative velocity frame can be accomplished by rotating the LVLH frame downward by the flight path angle, γ, which is the angle between the local horizontal and the air-relative velocity vector.

$$\mathcal{T}^{V}_{LVLH} = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix} \tag{2.6}$$

The reverse transformation, from the velocity frame to the LVLH frame, is given by the transpose of the previous matrix:

$$\mathcal{T}^{LVLH}_{V} = \left(\mathcal{T}^{V}_{LVLH}\right)^{T} \tag{2.7}$$

The transformation from the body frame to the velocity frame requires the following three rotation matrices, where ξ is the roll angle, η is the pitch angle, and ψ is the yaw angle.

$$\mathbf{R}_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\xi & -\sin\xi \\ 0 & \sin\xi & \cos\xi \end{bmatrix} \tag{2.8}$$

$$\mathbf{R}_p = \begin{bmatrix} \cos\eta & 0 & \sin\eta \\ 0 & 1 & 0 \\ -\sin\eta & 0 & \cos\eta \end{bmatrix} \tag{2.9}$$

$$\mathbf{R}_y = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

The body-to-velocity transformation is then given by the following equation:

$$\mathcal{T}_b^V = \mathbf{R}_r \mathbf{R}_p \mathbf{R}_y \tag{2.11}$$

Finally, the transformation from the body frame to the inertial frame can be written as the product of the following transformations:

$$\mathcal{T}_b^I = \mathcal{T}_{LVLH}^I \mathcal{T}_V^{LVLH} \mathcal{T}_b^V \tag{2.12}$$

2.3 Environment Models

Models of the Earth's environment are used to approximate the natural forces and torques on the vehicle during atmospheric flight. The ones described here are models of the Earth's atmosphere and gravity field.

2.3.1 Earth Gravity Model

The Earth's gravity field can be modeled in different ways, depending on the desired level of accuracy. For this simulation, the Earth gravity model includes the simple spherical relationship from Newton's Second Law plus the effects of the J_2 harmonic. Equation 2.13 is the acceleration due to gravity, written in the inertial frame.

$$\mathbf{a}_{g_I} = -\frac{\mu}{r^3}\mathbf{r}_I - \frac{3J_2\mu {R_E}^2}{2r^5}\left(1 - \frac{5{r_j}^2}{r^2}\right)\mathbf{r}_I - \frac{3J_2\mu {R_E}^2}{r^5} r_k \hat{\mathbf{k}}_I \tag{2.13}$$

where:

$$r = |\mathbf{r}_I|$$

2.3.2 Earth Atmosphere

This simulation uses the 1962 U.S. Standard Atmosphere to model the Earth's atmosphere [17]. The model provides the air density, speed of sound, and temperature based on the vehicle's inertial position and velocity. The air density, ρ, is important in determining the aerodynamic lift and drag forces experienced by the vehicle throughout the entry. The atmosphere is assumed to be fixed with the Earth, so it is not stationary in the inertial frame. Winds are not modeled in this environment.

2.4 Vehicle Model

The spacecraft model used for this thesis is a conceptual design for the Crew Exploration Vehicle, documented in Reference [6]. The spacecraft is a blunt body capsule that is a slightly larger, scaled version of the Apollo command module. The basic capsule shape is shown in Figure 2-3. The maximum diameter of the capsule is 16.5 ft at the heat shield, and the sidewall is inclined 32.5° from the centerline. The total length of the capsule from nose to heat shield is 10.83 ft. The reference area for aerodynamic calculations is approximately 213.8 ft^2, calculated by $S_{ref} = \pi r^2$, where r is the radius of the heat shield. The reference length, L_{ref}, is also needed for aerodynamic calculations, and it is the maximum diameter of the capsule. The total mass used for this thesis, scaled from the Apollo configuration, is $17,000\ lb_m$.

Aerodynamic properties of the capsule were provided by NASA [6] and are also taken from *Reentry Guidance with Extended Range Capability for Low L/D Spacecraft* [2]. This

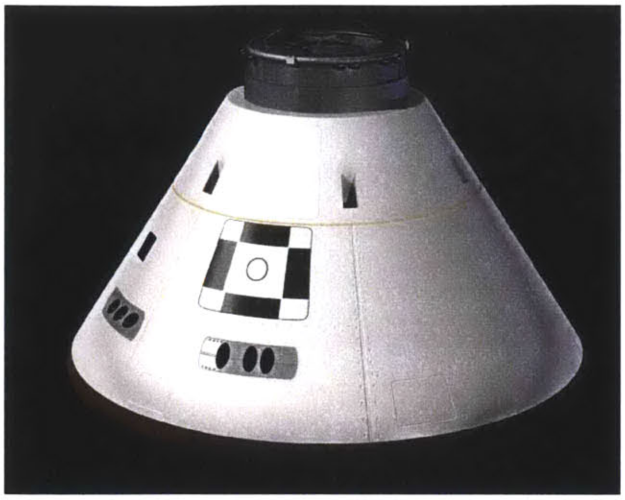


Figure 2-3: CEV concept's capsule shape [13]

includes moment data and the coefficients of lift and drag as functions of Mach number and angle of attack (α). The trim conditions, however, are not specified by NASA. The trim angle of attack (α_{trim}) is the angle at which sum of the moments about the vehicle's center of gravity is zero; in other words, there are no unbalanced forces that cause a change in the vehicle's pitch angle. α_{trim} is determined by the location of the center of gravity, and it changes with Mach number. The trim values of α, C_D, C_L, and L/D used in this thesis are plotted in Figure 2-4. This data is stored as a lookup table for use during the simulation. For Mach values that fall between data points, linear interpolation is used to determine the angle of attack and the aerodynamic coefficients.

The guidance algorithm requires constant values for C_D and C_L to use for its calculations and predictions. Since the vehicle is hypersonic for the majority of the entry, the constant values chosen for guidance are averages at high Mach values. For comparison, this L/D ratio of 0.36 is approximately 20% higher than the Apollo command module's L/D ratio at hypersonic speeds [5].

$$\begin{aligned} C_D &= 1.20 \\ C_L &= 0.43 \\ L/D &= 0.36 \end{aligned}$$

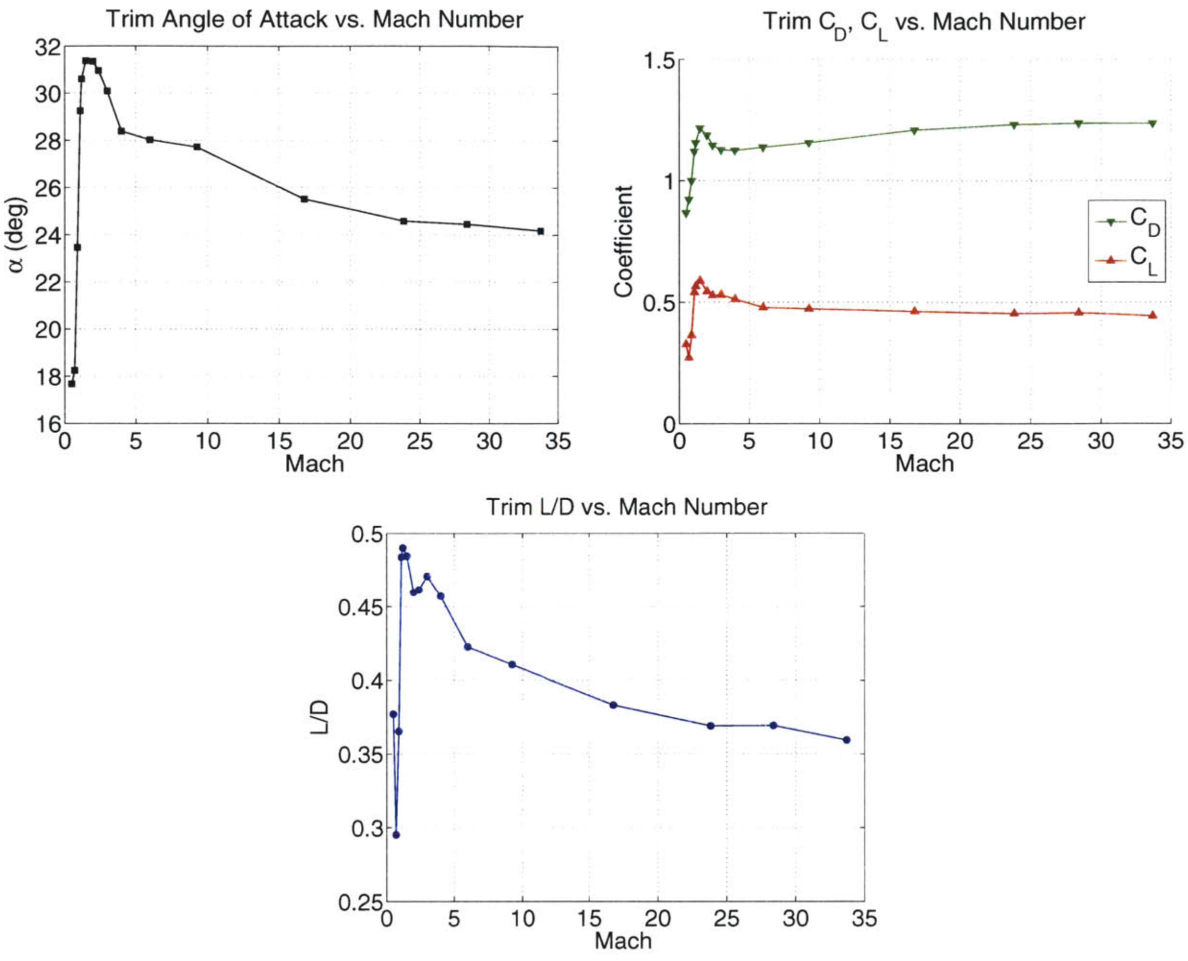


Figure 2-4: Aerodynamic data at trim angle of attack

2.5 Equations of Motion

The numeric simulation used in this thesis allows six degrees of freedom (DOF). The translational motion is governed by the total force on the vehicle, and the rotational motion is determined by the total torque on the spacecraft.

2.5.1 Translational Motion

The total force acting on the vehicle during its descent through the atmosphere is the sum of the gravity force, the aerodynamic forces, and the forces due to the Reaction Control System (RCS) engines.

$$\mathbf{F}_{total} = \mathbf{F}_{gravity} + \mathbf{F}_{aero} + \mathbf{F}_{RCS} \tag{2.14}$$

The vehicle's total acceleration is then related to the total force by Newton's Second Law, where m represents the total mass of the spacecraft:

$$\mathbf{F}_{total} = m\mathbf{a}_{total} \tag{2.15}$$

The gravity force, written in the inertial frame, is simply

$$\mathbf{F}_{grav_I} = m\mathbf{a}_{grav_I} \tag{2.16}$$

where $\mathbf{a}_{grav_I}$ is given by Equation 2.13.

In the stability frame, the aerodynamic force has two components: drag and lift (the lateral force is negligible). These forces are calculated as the product of the dynamic pressure, the reference area of the vehicle S_{ref}, and their aerodynamic coefficients. Dynamic pressure, q, is calculated from current flight conditions, using the air density ρ (determined from the atmosphere model) and the air-relative velocity magnitude, V_{rel}:

$$q = \frac{1}{2}\rho {V_{rel}}^2 \tag{2.17}$$

The aerodynamic coefficients of each of these forces are determined by the vehicle's total angle of attack and Mach number, and the total force is calculated using Equation 2.18.

$$\mathbf{F}_{aero_s} = qS_{ref}\begin{bmatrix} C_D \\ C_L \\ 0 \end{bmatrix} \tag{2.18}$$

The aerodynamic force is then transformed into the inertial frame through a series of coordinate transformations.

$$\mathbf{F}_{aero_I} = T_b^I T_s^b \mathbf{F}_{aero_s} \tag{2.19}$$

The Reaction Control System jet forces are calculated using Equation 2.20, written in the body frame:

$$\mathbf{F}_{RCS_b} = [\text{thrust directions}]_{3\times N}\,[\text{throttle cmds}]_{N\times 1}\,(\text{thrust/engine}) \tag{2.20}$$

The RCS force must also be transformed into the inertial frame:

$$\mathbf{F}_{RCS_I} = T_b^I \mathbf{F}_{RCS_b} \tag{2.21}$$

The total inertial acceleration can now be written as a sum of the individual components:

$$\mathbf{a}_{total_I} = \frac{1}{m}\left(\mathbf{F}_{grav_I} + \mathbf{F}_{aero_I} + \mathbf{F}_{RCS_I}\right) \tag{2.22}$$

This equation for inertial acceleration can be integrated to determine the inertial velocity:

$$\mathbf{v}_I = \int \mathbf{a}_I dt \tag{2.23}$$

Inertial velocity can then be integrated to determine inertial position:

$$\mathbf{r}_I = \int \mathbf{v}_I dt \tag{2.24}$$

The vehicle's initial velocity and position are the constants of integration for Equations 2.23 and 2.24.

2.5.2 Rotational Motion

Torques, or moments, on the vehicle cause rotational motion, and the total torque has the same three basic components as the total force:

$$\boldsymbol{\tau}_{total} = \boldsymbol{\tau}_{grav} + \boldsymbol{\tau}_{aero} + \boldsymbol{\tau}_{RCS} \tag{2.25}$$

The gravity gradient torque, in body coordinates, is given by Equation 2.26:

$$\boldsymbol{\tau}_{gg_b} = \frac{3\mu\left(\hat{\mathbf{r}}_b \times \mathbf{K}\hat{\mathbf{r}}_b\right)}{|\mathbf{r}_b|^3} \tag{2.26}$$

where:

$\boldsymbol{\tau}_{gg_b}$	=	gravity gradient torque vector in body coordinates
μ	=	standard gravitational parameter for Earth, GM_E
$\mathbf{r}_b$	=	position vector in body coordinates
$\hat{\mathbf{r}}_b$	=	unit position vector in body coordinates
$\mathbf{K}$	=	vehicle inertia matrix

The aerodynamic torque is divided into three parts. The first part comes from the nonzero pitching moment coefficient. Because this torque is calculated in the stability frame, the roll and yaw moment coefficients are zero.

$$\boldsymbol{\tau}_{CM_s} = qS_{ref}L_{ref}\begin{bmatrix} 0 \\ 0 \\ C_M \end{bmatrix} \tag{2.27}$$

The second part of the aerodynamic torque comes from the Euler angle rates, $[P\ Q\ R]$. This is calculated in the body frame using Equation 2.28.

$$\boldsymbol{\tau}_{PQR_b} = \frac{qS_{ref}{L_{ref}}^2}{V_{rel}}\begin{bmatrix} P \cdot C_{m_P} \\ Q \cdot C_{m_Q} \\ R \cdot C_{m_R} \end{bmatrix} \tag{2.28}$$

The last part of the aerodynamic torque comes from the offset center of gravity (CG) moment. This is simply the cross product of pivot arm and total aerodynamic force:

$$\boldsymbol{\tau}_{CG_b} = \mathbf{r}_{pos} \times \mathbf{F}_{aero_b} \tag{2.29}$$

where $\mathbf{r}_{pos}$ is the vector difference between the pivot location and the common moment reference location. The aerodynamic torque can now be written as the sum of the previous three equations:

$$\boldsymbol{\tau}_{aero_b} = \mathcal{T}_s^b \boldsymbol{\tau}_{C_{M\,s}} + \boldsymbol{\tau}_{PQR_b} + \boldsymbol{\tau}_{CG_b} \tag{2.30}$$

The RCS torque is similar to the RCS force:

$$\boldsymbol{\tau}_{RCS_b} = [\text{unit torque matrix}]_{3\times N}\,[\text{throttle cmds}]_{N\times 1}\,(\text{thrust/engine}) \tag{2.31}$$

The total angular acceleration, also in body coordinates, can now be written using Newton's Second Law for rotation:

$$\boldsymbol{\alpha}_{total_b} = \mathbf{K}^{-1}\boldsymbol{\tau}_{total_b} \tag{2.32}$$

Angular acceleration can be integrated to get angular rate:

$$\boldsymbol{\omega}_b = \int \boldsymbol{\alpha}_b \tag{2.33}$$

Angular rate can then be integrated to get the angular position:

$$\boldsymbol{\Theta}_b = \int \boldsymbol{\omega}_b \tag{2.34}$$

Again, the constants of integration for Equations 2.33 and 2.34 are the vehicle's initial angular rate and attitude.

2.5.3 Bank Angle Definition

The bank angle convention for this thesis is described in Section 1.1.3. Specifically, this simulation environment defines the bank angle ϕ as the angle between the following two planes:

1. The vertical plane defined by the inertial position vector, $\mathbf{r}_I$, and the air-relative velocity vector, $\mathbf{V}_{rel}$
2. The plane that contains the longitudinal axis of the vehicle, $\hat{\mathbf{i}}_b$, and the air-relative velocity vector, $\mathbf{V}_{rel}$

The bank angle is then computed as the angle between the unit vectors normal to each of those planes, $\hat{\mathbf{n}}_1$ and $\hat{\mathbf{n}}_2$, using the following method:

$$\hat{\mathbf{n}}_1 = \text{unit}\left(\mathbf{r}_I \times \mathbf{V}_{rel}\right) \tag{2.35}$$

$$\hat{\mathbf{n}}_2 = \text{unit}\left(\mathbf{V}_{rel} \times \hat{\mathbf{i}}_b\right) \tag{2.36}$$

$$|\phi| = \text{acos}\left(\hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_2\right) \tag{2.37}$$

$$\text{sign}\left(\phi\right) = \text{sign}\left[\left(\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2\right) \cdot \mathbf{V}_{rel}\right] \tag{2.38}$$

Using this definition, the bank angle is zero when the two planes are parallel, and the bank angle is positive (clockwise) when the vehicle's roll component is positive. In other words, from a vantage point that is behind the vehicle during reentry, a positive bank angle points the lift vector to the right.

The desired bank angle is achieved through coordinated roll/yaw control; this attitude maneuver is performed using the RCS jets. The flight control system used in this simulation to transform bank angle commands into RCS jet commands is documented in "MIMO Adaptive Bank-To-Steer Control Algorithms for Guided Re-Entry Vehicles." [8]

2.5.4 Planar Equations of Motion for a Point Mass

The equations in this section are not used in the simulation environment, but they are used to derive certain control laws and relationships used in the entry guidance code. These equations describe translational motion of a point mass in two dimensions, altitude and range, using the coordinate system shown in Figure 2-5. These are simplifications of the 6-DOF equations used in the simulation environment. The complete derivation of these equations can be found in Reference [10].

To start, Newton's Second Law can be written in component form as:

$$F_{Lift} - mg\cos\gamma = ma_n \tag{2.39}$$

$$-mg\sin\gamma - F_{Drag} = ma_t \tag{2.40}$$

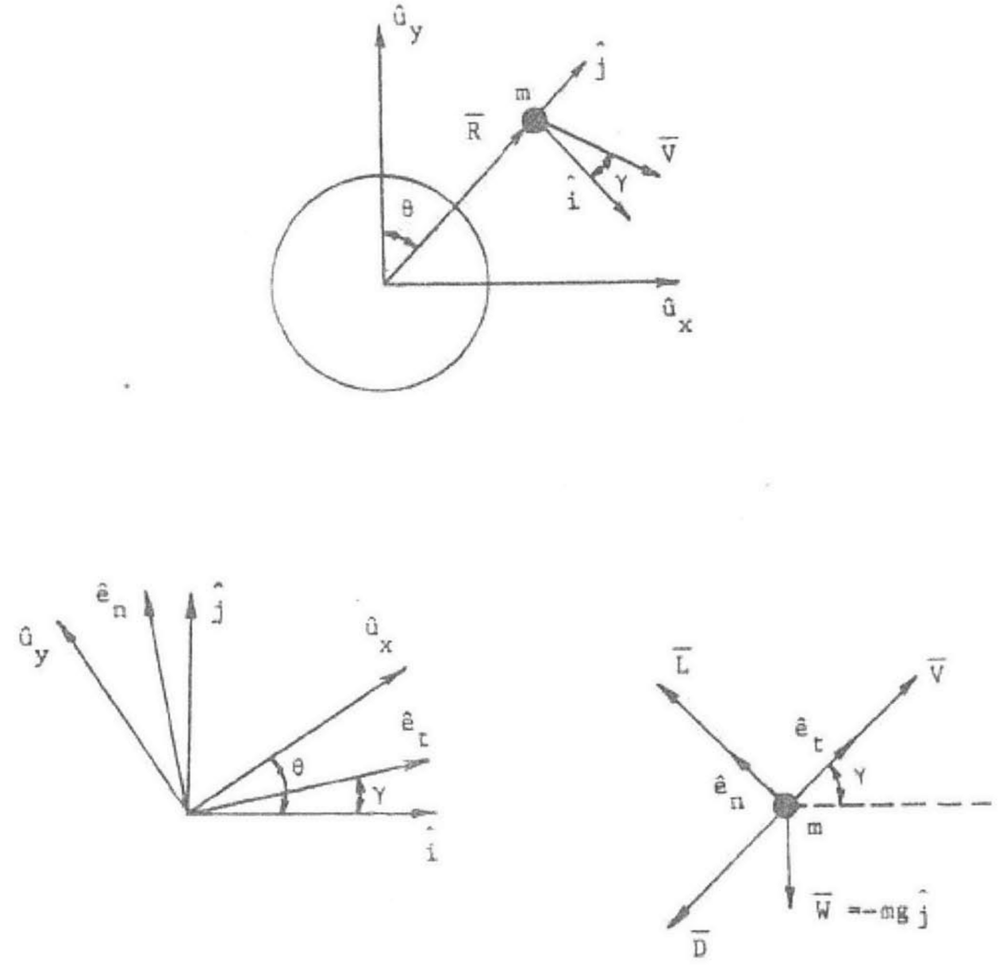


Figure 2-5: Coordinate system for planar, point mass equations

The kinematic relations are:

$$a_t = \dot{V} \tag{2.41}$$

$$a_n = V\left(\dot{\gamma} + \dot{\theta}\right) \tag{2.42}$$

$$V\sin\gamma = \dot{R} \tag{2.43}$$

$$V\cos\gamma = -\dot{\theta}R \tag{2.44}$$

Combining the dynamic and kinematic equations yields the non-linear set of equations:

$$\dot{V} = -g \sin\gamma - \frac{F_{Drag}}{m} \tag{2.45}$$

$$V\dot{\gamma} = g \cos\gamma \left[\frac{V^2}{gR} - 1\right] + \frac{F_{Lift}}{m} \tag{2.46}$$

$$\dot{R} = V \sin\gamma \tag{2.47}$$

To simplify these equations to a linear form, first define lift and drag accelerations as forces per unit mass:

$$D = \frac{F_{Drag}}{m} \tag{2.48}$$

$$L = \frac{F_{Lift}}{m} \tag{2.49}$$

Assume shallow flight path angles to eliminate trigonometric functions:

$$\sin\gamma \approx \gamma \tag{2.50}$$

$$\cos\gamma \approx 1 \tag{2.51}$$

Also assume that the drag acceleration is much greater than the product of gravity and the flight path angle:

$$D \gg g \sin\gamma \tag{2.52}$$

Combining the assumptions made in Equations 2.48–2.52, the nonlinear system of equations can be reduced to its simplified, linear form:

$$\dot{V} = -D \tag{2.53}$$

$$V\dot{\gamma} = L + g\left[\frac{V^2}{gR} - 1\right] \tag{2.54}$$

$$\dot{R} = V\gamma \tag{2.55}$$

where $\dot{R}$ is equivalent to altitude rate, $\dot{h}$.

2.6 Aerodynamic Heating

Aerodynamic heating is calculated in this simulation environment as the sum of convective and radiative heating. Convective heating is calculated using the Chapman equation [3] for heating rate on a reference sphere:

$$q_c = 17600\, {r_n}^{-0.5} \left(\frac{\rho}{\rho_0}\right)^{0.5} \left(\frac{V_{rel}}{V_{cir}}\right)^{3.15} \tag{2.56}$$

where:

q_c	$=$	convective heat rate $\left[\frac{BTU}{ft^2 \cdot s}\right]$
r_n	$=$	vehicle effective nose radius $[ft]$
ρ_0	$=$	air density at zero altitude
V_{rel}	$=$	vehicle's Earth-relative velocity
V_{cir}	$=$	circular orbital velocity, $\sqrt{g\,(R_E + h)}$

Radiative aeroheating is given by the Tauber-Sutton relation [15]:

$$q_r = C{r_n}^a \rho^b f\,(V_{rel}) \tag{2.57}$$

where:

q_r	$=$	radiative heat rate $\left[\frac{W}{cm^2}\right]$
C	$=$	4.736×10^4
r_n	$=$	vehicle effective nose radius $[m]$
a	$=$	$\begin{cases} 0.6, & r_n \leq 2 \\ 0.5, & r_n > 2 \end{cases}$
b	$=$	1.22

and $f\,(V_{rel})$ is tabulated for the Earth's atmosphere in Reference [15].

The total heat rate is the sum of q_c and q_r. These empirical relationships approximate stagnation point heating only; they do not include any effect of heat soak or irradiation. Heat load is simply the heat rate integrated over time.

2.7 Simulation Initialization

The vehicle's simulated trajectory is a result of its initial conditions and the bank angle commands generated throughout the trajectory to steer to the desired landing site. The spacecraft is assumed to be on a return trajectory from the Moon, rather than the International Space Station or other Low Earth Orbit (LEO) location. Since the CEV is used for the test cases presented in this thesis, the initial conditions for entry are derived from a CEV lunar return mission profile.

2.7.1 Vehicle Initial Conditions

The spacecraft's initial conditions at Entry Interface (EI) are described by altitude, velocity, and flight path angle:

$$\begin{aligned} \gamma &= -5.9 \quad deg \\ h &= 400,000 \quad ft \\ V &= 35,833 \quad ft/s \end{aligned}$$

The downrange distance that the vehicle travels from its initial EI position to the landing site is referred to as the *target range*. To yield a particular target range, this simulation adjusts the EI location backward from the landing site, rather than fixing the EI location and moving the landing site.

2.7.2 Target Location

For this thesis, the location of the target landing site is Edwards Air Force Base. Its location is fully described by its geodetic latitude, longitude, and azimuth:

$$\begin{aligned} \text{Latitude} &= 34.99^\circ \\ \text{Longitude} &= -117.85^\circ \\ \text{Azimuth} &= 45^\circ \end{aligned}$$

2.8 Simulation Termination Conditions

The simulation terminates under either one of two conditions. The first is if the vehicle's altitude exceeds a certain limit, indicating that it has skipped away from the Earth. The altitude limit for this condition is $250,000\ m$, or approximately $820,000\ ft$.

The more ideal termination condition is when the vehicle reaches the ground. Ideally, this would be exactly at zero altitude, but this simulation does not run all the way to the ground. In a typical entry scenario, the vehicle deploys parachutes once it reaches a low enough altitude and velocity and drifts the rest of the way to the ground. To accurately determine the true impact point, sophisticated models of the parachutes are needed to determine the drift characteristics. Those models are outside the scope of this work, so the simulation ends when the vehicle descends to an altitude of $25,000\ ft$, which is the approximate altitude at which the drogue parachute would be deployed. The landing point is taken to be this simulation termination location, projected onto the ground.

2.9 Monte Carlo Dispersed Parameters

Monte Carlo simulations are unlike nominal simulations because they include nondeterministic components. The Monte Carlo method uses random numbers to approximate typical day-of-flight conditions in which certain elements are not known exactly—small changes in the atmospheric entry position and attitude, or slight variations in the atmospheric density, for example. For simulation purposes, these parameters are allowed to vary by up to three standard deviations from the mean or nominal value. The dispersion parameters used in the Monte Carlo simulations for this thesis are listed in Table 2.1. In addition, the guidance algorithm uses the 1976 U.S. Standard Atmosphere in Monte Carlo simulations so that it does not have perfect knowledge of the entry environment, which is the 1962 U.S. Standard Atmosphere.

Table 2.1: 3σ Uncertainties and Dispersions for Monte Carlo Analysis

Parameter	***Mean***	***3σ dispersion***
Entry velocity	$35,833\ ft/s$	$500\ ft/s$
Entry velocity azimuth	$15° - 75°$	$0.1°$
Entry flight path angle	$-5.9°$	$0.1°$
Entry latitude	35°S – 32°N	$0.01°$
Entry longitude	$< 180°$	$0.01°$
Atmospheric density scale factor	35%	10 – 60%
C_L aerodynamic coefficient factor	5% (Mach > 6) 15% (Mach < 4)	5% (Mach > 10) 10% (Mach < 5)
C_D aerodynamic coefficient factor	5% (Mach > 6) 15% (Mach < 4)	3% (Mach > 10) 8% (Mach < 5)
C_M aerodynamic coefficient bias	-	0.007
RCS thrust (individual jets)	-	3%
Trim α knowledge	-	$3°$
Vehicle mass	$17,068\ lb_m$	5%
Moments of inertia	-	5%
Center of gravity offset	$-9.4\ in$	$0.5\ in$

Chapter 3

Guidance Algorithm Metrics

Several different metrics, or figures of merit, are used to evaluate the performance of the entry guidance algorithms considered here: the Draper baseline and the Predictor algorithm, which is a combination of the Draper PredGuid skip guidance and the enhanced energy management and transition logic developed for this thesis. The performance based metrics are used to determine how well the algorithm met a particular goal. These are:

1. Landing accuracy
2. g load
3. Aerodynamic heating

The margin based metrics are used to assess how far away the algorithm is from the edge of its capability, and they are the following:

1. Control saturation
2. Final phase energy bucket

All of these metrics are nearly identical to the ones found in Reference [2].

3.1 Performance Based Metrics

Since this algorithm is designed for precision landing, one important performance metric is landing accuracy—how well the algorithm can guide the vehicle to the desired ground landing site. Another very important concern is the deceleration force on the vehicle. Excessive g loads, especially over long durations, are very stressful on humans; therefore, the load must be kept below certain levels. The third metric is aerodynamic heating. The friction created by a blunt body flying through a dense atmosphere at lunar return velocities creates an enormous amount of heat. To survive this environment, reentry spacecraft have a thermal protection system covering the outside of the structure, but this protective covering has a limit to the amount of heat it can absorb or dissipate. Past this point, the heat starts to endanger the crew, damage the vehicle, and can ultimately destroy it.

3.1.1 Landing Accuracy

The requirement for landing accuracy is 2.3 nautical miles (nm), or 4.3 km [2]. In order for a particular trial to meet the requirement, its total distance from the target landing site must not exceed this value. A simple way to view this is in a downrange and crossrange error scatter plot, where the trials inside the circle are the ones that meet the requirement. A sample of this is shown in Figure 3-1.

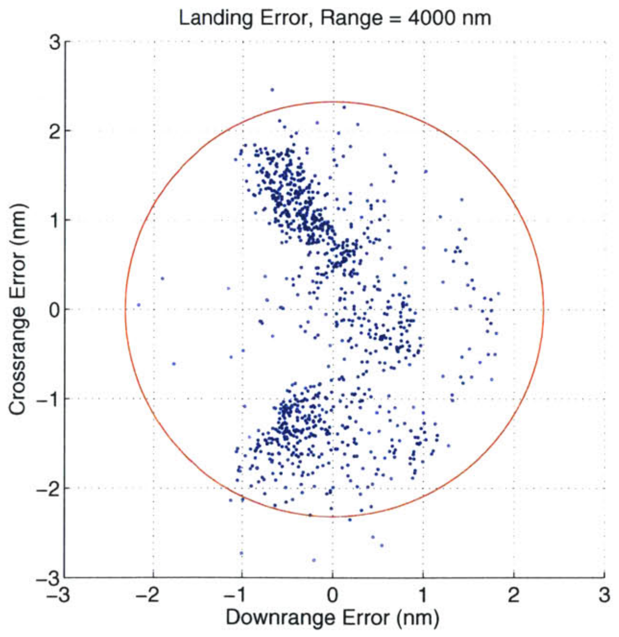


Figure 3-1: Landing accuracy

Landing accuracy can also be viewed in the type of plot shown in Figure 3-2. This is a useful way to view different sets of trials together. The hash mark in the center of the boxes in Figure 3-2 represents the mean miss distance for the total number of trials for a particular algorithm at that target range. The top and bottom hash marks represent the maximum and minimum miss distance, respectively. The box around the mean represents the ± 1 standard deviation from the mean, or $\pm 1\sigma$.

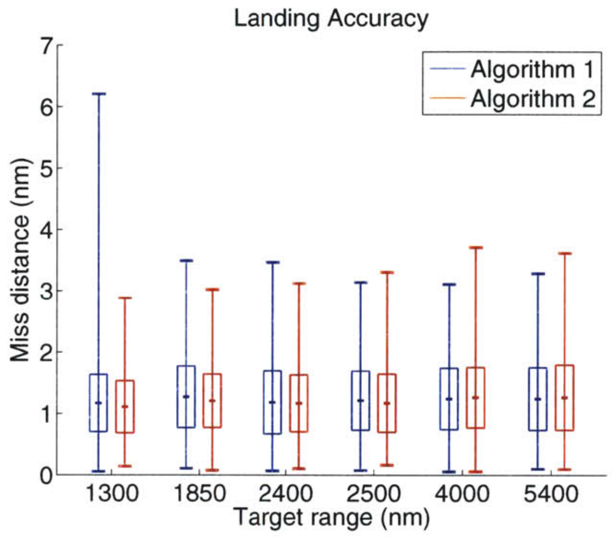


Figure 3-2: Sample landing accuracy comparison plot

This type of plot is also used for other single-value metrics that can be compared between algorithms: maximum g load, maximum heat rate, total heat load, and control saturation.

3.1.2 g Loads

g loads are the result of aerodynamic acceleration on the vehicle. The aerodynamic acceleration is equal to the sum of the lift and drag forces divided by the mass of the vehicle, and it is measured in units of gravitational acceleration at the surface of the Earth. Therefore, 1 g is equal to approximately 32.2 ft/s^2 (9.81 m/s^2). Throughout this thesis, the terms *drag* and *lift* are used to describe the components of the vehicle's aerodynamic acceleration. This is different than how these terms are traditionally used to describe the aerodynamic forces, which do include the mass of the vehicle. When it is necessary to refer to the true aerodynamic forces (the product of mass and acceleration), the terms *drag force* and *lift force* will be used.

There are two g load requirements: one for the maximum load and a set for sustained loads. The maximum allowable is 10 g's, and the duration-based loads are taken from NASA's Human-Systems Integration Requirements (HSIR) [7]. Sample trajectories and the duration-based g load limits are shown in Figure 3-3.

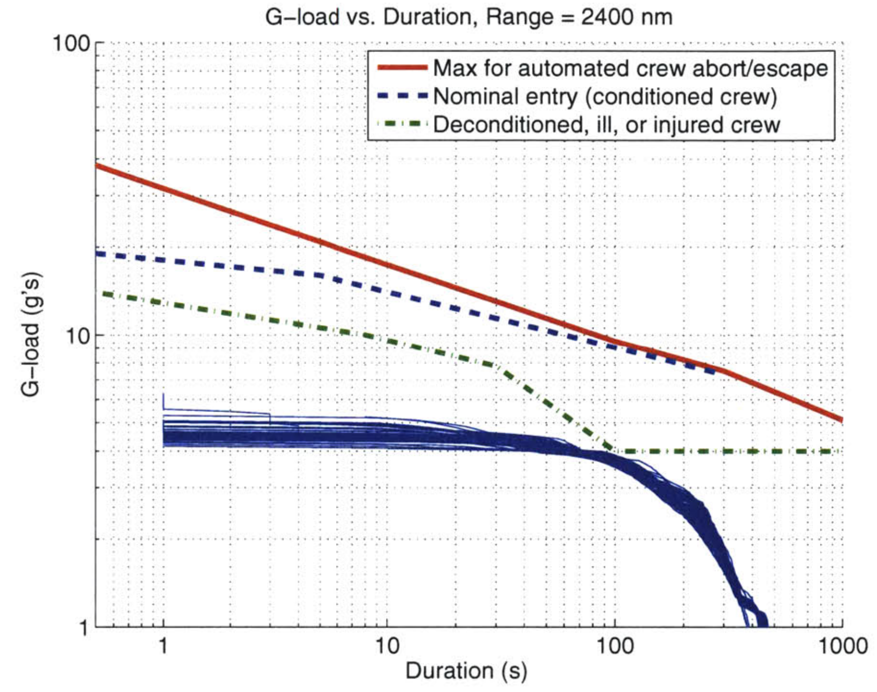


Figure 3-3: Duration-based g loads

Figure 3-3 depicts the acceleration limits in the $+X$ direction, which is sometimes referred to as "eyeballs in." This represents the force that seated crewmembers would feel pushing them into the backs of their seats; the majority of the deceleration force during reentry is in this direction. It is desirable for the entry loads to remain beneath the dotted green line denoting limits for deconditioned, ill, or injured crew; however, the limit defined by the "green corner" around 100 seconds is more a guideline than a hard limit.

3.1.3 Aeroheating

Aerodynamic heating covers two different parts: maximum heat rate and total heat load. The first is simply the highest heat rate seen at any point in the trajectory, and the second is the result of integrating the heat rate over the entire trajectory time, as shown in Figure

3-4. No limits are specified here for either of these metrics, but lower is generally better for both conditions. Unfortunately, heat rate and heat load are usually inversely related. To reduce the peak heat rate, it is necessary for the vehicle to decelerate more slowly; however, slow deceleration increases the total trajectory time, which increases the total heat load.

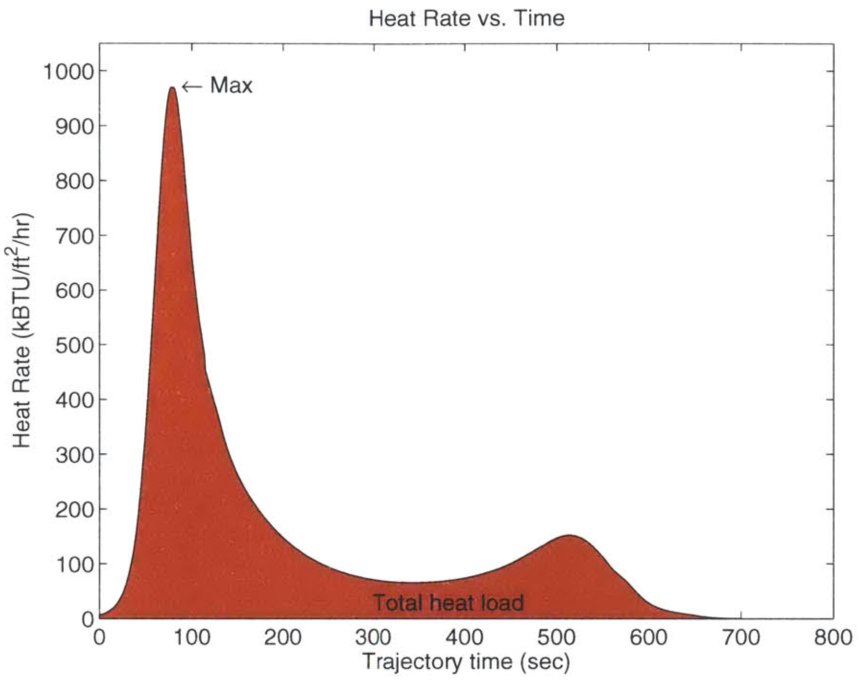


Figure 3-4: Heat rate, integrated to produce total heat load

3.2 Margin Based Metrics

These metrics help to show where the vehicle is performing with respect to the total entry corridor. The entry corridor is defined by certain limits on g loads, heating, and other factors. The amount of margin the vehicle has is determined by how close it is to these limits. Ideally, a guidance algorithm makes full use of the vehicle's total capability and keeps it near the center of the corridor.

3.2.1 Control Saturation

The mathematical relationship between L/D ratio and bank angle, ϕ, is shown in the following equation:

$$\frac{\left(\frac{L}{D}\right)_{vertical}}{\left(\frac{L}{D}\right)_{max}} = \cos\phi \tag{3.1}$$

The total L/D can be decomposed into a vertical and a horizontal component. The vertical direction controls the vehicle's rate of energy decay and its downrange capability, and the time history of the vertical component is what shapes the overall trajectory. This fraction of the total L/D gives some insight into how hard guidance is working to achieve the desired trajectory.

Throughout the entry, guidance generates bank angle commands to steer toward the target and keep the vehicle within acceptable limits on g load and other parameters. If the vehicle is generally on the right course to the target and its predicted trajectory requires only small modifications, guidance can command small changes in the bank angle (and thus vertical L/D fraction) to achieve the desired result. If, however, the required changes to the trajectory are large, guidance must command larger changes in the bank angle to achieve its goal. If putting the total amount of L/D in the vertical direction still does not accomplish the desired result, guidance is unable to command anything more to correct the trajectory. At this point, the controller is saturated.

The bank angle command history of a trajectory shows the amount of control effort guidance exerted to achieve its goals. Here again, there is a tradeoff. A guidance algorithm should not limit the capability of the vehicle, but it also should not use the vehicle's entire capability to make small changes to the trajectory. In this situation, minimizing control saturation is generally best. The amount of control saturation, or the length of time guidance commands full lift up ($\phi = 0°$) or full lift down ($\phi = 180°$), shows the time that the vehicle is operating at the edge of its capability. Furthermore, these algorithms limit the bank angle command to 15° and 185° under certain conditions (instead of 0° or 180°) to retain a marginal amount of lateral steering capability. This is also considered control saturation. Figure 3-5 shows a sample plot of bank angle command history with a lift up saturation from about 200 seconds to nearly 1200 seconds.

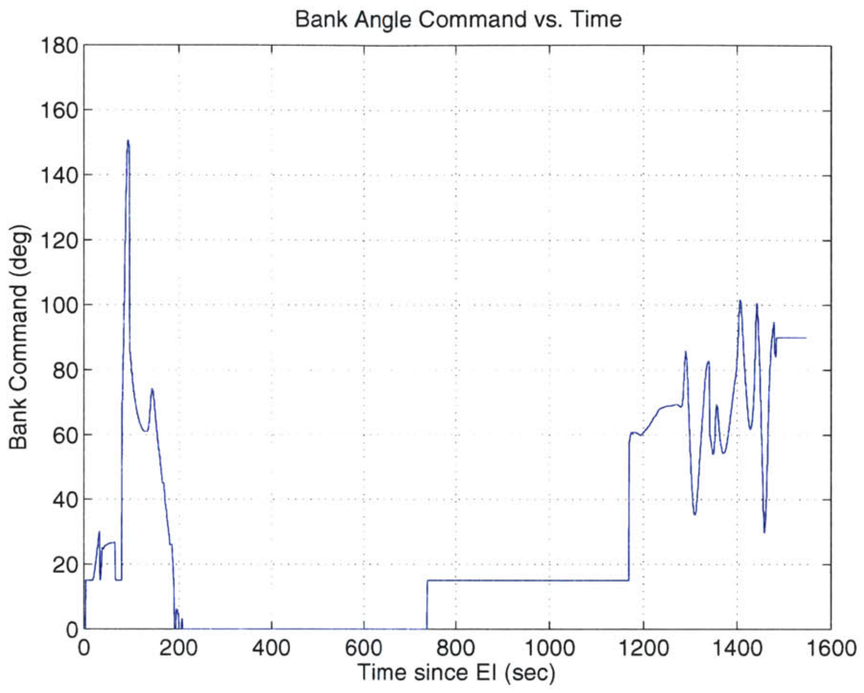


Figure 3-5: Bank angle command history

3.2.2 Final Phase Energy Bucket

The final phase of the guidance algorithms discussed in this thesis use a controller that follows a predetermined reference trajectory to the ground. The "energy bucket" is a way of measuring the margin during this phase. It is a plot of energy and range-to-go to the target landing site during the final phase. To keep this independent of vehicle parameters, "energy" here actually refers to energy per unit weight, or E/W.

$$E = mgh + \frac{1}{2}mv^2 \quad \rightarrow \quad E/W = h + \frac{1}{2}\frac{v^2}{g} \tag{3.2}$$

where:

E = total vehicle energy (potential + kinetic)

m = vehicle mass

g = gravitational acceleration at the surface of the Earth (32.2 ft/s)

h = altitude

v = inertial velocity magnitude

A sample energy bucket plot is shown in Figure 3-6. The left edge of the bucket, shown in red, is a trajectory in which the vehicle flies full lift down (while observing the maximum g load constraint) for the duration of the final phase. Full lift down decelerates the vehicle quickly, and it gives the shortest possible range that is required to not overfly the target. The right edge of the bucket, also in red, is a trajectory in which the vehicle flies full lift up for the duration of the final phase. This decelerates the vehicle slowly and yields the longest possible range to reach the target. These two boundaries represent the physical limitations of the vehicle as it makes its final descent to the ground. The reference trajectory, shown in green, is generally in the center of the bucket, and thus in the middle of the ranging capability. Ideally, this reference keeps a typical trajectory near the center of the bucket. The trajectory's margin is its distance away from the edges of the bucket. The amount of margin is not analytically calculated; the energy bucket is merely a visual representation of the vehicle's capability.

The energy bucket is also important when considering the transition point at which the vehicle enters the final phase. If the vehicle is anywhere inside the bucket at the transition time, it should be able to reach the target landing site. Conversely, if the vehicle transitions to the final phase outside the bucket, it will not be able to reach the target.

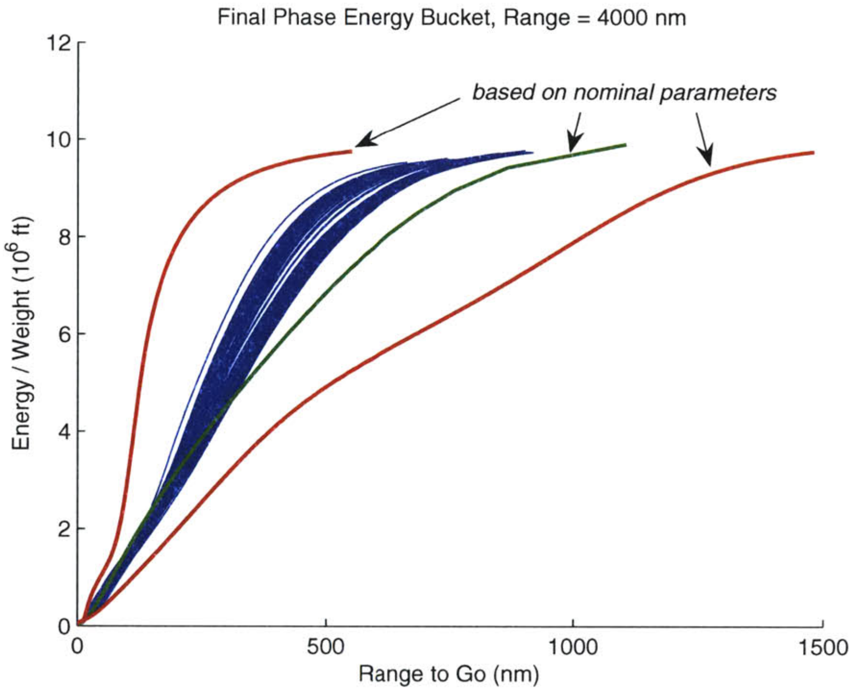


Figure 3-6: Sample energy bucket

Chapter 4

Apollo Guidance Algorithm

4.1 Overview

The goal of any atmospheric entry guidance algorithm is to manage energy and steer the vehicle toward the target landing site, from Entry Interface (EI) to the point at which the parachutes are deployed. A capsule-type spacecraft has only one way to accomplish both of these functions, and that is by modulating the bank angle. To this end, the bank angle command is updated once per guidance cycle, which is once every two seconds.

Energy and downrange are controlled by the portion of the lift vector in the vertical direction, while crossrange is controlled by the portion in the lateral direction. The vertical portion is given by $\cos\phi$ and the lateral portion is given by $\sin\phi$, where ϕ is the bank angle (see Figure 1-5). The Apollo guidance algorithm's top priority is managing energy and downrange, thereby controlling the vertical portion of the lift vector.

The original Apollo guidance algorithm[1] consists of five phases:

1. Initial Roll
2. Huntest
3. Upcontrol
4. Ballistic (Kepler)
5. Final

[1]The algorithm presented here is the version documented in "Reentry Guidance for Apollo" [9]

These phases correspond to the portions of the trajectory labeled in Figure 4-1. This figure assumes that the vehicle is performing a skip entry, and it should be noted that in a direct entry trajectory, the Upcontrol and Ballistic phases are omitted and Huntest transitions directly to the Final phase. Although none of Apollo's atmospheric reentries during the 1960's and 1970's required a skip, the entry guidance algorithm had the capability to perform them.

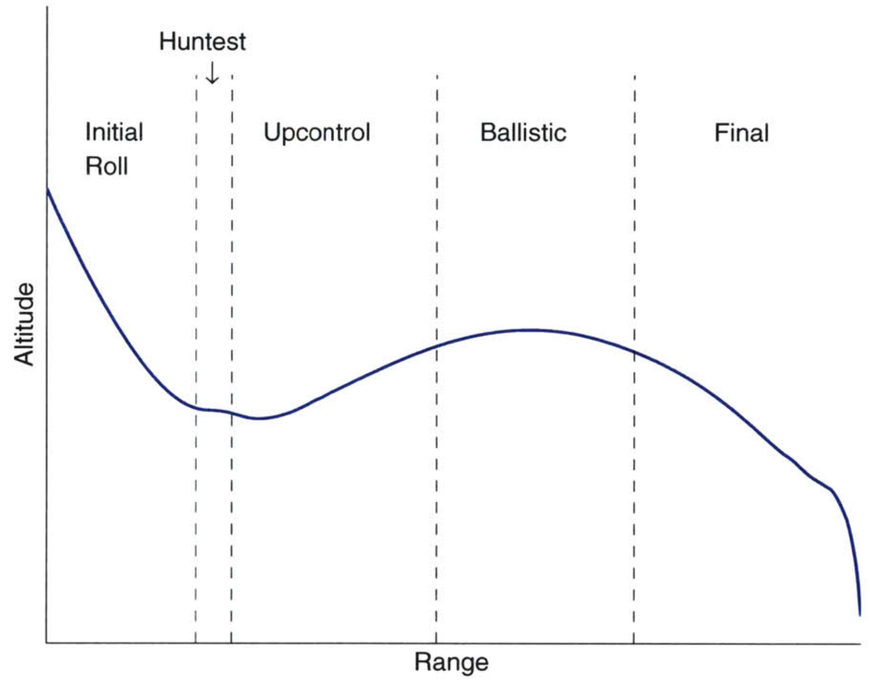


Figure 4-1: Apollo entry guidance phases

Each guidance cycle runs through multiple parts of the algorithm, depending on the point in the trajectory. For example, if the vehicle is currently in a skip and the phase selector is set to Ballistic, each call to guidance goes through Targeting first, then Ballistic, then Lateral Logic. This flow is documented in Figure 4-2.

A detailed description of the Apollo algorithm and its individual phases can be found in References [9] and [10]. All phases of the Apollo guidance algorithm perform important functions, but for this thesis and for the development of the algorithm presented in Chapter 7, particular attention was paid to the Initial Roll, Huntest, and Final phases, and to the transitions between these phases.

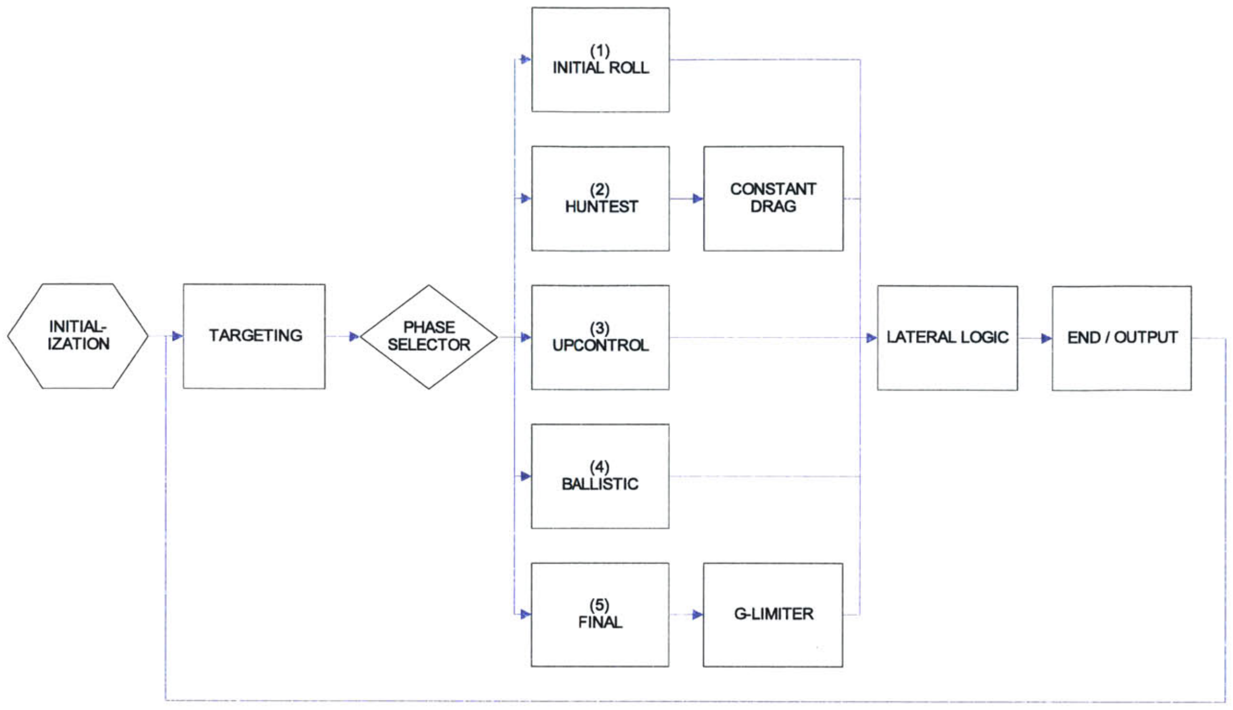


Figure 4-2: Apollo guidance flowchart

4.2 Targeting

During each guidance cycle, the Targeting subroutine is responsible for calculating the remaining downrange and crossrange to the landing site. This is done using geometric calculations documented in Reference [9], and these are the desired range values the algorithm attempts to achieve. This subroutine also determines whether the velocity vector used in the remainder of the guidance algorithm should be calculated in the inertial frame or relative to a fixed altitude above the Earth. This switch from inertial to relative velocity is made once the velocity drops below approximately 13,000 ft/s.

4.3 Initial Roll

The purpose of Apollo's Initial Roll phase is to assure atmospheric capture within the entry corridor by setting the initial bank angle appropriately. Initial Roll starts at Entry Interface (EI), the point at which the vehicle enters the Earth's atmosphere; however, until the vehicle's sensed acceleration reaches 1.6 ft/s^2 (approximately 0.05 g's), Initial Roll transitions directly to the Lateral Logic subroutine to manage crossrange error. Once the drag reaches 0.05 g's, Initial Roll performs a test to determine if the current velocity and flight path angle indicate a shallow entry. If the entry is too shallow, full lift down is commanded to steepen the entry. Otherwise, guidance commands full lift up.

If lift down has been commanded, Initial Roll maintains this orientation until the drag increases to 2 g's, at which point full lift up is commanded to slow the load increase and the vertical descent rate. Lift up is maintained until it is time to transition to Huntest, which occurs when the altitude rate, $\dot{h}$, increases to -700 ft/s. Refer to Figure 4.3 for a diagram of the Initial Roll logic.

4.4 Huntest

This section of the Apollo guidance algorithm determines whether a skip is necessary to reach the target and whether the Constant Drag subroutine is needed to bleed off excess energy first. The name "Huntest" is a contraction of sorts to describe the phase that is "hunting" for an "estimate" of the time in the trajectory, indicated by a velocity, to start the skip in order to meet the downrange requirement.

Huntest assumes a constant L/D trajectory from the start of Upcontrol to the start of the Final phase, and it adjusts the starting velocity of this trajectory to change the total range traveled. Using a series of analytic equations, Huntest first predicts the velocity at atmospheric exit (V_L), which is the beginning of the skip. If V_L is too low, Huntest determines that the vehicle does not have enough energy to complete a skip and Huntest transitions directly to the Final phase. On the other hand, if V_L is too high, Huntest calls the Constant Drag subroutine to bleed off excess energy before making another velocity prediction. The equations used to calculate V_L are described in "Reentry Guidance for Apollo." [9]

If, however, the predicted exit velocity is between the minimum and maximum allowable values, then Huntest uses another set of analytic equations to calculate the flight path angle at atmospheric exit and predict the total downrange traveled. These equations are also documented in Reference [9]. If the predicted range is within the allowable tolerance of

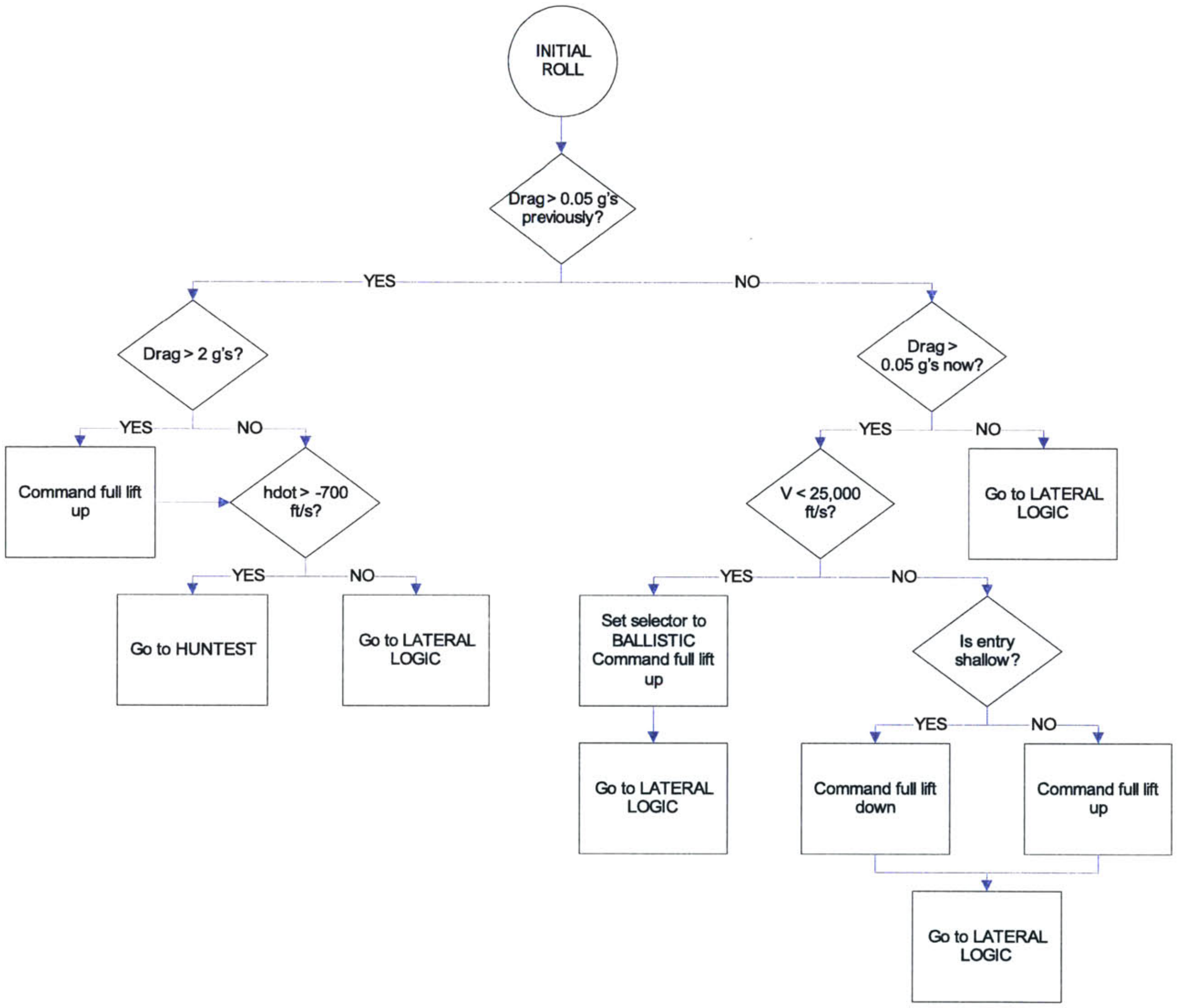


Figure 4-3: Initial Roll phase logic

the desired downrange, Huntest immediately transitions to Upcontrol to set up the skip. If the range error is greater than the tolerance, Huntest adjusts the velocity at which to transition to the constant L/D trajectory and performs another range prediction during the next guidance cycle. This iterative process continues until a suitable transition time to Upcontrol is found, or until the V_L prediction drops below the minimum value and a skip is no longer needed. See Figure 4-4 for a flowchart describing Huntest's logic.

4.4.1 Constant Drag

Constant Drag is a subroutine called by Huntest, and it is a method of decelerating the vehicle by converting kinetic energy into heat. There are several choices for managing energy during atmospheric entry, but this particular method of maintaining a drag level that is constant with time was chosen for Apollo because of its low total heat load. [10]

The Constant Drag controller acts as a regulator to eliminate errors in drag and altitude rate from their desired reference values. It also contains a reference L/D command. This is the open-loop command that would be necessary to maintain the Constant Drag level, D_0, even if there were no drag or $\dot{h}$ errors. The control law, with C_{16} and C_{17} as the control gains, is shown in Equation 4.1.

$$\left(\frac{L}{D}\right)_{cmd} = \left(\frac{L}{D}\right)_0 + C_{16}\left(D - D_0\right) - C_{17}\left(\dot{h} - \dot{h}_0\right) \tag{4.1}$$

The relationship between the reference altitude rate and drag level for Constant Drag can be determined by linearizing the equations of motion, making some simplifying assumptions, and solving for $\dot{h}$. Reference [10] contains the complete derivation; only the relevant parts will be presented here.

First, assume an exponential atmosphere model for the density of air, ρ, at a particular altitude h, where ρ_0 is the density of air at sea level and h_s is the scale height:

$$\rho = \rho_0 e^{-\frac{h}{h_s}} \tag{4.2}$$

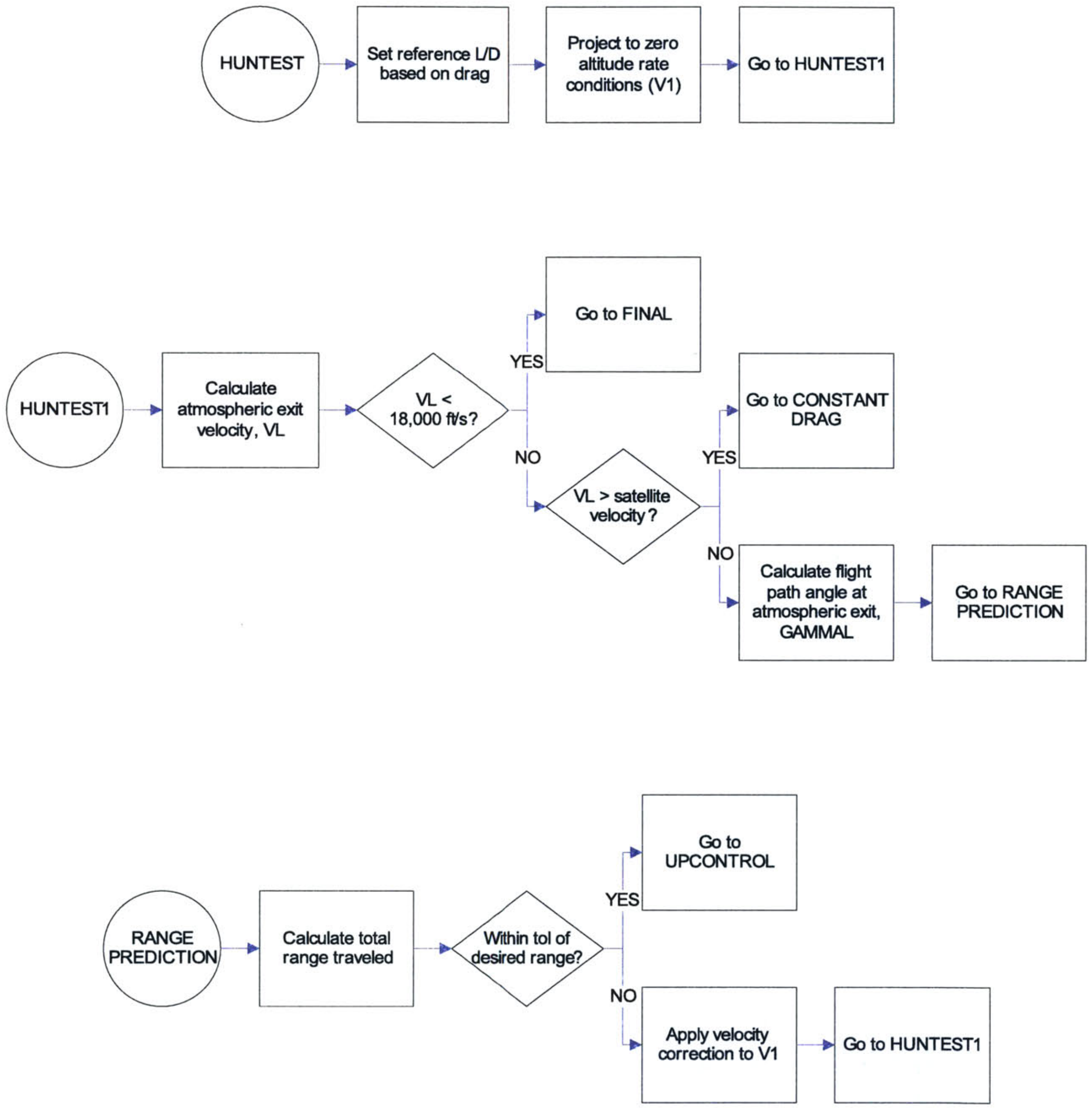

HUNTEST
Set reference L/D based on drag
Project to zero altitude rate conditions (V1)
Go to HUNTEST1
HUNTEST1
Calculate atmospheric exit velocity, VL
VL < 18,000 ft/s?
YES
Go to FINAL
NO
VL > satellite velocity ?
YES
Go to CONSTANT DRAG
NO
Calculate flight path angle at atmospheric exit, GAMMAL
Go to RANGE PREDICTION
RANGE PREDICTION
Calculate total range traveled
Within tol of desired range?
YES
Go to UPCONTROL
NO
Apply velocity correction to V1
Go to HUNTEST1

Figure 4-4: Huntest phase logic

Additionally, the aerodynamic drag acceleration is expressed as the drag force per unit mass:

$$D = \frac{\rho V^2 S_{ref} C_D}{2m} \tag{4.3}$$

where S_{ref} represents the reference area of the vehicle and C_D is the coefficient of drag. Next, take the derivatives of Equations 4.2 and 4.3 with respect to time, assuming that ρ_0, S_{ref}, h_s, m, and C_D are constants.

$$\dot{\rho} = -\rho \frac{\dot{h}}{h_s} \tag{4.4}$$

$$\dot{D} = D\left(\frac{2\dot{V}}{V} - \frac{\dot{h}}{h_s}\right) \tag{4.5}$$

Substitute $\dot{V} = -g \sin\gamma - D$ (Equation 2.45) and $\dot{h} = V \sin\gamma$ into Equation 4.5 and rearrange:

$$\dot{D} = D\left[\dot{h}\left(-\frac{2g}{V^2} - \frac{1}{h_s}\right) - \frac{2D}{V}\right] \tag{4.6}$$

Assuming D is nonzero and constant with time, then $\dot{D} = 0$ and Equation 4.6 can be rearranged to solve for $\dot{h}$:

$$\dot{h} = \frac{-2Dh_s}{\frac{2gh_s}{V} + V} \tag{4.7}$$

Assuming $V \gg \frac{2gh_s}{V}$ for lunar return velocities, Equation 4.7 can be simplified to give the relationship between the reference altitude rate and the reference drag level in Constant Drag:

$$\dot{h}_0 = \frac{-2D_0 h_s}{V} \tag{4.8}$$

The reference L/D value, the first component of the constant drag L/D command in Equation 4.1, is also derived in Reference [10]. Starting with the planar equations of motion for a point mass, introducing Equations 4.2 and 4.3, and dropping $\frac{1}{V^2}$ terms, the reference L/D value is shown to be:

$$\left(\frac{L}{D}\right)_0 = \frac{-g\left(\frac{V^2}{gR_E} - 1\right)}{D_0} \tag{4.9}$$

where:

g = gravitational acceleration near the surface of the Earth (32.2 ft/s)

R_E = radius of the Earth

This reference is plotted for various drag levels in Figure 4-5.

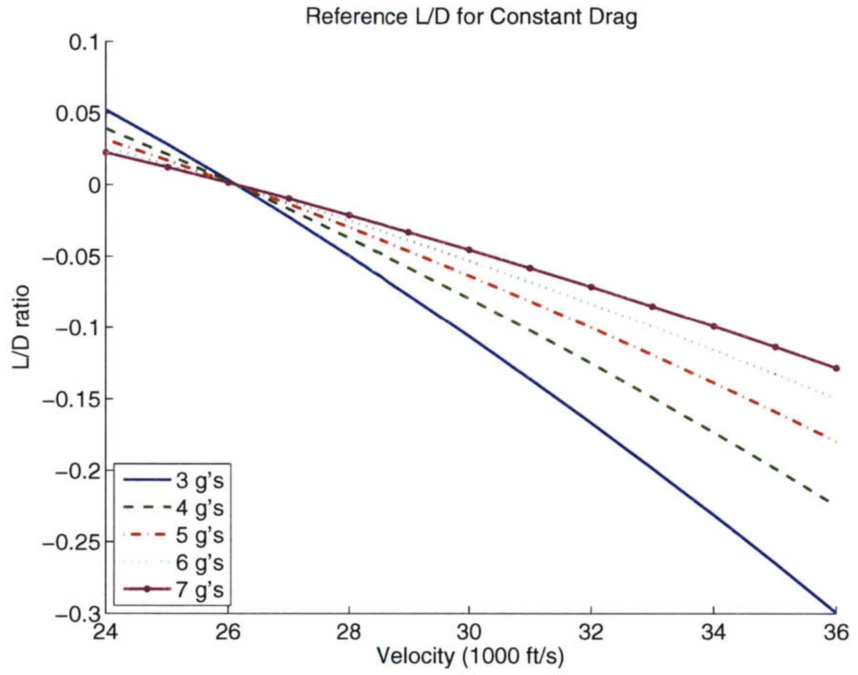


Figure 4-5: Reference L/D ratios for different Constant Drag values

4.5 Final

The Final phase of the Apollo guidance algorithm uses a path-following controller to generate steering commands. The controller follows a stored reference trajectory from the start of Final until the point at which the vehicle's Earth-relative velocity drops below 1000 ft/s. During the Final phase, if guidance determines that the vehicle will overshoot (fly past) the target, full lift down is commanded. Otherwise, if guidance projects that the vehicle will not overshoot the target, the appropriate L/D command is calculated based on the vehicle's state deviation from the reference trajectory. Next, Final modes to the G-Limiter subroutine. If Final's L/D command will cause the vehicle to exceed its maximum load limit of 10 g's, G-Limiter commands lift up. Otherwise, it does not change the command. The logic for this phase is summarized in Figure 4-6.

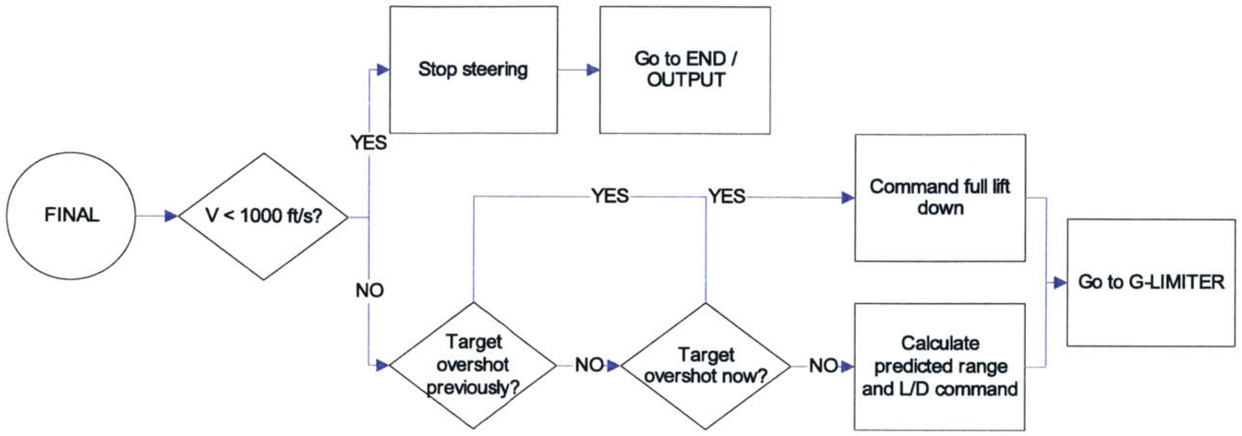


Figure 4-6: Final phase logic

4.5.1 Reference Trajectory

Application

The reference trajectory is stored as a lookup table of nominal values for drag, altitude rate, and range to go, and it uses velocity as the independent variable because velocity is continually decreasing throughout the phase. The lookup table also includes gains or sensitivities in downrange, θ, to the other parameters: drag, altitude rate, and L/D.

$$F_1 = \frac{\partial \theta}{\partial D} \qquad F_2 = \frac{\partial \theta}{\partial \dot{h}} \qquad F_3 = \frac{\partial \theta}{\partial (L/D)}$$

During the Final phase, guidance uses the current velocity to determines the corresponding reference trajectory values by interpolating in the table. Guidance then calculates a predicted range to go, based on the nominal range to go for the current velocity and the necessary corrections for off-nominal values of altitude rate and drag. This range calculation is shown in Equation 4.10.

$$\theta_{pred} = \theta_{ref} + F_2 \left(\dot{h} - \dot{h}_{ref}\right) + F_1 \left(D - D_{ref}\right) \tag{4.10}$$

Once the predicted range is determined, guidance converts this into an L/D command, using Equation 4.11.

$$\frac{L}{D} = \left(\frac{L}{D}\right)_{ref} + \frac{4}{F_3}\left(\theta_{des} - \theta_{pred}\right) \tag{4.11}$$

Generation

The Final phase reference is a constant bank angle trajectory in two dimensions, altitude and downrange, over a spherical non-rotating Earth. Generation of the reference trajectory can be posed as an initial value problem—the vehicle's initial conditions and its constant bank angle uniquely determine its path to the ground. The necessary initial conditions for the reference are altitude, velocity, and flight path angle. Instead of specifying altitude directly, drag can be used instead; the altitude is then determined via the drag equation (Equation 4.3) and the atmosphere model. The initial conditions for the reference trajectory are taken from the typical Final phase initial conditions for the longest target range. For the Apollo reference trajectory, those are:

$$\begin{aligned} V &= 23{,}500 \quad ft/s \\ \gamma &= -2.0 \quad deg \\ D &= 6.0 \quad ft/s^2 \end{aligned}$$

The Apollo program chose a constant L/D of 0.18 in the vertical direction (approximately 60% of the total L/D of the capsule) because it put the resulting landing point near the center of the Apollo capsule's downrange capability. The reference trajectory is generated pre-flight and then stored for use during the entry. Figure 4-7 shows the Apollo reference trajectory plotted from tabulated values for the Apollo capsule [9]. The energy bucket boundaries shown in Figure 4-7 do not represent the Apollo capsule's total ranging capability as they were generated using the CEV configuration; they are shown simply to provide a frame of reference for viewing the Apollo reference trajectory.

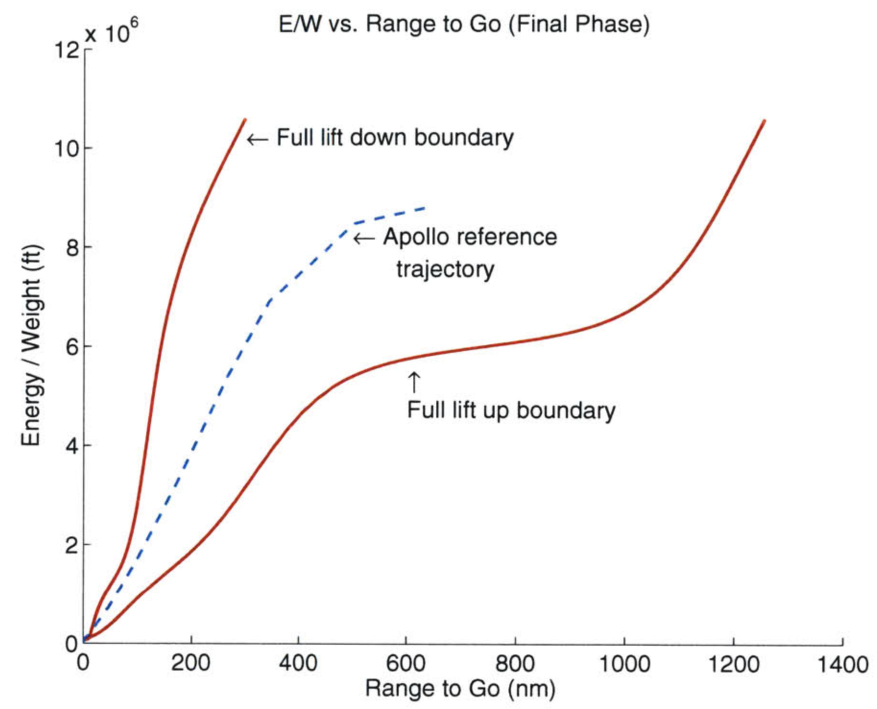


Figure 4-7: Final phase reference trajectory for Apollo guidance algorithm

4.6 Lateral Logic

The Lateral Logic subroutine accomplishes two things: lateral steering and generating the bank angle command from the desired L/D calculated by any of the five phases. This is why Lateral Logic is called after all of the other phases.

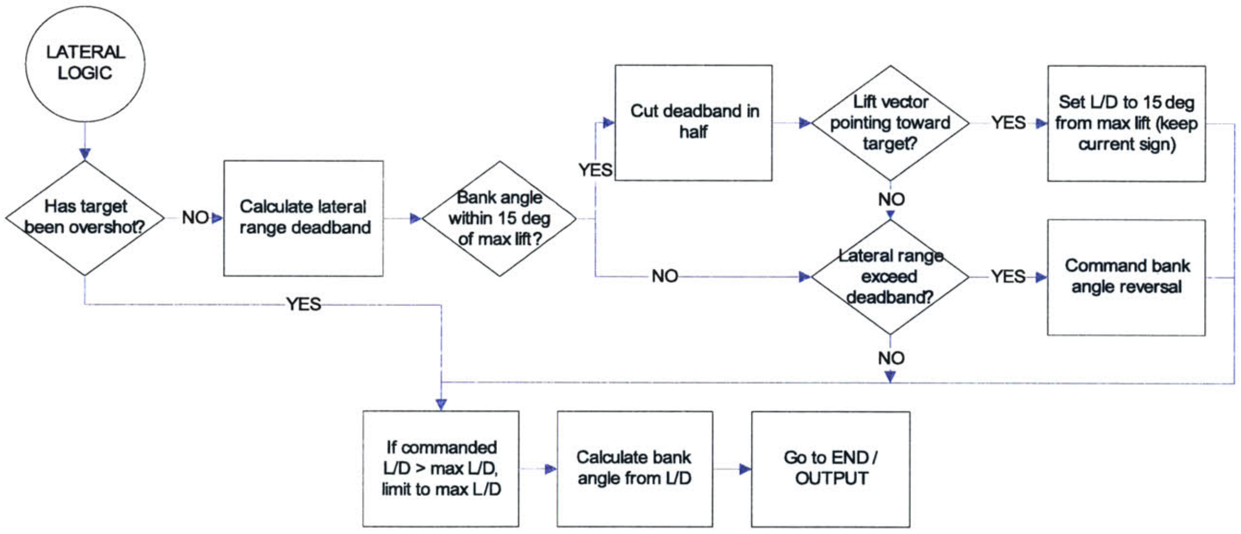
As the vehicle descends through the atmosphere, crossrange error changes over time due to the component of the lift vector in the lateral direction, $\sin\phi$. This error must be corrected in order to steer the vehicle toward the target, and it is accomplished by a series of bank reversals. In this manner, the vehicle "zigzags" its way toward the target, reversing the bank angle from ϕ to $-\phi$ when the crossrange error exceeds a certain amount. The width of this corridor of allowable crossrange error decreases as the vehicle approaches the target to ensure greater accuracy near the ground.

In each call to Lateral Logic, guidance first calculates the allowable crossrange error corridor, or deadband. Next, the deadband is halved if the bank angle command is within $15°$ of maximum lift up or down. This is done because bank angles in that range are severely limited in their lateral steering capability; therefore, the corridor must be narrowed accordingly. In this case, if the lift vector is pointing toward the target, the bank angle command is reset to either 15° or 165°, depending on whether the current lift vector is at full lift up or full lift down. Regardless of the current value of the bank angle however, if the lateral range exceeds the deadband, a bank reversal is commanded.

If guidance determines that the target will be overshot, the deadband is not updated from the previous guidance cycle and the bank angle is not limited to at least 15° away from maximum lift. Regardless of whether the projected target will be overshot, Lateral Logic's last two steps are to keep the commanded L/D within the total available L/D of the vehicle and to calculate the bank angle command using Equation 4.12.

$$\phi_{cmd} = \text{sign}\left(\phi_{current}\right) \cdot \text{acos}\left[\frac{(L/D)_{des}}{(L/D)_{max}}\right] \tag{4.12}$$

The logic for this subroutine is shown in Figure 4-8.



LATERAL LOGIC
Has target been overshot?
NO
Calculate lateral range deadband
Bank angle within 15 deg of max lift?
YES
Cut deadband in half
Lift vector pointing toward target?
YES
Set L/D to 15 deg from max lift (keep current sign)
NO
NO
Lateral range exceed deadband?
YES
Command bank angle reversal
NO
YES
If commanded L/D > max L/D, limit to max L/D
Calculate bank angle from L/D
Go to END / OUTPUT

Figure 4-8: Lateral Logic subroutine

Chapter 5

Draper PredGuid Algorithm

5.1 Overview

In 2005, the Charles Stark Draper Laboratory designed a new entry guidance algorithm based on the original Apollo algorithm and containing the same five phases [2]. The Draper versions of Initial Roll and Huntest are essentially unchanged from the Apollo versions, but Upcontrol and Ballistic have been significantly altered. The reference trajectory for the Final phase has been changed for the Draper algorithm, but the phase logic remains the same as Apollo's.

Apollo's Upcontrol and Ballistic phases use simplified analytic equations to make range predictions and calculate bank angle commands. The Draper algorithm eliminates those calculations and instead uses PredGuid, a model-based numeric predictor-corrector algorithm, to generate bank angle commands during the Upcontrol and Ballistic phases for skip trajectories.

5.2 PredGuid

PredGuid was originally developed by Draper Laboratory in the late 1980's as an aerocapture guidance algorithm, and it was modified in 2005 to be suitable for this application. A detailed description of PredGuid can be found in Reference [4], and the necessary modifications for the Draper entry guidance algorithm are presented in Reference [2].

PredGuid's role is to determine the bank angle commands that will manage the vehicle's energy and downrange from the beginning of Upcontrol to the beginning of the Final phase—the skip portion of the entry. Each guidance cycle, the input to PredGuid is the downrange distance remaining between the vehicle's current position and the start of the reference trajectory. PredGuid assumes a constant bank angle trajectory to cover this distance.

During each guidance cycle, PredGuid performs a number of iterations to find the constant bank angle trajectory that will meet its objective. The prediction phase is executed first—PredGuid makes an initial guess at the constant bank angle required to cover the desired downrange distance and integrates the equations of motion forward in time until the termination conditions are met. PredGuid then compares the predicted distance traveled with the desired downrange. If the difference between the prediction and the requirement is more than a certain tolerance, PredGuid moves to the correction phase and adjusts the bank angle. If the predicted downrange is too long, PredGuid increases the bank angle to shorten the range, and if the predicted downrange is too short, PredGuid decreases the bank angle to lengthen the range. Once the bank angle is adjusted, PredGuid goes back to the prediction phase and integrates the equations of motion again to determine the new range traveled. This process repeats until PredGuid has either found a bank angle solution for which the predicted range is within the tolerance of the downrange requirement, or the maximum allowable number of iterations is reached. If the maximum number of iterations is reached, PredGuid's last bank angle guess is output as the bank angle command, even if the predicted range does not meet the desired range.

5.2.1 PredGuid's Target Range

The primary input to PredGuid during each guidance cycle is the remaining range to go from the vehicle's current position to the start of the Final phase reference trajectory. This is the downrange that PredGuid is attempting to cover with a constant bank angle trajectory, and it is calculated by subtracting the Final phase range from the total desired range remaining. The Final phase range is not a static number however; it is a value that changes based on

PredGuid's predicted state at the end of the skip. It is a linear combination of the nominal reference trajectory range and range corrections based on drag and altitude rate.

As mentioned in Section 4.5, the Final phase reference trajectory is stored as a lookup table for range to go, altitude rate, and drag, with velocity as the independent variable. PredGuid estimates the Final phase range by using the predicted velocity at the end of the skip to interpolate in the reference trajectory table. The estimated Final phase range is the sum of the nominal range to go for the predicted velocity, the drag error, and the altitude rate error.

$$\theta_{Final_{est}} = \theta_{Final_{ref}} + F_2\left(\dot{h}_{PG} - \dot{h}_{ref}\right) + F_1\left(D_{PG} - D_{ref}\right) \tag{5.1}$$

where:

$\theta_{Final_{est}}$	=	PredGuid's estimate of the Final phase range
$\theta_{Final_{ref}}$	=	Nominal Final phase range for PredGuid's predicted velocity
$\dot{h}_{PG}$	=	PredGuid's predicted altitude rate at start of Final
D_{PG}	=	PredGuid's predicted drag at start of Final
$F_1,\ F_2$	=	gains for drag error and altitude rate error (see Section 4.5)

PredGuid's target range is then total range to go minus $\theta_{Final_{est}}$.

This is the same range estimation process that is performed during the Final phase, except that PredGuid is using estimated quantities at the end of the skip rather than actual values of drag and $\dot{h}$ that the Final phase controller uses.

5.2.2 PredGuid Termination Conditions

While PredGuid is integrating the equations of motion forward in time during its prediction phase, the predicted trajectory must meet several conditions in order to terminate the integration and consider the resulting trajectory complete. First, PredGuid stops integrating if it determines that the trajectory has escaped the Earth's gravitational field or crashed into the Earth. If the trajectory has neither escaped nor crashed, it is considered a good or "captured" solution and must meet three conditions to stop integrating:

1. The skip has started

2. Altitude rate is negative

3. Drag is greater than 0.2 g's

Condition 2 indicates that the capsule is past the apogee of its ballistic trajectory and its altitude is now decreasing. Condition 3 indicates that the capsule is back in the atmosphere—the higher air density increases the drag on the vehicle. The last two conditions, however, do not uniquely determine the end of the skip. Those conditions could be met during the initial descent into the atmosphere, and if PredGuid is started during this time, those conditions would signal to PredGuid to stop integrating immediately. Figure 5-1 shows the parts of a typical trajectory, both before and after the skip, in which conditions 2 and 3 are met. To distinguish between these two areas, PredGuid needs an additional flag which determines that the vehicle has already made its first descent into the atmosphere and the skip has begun.

This flag, signaling that the skip has started, looks for a point in the predicted trajectory at which the altitude rate becomes positive or the drag drops below 0.2 g's. This indicates to PredGuid that the predicted trajectory has started the skip. Once the predicted trajectory has passed this point, PredGuid is free to terminate the integration as soon as the last two conditions are met. Table 5.1 summarizes PredGuid's different cases and corresponding conditions required to terminate integration.

Table 5.1: PredGuid Integration Termination Conditions

Case	***Termination Conditions***
Escape	Altitude $> 600,000\ ft$
Crash	Altitude $< 0\ ft$
Capture	1. "Skip has started" flag = true 2. $\dot{h} < 0$ 3. Drag $> 0.2\ g$'s

5.3 Upcontrol and Ballistic

The Upcontrol and Ballistic phases of both the Draper baseline algorithm and the Apollo algorithm are responsible for managing the skip portion of the trajectory. Upcontrol is active

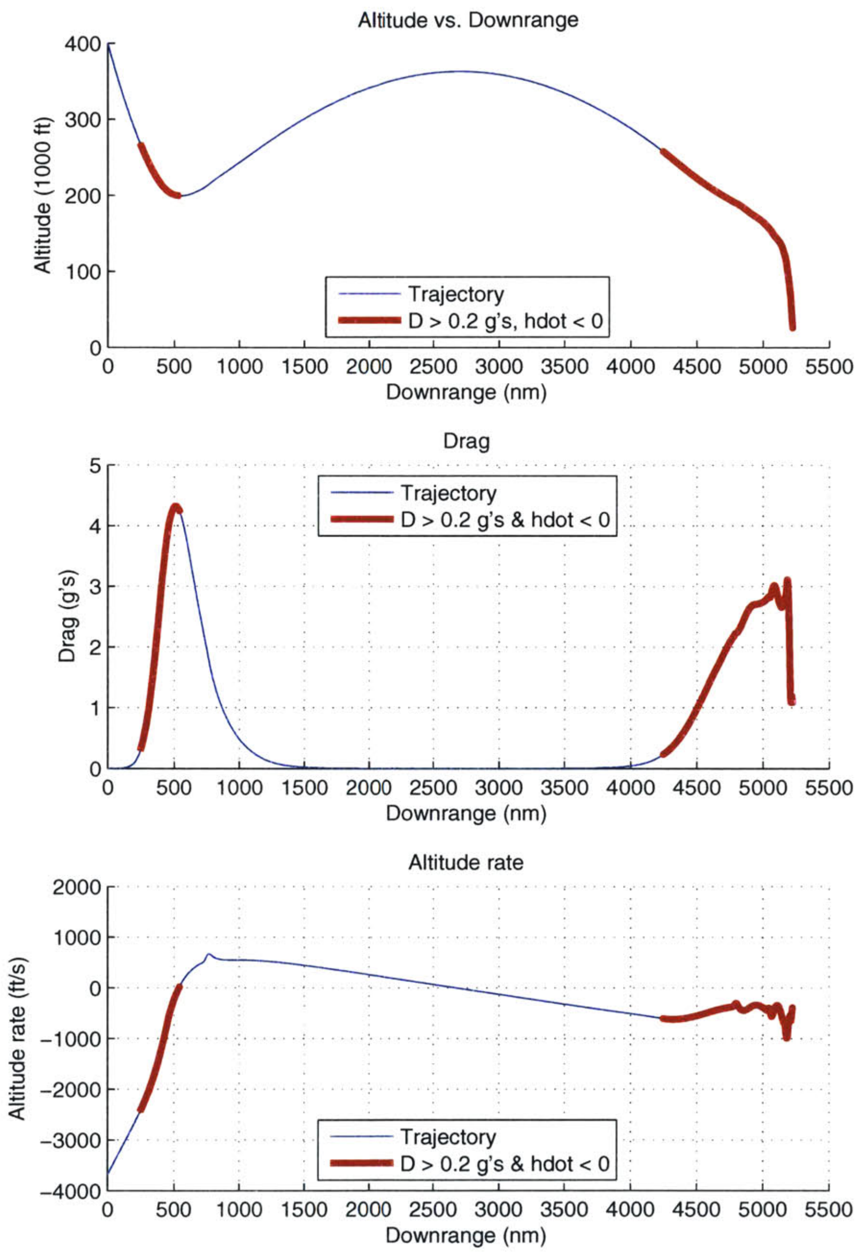


Figure 5-1: Drag and altitude rate for first and second entries

before Ballistic, during the atmospheric portion of the skip. Ballistic, as the name suggests, is active during the exoatmospheric portion of the skip, which is defined for these algorithms to be anytime the drag is below 0.2 g's. These two phases perform essentially the same function—running PredGuid until the skip is complete and then transitioning to the Final phase. They are two separate phases instead of one because they are checking for different requirements to transition to the Final phase.

5.3.1 Upcontrol

In the Apollo algorithm, Upcontrol guides the vehicle along a reference trajectory to the atmospheric exit conditions calculated in Huntest. In the Draper baseline, the Upcontrol reference trajectory and associated steering logic is eliminated and PredGuid's targeting algorithm is used instead to generate the bank angle commands. The result is that Upcontrol is simply a "wrapper" for PredGuid—Upcontrol checks for conditions that require a transition to a different phase, but if none of those conditions are met, it runs PredGuid instead. Upcontrol transitions to either Ballistic or Final, depending on which conditions are met.

For Upcontrol to transition to the Final phase, the current velocity must be within 500 ft/s of the predicted atmospheric exit velocity, V_L, and the altitude rate must be negative. This transition is typical of low altitude, short range skips that do not leave the atmosphere. For Upcontrol to transition instead to the Ballistic phase, the current drag must be less than 6 ft/s^2 (0.2 g's). This transition is more common for high altitude, long range skips. If no phase change is required, Upcontrol runs PredGuid to generate the bank angle command and then calls the Negative Test subroutine, which is carried over from Apollo. Negative Test commands an L/D of zero if the current L/D is negative and the drag is above 175 ft/s^2 (approximately 5.4 g's); otherwise, it does not change PredGuid's bank command. Negative Test then transitions to Lateral Logic. Figure 5-2 summarizes the logical flow of Upcontrol.

5.3.2 Ballistic

Like Upcontrol, the Ballistic phase in the Draper baseline is essentially just a wrapper for PredGuid. Ballistic checks for the condition that is required to transition to the Final phase;

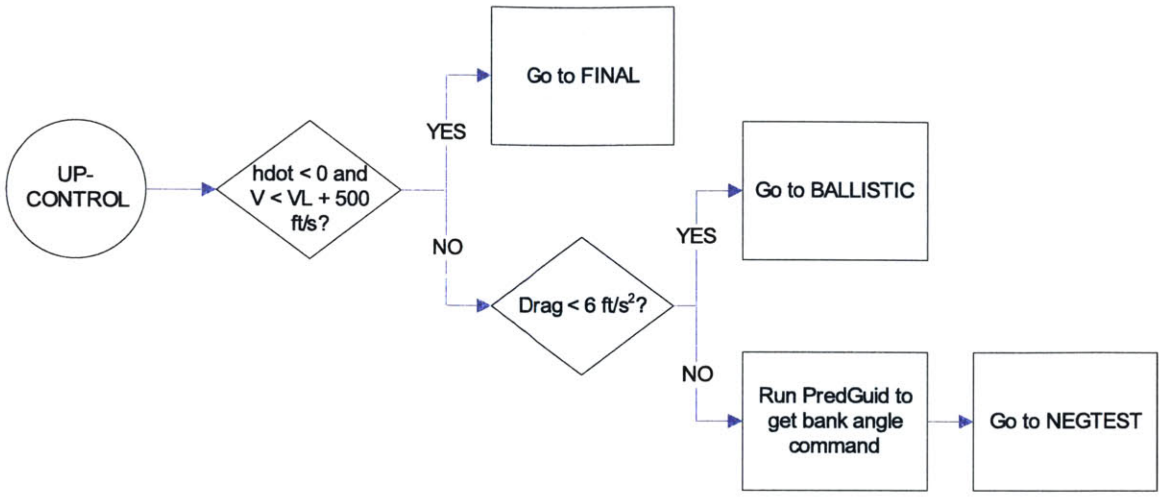


Figure 5-2: Upcontrol phase logic

if it is not met, Ballistic runs PredGuid to generate the bank angle command.

To transition to the Final phase, the current drag must be greater than 6.5 ft/s^2 (0.2 g's). This indicates that the vehicle has reentered the atmosphere and the skip is complete. This drag requirement is slightly higher than the drag required to transition from Upcontrol to Ballistic (6 ft/s^2), which is to ensure that guidance does not transition from Upcontrol to Ballistic to Final in the same guidance cycle. If the drag is below 0.2 g's but above 0.05 g's, PredGuid is run to generate the bank angle command. Ballistic then transitions to Lateral Logic to manage the lateral steering. If the drag is below 0.05 g's, PredGuid is not run at all. At such a high altitude and low air density, changing the bank command does not appreciably alter the trajectory, so the previous guidance cycle's bank command is held and PredGuid is shut off to conserve fuel. The guidance logic for the Ballistic phase is shown in Figure 5-3.

5.4 Final

Although the Final phase logic is the same for the Draper baseline and the Apollo algorithms, the Draper baseline uses an updated reference trajectory. The reference trajectory is dependent on the particular vehicle's mass, dimensions, and other physical characteristics, and since the Apollo and CEV capsules do not the same size-mass scale, the Apollo refer-

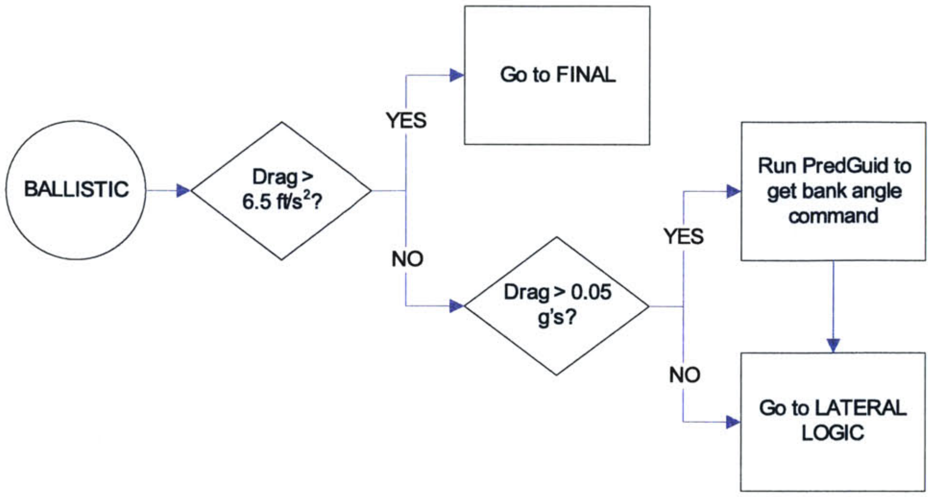


Figure 5-3: Ballistic phase logic

ence was not well suited for the larger CEV capsule. Additionally, the typical Final phase initial conditions for the two configurations are different, due to their different maximum target ranges. For the Draper/CEV configuration, the maximum target range was 5400 nm (10,000 km) [2]. The Final phase initial conditions for this target range, and thus the initial conditions for the Draper baseline reference trajectory, are:

$$\begin{aligned} V &= 25,262 \quad ft/s \quad (7,700\ km/s) \\ \gamma &= -1.0 \quad deg \\ D &= 6.0 \quad ft/s^2 \quad (0.2\ g\text{'s}) \end{aligned}$$

The reference L/D for the Draper baseline reference trajectory is 60% of the total L/D capability for the 2005 CEV capsule concept. In the same manner as Apollo, this is the approximate center of the vehicle's ranging capability. Figure 5-4 shows a comparison between the Apollo and Draper reference trajectories. A complete description of the Draper reference trajectory redesign can be found in Reference [2].

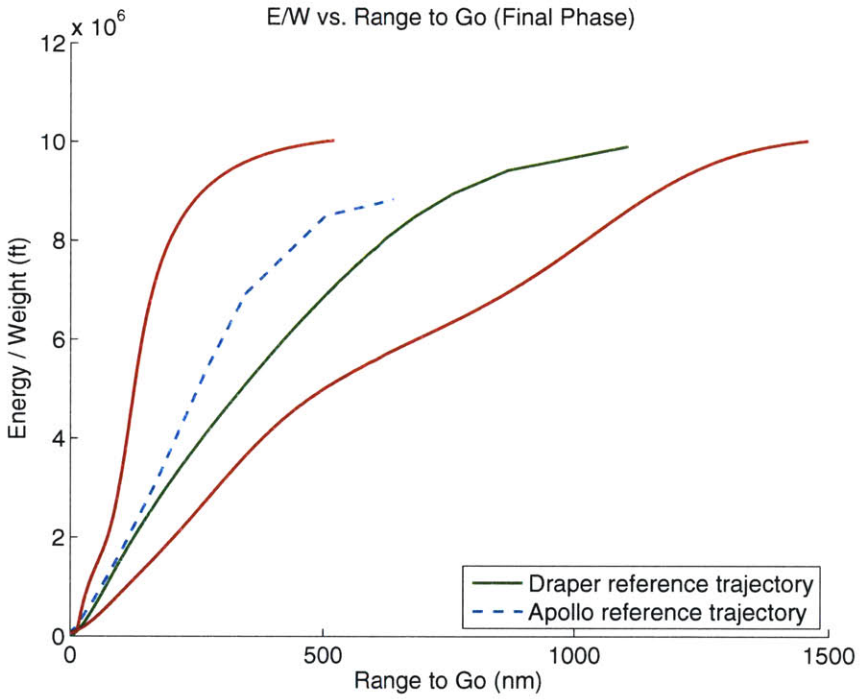


Figure 5-4: Draper baseline and Apollo reference trajectories

Chapter 6

Problem Description

6.1 Overview

The Apollo entry guidance algorithm is amazingly clever and flexible, considering the important functions it had to perform using such limited computing power. Accurate predictions and calculations are made throughout the algorithm using only analytic equations, but to do that with very limited resources, the predictions rely on many simplifications and heuristics. While the equations are tuned extremely well for the Apollo configuration, they are less accurate for other configurations.

The Draper baseline algorithm upgraded Apollo's skip guidance logic and extended the downrange capability to 5400 nm (10,000 km) by using PredGuid in the Upcontrol and Ballistic phases. PredGuid eliminated many of the empirical equations used in the skip portion of the Apollo algorithm, but many others still remain in the energy management and phase transition logic preceding the transition to PredGuid. The Draper baseline algorithm exhibits difficulty in managing energy, particularly for direct entry trajectories. In addition, some phase transitions require instantaneous changes from one reference vehicle state to another. The vehicle is obviously incapable of such transitions, and the resulting guidance commands can overwhelm the flight control system.

This chapter presents five major areas of concern regarding the Draper baseline, covering all five phases of the algorithm:

1. Energy management and Constant Drag
2. Final phase reference trajectory for short range entries
3. Initial Roll's shallow entry test
4. Determining whether a skip is necessary
5. Time of transition to PredGuid

These particular issues are discussed in detail in this chapter. Enhancements to the Draper baseline entry guidance algorithm have been designed to improve these areas, and they are presented as the Predictor algorithm in Chapter 7.

6.2 Energy Management and Constant Drag

A spacecraft on a lunar return trajectory enters the Earth's atmosphere with a tremendous amount of energy. Apollo capsules returning from the Moon, for example, entered the Earth's atmosphere at a speed of approximately $36,000\ ft/s$ [12]. If the vehicle's target landing site is far away from its EI location, much of that energy must be conserved in order to cover the large required distance. If the landing site is near its EI location, the vehicle must quickly reduce its initial energy so that it does not miss the landing site by overshooting it. In these algorithms, energy is depleted by increasing the aerodynamic drag on the vehicle and decelerating it. To deplete a fixed amount of energy, the spacecraft can either decelerate quickly using a high drag, or decelerate slowly using a low drag. The former case is desirable if the range to go is short, and the latter is desirable if the range to go is long.

The original Apollo guidance algorithm uses a single drag level in Constant Drag: 130 ft/s^2, or about 4 g's. With a single drag level, the only way to modify the amount of energy depleted is to vary the amount of time the vehicle flies at that drag level. This becomes restrictive for short target ranges— the spacecraft may be unable to deplete enough energy in the amount of time and downrange available. For example, if the vehicle is unable to deplete enough energy during Huntest and Constant Drag, the remainder of the energy must be depleted during the Final phase. This can increase the g load beyond desirable levels during the Final phase, as shown in Figure 6-1. If the g load exceeds the allowable limits, the single Constant Drag value limits the short range capability of the vehicle.

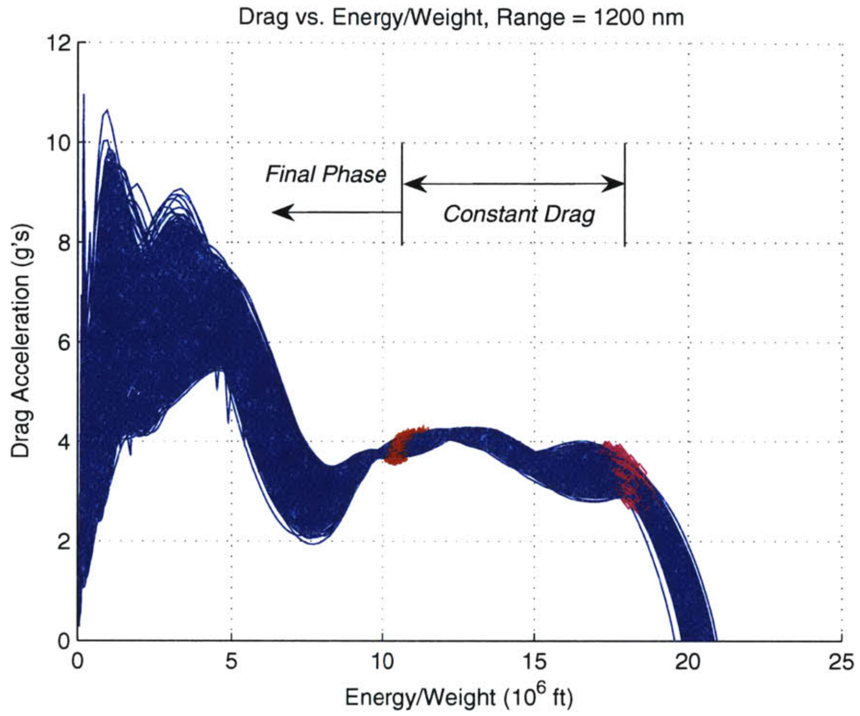


Figure 6-1: Drag vs. E/W for 1200 nm direct entry Monte Carlo trials

A guidance algorithm that allows various Constant Drag levels based on the desired energy reduction over a specific range would keep the drag level within desired limits throughout the entire trajectory, rather than just in phases prior to Final. This should increase the algorithm's short range direct entry capability and reduce the maximum g load.

6.3 Final Phase Reference Trajectory

The Draper baseline algorithm, as mentioned in Section 5.4, uses a different reference trajectory for the Final phase than the Apollo algorithm. This trajectory works very well for long skip trajectories, which is to be expected since the initial conditions for the Draper reference were taken from a typical long range skip entry. For direct entry cases, however, this reference trajectory does not work as well.

In a skip trajectory, the Final phase begins when the vehicle reenters the atmosphere after the skip, and at this point, the drag is quite low. In a direct entry trajectory, the Final phase begins after the vehicle completes Huntest and Constant Drag. Using the Apollo algorithm, this means that the vehicle will be flying at a drag level of 4 g's when it enters the Final phase, which is 3.8 g's higher than the initial drag of the reference trajectory.

To demonstrate the problems this initial drag "mismatch" can create in the Final phase, consider the following example from a direct entry target range of 1300 nm. Using the Apollo algorithm, the average entry conditions—velocity, drag, altitude rate, and desired range to go—for 1000 Monte Carlo trials are:

$$\begin{aligned} V &= 25,991 \quad ft/s \\ D &= 122 \quad ft/s^2 \\ \dot{h} &= -198 \quad ft/s \\ \theta_{des} &= 518 \quad nm \end{aligned}$$

First, interpolate in the reference trajectory to get the following reference parameters:

$$\begin{aligned} \dot{h}_{ref} &= -441 \quad ft/s \\ D_{ref} &= 6.4 \quad ft/s^2 \\ F_1 &= -18.2 \quad nm/(ft/s^2) \\ F_2 &= 0.755 \quad nm/(ft/s) \\ F_3 &= 2759 \quad nm \\ \theta_{ref} &= 1106 \quad nm \\ (L/D)_{ref} &= 0.22 \end{aligned}$$

Next, calculate the predicted range to go (θ_{pred}), noting that the calculated value is negative:

$$\theta_{pred} = \theta_{ref} + F_2\left(\dot{h} - \dot{h}_{ref}\right) + F_1\left(D - D_{ref}\right) = -817\ nm \tag{6.1}$$

A negative predicted range to go indicates that the vehicle will overfly the target. In this situation, a negative L/D command is necessary to shorten the range; however, the L/D command that is calculated based on the predicted range to go is actually positive:

$$\frac{L}{D} = \left(\frac{L}{D}\right)_{ref} + \frac{4}{F_3}\left(\theta_{des} - \theta_{pred}\right) = 2.15 \tag{6.2}$$

Not only does the calculated L/D command have the incorrect sign, it far exceeds the total L/D of the vehicle. This Final phase initial drag error ultimately results in saturating the controller as it tries to begin following the reference trajectory. Even if the controller is not saturated, forcing it to follow a reference that is far away from the vehicle's current position requires it to expend more fuel to reach the reference trajectory.

Figure 6-2 shows the Final phase energy bucket for this target range, further illustrating the "mismatch" between a typical direct entry trajectory and the reference trajectory. In the figure, the reference trajectory is represented by the single green line leaning toward the right half of the bucket, and the 1000 Monte Carlo trajectories, using the Apollo algorithm, are shown by the blue lines clustered on the far left side of the bucket. These direct entries transition to the Final phase with a significantly shorter range to go than the reference trajectory. The general trajectory shape is also substantially different, and this is due to the higher drag at the start of the Final phase.

Figure 6-3 shows how the shape of the reference trajectory changes as the initial drag is increased, while keeping the initial velocity, flight path angle, and L/D fraction constant. A reference trajectory with a higher initial drag would match the typical direct entry shape shown in Figure 6-2. Changing the reference trajectory for direct entries should reduce the controller's workload and improve the performance during the Final phase. This should reduce the amount of control saturation and increase the overall robustness of the algorithm.

6.4 Shallow Entry Test

Assuming the spacecraft has entered the Earth's atmosphere near its nominal orientation, the reentry guidance system must correct for any errors in the initial flight path angle. In the Apollo algorithm, this is accomplished by a single test after EI when the vehicle's drag reaches 1.6 ft/s^2, or 0.05 g's. Its purpose is to determine the vehicle's position in the entry

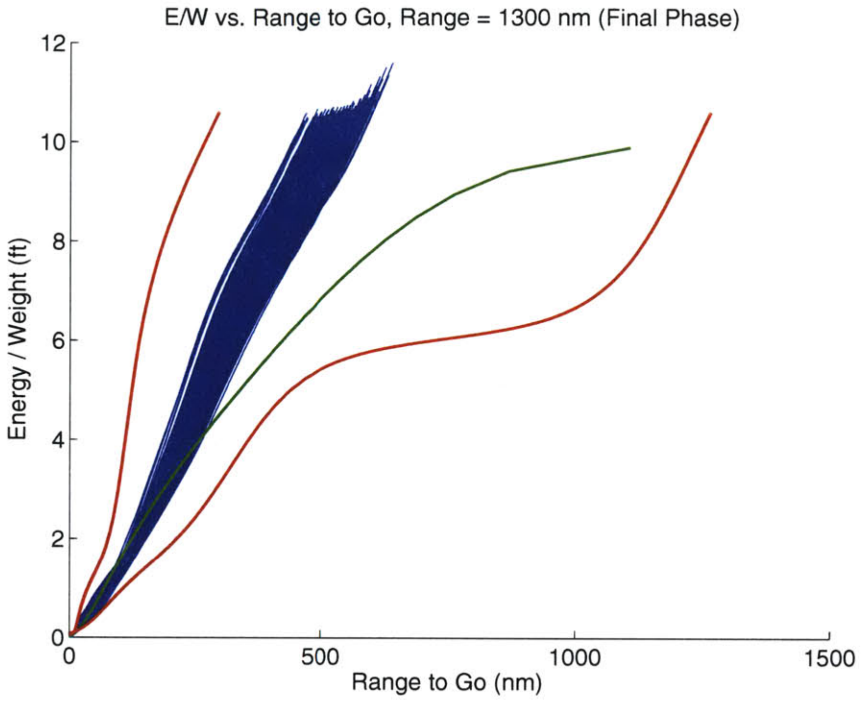


Figure 6-2: Draper baseline energy bucket for 1300 nm target range

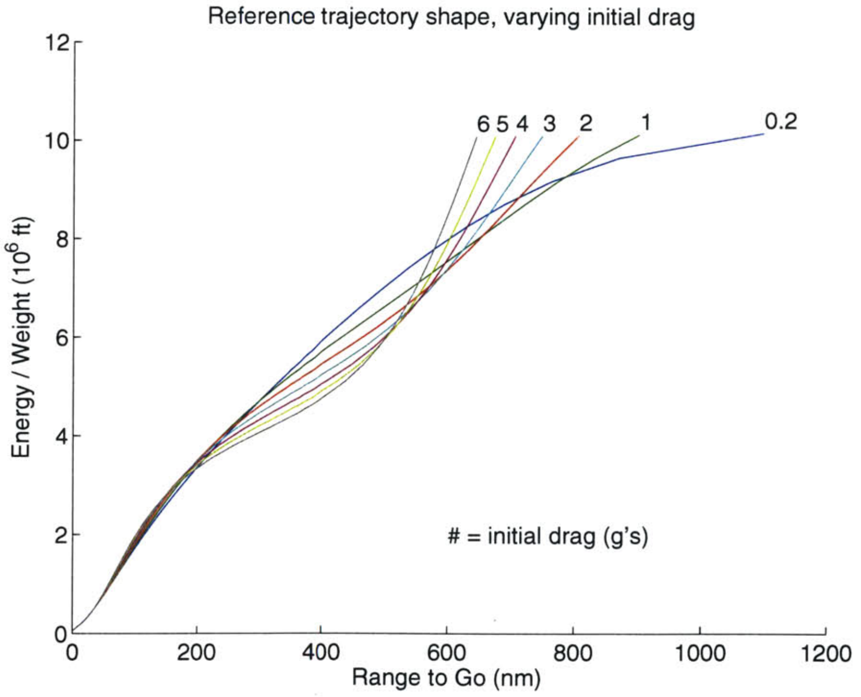


Figure 6-3: Reference trajectory shape change with initial drag

corridor, which is defined by flight path angle limits as a function of EI velocity. If the current velocity is greater than V_{up}, calculated using Equation 6.3, the entry is shallow and lift down is commanded. Otherwise, guidance commands lift up.

$$V_{up} = V_{final} - K_{44} (\sin \gamma)^3 \tag{6.3}$$

The values of V_{final} and K_{44} are $25,000$ ft/s and $44,389,312$ ft/s, respectively [9]. These empirical constants are derived from the Apollo overshoot and undershoot boundaries [5]. The overshoot boundary is defined by the shallowest flight path angle that achieves the minimum range. The undershoot boundary is constructed from a pair of limits:

1. The steepest flight path angle for which the maximum load will not exceed the limit
2. The steepest flight path angle for which the maximum range can be achieved

For Apollo 11, the minimum range requirement was 1285 nm, the maximum range requirement was 2500 nm, and the maximum load limit was 12 g's [5]. Apollo 11's resulting entry flight path angle boundaries, along with heating constraints, are shown in Figure 6-4. The V_{up} centerline is shown in the plot by the lift vector orientation line.

The Apollo shallow entry test is potentially unsuitable for the CEV configuration for a variety of reasons. First, the range requirements will almost certainly be different due to the constraint that the CEV land on land rather than in water, and the maximum load constraint may change. Second, the CEV capsule has different physical characteristics than the Apollo capsule, so it will not behave the same way as the Apollo capsule under the same guidance commands. Third, since the logic in the Draper PredGuid algorithm is different in several areas than the Apollo skip algorithm, it will shape the trajectory in a different manner. An inappropriate shallow entry test and resulting incorrect L/D command could cause two problems:

1. Commanding full lift up for a shallow entry increases the risk of the vehicle skipping out of the atmosphere
2. Commanding full lift down for a steep entry could cause the vehicle to exceed its maximum allowable g load constraint

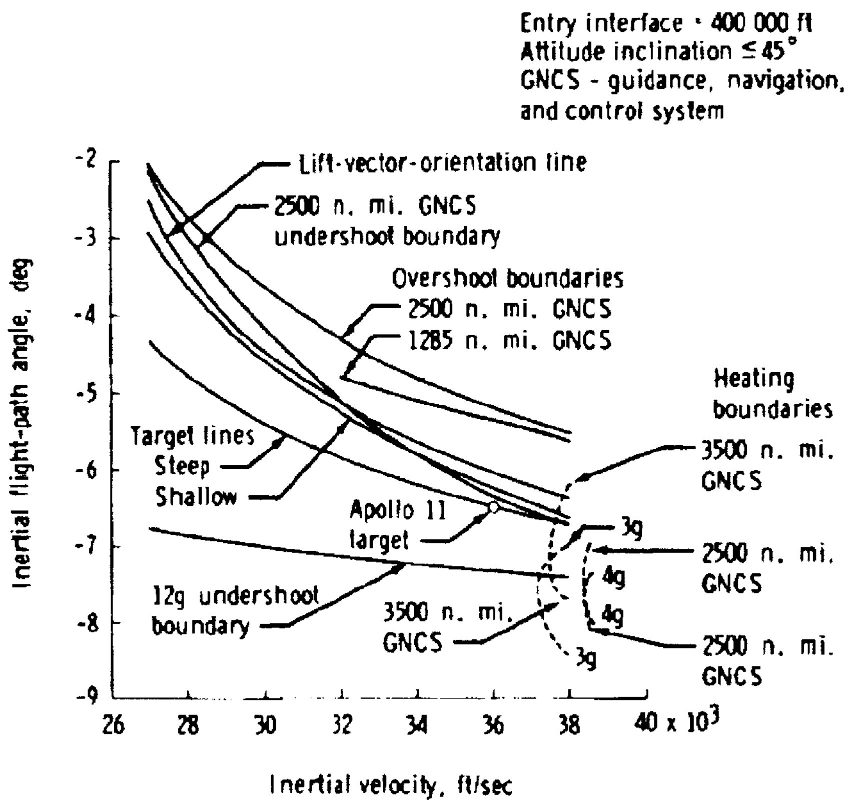


Figure 6-4: Apollo 11 entry corridor [5]

There are a few different approaches for updating this test for the CEV configuration. The first is to directly apply the Apollo version of the test, as is currently done in the Draper baseline algorithm. This assumes that the Apollo and CEV capsules have roughly the same L/D and ballistic coefficient; therefore, they have approximately the same entry corridor limits on velocity and flight path angle. This technique—using the Apollo test with the CEV configuration—results in no shallow entries for 1000 Monte Carlo trials. 100% of the CEV entries are commanded to fly lift up (see Figure 6-5).

Nearly three years after "Reentry Guidance for Apollo" was published, Phillip Moseley wrote "The Apollo Entry Guidance: A Review of the Mathematical Development and its Operational Characteristics," which contained updated values for the empirical constants in the V_{up} equation: $V_{final} = 26,600\ ft/s$ and $K_{44} = 19,749,550\ ft/s$ [10]. Using these new constants with the CEV yielded the following results for 1000 Monte Carlo trials: 93% of the entries were deemed shallow and were commanded to fly lift down; the remaining 7%

were not shallow and were commanded to fly lift up. This data is shown in Figure 6-5.

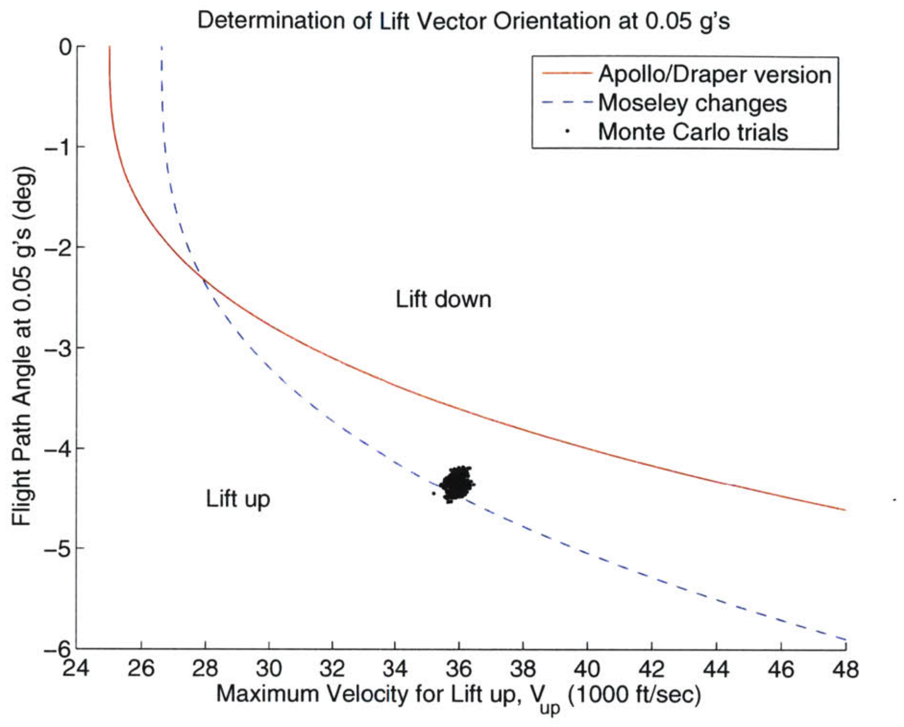


Figure 6-5: Orientation of lift vector at 0.05 g's

This presents some ambiguity in what the true entry corridor centerline is. Continuing to use Apollo's empirical constants from Reference [9] for the CEV could result in several shallow entries not receiving a lift down command. Furthermore, a comparison of the L/D and ballistic coefficient (B_c) values for the Apollo and CEV command modules (CM) reveals that they are not very similar, as shown in Table 6.1. Therefore, the two vehicles will not have the same entry corridor and should not use the same shallow entry test.

Table 6.1: L/D and ballistic coefficient values for the Apollo and CEV capsules

	Apollo CM	***CEV CM***
Average mass (lb_m)	388 [12]	528
Reference area (ft^2)	129 [12]	211
Hypersonic C_D	1.289 [5]	1.235
$B_c = \frac{m}{S_{ref}C_D}$ (lb_m/ft^2)	2.33	2.02
Hypersonic L/D	0.30	0.36

Another approach is to re-derive the overshoot and undershoot boundaries for the CEV configuration, which should yield new values for the empirical constants V_{final} and K_{44}. At this time, however, the derivation would be difficult because the CEV capsule is still in its design period, and requirements for loads and range are still in flux. All of the simulation data and analysis necessary to derive the boundaries would have to be recreated every time the design requirements or vehicle parameters changed.

A third and potentially more robust option is to develop a model-based shallow entry test which does not rely on empirical constants. This method is attractive because it would not require a redesign every time the requirements are updated, and it would be easily portable between vehicle configurations. This approach, however, would most likely require an increase in computing power and guidance logic complexity over a single-line test.

6.5 Determining Whether Skip is Necessary

As briefly discussed in Section 4.4, the Huntest phase determines whether a skip is necessary to reach the landing site by comparing its predicted atmospheric exit velocity (V_L) to a set limit of $18,000\ ft/s$. If V_L is below this limit, no skip is necessary and Huntest transitions directly to the Final phase.

Huntest calculates V_L using a series of empirical relationships and simplified analytic equations, and this is the only parameter used to determine whether a skip is necessary – Huntest performs no range prediction for direct entries. This became a problem for early versions of the Draper baseline algorithm. Results from the Draper algorithm with the CEV configuration showed some cases in which Huntest commanded a skip even though the target range was too short to require one. As a temporary solution, the exit velocity cutoff was raised to $20,000\ ft/s$. This improved the performance, but a more robust solution is desired.

Because computing has made such great advances since the 1960's, the accuracy of Huntest's calculations and predictions can be improved by updating the technology it uses. The decision to perform a skip can be based on more accurate models and a combination of parameters, not just velocity. Eliminating these empirical relationships and simplifications should make the entire algorithm more robust and more adaptable to different vehicle configurations.

6.6 Transitioning to PredGuid

As previously discussed in Chapter 5, the Draper baseline algorithm uses PredGuid to determine the bank angle commands during the Upcontrol and Ballistic phases. When the Draper algorithm was first created, there was some concern regarding how much of the original Apollo Upcontrol phase to keep and when to hand over control to PredGuid. Initially there were two versions of the algorithm, and they differed in the time that PredGuid took over during Upcontrol. In the "Low Loft" version, more of the original Upcontrol logic was kept and PredGuid started later in the phase; in the "High Loft" version (the baseline algorithm described in Chapter 5), less of the original Upcontrol was kept and PredGuid started as soon as Upcontrol began. The naming conventions came from the fact that turning PredGuid on earlier resulted in a steeper, higher altitude skip, while turning PredGuid on later gave a shallower, lower altitude skip. This is shown in Figure 6-6.

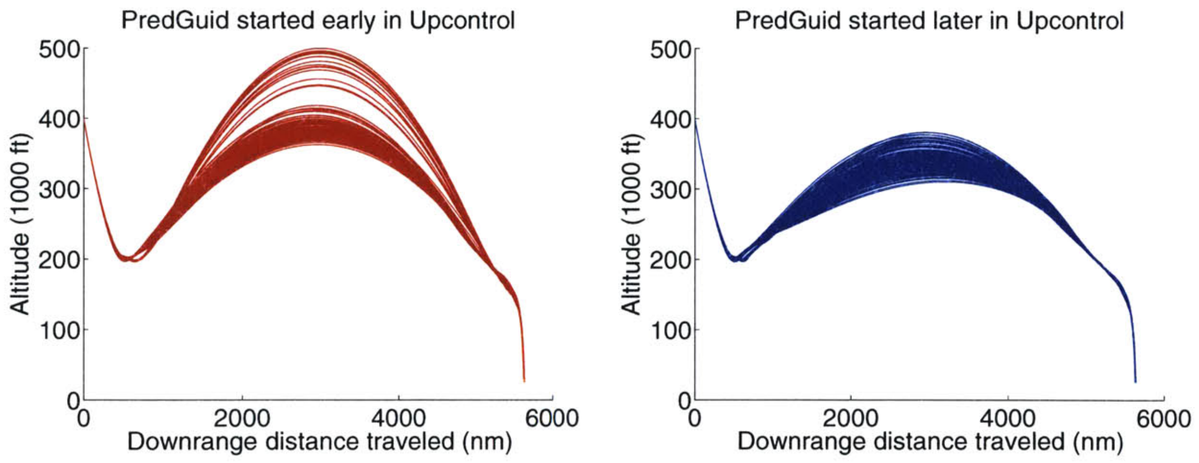


Figure 6-6: Trajectory shaping from PredGuid start time

The difference in the amount of lofting is due to the vehicle's remaining range to go at the time PredGuid starts. When PredGuid's target range is longer, it requires a smaller bank angle to reach the target, giving the vehicle more lift. Figure 6-7 shows the range to go at the start of PredGuid for each version of the Draper algorithm.

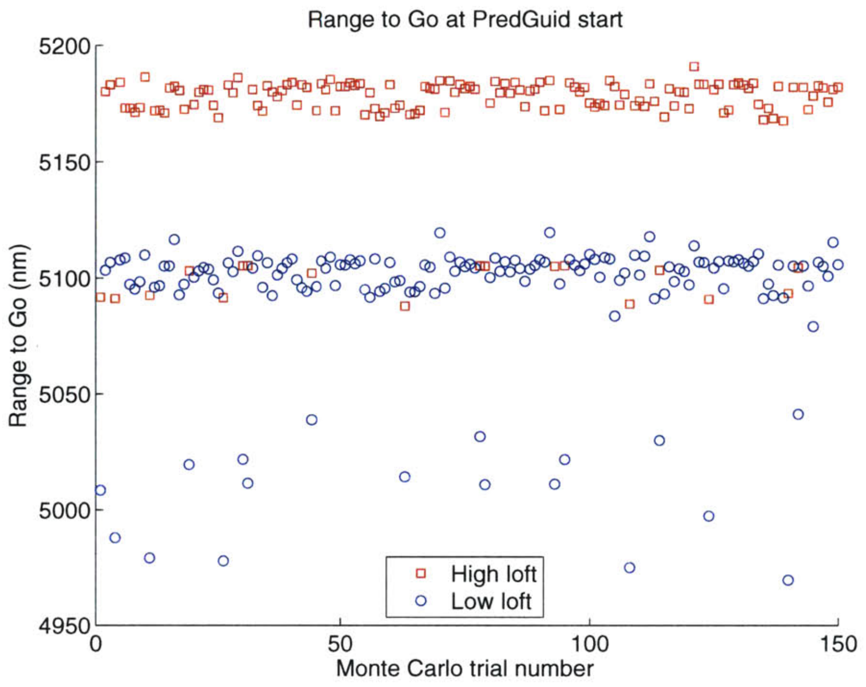


Figure 6-7: Range to go at PredGuid start time

While there is no appreciable difference in landing accuracy from the time PredGuid is started, there are other advantages and disadvantages of both trajectory shapes. A high loft trajectory, for example, has a lower total heat load because it spends less time in the atmosphere than a low loft trajectory. Table 6.2 summarizes the advantages and disadvantages of the low and high altitude skips.

These differences suggest that perhaps there is an optimal blend of the two trajectory shapes—a "best" time to transition to PredGuid. Determining this optimal point requires some knowledge of the desired trajectory characteristics, however. A trajectory that minimizes total flight time may not yield the same PredGuid transition time as a trajectory that minimizes total heat load, for example. Regardless of the specific objective function, an algorithm in which this type of tuning is possible would offer significantly more flexibility to a mission design team.

Table 6.2: Pros and cons of high and low altitude skip trajectories [2]

	Low Altitude Skip
PROS	Higher density $\rightarrow$ more aero control authority Shorter skip time in emergency scenario
CONS	Upper atmosphere highly uncertain and variable $\rightarrow$ potential for sudden loss of control authority

	High Altitude Skip
PROS	Additional time to navigate Less atmospheric uncertainty Smaller disturbing aerodynamic forces Greater heat dissipation, lower total heat load
CONS	Places vehicle in flight regime with no aerodynamic control authority

Chapter 7

EMT Predictor Algorithm

7.1 Overview

The Predictor entry guidance algorithm was developed to solve the problems discussed in Chapter 6 and it is described in this chapter. The Energy Management and Transition (EMT) Predictor algorithm adds critical energy management and phase transition logic to the Draper baseline algorithm described in Chapter 5, with significant alterations to the five major phases. The most important of these modifications is the change of control between Huntest and Initial Roll. In the Apollo algorithm, Huntest held the bulk of the decision-making authority, but Initial Roll now has that authority in the Predictor algorithm. Most importantly, Initial Roll determines whether a skip is necessary to meet the range requirement. If a skip is not required, Initial Roll transitions to Huntest to continue the direct entry logic. If the trajectory does require a skip, Initial Roll transitions directly to Upcontrol (instead of Huntest) to continue the skip logic. Regardless of whether the algorithm follows the direct entry path or the skip path, it always ends with the Final phase. The separation of phases used for direct entry and for skip is shown in Figure 7-1.

The Predictor algorithm also introduces the concept of a "loft" trajectory, which is a low altitude, short range skip. In terms of the guidance logic, this is a trajectory in which Upcontrol transitions directly to Final and does not enter the Ballistic phase. This means that the drag never drops below 0.2 g's during Upcontrol. The loft regime bridges the gap between

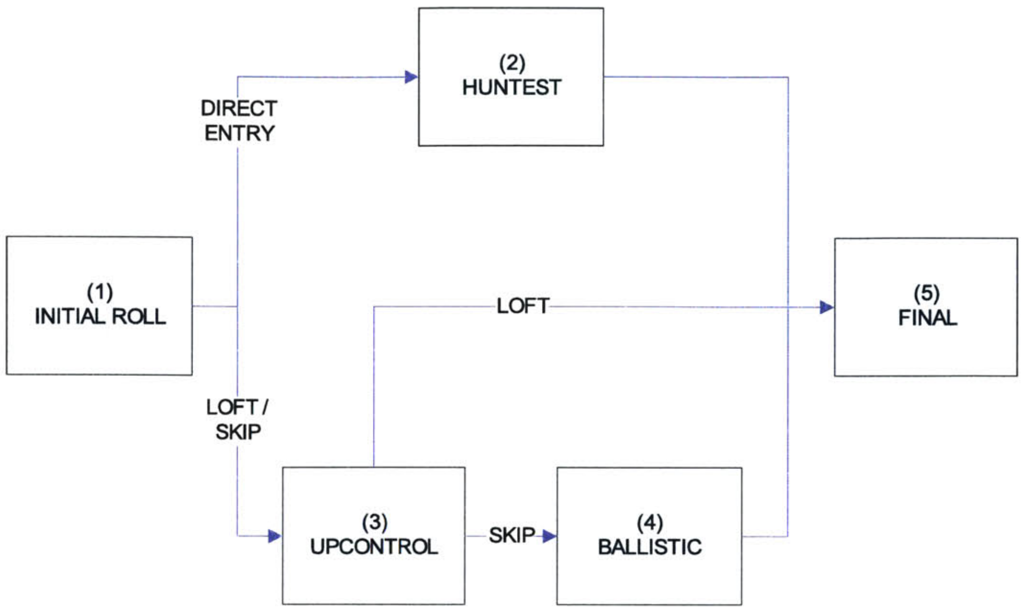


Figure 7-1: Predictor algorithm phases

direct entries and skips, and its development required slight modifications to PredGuid. The Draper baseline's versions of Upcontrol and Ballistic with PredGuid, however, remain unchanged.

Because Initial Roll makes the bulk of the major decisions in the Predictor algorithm, Huntest is reduced to a "wrapper" for Constant Drag, waiting until the conditions are met to transition to the Final phase. The Predictor algorithm allows for more than one Constant Drag value, based on the desired energy change over the range to be covered in Huntest. Initial Roll determines the required Constant Drag level.

Final's logic is unchanged from the Draper baseline version, but it uses a different reference trajectory for direct entry cases. Skip and loft trajectories use the original skip reference, but direct entries select an appropriate Final phase reference in flight from several stored trajectories. This is done during Initial Roll, and the reference trajectory is chosen to match the Constant Drag level.

This chapter first presents the design philosophy and objectives in designing the Predictor algorithm. The next section covers improvements to the direct entry logic, and the third section discusses changes to the skip/loft logic. Finally, Initial Roll's EMT Predictor is introduced.

7.2 Design Philosophy and Objectives

One major goal in revising the guidance algorithm was to match the typical final vehicle state of each phase to the desired initial state of the next phase. This makes the transitions between the phases smoother, reduces the controller's workload to modify the state, and makes the behavior of the trajectory easier to predict and manage. This is more important for direct entries than for skips or lofts because the total trajectory range is so short. There is very little time to compensate for incorrect or suboptimal bank commands. Because of this, it is important that the Initial Roll, Huntest, and Final phases line up correctly and that no large state changes are required in the transitions from one phase to the next.

Another goal was to determine the trajectory type—direct entry, loft, or skip—as early in the algorithm as possible. This allows the algorithm more time to make accurate decisions about how to manage the spacecraft's energy and when to transition to the next phase. This was accomplished by moving the trajectory type decision from Huntest to Initial Roll, the first phase. Specifically, Initial Roll is responsible for determining the following information:

- Is a skip necessary to reach the target?
  - If yes
    - Determine appropriate time to transition to PredGuid
    - Determine constant skip bank angle that will reach the target
  - If no
    - Determine appropriate Constant Drag value
    - Determine L/D fraction that will reach Constant Drag value

In addition to improving the problem areas discussed in Chapter 6, the overarching objective in designing the Predictor algorithm is to reduce the dependency on empirical equations, heuristics, and Apollo-specific data. Because of the greater computational power available, the Predictor algorithm uses fewer simplifications and more model-based predictions.

7.3 Direct Entry

The addition of the EMT Predictor enhancements to the Draper baseline algorithm results in significant changes to the direct entry logic. The main elements are the variable Constant

Drag policy and the redesigned Final phase reference trajectories. The variable Constant Drag policy requires changes in the Initial Roll L/D command and uses a new controller to facilitate the transition from Initial Roll to Huntest.

7.3.1 Variable Constant Drag Level

Several changes have been made to the direct entry logic. The first is that the Constant Drag value (D_0) is allowed to vary for each trajectory; it is no longer 130 ft/s^2 for every entry. Initial Roll determines the correct value of D_0 based on the required energy reduction over the available range. D_0 is held fixed once Initial Roll transitions to Huntest.

The idea behind the calculation for D_0 is to assume a constant acceleration trajectory during Constant Drag and then integrate acceleration twice to get the range traveled during the phase. If the range traveled is too long, D_0 must be increased so the vehicle decelerates faster, and if the range is too short, D_0 should be decreased to decelerate more slowly.

Rather than actually performing the integration, a few assumptions can be made to get an analytic equation for the range traveled as a function of the Constant Drag level. First, start with the simplified relationship between velocity, range, and Earth radius from the planar equations of motion for a point mass (Equation 2.44):

$$V\cos\gamma = -\dot{\theta}R_E$$

Assume shallow flight path angles ($\cos\gamma \approx 1$) and recall that Equation 2.44 assumes counterclockwise displacement in θ is positive. For this application, positive range traveled is in the direction opposite a traditional right-handed coordinate system, so the negative sign is not required.

$$V = \dot{\theta}R_E \tag{7.1}$$

Also recall from the simplified equations of motion for a point mass that $\dot{V} = -D$. Divide both sides of Equation 7.1 by dV/dt and simplify:

$$\frac{V}{dV/dt} = \frac{R_E\dot{\theta}}{dV/dt} \tag{7.2}$$

$$-\frac{V}{D} = R_E \frac{d\theta}{dt}\frac{dt}{dV} \tag{7.3}$$
$$-V dV = R_E D\, d\theta \tag{7.4}$$

Integrate Equation 7.4 to get the range traveled during the Constant Drag phase, as a function of the drag level:

$$\theta = \frac{V_1{}^2 - V_2{}^2}{2 R_E D_0} \tag{7.5}$$

where:

D_0	=	Constant Drag level
V_1	=	velocity at start of Constant Drag
V_2	=	velocity at end of Constant Drag
θ	=	range traveled during Constant Drag
R_E	=	Earth radius

Allowing different Constant Drag levels gives the algorithm flexibility to adjust the amount of energy depleted over a particular range, which addresses the energy management problem presented in Section 6.2. This improves the algorithm's ability to target the energy and range conditions at the start of the Final phase reference trajectory. Figure 7-2 shows the Constant Drag levels the Predictor algorithm selected for each direct entry target range, in order to meet approximately the same energy level at the start of the Final phase.

In an ideal scenario, the appropriate Constant Drag level could be calculated analytically at the start of Huntest by rearranging Equation 7.5 and assigning the input parameters in the following manner:

$$D_0 = \frac{V_1{}^2 - V_2{}^2}{2 R_E \theta} \tag{7.6}$$

V_1	=	velocity at start of Huntest
V_2	=	desired velocity at end of Huntest (velocity at start of Final phase reference trajectory)
θ	=	range to go from start of Huntest to start of Final phase reference trajectory

In reality, the process is not quite that simple. Consider the following example: the vehicle flies full lift up throughout Initial Roll and reaches a drag of 4 g's at the transition to

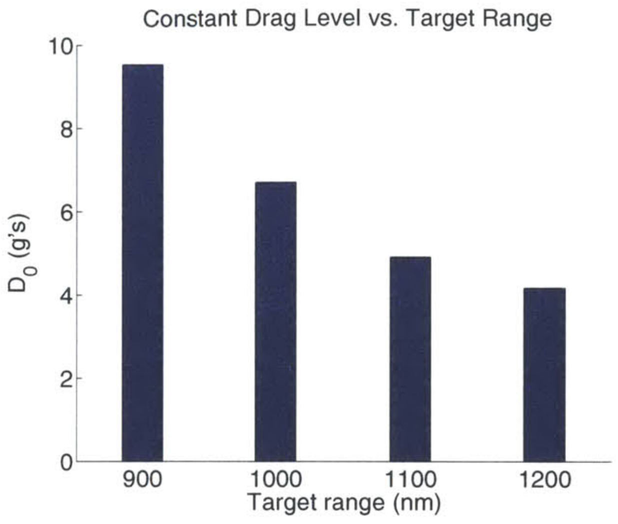


Figure 7-2: Constant Drag levels for direct entry target ranges

Huntest. Using Equation 7.6, guidance calculates that the drag level necessary to deplete the required amount of energy over the remaining range is 5 g's. Because the required Constant Drag level is different than the current drag, the vehicle will have to spend some amount of time and range increasing the drag to the desired value. By the time the drag reaches 5 g's, the Constant Drag level will be too low to deplete enough energy over the shortened range to go.

This method assumes that the drag level does not vary during Constant Drag. Although the drag level will never be completely steady throughout Huntest, the accuracy of the range prediction drops as the Constant Drag value drifts from or oscillates around its desired value. Starting Constant Drag with the current drag and altitude rate near their desired values will reduce the amplitude of the drag oscillations, reduce fluctuations in the energy decay rate, and improve the accuracy of the range prediction.

For this variable Constant Drag approach to be feasible, the goal of Initial Roll for direct entries must be to guide the vehicle to the required drag and altitude rate for Constant Drag. This can be accomplished by changing the L/D fraction during Initial Roll to achieve different drag levels, which is discussed in the next section.

7.3.2 Initial Roll L/D

In the Predictor algorithm, Initial Roll's constant L/D fraction for direct entries is dictated by the required Constant Drag level. To ease the Constant Drag controller's workload, Initial Roll's objective is to get the vehicle as close as possible to the required drag and altitude rate by the end of the phase. This should reduce the amplitude of the oscillations during Constant Drag and yield a steadier energy decay rate.

Different Initial Roll L/D fractions change the shape of the early trajectory, and this impact increases as the duration of the phase increases. A high L/D fraction (near +1) arrests the steep initial descent rate and results in a low drag. A low L/D fraction (near −1) does not arrest the steep initial descent rate and results in a very high drag. Different constant L/D fractions, ranging from −1 to +1 in increments of 0.1, were flown during the first 700 *nm* of the entry to create the plots shown in Figures 7-3 to 7-5. Figure 7-3 shows that different L/D fractions can be flown to reach different drag levels. Flying full lift up, for instance, can reach drag levels up to approximately 3.5 g's. There are often several different L/D fractions that will reach a particular Constant Drag level; the difference is how soon that drag level is achieved. Figures 7-4 and 7-5 show the corresponding altitude rate and flight path angle behavior for different Initial Roll L/D fractions.

The question of which L/D fraction to fly in order to reach a particular drag value can be answered by examining the altitude rate behavior. The Constant Drag controller generates L/D commands based on errors in drag and altitude rate, where the reference altitude rate for Constant Drag is given by the following relationship:

$$\dot{h}_0 = \frac{-2h_s D_0}{V}$$

Because the goal of Initial Roll is to guide the vehicle to the desired Huntest/Constant Drag initial conditions (drag and $\dot{h}$), the appropriate Initial Roll L/D fraction is the one in which the vehicle achieves the correct altitude rate at the same time as the correct drag.

As an example, assume the vehicle flies an L/D fraction of 0.5 during Initial Roll. Assume also that the transition to Huntest could occur at any time and that in each guidance cycle, the Constant Drag value will be set to the current drag. Therefore, the Constant Drag

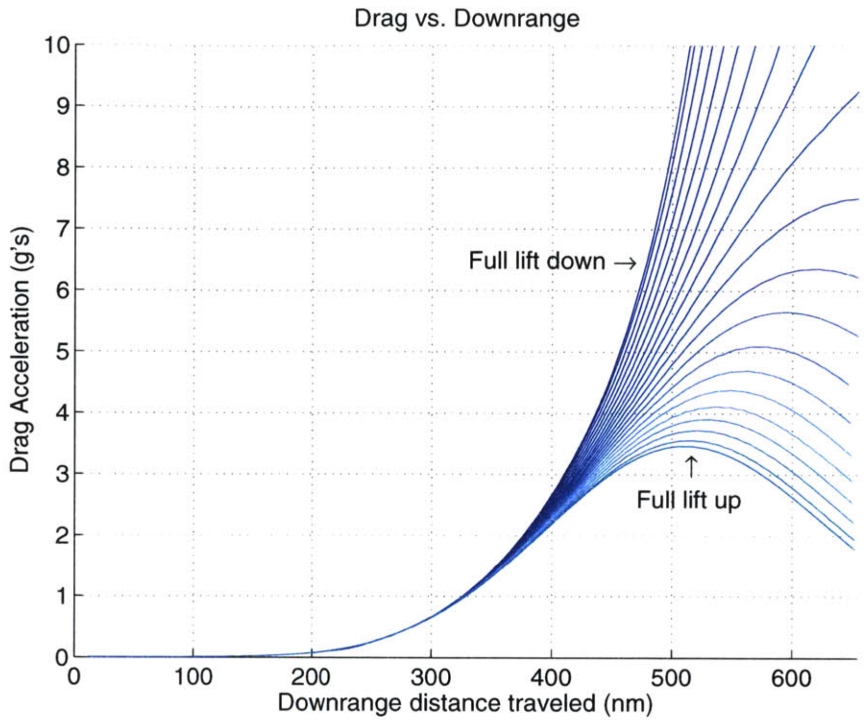


Figure 7-3: Drag for various Initial Roll L/D fractions

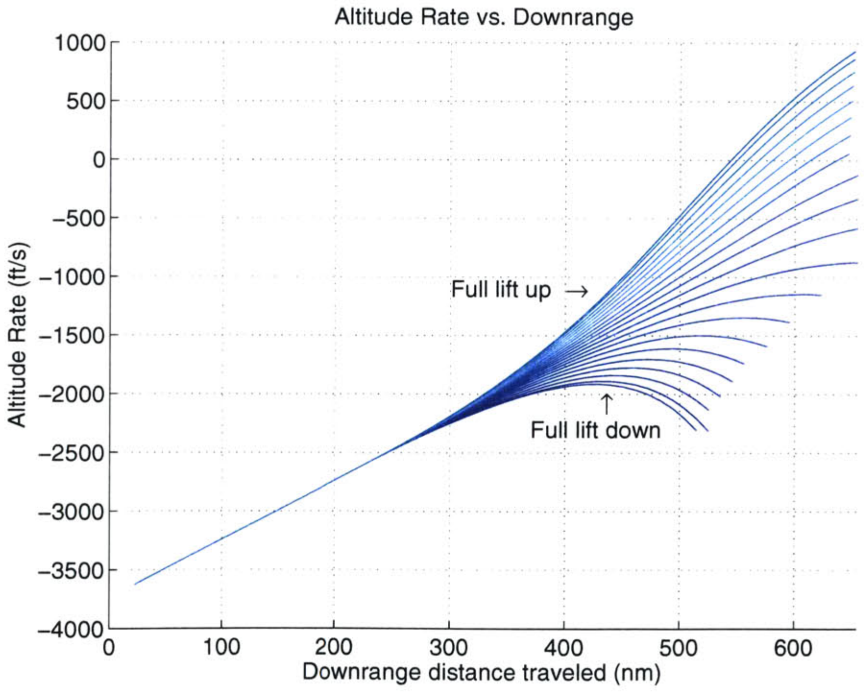


Figure 7-4: Altitude rate for various Initial Roll L/D fractions

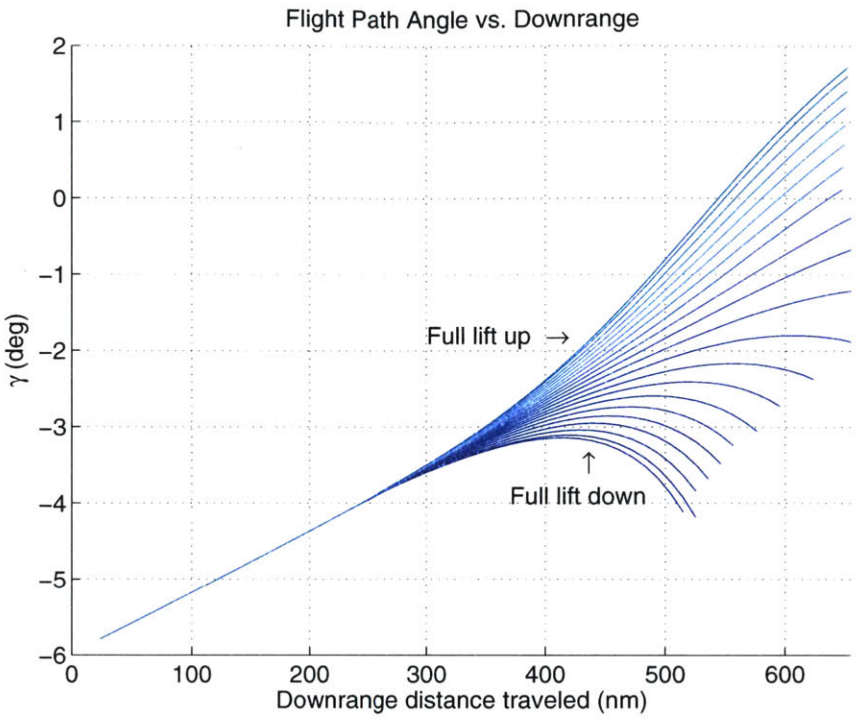


Figure 7-5: Flight path angle for various Initial Roll L/D fractions

reference altitude rate changes each guidance cycle with the current drag: $\dot{h}_0\left(t\right) = \frac{-2h_s D(t)}{V}$. Since the drag error from the Constant Drag reference is zero at each guidance cycle, the transition time will be determined by the point that yields the smallest altitude rate error. Figure 7-6 demonstrates that this happens to be the time at which the L/D fraction achieves its maximum drag. In fact, the minimum altitude rate error occurs at the maximum drag point for all positive L/D fractions (see Figure 7-7).

The results from Figures 7-6 and 7-7 imply that if the desired Constant Drag level were the maximum drag achievable for that L/D fraction, the appropriate time to transition to Huntest would be the time that maximum drag occurs. Using this logic, each L/D fraction in Initial Roll will yield a unique maximum drag, so changing the Initial Roll L/D fraction changes Constant Drag's D_0.

Using this approach, the lowest Constant Drag level for this configuration is approximately 3.5 g's, and it is achieved by flying full lift up during Initial Roll. Intermediate Constant Drag levels can be reached by decreasing the Initial Roll L/D fraction. Negative L/D fractions can achieve extremely high drag levels, but the Constant Drag value is limited

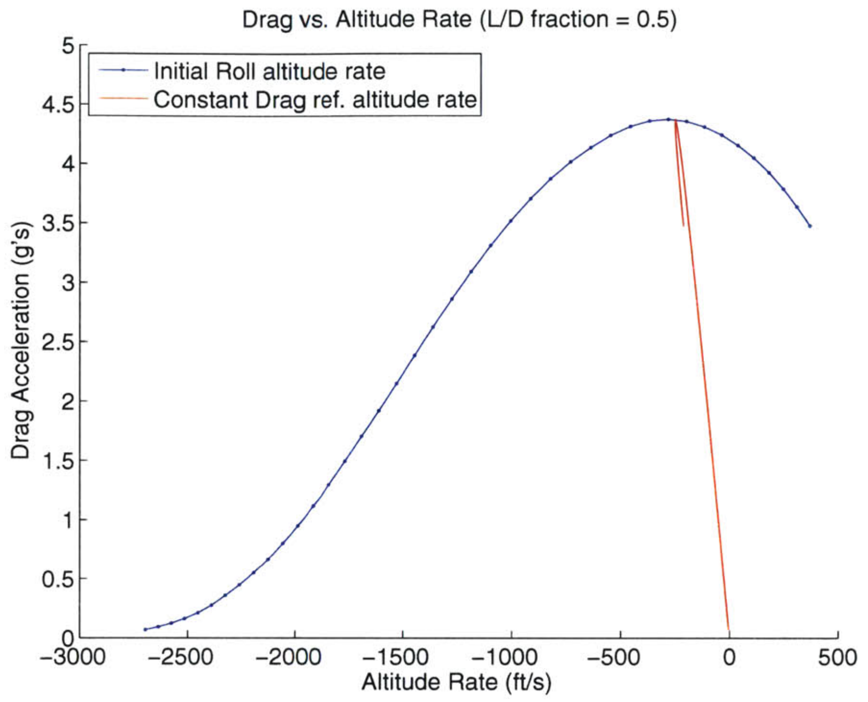


Figure 7-6: Initial Roll $\dot{h}$ and Constant Drag reference $\dot{h}$ for 0.5 L/D fraction

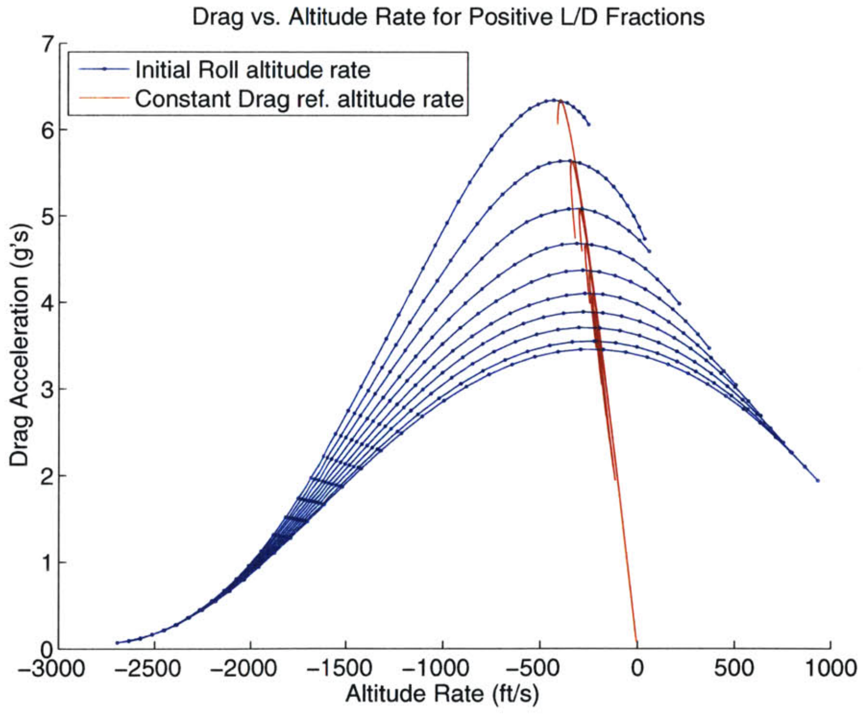


Figure 7-7: Initial Roll $\dot{h}$ and Constant Drag reference $\dot{h}$ for all positive L/D fractions

to less than 10 g's to avoid exceeding maximum load limits.

7.3.3 Gamma Turn Controller

Unfortunately, making the transition from Initial Roll to Huntest precisely at the time that maximum drag is achieved does not eliminate oscillations during Constant Drag. This is because while the altitude rate and flight path angle pass through their respective Constant Drag reference values at the transition time, they do not have the correct rates to maintain those values. An example of this is shown in Figure 7-8, where the blue lines are the flight path angles for different Initial Roll L/D fractions and the nearly horizontal lines are the Constant Drag reference flight path angles for different drag levels.

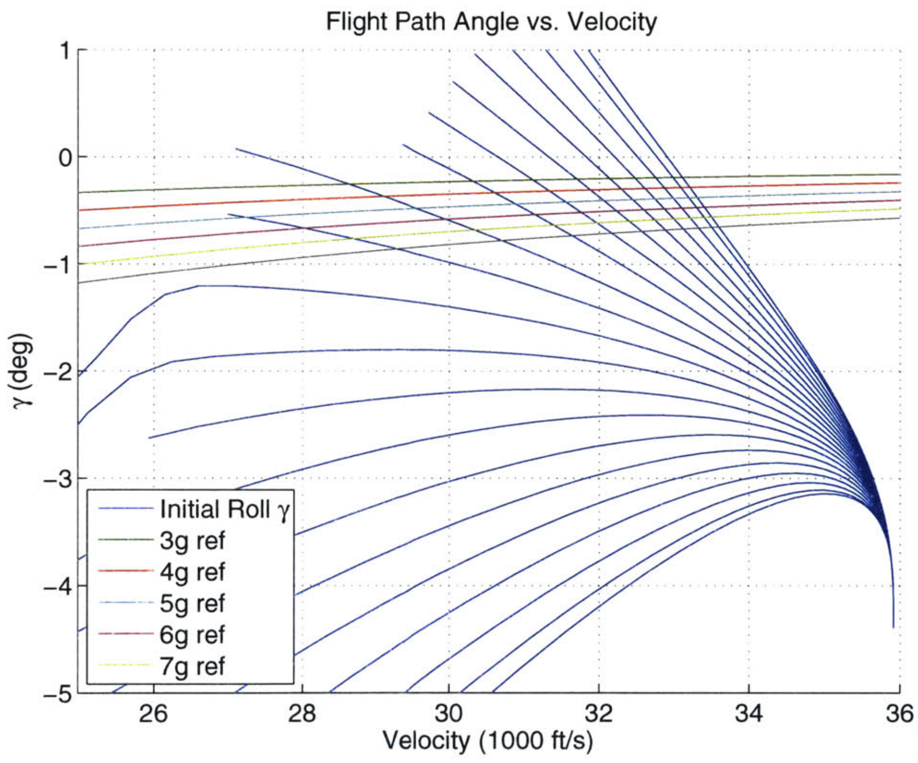


Figure 7-8: Initial Roll flight path angles and Constant Drag references

A solution to this problem is to add a controller to "turn" the flight path angle rate from its Initial Roll value to the desired Constant Drag value, just before the transition to Huntest. The necessary control law can be derived from the planar equations of motion for a point mass. Start with Equation 2.55, remembering that lift and drag are expressed here as forces per unit mass:

$$V\dot{\gamma} = L + g\left(\frac{V^2}{gR_E} - 1\right)$$

Substitute $L = \frac{L}{D} \cdot D$ and rearrange the above equation to get the Gamma Turn Controller's L/D command as a function of the desired Constant Drag flight path angle rate, $\dot{\gamma}_0$:

$$\left(\frac{L}{D}\right)_{turn} = \frac{V\dot{\gamma}_0 - g\left(\frac{V^2}{gR_E} - 1\right)}{D} \tag{7.7}$$

During each guidance cycle, all the parameters on the right hand side of Equation 7.7 are known except $gamma_0$. This can be determined from the reference altitude rate for Constant Drag, γ_0, which is given by Equation 4.8. Since flight path angle is related to altitude rate by $\dot{h} = V\gamma$ (assuming small angles), the reference flight path angle for Constant Drag is:

$$\gamma_0 = \frac{-2h_s D_0}{V^2} \tag{7.8}$$

Assuming h_s and D_0 are constants, the reference flight path angle rate can be determined by taking the derivative of Equation 7.8:

$$\begin{aligned}
\dot{\gamma}_0 &= \frac{\partial \gamma_0}{\partial V} \cdot \frac{\partial V}{\partial t} && (7.9)\\
&= \frac{4h_s D_0}{V^3} \cdot (-D_0) && (7.10)\\
\dot{\gamma}_0 &= \frac{-4h_s {D_0}^2}{V^3} && (7.11)
\end{aligned}$$

Using this equation for the flight path angle rate, the Gamma Turn Controller's L/D command in Equation 7.7 can be rewritten as an explicit function of the Constant Drag level, D_0:

$$\left(\frac{L}{D}\right)_{turn} = \frac{\frac{-4h_s {D_0}^2}{V^2} - g\left(\frac{V^2}{gR_E} - 1\right)}{D} \tag{7.12}$$

If the resulting L/D command from the Gamma Turn Controller is larger than the vehicle's total L/D, it is limited to the maximum L/D. The L/D command is further limited if the difference between the current L/D and the command results in a bank angle difference of more than 20°. This is a conservative estimate of the flight control system's maximum bank angle rate, and the limit is intended to avoid saturating the flight control system.

First, the difference between the current and desired bank angles is computed to determine whether the limit is necessary:

$$\phi_{des} = \operatorname{sign}(\phi_{current}) \cdot \operatorname{acos}\left[\frac{(L/D)_{turn}}{(L/D)_{max}}\right] \tag{7.13}$$

$$\Delta\phi = \phi_{des} - \phi_{current} \tag{7.14}$$

If this difference is greater than 20°, the L/D command is computed using Equation 7.15, which is simply the product of the vehicle's maximum L/D and the cosine of the bank angle difference. If the bank angle difference is less than 20°, the L/D command is calculated using Equation 7.12.

$$\left(\frac{L}{D}\right)_{cmd} = \left(\frac{L}{D}\right)_{max} \cos\left[\phi_{current} + \operatorname{sign}(\Delta\phi) \cdot (20^\circ)\right] \tag{7.15}$$

The Gamma Turn Controller is run during the last few guidance cycles before the transition to Huntest. During this time, the controller monitors the flight path angle error from the Constant Drag value ($\Delta\gamma$) and the flight path angle rate error from the Constant Drag value ($\Delta\dot{\gamma}$). The Gamma Turn Controller runs until any of the following conditions are met, and at that time, Initial Roll transitions to Huntest.

a. $|\Delta\gamma|$ and $|\Delta\dot{\gamma}|$ smaller than tolerances

b. Sign of $\Delta\gamma$ has changed from previous guidance cycle

c. Current drag exceeds maximum allowable

d. Maximum number of allowed guidance cycles reached

Figures 7-9 and 7-10 shows how the Gamma Turn Controller successfully modifies $\dot{\gamma}$ before transitioning to Huntest. This reduces the oscillations during Huntest and Constant Drag, and completes the variable Constant Drag approach.

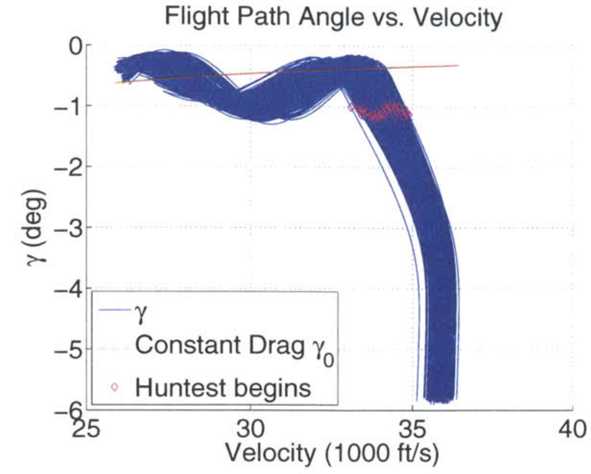


Figure 7-9: Without γ_{turn} Controller

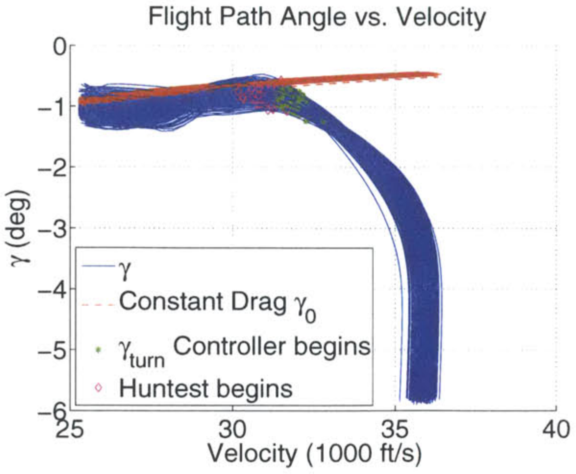


Figure 7-10: With γ_{turn} Controller

7.3.4 Final Phase Reference Trajectory

The Final phase reference trajectory has been redesigned for direct entry cases with the intent of matching the typical Huntest exit conditions. The method used to create the skip reference trajectory [2] was also used to create the direct entry reference trajectories and their corresponding gains.

As the target range increases for direct entries, the required Constant Drag level decreases; therefore, the Final phase initial drag decreases. To accommodate this variation, several reference trajectories were created in order to match different Final phase initial conditions. These references are created and stored pre-flight, and the Predictor algorithm can select the best reference trajectory during the flight. The selection is made during the Initial Roll phase, and it is done for direct entry cases only. All skip and loft cases use the original Draper PredGuid reference.

The initial conditions for the Final phase references are listed in Table 7.1, and the trajectories are plotted in Figure 7-11. The direct entry reference trajectories in Figure 7-11 are in two clusters due to their different initial flight path angles. The next section describes the origin of the initial conditions for these reference trajectories.

Table 7.1: Final phase reference trajectory initial conditions

Description	D_i *(g's)*	γ_i *(deg)*	V_i (ft/s)	*L/D fraction*
Skip/loft reference	0.2	-1	25,262	0.6
$3g$ reference	3	-0.5	25,262	0.2
$3.5g$ reference	3.5	-0.5	25,262	0.2
$4g$ reference	4	-0.5	25,262	0.2
$5g$ reference	5	-0.5	25,262	0.2
$6g$ reference	6	-1	25,262	0.2
$7g$ reference	7	-1	25,262	0.2
$8g$ reference	8	-1	25,262	0.2
$9g$ reference	9	-1	25,262	0.2
$10g$ reference	10	-1	25,262	0.2

Reference Trajectory Generation

Section 4.5.1 describes the constant L/D fraction and the three initial conditions that are required to generate the Final phase reference trajectory. Drag was chosen as the primary distinguishing feature between the different direct entry references so that it would match the Constant Drag levels coming out of Huntest. Due to the altitude rate and flight path angle associated with a particular Constant Drag level, the reference trajectory's initial flight path angle was changed with drag to accommodate this variation. Figure 7-12 shows how the Constant Drag reference flight path angle changes with velocity and drag. For simplicity, the initial flight path angles were rounded to the nearest half degree from their values at the initial velocity of the reference trajectory.

The direct entry reference trajectories use the same initial velocity as the Draper baseline reference used for skip and loft trajectories. This was done to keep the initial energy level for the Final phase relatively constant between trajectory types. The direct entry references do, however, use a different L/D fraction than the Draper baseline reference. Starting from the typical Final phase initial conditions for a direct entry, if the vehicle flies a constant L/D fraction of 0.6 throughout the Final phase, its trajectory will actually loft. While this phenomenon is not inherently bad, it does violate the defining characteristic of a direct entry, which is continuously decreasing altitude. Furthermore, the lofting associated with the 0.6 L/D fraction increases the Final phase range, which limits the short range capability of direct entry trajectories. To eliminate lofting and increase the short range capability,

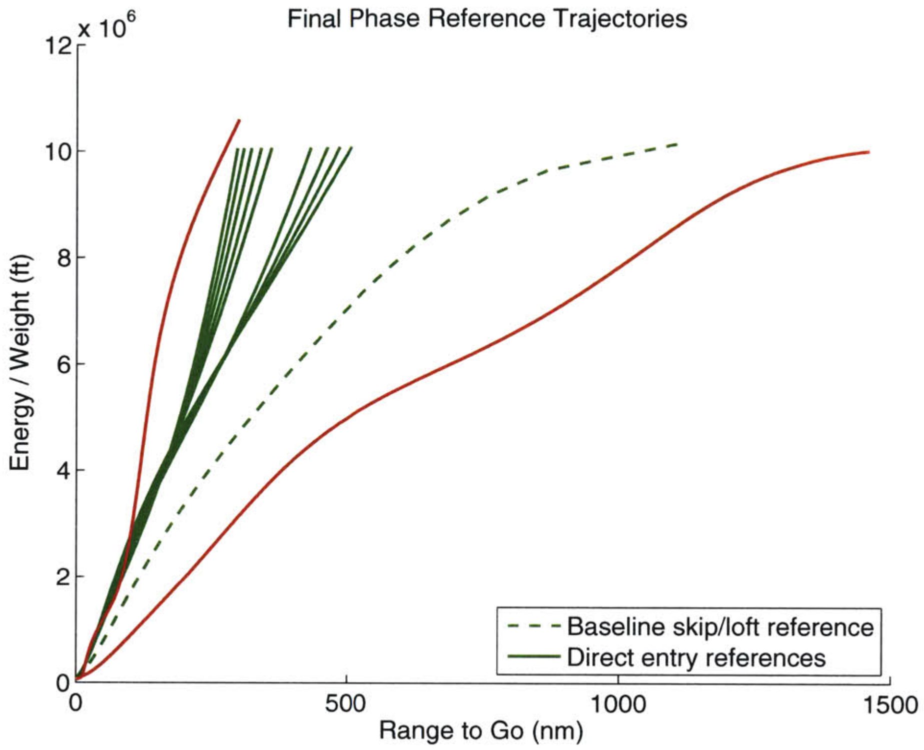


Figure 7-11: Final phase reference trajectories for direct entry and skip/loft

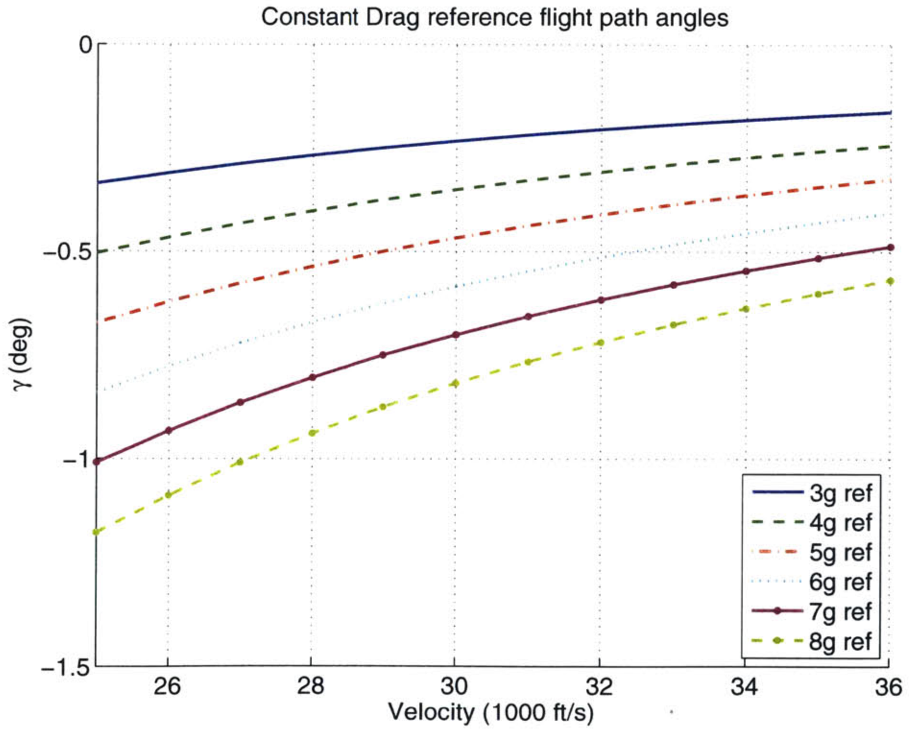


Figure 7-12: Reference flight path angles for Constant Drag

an L/D fraction of 0.2 was chosen as the center of the ranging capability for direct entry trajectories.

By eliminating the Final phase lofting, the bounds of the direct entry energy bucket are significantly smaller than for the loft or skip energy bucket. The left side of the bucket is still bounded by the full lift down trajectory, but the right side of the bucket is no longer bounded by the full lift up trajectory. The bucket is bounded on the right by a trajectory whose L/D fraction is less than 1, where the actual value depends on the initial conditions. Figure 7-13 shows a comparison of the new energy bucket for a direct entry and the original energy bucket that allows lofting. This slimmer energy bucket further illustrates the smaller margin for error for direct entries, and thus, the importance of accurately targeting the desired Final phase initial conditions.

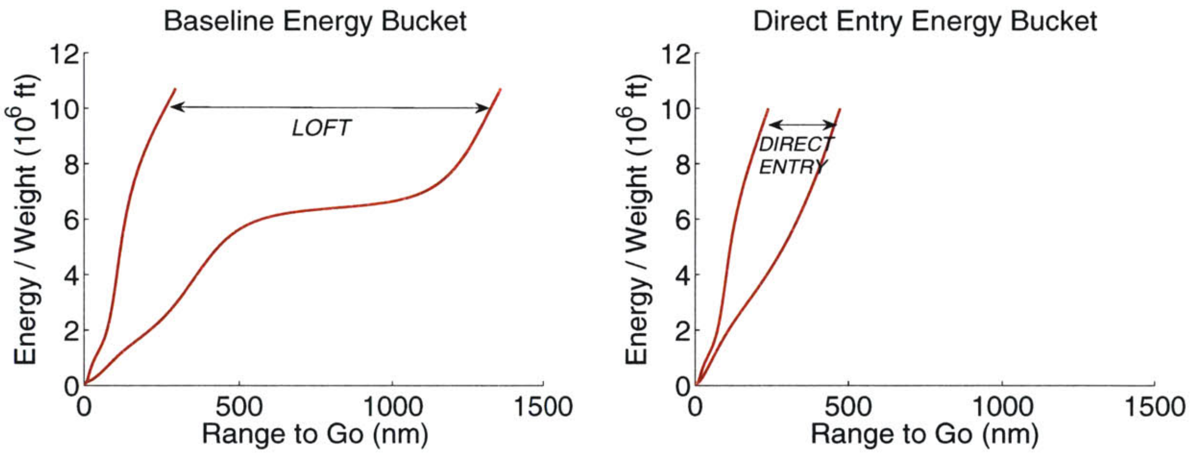


Figure 7-13: Reduction in Final phase energy bucket width for direct entry

Reference Trajectory Selection

For direct entries, the Predictor algorithm has the ability to dynamically select the best Final phase reference trajectory during the early portion of the entry, rather than having the reference trajectory pre-selected. The selection logic resides in the Targeting phase because several of the following phases use the reference trajectory for targeting and estimation. Although the selection logic is in Targeting, the reference trajectory can only be updated while the entry is in the Initial Roll portion of the algorithm. Once the algorithm transitions to either Huntest for direct entries or Upcontrol for skip/loft entries, the reference trajectory is fixed for the remainder of the entry.

The inputs to the reference trajectory selection subroutine are the predicted conditions at the start of Final. Using the Final Phase range estimation logic described in Section 4.5.1, the selector uses the predicted Final phase initial velocity and interpolates in each reference trajectory's lookup table to determine the corrected range to go for that velocity. The selector then subtracts the nominal range from the corrected range to get a range error, which is just the sum of the drag and altitude rate errors multiplied by their respective gains. The selector then chooses the reference trajectory that gives the smallest range error for the current predicted velocity. The selector's logic is summarized in Figure 7-14.

```
Initialize
for i = 1 : (number of reference trajectories)
    Interpolate in reference i's lookup table using predicted Final velocity
    Range error = corrected range – nominal range
    if range error < current smallest range error
        current smallest range error = range error
        ref_to_use = i
    end
end
Load reference ref_to_use
```

Figure 7-14: Reference trajectory selector pseudocode

7.4 Skip/Loft

The difference between loft and skip trajectories is slight, but it is an important distinction. A skip trajectory exits the atmosphere during a portion of the trajectory—that is, the aerodynamic drag drops below 0.2 g's. During this portion of the skip, the vehicle's control authority is greatly reduced. The air density is so low that changing the direction of the lift vector does not significantly affect its trajectory.

A loft trajectory, on the other hand, never exits the atmosphere. It has a lower apogee altitude than a skip and typically covers ranges that are too long for direct entry and too short for a skip. Because a loft never exits the atmosphere, it maintains control authority throughout the trajectory. In terms of the algorithm logic, a loft trajectory transitions from Upcontrol directly to Final, whereas a skip trajectory transitions from Upcontrol to Ballistic to Final.

The Apollo algorithm creates short range loft trajectories in an entirely different manner. The direct entry downrange capability was stretched to create a loft just after the transition to the Final phase. Using Final phase reference trajectory designed for a long range skip results in a drag mismatch for direct entry cases at the beginning of the phase. Because of this, the Final phase controller lofts the trajectory in an attempt to reduce the drag to the reference trajectory value. This type of lofting is shown in Figure 7-15. It is undesirable because it is not a planned loft; it is simply the result of a large predicted range error at the beginning of the Final phase. As previously discussed, these large errors can also lead to control saturation.

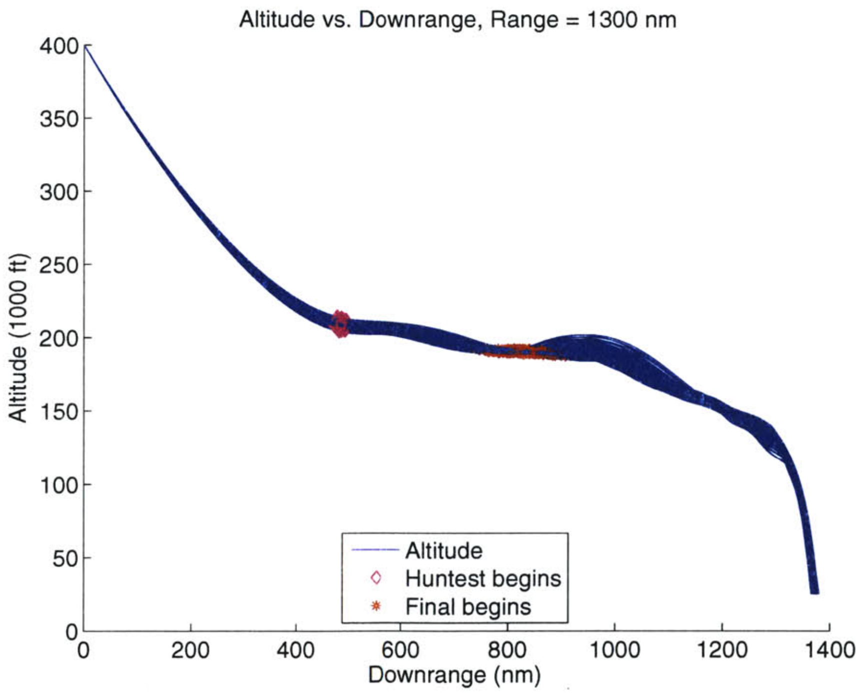


Figure 7-15: Apollo algorithm loft during Final phase

7.4.1 PredGuid Changes for Loft

Additional Integration Termination Condition

Two changes were made to PredGuid to make it more robust for loft trajectories. The most significant change is an additional integration termination condition. The Predictor algorithm's version of PredGuid uses the three original conditions (see Section 5.2.2) plus a fourth: $\gamma < -0.5°$.

The third original condition requires that the drag must be greater than 0.2 g's. In a loft, the drag never drops below 0.2 g's because the vehicle never leaves the atmosphere, so this condition is always true. The first two conditions—the skip has begun and the altitude rate is negative—ideally are sufficient to determine the end of the loft, but PredGuid struggles with this when the loft flight path angle is very shallow. Short loft trajectories essentially "skim" across the atmosphere rather than having distinct "up" and "down" portions. The very shallow flight path angle associated with the "skim" creates a problem because very minor changes in the flight path angle can change its sign, thereby tripping the negative altitude rate condition, often too early or too late in the predicted trajectory.

This creates problems in the vehicle's true trajectory. PredGuid uses the predicted skip end conditions to predict the Final phase range, and this range is saved to use in the next guidance cycle as part of PredGuid's total downrange objective. Having inconsistent predicted trajectory lengths from cycle to cycle causes large deviations in PredGuid's bank angle solution from the previous cycle.

Because of these inconsistencies, the vehicle can complete the loft too early, falling short of the desired range. As PredGuid determines that the range is short, it will command a small bank angle to "pull up" the trajectory. As the flight path angle becomes positive again, PredGuid's predicted range increases significantly and often overshoots the start of the Final phase reference trajectory. To compensate, PredGuid commands a large bank angle to push the trajectory back down. As this pattern repeats, a wave effect is observable in the vehicle's true trajectory, as seen in Figure 7-16.

This pattern is undesirable, primarily because PredGuid can saturate the bank angle trying to meet the range requirement at the end of the trajectory. Figure 7-17 shows the

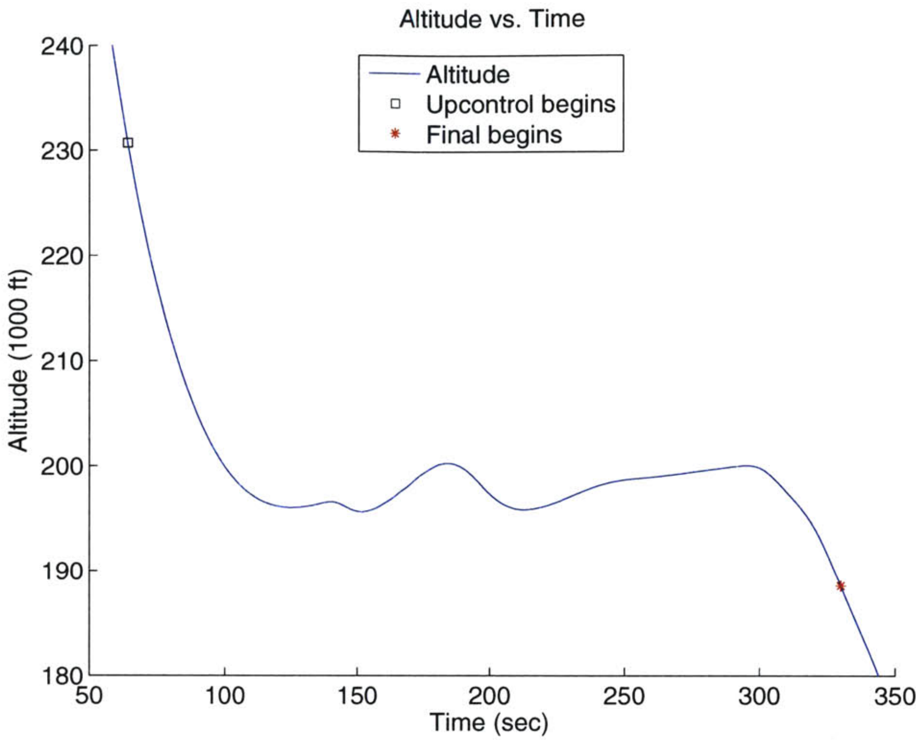


Figure 7-16: Altitude oscillations in loft trajectory due to PredGuid commands

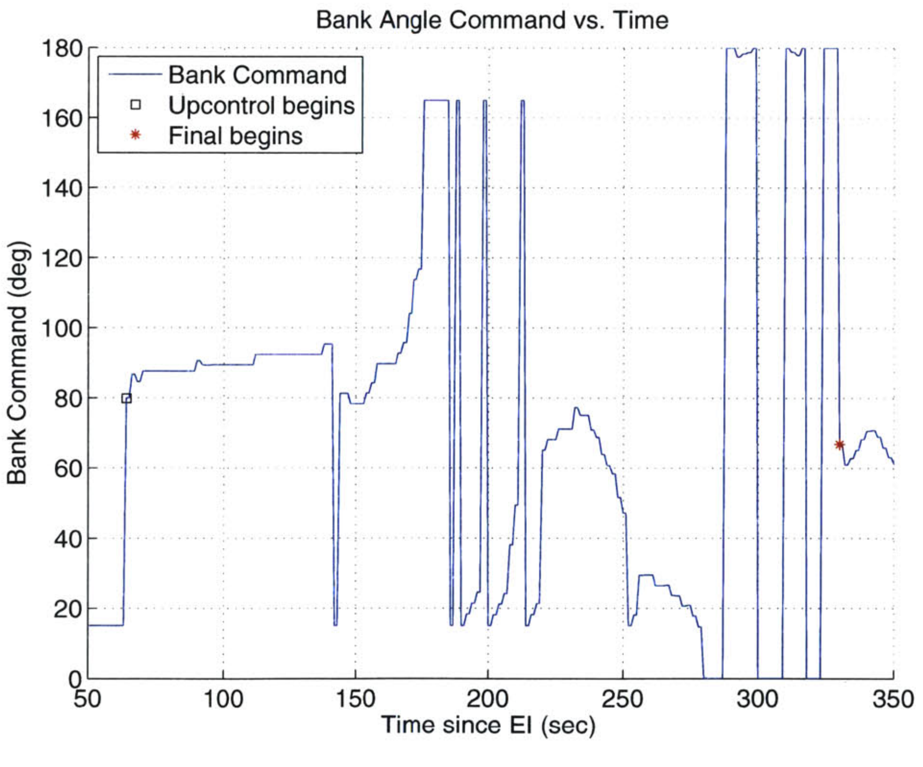
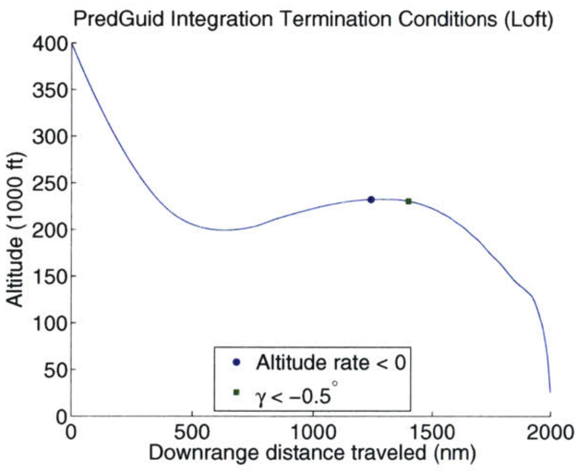


Figure 7-17: PredGuid command history for loft trajectory

bank angle command history for the same trajectory shown in Figure 7-16. By forcing PredGuid to continue integrating until the predicted flight path angle is steeper than $-0.5°$, which is well past the apogee of the loft, small oscillations in the predicted altitude do not trip the negative altitude rate condition. The PredGuid integration termination conditions for a loft are shown in Figure 7-18.

The flight path angle termination condition was included specifically for loft cases and has no effect on skips. For a typical skip, PredGuid terminates integration because of the drag trigger. Altitude starts decreasing once the predicted trajectory has passed apogee, and shortly afterward the flight path angle decreases below $-0.5°$. Much later, the drag increases above 0.2 g's and the predicted trajectory is terminated. This sequence of events is also shown in Figure 7-18. Because drag is the final trigger, adding the flight path angle condition does not change the time that the skip trajectory is terminated.



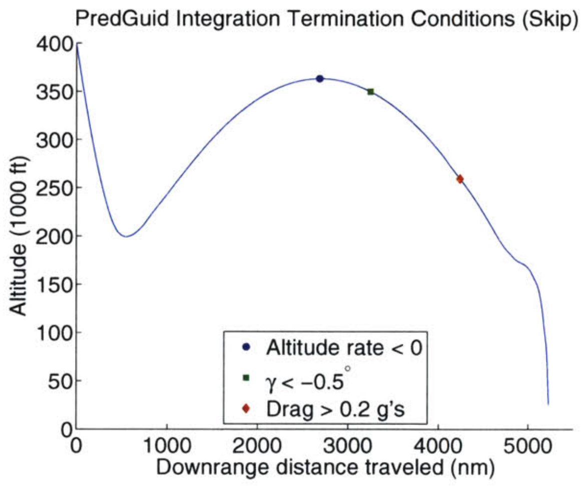


Figure 7-18: Sequence of PredGuid integration termination conditions for loft (left) and skip (right) trajectories

PredGuid Shutoff at End of Loft

The range calculation used in PredGuid's predictor exhibits precision errors for very short target range inputs [2]. Because of this, PredGuid's bank angle solution is often erratic at the end of the skip or loft as it approaches the transition to the Final phase. This is due in part to the low air density at the end of the skip—the vehicle has little control authority and PredGuid does not receive the desired response from its bank angle commands, so it commands more extreme bank angles to correct the predicted trajectory. This can saturate the controller. Fortunately, these large bank angle excursions do not have much effect on the overall trajectory for a long skip, precisely because the air density is so low, although they do cause the vehicle to expend fuel unnecessarily. For a loft however, the situation is different—the erratic bank commands will alter the trajectory since the vehicle is still in the atmosphere.

To create a smoother trajectory transition at the end of the loft and to avoid saturating the controller, PredGuid is not run in loft cases when its objective distance is less than 200 *nm*. This 200 *nm* limit is also used for skip cases in which PredGuid is not run because the drag is below 0.05 g's. Instead, a neutral bank command ($\phi = 90°$) is passed to the flight control system. While this has the potential to increase the initial deviation from the reference trajectory at the beginning of the Final phase, the error is not large because the neutral bank command is used for typically less than 40 seconds.

7.5 EMT Predictor

The most significant addition to the Draper baseline to create the Predictor algorithm is the introduction of the Energy Management and Transition (EMT) Predictor. The EMT Predictor runs during Initial Roll, starting when the drag reaches 0.05 g's, and it makes nearly all the major trajectory decisions. These are:

1. Whether a skip is necessary to reach the target
2. Skip/loft
   (a) The appropriate time to transition to PredGuid
   (b) The constant bank angle during the skip that will reach the target
3. Direct entry
   (a) The appropriate Constant Drag value
   (b) The Initial Roll L/D fraction that will reach that Constant Drag value

7.5.1 Trajectory Type Selection

The EMT Predictor's first task is to determine what type of trajectory is required to meet the downrange requirement. The predictor first tries to fly to the target using a direct entry; if the range is too short using the lowest Constant Drag value, then the range is not achievable via direct entry and a loft or skip is necessary.

The trajectory type decision is an ongoing process during Initial Roll. While the EMT Predictor is trying to determine the appropriate trajectory, the reference trajectory is also being updated with the predicted Final phase initial conditions. As the reference trajectory changes, the Final phase range changes, thereby modifying the EMT Predictor's range requirement.

To avoid too much flip-flopping between direct entry and loft, the EMT Predictor has two safeguards in place:

1. If the trajectory type during the previous guidance cycle was a skip, the predictor does not attempt to determine if the trajectory could be flown as a direct entry—it goes immediately to its loft/skip logic to determine the appropriate transition time to Upcontrol.

2. If the predictor has gone back and forth between a loft and a direct entry more than the allowable number of times during previous guidance cycles, direct entry is no longer allowed and the predictor goes immediately to its loft/skip logic to determine the appropriate transition time to Upcontrol.

This selection process is a solution to the problem of appropriately determining the trajectory type at the beginning of the entry, as discussed in Section 6.5.

7.5.2 Transition Time Determination

Once the EMT Predictor has determined what type of trajectory to fly, it must then determine the correct time to transition to the next phase. This process is different for a direct entry than a skip or loft.

Direct Entry

If the EMT Predictor has determined that a direct entry is necessary, the next step is to determine when to transition to Huntest. The goal is to transition just before the Initial Roll drag and altitude rate reach their desired Constant Drag values, so the Gamma Turn Controller has enough time to modify the flight path angle rate.

During each guidance cyle, the predictor first integrates the equations of motion forward in time from the current vehicle state until the time maximum drag is reached, using a constant bank angle of 15° (full lift up, limited by Lateral Logic's ±15° requirement). The maximum drag is recorded and used as the Constant Drag value. The predictor's range traveled from the current point to the maximum drag point is recorded as the Initial Roll range. The range traveled during Constant Drag is calculated using Equation 7.5. The Final phase range is calculated from the predicted Constant Drag exit conditions, using Equation 4.10. The total predicted range is the sum of the individual phase ranges, and this is compared to the desired range. If the predicted range is too long, the Initial Roll bank angle guess is increased (thereby increasing the Constant Drag level), and the process is repeated until a suitable Initial Roll bank angle and the corresponding Constant Drag value are found. If the predicted range is too short, the Initial Roll bank angle is reset to 15° and the predictor switches to its loft/skip logic.

Assuming an Initial Roll L/D fraction has been found, the predictor then decides what time to start the Gamma Turn Controller (described in Section 7.3.3). To do this, the predictor flies the Initial Roll constant bank angle until 1 second before the time of maximum drag ($T_{max\ drag}$). At $T_{max\ drag} - 1\ sec$, the predictor starts commanding the L/D required for the turn. During the turn, the predicted flight path angle approaches the Constant Drag reference flight path angle, but if the turn is commanded for too long, the predicted flight path angle starts to diverge from the Constant Drag reference. At the point that the error between the predicted flight path angle and the Constant Drag reference starts to increase, the turn predictor is terminated. The minimum flight path angle error during the predicted trajectory is recorded. This sequence is shown in Figure 7-19. The predictor then steps back to $T_{max\ drag} - 2\ sec$ to begin the turn and propagates the trajectory forward again. If the

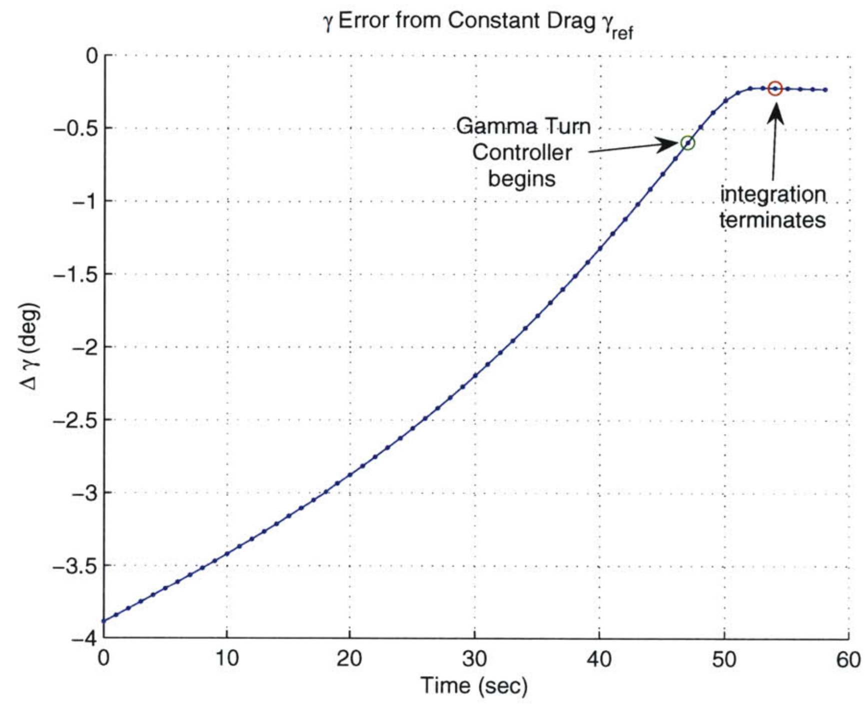


Figure 7-19: γ error from Constant Drag reference during direct entry prediction

minimum flight path angle error for this start time is smaller than the error for the previous start time, this start time is saved as the desired time to begin the Gamma Turn Controller. Using this process, the predictor keeps stepping backward in time until $T_{max\ drag} - 10\ sec.$ The trial that gives the smallest flight path angle error from the reference determines the current guidance cycle's predicted time to start the Gamma Turn Controller. Section 7.3.3 describes how the Gamma Turn Controller transitions to Huntest.

Loft/Skip

If a loft or skip is required instead, the EMT Predictor uses different logic to determine the appropriate time to transition to Upcontrol. In this case, the current vehicle state is propagated forward in time using the Initial Roll constant bank angle until the maximum drag time is reached. At this point, the predictor switches to a reference skip bank angle and propagates the trajectory forward until the skip is complete. As with PredGuid, the predictor terminates integration when the drag is above 0.2 g's and the flight path angle is steeper than $-0.5°$.

Once the integration is complete, the total range is calculated as the sum from Initial Roll, the skip, and the Final phase. If the difference between the total predicted range and the desired range is greater than the allowable tolerance, the transition time to the reference skip bank angle is adjusted. If the predicted range is long, the transition time guess is decreased so that it occurs earlier in the trajectory. If the predicted range is short, the transition time guess is increased so that it occurs later.

This process repeats until one of the termination criteria below is met:

1. The range error is within the allowable tolerance
2. The range error has changed sign from the previous iteration
3. The maximum allowable number of iterations is reached

If the range error is within the allowable tolerance, the last guess is saved as the transition time. If the range error is still too large but it has changed sign since the previous guess, this indicates that the zero range error point lies somewhere between the previous transition time guess and the current one. In this case, the time guess with the smaller error is saved as the transition time and the predictor makes small adjustments to the reference skip bank angle until the predicted range is within the tolerance of the desired range. If the maximum number of iterations is reached, the EMT Predictor uses $T_{max\ drag}$ as the transition time initial guess in the next guidance cycle.

The Draper baseline's method of defining the transition to the skip is to start Upcontrol at a fixed time and then use PredGuid to adjust the skip bank angle to meet the range requirement. The EMT Predictor's approach is the reverse – a desired bank angle is chosen for the skip and the Upcontrol transition time is adjusted until the desired range is achieved. This addresses the dilemma of when to transition to Upcontrol and PredGuid, described in Section 6.6.

7.5.3 Shallow Entry Correction

Unlike the Apollo algorithm, the Predictor algorithm does not have a dedicated shallow entry test at 0.05 g's in Initial Roll. However, the algorithm compensates for shallow entries during Initial Roll using the EMT Predictor.

For direct entries, no additional logic is needed to correct for shallow entries. If the vehicle enters the atmosphere in the shallow end of the corridor, its predicted range will be longer than if it entered in the center of the corridor. To shorten the range, a higher Constant Drag level will be necessary, and this is achieved by a larger bank angle (lower L/D fraction) during Initial Roll. Rolling the lift vector toward lift down to increase the drag will automatically steepen the entry.

For loft and skip entries, additional logic is necessary to steepen the entry. The default bank orientation during Initial Roll for loft and skip entries is 15°. This is chosen to keep g loads low. If the EMT Predictor determines that the transition to Upcontrol will take place too early in the trajectory, this indicates that the entry is shallow and the EMT Predictor commands a lower L/D fraction to steepen the entry.

This phenomenon is a result of the different constant bank angles for Initial Roll and the skip. For a given reference skip bank angle, which is typically between 60° and 120°, the later the transition to Upcontrol, the more time the vehicle spends flying lift up during Initial Roll, which increases the downrange. Conversely, if the transition to Upcontrol is earlier, the vehicle spends more time flying at a larger bank angle, thereby decreasing the downrange. As is the case with direct entries, if the vehicle enters the atmosphere at a shallow flight path angle, the predicted range will be long. This means that the transition to Upcontrol will need to take place earlier to shorten the range. If the predictor determines that the transition will happen at too low a drag level (too early in the trajectory), the Initial Roll bank angle is increased incrementally to steepen the initial descent into the atmosphere. In each guidance cycle that the predicted transition to Upcontrol is too early, the Initial Roll bank angle is increased by 30°. Once the predicted transition time occurs at an acceptable drag level, the Initial Roll bank angle is no longer increased.

The EMT Predictor's method of determining whether the entry is shallow and adjusting the Initial Roll bank angle accordingly is a solution to the shallow entry test problem discussed in Section 6.4. By using model based predictions to determine the vehicle's position in the entry flight path angle corridor, the dependency on empirical constants or relationships is eliminated. Additionally, the process of increasing the Initial Roll bank angle incrementally gives the algorithm more flexibility to steepen the entry while keeping the g loads low, rather than flying full lift down.

7.6 Lateral Steering

No significant changes have been made to the lateral steering logic in the development of the Predictor algorithm. A minor change has been made for the calls to Lateral Logic during the Initial Roll and Huntest phases—during this time, bank reversals are prohibited. The Initial Roll and Huntest/Constant Drag phases are crucial in the effort to bleed off the correct amount of energy, and rolling through full lift up or full lift down to change the sign of the bank angle can significantly change the amount of energy lost over the time required for the bank reversal. In order to keep the energy decay rate constant, Lateral Logic is prohibited from commanding reversals during these first two phases.

7.7 Summary of Initial Roll and Huntest

Initial Roll and Huntest were the two phases affected the most by the addition of the EMT Predictor. The Predictor algorithm's logic for these phases is shown in Figures 7-20 and 7-21.

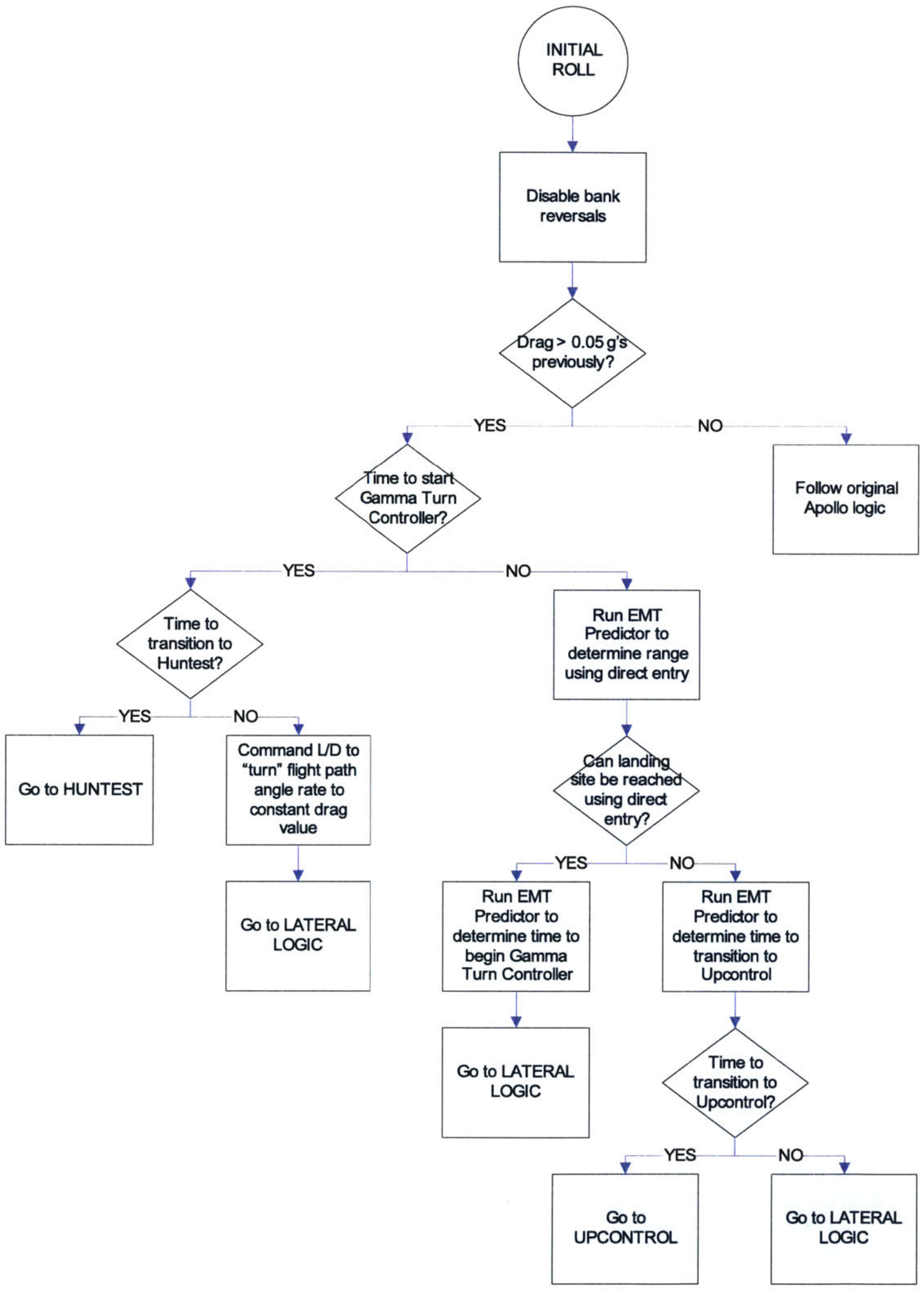

INITIAL ROLL
Disable bank reversals
Drag > 0.05 g's previously?
YES
NO
Follow original Apollo logic
Time to start Gamma Turn Controller?
YES
NO
Time to transition to Huntest?
YES
NO
Go to HUNTEST
Command L/D to "turn" flight path angle rate to constant drag value
Go to LATERAL LOGIC
Run EMT Predictor to determine range using direct entry
Can landing site be reached using direct entry?
YES
NO
Run EMT Predictor to determine time to begin Gamma Turn Controller
Go to LATERAL LOGIC
Run EMT Predictor to determine time to transition to Upcontrol
Time to transition to Upcontrol?
YES
NO
Go to UPCONTROL
Go to LATERAL LOGIC

Figure 7-20: Predictor algorithm: Initial Roll phase logic

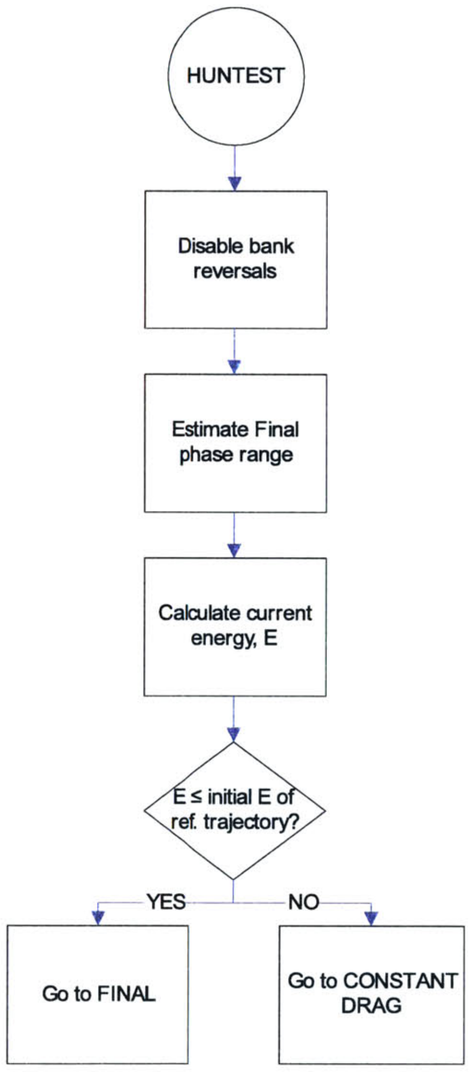

HUNTEST
Disable bank reversals
Estimate Final phase range
Calculate current energy, E
E ≤ initial E of ref. trajectory?
YES
NO
Go to FINAL
Go to CONSTANT DRAG

Figure 7-21: Predictor algorithm: Huntest phase logic

Chapter 8

Results

8.1 Overview

The Draper PredGuid algorithm was tested with and without the EMT Predictor algorithm enhancements through a variety of methods. The Predictor enhancements show improvement in several areas, namely reduced control saturation and increased robustness during the Final phase. The Predictor algorithm also increases the short-range direct entry capability by 300 *nm*.

This chapter first describes the trajectory shaping that results from the combination of the Initial Roll bank angle and the reference skip bank angle. It has been found that a skip trajectory's heating and load characteristics can be adjusted to meet certain criteria by changing the skip bank angle. Next, the Predictor enhancements were tested under nominal conditions to determine the range of downrange distances and entry flight path angles they could achieve. This is considered to be stress-to-failure testing. Finally, the Monte Carlo results are presented for nine different target ranges. This set includes ranges considered to be representative of the Predictor algorithm's different entry types, as well as ranges that stress the algorithm's ability to transition between those types.

8.2 Skip/Loft Trajectory Shaping

The results presented in this chapter are based on two important user-defined inputs. The first is the Initial Roll bank angle for skip and loft trajectories; for the results presented here, the default Initial Roll bank angle is $\pm 15°$ (full lift up). The second input is the reference skip bank angle. This thesis uses $80°$ for both loft and skip trajectories, although different values could be set for each trajectory type. The combination of these two bank angles impacts the aerodynamic heating and loading of the vehicle as well as the overall shape of the trajectory. In the future, these reference values can be selected by the guidance designer to address specific mission objectives, such as minimizing total heat load, g load, or heat rate.

8.2.1 Initial Roll Bank Angle

For a nominal skip or loft entry using the Predictor algorithm, the vehicle flies a constant bank angle of $15°$ during Initial Roll. This decision is based on the Apollo algorithm's L/D policy during Initial Roll, which is that lift up is commanded if the entry is not shallow. Flying lift up reduces the initial g load on the vehicle as it descends into the atmosphere and it prevents the vehicle from losing too much energy, and therefore downrange capability, before entering the skip.

The choice of Initial Roll bank angle, however, has an impact on the remainder of the trajectory. For a fixed bank angle during the skip, a small Initial Roll bank angle extends the trajectory's downrange. This also impacts the transition time to Upcontrol. As long as the skip bank angle is larger than the Initial Roll bank angle, the following trend is observed: smaller Initial Roll bank angles force earlier transitions to Upcontrol, which is a result of the algorithm compensating for the longer predicted range. This is shown in Figure 8-1.

8.2.2 Skip Bank Angle

The reference or desired skip bank angle for this thesis is set to $80°$ for both skip and loft trajectories. This value was chosen after some consideration of the possible constant bank angles

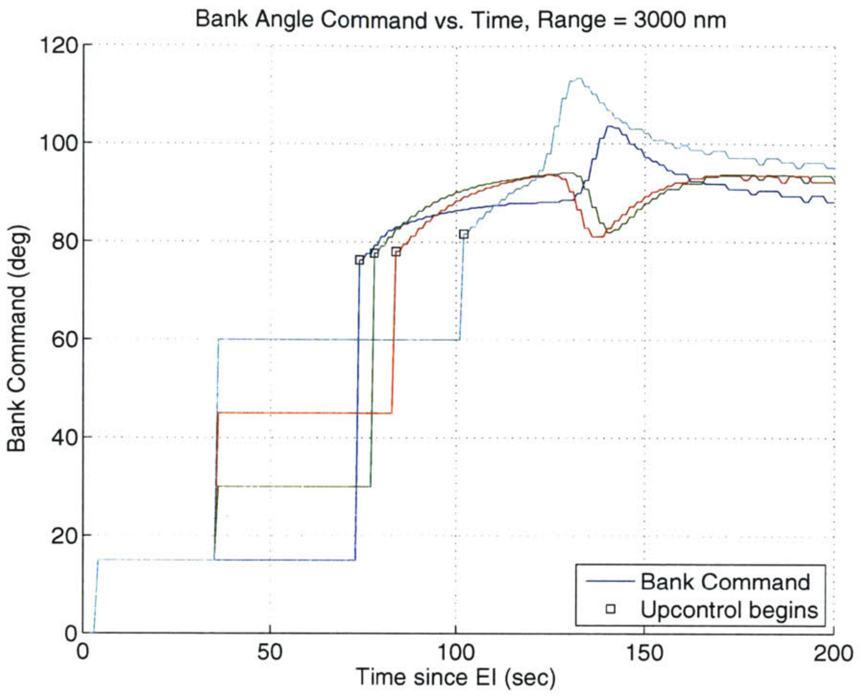


Figure 8-1: Larger Initial Roll bank angles delay transition to Upcontrol

and their impact on the trajectory. Small skip bank angles (lift up) increase the predicted range, and bank angles that are too small are infeasible for short range trajectories—the algorithm needs to transition to Upcontrol as soon as Initial Roll begins and the predicted range is still too long. Bank angles that are too large (lift down) delay the transition to Upcontrol so much that it occurs once the vehicle has left the atmosphere during the skip. This can be dangerous because it reduces the time that PredGuid has to perform the necessary targeting prior to the Final phase.

Using an Initial Roll bank angle of 15°, the general range of feasible skip bank angles for the CEV configuration was 60° – 120°; however, the range of feasible skip bank angles changes slightly with the downrange distance. The low end of the bank angle range is given by the smallest bank angle that will not overfly the target, and the high end of the range is often limited by the Negative Test subroutine, which commands a neutral bank angle if the current L/D command is negative and the current drag exceeds 175 ft/s^2. Smaller skip bank angle trajectories typically have higher maximum heat rates but lower total heat loads, due to the shorter amount of time spent in the atmosphere. These trajectories have higher maximum altitudes and reenter the atmosphere at a steeper flight path angle. Larger skip bank angle trajectories showed lower maximum heat rates but higher total heat loads. These trajectories have lower maximum altitudes and shallower flight path angles upon reentry into the atmosphere. These differences can be seen in Figure 8-2. Smaller skip bank angles also tend to increase the maximum g load, shown in Figure 8-3.

Another consideration in choosing the reference skip bank angle is the constant L/D fraction used in the design of the Final phase reference trajectory. Small bank angle differences at the start of the Final phase are desirable because the control effort required to correct the error is small. The skip and loft trajectories considered in this thesis use a Final phase reference trajectory that has an L/D fraction of 0.6, which corresponds to a bank angle of approximately 53°.

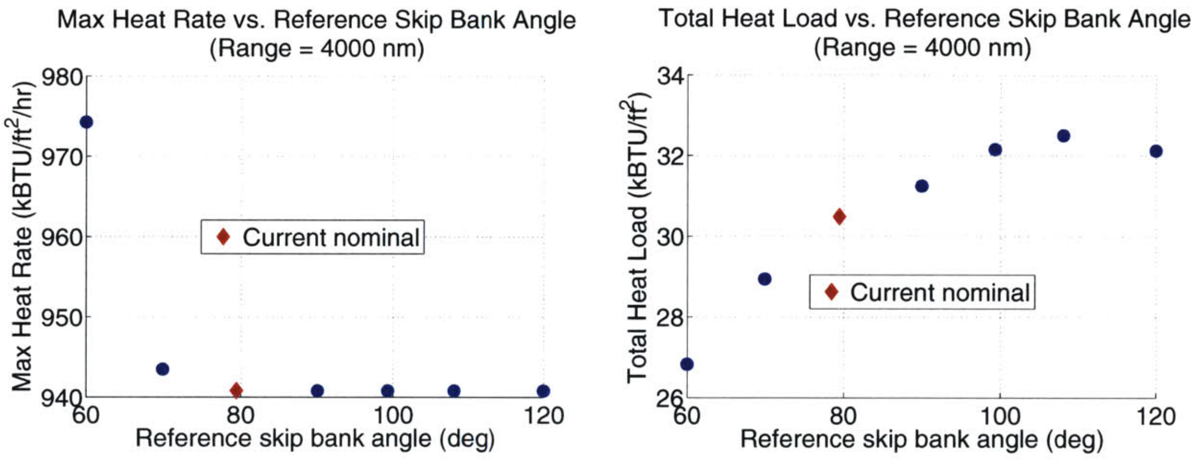


Figure 8-2: Variation in aeroheating due to reference skip bank angle

This is not an exhaustive list of the mission design elements to consider when selecting an appropriate reference skip bank angle, and it was not the goal of this thesis to perform a comprehensive analysis to arrive at the optimal value. 80° was chosen because it is the approximate center of the feasible corridor and it performed well for both loft and skip trajectories at all target ranges tested. The key point is that the Predictor algorithm allows an engineer to optimize the trajectory for desired characteristics. This is an area where the Predictor algorithm offers significantly more freedom to the mission design team.

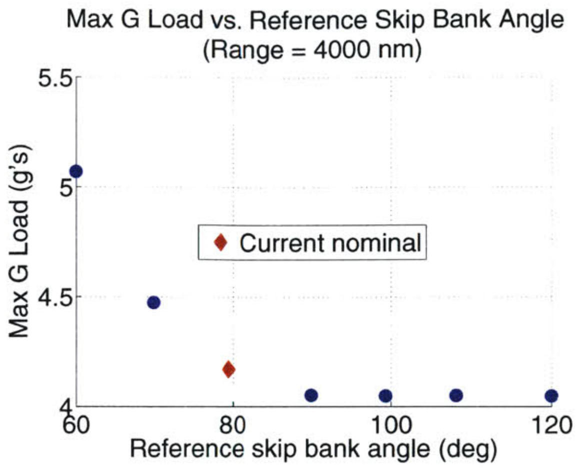


Figure 8-3: Variation in max g load due to reference skip bank angle

8.3 Precision Landing Downrange Capability

The objective of this test was to determine the maximum and minimum downrange capability of the Predictor algorithm, as well as ensure that there were no intermediate downrange distances that could not be reached. This test was not a true footprint analysis because the crossrange capability was not tested—only the centerline of the footprint was examined. The crossrange capability was not tested because the lateral steering logic is essentially unchanged from its Apollo predecessor.

The set of target ranges tested was 800 nm to 8100 nm in increments of 100 nm. All vehicle and environmental parameters were set at their nominal values for this test. A particular downrange distance was considered to be successful if it met the precision landing requirement of 2.3 nm and did not exceed the maximum g load limit of 10 g's. The successful upper limit for both algorithms was 8000 nm, but the limit on the shortest target range for the Draper baseline was only 1200 nm, while the Predictor algorithm's lower limit was 900 nm. Neither algorithm showed any intermediate downrange distances that could not be met. Figure 8-4 shows how the Draper and Predictor algorithms shape the trajectories differently for these downrange targets.

This exercise also provided limits of the downrange capability of each trajectory type for each algorithm, as shown in Figure 8-5. These results show that the Predictor algorithm increases the nominal direct entry capability by 300 nm, which can also be seen in Figure 8-6.

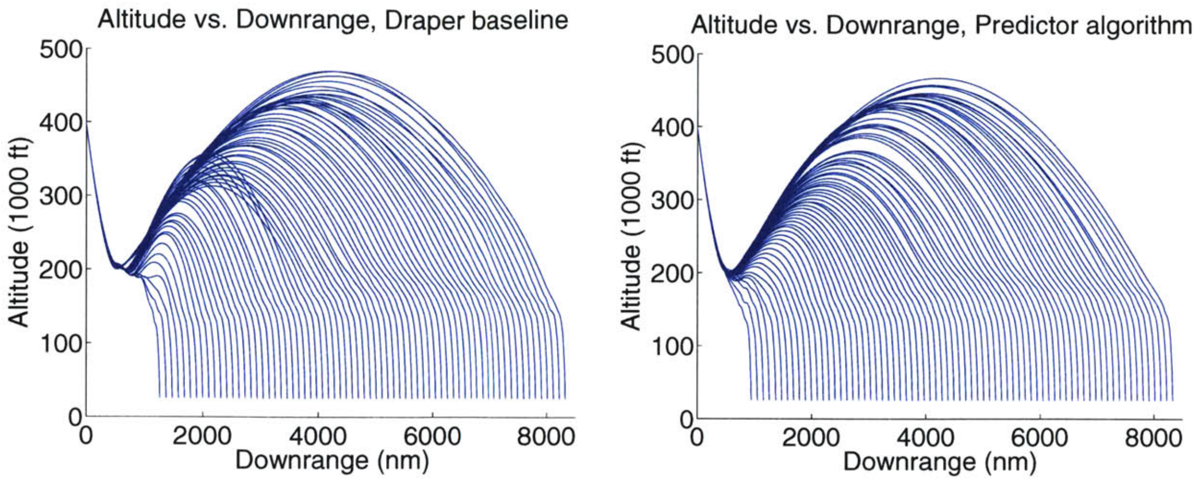


Figure 8-4: Draper baseline (left) and Predictor (right) downrange footprints

This data also provided target ranges to examine more fully in the Monte Carlo analysis, in order to assess how well the Predictor algorithm transitions between direct entry and loft, and loft and skip trajectories.

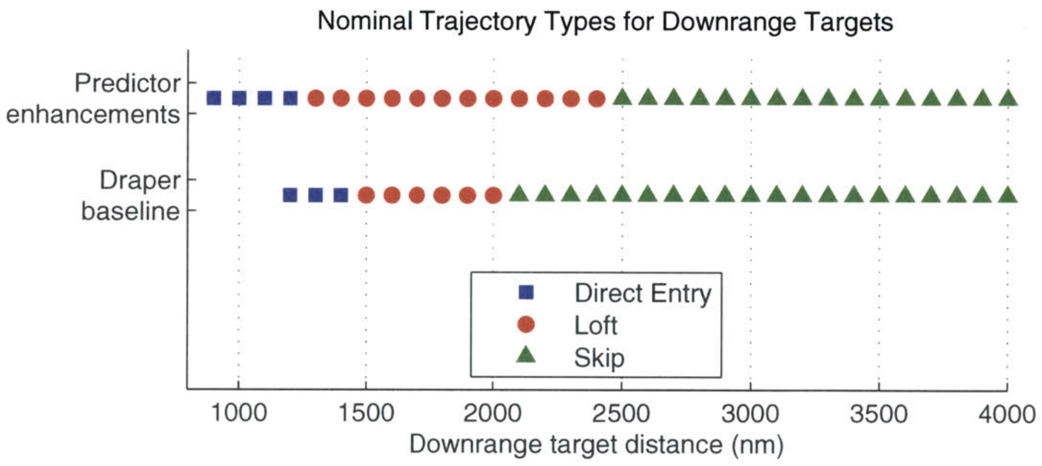


Figure 8-5: Comparison of trajectory types for different algorithms

8.4 Entry Flight Path Angle

The robustness of the Predictor algorithm's shallow entry correction was tested by varying the flight path angle at EI. The nominal value was -5.9°, and the entry flight path angle was varied from -5° to -8°. This test was performed using three different target ranges:

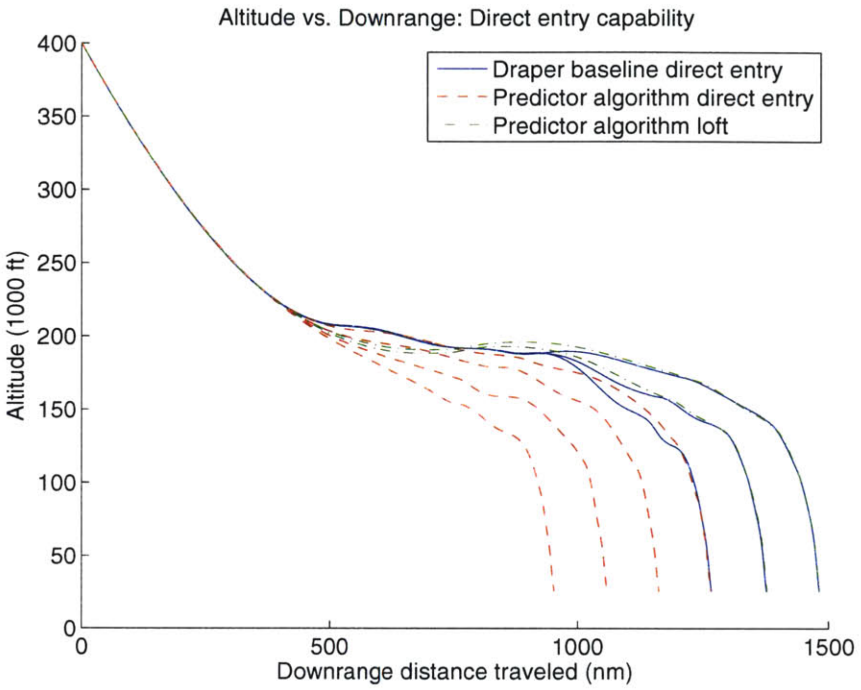


Figure 8-6: Predictor algorithm's increased direct entry capability

1100 *nm*, 1700 *nm*, and 5200 *nm*. These were chosen because they were the approximate center of the Predictor algorithm's ranging capability for direct entry, loft, and skip entries, respectively. Figure 8-7 shows an overall comparison of the entry flight path angles that each algorithm is able to fly while meeting the landing accuracy and maximum g load requirements.

In the 1100 *nm* case, the Predictor enhancements show significant improvement in the range of EI flight path angles it can accommodate. This is to be expected because the modified Apollo algorithm used in the Draper baseline overshot the landing site for this target range under nominal conditions; therefore, the entry flight path angle must be steeper than the nominal value for the algorithm to be able to meet the requirements.

The range of successful flight path angles for the 1700 *nm* and 5200 *nm* cases are relatively similar between the two algorithms. In the 5200 *nm* case, the Predictor algorithm's EI γ capability is smaller by one data point, $-6.6°$. This failure was due to the trial's miss distance. For this flight path angle, the Predictor algorithm determined that the transition to the reference skip bank angle was going to happen after the pullout ($\dot{h} = 0$) and at

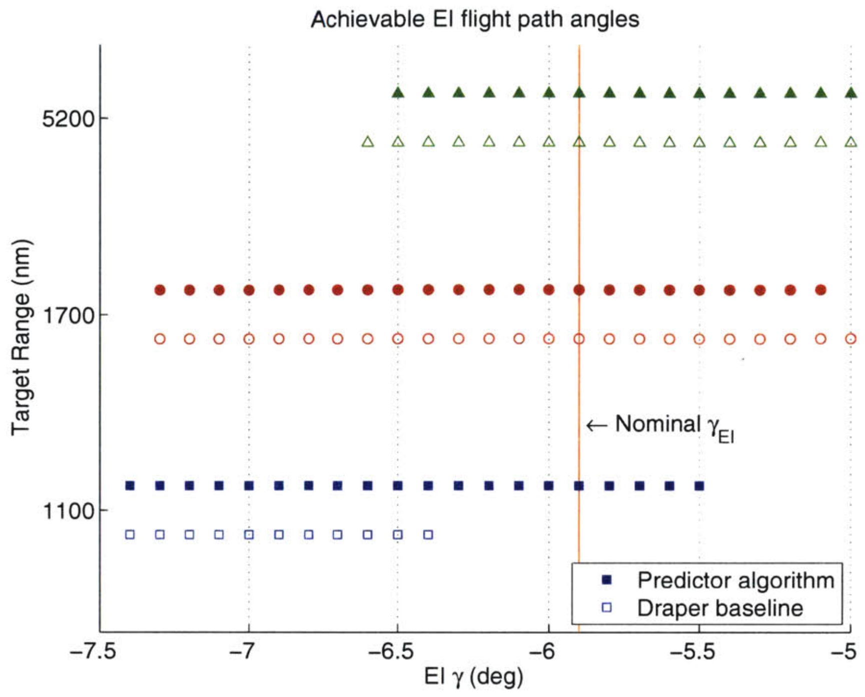


Figure 8-7: Entry flight path angle capability comparison

a very low g load. This indicates that the transition to Upcontrol would not take place until after the skip had started. Because the EMT Predictor's targeting method is not as accurate as PredGuid's, it is advantageous to transition to PredGuid before the vehicle leaves the atmosphere. In order to force the transition to Upcontrol to occur earlier, the EMT Predictor kept decreasing the reference skip bank angle until it reached 15°, the same value as the Initial Roll bank angle. This resulted in a constant bank angle predicted trajectory and the EMT Predictor could do nothing more to extend the range, so it transitioned to PredGuid. When PredGuid took over, it commanded bank angles near 90° for approximately 30 seconds before commanding full lift up to lengthen the range (see Figure 8-8), but by that point, the vehicle had lost too much energy and could not reach the landing site.

This result suggests that in these situations, perhaps the EMT Predictor should remain in Initial Roll for a certain amount of time before transitioning to Upcontrol, even if the range requirement is not met. By waiting to transition until after the pullout point, PredGuid may be in a better position to accurately predict the range and necessary bank commands.

For the 1700 *nm* set, the situation is slightly different. Again, the Draper baseline's

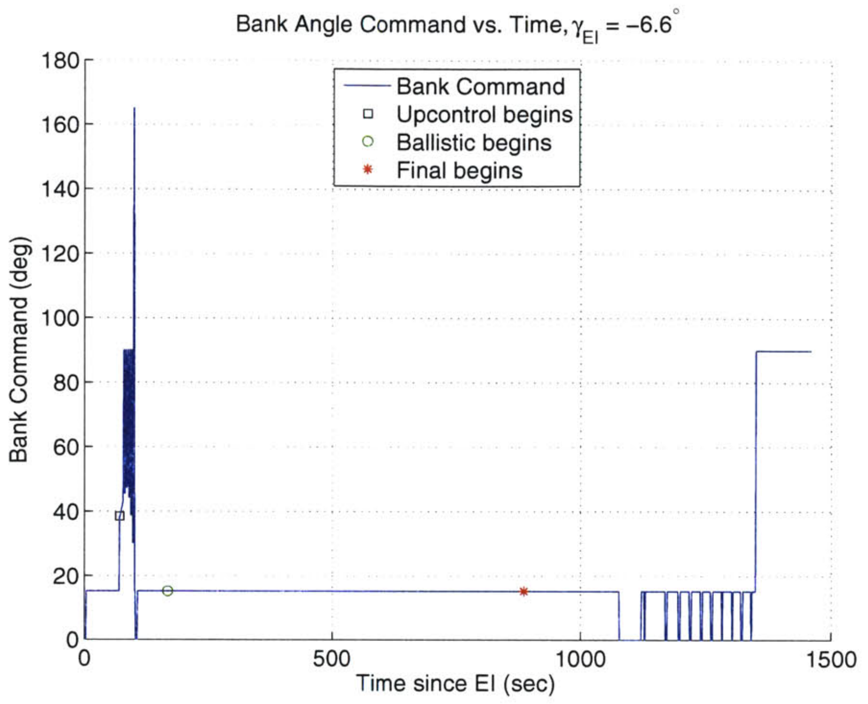


Figure 8-8: Bank angle command history, $\gamma_{EI} = -6.6^\circ$, range $= 5200\ nm$

range exceeds the Predictor algorithm's by one data point, -5°, but in this case it is because that trajectory skips out of the atmosphere using the Predictor algorithm. The Predictor algorithm treats the -5° case as a direct entry, and at 0.05 g's it commands a bank angle of 120° to get to the necessary Constant Drag value (see Figure 8-9). This negative L/D fraction should also steepen the flight path angle.

Because the vehicle does not achieve this bank angle instantaneously, the algorithm commands increasingly negative L/D fractions until it is nearly saturated, but by this time it is too late for the algorithm to compensate for not flying full lift down earlier. The modified Apollo algorithm used in the Draper baseline is successful at this shallow angle because it immediately commands full lift down at 0.05 g's. This suggests that it may be necessary to model the flight control system's maneuver rate in the EMT Predictor to compensate for the time required to bank to the desired angle. Another possible option is to command full lift down if the EMT Predictor's desired L/D fraction is less than some threshold.

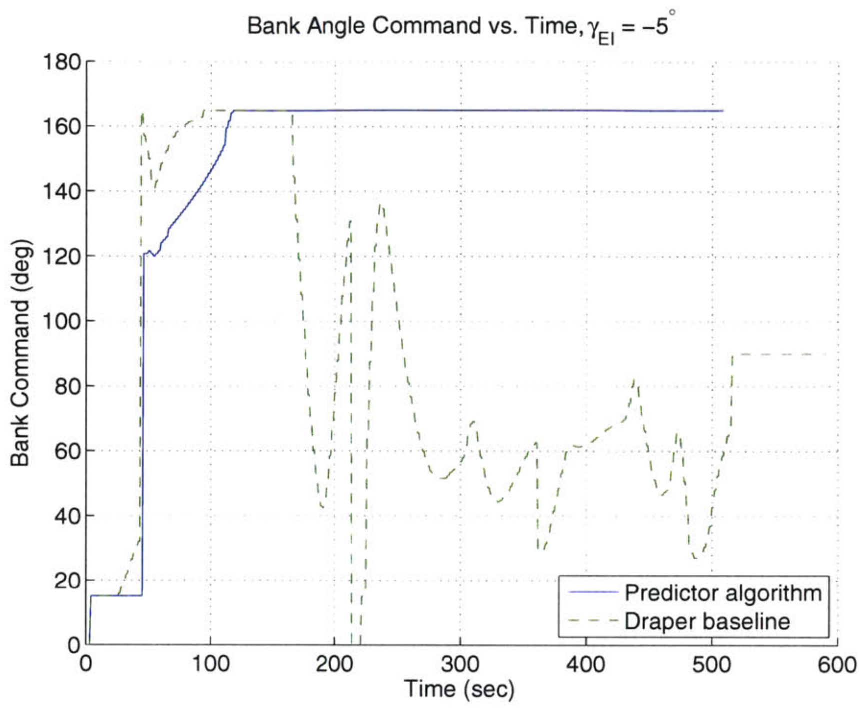


Figure 8-9: Bank angle command history for both algorithms, $\gamma_{EI} = -5°$, range = 1700 nm

8.5 Monte Carlo Results

Monte Carlo simulations were performed to assess the Predictor algorithm's capability in off-nominal situations and to compare its performance against the Draper baseline. Nine different target ranges were tested, and various vehicle-specific and environmental parameters were varied randomly in order to simulate typical day-of-flight uncertainties. These parameters, along with their nominal and 3-σ values, are listed in Table 2.1. Some target ranges were chosen to test the Predictor algorithm's ability to transition between trajectory types. Additional target ranges were selected to test the Predictor's midrange capability for direct entry, loft, and skip. These are summarized in Table 8.1. The Draper baseline algorithm was also tested at these target ranges so that a comparison of capability could be made. 1000 Monte Carlo trials were run at every target range for each algorithm, for a total of 18,000 trials.

It should be emphasized that while the Draper baseline algorithm was originally tested in *Reentry Guidance with Extended Range Capability for Low L/D Spacecraft*, the results

Table 8.1: Monte Carlo target ranges

Target Range	***Description***
900 *nm*	Shortest direct entry
1050 *nm*	Midrange direct entry
1200 *nm*	Longest direct entry
1300 *nm*	Shortest loft
1850 *nm*	Midrange loft
2400 *nm*	Longest loft
2500 *nm*	Shortest skip
4000 *nm*	Midrange skip
5400 *nm*	Longer skip

shown here cannot be directly compared to those results because they were generated using a 4 degree of freedom (DOF) simulation. The simulation environment utilized for this thesis is 6-DOF.

8.5.1 Landing Accuracy

Following the results from the nominal downrange footprint analysis in Section 8.3, the Predictor algorithm achieves much better landing accuracy for the short range targets, which are the direct entry cases.

Figure 8-11 is an enlarged version of Figure 8-10 which eliminates the Draper baseline's very large miss distances for the 900 *nm* and 1050 *nm* ranges. In Figure 8-11, it is obvious that the Predictor algorithm's landing accuracy for the 900 *nm* target range is poor. Similarly, the Draper baseline algorithm shows unacceptable accuracy for the 1200 *nm* target range, which is the shortest range it could achieve under nominal conditions.

Figure 8-12 zooms in even further on the remaining target ranges. The results for both algorithms are quite similar from the midrange loft range up to the longest skip range. The Predictor algorithm's performance is somewhat worse at 1300 *nm*, where the largest miss exceeds 6 *nm*. This case has a small downrange error but a significant crossrange error, as seen in Figure 8-13. This is most likely due to the fact that the very short loft cases transition to the Final phase quite low in the energy bucket, as seen in Figure 8-14. This gives the Final phase controller very little time to guide the vehicle onto the reference trajectory and to eliminate crossrange errors before impact.

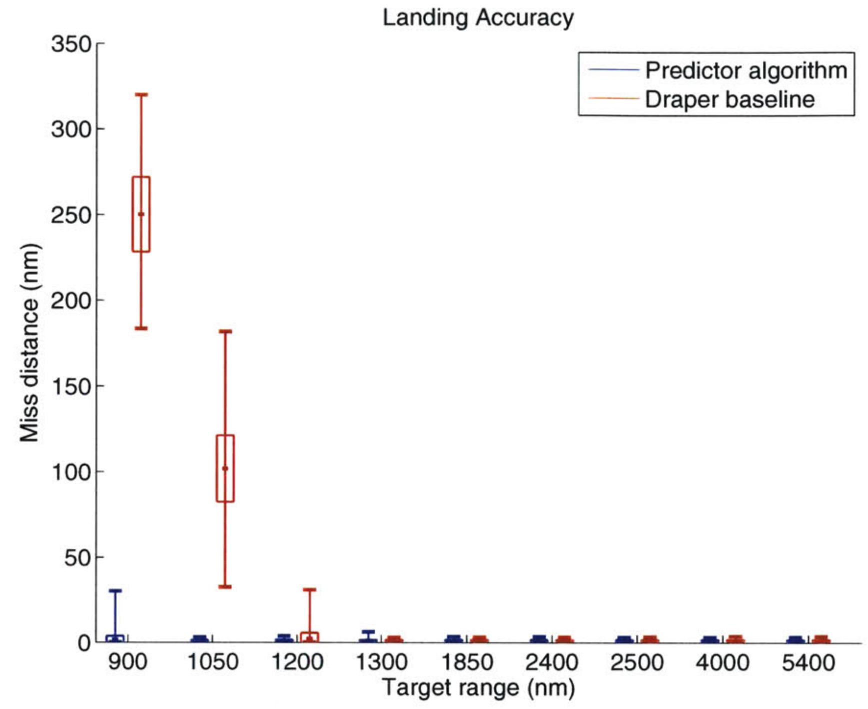


Figure 8-10: Summary of landing accuracy for all Monte Carlo target ranges

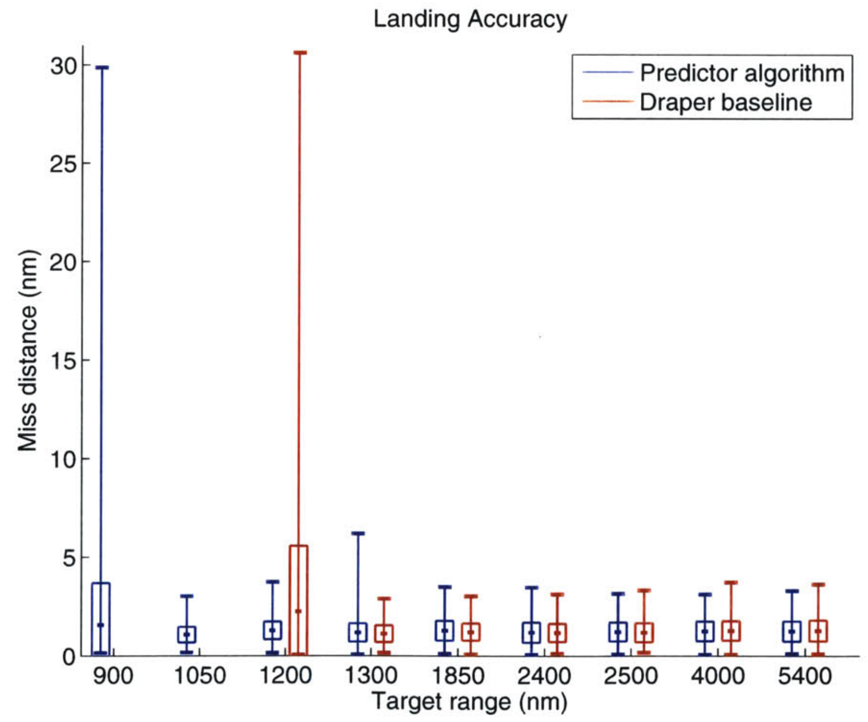


Figure 8-11: Monte Carlo landing accuracy, enlarged

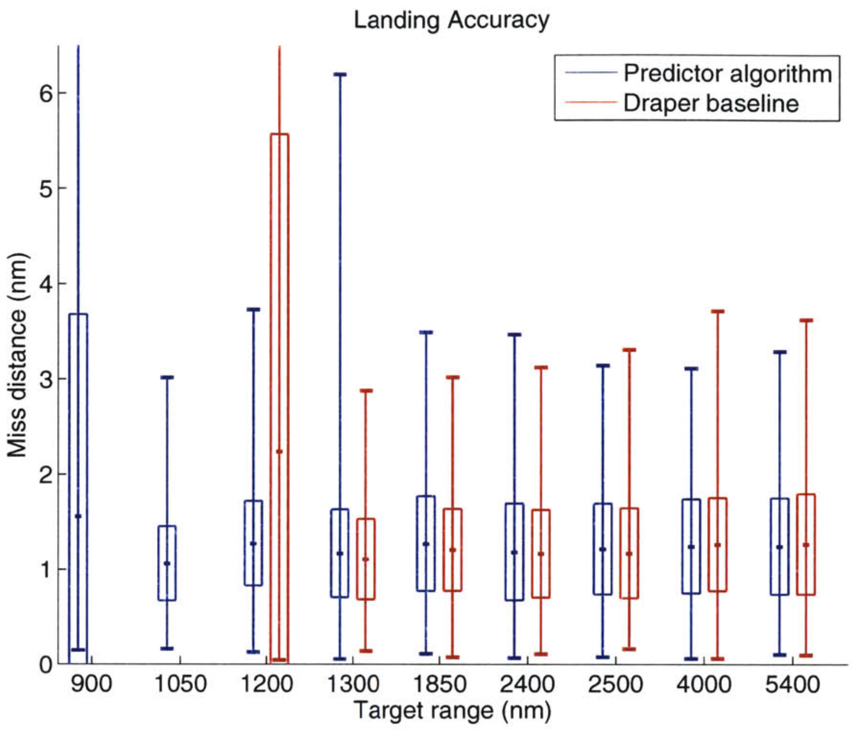


Figure 8-12: Monte Carlo landing accuracy, further enlarged

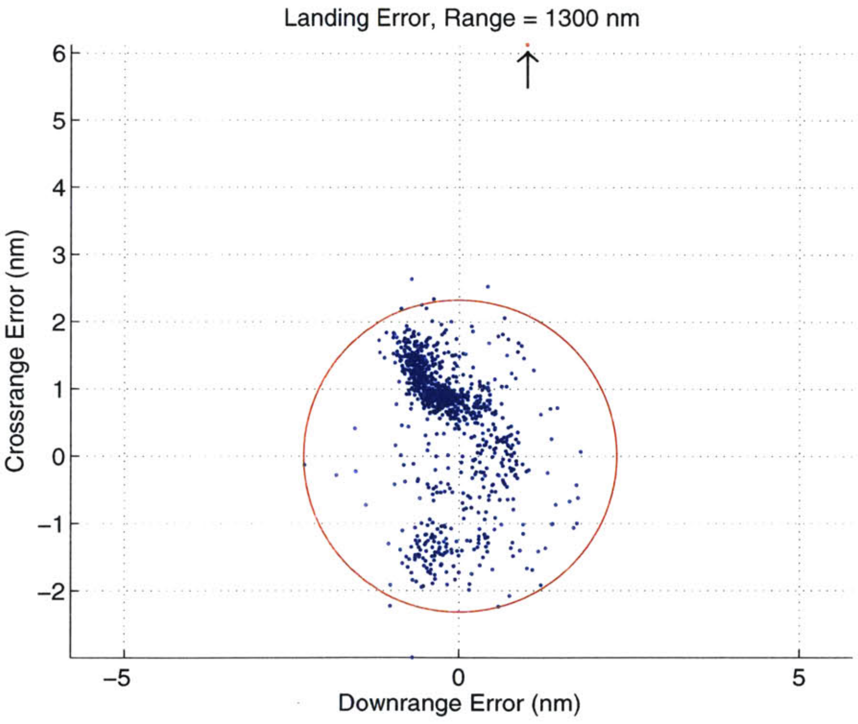


Figure 8-13: Large crossrange error in 1300 nm target range

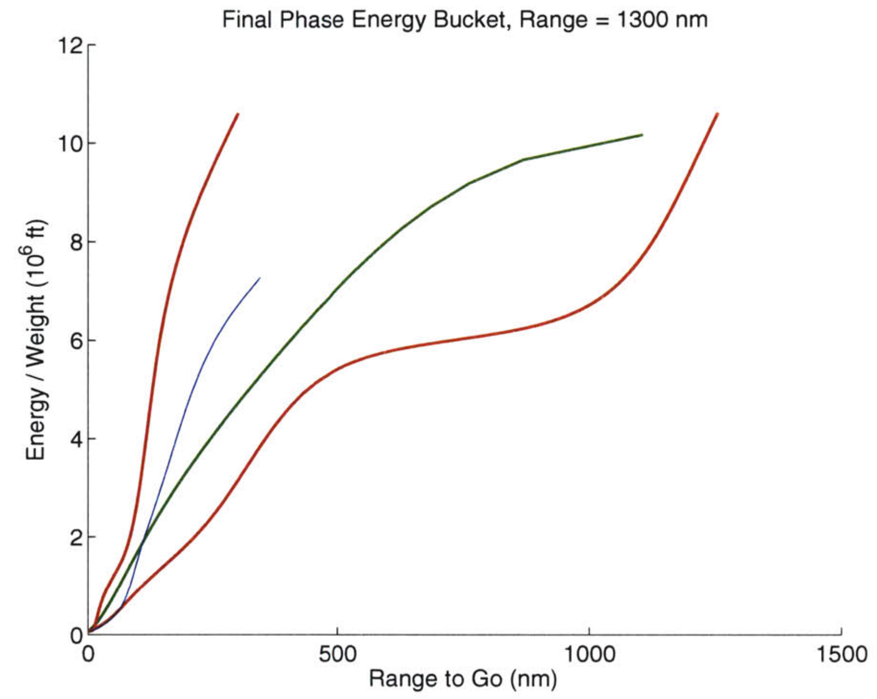


Figure 8-14: Crossrange error case in 1300 *nm* target range

It is interesting to note that neither algorithm met the 2.3 *nm* landing accuracy requirement for all 1000 Monte Carlo trials at any target range; however, for target ranges above 1200 *nm* for the Draper baseline and above 900 *nm* using the Predictor enhancements, the miss distances are not significantly greater than the requirement. A possible explanation for this error is the simulation's assumption that the drogue parachute deployment occurs exactly at $25,000\ ft$ altitude. In reality, the acceptable drogue chute deployment conditions are functions of limits on dynamic pressure, which imply limits on altitude and Mach number, so the drogue deployment point would be slightly different for each entry. For each target range, using the EMT Predictor enhancements, the transition to the Final phase occurs inside the energy bucket (see Section 8.5.6); this implies that the vehicle should be able to reach the landing site. The fact that misses still occur indicates that an additional controller may needed for the final descent to chute deploy. Landing accuracy could be improved by continuing to steer past $1000\ ft/s$, although this would naturally increase the fuel use.

8.5.2 g Loads

Maximum

Figure 8-15 shows that the Predictor algorithm significantly reduces the g load from the Draper baseline algorithm for direct entry cases. This reduction can be attributed to the Predictor's variable Constant Drag level, giving it the ability to bleed off more energy earlier in the trajectory. The Predictor algorithm shows marginally higher loads for the 1300 nm range and marginally lower loads for the longer target ranges.

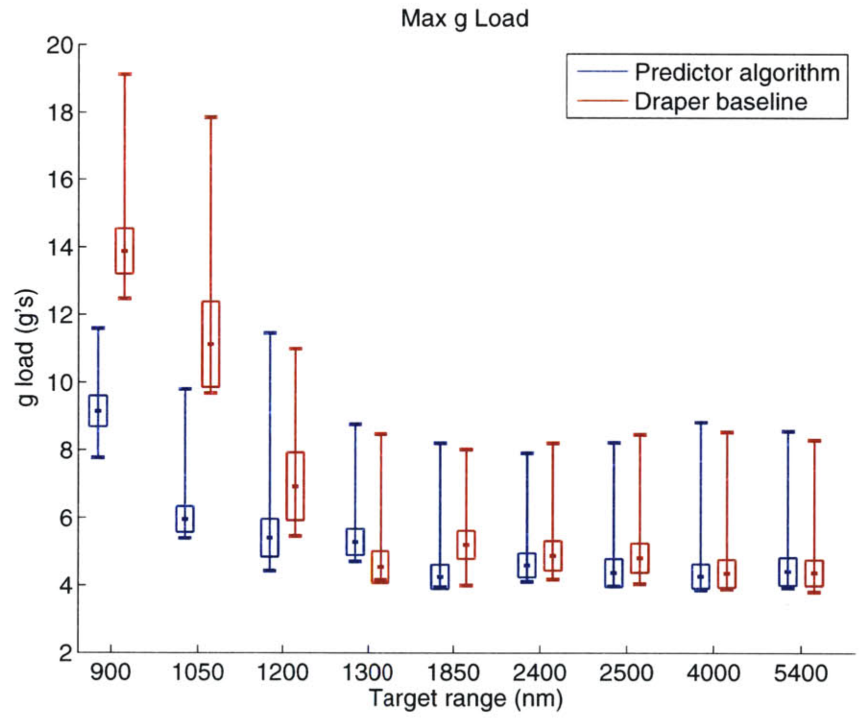


Figure 8-15: Summary of maximum g loads for all Monte Carlo target ranges

The Draper algorithm flies all of the Monte Carlo cases in the 1300 nm target range as lofts. The Predictor algorithm flies mostly lofts but has some direct entry cases. As Figure 8-15 demonstrates, direct entries typically experience higher maximum g loads than lofts. This increases the Predictor algorithm's average g load for this distance. Additionally, the Draper algorithm goes through Constant Drag prior to beginning the loft. This reduces the vehicle's velocity before the pullout, which is often the point where the highest g loads are seen.

Duration

As the target range decreases, the duration-based g loads begin to overrun the deconditioned crew limit at 100 seconds—the "green corner." The Draper baseline, using the modified Apollo algorithm, begins to also overrun the nominal low-duration (less than 10 seconds) acceleration limits for very short target ranges, 900 nm and 1050 nm. For the longer target ranges, 1850 nm and above, none of the Predictor algorithm's Monte Carlo cases exceed the duration-based g load limits. The duration-based g load results for the remaining target ranges can be found in Appendix A.

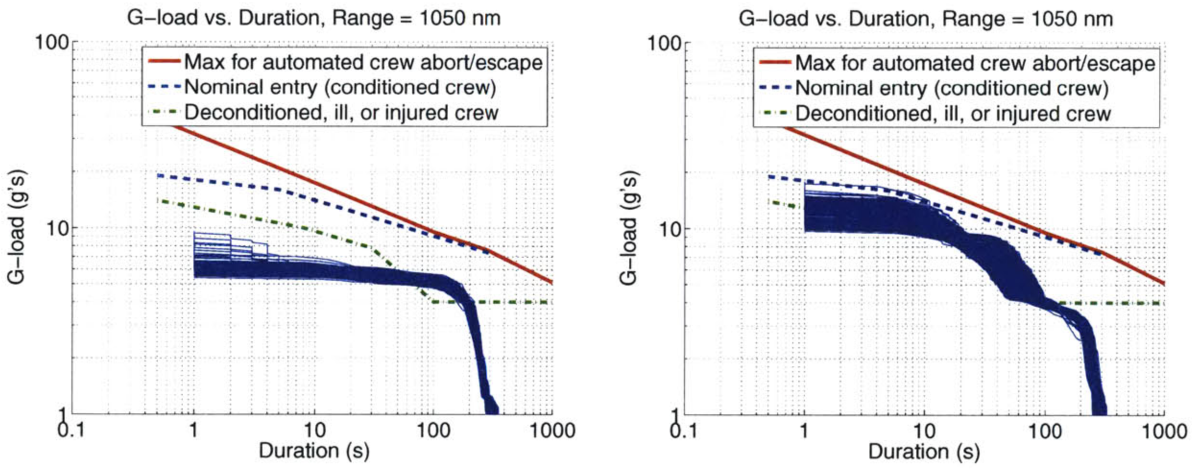


Figure 8-16: Duration-based g loads for 1050 nm target range: Predictor (left) and Draper (right)

8.5.3 Maximum Heat Rate

Figure 8-17 shows the peak heat rate data for the Monte Carlo trials. The overall increase in maximum heat rate in the Predictor algorithm, particularly for the shorter target ranges, can be explained by the shallow entry test (see Section 6.4). The Draper baseline commands full lift up at 0.05 g's for every entry, while the Predictor algorithm commands larger bank angles during Initial Roll if the Constant Drag level needs to be increased or the transition to Upcontrol will occur below a certain drag level. These larger bank angles during the initial descent into the atmosphere will increase the peak heat rate. As the target range increases and the Predictor algorithm no longer needs to command larger Initial Roll bank angles, the maximum heat rate behavior becomes essentially the same as the Draper baseline.

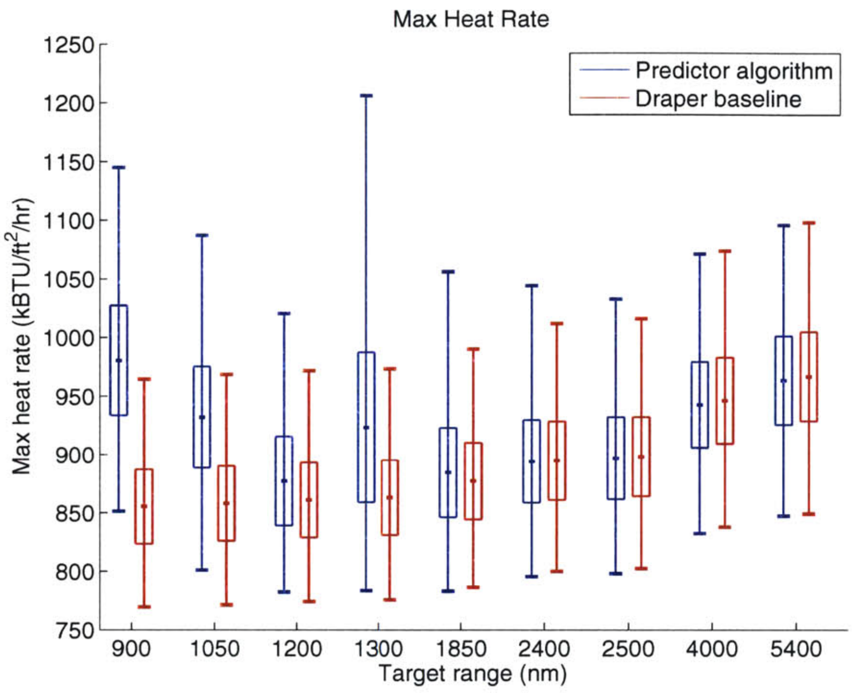


Figure 8-17: Summary of maximum heat rate for all Monte Carlo target ranges

8.5.4 Total Heat Load

As Figure 8-18 shows, average total heat load increases as the target range increases, due to the longer total trajectory time. For the short target ranges, the Draper baseline algorithm significantly overshoots the landing site, resulting in a longer trajectory and a higher total heat load. For the 1200 *nm* and 1300 *nm* targets, the algorithms display similar heat load characteristics because they have similar trajectory times and are flying at similar altitudes.

For the long target ranges, the Predictor algorithm has higher total heat loads because the average bank angle during the skip is larger than the Draper algorithm's average skip bank angle. This keeps the trajectory in the denser portion of the atmosphere. However, Section 8.2.2 shows that the Predictor algorithm can be "tuned" to reduce the total heat load by adjusting the reference skip bank angle, whereas the original PredGuid implementation cannot.

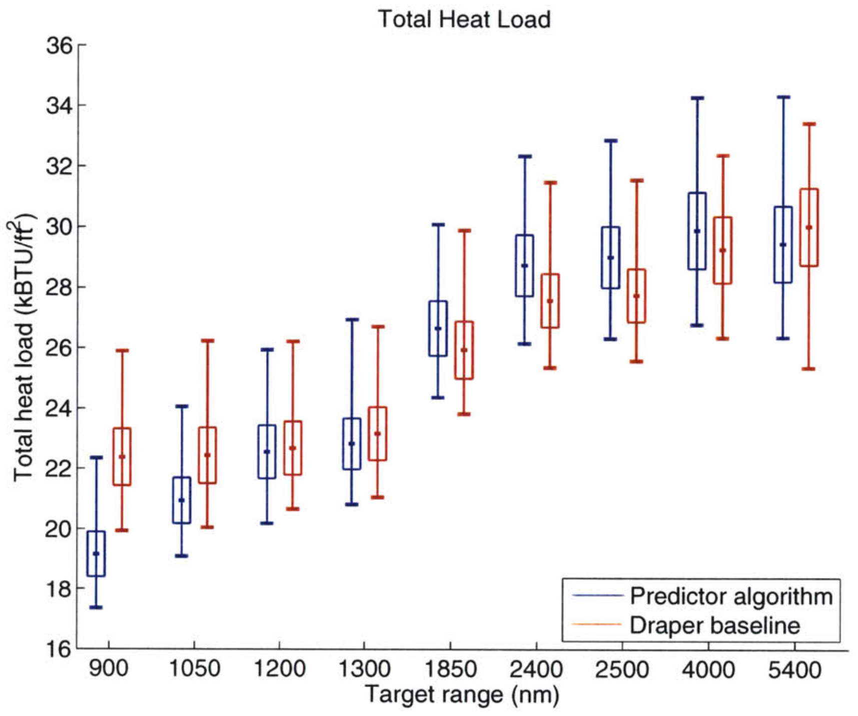


Figure 8-18: Summary of total heat load for all Monte Carlo target ranges

8.5.5 Control Saturation

A major objective in designing the Predictor algorithm was to improve robustness. One way to measure robustness is by looking at control saturation. The data presented in this section is separated into two groups—lift up and lift down. In order to compare directly between algorithms and target ranges, saturation is measured for each trial as the total length of time the controller is commanded its extreme bank angles, regardless of whether it is continuous. Saturation is defined as anytime the bank command is $\phi = 0°$ or $\phi = \pm 180°$, and when the bank command is limited by Lateral Logic to $\phi = \pm 15°$ or $\phi = \pm 165°$. Saturation is not measured during the Initial Roll and Huntest phases (the energy management phases); it is only measured during the Upcontrol, Ballistic, and Final phases (the targeting phases). Additionally, any saturation during the portion of the Ballistic phase in which PredGuid is turned off is not included in this data. This is because the bank angle command is not being updated while PredGuid is off, regardless of its value.

Lift Up

Figure 8-19 shows that the amount of saturation at lift up saturation has been reduced all target ranges above 900 *nm*. The Predictor algorithm's improvement is moderate for the 1300 *nm* and 1850 *nm* target ranges, but it is significant for the target ranges longer than 1850 *nm*.

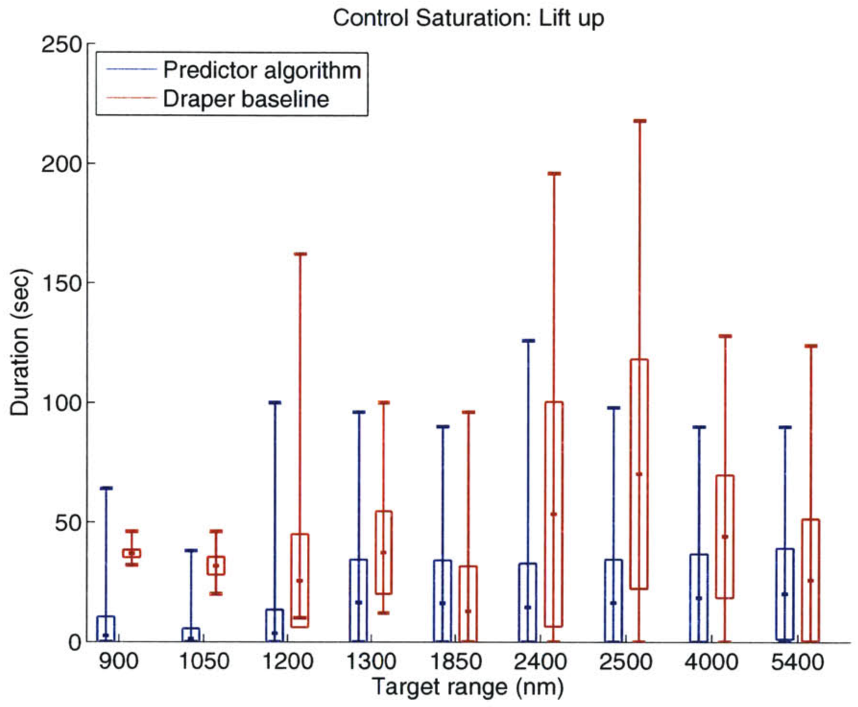


Figure 8-19: Summary of lift up saturation for all Monte Carlo target ranges

The amount of lift up saturation is smaller for the direct entry cases, which is to be expected because of the shorter range. However, the Predictor algorithm shows similar average saturation values for the loft and skip target ranges, whereas the amount of control saturation using the Draper baseline shows some amount of correlation with increasing target range.

The slight increase in control saturation for the Predictor algorithm at 2400 *nm* may be explained by the thick-to-thin density shear that is a result of the different atmosphere models used in the guidance system and the simulation environment. Figure 8-21 shows that the atmospheric density scale factor increases during the loft portion of the trajectory, indicating that the actual density is higher than expected. To compensate, PredGuid must command steadily smaller bank angles to compensate for the energy lost by flying through a thicker atmosphere (see Figure 8-21). By the time the loft is completed, the vehicle is short of Final phase reference trajectory, shown in Figure 8-22, requiring it to fly lift up for a larger portion of the Final phase to increase the range.

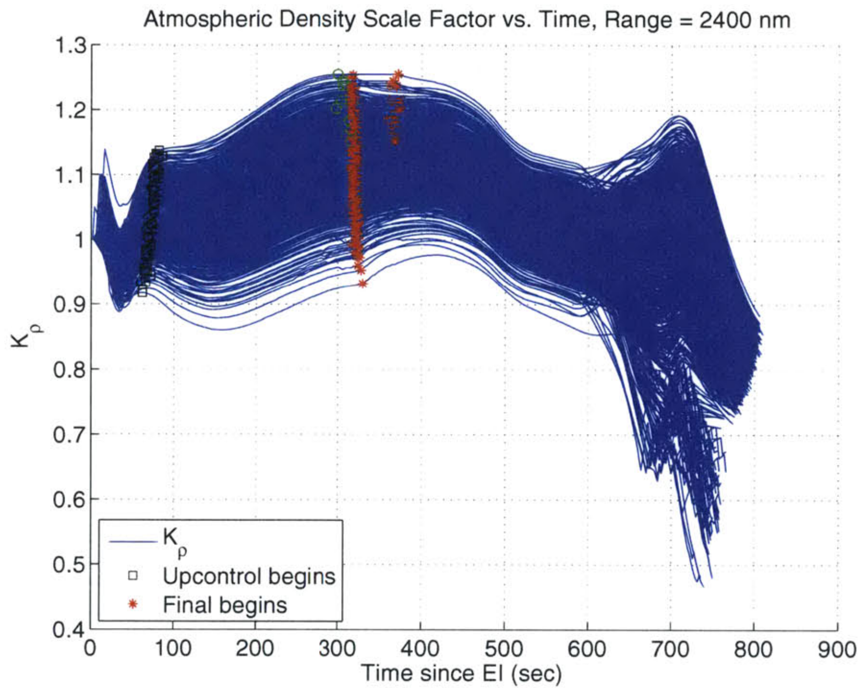


Figure 8-20: Atmospheric density scale factor, 2400 *nm* target range

Lift Down

Figure 8-23 shows the lift down control saturation for the all of the Monte Carlo target ranges. The Predictor algorithm shows clear improvement in this area over the Draper baseline algorithm.

As expected, the time spent commanding full lift down is generally longer for the short target ranges. The Predictor algorithm's slight increase in lift down saturation for the 4000 *nm* range is a result of PredGuid's inaccuracies for short objective distances. In 40% of the trials for this target range, PredGuid's objective distance is over 200 *nm* when it is restarted just before the second entry. PredGuid commands erratic bank angles, shown in Figure 8-24,

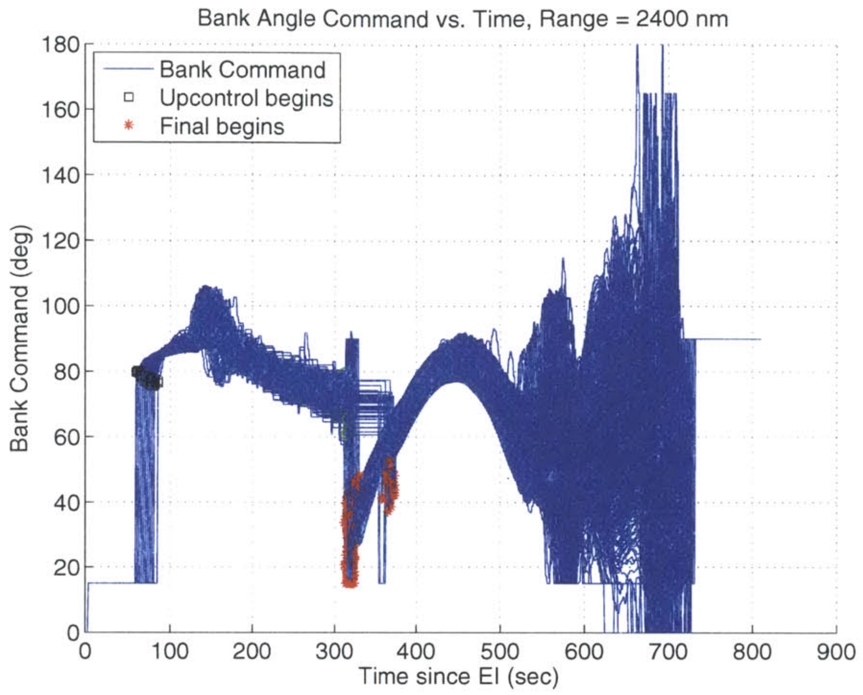


Figure 8-21: Bank angle command history, 2400 nm target range

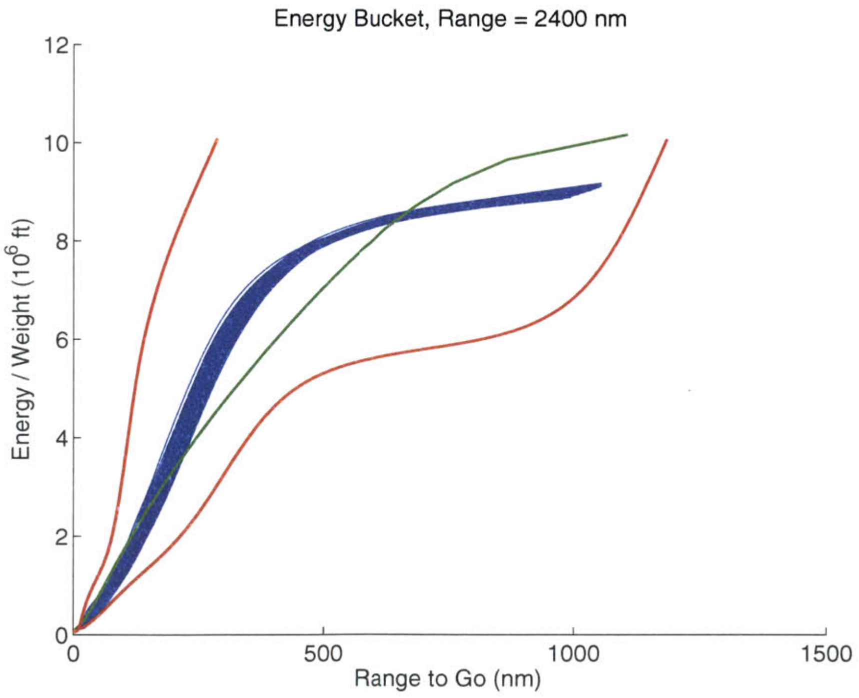


Figure 8-22: Final phase energy bucket, 2400 nm target range

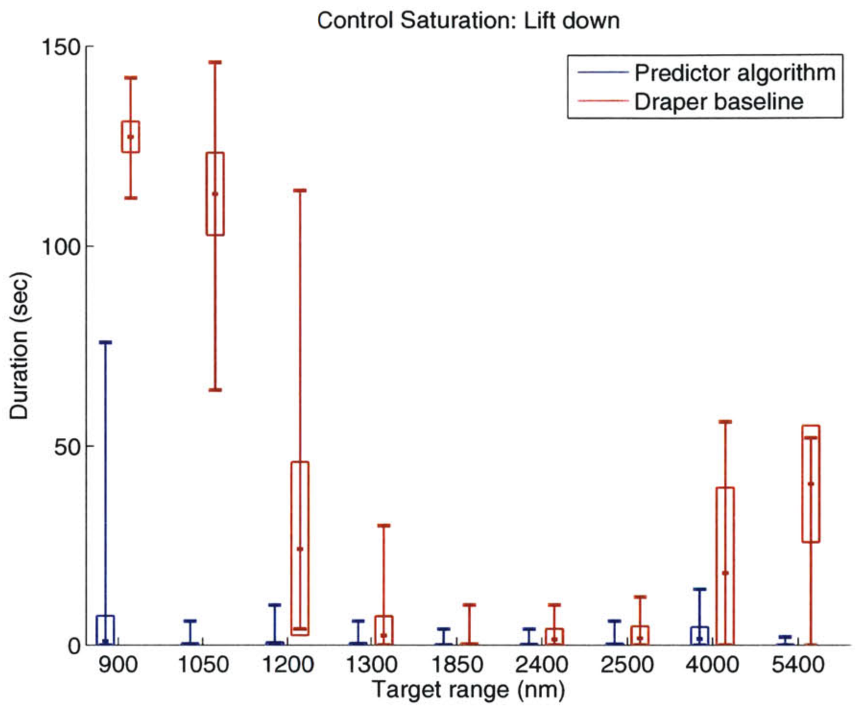


Figure 8-23: Summary of lift down saturation for all Monte Carlo target ranges

until the objective distance falls below 200 nm and PredGuid's solution is replaced with a lift neutral command. This phenomenon also occurs using the Draper baseline, and raising PredGuid's objective distance cutoff could be a temporary solution, but PredGuid should ultimately be updated to provide better solutions for short objective distances.

8.5.6 Energy Bucket

Another way to measure robustness is by examining the Final phase energy bucket. There are three desirable characteristics to look for in the Monte Carlo trajectories, when viewed in these plots:

1. The trajectories are near the center of the energy bucket. This gives the most margin from either boundary.

2. The trajectories are tightly clustered. This shows that the algorithm has been successful at eliminating large variations prior to the Final phase.

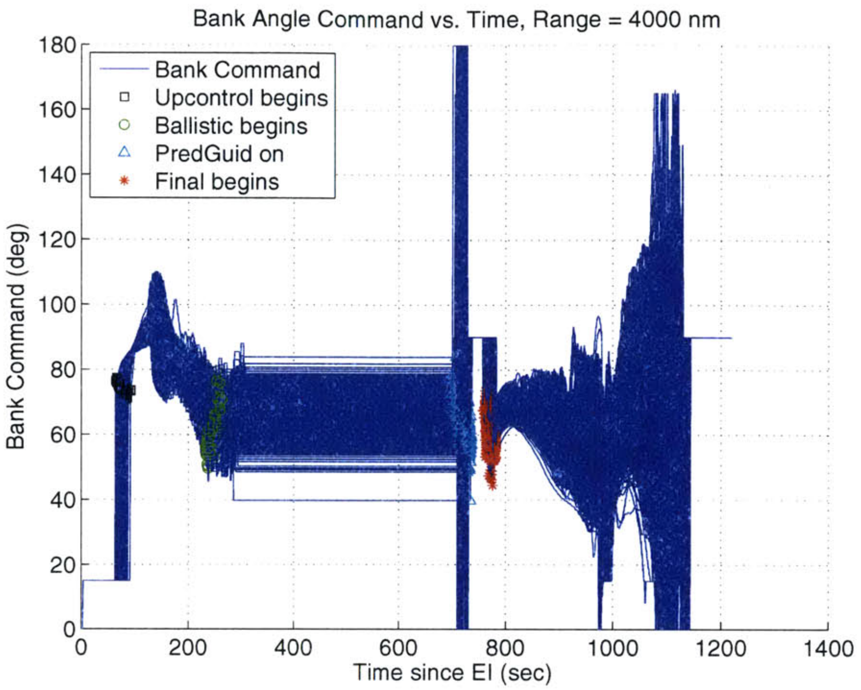


Figure 8-24: Predictor algorithm bank command history, range = 4000 *nm*

3. The trajectories are following the reference trajectory. This ensures that the controller is successfully guiding the vehicle to the ground.

This section presents results for the following target ranges: 1050 *nm*, 1300 *nm*, 1850 *nm*, and 4000 *nm*. 1050 *nm* shows the direct entry capability, and 1300 *nm* shows the Predictor algorithm's ability to transition between direct entry and loft. 1850 *nm* is a midrange loft, and 4000 *nm* shows the skip entry capability. The results for the remaining target ranges are included in Appendix A.

1050 *nm*

Figure 8-25 shows the direct entry energy bucket for the 1050 *nm* target range for both algorithms.

The results from the Predictor algorithm (left) show that the trajectories are very tightly clustered and are in the center of the energy bucket. There are two reference trajectories on the left plot because the algorithm chose the 6 *g* reference for some trajectories and the 7 *g* reference for the rest. Using the Draper baseline algorithm (right), the vehicle transitions to

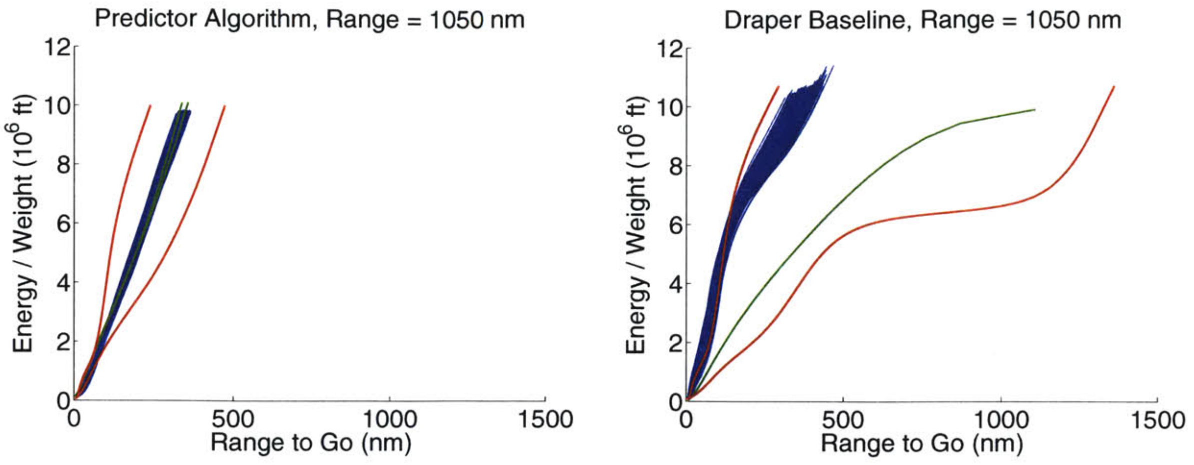


Figure 8-25: Energy buckets for 1050 *nm* target range

the Final phase quite close to the lift down boundary of the energy bucket and is unable to follow the reference trajectory.

1300 *nm*

The results from the 1300 *nm* target range, presented in Figure 8-26, show the Predictor algorithm's ability to fly direct entry or loft, depending on the particular trial's dispersions. The Draper algorithm flies all trajectories at this range as direct entries.

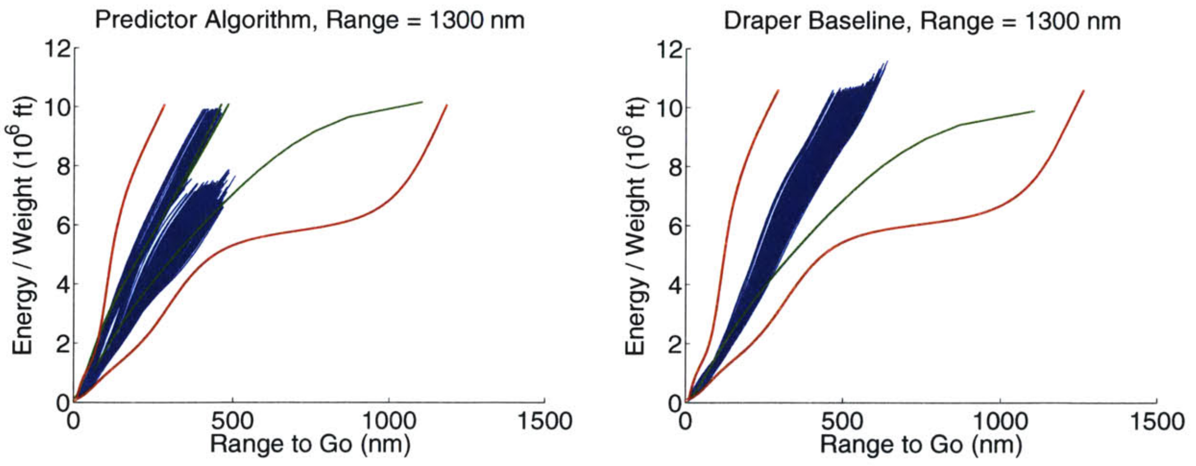


Figure 8-26: Energy buckets for 1300 *nm* target range

The direct entry cases follow either the 3.5 *g* reference or the 4 *g* reference, and the loft cases follow the original Draper reference. While the direct entry cases look very similar to the results shown in Figure 8-25, the loft cases are quite different. They transition to the Final phase very low in the energy bucket and are not tightly clustered. While some "spread" is expected in the energy and range to go at the start of the Final phase, this is generally reduced as the trajectories descend further into the bucket. Because these loft cases transition to the Final phase with such low energy however, there is less time to eliminate errors. This may explain why the landing accuracy for this target range is slightly worse than the other ranges (see Figure 8-12). The loft regime's performance using the Predictor algorithm is worst at these short ranges and improves as the target range increases.

These results from these short range loft cases suggest that the Predictor's loft logic may need further study. One possible improvement would be to stretch the direct entry downrange capability slightly so the loft regime is only used for longer target ranges in which it demonstrates more robustness. This could be implemented by simply allowing a larger error between the predicted and desired ranges during the EMT Predictor's direct entry range estimation. The results from this target range could also merely indicate that a single reference skip bank angle may not be suitable for both loft and skip entries. A smaller reference skip bank angle for short loft cases would force the Initial Roll bank angle to be larger in order to meet the same range requirement. This would increase the drag on the vehicle during the initial descent into the atmosphere, thereby increasing the amount of energy dissipated before the loft begins. This would allow the loft to have a higher altitude (less "skimming" through the atmosphere) and potentially improve its performance.

Preliminary results using a reference skip bank angle of 70° instead of 80° visibly improve the Predictor algorithm's performance at 1300 *nm*. Figure 8-27 shows the energy bucket for the loft cases at 1300 *nm*, as well as the improved landing accuracy using this change. In the energy bucket, the trajectories are more tightly clustered and are more closely aligned with the reference trajectory. This implies that more analysis may be necessary to determine the best reference skip bank angle for skip and loft cases at various target ranges.

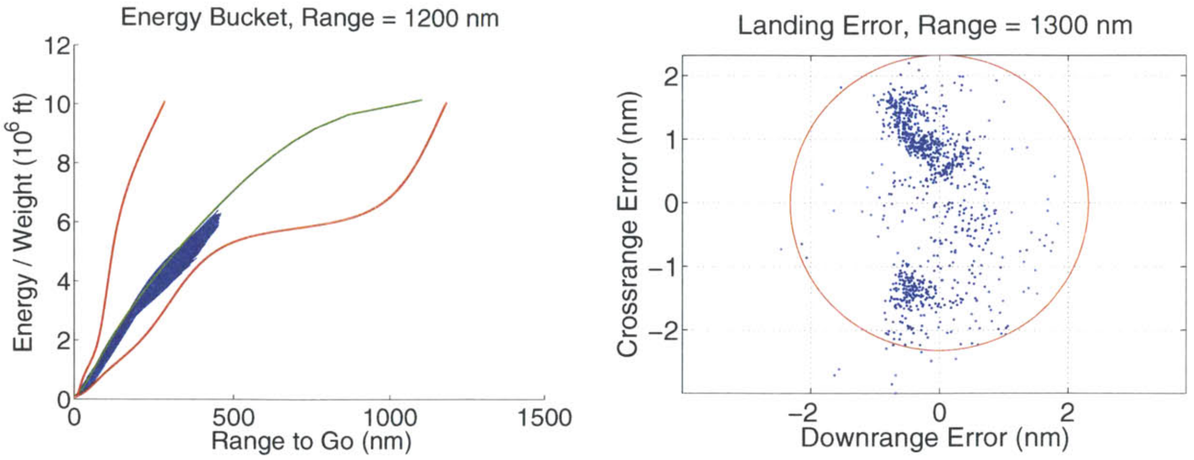


Figure 8-27: Energy bucket and landing accuracy for 70° skip bank angle, range = 1300 *nm*

1850 *nm*

The 1850 *nm* target range is the middle of the Predictor algorithm's loft range, so all of the trajectories follow the original skip reference. The loft performance has improved with a longer target range—the trajectories are tightly clustered and very near the center of the energy bucket. Using both algorithms, the transition to the Final phase occurs lower in the energy bucket than the start of the reference trajectory (which was designed for the maximum skip distance). This phenomenon occurs because the trajectories are losing energy throughout loft, due to their low altitude. In a skip, the vehicle retains energy by flying outside the atmosphere in order to extend the range. This increases the vehicle's energy at the start of the reference trajectory (see Figure 8-29).

The Draper baseline algorithm is better suited for longer target ranges than short target ranges, as the results in Figure 8-28 show. The trajectories are displaying a shape similar to the reference.

4000 *nm*

The 4000 *nm* target range is representative of typical ballistic skip trajectories. The trajectories shapes are much closer to the reference trajectory, which was designed for a 5400 *nm* target range, and both algorithms show that the trajectories are near the center of the bucket. The Predictor enhancements show improvement in the "tightness" of the trajecto-

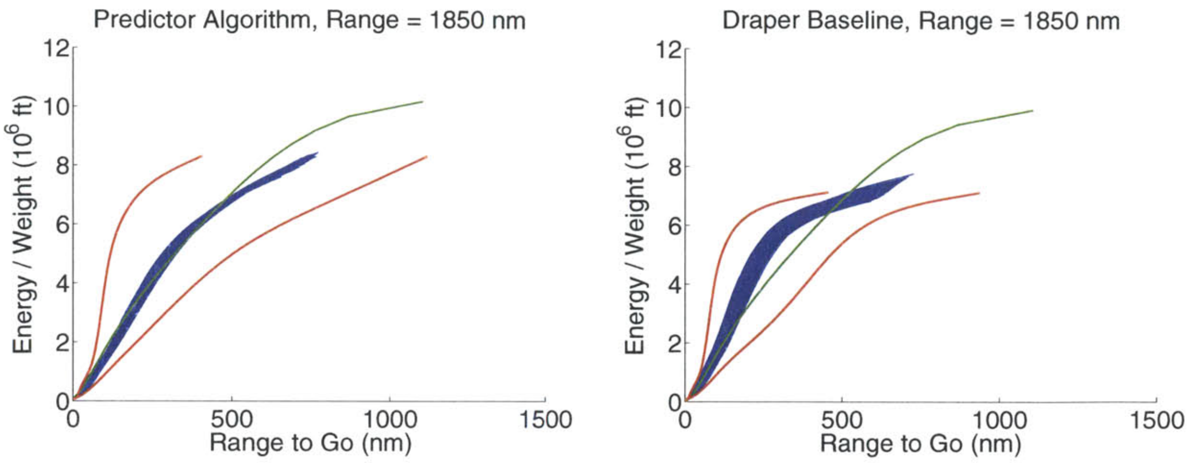


Figure 8-28: Energy buckets for 1850 *nm* target range

ries, however, which implies an increase in robustness.

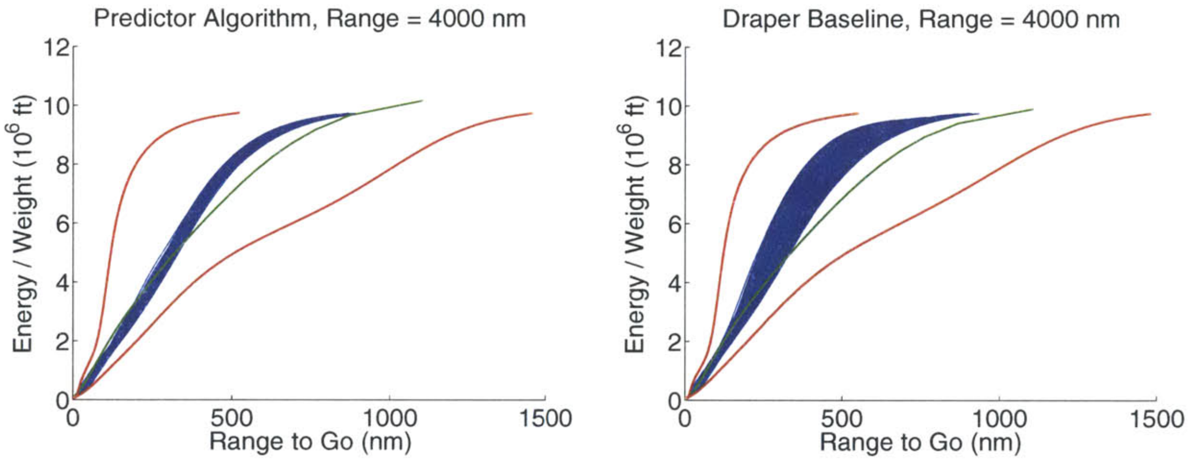


Figure 8-29: Energy buckets for 4000 *nm* target range

Chapter 9

Conclusion

9.1 Summary

The objective of this thesis was to develop an atmospheric entry guidance algorithm, based on the original Apollo algorithm and the PredGuid enhancements made by Draper Laboratory, which features improved energy management and phase transition logic, as well as reduced dependency on empirical relationships. The resulting algorithm is suitable for both direct entry and skip trajectories, as well as the intermediate range of "loft" trajectories. These are low altitude, short range skips in which the vehicle never exits the atmosphere. The energy management logic has been expanded to include the Energy Management and Transition (EMT) Predictor, which is responsible for determining 1) whether a skip is necessary to reach the target landing site, and 2) the appropriate transition time to the next phase, depending on the trajectory type. This adaptable energy management policy increases the direct entry capability of the algorithm. This algorithm also uses new Final phase reference trajectories that are specifically designed for direct entry cases. The best Final phase reference is selected during flight from a set of stored trajectories.

The EMT Predictor enhancements increase the algorithm's direct entry downrange capability by 300 *nm*. Results from Monte Carlo analysis show a significant reduction in the amount of control saturation and improved robustness throughout the entry, particular during the skip and final descent phases. The Predictor algorithm noticeably decreases deceleration loads and total heat load for short range entries, and a moderate decrease in deceleration loads is observed for the longer target ranges. Finally, the aerodynamic heating and load characteristics of a skip trajectory, using this algorithm, can be optimized to meet a particular objective by adjusting the reference skip bank angle.

9.2 Future Work

The performance of this algorithm demonstrates the feasibility of this type of atmospheric entry guidance, and it has revealed several areas that merit future study which are listed below. It should be noted that this algorithm was tested with a capsule whose L/D is moderately high; the EMT Predictor algorithm may exhibit different capability with a lower L/D vehicle.

Skip Bank Angle Selection

For a nominal skip entry using this algorithm, the trajectory prior to the Final phase is shaped by the Initial Roll bank angle and the skip bank angle. The default values for these two phases are 15° and 80°, respectively. The goal for this thesis was to select a skip bank angle in the approximate center of the corridor. A more rigorous method to determine the optimal bank angles for these phases, however, is necessary. The optimal values will depend on the objective of a particular program or vehicle configuration, as well as the expected target range. For example, this could be to minimize g load, total heat load, or total trajectory time.

Final Phase Reference Trajectory Parameters

This algorithm demonstrates that the performance of direct entries is improved using Final phase reference trajectories whose initial conditions match the typical Huntest exit conditions. The number of direct entry reference trajectories used to generate the results in Chapter 8 however, is probably higher than necessary. A more complete analysis of the range of Final phase initial conditions and the relationship between initial drag, flight path angle, and velocity would likely yield a smaller set of improved references. Additionally, a dedicated Final phase reference trajectory for loft cases may improve the performance of this trajectory type.

Lateral Steering

Lateral Logic, the phase of the algorithm responsible for lateral steering, was not modified in the development of this algorithm. Because guidance uses one control, bank angle, to affect two channels, vertical and lateral, it is essential that these two channels are managed in tandem. Bank angle commands to control both channels should not be contradictory.

In this algorithm, Lateral Logic is prohibited from commanding bank reversals during the first two phases of the entry. This was done to maintain the desired energy decay rate, particularly during Constant Drag; however, this can degrade the crossrange landing accuracy for very short direct entries. The decision to prohibit reversals during these phases should be revisited in the future. The vehicle's fuel use is also directly tied to Lateral Logic. Reducing the number of bank reversals reduces the total fuel consumption throughout the entry. If the lateral corridor is set correctly, only one bank reversal is needed to eliminate crossrange error at landing, as shown in Figure 9-1. This could also be done by incorporating the bank reversal time into the EMT Predictor and adjusting it to meet the crossrange requirement.

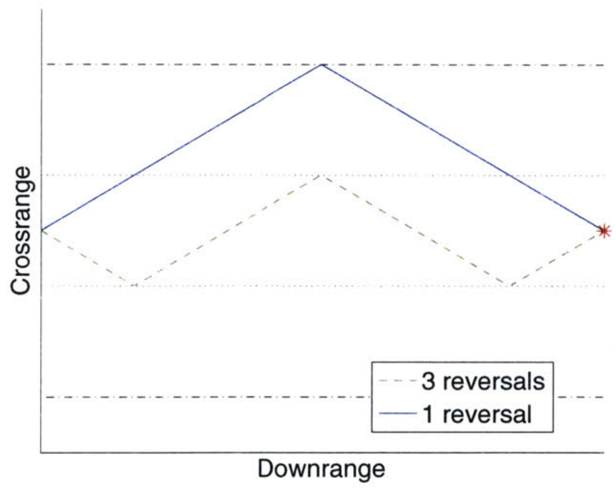


Figure 9-1: Meeting crossrange requirement using one bank reversal

Alternatives to Constant Drag

The energy depletion method of maintaining a drag level that is constant with time was used for the original Apollo algorithm, and it was retained for this algorithm. Constant Drag yields the lowest total heat load of the methods considered for the Apollo program [5], but it is a difficult flight regime to control. Other energy management methods should be investigated to determine their suitability for this application.

Controller for Transition to Final

The transition to the Final phase for direct entries is much smoother with the addition of the direct entry reference trajectories and the algorithm's ability to choose the best reference to match the predicted conditions. It is very rare, however, that the vehicle transitions to the Final phase at exactly the correct conditions to match the reference trajectory. Because of this, it may be beneficial to develop a controller to facilitate the transition to the Final phase.

Terminal Steering to Drogue Deploy Conditions

Neither the baseline nor the Predictor algorithm display acceptable landing accuracy in Monte Carlo simulations for the target ranges tested. This entry guidance algorithm, with and without the EMT Predictor enhancements, terminates once the relative velocity drops below 1000 ft/s; however, preliminary investigation shows that the landing accuracy is improved if this velocity cutoff is reduced, or if an additional controller is implemented to steer from 1000 ft/s to the drogue parachute deploy conditions. Naturally, the cost of this reduction in landing error is increased fuel usage; therefore, it is necessary to determine the relative importance of both metrics.

PredGuid Solution Near Transition to Final Phase

PredGuid continues to exhibit difficulty in providing accurate bank angle solutions near the transition to the Final phase, where PredGuid's target range is short. Possible solutions to this problem include reducing PredGuid's integration time step near the end of the skip, or reducing the sensitivity of the bank angle guess to the previous iteration's range error toward the end of the skip. If no acceptable adjustments to PredGuid can be made, the decision to output a neutral lift command at the end of the skip should be revisited. It is possible that a bank command more appropriate than 90° can be found, either by some method of extrapolating the Final phase reference trajectory up into the skip, or by selecting a bank command that is closer to either the Final phase constant L/D fraction or the reference skip bank angle.

Short Range Loft Trajectories

The Predictor algorithm's loft regime performs well for longer target ranges, but the extremely short ranges are more difficult to control, partly because these trajectories enter the Final phase at a very low energy level. This gives the Final phase controller very little time to guide the vehicle onto the reference trajectory and steer to the landing site. These short range trajectories may benefit from a reduction in energy before the beginning of the loft, which would increase the trajectory's altitude during the loft and may allow for better targeting of the Final phase reference trajectory. This energy reduction could be achieved by entering a short Constant Drag phase prior to beginning the loft or by setting a smaller reference skip bank angle for loft cases. A smaller skip bank angle forces an increase in the Initial Roll bank angle in order to meet the same range requirement. This larger Initial Roll L/D fraction will increase the vehicle's energy decay rate during its initial descent into the atmosphere.

Negative Test Subroutine

The Upcontrol phase transitions to the Negative Test subroutine after calling PredGuid to generate bank angle commands during the skip. Negative Test has been carried over from the original Apollo guidance, and its purpose is to increase the L/D command to zero if Upcontrol's desired L/D command is negative and the current drag is greater than 175 ft/s^2. This subroutine may not be suitable for the PredGuid implementation of Upcontrol, especially if the reference skip bank angle corresponds to a negative L/D fraction.

Appendix A

Additional Results

A.1 Predictor algorithm results

Table A.1: Landing accuracy, Predictor algorithm

Target Range (nm)	Downrange Error (nm)		Crossrange Error (nm)		97% CEP (nm)
	μ	3-σ	μ	3-σ	
900	0.75	5.12	-0.78	5.05	19.95
1050	0.17	1.16	0.79	2.06	2.60
1200	0.12	1.57	0.85	2.70	3.23
1300	-0.18	1.54	0.56	2.96	2.56
1850	-0.17	1.47	-0.47	3.52	2.47
2400	-0.14	1.68	-0.12	3.44	3.16
2500	-0.16	1.60	-0.21	3.49	2.79
4000	-0.08	1.76	-0.02	3.61	2.70
5400	-0.09	1.69	-0.22	3.59	2.92

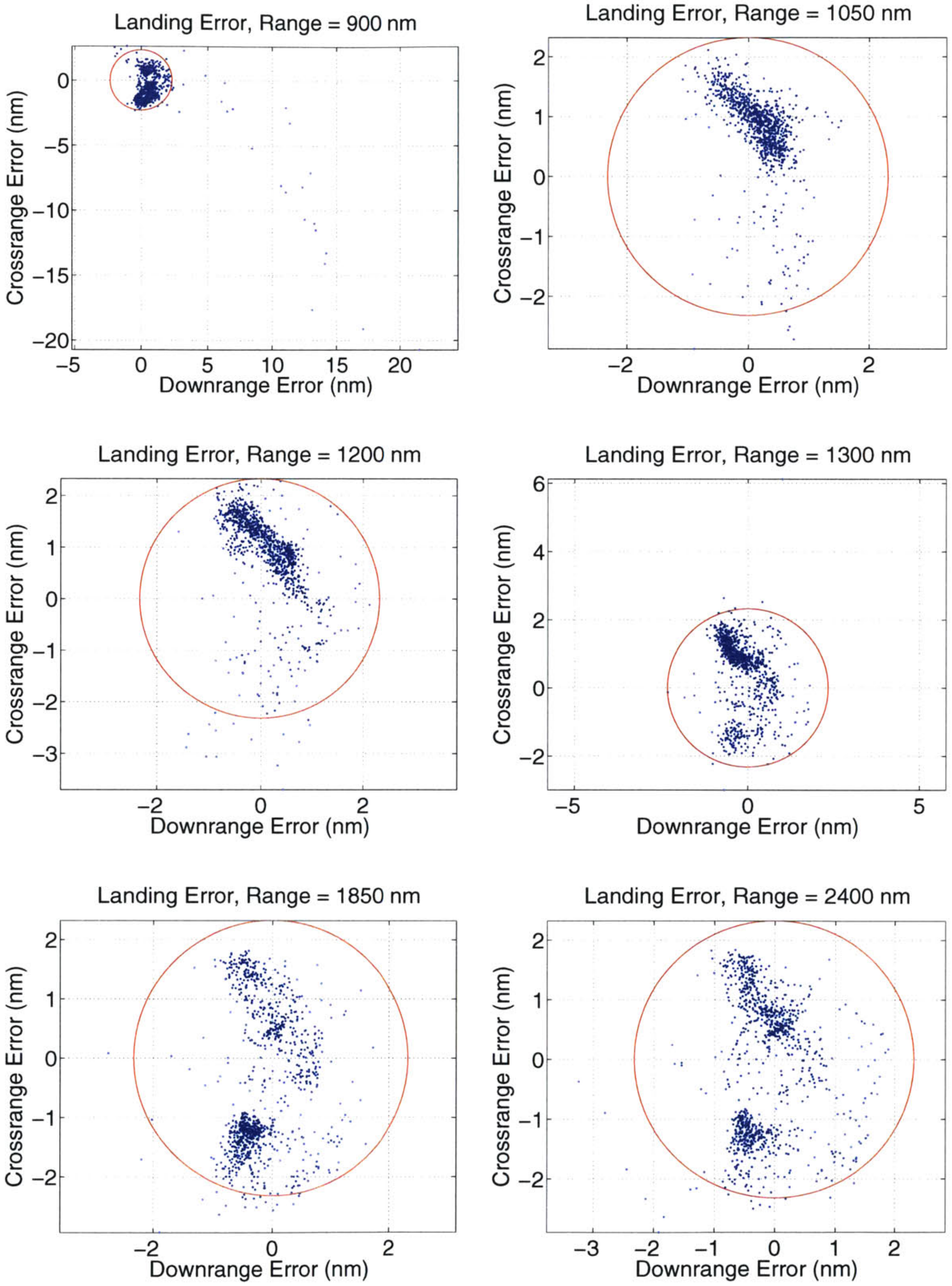


Figure A-1: Landing accuracy, Predictor algorithm, target ranges $900 - 2400\ nm$

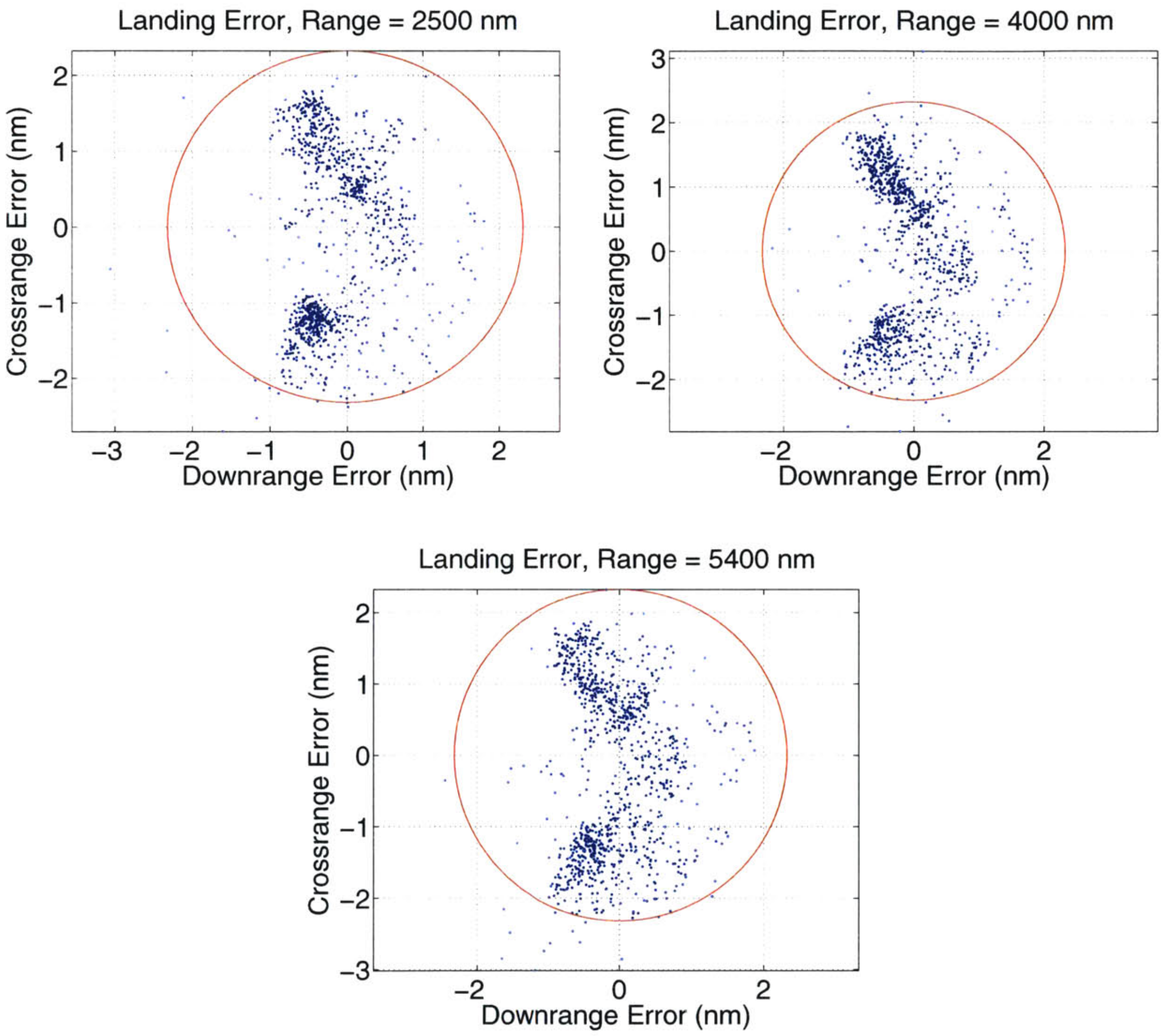


Figure A-2: Landing accuracy, Predictor algorithm, target ranges 2500 − 5400 *nm*

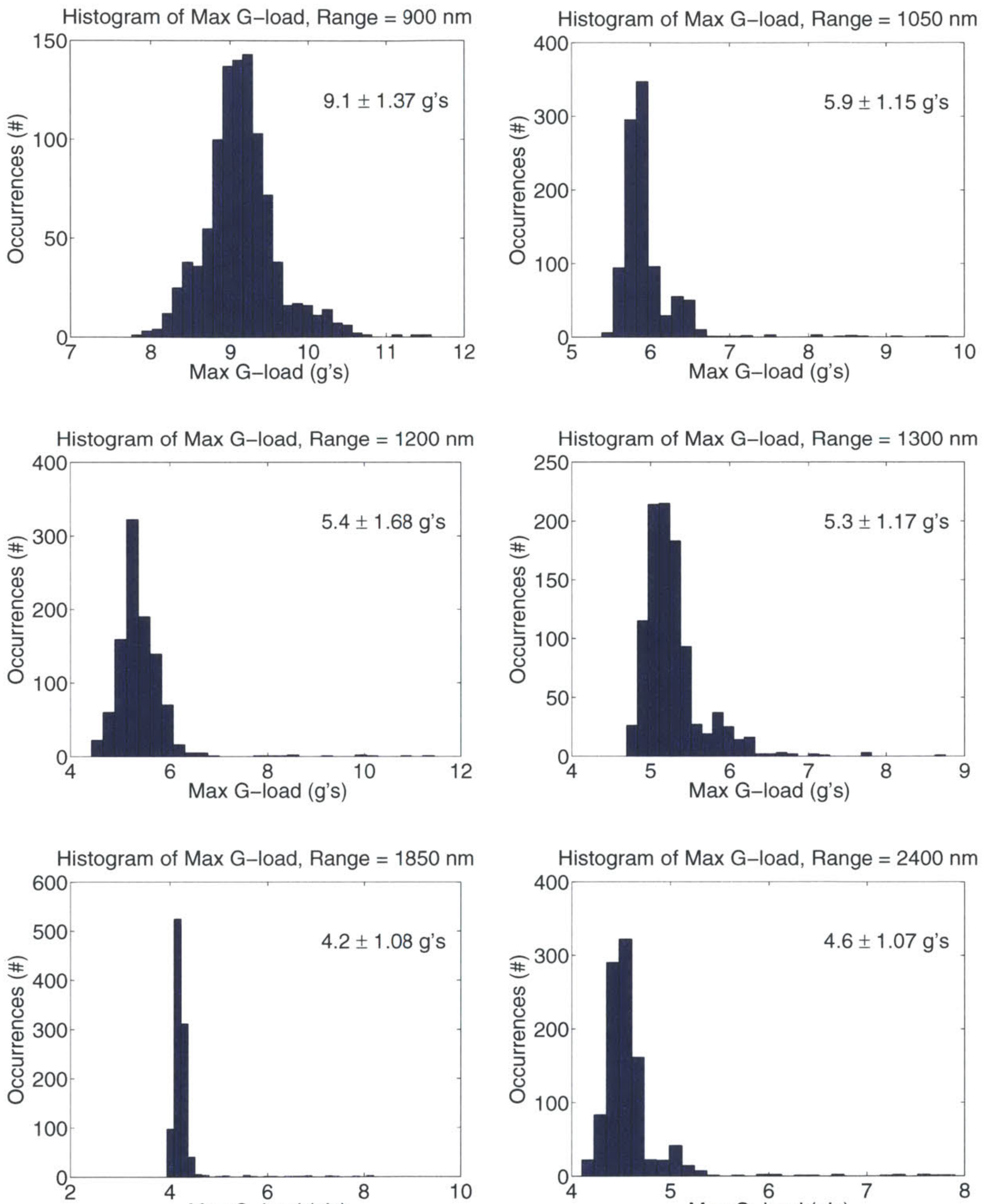


Figure A-3: Max g load histograms, Predictor algorithm, target ranges $900 - 2400$ nm

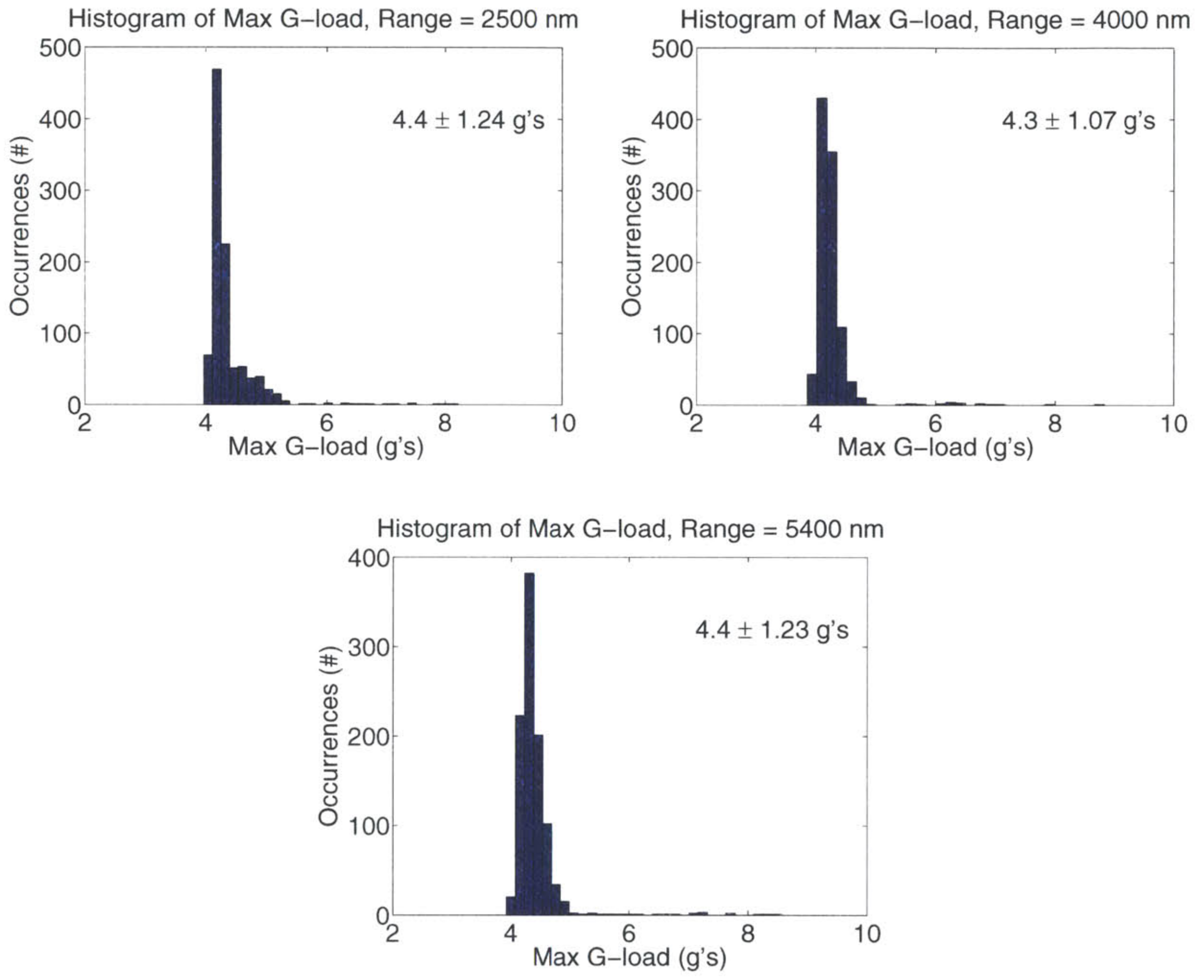


Figure A-4: Max g load histograms, Predictor algorithm, target ranges $2500 - 5400$ nm

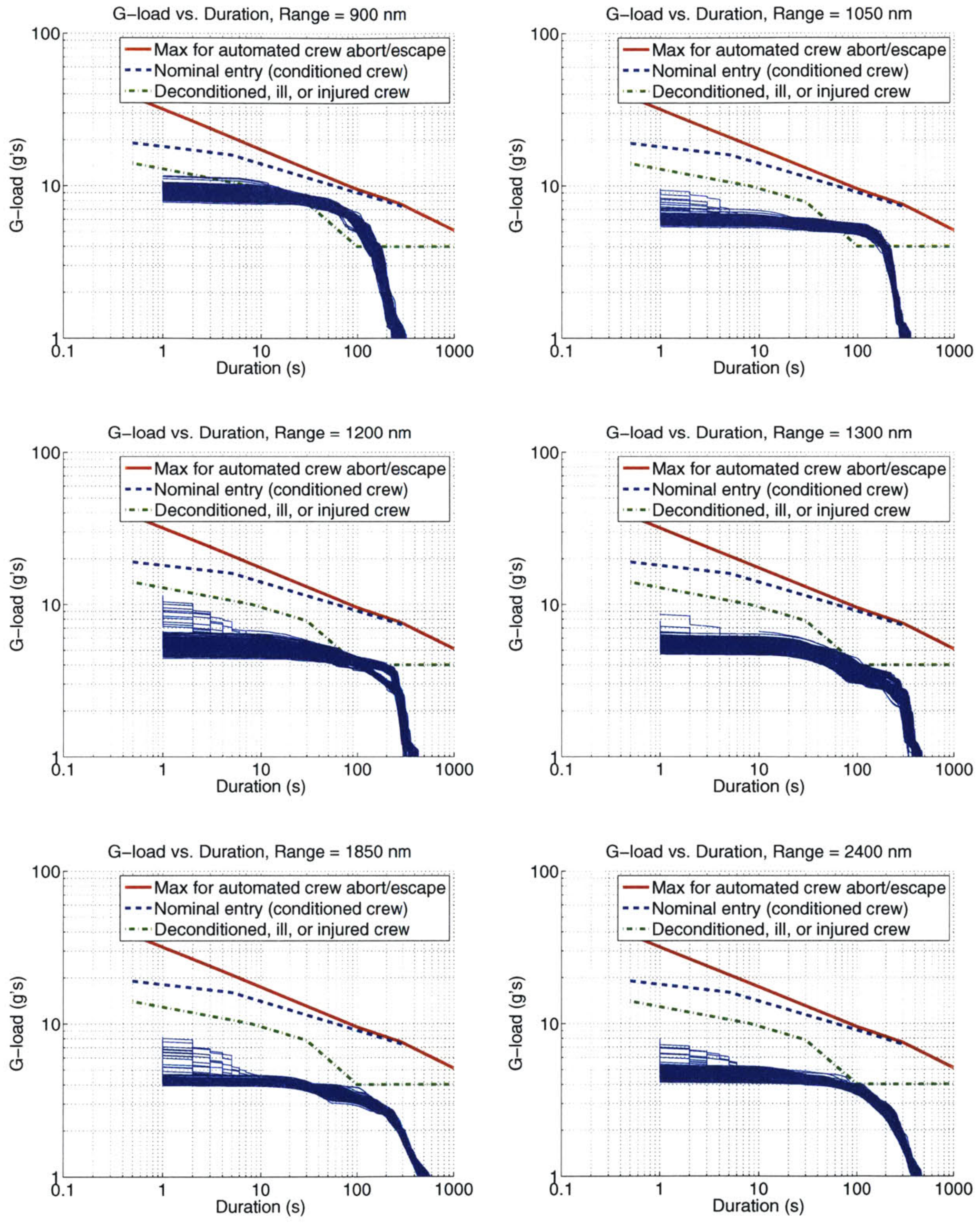
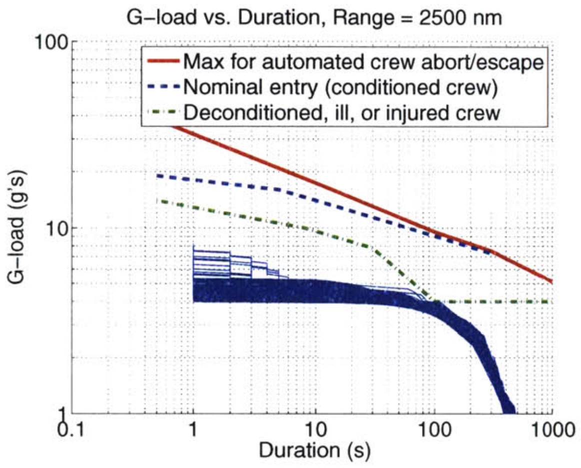
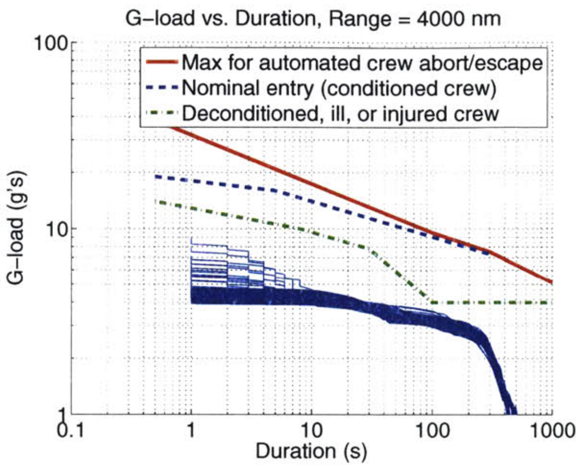


Figure A-5: Duration-based g loads, Predictor algorithm, target ranges $900 - 2400$ nm





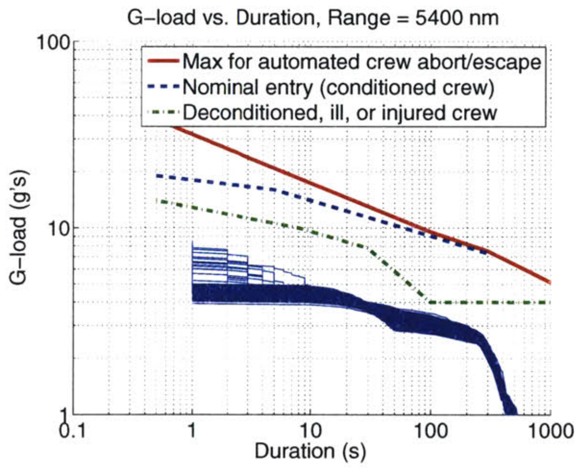


Figure A-6: Duration-based g loads, Predictor algorithm, target ranges $2500 - 5400$ nm

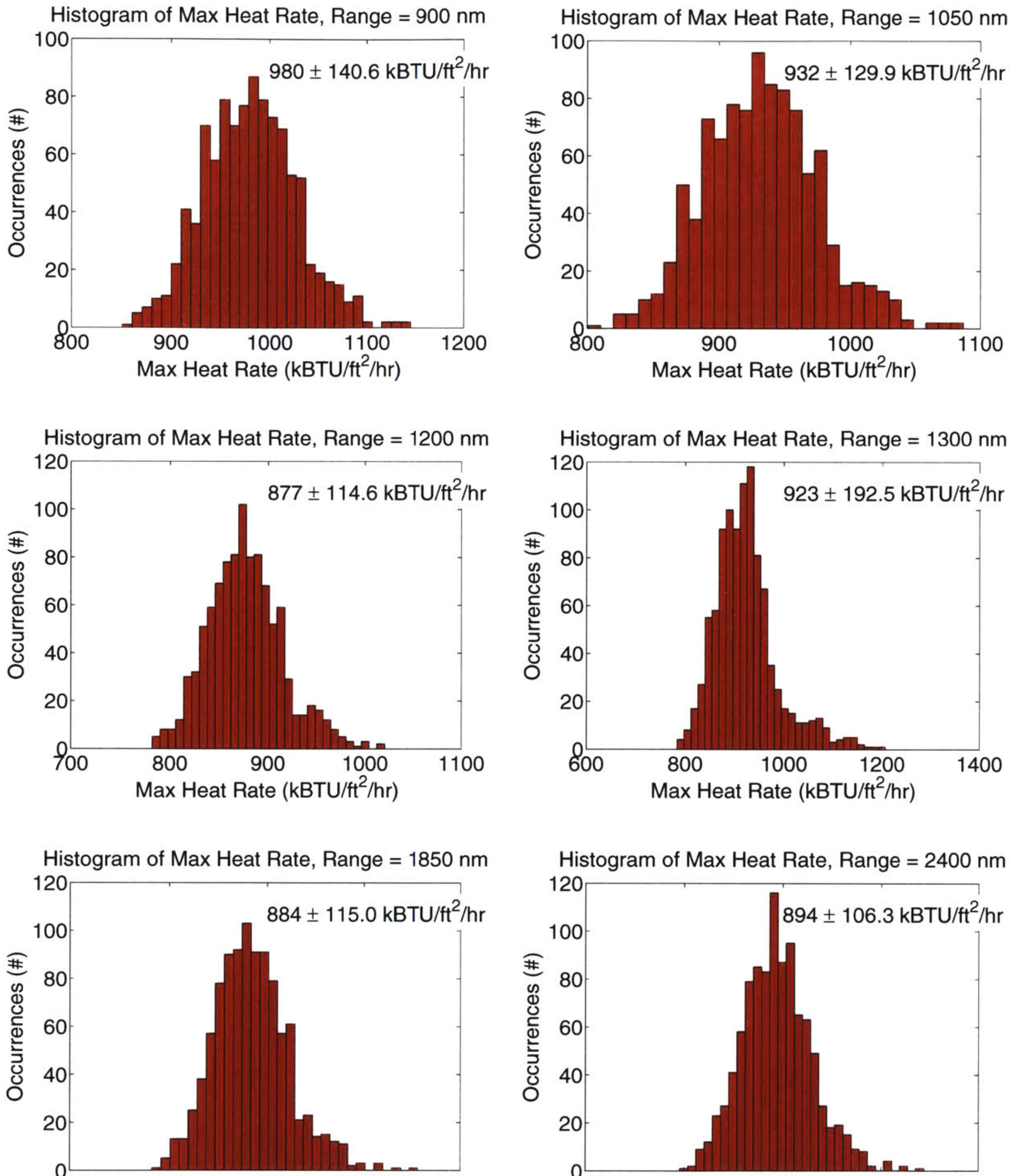
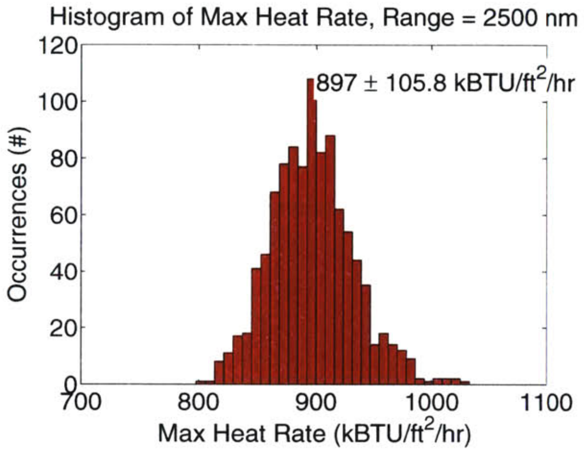
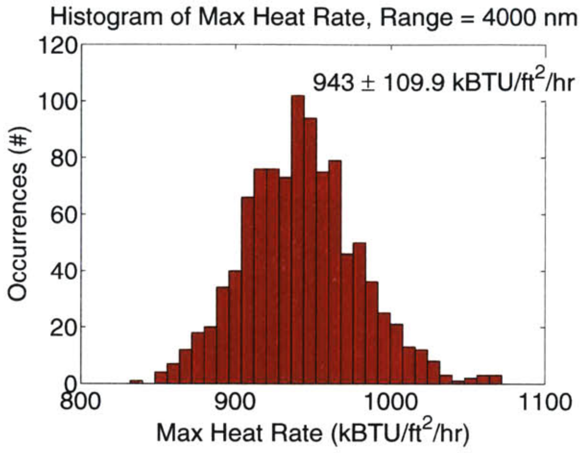


Figure A-7: Maximum heat rate histograms, Predictor algorithm, target ranges 900 – 2400 *nm*





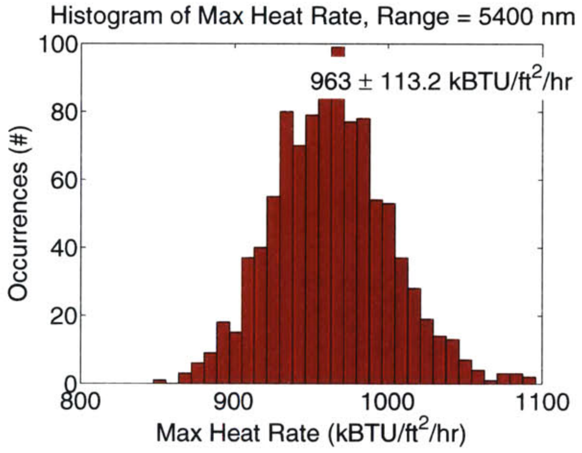


Figure A-8: Maximum heat rate histograms, Predictor algorithm, target ranges $2500 - 5400$ nm

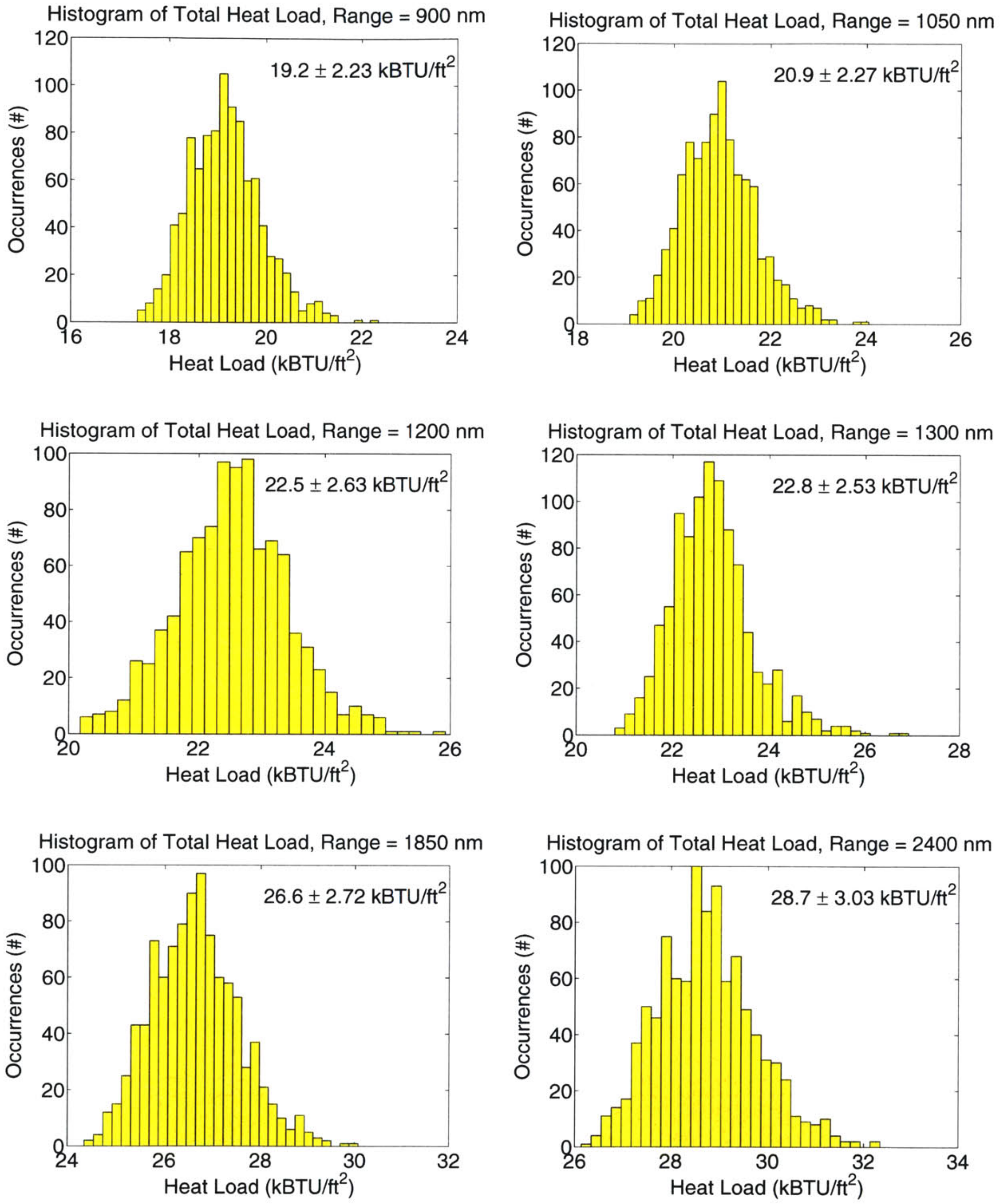


Figure A-9: Total heat load histograms, Predictor algorithm, target ranges 900 – 2400 *nm*

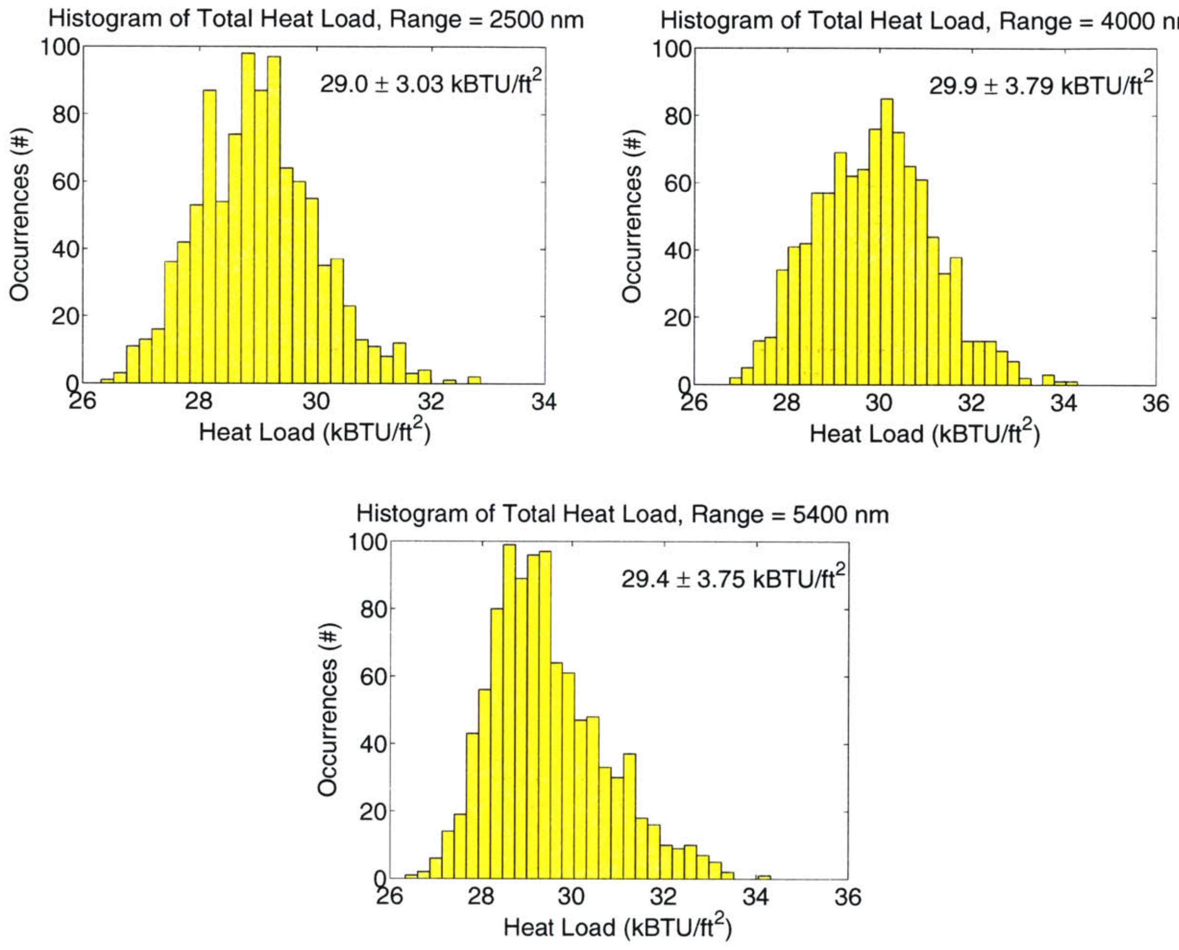


Figure A-10: Total heat load histograms, Predictor algorithm, target ranges $2500 - 5400$ nm

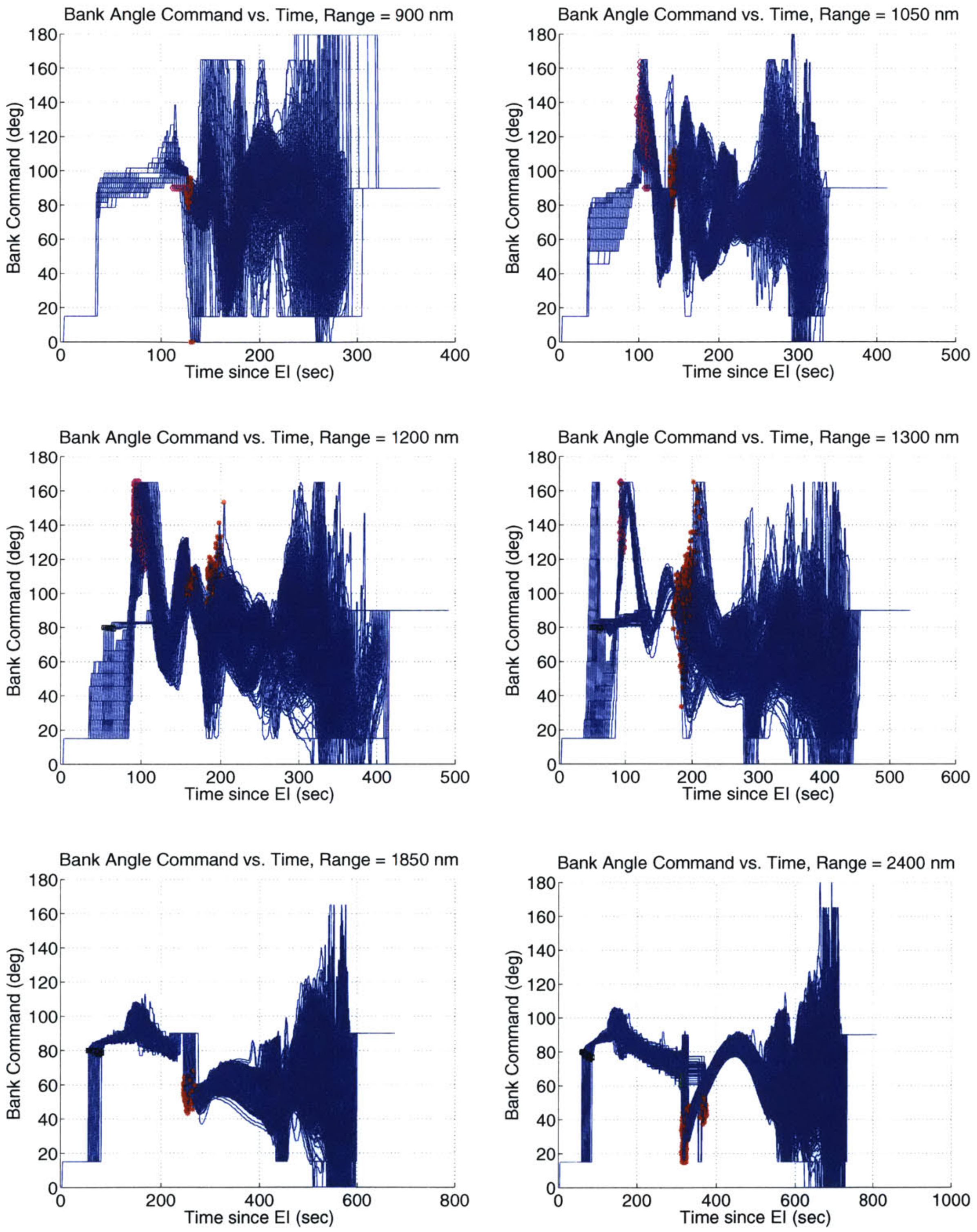


Figure A-11: Bank angle command history, Predictor algorithm, target ranges 900 – 2400 *nm*

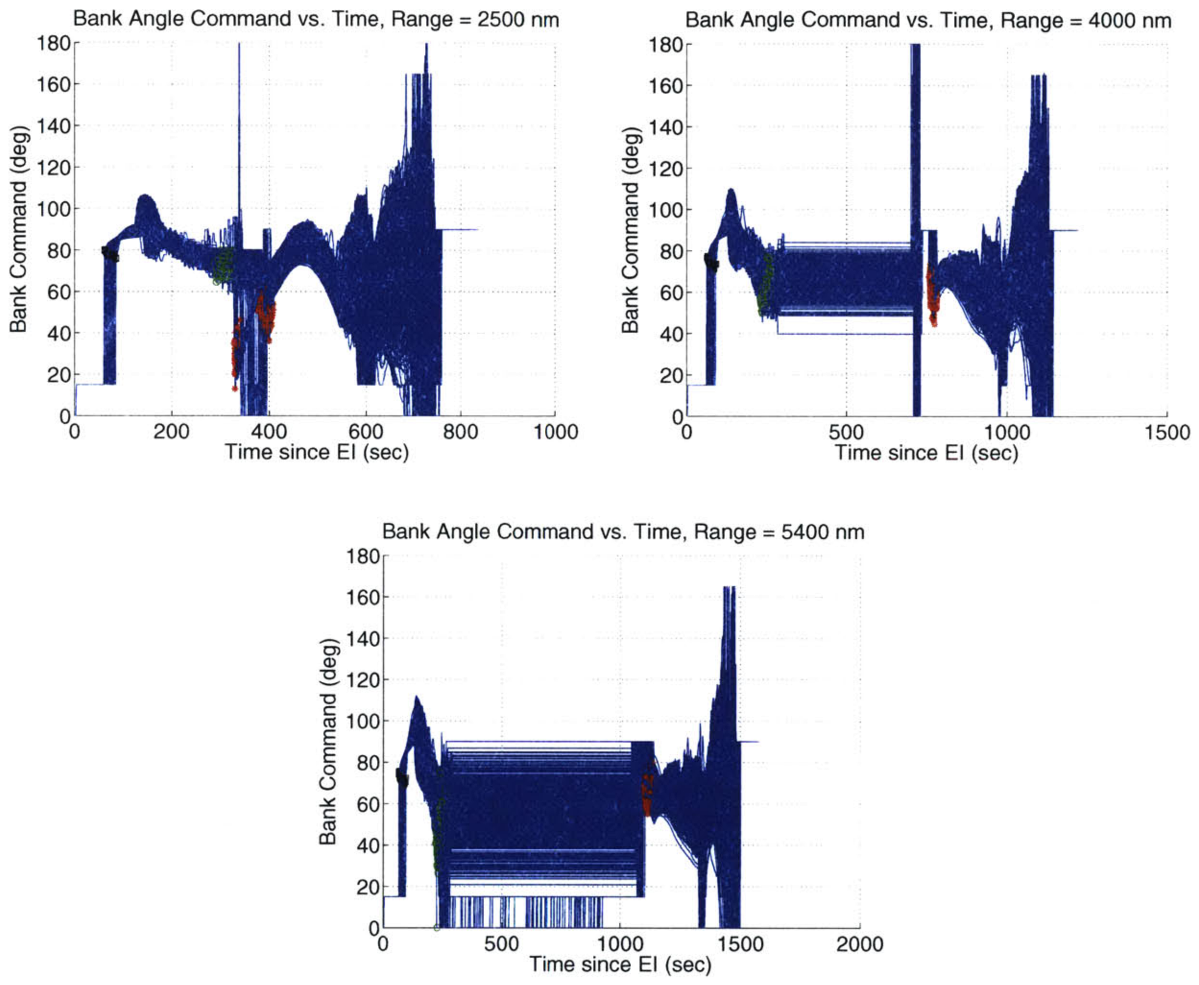


Figure A-12: Bank angle command history, Predictor algorithm, target ranges 2500 – 5400 *nm*

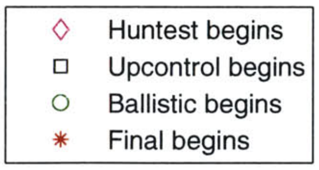


Figure A-13: Legend for bank angle command history figures

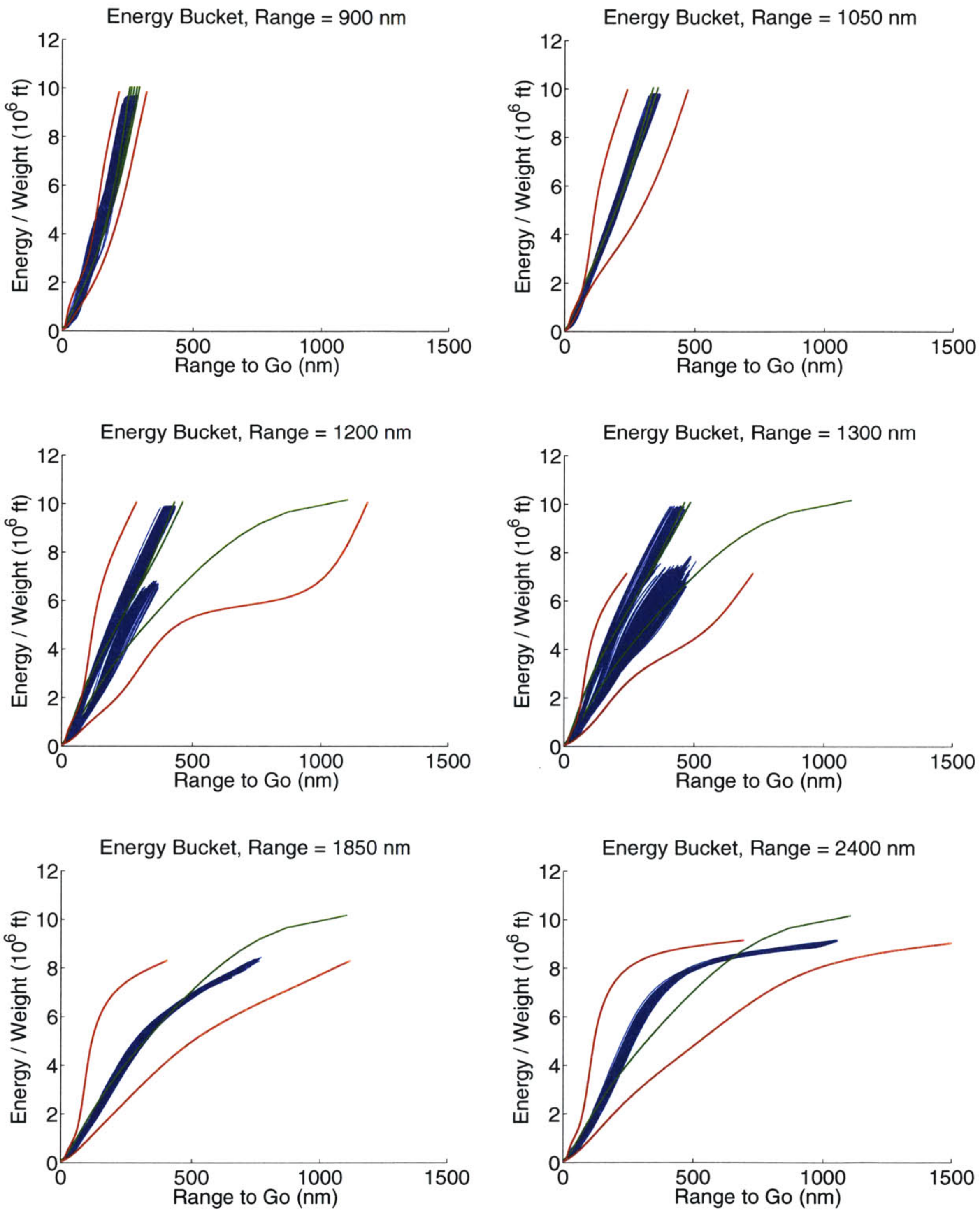
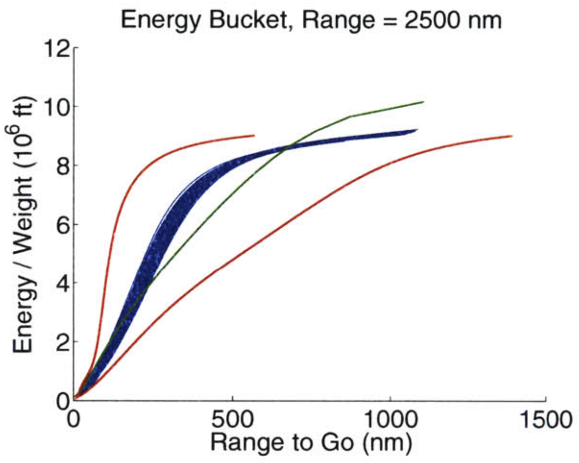
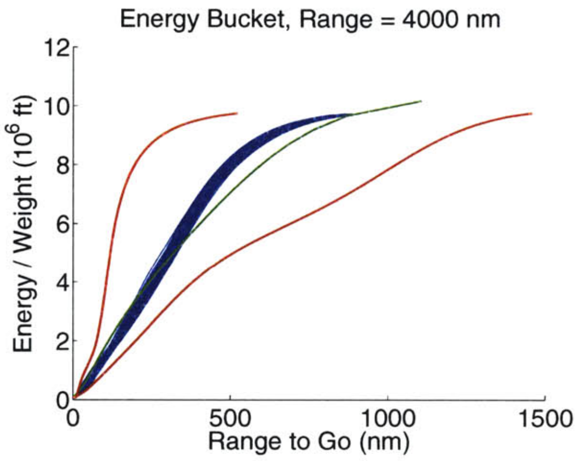


Figure A-14: Final phase energy buckets, Predictor algorithm, target ranges $900 - 2400$ nm





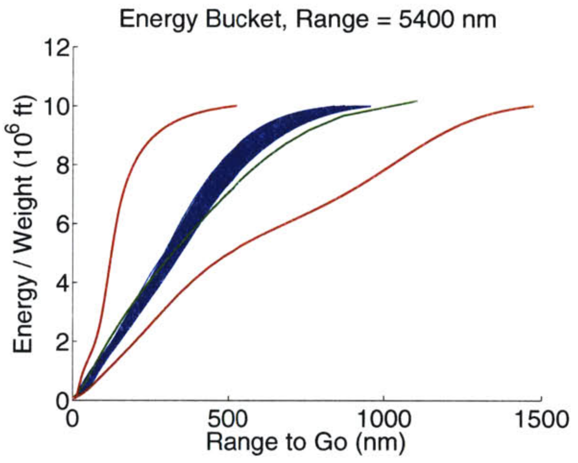


Figure A-15: Final phase energy buckets, Predictor algorithm, target ranges $2500 - 5400$ nm

A.2 Draper Baseline Algorithm Results

Table A.2: Landing accuracy, Draper baseline algorithm

Target Range (nm)	Downrange Error (nm)		Crossrange Error (nm)		97% CEP (nm)
	μ	3-σ	μ	3-σ	
900	249.70	65.23	-17.29	11.21	313.03
1050	100.51	58.58	-15.57	11.08	175.92
1200	1.06	9.05	0.17	7.25	21.01
1300	-0.15	1.49	0.65	2.54	2.17
1850	-0.23	1.38	0.63	2.96	2.51
2400	-0.19	1.63	0.25	3.25	2.89
2500	-0.19	1.63	0.16	3.34	2.91
4000	-0.12	1.75	-0.06	3.65	2.92
5400	-0.10	1.68	-0.30	3.64	2.80

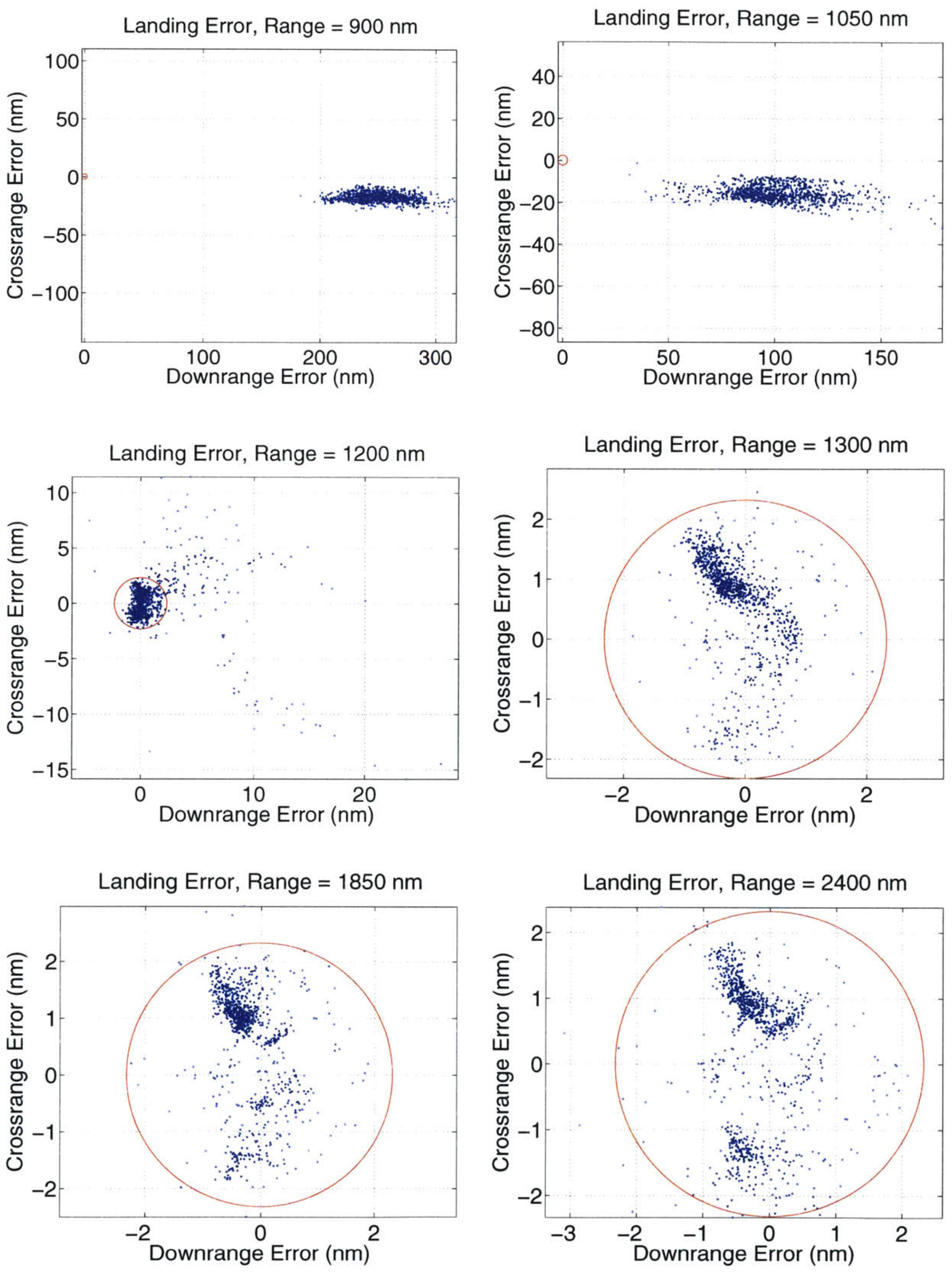


Figure A-16: Landing accuracy, Draper baseline algorithm, target ranges 900 – 2400 nm

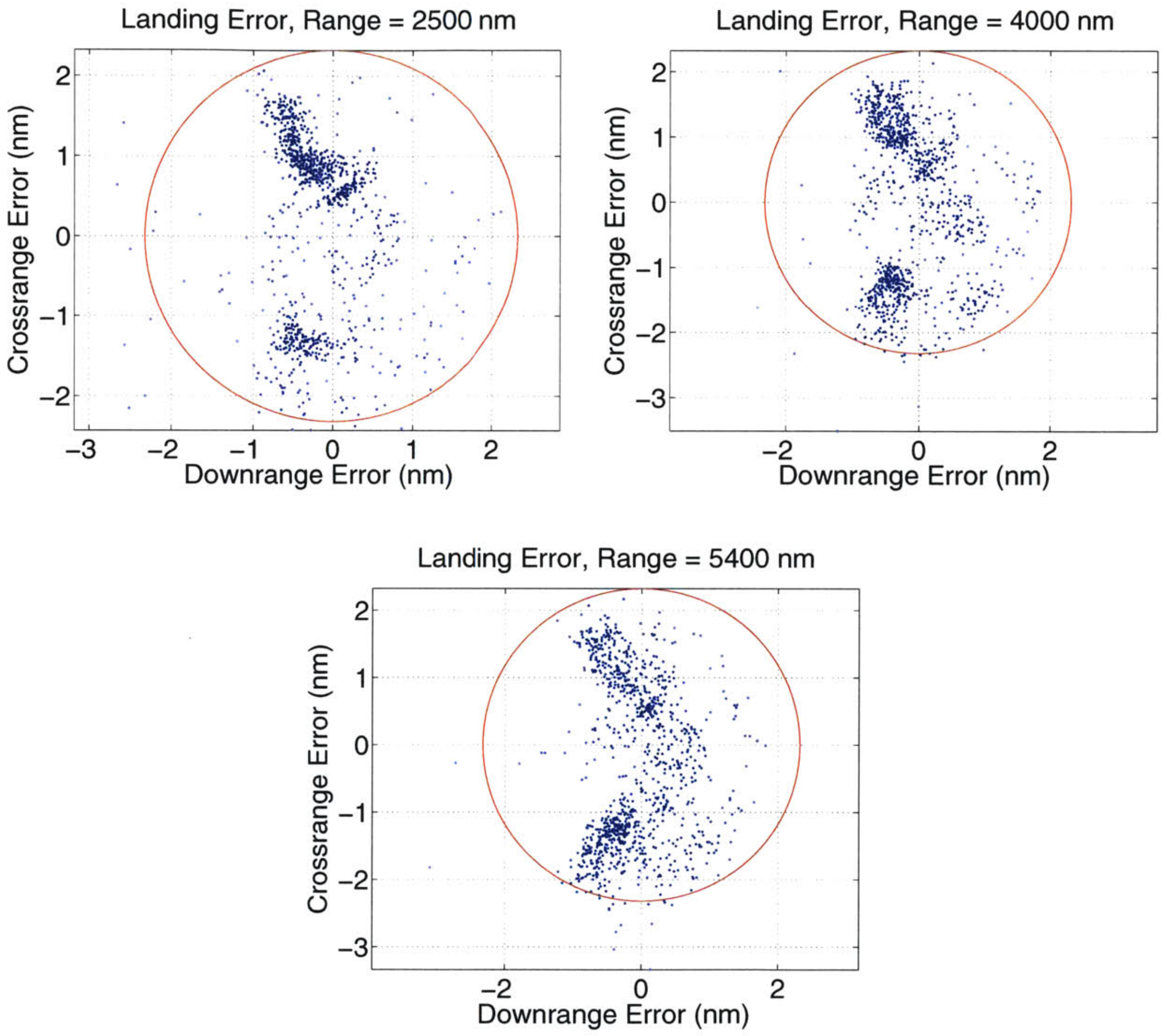


Figure A-17: Landing accuracy, Draper baseline algorithm, target ranges 2500 − 5400 nm

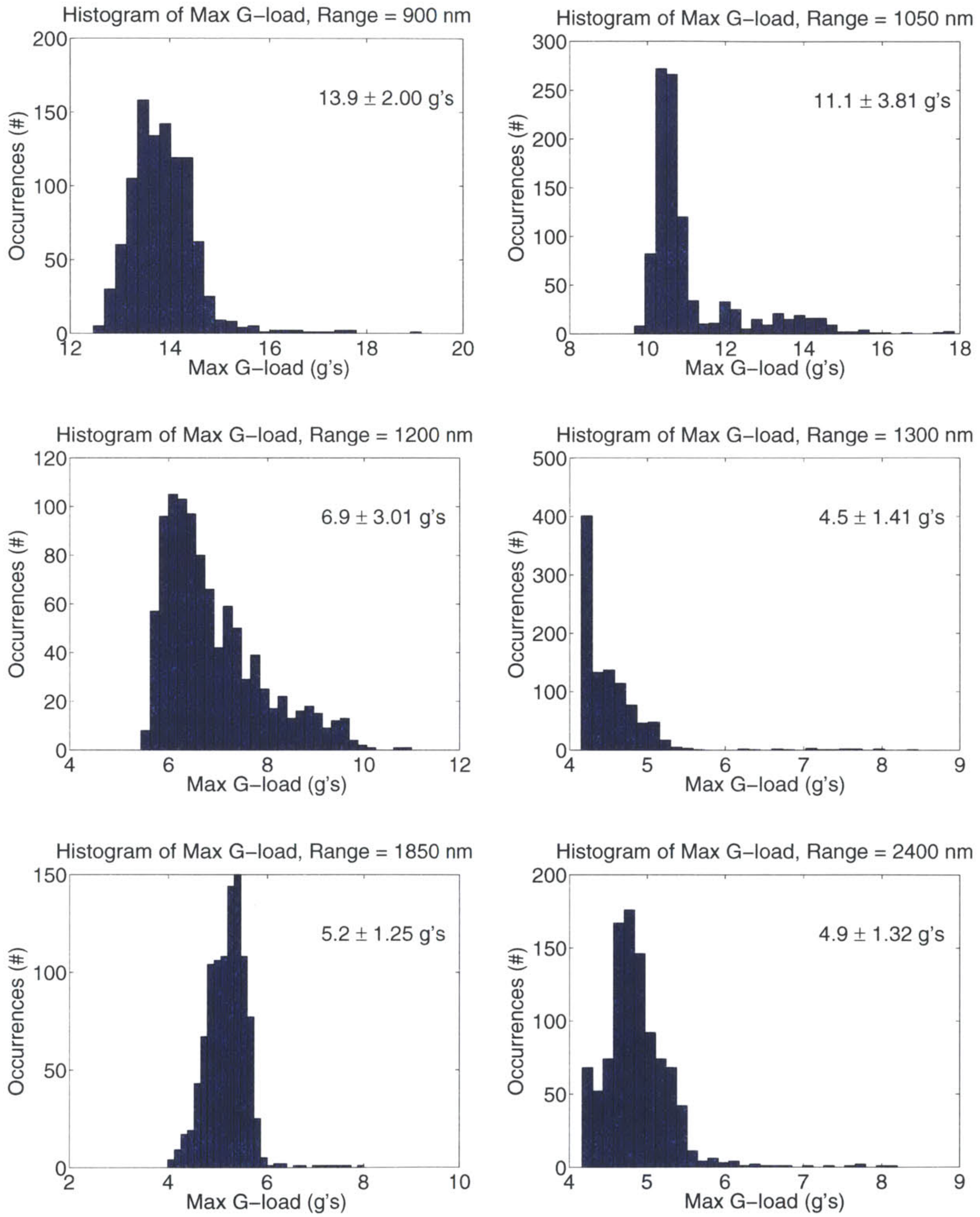


Figure A-18: Max g load histograms, Draper baseline algorithm, target ranges $900 - 2400$ nm

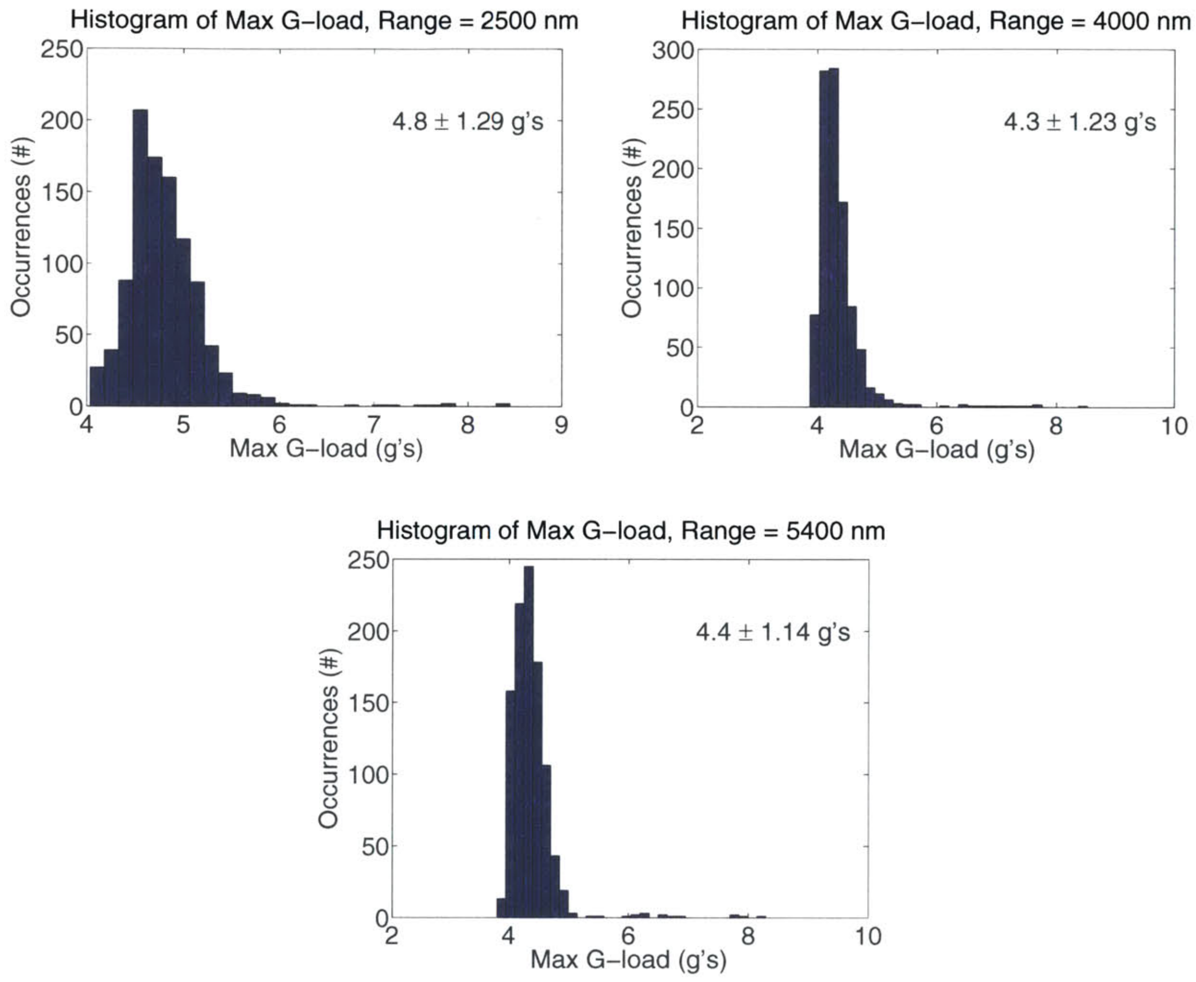


Figure A-19: Max g load histograms, Draper baseline algorithm, target ranges $2500 - 5400$ nm

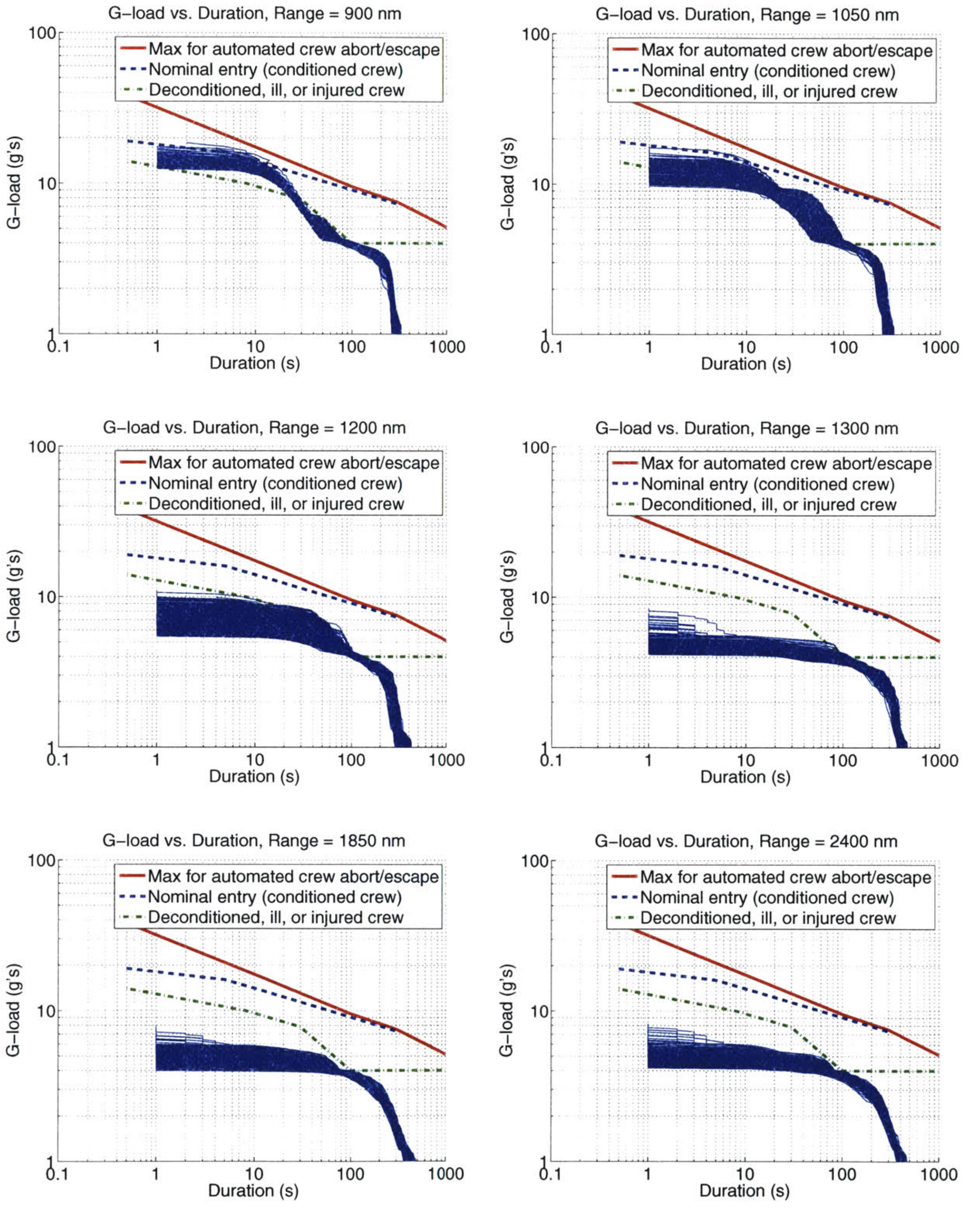


Figure A-20: Duration-based g loads, Draper baseline algorithm, target ranges 900 – 2400 nm

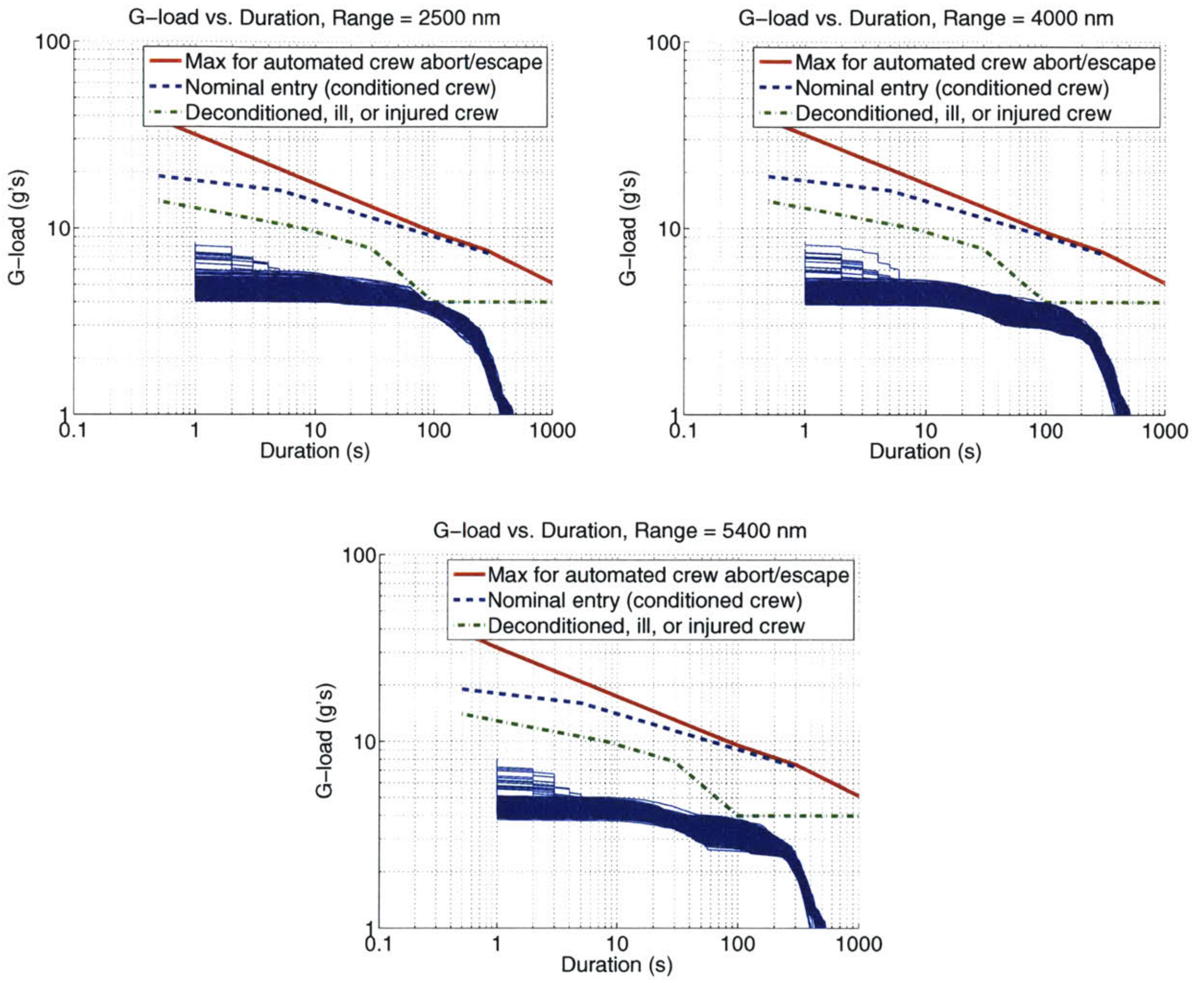


Figure A-21: Duration-based g loads, Draper baseline algorithm, target ranges $2500 - 5400$ nm

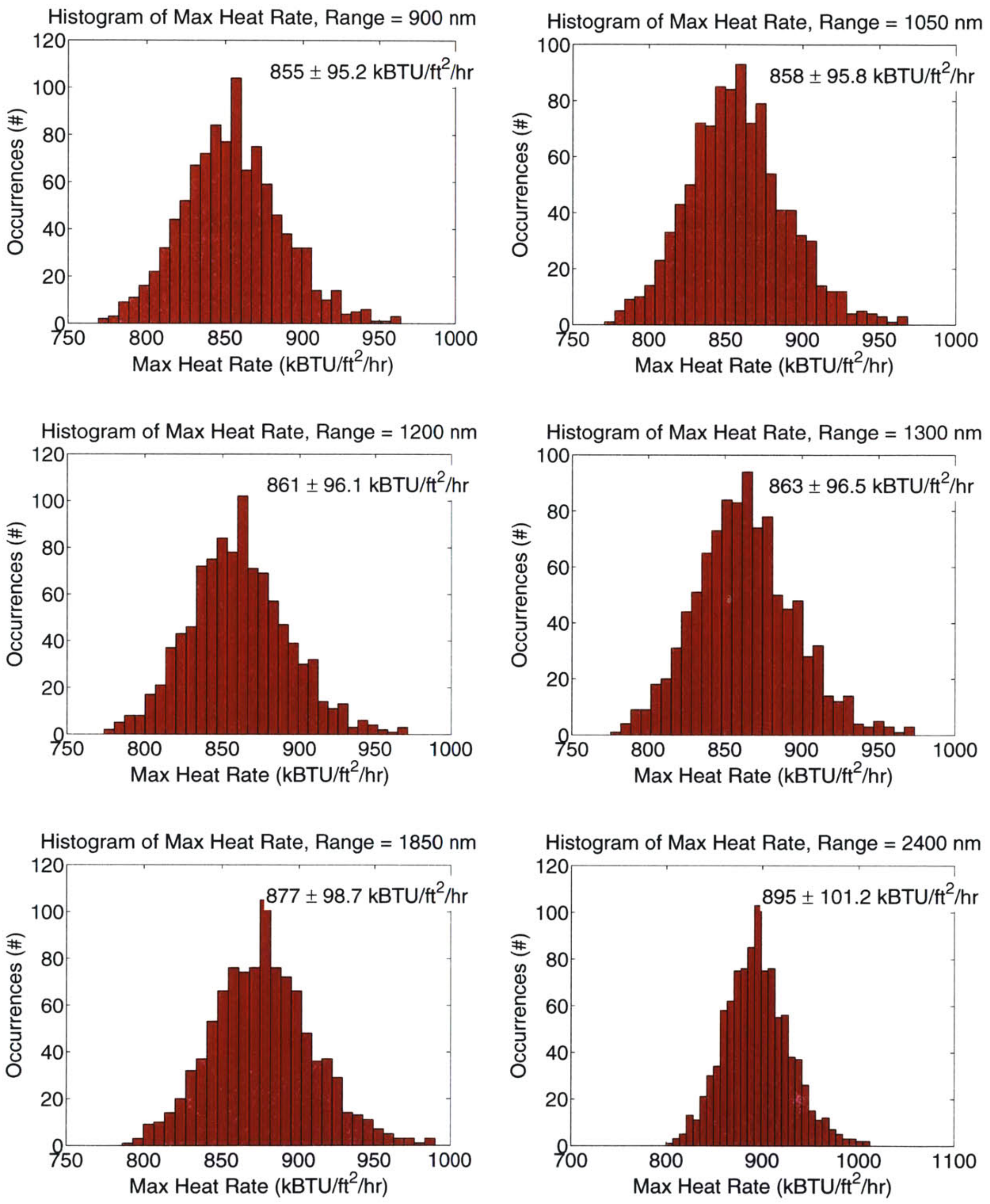


Figure A-22: Maximum heat rate histograms, Draper baseline algorithm, target ranges 900 – 2400 *nm*

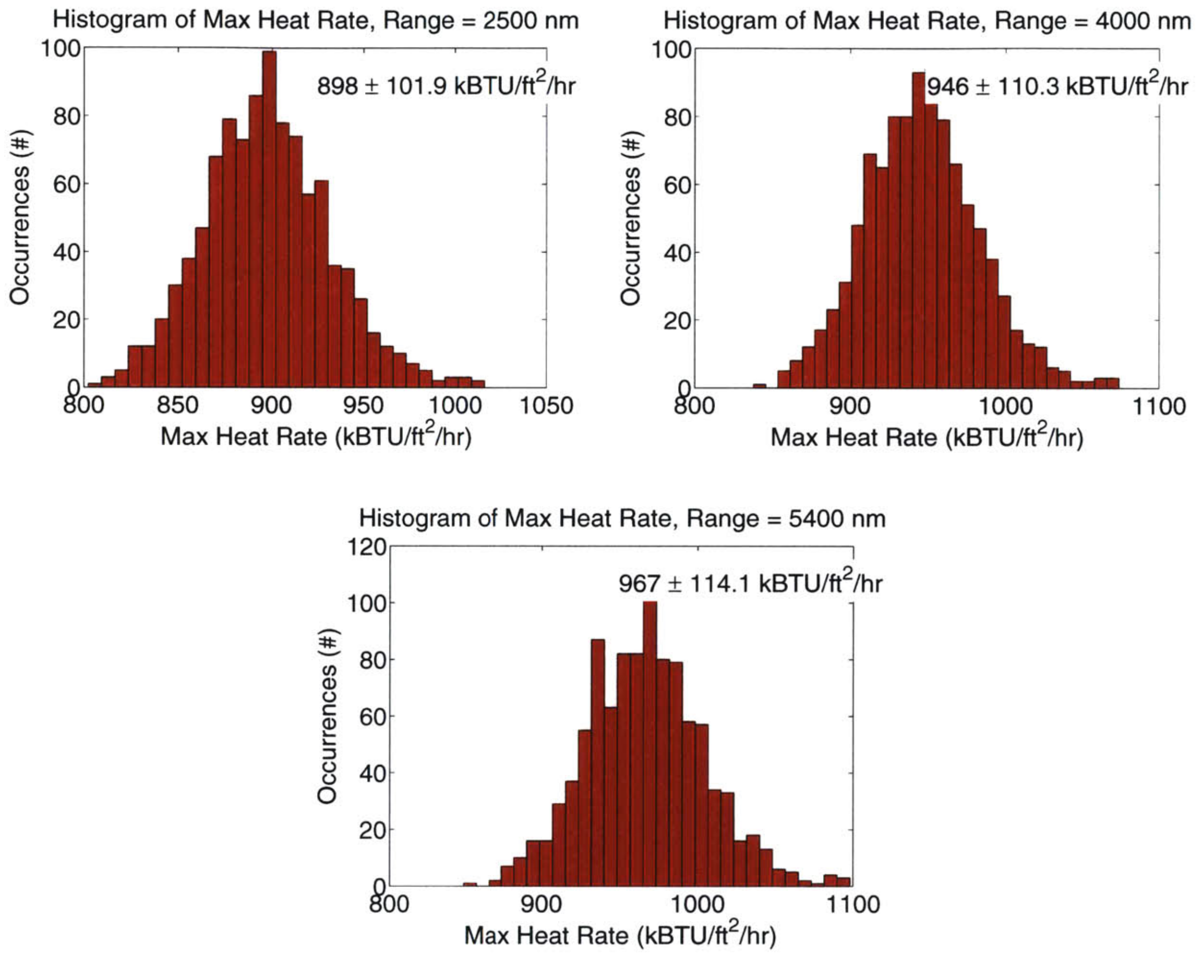


Figure A-23: Maximum heat rate histograms, Draper baseline algorithm, target ranges $2500 - 5400\ nm$

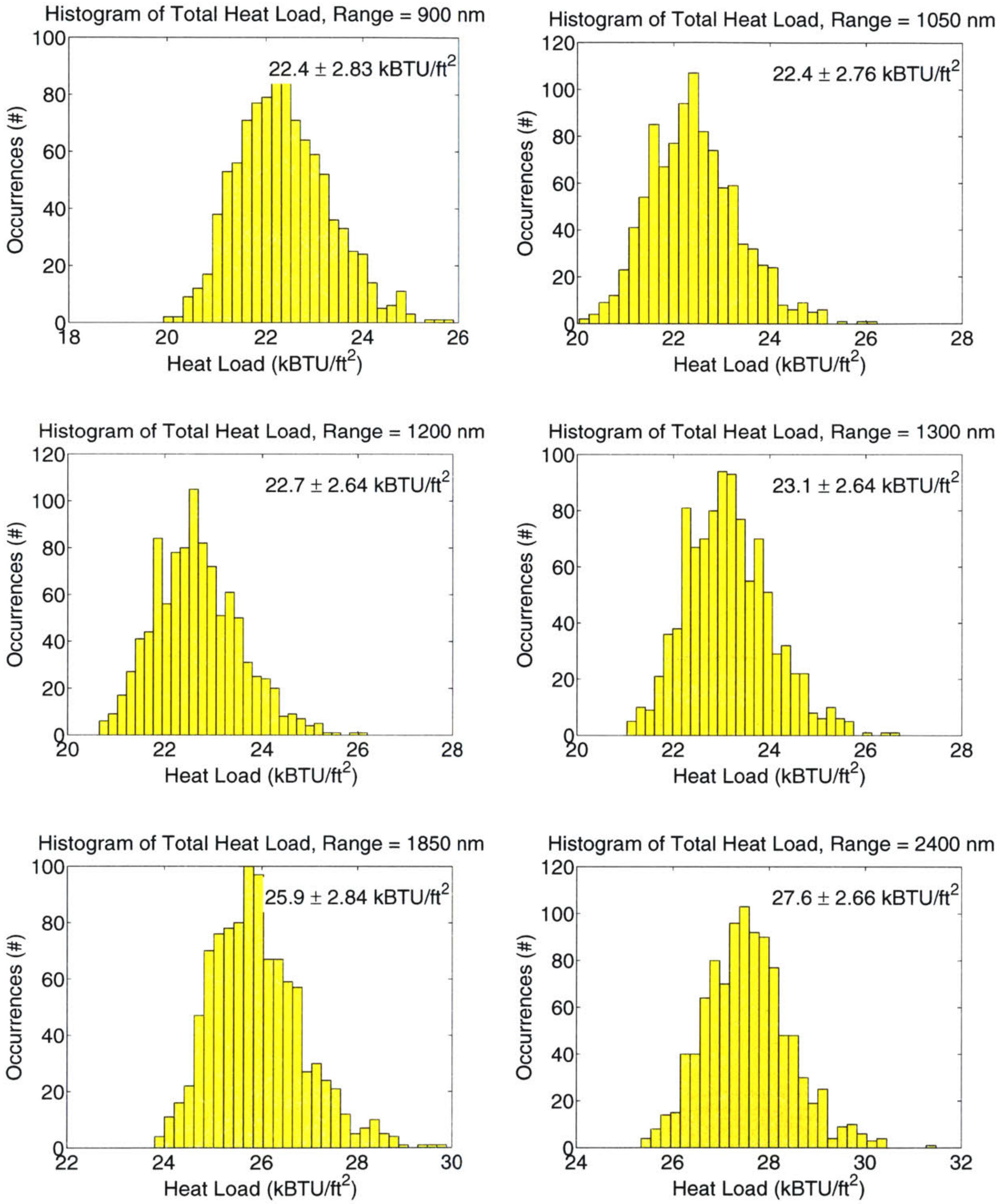


Figure A-24: Total heat load histograms, Draper baseline algorithm, target ranges 900 − 2400 nm

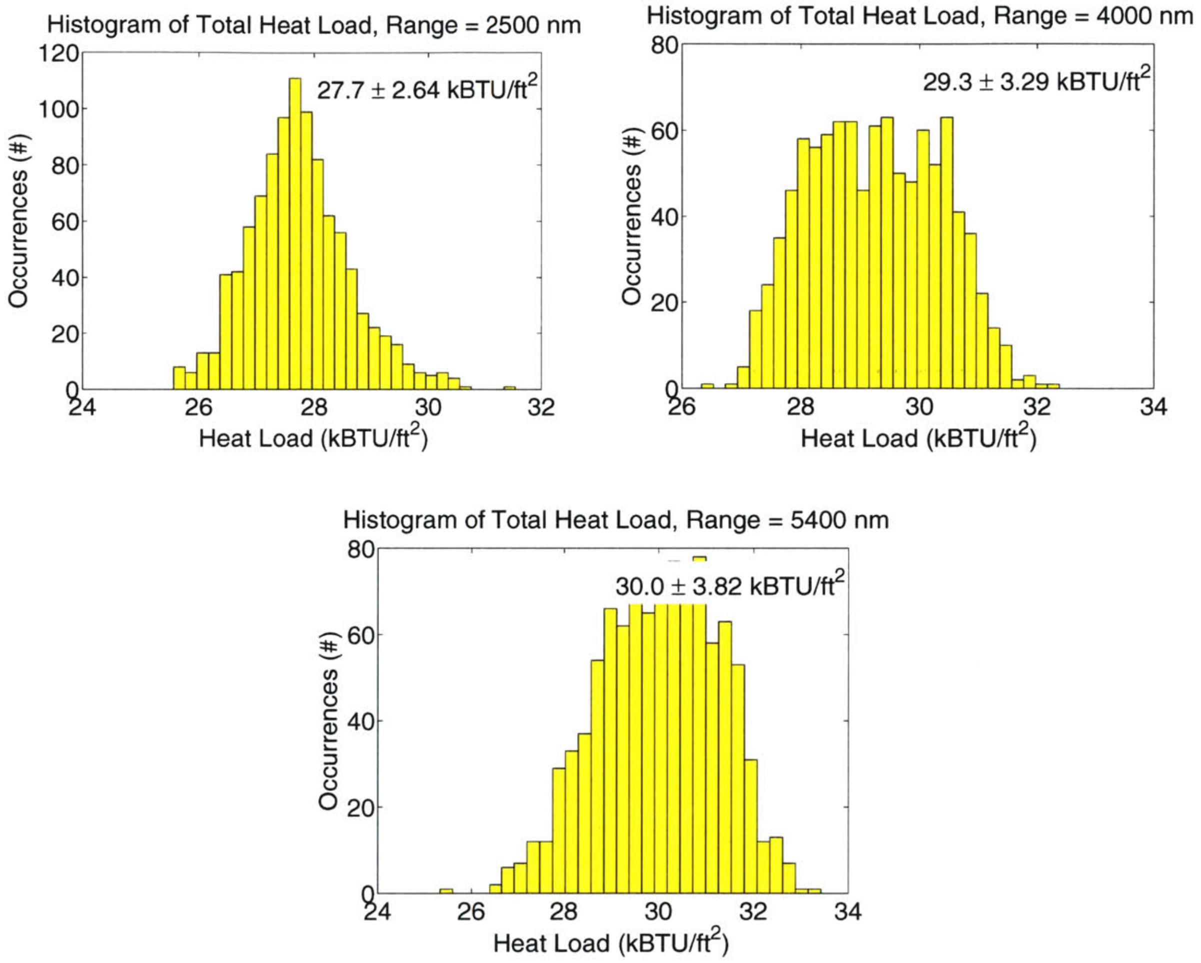


Figure A-25: Total heat load histograms, Draper baseline algorithm, target ranges 2500 – 5400 *nm*

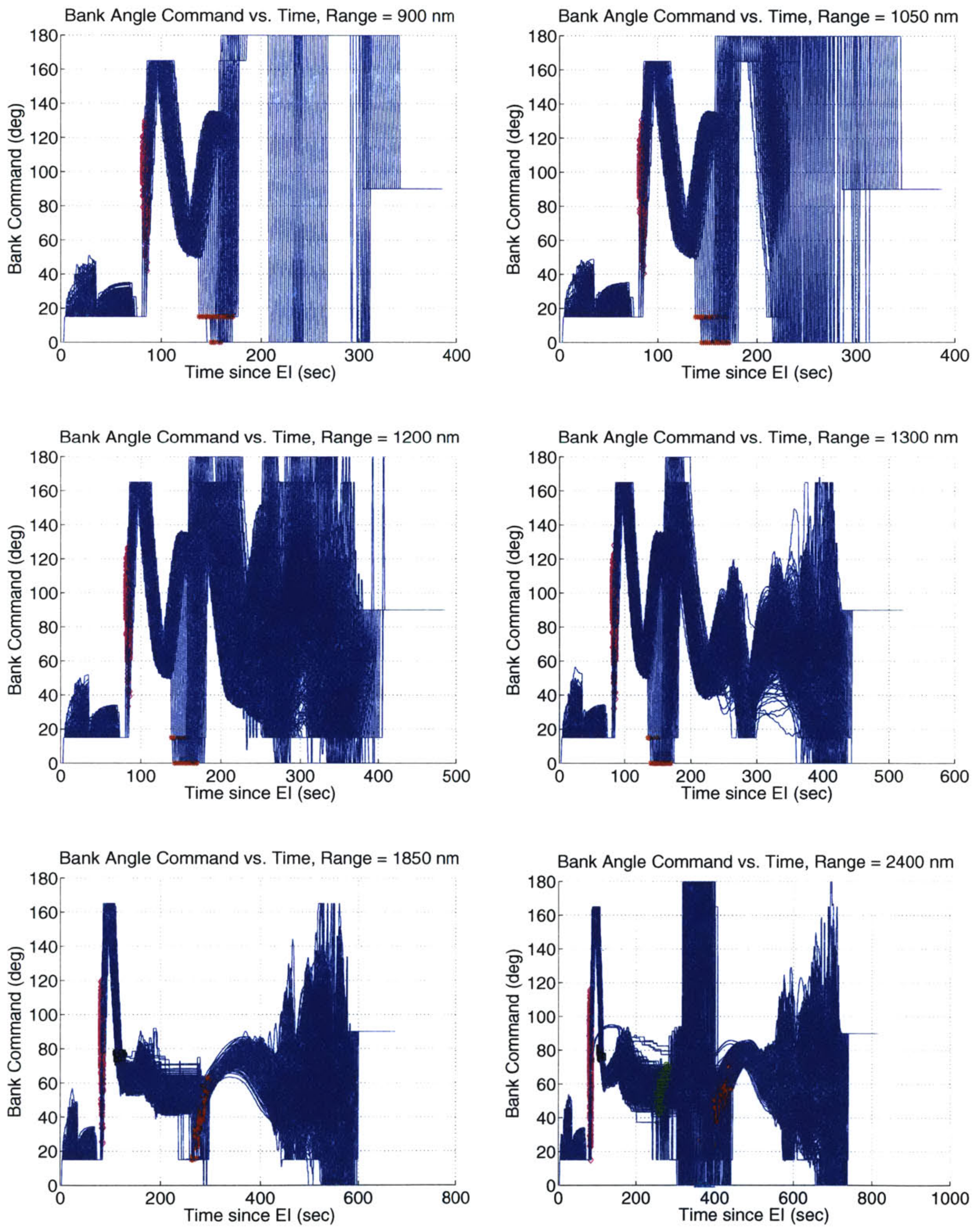


Figure A-26: Bank angle command history, Draper baseline, target ranges 900 − 2400 *nm*

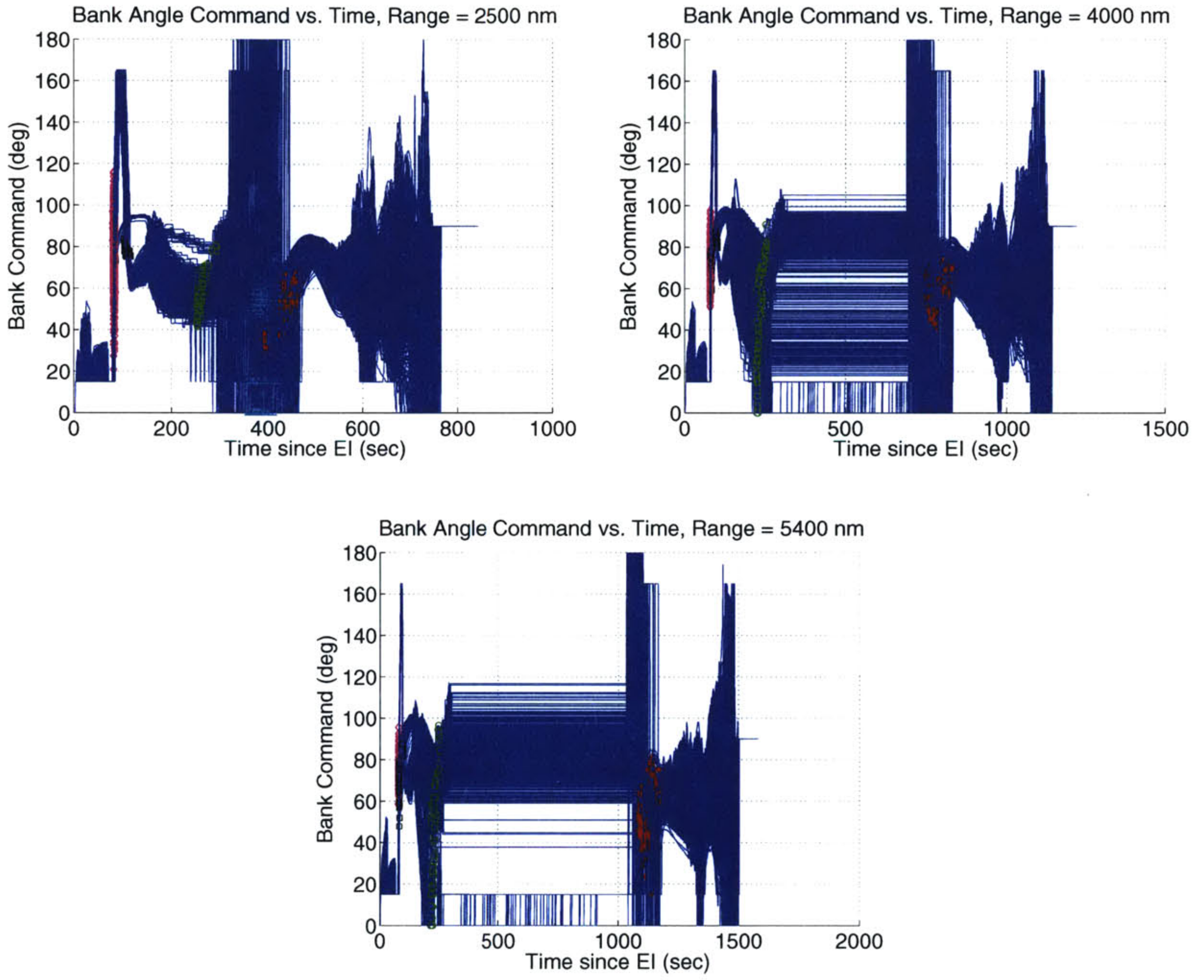


Figure A-27: Bank angle command history, Draper baseline, target ranges 2500 − 5400 nm

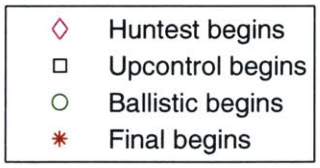


Figure A-28: Legend for bank angle command history figures

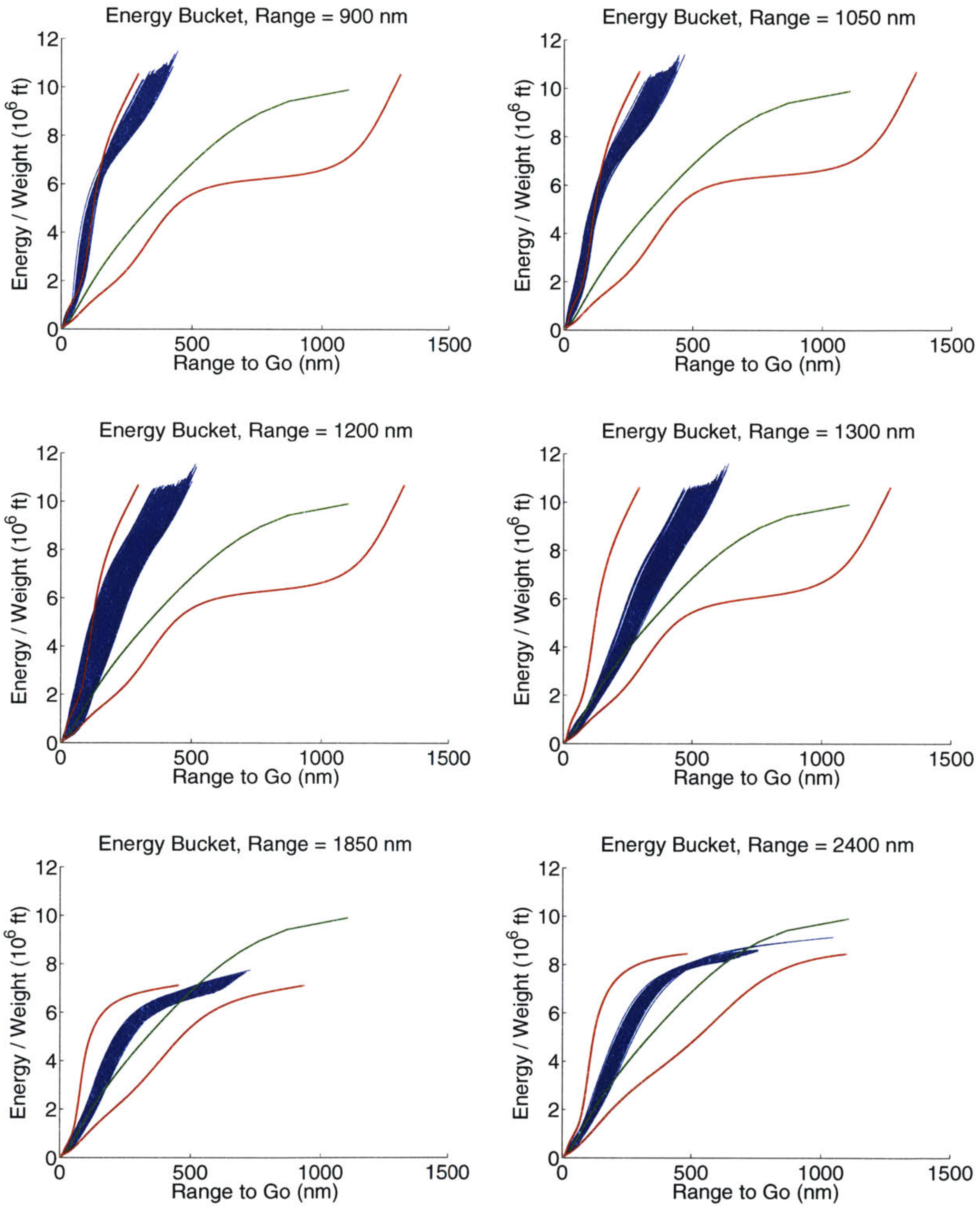


Figure A-29: Final phase energy buckets, Draper baseline algorithm, target ranges 900–2400 *nm*

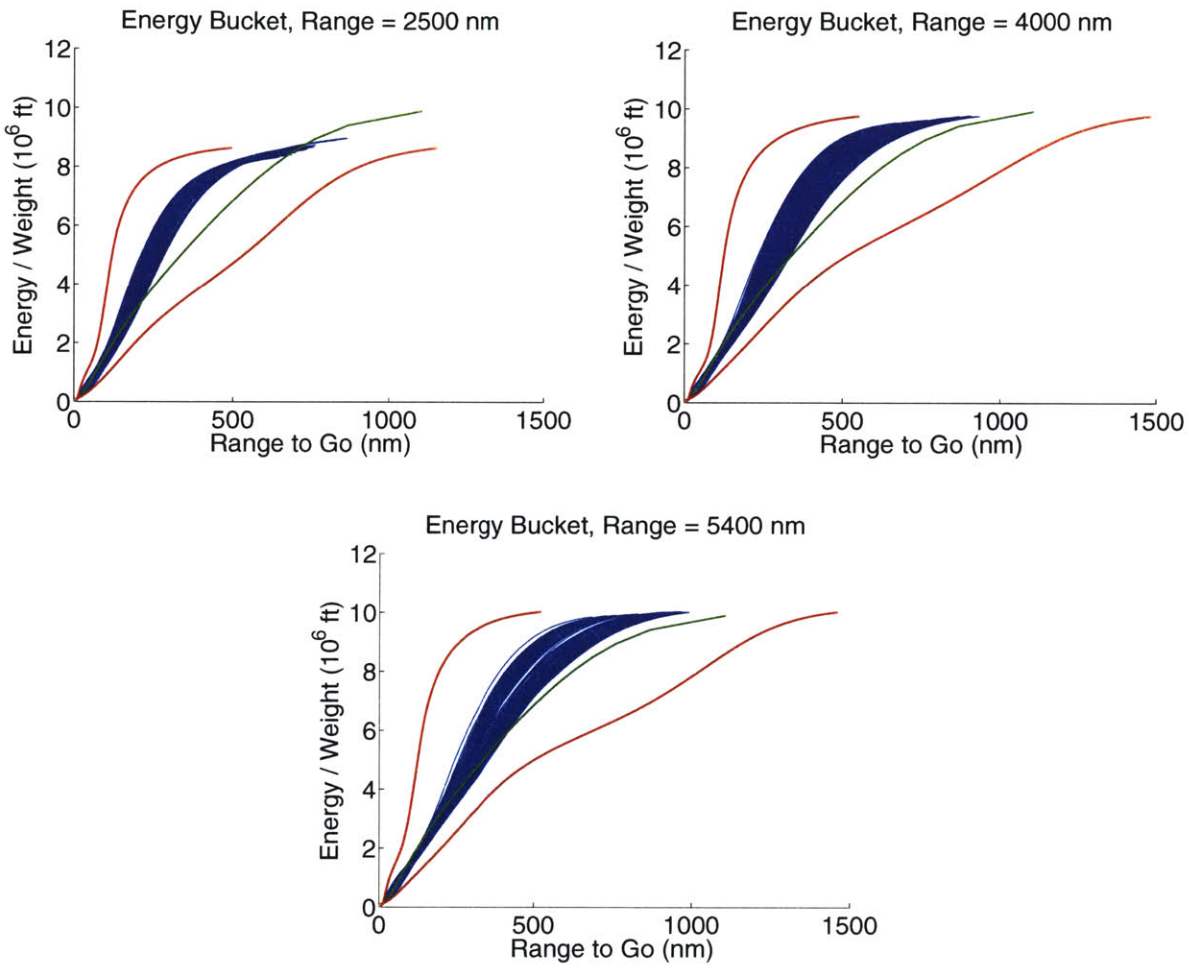


Figure A-30: Final phase energy buckets, Draper baseline algorithm, target ranges 2500 – 5400 nm

Appendix B

Detailed Description of EMT Predictor Enhancements

B.1 Overview

The Predictor algorithm is a derivative of the Draper baseline algorithm; therefore, much of the guidance executive and initialization of the algorithm is the same. The Draper baseline is fully described in Reference [2]. Phases or subroutines not mentioned in this appendix have not been modified from their Draper baseline versions.

This appendix describes the changes made to the Initial Roll, Huntest, Upcontrol, and Ballistic phases, as well as the Targeting, PredGuid, and Lateral Logic subroutines. Guidance code, written in Matlab, is also included where appropriate. Tables of the guidance variables and constants relevant to these sections are also included.

B.2 Targeting Subroutine

There have been a few changes made to the Targeting phase since its documentation for the Draper baseline algorithm [2]:

1. Addition of an L/D estimator
2. Atmospheric density estimator shut off during Ballistic phase

3. Final phase reference trajectory selector

The L/D estimator is called during Targeting to update guidance's constant L/D value with the vehicle's current L/D. This estimation, however, is only performed if the current drag is greater than the constant *ASENSIBLE* (1.6 ft/s^2) and guidance is not in the Ballistic or Final phases.

The atmospheric density estimator which calculates the current air density scale factor is only called if guidance is not in the Ballistic phase. The air density is extremely low during the Ballistic phase, and this causes large oscillations the density estimator's scale factor between between guidance cycles. This in turn causes large oscillations in PredGuid's bank angle commands during the Ballistic phase. To prevent these erratic bank commands, the density estimator is shut off during the Ballistic phase.

The Final phase reference trajectory selector (RT Selector) is used to choose the best reference trajectory for the Final phase. The selector is only called if the vehicle is performing a direct entry; the Draper baseline reference trajectory is used for all skip and loft entries. At each guidance cycle during the Initial Roll phase, the RT Selector chooses the best reference trajectory for the Final phase based on the predicted velocity, drag, and flight path angle at the start of the Final phase. During all phases after Initial Roll, the RT Selector's only task is to load the chosen reference trajectory.

B.2.1 L/D Estimation

The L/D estimator first calculates the current L/D based on the angle between the aerodynamic acceleration vector and the air-relative velocity vector. If the calculated L/D exceeds certain minimum or maximum values, it is limited to those min/max values. The estimated L/D is then incorporated into a weighted average of previous measurements.

```
%=====================================================
% LADEst: estimate L/D
%=====================================================
function [LAD_now LAD_avg] = LADEst(DATA)

% Inputs:  guidance DATA structure
% Outputs: LAD_now = current calculated L/D
%           LAD_avg = weighted average of L/D values

% Reassign constants
c = DATA.CONST.c;

% Calcuate current L/D based on angle between the aerodynamic acceleration
% vector and air-relative velocity vector
LAD_now = tan(acos(dot(unit(DATA.Dbar), -unit(DATA.V_rel))));

% Limit L/D estimate to certain min/max values
LAD_now = max(min([c.L_OVER_D_MAX, LAD_now]),c.L_OVER_D_MIN);

% Incorporate L/D estimation into weighted average of previous measurements
LAD_GAIN = c.L_OVER_D_FILTER_GAIN;
LAD_avg = LAD_GAIN*LAD_now + (1 - LAD_GAIN)*DATA.LAD_est;

%====================================================
% end LADEst
%====================================================
```

B.2.2 RT Selector

The Final phase reference trajectory selector (RT Selector) logic is located in the Targeting subroutine. Two conditions must be met to update the Final phase reference trajectory:

1. The predicted Final phase initial conditions have been updated from their initialization values
2. Guidance is in the Initial Roll phase

If these conditions are met, the RT Selector calls a separate routine, NewRefTraj, to pick the best reference trajectory. NewRefTraj can select between different references for direct entry, but only one reference trajectory is available for loft and skip entries. If the conditions are not met, the RT Selector simply loads the correct reference trajectory.

```
%======================================
% RT SELECTOR/LOADER
%======================================

% Only perform this if VL has been updated from initialized value
if DATA.VL ~= 0

    if DATA.SELECTOR == 1
        % If in Initial Roll, select Final phase ref traj
        DATA = NewRefTraj(DATA);
    elseif DATA.SELECTOR > 1
        % If not in Inital Roll, just LOAD the correct reference

        clear temp
        refs = [0.2, 3.0, 3.5, 4:10];
        lod  = [0.6, 0.2*ones(1,length(refs)-1)]*0.36;
        temp = load([DATA.CONST.RefTrajPath filesep num2str(DATA.FPGREF) 'Gref.mat']);
        DATA.CONST.FTABLE = temp.FTABLE;
        DATA.CONST.Q2  = temp.Q2;
        DATA.CONST.Q3  = temp.Q3;
        DATA.CONST.Q5  = temp.Q5;
        DATA.CONST.Q6  = temp.Q6;
        DATA.CONST.LOD = lod(find(refs == DATA.FPGREF,1));

    end

    % Update the trajectory type and count
    if DATA.TRAJTYPE ~= DATA.TRAJTYPE_PREV
        DATA.TRAJCOUNT = DATA.TRAJCOUNT + 1;
    end
    DATA.TRAJTYPE_PREV = DATA.TRAJTYPE;
end

%======================================
% end RT SELECTOR
%======================================




%==========================================================================
% NewRefTraj: function to determine the appropriate Final phase reference
% trajectory, based on predicted velocity, flight path angle, and drag at
% start of Final phase
%==========================================================================

function DATA = NewRefTraj(DATA)

switch DATA.TRAJTYPE
    case 0  % DIRECT ENTRY
        refs = [3.0, 3.5, 4:10];
        lod  = 0.2*ones(length(refs),1)*0.36;
    case 1  % LOFT
        refs = 0.2;
        lod  = 0.6*0.36;
    case 2  % SKIP
        refs = 0.2;
        lod  = 0.6*0.36;
end

% Loop through stored reference trajectories
for i = 1:length(refs)
```

```
    % Load reference trajectory data: FTABLE, Q2, Q3, Q5, Q6
    load([DATA.CONST.RefTrajPath filesep num2str(refs(i)) 'Gref.mat']);

    if DATA.VL > FTABLE.x(length(FTABLE.x) - 1)
        % If VL > velocity at start of reference trajectory, project upward
        % from beginning of reference trajectory using a Taylor series
        % expansion (Apollo method)
        ASP1 = Q2 + Q3*DATA.VL;          % velocity correction
        ASP3 = Q5*(Q6 + DATA.GAMMAL);    % flight path angle correction
        final_range = ASP1 + ASP3;       % total range
        RTOGO = FTABLE.Y(end-1,5);       % nominal range-to-go at start of ref traj

    else
        % If VL < velocity at start of reference trajectory,
        % interpolate in the table, just as in Final phase
        yi = interp1(FTABLE.x,FTABLE.Y,DATA.VL,'linear');

        % Assign values:
        RDOTREF = yi(1);  % altitude rate
        DREFR   = yi(2);  % drag
        F2      = yi(3);  % dRange/d(Altitude rate)
        F1      = yi(4);  % dRange/dDrag
        RTOGO   = yi(5);  % range to go
        F3      = yi(6);  % dRange/d(L/D)

        % Calculate estimated Final phase range
        final_range = RTOGO + F2*(DATA.GAMMAL*DATA.VL - RDOTREF) + F1*(DATA.Q7 - DREFR);
    end

    % If range error < current smallest range error or first iteration,
    % store current range error as smallest
    if abs(final_range - RTOGO) < range_small || i == 1
        range_small = abs(final_range - RTOGO);
        Gref = refs(i);     % keep track of reference #
    end
end

% Reference that has smallest range error
DATA.FPGREF = Gref;
idx = find(refs == DATA.FPGREF,1);

% Load reference trajectory & assign values
temp = load([DATA.CONST.RefTrajPath filesep num2str(DATA.FPGREF) 'Gref.mat']);
DATA.CONST.FTABLE = temp.FTABLE;
DATA.CONST.Q2  = temp.Q2;
DATA.CONST.Q3  = temp.Q3;
DATA.CONST.Q5  = temp.Q5;
DATA.CONST.Q6  = temp.Q6;
DATA.CONST.LOD = lod(idx);

%==========================================================================
% end NewRefTraj
%==========================================================================
```

B.3 Initial Roll Phase

The Initial Roll phase has been significantly modified from its Apollo and Draper predecessors. The basic logic is shown in Figure B-1. For clarity, the guidance code for Initial Roll will be presented here in sections.

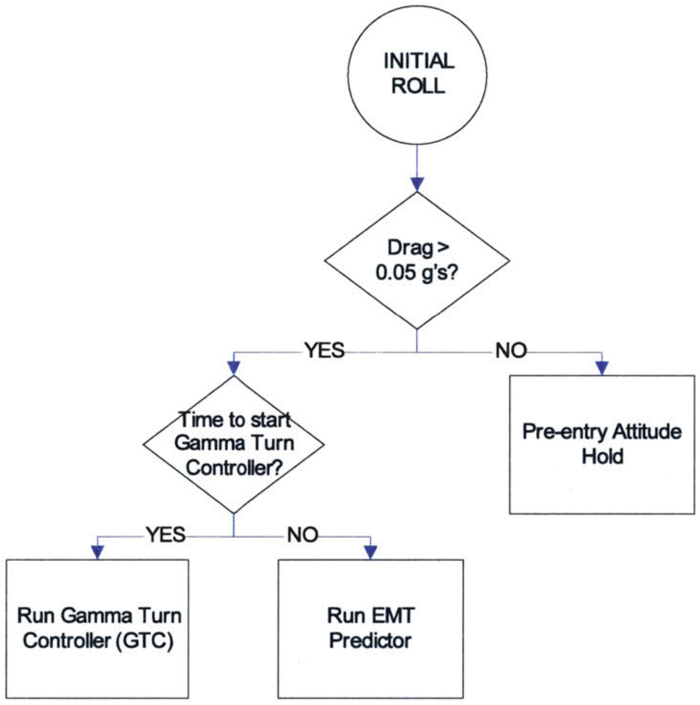


Figure B-1: EMT Predictor algorithm: Initial Roll

B.3.1 Preentry Attitude Hold

"Preentry Attitude Hold" is the first segment of Initial Roll, and it is maintained during the first part of the entry in which the sensed acceleration is below 0.05 g's. During this segment, guidance transitions directly to Lateral Logic without changing the bank command. Once the sensed acceleration exceeds 0.05 g's, full lift up is commanded. If the current velocity is below V_{final} (indicating a very low-energy entry), Initial Roll transitions to Ballistic. For a standard lunar return in which the entry velocity is above V_{final}, guidance continues with the rest of the Initial Roll logic.

```
%==========
% INITROLL
%==========

function DATA = INITROLL(DATA)

DATA.NOSWIT = 1; % Disable roll reversals

% TEST: Has sensible atmosphere been reached in previous iterations?
if DATA.INRLSW == 0
    % NO, sensible atmosphere has NOT previously been reached

    % TEST: Has sensible atmosphere been reached in THIS iteration?
    if (DATA.D - DATA.CONST.ASENSIBLE) > 0

        % YES, sensible atmosphere has been reached in this iteration
        % Indicate that sensible atmosphere has been reached (INRLSW flag)
        DATA.INRLSW = 1;

        % TEST: Is entry velocity too low for standard entry?
        if DATA.V < DATA.CONST.VFINAL
            % YES, entry velocity is low. Command full lift up
            DATA.LD = DATA.CONST.LAD;

            % Transition to Ballistic phase, but go to LATLOGIC first
            DATA.SELECTOR = 4;  % BALLISTIC
            DATA.NEXTSTEP = 11; % LATLOGIC
            return;
        else
            % NO, entry velocity not too low. Command full lift up
            DATA.LD = DATA.CONST.LAD;

            % Continue to LATLOGIC
            DATA.NEXTSTEP = 11; % LATLOGIC
            return;
        end

    else
        % NO, sensible atmosphere has not yet been reached
        % Continue directly to LATLOGIC without changing roll command
        DATA.NEXTSTEP = 11; %LATLOGIC
        return;
    end
else

    ....
```

Once the drag is above 0.05 g's, Initial Roll continues to one of two segments. If it is time to begin or continue the Gamma Turn Controller (indicated by the GTC flag), Initial Roll skips the EMT Predictor segment and moves directly to the Gamma Turn Controller. If it is not time to begin or continue the Gamma Turn Controller, Initial Roll continues with the EMT Predictor segment.

B.3.2 EMT Predictor

The EMT Predictor is divided into three smaller predictors (all Simulink models):

1. `predictor_initroll`, for predicting the range traveled during Initial Roll
2. `predictor_turn`, for predicting the range traveled during Initial Roll and while the Gamma Turn Controller is operating
3. `predictor_ballistic`, for predicting the range traveled during Initial Roll and the skip

The "correction" portion that would correspond to the predictors listed above to make them "predictor-correctors" is written in Matlab code rather than embedded in Simulink.

The predictors use the planar equations of motion for a point mass (see Section 2.5.4); therefore, they perform trajectory prediction in only two dimensions: altitude and downrange.

Direct Entry Range Prediction

The EMT Predictor's first task is to determine the trajectory type—whether a skip is necessary. This is accomplished by first predicting forward in time until the maximum drag point is reached, using the current guess for the appropriate Initial Roll L/D fraction. The drag, velocity, range, and position are recorded at the maximum drag point, and these values are used to predicted the range traveled during Constant Drag and the Final phase.

```
....

% YES, sensible atmosphere has previously been reached.

% Is it time to begin the gamma turn controller?
if DATA.GTC == 0
    % NO, it's NOT time to begin gamma turn controller

    % Run predictor to determine time to transition to
    % either Huntest (direct entry) or Upcontrol (loft/skip)
    LDfrac = DATA.LDfrac;
```

```
% Initialize variables for predictor_initroll
clear PredIn
PredIn.LDratio = LDfrac*DATA.CONST.LAD;
if abs(PredIn.LDratio) > DATA.CONST.LDCMINR
    PredIn.LDratio = sign(PredIn.LDratio)*DATA.CONST.LDCMINR;
end
PredIn.tstep = 1;                           % integration time step [sec]
PredIn.V = DATA.V;                          % current velocity [ft/s]
PredIn.R = DATA.CONST.RE + DATA.alt;        % current position [ft]
PredIn.gamma = asin(DATA.RDOT/DATA.V);      % current flight path angle [rad]
PredIn.GS = DATA.CONST.GS;                  % gravity [ft/s^2]
PredIn.Dco = DATA.CONST.GMAX;               % max g limit
PredIn.Vco = DATA.CONST.FTABLE.x(end-1);    % velocity at start of ref traj
PredIn.RE = DATA.CONST.RE;                  % radius of Earth [ft]
PredIn.LAD = DATA.CONST.LAD;                % max L/D
PredIn.LDCMINR = DATA.CONST.LDCMINR;        % cos(15)*LAD
PredIn.atm_table = DATA.CONST.atm_table;    % atmosphere lookup table)
PredIn.Krho = DATA.Krho_est;                % atmospheric density scale factor
PredIn.SCd = DATA.CONST.c.S_REF*...
    DATA.CONST.c.CD_EST_INITIAL/DATA.CONST.c.MASS_EI;   % S*Cd/mass [ft^2/lbm]

% Run Initial Roll predictor
clear IR* tout
assignin('base','PredIn',PredIn);
sim('predictor_initroll');

% Record parameters at max drag point (idx_maxd)
idx_maxd = find(IRdrag(:,2) == max(IRdrag(:,2)),1);
PredIn.D0 = IRdrag(idx_maxd,2);                     % Constant Drag level
InitrollRange = IRrange(idx_maxd,2);
InitrollVelocity = IRvelocity(idx_maxd,2);
InitrollR = DATA.CONST.RE + IRalt(idx_maxd,2);

% Calculate range traveled during Constant Drag
CDtheta = (InitrollVelocity^2 - PredIn.Vco^2)/(2*DATA.CONST.RE*PredIn.D0);%[rad]
CDrng   = CDtheta*InitrollR*(1/6076.11549);      % [rad] -> [ft] -> [nm]

% Estimate Final phase range
DATA.VL     = PredIn.Vco;
DATA.GAMMAL = -2*DATA.CONST.HS*PredIn.D0/PredIn.Vco^2;
DATA.Q7     = PredIn.D0;
DATA = FinalRangeEst(DATA);

% Add ranges: Initial Roll + Constant Drag + Final
PredictedRange = InitrollRange + CDrng + DATA.final_range;    % [nm]

....
```

Direct Entry Range Correction

If the difference between the predicted range and the desired range is more than the allowed tolerance, the Initial Roll L/D fraction guess (`LDfrac`) must be adjusted. If the predicted range is too long, the L/D fraction is decreased so that the Constant Drag level will increase, thereby shortening the predicted range. If the predicted range is too short, the L/D fraction is increased so that the Constant Drag level is decreased. If the Initial Roll L/D fraction guess is already at 1 and the predicted range is too short, the desired range is not achievable via a direct entry and a loft or skip is necessary. If this is the case, Initial Roll exits the direct entry adjustment logic and continues to the loft/skip logic. Guidance exits the direct entry adjustment *while* loop in three other scenarios:

1. Predicted range is within tolerance of desired range
2. The current trajectory type is a skip
3. Amount to increment L/D fraction guess is below tolerance

Once guidance exits the loop, it assigns the maximum drag point to $T_{max\ drag}$ and continues to the next segment of the logic, depending on the trajectory type.

```
....

% DETERMINE TRAJECTORY TYPE: DIRECT ENTRY OR LOFT
% Do this while predicted range not within tolerance of desired
% range, predicted trajectory not a skip, and amount to increment
% L/D fraction not too small
kk = 0;
LDinc = 0.1;
while abs(PredictedRange - DATA.THETNM) > DATA.CONST.TOL ...
    && DATA.TRAJTYPE ~= 2 && LDinc > 0.02

    % If currently flying full lift up and predicted range is more than
    % 25 nm short, direct entry is infeasible -> loft
    % Or if flip-flopped btw DE and loft too many times -> loft
    if PredictedRange < DATA.THETNM && LDfrac > 0.99 || DATA.TRAJCOUNT >= 4
        % If current direct entry, change to loft.
        DATA.TRAJTYPE = 1;  % LOFT
        break;
    else
        DATA.TRAJTYPE = 0;  % DIRECT ENTRY
    end

    % Reduce L/D increment, if necessary
```

```
if kk >= 21 || (LDinc < 0.05 && kk > 3)
    LDinc = LDinc/2;
    kk = 0;
end

% If range is short, increase L/D fraction to decrease constant drag level
if PredictedRange < DATA.THETNM && LDfrac < 1
    LDfrac = LDfrac + LDinc;
    kk = kk + 1;

% If range is long, decrease L/D fraction to increase constant drag level
elseif PredictedRange > DATA.THETNM && LDfrac > -0.99
    LDfrac = LDfrac - LDinc;
    kk = kk + 1;
end

% Stay outside of +/- 15 deg of max lift up/down
PredIn.LDratio = LDfrac*DATA.CONST.LAD;
if abs(PredIn.LDratio) > DATA.CONST.LDCMINR
    PredIn.LDratio = sign(PredIn.LDratio)*DATA.CONST.LDCMINR;
end

% Re-run Initial Roll predictor
clear IR* tout
assignin('base','PredIn',PredIn);
sim('predictor_initroll');

% Record find parameters at max drag point
idx_maxd = find(IRdrag(:,2) == max(IRdrag(:,2)),1);
PredIn.D0 = IRdrag(idx_maxd,2);                    % Constant Drag level
InitrollRange = IRrange(idx_maxd,2);
InitrollVelocity = IRvelocity(idx_maxd,2);
InitrollR = DATA.CONST.RE + IRalt(idx_maxd,2);

% Calculate range traveled during Constant Drag
CDtheta = (InitrollVelocity^2 - PredIn.Vco^2)/...
    (2*DATA.CONST.RE*PredIn.D0); % [rad]
CDrng   = CDtheta*InitrollR*(1/6076.11549);      % [rad] -> [ft] -> [nm]

% Estimate Final phase range
DATA.VL     = PredIn.Vco;
DATA.GAMMAL = -2*DATA.CONST.HS*PredIn.D0/PredIn.Vco^2;
DATA.Q7     = PredIn.D0;
DATA = FinalRangeEst(DATA);

% Add ranges: Initial Roll + Constant Drag + Final
PredictedRange = InitrollRange + CDrng + DATA.final_range;    % [nm]

% If already flying full lift down and predicted range still too long,
% get out of the 'while' loop
if LDfrac < -0.99 && PredictedRange > (DATA.THETNM + DATA.CONST.TOL)
    break;
end
```

```
end
% end TRAJECTORY TYPE 'while' loop

% Time at which max drag is reached during Initroll [sec]
TmaxDrag = IRdrag(find(IRdrag(:,2) == PredIn.D0,1),1);

....
```

GTC Start Time Determination

If the trajectory is a direct entry, guidance begins predicting when to start the Gamma Turn Controller (GTC). This is done by starting at the max drag time and commanding the L/D required to "turn" the flight path angle rate to its desired Constant Drag value. The minimum error in the predicted trajectory from the Constant Drag reference flight path angle is recorded and the loop starts again, this time starting the GTC L/D command 1 second earlier in the predicted trajectory. The GTC start time is stepped backward 10 seconds, and the start time that gives the smallest flight path angle error is saved as T_{turn}, the predicted time to start the Gamma Turn Controller.

The remainder of this section updates the predicted Final phase initial conditions and checks for the conditions that must be met in order to begin the Gamma Turn Controller. These conditions are:

- The predicted time to begin the GTC (T_{turn}) is less than 5 seconds from the current time
- The current drag and current flight path angle are within 1.5 g's and 0.1° of their respective Constant Drag values
- The desired range is less than the predicted Final phase range plus 25 nm

If any of these three conditions are true, the flag to start the Gamma Turn Controller is set. This causes Initial Roll to bypass the trajectory selection and transition time prediction logic (the EMT Predictor) in the next guidance cycle and start the Gamma Turn Controller instead. If none of these conditions are met, the appropriate L/D is commanded and Initial Roll transitions to Lateral Logic.

```
....
```

```
% TEST: Is trajectory a direct entry?
if DATA.TRAJTYPE == 0
    % YES, it is a direct entry -> predict time to begin Gamma Turn Controller

    % Initialize variables
    DATA.LDfrac = LDfrac;
    PredIn.LDratio = DATA.LDfrac*DATA.CONST.LAD;
    if abs(PredIn.LDratio) > DATA.CONST.LDCMINR
        PredIn.LDratio = sign(PredIn.LDratio)*DATA.CONST.LDCMINR;
    end
    backstep = 0;
    gammaSmall = [100 0 0];

    % Step backward in time from TmaxDrag to determine time to
    % begin Gamma Turn Controller
    while backstep <= 10 && backstep*PredIn.tstep < TmaxDrag
        clear tout idx IRT*
        PredIn.Tturn = TmaxDrag - backstep*PredIn.tstep;

        % Run turn predictor
        assignin('base','PredIn',PredIn);
        sim('predictor_turn');

        % Find smallest difference between actual gamma and gamma_ref
        idx = find(abs(IRTgamma_diff) == min(abs(IRTgamma_diff)),1);

        % If this gamma difference is smaller than previous, replace
        % values in gammaSmall
        if abs(IRTgamma_diff(idx)) < gammaSmall(1)
            gammaSmall = [abs(IRTgamma_diff(idx)) abs(IRTGDdiff(idx)) backstep];
        end

        backstep = backstep + 1;
    end

    % Update time to begin GTC
    backstep = gammaSmall(3);
    PredIn.Tturn = TmaxDrag - backstep*PredIn.tstep;

    % Update running average of predicted Constant Drag value
    DATA.D0 = (DATA.D0ct*DATA.D0 + PredIn.D0)/(DATA.D0ct + 1);
    DATA.D0ct = DATA.D0ct + 1;

    % Update predicted velocity, drag, and altitude rate at
    % start of Final for reference trajectory selection
    DATA.VL     = PredIn.Vco;
    DATA.GAMMAL = -2*DATA.CONST.HS*DATA.D0/(PredIn.Vco^2);
    DATA.Q7     = DATA.D0;
    DATA = FinalRangeEst(DATA);

    % Flight path angle required at beginning of Constant Drag [deg]
    if length(IRdrag(1:end-1,2)) >= 2
    vel = interp1(IRdrag(1:end-1,2),IRvelocity(1:end-1,2),DATA.D0,'linear');
else
    vel = IRvelocity(end-1,2);
end
if isnan(vel)
    vel = IRvelocity(find(IRdrag(:,2) == PredIn.D0,1),2);
end
gammaCD = -2*DATA.CONST.HS*DATA.D0/(vel^2);

% Are conditions met to begin Gamma Turn Controller?
if PredIn.Tturn <= 5 || ...
        (DATA.D + 1.5*DATA.CONST.GS >= DATA.D0 && ...
        asin(DATA.RDOT/DATA.V) > gammaCD - 0.1*pi/180) || ...
        (DATA.THETNM < DATA.final_range + DATA.CONST.TOL)

    % YES, set flag to begin Gamma Turn Controller
    DATA.GTC = 1;

else
    % NO, continue with INITIAL ROLL
    % Command L/D
    DATA.LD = DATA.LDfrac*DATA.CONST.LAD;

    % Continue directly to LATLOGIC
    DATA.NEXTSTEP = 11; % LATLOGIC
    return;
end

....
```

Skip/Loft Range Prediction

If the entry is a skip or loft, the logic to predict the time to start the Gamma Turn Controller is bypassed and guidance starts predicting when to transition to Upcontrol. The first thing to do in this segment is increase the Initial Roll bank angle if, in a previous guidance cycle, the predicted drag at the transition to Upcontrol was too low. This indicates a shallow entry and is recorded using the *RLOVER* flag. If the flag is set to 1, the Initial Roll bank angle is increased by 30°. If the previous trajectory type was a loft or skip and the entry is not shallow, no change is made to the Initial Roll bank angle. If this is the first iteration or the predicted transition to Upcontrol was not too early in the previous guidance cycle, the Initial Roll L/D is set to full lift up (limited, of course, by $\pm 15°$). Once the correct Initial Roll L/D is set, the first prediction of the loft/skip trajectory range is made.

```
....
```

```
else
    % NO, trajectory is a loft or skip -> predict time to
    % transition to Upcontrol

    % If entry is shallow and current L/D command is not full lift
    % down, reduce the L/D fraction
    if DATA.RLOVER ~= 0 && DATA.LDfrac > -0.99
        DATA.LDfrac = cosd(acosd(DATA.LDfrac) + 30);

    % If entry is not shallow and previous trajectory type was
    % direct entry, set Initroll L/D to full lift up
    elseif DATA.RLOVER == 0 && DATA.TRAJTYPE_PREV == 0
        DATA.LDfrac = 1;
    end

    bank_ref = DATA.BANKREF;    % desired skip bank angle

    % Initialize variables
    PredIn.Q2      = DATA.CONST.Q2;
    PredIn.Q3      = DATA.CONST.Q3;
    PredIn.Q5      = DATA.CONST.Q5;
    PredIn.Q6      = DATA.CONST.Q6;
    PredIn.V       = DATA.V;
    PredIn.R       = DATA.alt + DATA.CONST.RE;
    PredIn.gamma   = asin(DATA.RDOT/DATA.V);
    PredIn.FTABLE  = DATA.CONST.FTABLE;  % Final phase ref traj lookup table
    PredIn.LDratio = DATA.LDfrac*DATA.CONST.LAD;
    if abs(PredIn.LDratio) > DATA.CONST.LDCMINR
        PredIn.LDratio = sign(PredIn.LDratio)*DATA.CONST.LDCMINR;
    end
    PredIn.LDratio2 = cosd(bank_ref)*DATA.CONST.LAD;    % skip L/D

    % If time to switch to reference skip bank angle not
    % previously set, use max drag time
    if DATA.Tswitch == 1000
        PredIn.Tswitch = TmaxDrag;
    else
        PredIn.Tswitch = max(0,DATA.Tswitch);
    end

    % Run ballistic predictor
    % Variables out to workspace: PredictedRange, FinalDrag,
    %   FinalGamma, SkipawayFlag, Brange, Bdrag, Bvelocity, Bgamma,
    %   Balt
    clear B* tout;
    assignin('base','PredIn',PredIn);
    sim('predictor_ballistic_gamma');

    ....
```

If the difference between the predicted trajectory range and the desired range is within the tolerance and the predicted trajectory has not skipped away (indicated by a positive final flight path angle), the "correction" portion is skipped and guidance continues to the logic that determines if the entry is shallow. If the range error is larger than the tolerance, however, or the predicted trajectory has skipped away, the correction portion begins.

Upcontrol Transition Time Correction

In order to meet the range requirement, the transition time to Upcontrol is adjusted first. The transition time guess is decreased if any of the following are true:

1. The predicted range is too long and the Initial Roll bank angle is smaller than the skip bank angle
2. The predicted range is too short and the Initial Roll bank angle is larger than the skip bank angle
3. The predicted trajectory skipped away and the Initial Roll bank angle is smaller than the skip bank angle

Conversely, the transition time guess is increased if any of the following are true:

1. The predicted range is too short and the Initial Roll bank angle is smaller than the skip bank angle
2. The predicted range is too long and the Initial Roll bank angle is larger than the skip bank angle
3. The predicted trajectory skipped away and the Initial Roll bank angle is larger than the skip bank angle

For a typical skip entry using this algorithm, the Initial Roll bank angle is smaller than the skip bank angle. For this situation, delaying the transition to Upcontrol allows the vehicle to spend a longer amount of time flying lift up, thereby increasing the downrange. An earlier transition to Upcontrol shortens the range because more of the trajectory is flown at a larger bank angle. A case in which the final predicted γ value is positive (skipaway) indicates that the vehicle has spent too much time flying lift up.

If the range error is larger than the tolerance or the predicted trajectory has skipped away and the transition is predicted to occur at the current guidance cycle, the desired skip bank angle is adjusted. For a skipaway condition or if the predicted range is too long, the desired skip bank angle is increased. This case is encountered most often for very short range loft entries. If the predicted range is too short, the desired skip bank is decreased. This is more typical of very long skips.

```
....

% Initialize variables for corrector
iter = 0;
count = 0;
max_iter = 100;
max_count = 100;

% CORRECTOR: UPCONTROL TRANSITION TIME
% If range error > 25 nm or flight path angle at end of
% predicted trajectory is positive (skipaway), adjust switch time
while (abs(PredictedRange - DATA.THETNM) > DATA.CONST.TOL || ...
        FinalGamma > 0) && iter < max_iter && count < max_count

    % Record previous trajectory parameters
    PredRange_prev  = PredictedRange;
    FinalGamma_prev = FinalGamma;
    Tswitch_prev    = PredIn.Tswitch;

    % Adjust transition time (Tswitch)
    %--------------------------------
    % If skipaway and current L/D > skip L/D, transition to
    % loft/skip earlier
    if FinalGamma > 0 && DATA.LDfrac > cosd(bank_ref)
        PredIn.Tswitch = PredIn.Tswitch - 1;
        iter = iter + 1;

    % Else, if skipaway and current L/D < skip L/D, transition
    % to loft/skip later
    elseif FinalGamma > 0 && DATA.LDfrac < cosd(bank_ref)
        PredIn.Tswitch = PredIn.Tswitch + 1;
        iter = iter + 1;

    % Else, if range is short and current L/D > skip L/D OR
    % range is long and current L/D < skip L/D, transition to
    % loft/skip later
    elseif (PredictedRange < DATA.THETNM && DATA.LDfrac > cosd(bank_ref)) ...
            || (PredictedRange > DATA.THETNM && DATA.LDfrac < cosd(bank_ref))
        PredIn.Tswitch = PredIn.Tswitch + 1;
        iter = iter + 1;

    % Else, if range is long and current L/D > skip L/D OR
% range is short and current L/D < skip L/D, transition to
% loft/skip earlier
elseif (PredictedRange > DATA.THETNM && DATA.LDfrac > cosd(bank_ref)) ...
        || (PredictedRange < DATA.THETNM && DATA.LDfrac < cosd(bank_ref))
    PredIn.Tswitch = PredIn.Tswitch - 1;
    iter = iter + 1;

% Else, if current L/D = skip L/D, adjusting transition
% time will make no difference in predicted trajectory
% Do not adjust transition time
elseif DATA.LDfrac == cosd(bank_ref)
    iter = max_iter;
end

% If transition should occur NOW or earlier and range is
% long, increase skip bank angle and reset transition time
if PredIn.Tswitch <= 0 && (PredictedRange > DATA.THETNM || ...
        FinalGamma > 0)
    bank_ref = bank_ref + 1;
    PredIn.LDratio2 = cosd(bank_ref)*DATA.CONST.LAD;
    PredIn.Tswitch = 0;
    iter = iter + 1;

% Else, if transition should occur NOW or earlier and range
% is short, decrease skip bank angle and reset transition time
elseif PredIn.Tswitch <= 0 && PredictedRange < DATA.THETNM
    bank_ref = bank_ref - 1;
    PredIn.LDratio2 = cosd(bank_ref)*DATA.CONST.LAD;
    PredIn.Tswitch = 0;
    iter = iter + 1;
end
%-------------------------------

% Run ballistic predictor with new transition time and/or skip bank angle
% Variables out to workspace: PredictedRange, FinalDrag,
%   FinalGamma, SkipawayFlag, Brange, Bdrag, Bvelocity, Bgamma,
%   Balt
clear B* tout;
assignin('base','PredIn',PredIn);
sim('predictor_ballistic_gamma');

....
```

Skip Bank Angle Correction

It is possible that the range requirement cannot be met simply by adjusting the time at which to transition to Upcontrol. If this is the case, the reference skip bank angle must also be modified slightly to meet the range requirement. The transition time is adjusted until the sign of the range error changes, indicating that the zero error point is in between the previous transition time guess and the current one.

Because the transition time is adjusted only in 1 second increments, the remaining range error must be eliminated by changing the desired skip bank angle. This portion of the corrector starts with either the current transition time guess or the previous guess—whichever yields the smaller range error (absolute value). If the predicted range is short, the desired skip bank angle is decreased slightly; if the range is too long, the skip bank angle is increased. For the first iteration, the increment added to the skip L/D is equivalent to a 0.5° change in the bank angle. For all subsequent iterations, the algorithm calculates the appropriate amount to change the skip L/D by computing a sensitivity in range to the previous change in the skip L/D. This sensitivity, $\frac{\Delta\left(L/D\right)}{\Delta\text{range}}$, greatly improves converge of the iterator. If the calculated amount to change the skip L/D is greater than the equivalent of 1°, the L/D change is limited to a 1° change from the previous value.

```
....

% CORRECTOR: SKIP BANK ANGLE
% If the sign of the range difference changes, that
% means the diff = 0 point is somewhere in between
if sign(PredRange_prev - DATA.THETNM) ~= sign(PredictedRange - ...
        DATA.THETNM) && FinalGamma < 0 && FinalGamma*180/pi > -10 && ...
        FinalGamma_prev < 0

    % If previous transition time guess had smaller range error
    % and did not skip away, use previous guess
    if abs(PredRange_prev - DATA.THETNM) <= abs(PredictedRange - ...
            DATA.THETNM) && FinalGamma_prev < 0 && ...
            FinalGamma*180/pi > -10 && Tswitch_prev >= 0

        PredIn.Tswitch = Tswitch_prev;
        PredictedRange = PredRange_prev;
    end

    % Adjust skip bank angle to match range requirement
    while abs(PredictedRange - DATA.THETNM) > DATA.CONST.TOL && ...
            count < max_count

        % If pass through this loop, change skip bank angle
        % by 0.5 deg
        if count == 0
            LDinc = (cosd(bank_ref - 0.5) - ...
                cosd(bank_ref))*DATA.CONST.LAD;
            LDratio_prev = PredIn.LDratio2;

            % If range is too short, increase skip L/D
```

```
                % Else, decrease skip L/D
                if PredictedRange < DATA.THETNM
                    PredIn.LDratio2 = PredIn.LDratio2 + LDinc;
                else
                    PredIn.LDratio2 = PredIn.LDratio2 - LDinc;
                end

            % If not first pass through loop, change skip bank
            % angle by [d(L/D)/dRange]prev * dRange
            else
                LDinc = (PredIn.LDratio2 - LDratio_prev)/...
                    (PredictedRange - PredRange_prev)*...
                    (DATA.THETNM - PredictedRange);

                % If current or previous iteration was a
                % skipaway, decrease skip L/D
                if FinalGamma > 0 || FinalGamma_prev > 0
                    LDinc = -abs(LDinc);
                end

                % Limit LDinc if bank angle change is greater than 1 deg
                if abs(LDinc) > abs((cosd(bank_ref + 1) - ...
                        cosd(bank_ref))*DATA.CONST.LAD)

                    LDinc = sign(LDinc)*abs((cosd(bank_ref + 1) - ...
                        cosd(bank_ref))*DATA.CONST.LAD);
                end

                LDratio_prev = PredIn.LDratio2;

                % Change skip L/D (bank angle)
                PredIn.LDratio2 = PredIn.LDratio2 + LDinc;
            end
            count = count + 1;  % increment counter

            % Record previous iteration's values
            PredRange_prev = PredictedRange;
            FinalGamma_prev = FinalGamma;

            % Run ballistic predictor
            % Variables out to workspace: PredictedRange, FinalDrag,
            %   FinalGamma, SkipawayFlag, Brange, Bdrag,
            %   Bvelocity, Bgamma, Balt
            clear B* tout;
            assignin('base','PredIn',PredIn);
            sim('predictor_ballistic_gamma');

        end
        % end WHILE abs(PredictedRange - DATA.THETNM) > DATA.CONST.TOL ...
        %           && count < max_count

    end
    % end CORRECTOR: SKIP BANK ANGLE
```

```
        end
        % end CORRECTOR: UPCONTROL TRANSITION TIME

        ....
```

Shallow Entry Test

The bulk of the "correction" phase ends here. The next segment of Initial Roll is the logic to determine whether the loft/skip entry is shallow. This logic was originally intended for situations in which the drag at the predicted transition to Upcontrol was too low—below 1 g. However, the logic has been expanded to include three other situations:

1. The final flight path angle is not within acceptable limits (positive or too negative)

   → Indicates a skipaway ($\gamma > 0$) or a crash

2. The bank angles for Initial Roll and the skip are too similar; they approximate a constant bank angle trajectory

   → Transition from Initial Roll to Upcontrol marked by a change in the bank angle. If there is very little or no change, the transition is difficult to predict. Often, this pushes the transition time far into the skip.

3. The maximum number of iterations to adjust the Upcontrol transition time has been reached

   → The range requirement has not been met using this combination of bank angles

In any of these three situations, two more conditions must be met in order to increase the Initial Roll bank angle to steepen the entry:

1. The desired skip bank angle is still at its initialized value (80°)

2. The Upcontrol transition time is before maximum drag, or the difference between the bank angles for Initial Roll and the skip is less than 10°

This section also decreases the desired skip bank angle if the predicted Upcontrol transition time is after maximum drag and the maximum number of iterations has been reached. This indicates that the predictor is trying to delay the transition as long as possible to increase the range but is still falling short. In this case, the desired skip bank angle is decreased to increase the downrange capability. This should keep the transition time closer to the maximum drag time. The T_{switch} initial guess for the next guidance cycle is reset to the maximum drag time.

```
....

% SHALLOW ENTRY TEST: Set flag to reduce current L/D fraction
% if predicted drag at transition to Upcontrol is < 1g

% Drag at transition to Upcontrol
upc_drag = interp1(Bdrag(:,1),Bdrag(:,2),PredIn.Tswitch);

% Enter this logic if upc_drag too low,
% final gamma not within acceptable limits,
% Initroll bank angle and skip bank angle are too similar,
% or max iterations reached (no solution found)
if upc_drag < 1*DATA.CONST.GS || FinalGamma > 0 || ...
        FinalGamma < -10*pi/180 || iter == max_iter || ...
        abs(DATA.BANKREF - acosd(DATA.LDfrac)) <= 10

    % If final gamma too negative, reset Upcontrol transition
    % time to TmaxDrag
    if FinalGamma < -10*pi/180
        PredIn.Tswitch = TmaxDrag;
    end

    % If skip bank angle is at initialized value and Tswitch
    % before max drag OR Initroll and skip bank angles too
    % similar, set flag to decrease Initial Roll L/D fraction
    if DATA.BANKREF == 80 && (PredIn.Tswitch < TmaxDrag || ...
            abs(DATA.BANKREF - acosd(DATA.LDfrac)) <= 10)

        DATA.RLOVER = 1;    % roll over

    % Else, if Tswitch after max drag and max # iterations reached,
    % decrease skip bank angle and reset transition time
    elseif iter == max_iter && PredIn.Tswitch > TmaxDrag && ...
            DATA.BANKREF <= 80

        DATA.BANKREF = DATA.BANKREF - 10;
        if DATA.BANKREF < 10
            DATA.BANKREF = 10;
        end
        PredIn.Tswitch = TmaxDrag;
    % Else, if Tswitch after max drag, solution found, and skip
    % bank angle below initialized value, reset skip bank angle
    elseif iter < max_iter && PredIn.Tswitch > TmaxDrag && DATA.BANKREF < 80
        DATA.BANKREF = 80;
    end

% If upc_drag not too low this guidance cycle but skip bank
% angle was decreased previously, increase it back to nominal
% value in 20 deg increments.
% Either way, set flag to stop increasing Initial Roll L/D fraction
elseif DATA.BANKREF < 80
    DATA.BANKREF = min(DATA.BANKREF + 20,80);
    DATA.RLOVER = 0;
else
    DATA.RLOVER = 0;
end
% end SHALLOW ENTRY TEST

....
```

Skip/Loft Determination and Transition to Upcontrol

The last segment of the loft/skip logic determines whether the trajectory is a true skip or a loft, and it determines whether the conditions are met to transition to Upcontrol. If the drag during the predicted trajectory decreases below 0.2 g's, the entry is considered to be a skip and $TRAJTYPE$ is set to 2. Otherwise, the entry is a loft and $TRAJTYPE$ is set to 1.

For simplicity, the EMT Predictor assumes an instantaneous transition from the Initial Roll bank angle to the skip bank angle. Because the vehicle obviously cannot perform that kind of maneuver, guidance commands the transition to Upcontrol several seconds early to allow the vehicle time to bank to the desired angle. The flight control system used for this thesis requires approximately 8 seconds to bank from 15° to 80°. This is an empirically-derived value and is specific to this flight control system.

The actual transition to Upcontrol is commanded if the following two conditions are met:

1. The predicted switch time, minus the time required to change the bank angle, falls either in this guidance cycle or before the next guidance cycle

2. Either the current drag or the drag prediction is at least 1 g

If the conditions are met, the transition to Upcontrol is commanded. Otherwise, the appropriate L/D is commanded and guidance continues to Lateral Logic. This ends the loft/skip logic to determine the Upcontrol transition time.

```
....

% Determine whether trajectory 'exits' the atmosphere (D < 0.2 g's)
jj = find(Bdrag(:,2) == max(Bdrag(:,2)),1);
out_of_atm = find(Bdrag(jj:end,2) < 0.2*DATA.CONST.GS);

if isempty(out_of_atm)
    DATA.TRAJTYPE = 1;  % LOFT
else
    DATA.TRAJTYPE = 2;  % SKIP
end

% Update VL, GAMMAL, Q7 for FinalRangeEst
DATA.VL     = Bvelocity(end,2);
DATA.GAMMAL = Bgamma(end,2);
DATA.Q7     = Bdrag(end,2);

% Update parameters
DATA.Tswitch = PredIn.Tswitch - 2;  % subtract length of guid cycle
DATA.LDratio2 = PredIn.LDratio2;
DATA.TmaxDrag = TmaxDrag;

% Time required to roll to desired loft/skip bank angle
time_to_roll = 8;   % empirically derived from FCS (15 - 80 deg)

% TEST: Are conditions met to transition to Upcontrol?
if DATA.Tswitch - time_to_roll <= 1 && ...
        (upc_drag >= 1*DATA.CONST.GS || DATA.D >= 1*DATA.CONST.GS)

    % YES, go to Upcontrol / PredGuid
    DATA.SELECTOR = 3;  % UPCONTRL
    DATA.NEXTSTEP = 7;  % UPCONTRL
    return;
else
    % NO, it is not time to go to Upcontrol

    if iter == max_iter
        DATA.Tswitch = TmaxDrag;    % reset initial guess for next guid cycle
    end
    % Command L/D
    DATA.LD = DATA.LDfrac*DATA.CONST.LAD;

    % Continue directly to LATLOGIC
    DATA.NEXTSTEP = 11; % LATLOGIC;
    return;
end
```

```
        end
        % end IF/ELSE DATA.TRAJTYPE == 0

        ....
```

B.3.3 Gamma Turn Controller

The last segment of the Initial Roll phase contains the Gamma Turn Controller. This logic is entered if the *GTC* flag is set to 1.

First, guidance calculates the current flight path angle and flight path angle rate, as well as the Constant Drag reference values for both. If any of the following conditions are met, the transition to Huntest is commanded.

1. $|\Delta\gamma| < 0.1°$ and $|\Delta\dot{\gamma}| < 0.05°/s$
2. The maximum number of guidance cycles for the GTC has been exceeded
3. The sign of $\Delta\gamma$ has changed from the previous guidance cycle
4. The current drag exceeds the maximum allowable

If none of the conditions are met, the L/D for the Gamma Turn Controller is calculated and commanded. This ends the portion of Initial Roll that is not Preentry Attitude Hold.

```
        ....

    else
        % YES, it is time to begin the Gamma Turn Controller

        % Calculate current flight path angle and reference flight path
        % angle for Constant Drag
        gamma      = asin(DATA.RDOT/DATA.V);                  % [rad]
        gamma_ref  = -2*DATA.CONST.HS*DATA.D0/DATA.V^2;       % [rad]
        gamma_diff = gamma - gamma_ref;                       % [rad]

        % Calculate current flight path angle rate and reference flight
        % path angle rate for Constant Drag
        gamma_dot      = (1/DATA.V)*(DATA.CONST.GS*cos(gamma)*...
            (DATA.V^2/(DATA.CONST.GS*DATA.CONST.RE) - 1) + ...
            DATA.CONST.LAD*cos(DATA.ROLLC)*DATA.D);               % [rad/s]
        gamma_dot_ref  = -4*DATA.CONST.HS*DATA.D0^2/(DATA.V^3); % [rad/s]
        gamma_dot_diff = gamma_dot - gamma_dot_ref;             % [rad/s]
```

```
% Conditions to transition to Huntest
a = (abs(gamma_diff) < 0.1*pi/180 && abs(gamma_dot_diff) < 0.05*pi/180);
b = (DATA.GTURNCT > 3);
c = sign(DATA.GDIFFPREV) ~= sign(gamma_diff);
d = DATA.D >= DATA.CONST.GMAX;

% TEST: Are any of conditions met to transition to Huntest?
if a || b || c || d

    % YES, transition to Huntest
    DATA.SELECTOR = 2;  % HUNTEST
    DATA.NEXTSTEP = 3;  % HUNTEST
    return;

else
    % NO, continue commanding the gamma turn

    % Calculate L/D command for Constant Drag
    LDref = (DATA.V*gamma_dot_ref - ...
        DATA.CONST.GS*(DATA.V^2/(DATA.CONST.GS*DATA.CONST.RE) - 1))/DATA.D;

    % Limit L/D command if necessary
    if abs(LDref) > DATA.CONST.LAD
        LDref = sign(LDref)*DATA.CONST.LAD;
    end

    % Calculate bank difference
    PHIref = sign(DATA.ROLLC)*acos(LDref/DATA.CONST.LAD);
    PHIdiff = PHIref - DATA.ROLLC;
    PHImax = 20*pi/180;    % max allowable

    % TEST: Is the bank angle difference greater than the maximum?
    if abs(ROLLdiff) > ROLLmax

        % YES, limit the L/D command
        DATA.LD = DATA.CONST.LAD*cos(DATA.ROLLC + sign(PHIdiff)*PHImax);

    else

        % NO, the L/D command does not need to be limited
        DATA.LD = LDref;

    end

    % Update stuff
    DATA.GTURNCT = DATA.GTURNCT + 1;    % iteration counter
    DATA.GDIFFPREV = gamma_diff;        % previous fpa difference

    % Continue to LATLOGIC
    DATA.NEXTSTEP = 11; % LATLOGIC
end
% end IF/ELSE (a || b || c || d)
```

```
    end
    % end IF (DATA.GTC ~= 0)

end
% end IF/ELSE (DATA.INRLSW == 1)

%==============
% end INITROLL
%==============
```

B.4 Huntest Phase

The Huntest phase is simply a wrapper for the Constant Drag subroutine. Huntest calculates the current energy per unit weight and compares it to the Final phase reference trajectory's initial energy per unit weight. If the current energy is less than the initial energy of the reference trajectory, Huntest transitions to the Final phase. Otherwise, Huntest continues to call Constant Drag to deplete energy.

```
%=========
% HUNTEST
%=========

function DATA = HUNTEST(DATA)

% Assign current values to predicted-quantity variables
% and calculate Final Phase range
DATA.VL     = DATA.V;            % velocity
DATA.GAMMAL = DATA.RDOT/DATA.V;  % flight path angle
DATA.Q7     = DATA.D;            % drag
DATA = FinalRangeEst(DATA);

% Calculate energy/weight: E/W = h + (1/2)*v^2/g
energy     = DATA.alt + 0.5*DATA.V^2/DATA.CONST.GS;
energy_ref = DATA.CONST.FTABLE.Y(end-1,7) ...
    + 0.5*DATA.CONST.FTABLE.x(end-1)^2/DATA.CONST.GS;

% Is current energy less than initial energy of reference trajectory?
if energy <= energy_ref

    % YES, transition to Final
    DATA.SELECTOR = 5;  % FINAL
    DATA.EGSW = 1;
    DATA.NEXTSTEP = 9;  % FINAL
    return;

else
```

```
    % NO, remain in Huntest and continue Constant Drag
    DATA.SELECTOR = 2; % HUNTEST
    DATA.NEXTSTEP = 6; % CONSTD

    DATA.NOSWIT = 1; % Disable lateral switches during this time

    return;

end
%=============
% end HUNTEST
%=============
```

B.4.1 Constant Drag Subroutine

The only change in the Constant Drag subroutine from its description in *Reentry Guidance with Extended Range Capability for Low L/D Spacecraft* is the replacement of the Constant Drag value, D_0, as a variable rather than a constant.

```
%========
% CONSTD
%========

function DATA = CONSTD(DATA)

% Calculate appropriate L/D command to maintain constant drag
DATA.LD = -DATA.LEQ/DATA.DO + DATA.CONST.C16*(DATA.D - DATA.DO) ...
    - DATA.CONST.C17*(DATA.RDOT + 2*DATA.CONST.HS*DATA.DO/DATA.V);

% Continue to Negative Test
DATA.NEXTSTEP = 12; % NEGTEST
return;

%============
% end CONSTD
%============
```

B.5 Upcontrol Phase

The Upcontrol phase is nearly identical to the Draper baseline version [2]. The only change is the replacement of Q_7 with Q_{7F} as the trigger to transition to the Ballistic phase. This is because Q_7 is continually updated as the predicted drag at the start of the Final phase, which for loft cases is significantly above 6 ft/s^2, the value of Q_{7F}. This could cause a transition to Ballistic in cases where the transition is not necessary or desirable.

```
%==========
% UPCONTRL
%==========
function DATA = UPCONTRL(DATA)

% TEST: Is altitude decreasing and has velocity reached
% the cutoff to move to final phase (VL + C18)?
if (DATA.RDOT < 0) && (DATA.V < DATA.VL + DATA.CONST.C18)

    % YES, altitude is decreasing and velocity has reached the cutoff
    % Continue directly to Final phase
    DATA.SELECTOR = 5; % FINAL
    DATA.EGSW = 1;
    DATA.NEXTSTEP = 9; % FINAL
    return

else

    % TEST: Is drag below cutoff (6 fpss) to go to Ballistic phase?
    if DATA.D < DATA.CONST.Q7F

        % YES, drag is below cutoff
        % Transition to Ballistic
        DATA.SELECTOR = 4; % BALLISTIC
        DATA.NEXTSTEP = 8; % BALLISTIC
        return

    else

        % NO, drag is not below cutoff
        % Run PredGuid to get bank angle
        DATA = PREDGUID(DATA);

        % Continue to Negative Test
        DATA.NEXTSTEP = 12; %NEGTEST

        return
    end
end

%==============
% end UPCONTRL
%==============
```

B.6 Ballistic Phase

There are two elements in the Ballistic phase that are different than the phase description in *Reentry Guidance with Extended Range Capability for Low L/D Spacecraft*:

1. The drag trigger to transition to Final has been changed from Q_7+0.5 ft/s^2 to Q_{7F}+0.5 ft/s^2. The reason is the same as for the replacement in Upcontrol—Q_7 is continually updated throughout the Initial Roll, Upcontrol, and Ballistic phases.

2. PredGuid's minimum input distance is set to 0 nm if PredGuid has not previously been shut off. This is for skips in which the drag never drops below 0.05 g's.

```
%===========
% BALLISTIC
%===========

function DATA = BALLISTIC(DATA)

% TEST: is drag than greater minimum drag plus a certain increment (Q7F + KDMIN)?
if DATA.D > (DATA.CONST.Q7F + DATA.CONST.KDMIN)

    % YES, drag is larger than the reference drag.
    % i.e. vehicle is back in the atmosphere.

    % Go to Final phase
    DATA.EGSW = 1;
    DATA.SELECTOR = 5;  % FINAL
    DATA.NEXTSTEP = 9;  % FINAL
    return;

else
    % NO, drag is NOT larger than reference drag.
    % i.e. still outside the atmosphere

    % TEST: is drag high enough to continue steering?
    if DATA.D > DATA.CONST.ASENSIBLE
        % YES, drag is high enough to continue steering

        % If drag has not previously dropped below ASENSIBLE (PredGuid
        % not previously off), set PG_tgt_min to 0
        if DATA.PG_off == 0
            DATA.CONST.PG_tgt_min = 0;  % nm
        end

        % Run PredGuid to get bank angle
        DATA = PREDGUID(DATA);
    else
        % NO, drag is too low to run PredGuid
        % Reset PG_tgt_min back to 200 nm
        DATA.PG_off = 1;    % indicates PredGuid not run
        DATA.CONST.PG_tgt_min = 200;    % nm

    end

end

% Continue to Lateral Logic to manage crossrange error
DATA.NEXTSTEP = 11;  % LATLOGIC
return;

end

%===============
% end BALLISTIC
%===============
```

B.7 PredGuid Subroutine

PredGuid has been altered from its description in *Reentry Guidance with Extended Range Capability for Low L/D Spacecraft*. The most significant change is an additional integration termination condition. This is indicated by the name of the Simulink model. In addition, the minimum range necessary to run PredGuid (`PG_tgt_min`) has been increased from 100 *nm* to 200 *nm*. This value, however, can be reduced to 0 *nm* in the Ballistic phase for skip cases in which PredGuid is never shut off. If PredGuid's target range is less than the minimum, the subroutine outputs a neutral lift command instead of maintaining the previous guidance cycle's bank command.

A few smaller things have also been changed:

1. PredGuid's initial guess for the appropriate bank angle command (`PG.CPhi_Desired`) is the EMT Predictor's last guess at the end of Initial Roll instead of 0

2. `PG.Q7` is calculated using Q_{7F} instead of Q_7

3. The upper limit on PredGuid's target miss to record the predicted values at the end of the skip has been reduced from 1000 *nm* from 100 *nm*

4. Predicted drag is also recorded (in addition to velocity and flight path angle) for cases in which PredGuid's target miss is less than the limit

5. The sign of γ_L has been changed to accommodate a sign change in the FinalRangeEst function

```
%==========
% PREDGUID
%==========

function DATA = PREDGUID(DATA)

% Estimate Final phase range
DATA = FinalRangeEst(DATA);

% Calculate remaining range to reference trajectory
DATA.PG_target_range = DATA.THETNM - DATA.final_range;

% TEST: Is remaining range to reference trajectory greater than minimum or
% is this the first call to PredGuid?
if DATA.PG_target_range > DATA.CONST.PG_tgt_min || DATA.PIND == 0

    % YES -> must run PredGuid

    % Assign values before running PredGuid
    c = DATA.CONST.c;

    c.APOGEE_TARGET = DATA.PG_target_range;
    c.EARTH_POLE    = DATA.UZbar;

    PG.Cd_est       = c.CD_EST_INITIAL;
    PG.LD_est       = DATA.CONST.LAD;
    PG.Krho_est     = DATA.Krho_est;
    PG.CPhi_Desired = DATA.LD/DATA.CONST.LAD;
    PG.Sign_Of_Bank = 1;

    PG.Position     = DATA.Rbar;
    PG.Velocity     = DATA.VIbar;
    PG.Acceleration = DATA.Dbar;
    PG.Altitude     = DATA.alt;
    PG.Velocity_Mag = abval(DATA.VIbar);
    PG.Q7           = DATA.CONST.Q7F + DATA.CONST.KDMIN;

    PG.R_ini = DATA.R_ini;
    PG.V_ini = DATA.V_ini;
    PG.VLMIN = c.VLMIN;

    if (DATA.RDOT < 0) && (DATA.D > PG.Q7)
        % i.e. vehicle has not passed pullout (rdot = 0)
        % (only used for initialization)
        PG.IND_ini = 0;   % ‘skip has begun’ flag = false
    else
        PG.IND_ini = 1;   % ‘skip has begun’ flag = true
    end

    c.LIFT_INC_CAPTURE = -3;
    c.MAX_NUMBER_RUNS  = 10;

    % If first call to PredGuid, initialize additional variables
    if DATA.PIND == 0
        PG.CPhi_Desired    = DATA.LDratio2/DATA.CONST.LAD;
        c.LIFT_INC_CAPTURE = -10;
        c.MAX_NUMBER_RUNS  = 20;
        DATA.PIND          = 1;
    end

    assignin(‘base’,‘c’,c);
    assignin(‘base’,‘PG’,PG);
    % ------------------------------
    % Run PredGuid
    sim(‘PredGuidTarget_gamma.mdl’);
    % ------------------------------
    % Process PredGuid data
    DATA.CPHI = CPHI;
    DATA.LD   = DATA.CPHI*DATA.CONST.LAD;

    DATA.PG_target_miss      = TARGET_MISS;
    DATA.PG_target_tolerance = TARGET_TOLERANCE;
    DATA.PG_num_iter         = N_RUNS;

    % Record predicted quantities at end of skip/loft:
    if abval(DATA.PG_target_miss) < 100
        DATA.VL     = VELMAG;                 % velocity
        DATA.GAMMAL = RDOT/VELMAG;            % flight path angle
        DATA.Q7     = AMAG(length(AMAG));    % drag
    end

else
    % NO, remaining range to reference trajectory less than PG_tgt_min
    % Output neutral lift command
    DATA.CPHI = cosd(DATA.CONST.PG_lim_phi);   % lift neutral
    DATA.LD   = DATA.CPHI*DATA.CONST.LAD;
end

%===============
% end PREDGUID
%===============
```

B.7.1 Final Range Estimation

The FinalRangeEst subroutine estimates the range traveled during the Final phase. It uses the predicted velocity, flight path angle, and drag at the start of the Final phase to adjust the nominal range. For instance, if the predicted velocity is equal to the nominal velocity at the start of the reference trajectory but the flight path angle is shallower than the nominal, the estimated Final phase range will be longer than the nominal range corresponding to the predicted velocity. This routine is used in several places throughout the algorithm: Initial Roll, Huntest, and PredGuid. A similar estimation routine is used in the RT Selector.

This function requires the predicted flight path angle at the start of the Final phase as one of its inputs. Assuming a truly ballistic trajectory, the flight path angle at the start of the skip should be equal in magnitude but opposite in sign to the flight path angle at the end of the skip. The Draper baseline version of this function assumed the flight path angle input was from the start of the skip, which is positive. In the Predictor algorithm, this range estimator is used for more than just skip entries, so the routine was changed to assume the input flight path angle is at the start of the Final phase, so it will be negative for skip entries. The code for this subroutine is shown below.

```
%===============
% FinalRangeEst
%===============
function DATA = FinalRangeEst(DATA)

% NOTE: GAMMAL is assumed to be the flight path angle at entry to Final
% phase, NOT the flight path angle at Upcontrol exit. Therefore, GAMMAL
% should be negative, not positive.

% TEST: Is VL greater than initial velocity of Final phase ref traj?
if DATA.VL > DATA.CONST.FTABLE.x(length(DATA.CONST.FTABLE.x) - 1)

    % YES, VL is greater than reference initial velocity. Project upward
    % from highest point (Apollo method)
    DATA.ASP1 = DATA.CONST.Q2 + DATA.CONST.Q3*DATA.VL;          % FINAL PHASE RANGE
    DATA.ASP3 = DATA.CONST.Q5*(DATA.CONST.Q6 + DATA.GAMMAL);    % GAMMA CORRECTION
    DATA.final_range = DATA.ASP1 + DATA.ASP3;   % estimated Final phase range

else
    % NO, VL is less than ref traj initial velocity
    % Interpolate in the table, just as in Final phase
    DATA.yi = interp1(DATA.CONST.FTABLE.x,DATA.CONST.FTABLE.Y,DATA.VL,'linear');
    % Assignment of values:
    DATA.RDOTREF = DATA.yi(1);  % altitude rate
    DATA.DREFR   = DATA.yi(2);  % drag
    DATA.F2      = DATA.yi(3);  % dRange/d(Altitude rate)
    DATA.F1      = DATA.yi(4);  % dRange/dDrag
    DATA.RTOGO   = DATA.yi(5);  % range to go
    DATA.F3      = DATA.yi(6);  % dRange/d(L/D)

    DATA.ASP1 = DATA.RTOGO;
    DATA.ASP3 = (DATA.GAMMAL*DATA.VL - DATA.RDOTREF)*DATA.F2;
    DATA.PREDANGL = DATA.RTOGO + DATA.F2*(DATA.GAMMAL*DATA.VL - DATA.RDOTREF) ...
        + DATA.F1*(DATA.Q7 - DATA.DREFR);
    DATA.final_range = DATA.PREDANGL;   % estimated Final phase range

end

% Final phase range should not be negative
if DATA.final_range < 0
    DATA.final_range = 0;
end
%===================
% end FinalRangeEst
%===================
```

B.8 Lateral Logic Subroutine

The only change to Lateral Logic is to prohibit the subroutine from commanding bank reversals if the *NOSWIT* flag is set to 1. *NOSWIT* is reset to 0 at the end of each call to Lateral Logic.

Bibliography

[1] *Apollo 13.* Dir. Ron Howard. Universal City Studios, Inc., 1995.

[2] S. H. Bairstow, *Reentry Guidance with Extended Range Capability for Low L/D Spacecraft.* S.M. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, February 2006.

[3] D. R. Chapman, "An Approximate Method for Studying Entry into Planetary Atmospheres," NACA Technical Note 4276, May 1958.

[4] J. L. DiCarlo, *Aerocapture Guidance Methods for High Energy Trajectories.* S.M. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, May 2003.

[5] C. A. Graves and J. C. Harpold, "Apollo Experience Report – Mission Planning for Apollo Entry," NASA TN D-6725, March 1972.

[6] J. Greathouse, R. Lillard, "CEV CM OML, Aerodynamic, and Aerothermal Database Background and Usage Information," Rev. 7, 1 March 2006.

[7] "Human-Systems Integration Requirements (HSIR)," NASA CFI Release Draft, 30 December 2005.

[8] S. Lim, R. Pileggi, and G. Barton, "MIMO Adaptive Bank-To-Steer Control Algorithms for Guided Re-Entry Vehicles," AIAA Paper 2007-6431, AIAA Guidance, Navigation, and Control Conference and Exhibit, Hilton Head, SC, 20-23 August 2007.

[9] R. Morth, "Reentry Guidance for Apollo," Massachusetts Institute of Technology Instrumentation Laboratory, R-532, Cambridge, MA, January 1966.

[10] P. E. Moseley, "The Apollo Entry Guidance: A Review of the Mathematical Development and its Operational Characteristics," TRW Note No. 69-FMT-791, 1 December 1969.

[11] "NASA's Exploration Systems Architecture Study," NASA-TM-2005-214062, November 2005, p. 218.

[12] R. W. Orloff, *Apollo by the Numbers: A Statistical Reference.* NASA SP-2000-4029, NASA History Division, Washington, D.C., 2000.

[13] J. I. Petty and K. Dismukes, "The Vision For Space Exploration: Technology Imagery," NASA Human Spaceflight,
http://spaceflight.nasa.gov/gallery/images/vision/technology/ndxpage1.html,
JSC2006-E-21479 (June 2006) and JSC2007-E-20978 (May 2007) [accessed 21 May 2007].

[14] "President Bush Announces New Vision for Space Exploration," White House Presidential News and Speeches, 14 January 2004,
http://www.whitehouse.gov/news/releases/2004/01/20040114-3.html
[accessed 13 May 2007].

[15] M. E. Tauber and K. Sutton, "Stagnation-Point Radiative Heating Relations for Earth and Mars Entries," *Journal of Spacecraft and Rockets*, Vol. 28, No. 1, 1991, pp. 40–42.

[16] C. Tillier, image file: Skip_reentry_trajectory.svg. Wikipedia online encyclopedia,
http://commons.wikimedia.org/wiki/Image:Skip_reentry_trajectory.svg,
28 October 2006 [accessed 10 April 2007].

[17] *United States Standard Atmosphere, 1962.* U.S. Government Printing Office, Washington, D.C., 1962.