Aerocapture Guidance Methods for High Energy Trajectories

by

Jennifer L. DiCarlo
B.S. Aeronautical and Astronautical Engineering
United States Air Force Academy, 2001

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

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2003

Signature of Author: ____

Department of Aeronautics and Astronautics
23 May, 2003

Approved by: ____

Gregg H. Barton
Principal Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by: ____

George T. Schmidt, Sc.D.
Lecturer, Department of Aeronautics and Astronautics
Director, Education, The Charles Stark Draper Laboratory, Inc.
Thesis Advisor

Accepted by: ____

Edward M. Greitzer, Ph.D.
H.N. Slater Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

[This Page Intentionally Left Blank]

Aerocapture Guidance Methods for High Energy Trajectories

by

Jennifer L. DiCarlo

Submitted to the Department of Aeronautics and Astronautics on
23 May 2003, in partial fulfillment of the requirements
for the Degree of Master of Science in Aeronautics and Astronautics

Abstract

This thesis investigates enhancements of an existing numerical predictor-corrector aerocapture guidance algorithm (PredGuid). The study includes implementation of an energy management phase prior to targeting with a generic method of transition and replacement of heuristic features with more generic features.

The vehicle response during energy management was modeled as a second-order spring/mass/damper system. Phase change occurred when two conditions were met: First, the vehicle could fly a constant bank angle of 110° for the remainder of the trajectory and have the resulting apogee below or within a given tolerance above the target apogee. Second, the predicted final energy indicated that the vehicle would be on an elliptical, not hyperbolic, trajectory. So as to incorporate generic features, modeling of a separate lift down phase was replaced by using a lift-down condition to determine phase change and biasing to the same lift-down condition during targeting. Also, use of a heuristic sensitivity to calculate the first corrected bank angle was replaced by a simple 'smart guessing' algorithm. Finally, heuristic lateral corridor boundaries were replaced by boundaries based on percentage of forward velocity.

Analysis of the resulting entry corridor revealed that the enhanced algorithm generated a mean improvement of 261% over the PredGuid corridor. Most of the gain occurred for steeper flight path angles. Results also indicated that the enhanced algorithm yielded lower maximum accelerations and comparable heating rates, heating loads, and ΔV to raise perigee. These results are intended to provide a starting point for further enhancement and applicability to interplanetary travel.

Technical Supervisor:	Gregg H. Barton
Title:	Principal Member of the Technical Staff, The Charles Stark Draper Laboratory, Inc.
Thesis Supervisor:	George T. Schmidt, Sc.D.
Title:	Lecturer, Department of Aeronautics and Astronautics Director, Education, The Charles Stark Draper Laboratory, Inc.

[This Page Intentionally Left Blank]

Acknowledgements

First and foremost, I would like to thank the Charles Stark Draper Laboratories for providing me with the opportunity to pursue a graduate degree at MIT. Many thanks go to Gregg Barton, my thesis advisor at Draper. Though you missed the daily thrill of plot after plot arriving on your desk, I assure you, I generated thousands. I hope that someday you actually get to use this stuff! Thanks to George Schmidt, my MIT thesis advisor, for the help on such short notice. Eternal gratitude and thanks go to Doug Fuhry who helped me more than I can even say. You never ceased to amaze me by being able to simply *look* at a plot and *know* what was wrong with my code. You saved me many hours of banging my head against a wall! And to everyone else at Draper who answered any of my endless (and sometimes stupid) questions along the way, thank you! I feel privileged to have known so many wonderful and intelligent individuals.

Next, I would like to thank the '03 Fellows who made sure the last two years were not *all* work. I couldn't have asked for a better group of people to go to school with! Geoff, you were the greatest officemate I have ever had. Okay, so you are the only officemate I've ever had—but you were still great! Thanks for all the talks, debates, and swimming advice (Coach Geoff!), and I wish you the best of luck in everything you do. Kim, thanks for swimming with me and for listening while we kicked all those yards. Believe it or not, I had a lot of fun those last few weeks of late nights in the lab. It was nice having someone else around to share it with! Steve, good luck in the Navy! Christine, I will never see a salt bagel for the rest of my life without thinking of you. Good luck with the Ph.D. Dave, thanks for all the optimal control help and for just being an all-around swell guy. Stuart, good luck in LA and maybe I'll see you at AFA down the road. To the '04 Fellows: Steve, Drew, and Dave, thanks for the fun year and best of luck finishing up!

I would also like to thank the people *outside* of the lab who were my distractions and let me lean on their shoulders and talk their ears off during my time in Boston. Thanks to the Boston Hash House Harriers for helping me keep my sanity, as backwards as that may sound. I will never forget seeing so many familiar cheering faces at Mile 20! There will always be crash space for you if your hash travels ever take you to my neck of the woods. On on! Sandy, thanks for all the rides, baseball, hockey, beer, and Thai and for getting me involved with GBTC. Without you I would never have run the Boston Marathon! Stephanie, my roommate, thanks for letting me invade your apartment. I know my room sometimes looked like a war zone and I left more stuff hanging around than I can even remember, but I really enjoyed my time there. Chiara, thanks for all the wine, cheese, and lemon square nights, countless rides home, being my e-mailing buddy, and generally just being there! Why we never became friends sooner I may never figure out. Steve, thanks for sticking by me for a few more years and for being there in the middle of the night when I needed someone to talk to. Thanks for letting me constantly talk through my ideas with you and for listening to my delirious recounting of numbers scrolling across the screen. Lyle, I don't know if thanks is enough to say. It wasn't all fun, but it was enlightening. In knowing you I have learned more about myself than I ever thought possible. Thanks for all the beers, baseball, football, wings, e-mail, phone calls, advice, running, favors, rides, editing, calming talks, and everything else you've done for me! Can I say another round of thanks for all the beer? We should have kept track...it would have been fun—scary, but fun!

Finally, I would like to thank Mom, Dad, Carolyn, and the rest of my family for their love and support. I know you all think I'm just plain crazy, but you love me anyway! Thanks for sticking by me no matter what lunatic idea I come up with next.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc. under Internal Research and Development, Project GCDLF-Support, 03-0-5043.

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

As a member of the Air Force, I acknowledge that the views expressed in this thesis are mine and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

Jennifer L DiCarlo

Jennifer L. DiCarlo

2nd Lieutenant, USAF

Table of Contents

[This Page Intentionally Left Blank]

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Aerocapture

Aerocapture is an expeditious, fuel-efficient method of orbit insertion. Payload and vehicle mass are often limited by the amount of chemical propellant required for propulsive orbit insertions. As a result, methods reducing the required propellant are highly desirable. The most readily available, non-propulsive, means of slowing a vehicle from hyperbolic speeds to low orbital speeds is atmospheric drag. Variants of this aeroassist approach have surfaced over the last several decades. One popular variant, aerobraking, uses a propulsive maneuver to insert the vehicle into a high elliptical orbit. Multiple passes through the atmosphere slowly remove velocity in small increments until the orbit is circular. While aerobraking can considerably reduce the amount of propellant required, a significant amount is still necessary for the initial orbit insertion.

Aerocapture completely eliminates the need for propulsive insertion by use of a single pass through the atmosphere to slow a vehicle from the high-energy approach trajectory to a low-energy orbit. Drag forces remove energy until the vehicle is slowed below escape velocity. Unlike aerobraking, there is no intermediary elliptical orbit. Thruster firings are used in the process, but they are limited to controlling orientation and provide no reduction in velocity. Figure 1.1 highlights differences between the two techniques.

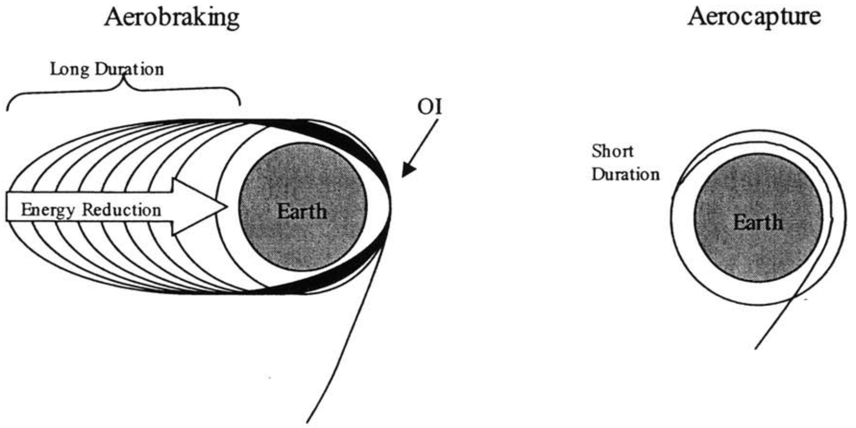


Figure 1.1: Aerocapture vs. Aerobraking

Numerous studies have been conducted on the benefits of aerocapture. Reference [4] compared aerobraking and aerocapture for the Mars Orbiter. These studies indicated a reduction in total vehicle mass from 1600 kg to 1100 kg and a reduction in time required for orbit insertion from 3 months to 3 hours through use of aerocapture.

Even with these promising figures and availability of the technology since the Apollo missions, aerocapture has never been used. The Apollo Program had a technique on paper similar to aerocapture, allowing entry capsules to enter the atmosphere and exit at near-orbital velocity as a method of landing area weather avoidance. However, weather avoidance was never required during the program, so the technology was developed and man-rated but not demonstrated [3].

1.2 Previous Research

Several proposals for an aerocapture flight demonstration have emerged in the last decade, including the Aeroassist Flight Experiment in the early 1990's and more recently, the joint NASA-CNES Mars Sample Return Mission [3]. These proposals led to the development of several important algorithms used in simulations to verify aerocapture. They are an analytic predictor-corrector, a numeric predictor-corrector, a terminal point controller, and an energy controller.

1.2.1 Analytic Predictor-Corrector

The analytic predictor-corrector is the second phase of a two-phase aerocapture maneuver. Dividing the aerocapture maneuver into two phase allows separate control of trajectory loads and apogee targeting. An equilibrium glide phase comprises the first part of the trajectory and the analytic predictor-corrector is used for the exit phase. In the equilibrium glide phase, the commanded bank angle is modeled with a linear second-order differential equation for altitude. After the vehicle has been slowed to a particular velocity, it transitions to the exit phase.

The analytic integration used by the predictor-corrector reduces the onboard computer requirements as compared to a numerical method. Analytic integration is possible by using altitude rate, $\dot{h}$, as the only control variable. Reference [1] contains a detailed derivation of the analytic equations of motion. These analytic equations predict the vehicle relative velocity at exit, assuming a constant altitude rate. Relative velocity is converted to inertial velocity and the predicted altitude rate and inertial velocity at exit are used to predict the resulting apogee. The algorithm iterates on the initial altitude rate until the desired apogee is achieved. This altitude rate is used to calculate commanded bank angle with an equation similar to the control equation from the equilibrium glide phase.

Lateral control determines the sign of the bank angle in order to keep the orbital plane as close as possible to the target orbital plane. Since there is only one control variable, bank angle, it is impossible to simultaneously null both position and velocity errors. However, it is possible to minimize both errors by nulling the wedge angle, δ, which is the angle between the actual and desired orbital planes.

References [3] and [2] provide a more detailed description of this analytic predictor-corrector method.

1.2.2 Numeric Predictor-Corrector

The numeric predictor-corrector algorithm controls the orientation of the lift vector about the relative velocity vector by altering the bank angle. The more deceleration required, the deeper the vehicle will penetrate into the atmosphere. The algorithm numerically integrates the current position and velocity vectors forward to atmospheric exit using a constant bank angle assumption. At each integration step, gravitational and aerodynamic accelerations are calculated using simplified models. The predicted states are used to calculate the resulting apogee. Bank angle is then adjusted to null the target apogee miss and this cycle is continued until the final apogee is within acceptable tolerances.

Lateral control, as with the analytic predictor-corrector, determines the sign of the bank angle in order to keep the orbital plane as close as possible to the target orbital plane. As discussed in the last section, it is impossible to simultaneously null both position and velocity errors. This algorithm controls the velocity error due to the short duration of the aerocapture trajectory.

The numeric predictor-corrector is described in reference [1] and is the basis for the aerocapture algorithm used in this thesis.

1.2.3 Terminal Point Controller

The terminal point controller algorithm attempts to drive the vehicle to follow a predetermined reference trajectory to a fixed terminal point or set of terminal conditions. The most crucial part of using a terminal point controller is generating an optimal reference trajectory. Simple feedback guidance schemes correct for dispersions and other errors along the trajectory.

As with the numeric and analytic predictor-corrector methods, the terminal point controller maintains lateral control through the use of predetermined limits on corridor error.

The terminal point controller is described in references [7] and [8].

1.2.4 Energy Controller

In the energy controller algorithm, the vehicle energy is controlled to a targeted energy state (determined by the target apogee) by altering the energy gain. The energy gain is calculated by taking the ratio of energy rate (a function of drag) to energy error. The gain is controlled so that the error and the energy rate approach zero. Energy gain is translated into altitude rate and the bank angle is determined using an analytical equation for vehicle vertical acceleration.

Lateral control is similar to that used in the other algorithms. If the out of plane velocity exceeds a given deadband, the vehicle executes a roll reversal to reduce the out of plane velocity error.

This algorithm is described in detail, including all relevant derivations, in reference [1].

1.3 Problem Definition and Thesis Objective

Over the last two decades, The Charles Stark Draper Laboratory (CSDL) has investigated the aerocapture concept. In 1984, John Higgins developed the original numeric predictor-corrector targeting guidance algorithm for Earth orbit transfer applications [1]. This algorithm has been periodically updated for use in various aerocapture guidance proposals but has its limitations. An aerocapture vehicle is capable of capturing within a certain range of flight path angles. The current application of Higgins' algorithm, PredGuid, is very useful for low-energy, shallow trajectories. As the energy of the orbits increases and the entry flight path angles become steeper, the portion of the theoretical corridor successfully captured by the algorithm decreases. The *theoretical* corridor is defined by the full lift up and full lift down flight path angle boundaries as well as structural and thermal constraints. The *capturable* corridor is the percentage of the theoretical corridor for which the algorithm allows the vehicle to aerocapture successfully within target, structural, and thermal constraints. These corridors will be further described in Chapter 4.

One major reason for the breakdown of Higgins' algorithm at high-energy cases is the inability of the algorithm logic to handle hyperbolic trajectories. This algorithm calculates updated bank angle guesses based on a slope calculated from previous guesses. Hyperbolic trajectories have negative values for apogee and semi-major axis which are incorrectly interpreted by the algorithm as 'low' misses. The second reason for high-energy case failure is the constant bank angle assumption. The algorithm's only recourse for reducing energy is digging deeper into the atmosphere which in turn raises acceleration loads to unacceptable levels.

In his 1988 Master of Science thesis, Doug Fuhry added an energy management capability to Higgins' algorithm with a numerical predictor-corrector entry phase targeting into a constant altitude cruise phase [6]. At a specified velocity the vehicle would then transition to Higgins' targeting algorithm. These improvements allowed enhanced aerocapture coverage for applicability at Mars. However, the algorithm was fairly complex, containing three phases for the aerocapture maneuver.

Although Higgins' numeric predictor-corrector method is generic in nature and can be used for any planet or moon with an atmosphere, the algorithm was *designed* for use at *Earth*. Similarly, Fuhry's method, like those discussed in the previous section, while generic in concept, was *specialized* for aerocapture at *Mars*. Through empirical observations from thousands of test cases, these algorithms incorporated new heuristic approximations and features for optimization. Use of the existing algorithms for a different case would involve empirically determining and I-loading a large number of factors. However, a more generic algorithm, capable of expanding the capturable corridor and easily applicable to different planets and moons with atmospheres, is desirable.

This thesis seeks to enhance the numerical predictor-corrector aerocapture guidance algorithm (PredGuid) by implementing a single energy management phase prior to targeting, developing a generic method of transitioning between the energy management and targeting phases, and replacing other heuristic features with more generic features. The resulting flight path angle entry corridor will be compared to the flight path angle entry corridor of the original algorithm. Additionally, comparisons will be made for various characteristics of the trajectories including maximum g load, heating rate, heating load, and ΔV to raise perigee. The resulting algorithm is intended to provide a starting point for further enhancement for applicability to interplanetary travel.

1.4 Thesis Overview

This thesis contains details and results of the design and testing of the enhanced aerocapture guidance algorithm discussed above.

Chapter 2, Equations of Motion, describes the equations used in the simulation dynamics. Reference frames, coordinate systems, assumptions, environmental models, and vehicle models used in the dynamics are described.

Chapter 3, Guidance Design, describes the algorithm used in the aerocapture guidance. Design factors, guidance phases, phase change criteria, and important alterations to the original algorithm are described.

Chapter 4, Capture Envelope and Test Case Determination, describes the theoretical and capturable corridors. Corridor approximations, vehicle constraint concerns, and determination of initial conditions are described.

Chapter 5, Results, discusses the results of the various cases tested. Comparisons to the original algorithm as well as evaluation of the enhanced algorithm are described.

Chapter 6, Conclusions, discusses conclusions of this study as well as suggestions for future study.

Chapter 2
Equations of Motion

2.1 Overview

Equations of motion model the environment to provide an accurate prediction of the vehicle trajectory within the simulation. Components of the equations of motion are computed in different reference coordinate frames, making it necessary to define both the frames and relevant rotation and transformation matrices between them. This chapter provides a brief guide to the reference frames and rotations used in this thesis as well as the basic equations of motion used in the environment models.

2.2 Reference Coordinate Frames

Inertial Reference Frame ($\hat{i}_i, \hat{j}_i, \hat{k}_i$): a non-rotating Earth-centered coordinate system in which the origin lies at the center of the Earth. The $\hat{i}_i$ axis points through zero longitude at time zero, the $\hat{k}_i$ axis points through the North Pole, and the $\hat{j}_i$ axis completes the right-handed coordinate system.

Local Horizontal Reference Frame ($\hat{i}_h, \hat{j}_h, \hat{k}_h$): a coordinate system with the origin at the vehicle center of gravity (CG). The $\hat{j}_h$ axis is in the direction of the angular momentum vector calculated from the relative velocity ($\vec{V}_{rel} \times \vec{R}$). The $\hat{k}_h$ axis points towards the center of the Earth along the vehicle inertial position vector, and $\hat{i}_h$ completes the right-handed coordinate system.

Velocity Reference Frame ($\hat{i}_v, \hat{j}_v, \hat{k}_v$): a coordinate system with the origin at the vehicle CG and the $\hat{i}_v$ axis pointing along the vehicle relative velocity vector. The $\hat{j}_v$ axis remains in the local horizontal and the $\hat{k}_v$ axis completes the right-handed coordinate system.

Body Reference Frame ($\hat{i}_b, \hat{j}_b, \hat{k}_b$): a coordinate system with the origin at the vehicle CG. The $\hat{i}_b$ axis points along the longitudinal axis of the vehicle. The $\hat{j}_b$ axis is positive out the 'right wing'. The $\hat{k}_b$ axis completes the right-handed coordinate system and points positive downward (towards Earth).

Stability Reference Frame ($\hat{i}_s, \hat{j}_s, \hat{k}_s$): a coordinate system with the origin at the vehicle CG with the $\hat{i}_s$ axis along the projection of the velocity vector ($\vec{V}$) onto the body $\hat{i}_b$ - $\hat{k}_b$ plane. Assuming zero sideslip, the $\hat{i}_s$ and $\hat{i}_v$ axes are coincident. The $\hat{j}_s$ axis is coincident with the $\hat{j}_b$ axis and the $\hat{k}_s$ axis completes the right-handed coordinate system.

2.3 Coordinate Transformations

Vector transformations are accomplished by taking the dot product of a vector in a given frame with the basis vectors of a desired reference frame. This operation can be written in matrix form as a multiplication of a transformation matrix (denoted T_{a2b}) with a vector. The transformation matrix contains the unit vectors of the given frame expressed in the desired reference frame. For example, a transformation matrix from reference frame A to reference frame B would contain the basis vectors of A written in the B frame as shown in Eq. 2.1.

$$\begin{bmatrix} V_{b1} \\ V_{b2} \\ V_{b3} \end{bmatrix} = \begin{bmatrix} b_1 \cdot a_1 & b_1 \cdot a_2 & b_1 \cdot a_1 \\ b_2 \cdot a_1 & b_2 \cdot a_2 & b_2 \cdot a_2 \\ b_3 \cdot a_1 & b_3 \cdot a_2 & b_3 \cdot a_3 \end{bmatrix} \begin{bmatrix} V_{a1} \\ V_{a2} \\ V_{a3} \end{bmatrix} \tag{2.1}$$

Reference frames are generally related by an easily identifiable angle rotated about one axis. As a result, transformation matrices are most frequently seen as combinations of sines and cosines on two axes while the third axis remains aligned with the given frame. Such transformation matrices are referred to as Euler rotations and are extremely useful in aerospace applications. Roll, pitch, and yaw Euler rotations are the most common. Rotation or transformation in the opposite direction is accomplished by simply using the transpose of the relevant matrix. Eq. 2.2 and 2.3 demonstrate this principle.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = T_{a2b} \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} \tag{2.2}$$

$$\begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} = T_{a2b}^T \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \tag{2.3}$$

Velocity to Inertial Transformation: The transformation matrix between the velocity and inertial reference frame contains the unit vectors of the velocity reference frame written in the inertial frame.

$$T_{i2v} = \begin{bmatrix} \uparrow & \uparrow & \uparrow \\ \hat{i}_v & \hat{j}_v & \hat{k}_v \\ \downarrow & \downarrow & \downarrow \end{bmatrix} \tag{2.4}$$

where:

$$\hat{i}_v = \frac{\vec{v}_{rel}}{\left|\vec{v}_{rel}\right|} \tag{2.5}$$

$$\hat{j}_v = \frac{\vec{v}_{rel} \times \vec{r}}{\left|\vec{v}_{rel} \times \vec{r}\right|} \tag{2.6}$$

$$\hat{k}_v = \hat{i}_v \times \hat{j}_v \tag{2.7}$$

Bank Angle Rotation: The stability reference frame is the velocity reference frame rotated by only the bank angle ϕ about their common $\hat{i}$ axes. This relationship is shown in Figure 2.1 below and the transformation matrix between the frames is given in Eq. 2.8.

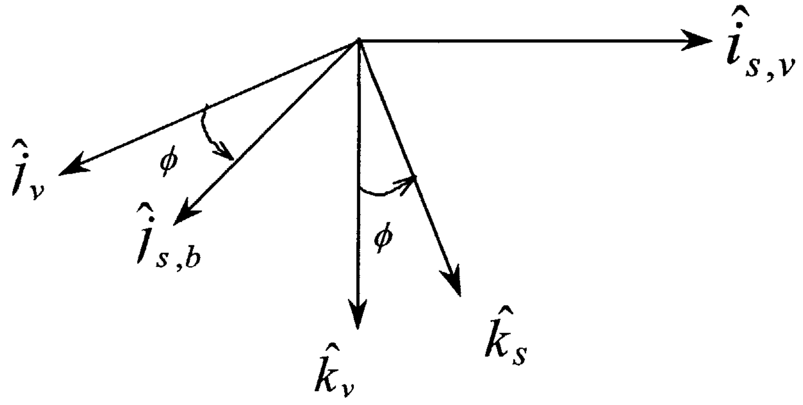


Figure 2.1: Bank Angle Rotation

$$T_{v2s} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{2.8}$$

Angle of Attack Rotation: The body reference frame is the stability reference frame rotated by the angle of attack α about their common $\hat{j}$ axes. This relationship is shown in Figure 2.2 below and the transformation matrix between the frames is given in Eq. 2.9.

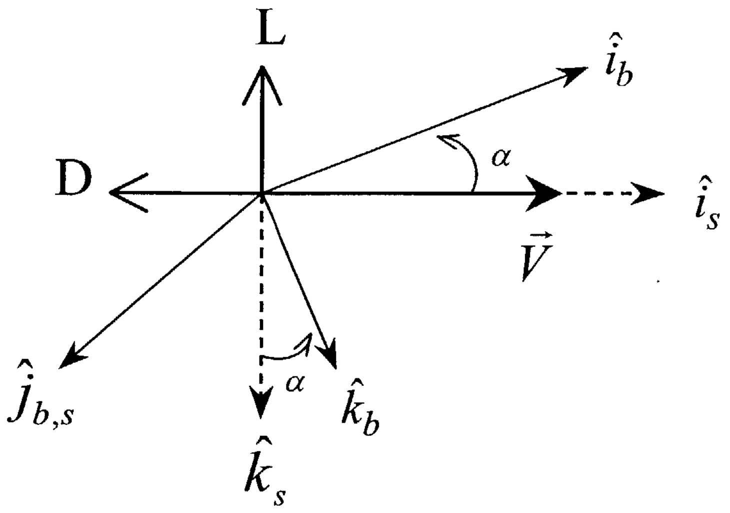


Figure 2.2: Angle of Attack Rotation

$$T_{s2b} = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \tag{2.9}$$

Aerodynamic accelerations act on the body in the stability frame. However, propagation of the states is accomplished in the inertial frame. Therefore, a single transformation matrix to rotate between the stability and inertial reference frames is desirable. This transformation matrix can be expressed as a product of the above matrices and is given in Eq. 2.10.

$$T_{s2i} = T_{v2i} T_{s2v} \tag{2.10}$$

Similarly, the body to inertial transformation can be expressed as:

$$T_{b2i} = T_{v2i} T_{s2v} T_{b2s} \tag{2.11}$$

2.4 Environment Models

Environment models describe the surroundings to which the vehicle is exposed. In this thesis, there are two basic models that must be considered: atmosphere and gravity.

2.4.1 Earth Atmosphere Models

Atmosphere models are used to calculate the density of the atmosphere at a given height above the surface. This density is important for lift and drag calculations. This thesis uses the US Standard Atmosphere, 1976 for the dynamic simulation [10].

2.4.2 Earth Gravity Model

Gravity models are used to calculate the acceleration due to gravity of the vehicle at a given position with respect to the planet. Earth-based gravity models can be simple in nature and increase in complexity depending on the desired accuracy. For this thesis, normal conical acceleration and J2 effects were considered:

$$\begin{aligned} a_g = \frac{\mu_\oplus}{r^3}\vec{r}\Big|_I + -\frac{3J_2\mu R_\oplus^2 r_I}{2r^5}\left(1-\frac{5r_K^2}{r^2}\right)\hat{i}_i + \\ -\frac{3J_2\mu R_\oplus^2 r_J}{2r^5}\left(1-\frac{5r_K^2}{r^2}\right) + -\frac{3J_2\mu R_\oplus^2 r_K}{2r^5}\left(3-\frac{5r_K^2}{r^2}\right) \end{aligned} \tag{2.12}$$

Eq. 2.12 is derived in reference [9].

2.5 Equations of Motion

The full equations of motion for an aerocapture vehicle are derived using Newton's Laws. Several assumptions are used to simplify the derivation. These assumptions are as follows:

- The vehicle is symmetrical.
- The vehicle has constant mass (i.e. fuel burn is not accounted for).
- The vehicle produces no thrust.
- There are no aerodynamic moments (vehicle is statically trimmed).
- There are no side forces (no sideslip, β).

With these assumptions, acceleration acting on the vehicle is comprised of two parts: acceleration due to gravity and acceleration due to aerodynamic forces:

$$\vec{a} = \vec{a}_g + \vec{a}_a \tag{2.13}$$

Aerodynamic acceleration is further broken down into two components: acceleration due to drag and acceleration due to lift.

$$\vec{a}_a = \vec{a}_d + \vec{a}_l \tag{2.14}$$

Since this simulation was developed without a solid vehicle description, general parameters such as ballistic coefficient (C_B) and lift-to-drag ratio (L/D)were sufficient to calculate the aerodynamic accelerations. The force due to drag (F_D) acting on a vehicle is defined using the vehicle coefficient of drag (C_d), dynamic pressure (q), and planform (or reference) area (S).

$$F_D = C_D q S \tag{2.15}$$

$$q = \frac{1}{2}\rho v_{rel}^2 \tag{2.16}$$

Eq. 2.17 defines ballistic coefficient:

$$C_B = \frac{m}{C_d S} \tag{2.17}$$

Using these definitions and Newton's Law of gravitation with constant mass,

$$F_D = m a_D \tag{2.18}$$

acceleration due to drag (a_D) can be derived as:

$$a_D = \frac{\rho v_{rel}^2}{2C_B} \tag{2.19}$$

Then acceleration due to lift (a_L) is found simply:

$$a_L = a_D \frac{L}{D} \tag{2.20}$$

Drag acceleration acts in the direction opposite the relative velocity while lift acceleration acts perpendicular to relative velocity. Written in the stability frame, the resulting aerodynamic acceleration is:

$$\vec{a}_a = -\frac{\rho V_{rel}^2}{2C_B}\hat{i}_s + -\frac{\rho V_{rel}^2}{2C_B}\frac{L}{D}\hat{k}_s \tag{2.21}$$

For integration, acceleration must be rotated into the inertial frame.

$$\vec{a}_a = T_{v2i} T_{s2v}\left(-\frac{\rho V_{rel}^2}{2C_B}\hat{i}_s + -\frac{\rho V_{rel}^2}{2C_B}\frac{L}{D}\hat{k}_s\right) \tag{2.22}$$

Yielding:

$$\begin{aligned}
a_{a_i} &= -a_D i_{v_1} + a_L \sin(\phi) i_{v_2} - a_L \cos(\phi) i_{v_3} \\
a_{a_j} &= -a_D j_{v_1} + a_L \sin(\phi) j_{v_2} - a_L \cos(\phi) j_{v_3} \\
a_{a_k} &= -a_D k_{v_1} + a_L \sin(\phi) k_{v_2} - a_L \cos(\phi) k_{v_3}
\end{aligned} \tag{2.23}$$

Acceleration due to gravity can also be expressed as a sum of components: conical acceleration and acceleration due to a nonspherical body.

$$\vec{a}_g = \vec{a}_c + \vec{a}_{ns} \tag{2.24}$$

Conical acceleration is expressed as:

$$\vec{a}_c = \frac{\mu}{r^3}\vec{r} = \frac{\mu}{r^3}\left(r_I\hat{i}_i + r_J\hat{j}_i + r_K\hat{k}_i\right) \tag{2.25}$$

The nonspherical terms can be added to the conical equation to account for the Earth's oblateness, non-uniform density, etc. For the purposes of this thesis the J2 terms are sufficient and the resulting total gravitational acceleration is the expression from Eq. 2.12 above. Note that this acceleration is already written in the inertial frame and no further rotation is required.

Finally, the total acceleration in the inertial frame:

$$\vec{a}^I = \begin{bmatrix} -a_D i_{v_1} + a_L\sin(\phi) i_{v_2} - a_L\cos(\phi) i_{v_3} - \frac{\mu_\oplus}{r^3} r_I - \frac{3J_2\mu R_\oplus^2 r_I}{2r^5}\left(1 - \frac{5r_K^2}{r^2}\right) \\ -a_D j_{v_1} + a_L\sin(\phi) j_{v_2} - a_L\cos(\phi) j_{v_3} - \frac{\mu_\oplus}{r^3} r_J - \frac{3J_2\mu R_\oplus^2 r_J}{2r^5}\left(1 - \frac{5r_K^2}{r^2}\right) \\ -a_D k_{v_1} + a_L\sin(\phi) k_{v_2} - a_L\cos(\phi) k_{v_3} - \frac{\mu_\oplus}{r^3} r_K - \frac{3J_2\mu R_\oplus^2 r_K}{2r^5}\left(3 - \frac{5r_K^2}{r^2}\right) \end{bmatrix} \tag{2.26}$$

Integrating acceleration once yields the change in *velocity* in the inertial frame. Integrating twice yields the change in *position* in the inertial frame. Appendix B contains a detailed verification that these equations of motion were properly implemented into the system dynamics.

2.6 Vehicle Models

The vehicle used in this study was described solely by mass, planform area, lift-to-drag ratio, and coefficient of drag. The values used in this thesis are representative of a conceivable aerocapture vehicle.

2.7 Vehicle Properties

The following values were used for mass, planform area, and drag coefficient:

$$m = 15.4783\ slugs \quad (225.89\ kg)$$

$$S = 12.163\ ft^2 \quad (1.13\ m^2)$$

$$C_d = 1.4286$$

These values yield a ballistic coefficient, $C_B = 0.891157\ slugs/ft^2\ (139.93\ kg/m^2)$.

The algorithm was highly sensitive to the lift-to-drag ratio. As a result, for testing, lift-to-drag ratio was varied between 0.25 and 1.5 in order to gain a better understanding of the range of possible trajectories.

2.7.1 Aerodynamic Heating

Heating rate and heating load are important factors in aerocapture vehicle performance. The convective heating rate is calculated empirically from an equation in reference [5]:

$$\dot{Q} = \frac{17600}{\sqrt{R_n}} \sqrt{\frac{\rho}{\rho_{SL}}} \left(\frac{v_{rel}}{v_e} \right)^{3.15} \tag{2.27}$$

where R_n is the vehicle nose radius, ρ_{SL} is the density at the surface of the earth, and v_e is the reference spherical velocity given by:

$$v_e = \sqrt{\frac{\mu_\oplus}{R_\oplus}} \tag{2.28}$$

The following values were used:

$$R_n = 1.96764\ ft\ \ (0.60\ m)$$

$$v_e = 25936.241\ ft/s\ \ (7.91\ km/s)$$

$$\rho_{SL} = 0.00237688\ slugs/ft^3\ \ (1.22\ kg/m^3)$$

Nose radius was calculated by assuming that the planform area of the vehicle is a circle and the radius of that circle is the nose radius.

The heat load due to aerodynamic heating is found by integrating the heating rate over time:

$$Q(t) = \int_0^t \dot{Q}(\tau) d\tau \tag{2.29}$$

Chapter 3
Guidance Design

3.1 Overview

The starting point for this thesis was a targeting algorithm, known as PredGuid, based on Higgins' numeric predictor-corrector [1]. The stated objective of this thesis was to enhance this existing targeting algorithm by implementing an energy management phase prior to targeting, developing a generic method of transitioning between the energy management and targeting phases, and replacing other heuristic features with more generic features. A detailed description of PredGuid, with all the mission-specific heuristics, is contained in Appendix A. This chapter describes the enhanced guidance algorithm and phase change logic in entirety and specifically details the enhancements to Higgins' targeting algorithm.

The enhanced algorithm *flow* (depicted in Figure 3.1) is identical to PredGuid. However, in the enhanced algorithm, Bank Angle Determination, which was originally only a targeting phase, is now further broken down into two phases, Energy Management and Targeting, discussed in sections 3.2 and 3.3, respectively. Lateral Control, discussed in section 3.4, and Command Incorporation, discussed in section 3.5 retain their PredGuid objectives, but have been modified appropriately to remove heuristics and support the addition of the Energy Management Phase. Initialization and Aerodynamic Properties were not substantially modified from PredGuid. Changes were limited to cosmetics and substitution of the US Standard Atmosphere, 1976 in place of the original exponential atmospheric model. Initialization is executed once on the first guidance cycle. Aerodynamic Properties calculates estimated density, lift-to-drag ratio, and drag coefficient. These values are updated throughout the sensible atmosphere, and for a short time before guidance begins cycling, to ensure that guidance is receiving good data. Appendix A contains a detailed description of these procedures.

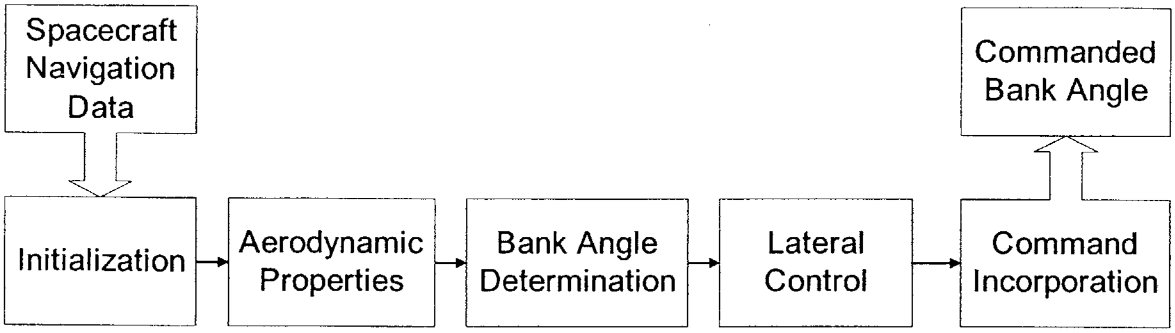


Figure 3.1: Guidance Algorithm Flow

During aerocapture, the vehicle steers solely by rotating the lift vector about the relative velocity vector. Guidance assumes the vehicle makes coordinated turns and trims to a constant angle of attack. Therefore, bank angle (ϕ) is the only control factor. Positive bank is defined as bank to the right according to the right hand rule. Bank angle can vary between $\pm 180°$ where $0°$ is full lift up (away from Earth) and $180°$ is full lift down (towards Earth). However, bank control is typically limited between $15°$ and $165°$ (in either direction) by the guidance algorithm in order to constantly maintain some control authority for out-of-plane corrections. The algorithm assumes the in-plane and out-of-plane velocity controls are decoupled. Therefore, the bank angle required to reach the target apogee is computed independently of the bank angle direction to remain in the desired orbital plane.

Guidance is initialized at entry interface (EI) by running the energy management guidance cycle once. Guidance does not cycle again until the vehicle is in the sensible atmosphere in order to prevent control corrections when there is insufficient atmospheric density to have reasonable command authority. Entry into the sensible atmosphere is determined by aerodynamic acceleration, calculated in Eq. 3.1:

$$a_{aero} = \frac{|a_{aero}|}{g_{\oplus}} \tag{3.1}$$

While the vehicle is experiencing loads above the acceleration minimum (0.075 g's), guidance cycles at a frequency of 0.5 Hz.

3.2 Energy Management Phase

The purpose of the energy management phase is to deplete sufficient energy to allow targeting of the exit conditions in the next phase. Several approaches have been explored including maintaining constant altitude, specified dynamic pressure, or reference drag.

All of these approaches adhered to a similar format, containing an altitude rate damper and a reference following term. The algorithm used in this thesis attempts to null out the altitude rate while maintaining a reference drag. This method was chosen for ease of implementation. The reference drag term can easily be replaced with a reference altitude term or dynamic pressure term and achieve similar results. While maintaining the constant cruise condition, the guidance runs periodic checks to determine when to initiate the targeting phase. The two major components of the Energy Management Phase are discussed in the following sections. The Energy Management Phase is depicted in Figure 3.2.

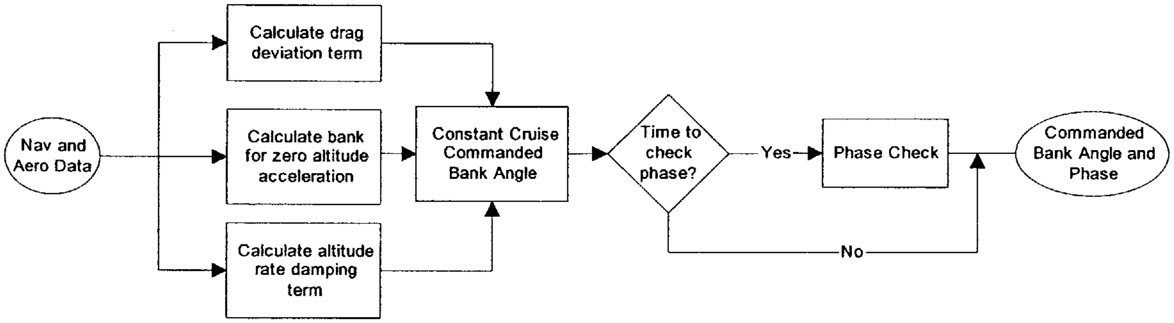

Nav and Aero Data
Calculate drag deviation term
Calculate bank for zero altitude acceleration
Calculate altitude rate damping term
Constant Cruise Commanded Bank Angle
Time to check phase?
Yes
No
Phase Check
Commanded Bank Angle and Phase

Figure 3.2: Energy Management Phase

3.2.1 Constant Condition Cruise

The constant condition cruise phase modeled in this thesis draws upon a conglomeration of the algorithms presented in references [2], [1] and [6]. The vehicle response can be modeled as a second-order spring/mass/damper system. Beginning with Newton's Second Law, the acceleration of the vehicle in the inertial reference frame can be written:

$$\frac{F}{m} = \left.\frac{d^2\vec{r}}{dt^2}\right|_I \tag{3.2}$$

Aerodynamic forces are most often expressed in local coordinate systems. Because the local reference frame is rotating, we must include the coriolis terms in the derivatives. The first derivative of position in the inertial frame is then:

$$\left.\frac{d\vec{r}}{dt}\right|_I = \left.\frac{d\vec{r}}{dt}\right|_h + {}^I\vec{\omega}^h \times \vec{r} \tag{3.3}$$

where h denotes the local horizontal reference frame from Chapter 2. Taking the second derivative,

$$\left.\frac{d^2\vec{r}}{dt^2}\right|_I = \left.\frac{d^2\vec{r}}{dt^2}\right|_h + \left.\frac{d\,{}^I\vec{\omega}^h}{dt}\right|_h \times \vec{r} + 2\,{}^I\vec{\omega}^h \times \left.\frac{d\vec{r}}{dt}\right|_h + {}^I\vec{\omega}^h \times \left({}^I\vec{\omega}^h \times \vec{r}\right) \tag{3.4}$$

Position and relative velocity in the inertial frame are defined in the local horizontal frame in Eq. 3.5 And 3.6, respectively:

$$\vec{r}^I = -r^h\hat{k}_h \tag{3.5}$$

$$\vec{v}^I_{rel} = v_{rel}\cos(\gamma)\hat{i}_h - v_{rel}\sin(\gamma)\hat{k}_h \tag{3.6}$$

where γ is the flight path angle. Angular velocity of the local horizontal frame with respect to the inertial frame is defined in Eq. 3.7.

$${}^I\vec{\omega}^h = -\frac{|\vec{v}_{rel}|}{|\vec{r}|}\cos(\gamma)\hat{j}_h \tag{3.7}$$

For small γ,

$$\vec{v}^h \approx v^h\hat{i}_h \tag{3.8}$$

$${}^I\vec{\omega}^h \approx -\frac{|\vec{v}_{rel}|}{|\vec{r}|}\hat{j}_h \tag{3.9}$$

For the energy management phase, we are only concerned with forces and acceleration in the $\hat{k}_h$ direction. Substituting the simplifications above into Eq. 3.2 and Eq. 3.4, we find that, in the $\hat{k}_h$ direction,

$$\frac{F}{m} = -\ddot{r} + \frac{v_{rel}^2}{r} \tag{3.10}$$

Lift and gravity are the only forces in the $\hat{k}_h$ direction, therefore,

$$\frac{F}{m} = \frac{-L\cos(\phi)}{m} + g = -\ddot{r} + \frac{v_{rel}^2}{r} \tag{3.11}$$

Using the definitions of drag and ballistic coefficient,

$$D = C_D qS \tag{3.12}$$

$$C_B = \frac{m}{C_D S} \tag{3.13}$$

acceleration in the $\hat{k}_h$ is presented in Eq. 3.14.

$$\ddot{r} = \frac{L}{D}\frac{q}{C_B}\cos\phi - g + \frac{v^2}{r} \tag{3.14}$$

Setting this acceleration equal to zero, the cosine of bank required to achieve zero altitude rate is:

$$\cos\phi\Big|_{\dot{r}=0} = \frac{C_B}{q}\frac{1}{L/D}\left[g - \frac{v^2}{r}\right] \tag{3.15}$$

The commanded bank angle from guidance is calculated by taking the value of bank required for zero altitude rate and adding an altitude rate damper and reference drag following terms:

$$\cos\phi_C = \cos\phi\Big|_{\dot{r}=0} - K_{\dot{r}}\frac{\dot{r}}{\bar{q}} + K_D\frac{\left(D - D_{ref}\right)}{\bar{q}} \tag{3.16}$$

Substituting Eq. 3.14 into Eq. 3.16 for the cosine term, the second-order equation becomes:

$$\ddot{r} + \frac{1}{C_B}\frac{L}{D}K_{\dot{r}}\dot{r} + \frac{1}{C_B}\frac{L}{D}K_D\left(D - D_{ref}\right) = 0 \tag{3.17}$$

The following equation for reference drag (with a gain value of $K = 1.5$) was taken from reference [2].

$$D_{ref} = K\left(\frac{v^2}{r} - g\right)\frac{C_D}{C_L} \tag{3.18}$$

The general equation of a second order spring/mass/damper system is:

$$\ddot{x} + 2\omega_n\xi\dot{x} + \omega_n^2 = 0 \tag{3.19}$$

Applying this general equation to Eq. 3.17, natural frequency and damping ratio are:

$$\omega_n^2 = \frac{1}{C_B}\frac{L}{D}K_D \tag{3.20}$$

$$2\omega_n\xi = \frac{1}{C_B}\frac{L}{D}K_{\dot{r}} \tag{3.21}$$

The following values were extracted from references [1] and [6] and worked well for this application:

$$\omega_n = 0.06\ rad/s$$

$$\xi = 1.5$$

Note that since the lift-to-drag ratios were varied in this study, the resulting gain values are functions of L/D.

$$K_D = f(L/D)$$
$$K_r = f(L/D)$$

3.2.2 Phase Change Conditions

The vehicle remains in the energy management phase until sufficient energy has been depleted to allow targeting. This condition is quantified by determining if, at the given point in the trajectory, the vehicle could fly a constant bank angle of 110° for the remainder of the trajectory and have the resulting apogee be below or within a given tolerance above the target apogee. The bank angle of 110° was not chosen arbitrarily. Extensive research completed in past studies indicates that flying a slightly lift down bank angle when exiting the atmosphere raises the resulting periapse altitude [1]. Reducing the amount of ΔV required to raise the periapse altitude after aerocapture greatly reduces the fuel budget.

One more condition is required in order to ensure guidance does not change phases prematurely. Hyperbolic orbits have negative values of semi-major axis. If the predicted final position and velocity are still on a hyperbolic orbit, the radius of apogee will be calculated to be a negative number. Obviously, a hyperbolic trajectory is not a low trajectory; however, a negative apogee will be interpreted by targeting as low. If guidance exits energy management too early based on the 'low' predicted apogee, the orbit will remain hyperbolic and the targeting algorithm will be unable to resolve this condition to reach the target apogee. To avoid premature exit, guidance checks the energy of the predicted exit conditions. Energy will be positive on a hyperbolic orbit, zero on a parabolic orbit, and negative on an elliptical orbit. The algorithm does not change phases until the energy of the predicted exit conditions is negative. This ensures that the initial phase check does not give a false indication of requiring a phase change.

$$\varepsilon = \frac{1}{2}v^2 - \frac{\mu}{r} = -\frac{\mu}{2a} \tag{3.22}$$

The phase change check executes at a slower frequency than the main guidance, running at a frequency of 0.1667 Hz. Figure 3.3 depicts the functional flow of the phase change logic.

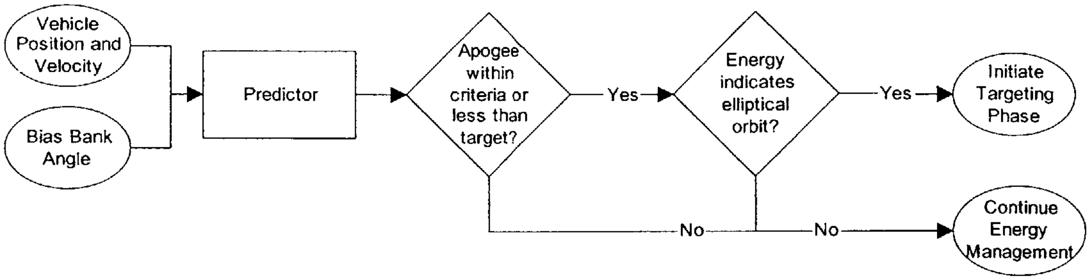

Vehicle Position and Velocity
Bias Bank Angle
Predictor
Apogee within criteria or less than target?
Yes
Energy indicates elliptical orbit?
Yes
Initiate Targeting Phase
No
No
Continue Energy Management

Figure 3.3: Phase Change Functional Flow

3.3 Targeting Phase

The targeting portion of the guidance algorithm relies on a numeric predictor-corrector method and uses a constant bank angle assumption to target the exit conditions. Prior to the addition of the energy management phase, PredGuid used only the targeting algorithm throughout the trajectory. Several heuristic approximations were incorporated into PredGuid in order to optimize it for a particular vehicle and mission. These heuristics and the constant bank angle assumption imposed severe limitations on the usability of the algorithm. Numerous improvements and changes were made over the course of this study while still preserving the basic intent of the targeting algorithm. The description contained in this chapter describes the current targeting algorithm while highlighting major changes and improvements to the original algorithm.

Figure 3.4 depicts the functional flow of the targeting algorithm. The following brief outline summarizes the basic algorithm:

1. Set the bank angle to the current commanded bank angle (ϕ_{CMD})
2. Predict the final apogee using current states and a constant bank angle assumption
3. Update the bank angle by an amount $\Delta\phi$ to null the target miss
4. Repeat steps 2 and 3 until guidance converges to the bank angle which will hit the target conditions within tolerance, or until the correction limit is reached

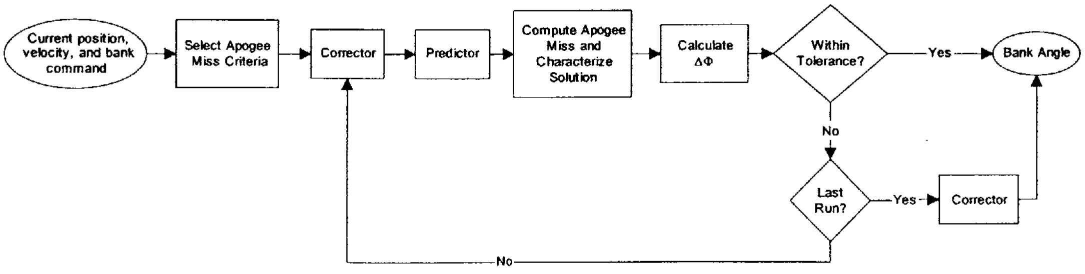

Current position, velocity, and bank command
Select Apogee Miss Criteria
Corrector
Predictor
Compute Apogee Miss and Characterize Solution
Calculate ΔΦ
Within Tolerance?
Yes
Bank Angle
No
Last Run?
Yes
Corrector
No

Figure 3.4: Targeting Functional Flow

3.3.1 Predictor-Corrector Method

The predictor-corrector is the main portion of the targeting algorithm. In the predictor, the current states are propagated forward to determine the resulting apogee based on a constant bank angle assumption. The corrector then updates the bank angle guess in order to null the target miss. The cycle continues until a bank angle is found which meets the apogee miss tolerance or until a maximum number of corrections is reached. The number of corrections is limited to improve running time while still allowing guidance to converge to a solution.

3.3.1.1 Predictor

The numeric prediction algorithm uses a Fourth Order Runge-Kutta method to accomplish the integration. The vector equations to be integrated are:

$$\frac{d\vec{r}}{dt} = \vec{v} \tag{3.23}$$

$$\frac{d\vec{v}}{dt} = \vec{a} \tag{3.24}$$

The initial conditions are the current position and velocity vectors. Acceleration is defined as the vector sum of the gravitational and aerodynamic accelerations:

$$\vec{a} = \vec{a}_g + \vec{a}_a \tag{3.25}$$

The Earth gravity model used is identical to the model used in the environment and is described in Chapter 2 with the final acceleration due to gravity described in Eq. 2.12 and presented below as Eq. 3.26.

$$\begin{aligned} a_g = \frac{\mu_\oplus}{r^3}\vec{r}\Big|_I + -\frac{3J_2\mu R_\oplus^2 r_I}{2r^5}\left(1-\frac{5r_K^2}{r^2}\right)\hat{i}_i + \\ -\frac{3J_2\mu R_\oplus^2 r_J}{2r^5}\left(1-\frac{5r_K^2}{r^2}\right) + -\frac{3J_2\mu R_\oplus^2 r_K}{2r^5}\left(3-\frac{5r_K^2}{r^2}\right) \end{aligned} \tag{3.26}$$

Aerodynamic acceleration is further decomposed into acceleration due to drag and acceleration due to lift. Drag acts in the opposite direction of the relative velocity and lift acts perpendicular to the relative velocity.

$$\vec{a}_a = -\vec{a}_d\hat{i}_s - \vec{a}_l\hat{k}_s \tag{3.27}$$

where

$$a_D = \frac{\rho v_{rel}^2}{2C_B} \tag{3.28}$$

$$a_L = a_D \frac{L}{D} \tag{3.29}$$

$$\hat{i}_s = unit(\vec{v}_{rel}) \tag{3.30}$$

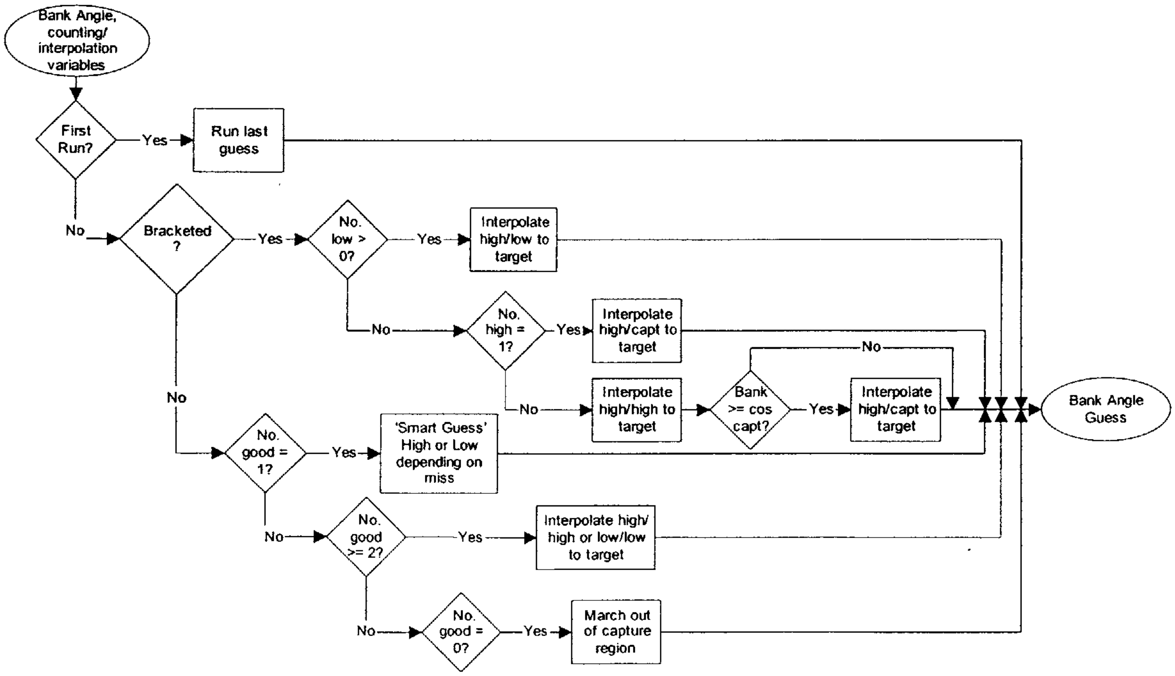
$$\hat{j}_s = unit(\hat{i}_s \times \vec{r}) \tag{3.31}$$

$$\hat{k}_s = (\hat{j}_s \times \hat{i}_s)\cos(\phi_p) + \hat{j}_s \sin(\phi_p) \tag{3.32}$$

ϕ_p is the bank angle used in the prediction. The prediction program propagates forward in time until the vehicle captures or escapes the atmosphere. Capture is indicated by the vehicle dipping below 200,000 feet $(60.96\ km)$in altitude while simultaneously having both negative altitude rate and acceleration.

3.3.1.2 Corrector

The correction algorithm uses various methods to compute the corrected bank angle based on the direction and severity of the target misses from previous guesses. These methods include interpolation, extrapolation, marching out of the capture region, and 'smart-guessing'. The corrector functional flow is one of the most complex in the entire program and is depicted in Figure 3.5.



Bank Angle, counting/ interpolation variables
First Run?
Yes
Run last guess
No
Bracketed ?
Yes
No. low > 0?
Yes
Interpolate high/low to target
No
No. high = 1?
Yes
Interpolate high/capt to target
No
No
Interpolate high/high to target
Bank >= cos capt?
Yes
Interpolate high/capt to target
No
No. good = 1?
Yes
'Smart Guess' High or Low depending on miss
No
No. good >= 2?
Yes
Interpolate high/ high or low/low to target
No
No. good = 0?
Yes
March out of capture region
Bank Angle Guess

Figure 3.5: Corrector Functional Flow

From this figure, the six distinct methods of generating a new bank angle guess are visible:

1. Use previous bank angle
2. Interpolate a high apogee bank angle and low apogee bank angle
3. Interpolate between a high apogee bank angle and a captured bank angle
4. Extrapolate from two high or two low apogee bank angles
5. March out of the capture region
6. 'Smart guess' based on apogee miss

Two significant changes were made to this portion of the algorithm. First, in the original targeting algorithm, Method 6 in the list above used a stored sensitivity of final apogee to bank angle $(dR_a/d\phi)$to determine the bank angle to try based on one 'good' guess. A 'good' guess is defined as a guess not resulting in a captured trajectory. The stored sensitivity was specific to a mission and vehicle. Applying it to different conditions was resulting in poor bank angle corrections. This method was replaced with a simple 'smart-guessing' algorithm which reduced or increased the bank angle by one degree if the predicted apogee was lower or higher than the target, respectively. While less computationally efficient than use of a stored sensitivity, this method is more generic and was better able to manage the wide range of entry conditions examined.

The second change resulted from the fact that the trajectories studied had a much higher sensitivity of final apogee to bank angle change than the original application for which the targeting algorithm was designed. With these high-energy trajectories, extremely small bank angle changes yield large changes in the predicted apogee. Guidance was unable to converge to a solution on each execution because the number of iterations was insufficient and the minimum allowable bank angle correction was too small. The number of iterations allowed per guidance cycle was increased and the criteria for hitting the target was loosened at high velocities. As the velocity decreases, the criteria for hitting the target narrows. Additionally, the lower limit imposed on the size of the bank angle correction was removed so that corrections could be as small as the sensitivity required.

3.4 Lateral Control

Bank maneuvers generate a component of lift in the out-of-plane direction causing out-of-plane position (θ_r) and out-of-plane velocity (θ_v) errors. Since bank angle is the only control variable, both errors cannot be controlled simultaneously. For this thesis, the desired orbital plane is assumed to be the same as the initial orbital plane. The perpendicular to the desired orbital plane is calculated using the initial position and velocity vectors at EI.

$$I_{yd} = \frac{\vec{v}_{init} \times \vec{r}_{init}}{\left|\vec{v}_{init} \times \vec{r}_{init}\right|} \tag{3.33}$$

Out of plane position and velocity are calculated by taking dot product of the position and velocity vectors, respectively, with the perpendicular.

$$\theta_v = \vec{v} \cdot I_{yd} \tag{3.34}$$

$$\theta_r = \vec{r} \cdot I_{yd} \tag{3.35}$$

This thesis elects to control out of plane velocity. If bank reversals are kept frequent and even, position error should be minimal. Previously, in PredGuid, a lateral corridor with maximum out of plane velocity limits was determined by the user. One change incorporated into the targeting algorithm was the removal of heuristic out of plane velocity limits. They were replaced with limits that are based on a percentage of the forward velocity. As a result, the lateral corridor still narrows with time, but narrows as a function of forward velocity rather than by heuristic limits which would only work for specific cases. Figure 3.6 demonstrates a nominal lateral corridor.

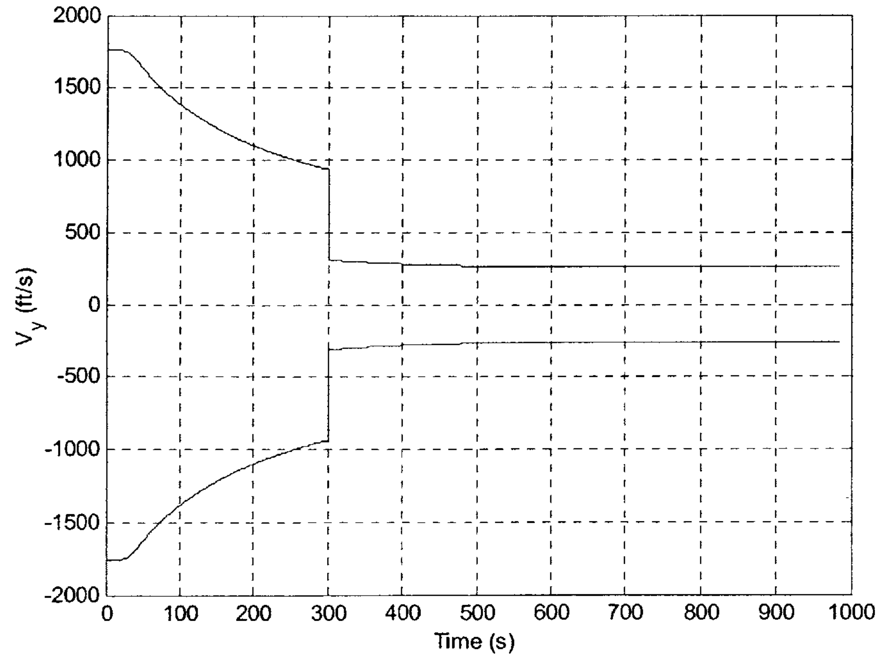


Figure 3.6: Lateral Corridor Description

3.5 Command Incorporation

The PredGuid was designed to bias to a neutral bank angle in order to save command margin for possible dispersions late in the trajectory. Without command biasing, dispersions could cause control saturation and subsequent target miss. An empirical velocity was chosen at which to begin biasing the commanded bank angle during targeting. The bias feature remains in the guidance algorithm, however the velocity restriction was removed. The algorithm biases during the *entire* targeting phase. Biasing is unnecessary during the energy management phase and actually inhibits the performance of that phase. In order to avoid sudden control corrections acceleration spikes at the phase change point, the bias angle was set to $110°$. By matching the phase change criteria, there should be a smooth transition between phases.

Chapter 4
Corridor and Test Case Determination

4.1 Overview

As discussed in Chapter 1, expansion of the capturable corridor was one of the main purposes of this thesis. The first step in measuring improvement is to determine current capability. This chapter describes how the theoretical and capturable corridors are determined and presents the corridors for the lift-to-drag ratios and velocities examined. Furthermore, this chapter describes initial conditions used in the dynamic simulation.

4.2 Earth Corridor Approximation

The theoretical corridor is defined as the range of flight path angles through which the vehicle is theoretically capable of accomplishing successful aerocapture where successful aerocapture is defined as reaching the target apogee with no other constraints. The shallow end of the corridor is the flight path angle at which the vehicle stays just below the target apogee by flying a full lift down bank angle throughout the entire atmosphere. The steep end of the corridor is the flight path angle at which the vehicle just reaches the target apogee by flying a full lift up bank angle throughout the atmosphere. The steep trajectory sees the greatest acceleration, heating rate, and dynamic pressure while having a shorter duration. The shallow trajectory has low acceleration, heating rate, and dynamic pressure, but is long in duration and yields higher heating loads. The theoretical corridor is dependent on lift-to-drag ratio. Figure 4.1 illustrates the theoretical corridor.

Realistically, a vehicle cannot fly the full theoretical corridor as defined above. Structural considerations limit the range of flight path angles which the vehicle can successfully fly. As already discussed, on the steep end of the corridor, the vehicle must fly a full lift up bank angle for the entire trajectory. If structural constraints limit the maximum deceleration, a more accurate description of the steep end of the corridor would be the steepest flight path angle for which the vehicle can fly full lift up and *remain below an established deceleration limit*. This thesis chose a maximum limit of 10 g's, typical of a robotic mission.

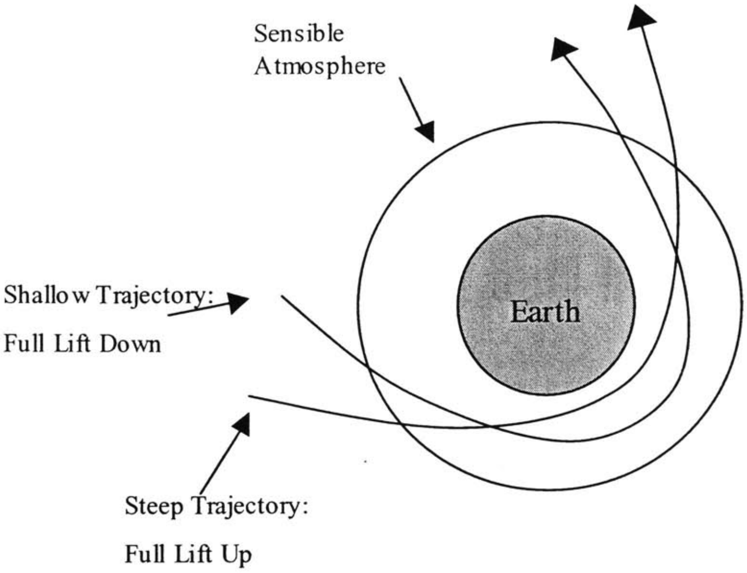


Figure 4.1: Theoretical Corridor

The deceleration is calculated using Eq. 3.1 and presented below as Eq. 4.1.

$$a_{aero} = \frac{|a_{aero}|}{g_{\oplus}} \tag{4.1}$$

Constraints may also be place on heating rate, heating load, and ΔV to raise perigee; however, this study did not examine these constraints. Flight path angles meeting all the constraints make up the true theoretical corridor. Figure 4.2 through Figure 4.7 summarize the theoretical corridors for the lift-to-drag ratios examined as a function of hyperbolic excess velocity (V_∞). These theoretical corridors include only the acceleration limit constraints. The width of the theoretical corridor increases with increased lift-to-drag ratio and decreasing V_∞.

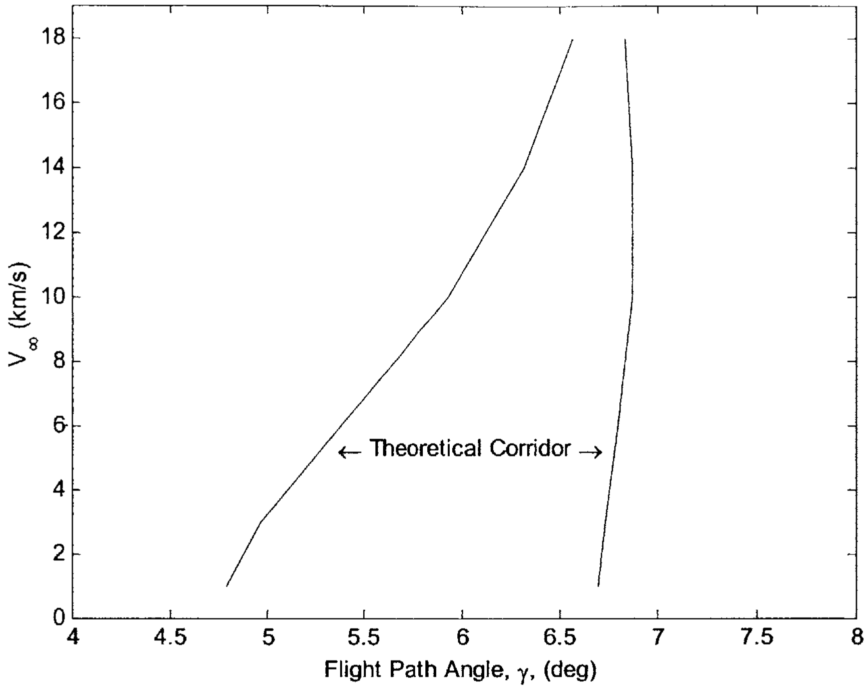


Figure 4.2: Theoretical Corridor, L/D = 0.25

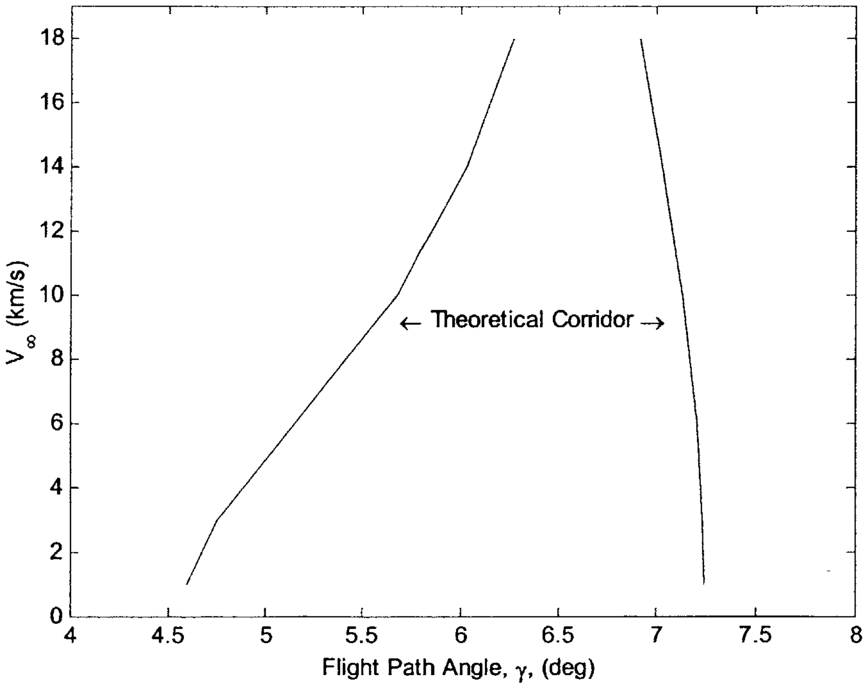


Figure 4.3: Theoretical Corridor, L/D = 0.50

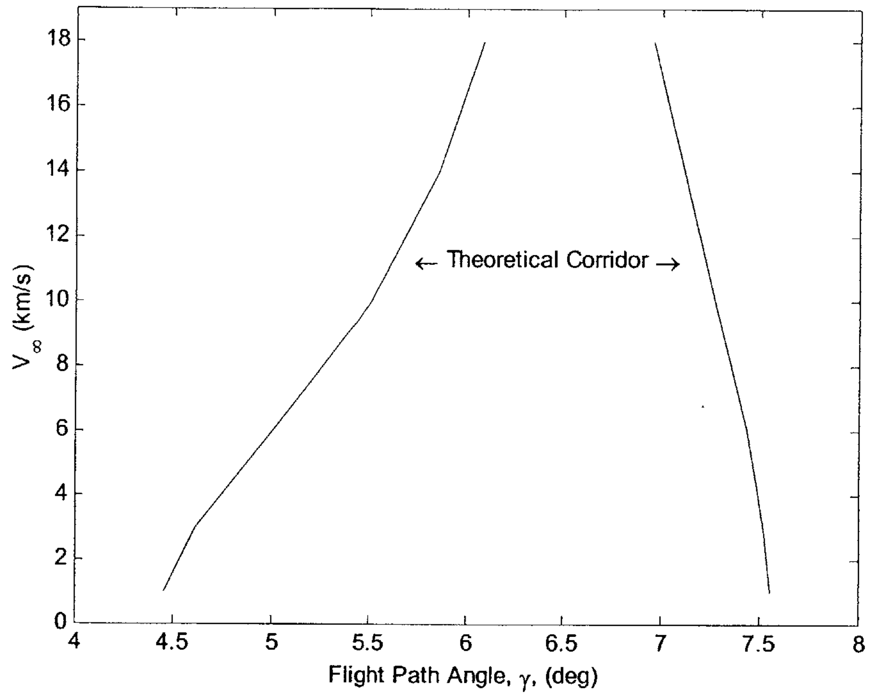


Figure 4.4: Theoretical Corridor, L/D = 0.75

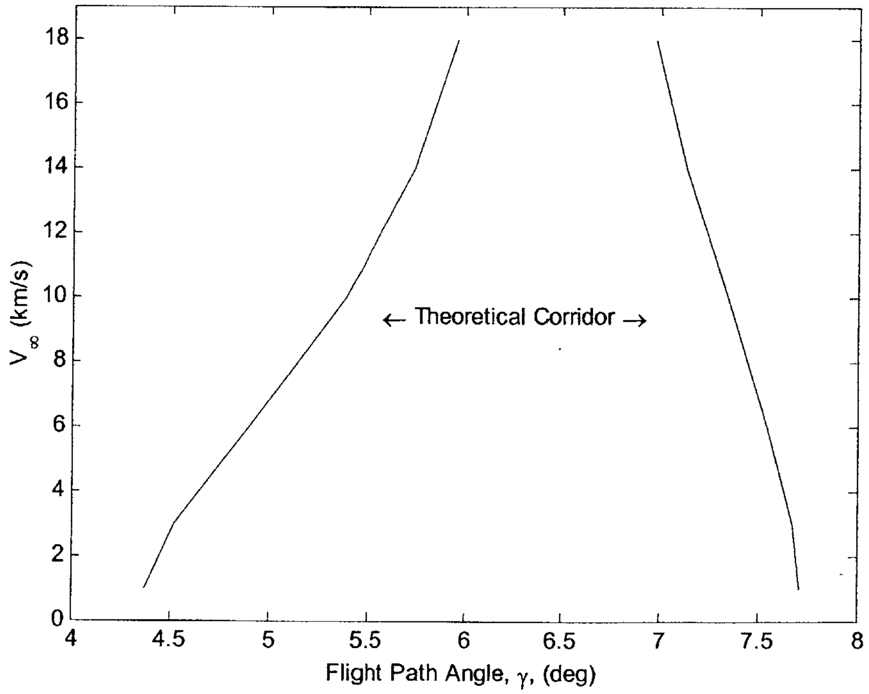


Figure 4.5: Theoretical Corridor, L/D = 1.00

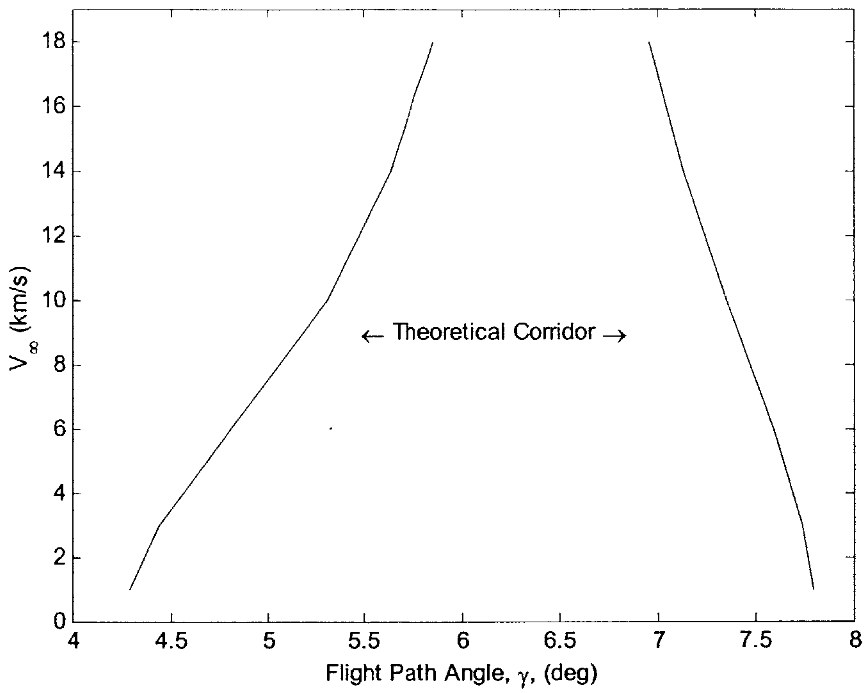


Figure 4.6: Theoretical Corridor, L/D = 1.25

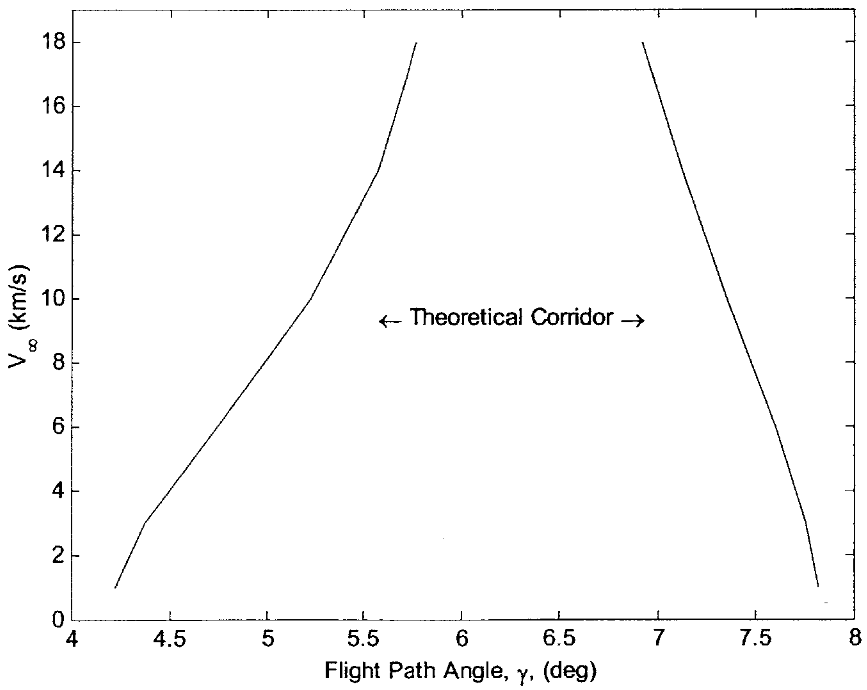


Figure 4.7: Theoretical Corridor, L/D = 1.50

Examination of the acceleration, dynamic pressure, and heating rate over time demonstrates the large difference between trajectories on the opposite ends of the corridor. Figure 4.8 through Figure 4.10 demonstrate this concept for a representative sample case with lift-to-drag ratio of 0.75 and $V_{\infty} = 6\ km/s$. For this case, the heating load on the shallow end of the corridor was 24,214 Btu/ft^2 (275,199 kJ/m^2) and the heating load on the steep end of the corridor was 18,062 Btu/ft^2 (204,984 kJ/m^2).

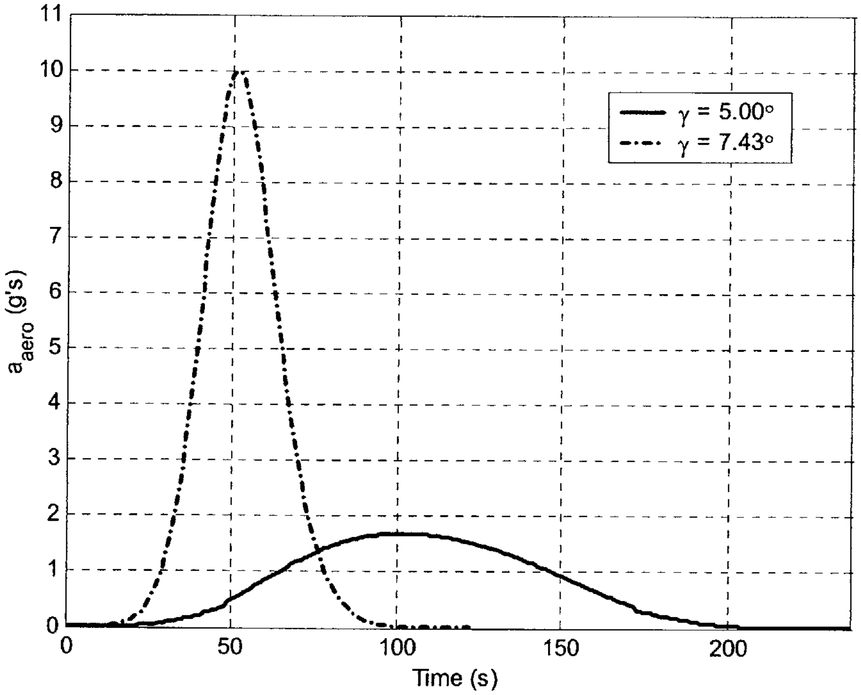


Figure 4.8: Acceleration vs. Time, Corridor Extremes

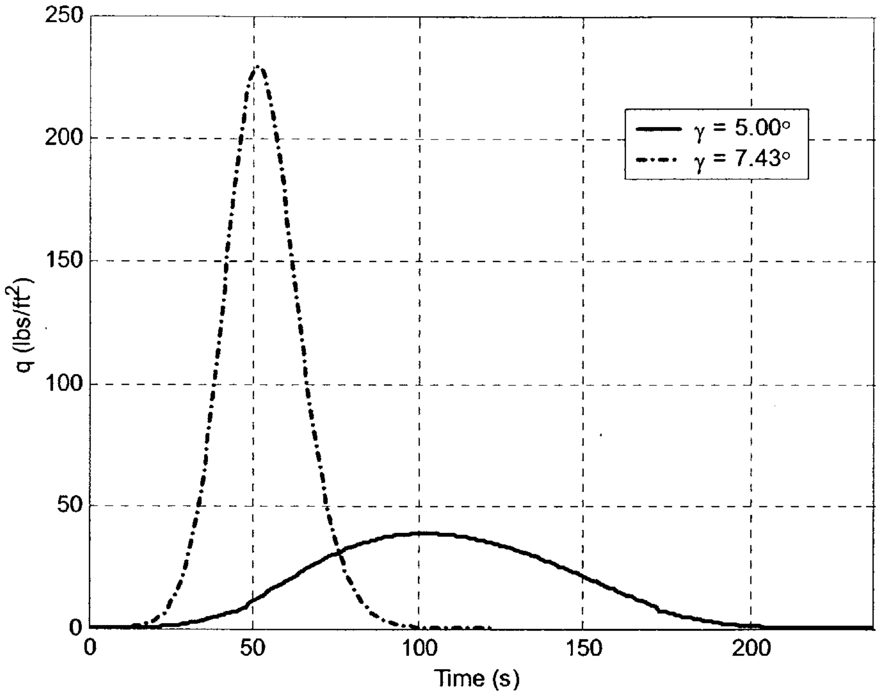


Figure 4.9: Dynamic Pressure vs. Time, Corridor Extremes

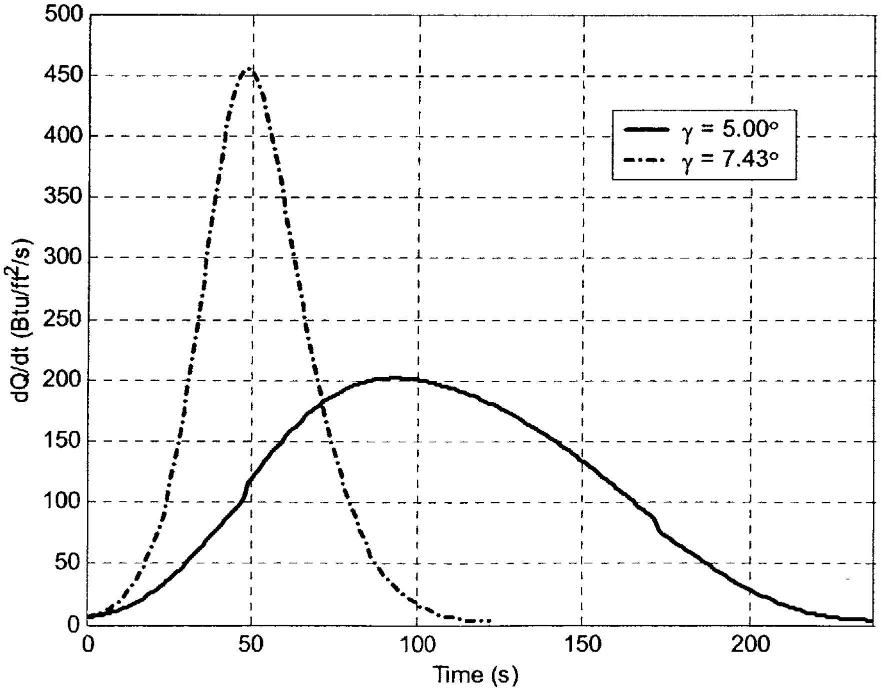


Figure 4.10: Heating Rate vs. Time, Corridor Extremes

4.3 Test Case Determination

Entry flight path angles were determined using basic astrodynamics. Six values are required to fully describe an orbit and the vehicle position on the orbit. The simulation was initialized with position and velocity vectors derived from these values. The six values chosen for this thesis were hyperbolic excess velocity (V_∞), altitude at EI (a_{EI}), inclination (i), vacuum periapse altitude (h_p), argument of perigee (ω), and longitude of the ascending node (Ω). Four of these values remained constant for all cases:

$$a_{EI} = 400{,}000\ ft \quad (121.92\ km)$$

$$\omega = 0^\circ$$

$$\Omega = 0^\circ$$

$$i = 5^\circ$$

Altitude at EI was chosen based on the top of the sensible atmosphere, standardized for Earth at 400,000 ft. Argument of perigee and longitude of the ascending node were chosen for simplicity. Inclination was chosen to ensure that argument of perigee and longitude of the ascending node would be defined.

Reference [3] compares hyperbolic excess velocities for other planets and moons and translates them to equivalent Earth demonstration velocities. Earth approach velocities which effectively simulate approaches to other planets and moons range from 9 to 14 km/s (29528 to 45931 ft/s). The velocities examined in this thesis were 1, 3, 6, 10, 14, and 18 km/s (3281, 9843, 19685, 32808, 45931, and 59055 ft/s) which completely encompasses the Earth demonstration velocity range from reference [3] and extends beyond it for further examination. At each velocity, vacuum periapse altitudes were chosen to yield the desired flight path angles. The initial conditions were independent of lift-to-drag ratio.

Chapter 5
Results

5.1 Overview

This chapter presents the results of the algorithm enhancements. The first section provides verification that the energy management guidance is working correctly. Capturable corridor results are presented in the next section. Finally, the third section discusses various metrics investigated in an effort to analyze the effectiveness of the enhanced algorithm.

5.2 Guidance Verification

As discussed in Chapter 3, the energy management phase seeks to maintain a constant cruise condition until sufficient energy has been depleted to allow successful targeting. Select cases were examined to ensure correct performance. For a nominal case, several observations would be expected. The vehicle would attempt to drive the altitude rate to zero by initially flying full lift up. After the altitude rate settled to zero, the bank angle would need to be lift down to keep the vehicle in the atmosphere. However, the vehicle will slowly push more lift up to generate enough lift to maintain altitude as velocity decreases. Acceleration would initially peak as the vehicle attempts to null the altitude rate. Finally, drag attempts to follow the reference drag curve. Figure 5.1 through Figure 5.5 depict a nominal case ($V_\infty = 14\,km/s$, $L/D = 0.75$, $\gamma = 5.6°$) demonstrating the energy management phase behaves in this manner prior to phase change. Note that due to the decoupling of vertical and lateral guidance and the assumption of instantaneous bank angle changes, bank angle results are presented as the absolute value of the commanded bank angle, $|\phi_c|$. This method allows easier observation of bank angle values and trends without losing vertical guidance information.

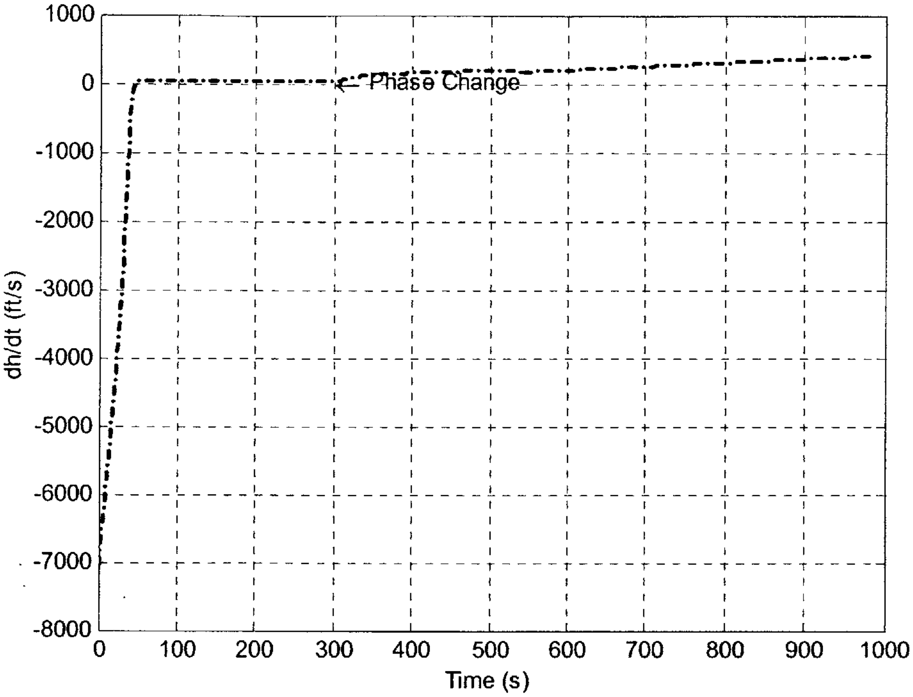


Figure 5.1: Altitude Rate vs. Time, $V_\infty = 14$ km/s, L/D = 0.75, $\gamma = 6.9°$

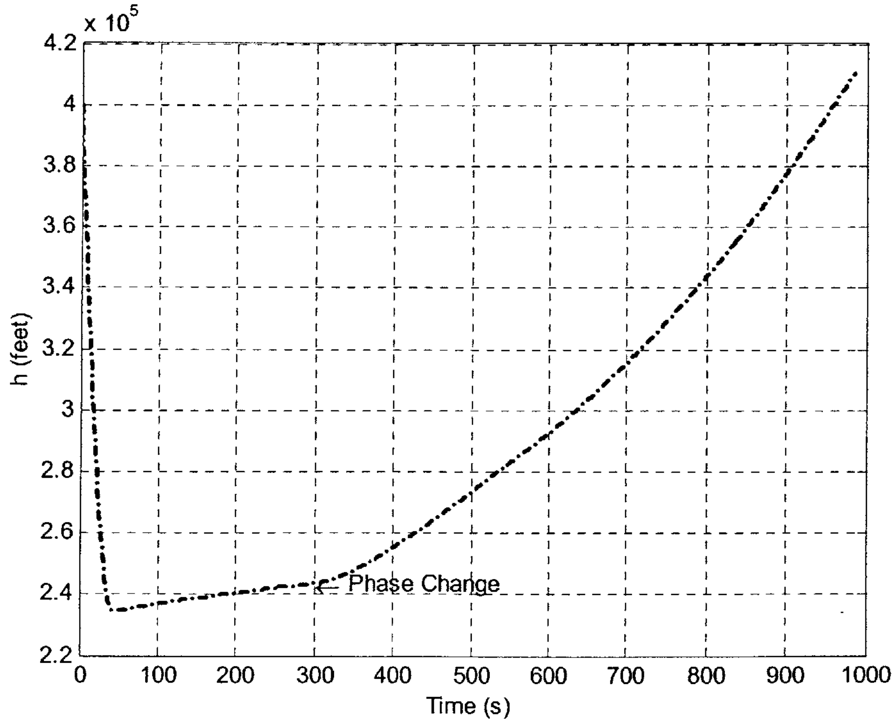


Figure 5.2: Altitude vs. Time, $V_\infty = 14$ km/s, L/D = 0.75, $\gamma = 6.9°$

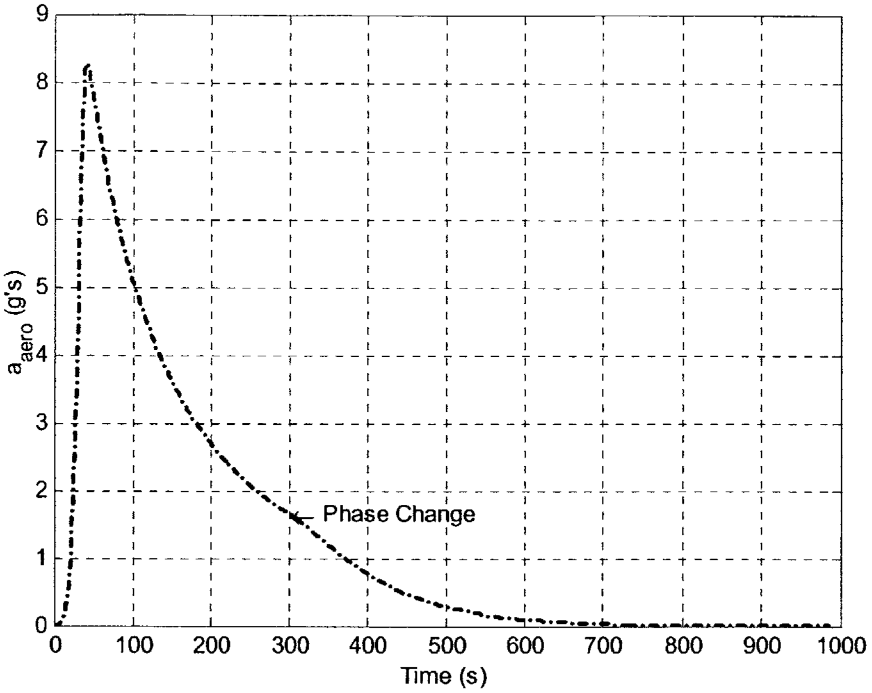


Figure 5.3: Acceleration vs. Time, $V_\infty = 14$ km/s, L/D = 0.75, $\gamma = 6.9°$

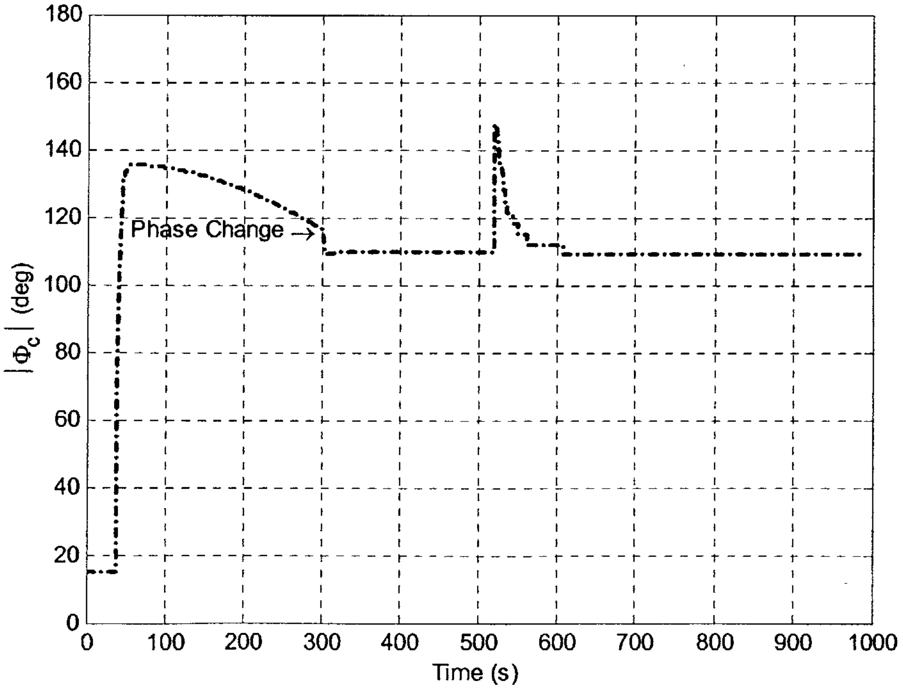


Figure 5.4: Commanded Bank Angle vs. Time, $V_\infty = 14$ km/s, L/D = 0.75, $\gamma = 6.9°$

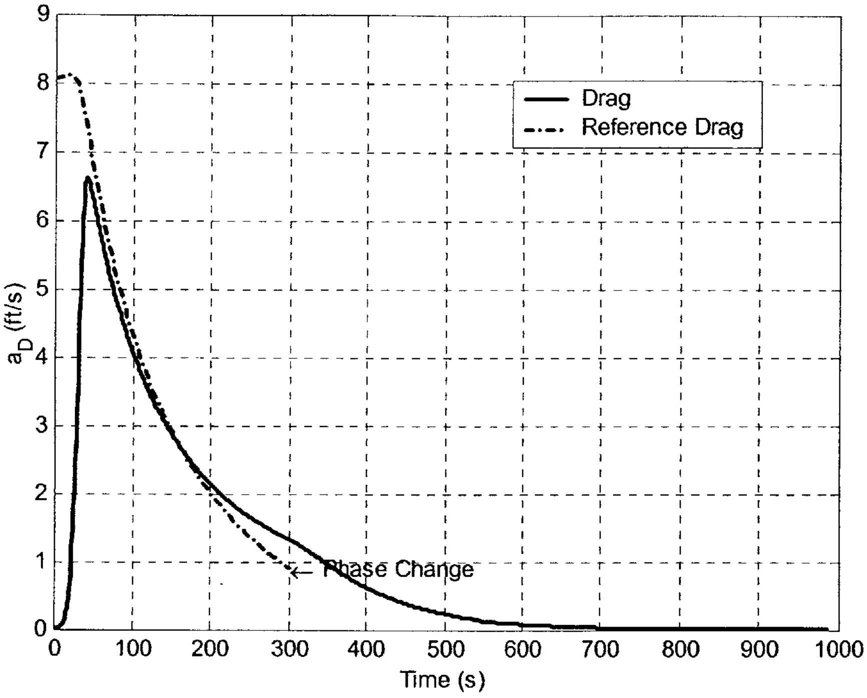


Figure 5.5: Drag Acceleration vs. Time, V_∞ = 14 km/s, L/D = 0.75, γ = 6.9°

Figure 5.6 through Figure 5.19 depict commanded bank angle histories for various velocities and lift-to-drag ratios. Each figure depicts a velocity and lift-to-drag ratio pair. The lift-to-drag ratios depicted are for the low (0.25), middle (0.75), and high (1.5) values in the range studied. All velocities for which guidance was able to capture a portion of the corridor are graphed for each lift-to-drag ratio. Each figure plots bank angle trajectories for the steepest and shallowest flight path angles captured as well as a flight path angle in the middle between these extremes.

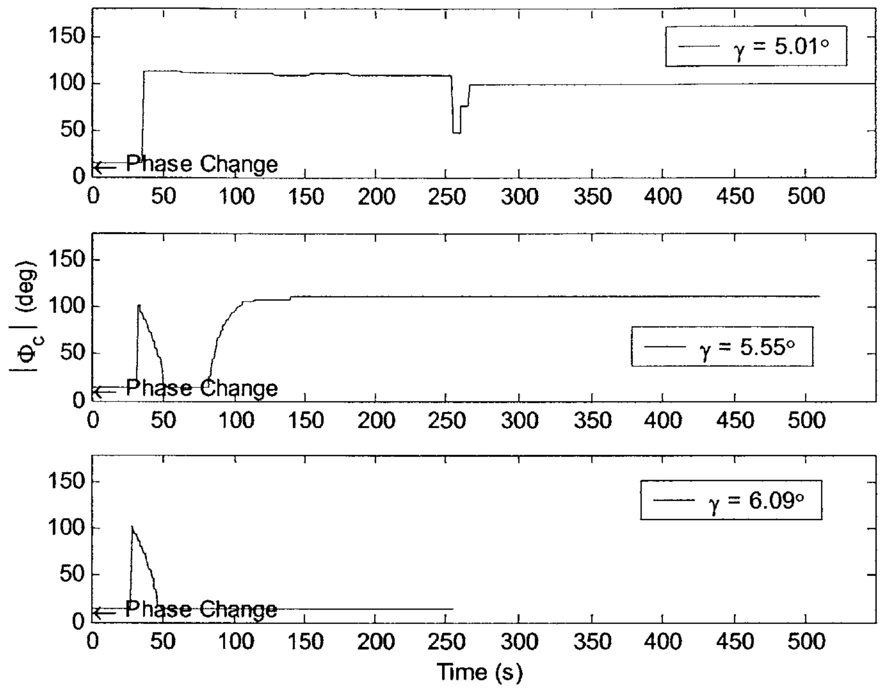


Figure 5.6: Commanded Bank Angle vs. Time, $V_\infty = 1$ km/s, L/D = 0.25

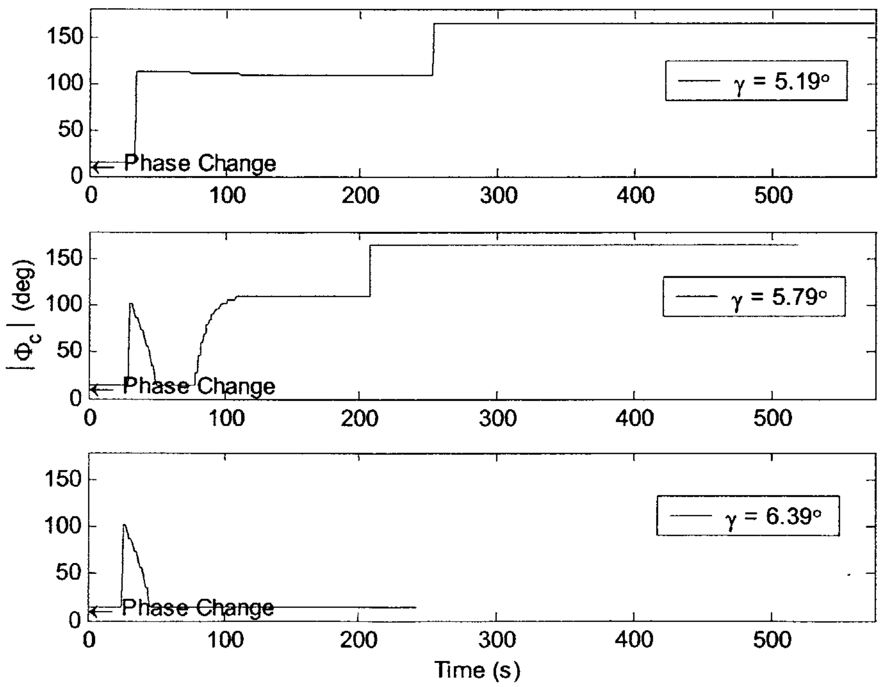


Figure 5.7: Commanded Bank Angle vs. Time, $V_\infty = 3$ km/s, L/D = 0. 25

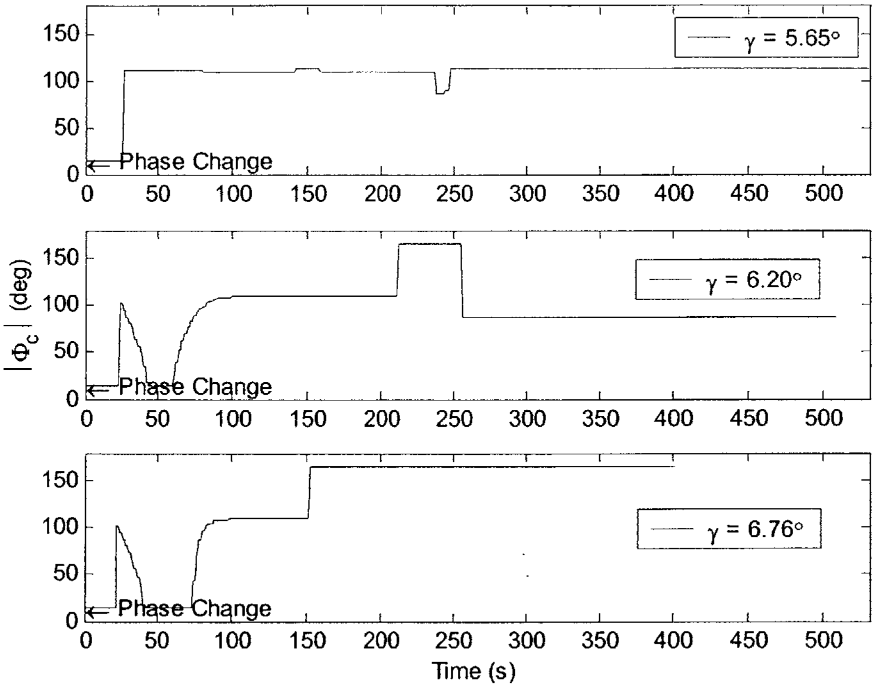


Figure 5.8: Commanded Bank Angle vs. Time, V_∞ = 6 km/s, L/D = 0.25

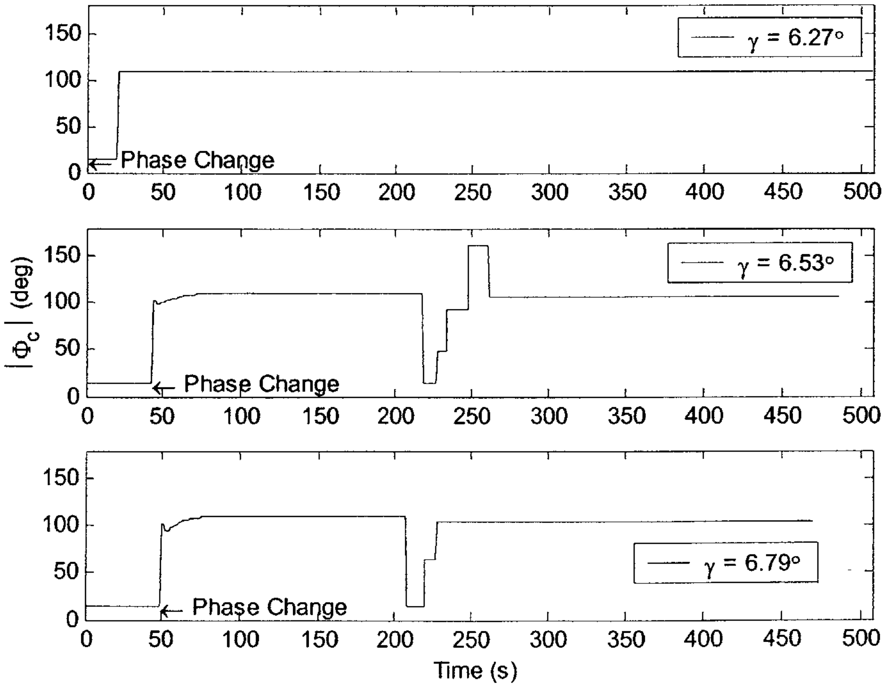


Figure 5.9: Commanded Bank Angle vs. Time, V_∞ = 10 km/s, L/D = 0.25

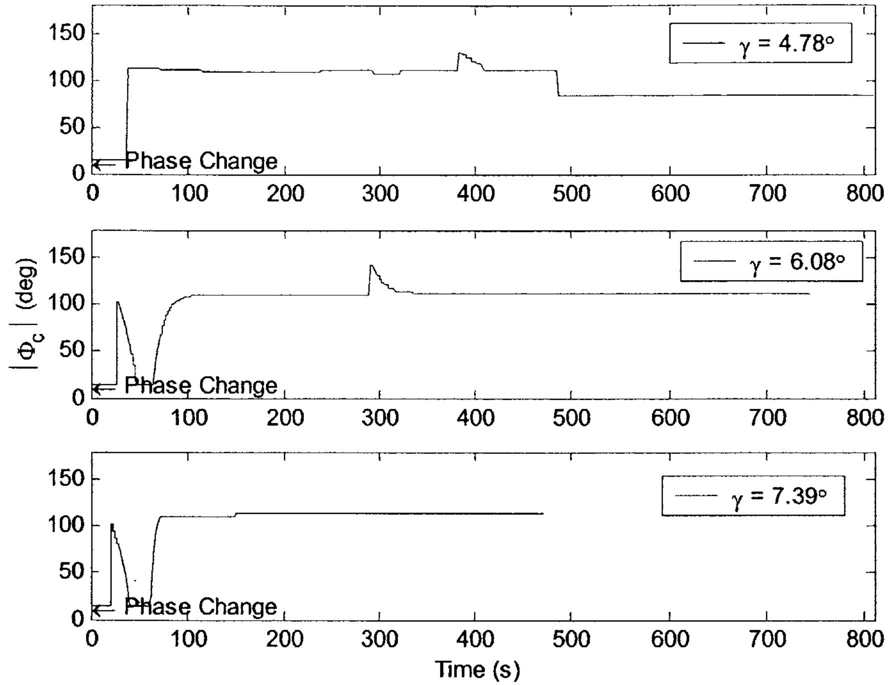


Figure 5.10: Commanded Bank Angle vs. Time, V_∞ = 1 km/s, L/D = 0.75

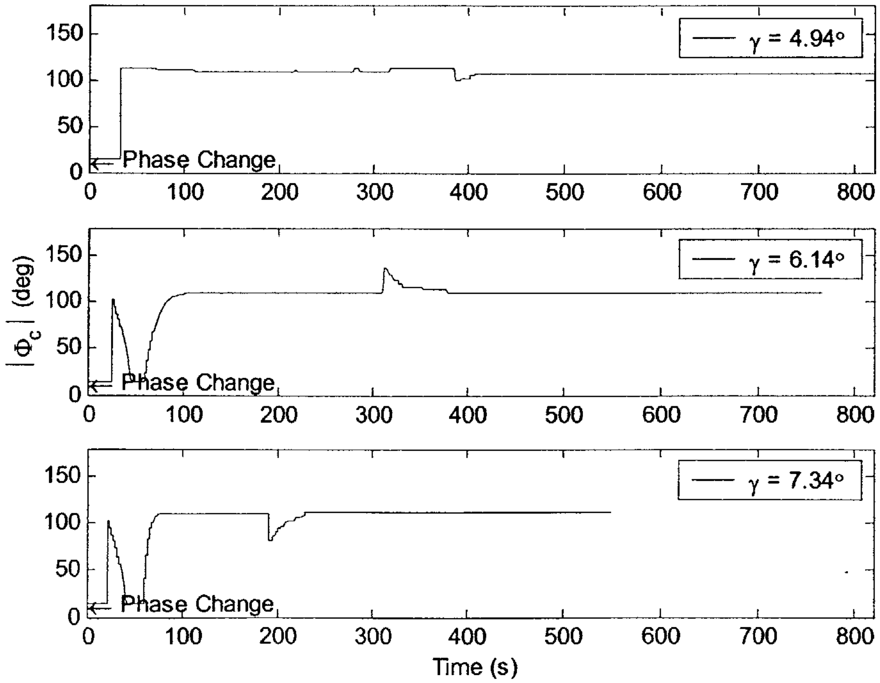


Figure 5.11: Commanded Bank Angle vs. Time, V_∞ = 3 km/s, L/D = 0.75

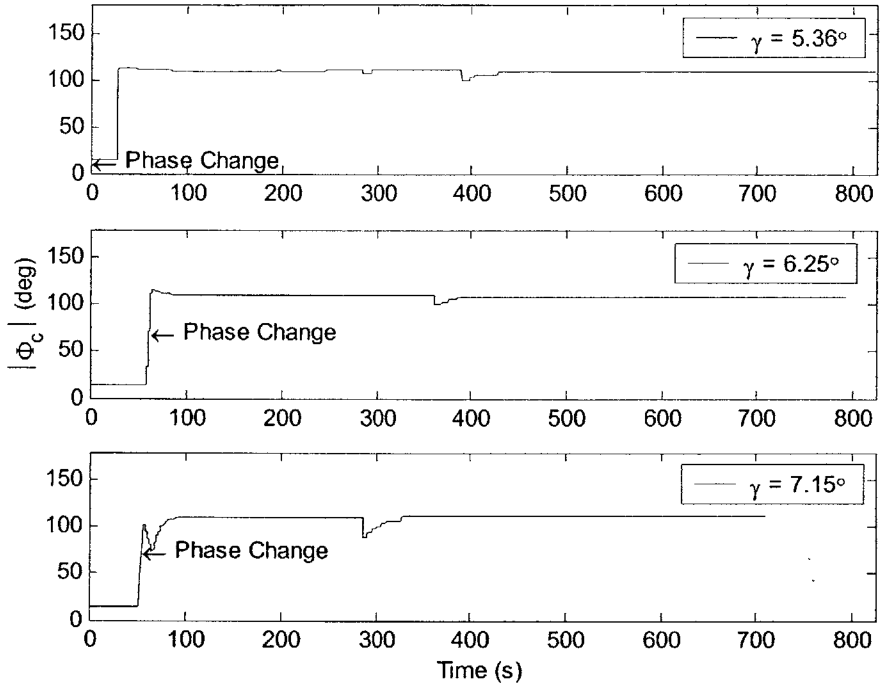


Figure 5.12: Commanded Bank Angle vs. Time, V_∞ = 6 km/s, L/D = 0.75

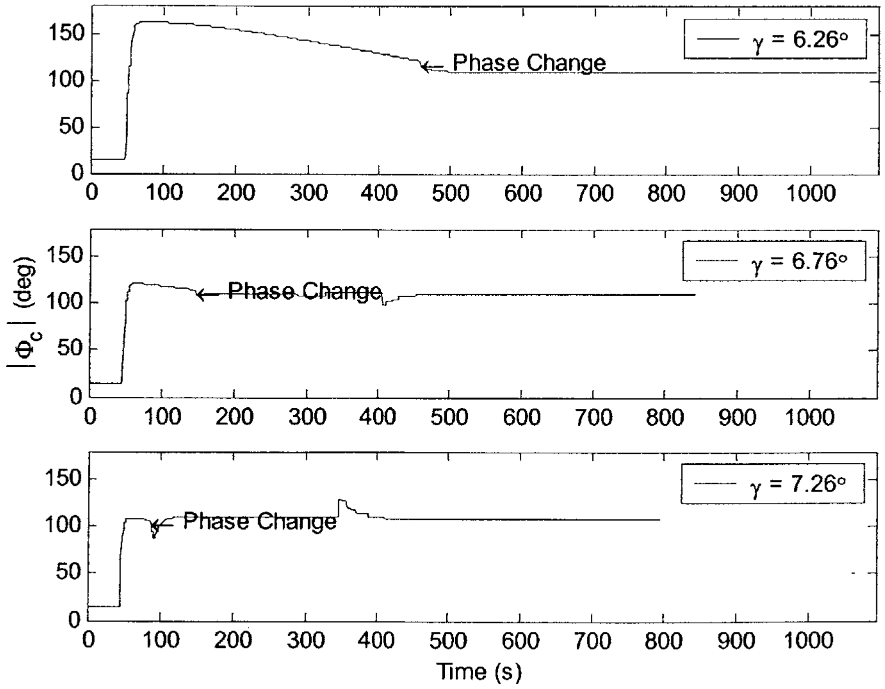


Figure 5.13: Commanded Bank Angle vs. Time, V_∞ = 10 km/s, L/D = 0.75

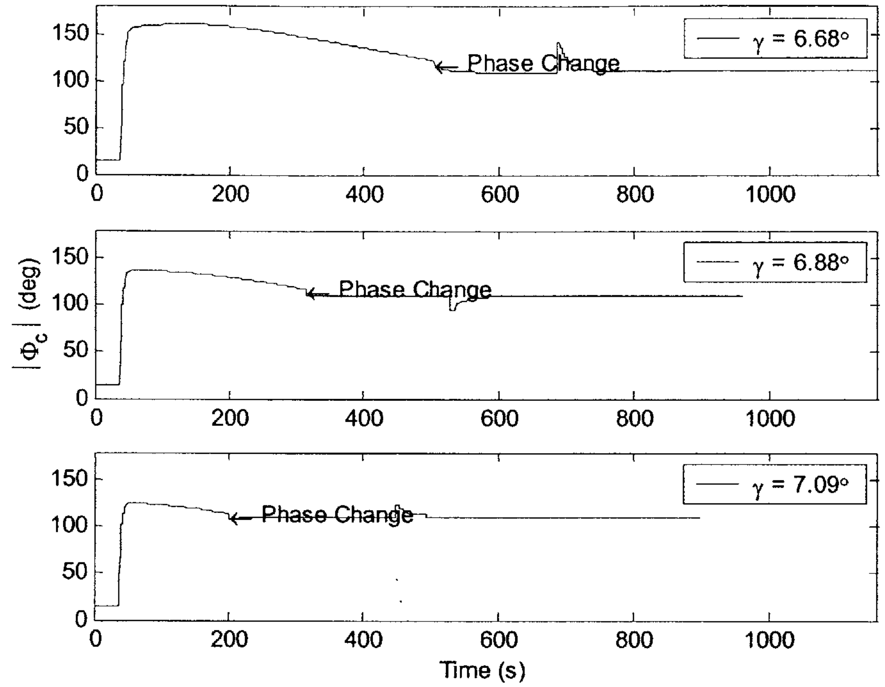


Figure 5.14: Commanded Bank Angle vs. Time, V_∞ = 14 km/s, L/D = 0.75

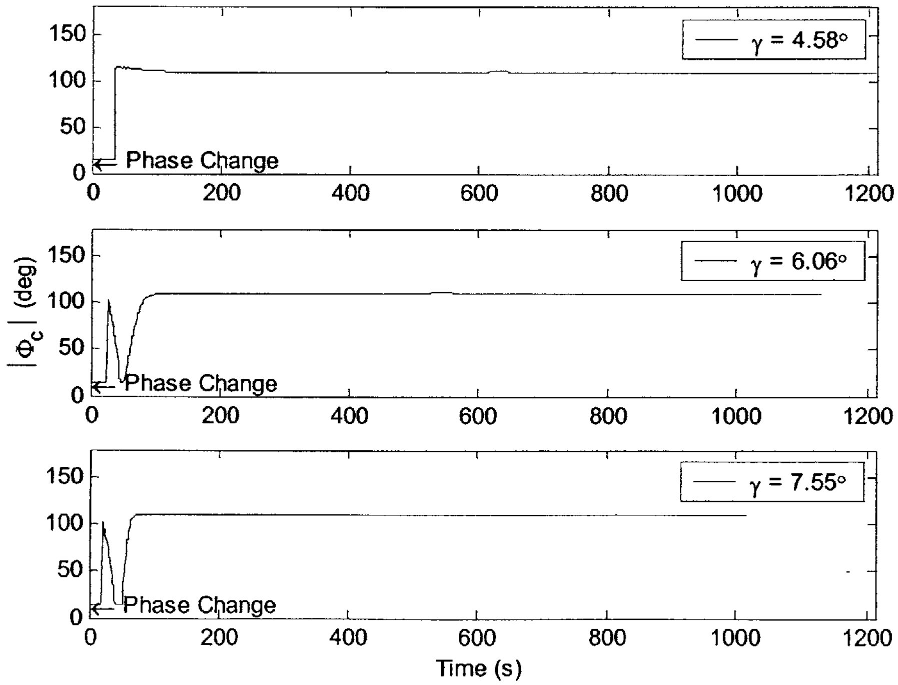


Figure 5.15: Commanded Bank Angle vs. Time, V_∞ = 1 km/s, L/D = 1.5

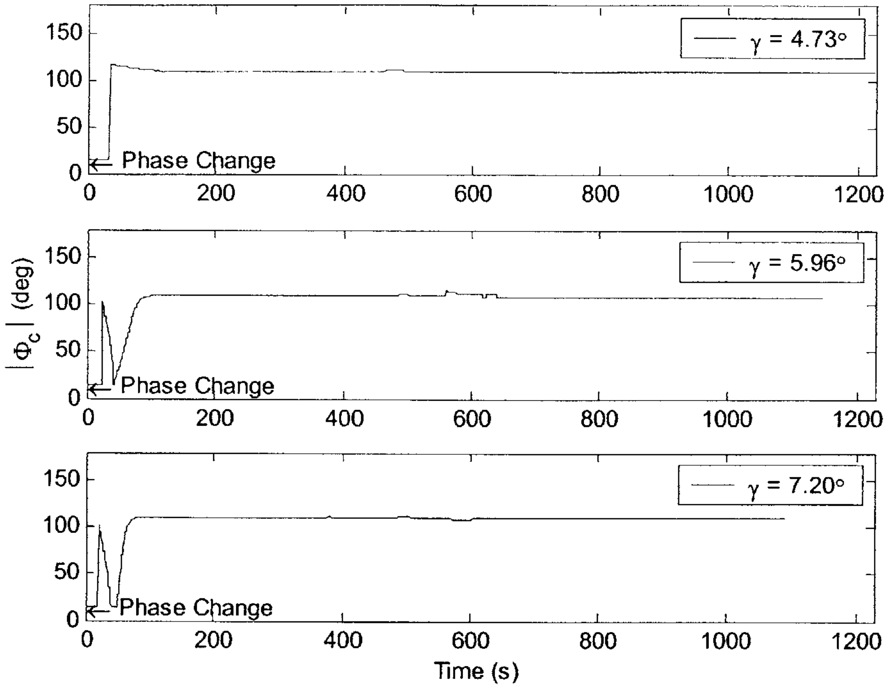


Figure 5.16: Commanded Bank Angle vs. Time, V_∞ = 3 km/s, L/D = 1.5

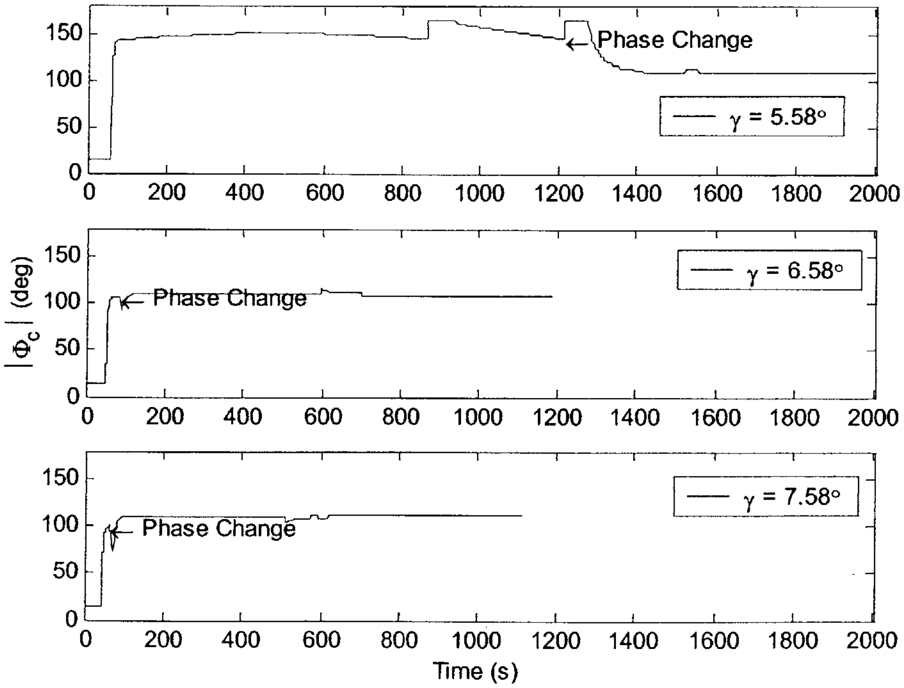


Figure 5.17: Commanded Bank Angle vs. Time, V_∞ = 6 km/s, L/D = 1.5

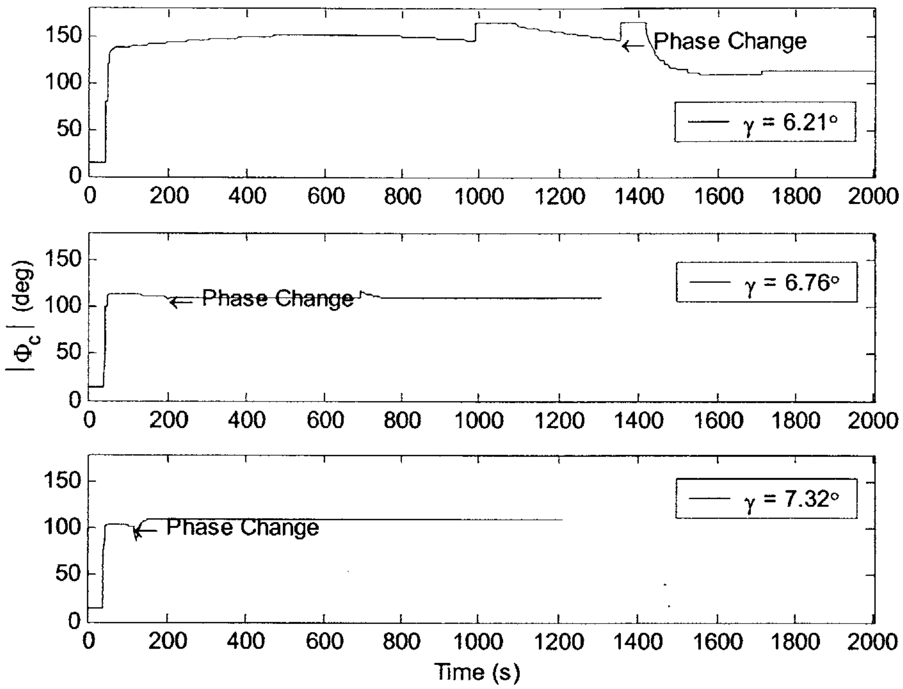


Figure 5.18: Commanded Bank Angle vs. Time, V_∞ = 10 km/s, L/D = 1.5

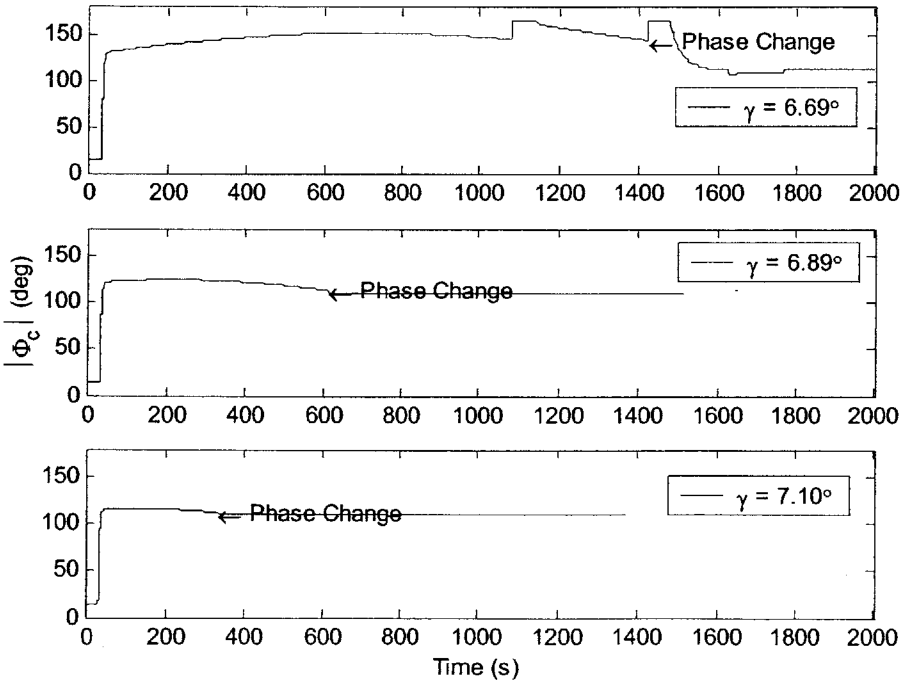


Figure 5.19: Commanded Bank Angle vs. Time, V_∞ = 14 km/s, L/D = 1.5

These figures show several important trends. First, at low velocities, for all lift-to-drag ratios, the guidance transitions to targeting almost immediately. This trend demonstrates that the phase check is working properly. Lower velocities do not require as much energy depletion for successful targeting and should transition to targeting earlier.

Second, at higher velocities, shallow flight path angles require longer energy management phases prior to transition to targeting. As lift-to-drag ratio increases at high velocities, regardless of flight path angle, the vehicle also spends longer in the energy management phase. These trends demonstrate that the energy management phase and phase change logic are working correctly. At shallow flight path angles, the vehicle does not dig as deeply into the atmosphere. Vehicles with high lift-to-drag ratios null the altitude rate more quickly so they also do not dig as deeply into the atmosphere. Due to lower drag at these atmospheric levels, the vehicle must spend longer in the energy management phase to deplete the same ΔV.

Third, almost all of the trajectories have a ripple in the commanded bank angle histories during later portions of the targeting phase. This ripple corresponds to the apogee miss criteria changeover point. At this point in targeting, the apogee miss criteria for a 'good' solution is reduced from 40 nm (74.08 km) to 3 nm (5.57 km). This trend demonstrates that the targeting guidance is successfully converging to bank angle solutions. When the acceptance criteria changes, guidance immediately takes action to hone in on the target.

5.3 Capturable Corridor Coverage

As discussed in Chapter 4, the capturable corridor is defined as the portion of the theoretical corridor for which the algorithm is able to successfully reach the target without exceeding a maximum acceleration of 10 g's. Figure 5.20 through Figure 5.25 show how the enhanced and original algorithms' capturable corridors compare to the theoretical corridors presented in Chapter 4. Corridors are depicted for lift-to-drag ratios as a function of V_∞.

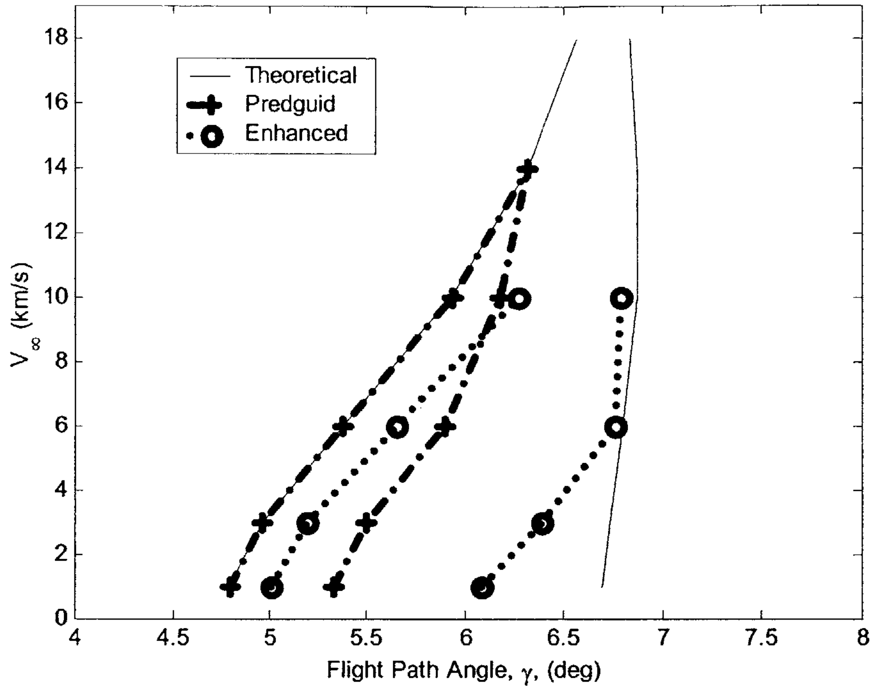


Figure 5.20: Theoretical and Capturable Corridors, L/D = 0.25

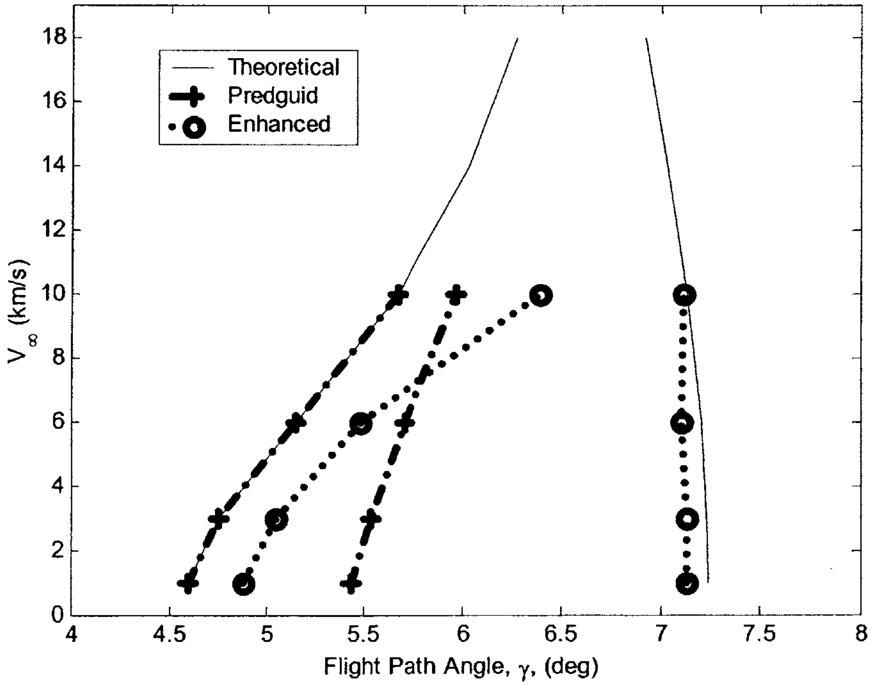


Figure 5.21: Theoretical and Capturable Corridors, L/D = 0.50

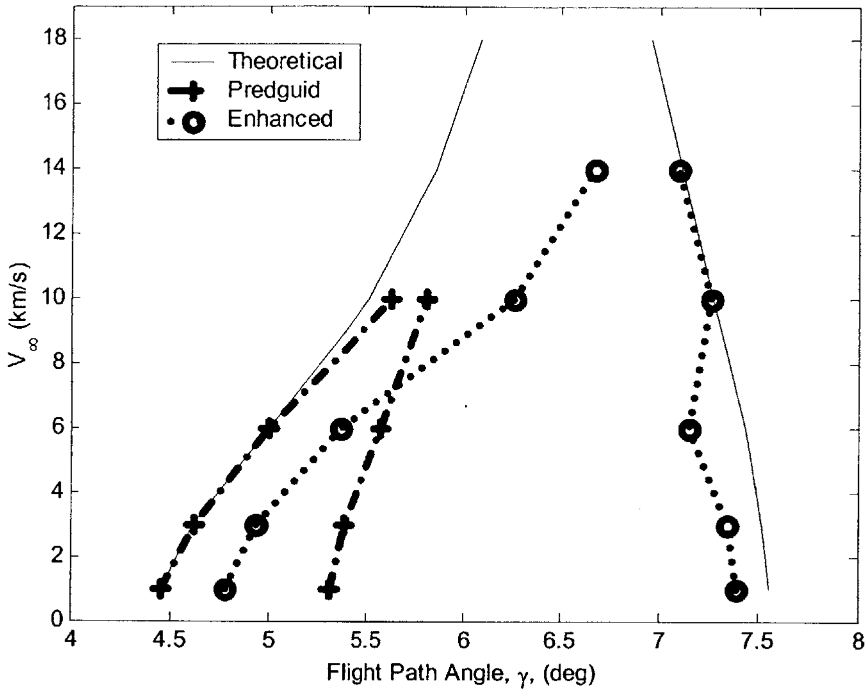


Figure 5.22: Theoretical and Capturable Corridors, L/D = 0.75

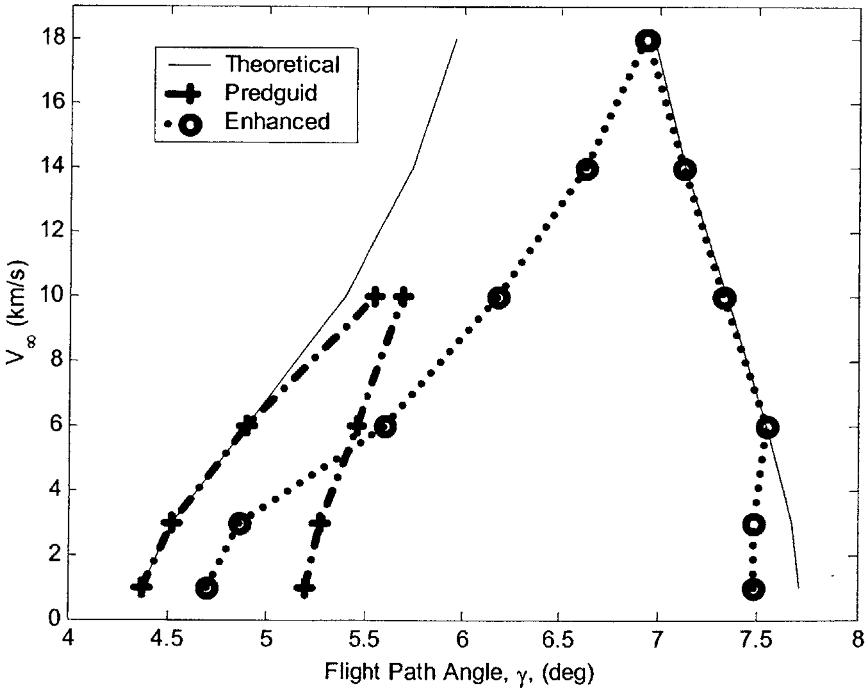


Figure 5.23: Theoretical and Capturable Corridors, L/D = 1.00

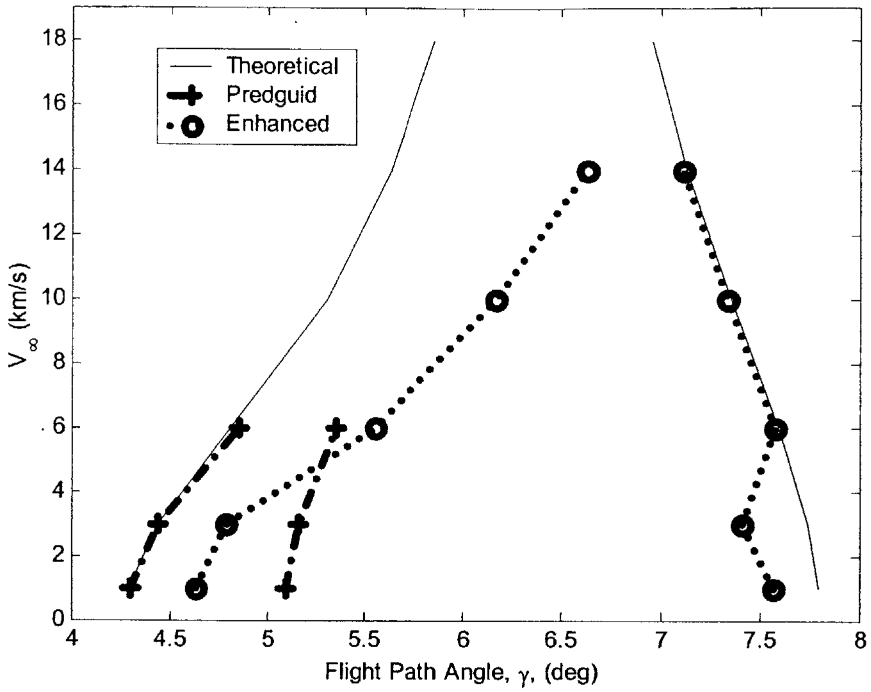


Figure 5.24: Theoretical and Capturable Corridors, L/D = 1.25

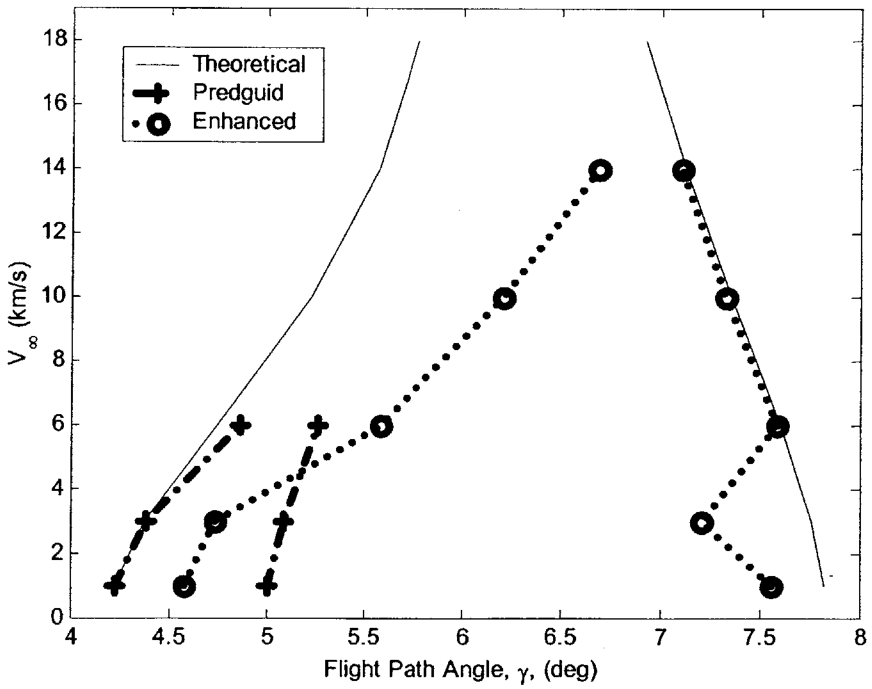


Figure 5.25: Theoretical and Capturable Corridors, L/D = 1.50

As you can see from the plots, the algorithm enhancements greatly improved corridor coverage, especially for the 14 km/s cases. Figure 5.26 shows the percent improvement over PredGuid of corridor captured for the various lift-to-drag ratios and values of V_∞.

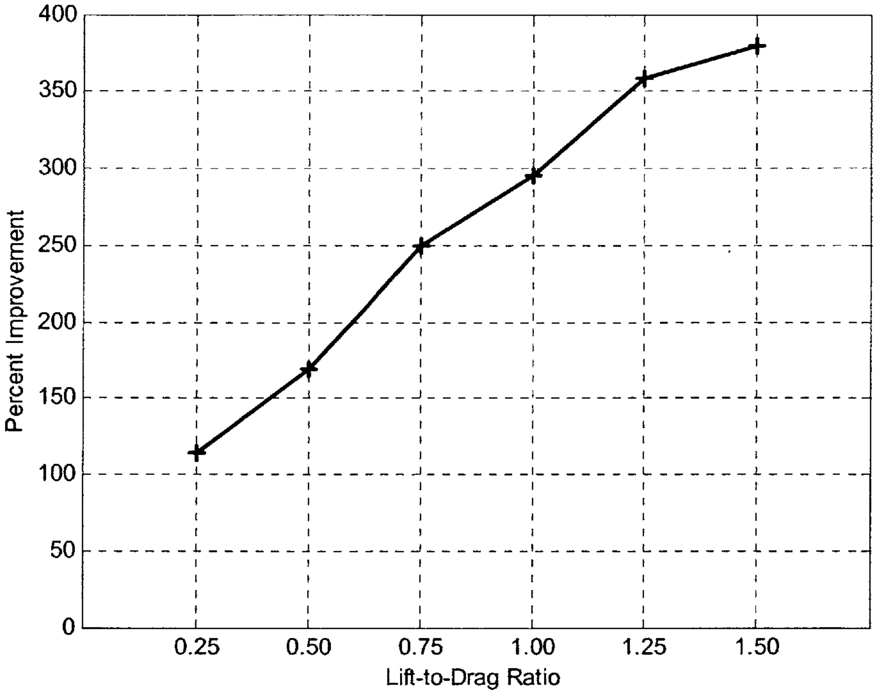


Figure 5.26: Percent Improvement of Corridor Captured

Most of the corridor coverage gain occurred on the steep end of the corridor. Also, two distinct areas of the theoretical corridor lacked coverage: the shallow end of the corridor (much of which lost area that PredGuid was able to cover) and the steep flight path angles with low velocity. These results can be explained by the nature of how each algorithm works.

5.3.1 Steep Flight Path Angle Corridor Gain

Steep end corridor coverage gain is the result of the enhanced algorithm's ability to remain at a particular altitude as long as necessary to remove energy. The original algorithm could only penetrate deeper into the atmosphere to generate more drag and deeper vehicle atmospheric penetration yields high acceleration forces. Figure 5.27 through Figure 5.29 depict an example case ($V_\infty = 6\ km/s$, $L/D = 0.75$, and $\gamma = 5.9°$) for which both algorithms are able to reach the target apogee, but the original algorithm causes a much higher (and unacceptable) acceleration.

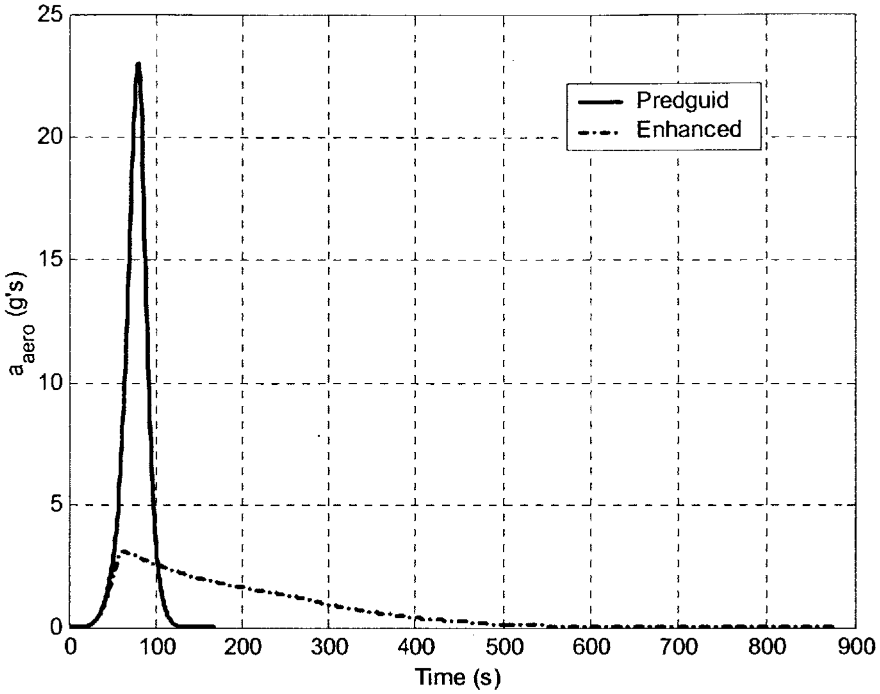


Figure 5.27: Acceleration vs. Time, V_∞ = 6 km/s, L/D = 0.75, γ = 5.9°

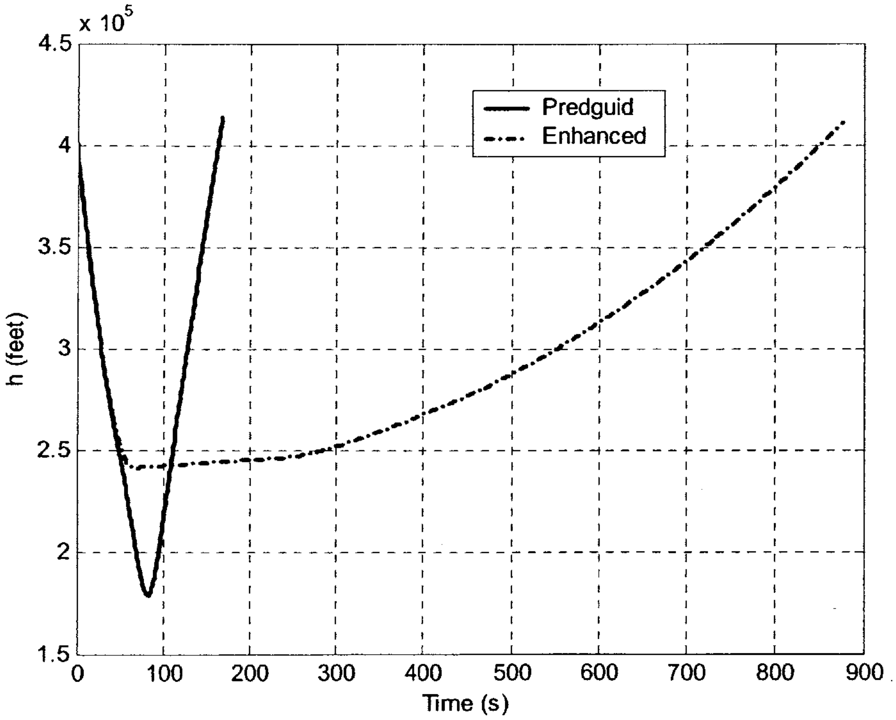


Figure 5.28: Altitude vs. Time, V_∞ = 6 km/s, L/D = 0.75, γ = 5.9°

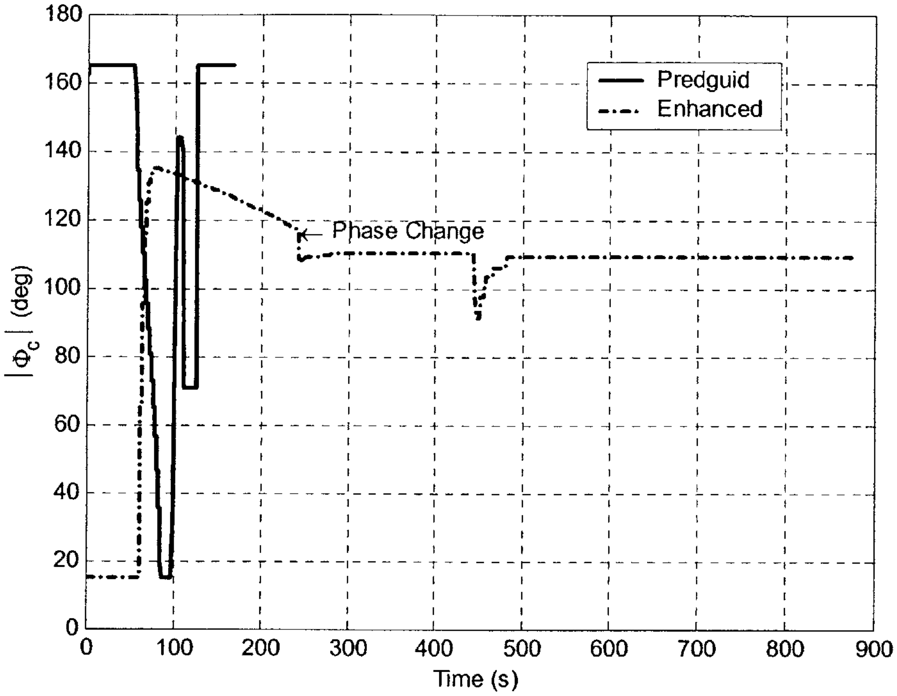


Figure 5.29: Commanded Bank Angle vs. Time, $V_\infty = 6$ km/s, L/D = 0.75, $\gamma = 5.9°$

5.3.2 Enhanced Algorithm Corridor Deficiencies

There are two distinct areas of deficiency: the shallow end of the corridor and the steep flight path angles with low hyperbolic excess velocity.

5.3.3 Shallow End Corridor Loss

On the shallow end of the corridor, the vehicle needs to fly more *lift down* throughout the trajectory in order to reach the target. However, the enhanced algorithm automatically attempts to first null the altitude rate by flying *full lift up* before checking for a phase change. By the time the vehicle flips to full lift down, the altitude rate cannot be slowed and the vehicle skips out of the atmosphere. Alternatively, PredGuid allowed the vehicle to fly lift down for the entire trajectory. As a result, the original algorithm had better shallow flight path angle coverage. Figure 5.30 through Figure 5.32 depict the bank angle, altitude and altitude rate profiles for an example case ($V_\infty = 3\ km/s$, $L/D = 0.75$, and $\gamma = 4.75°$) in which PredGuid was able to target the final apogee but the enhanced algorithm was not. The enhanced algorithm never transitioned out of the energy management phase and was unable to remain in the atmosphere even flying full lift down because the vehicle did not penetrate far enough into the atmosphere to generate enough lift to null the altitude rate.

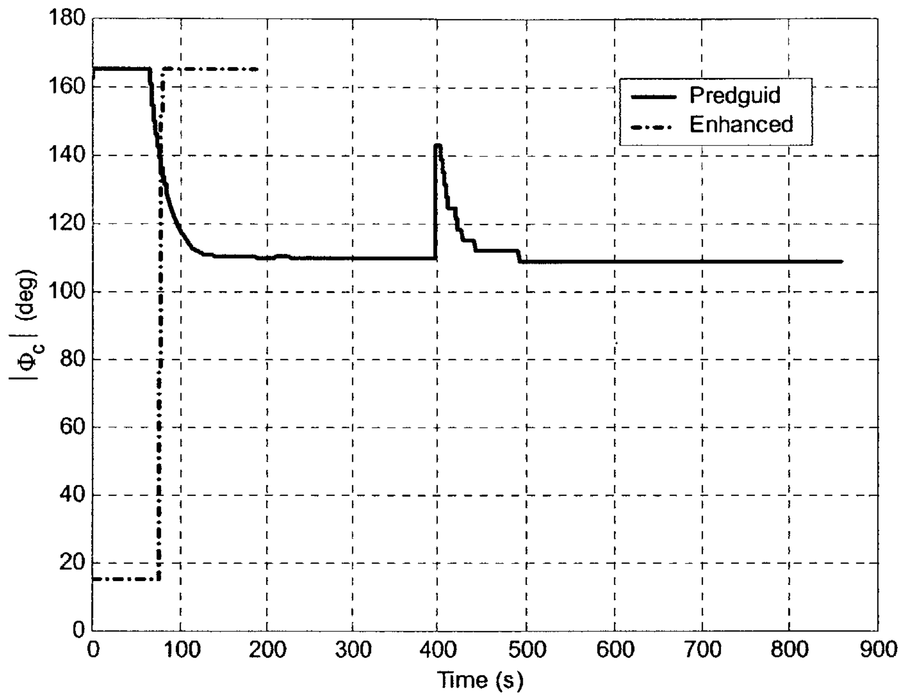


Figure 5.30: Commanded Bank Angle vs. Time, $V_\infty = 3$ km/s, L/D = 0.75, $\gamma = 4.75^\circ$

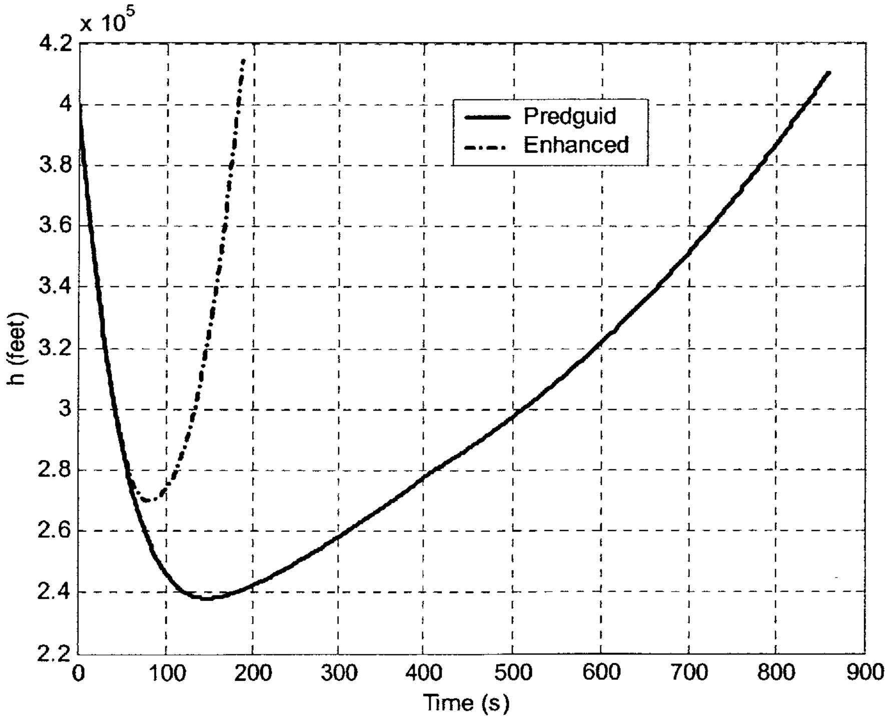


Figure 5.31: Altitude vs. Time, $V_\infty = 3$ km/s, L/D = 0.75, $\gamma = 4.75^\circ$

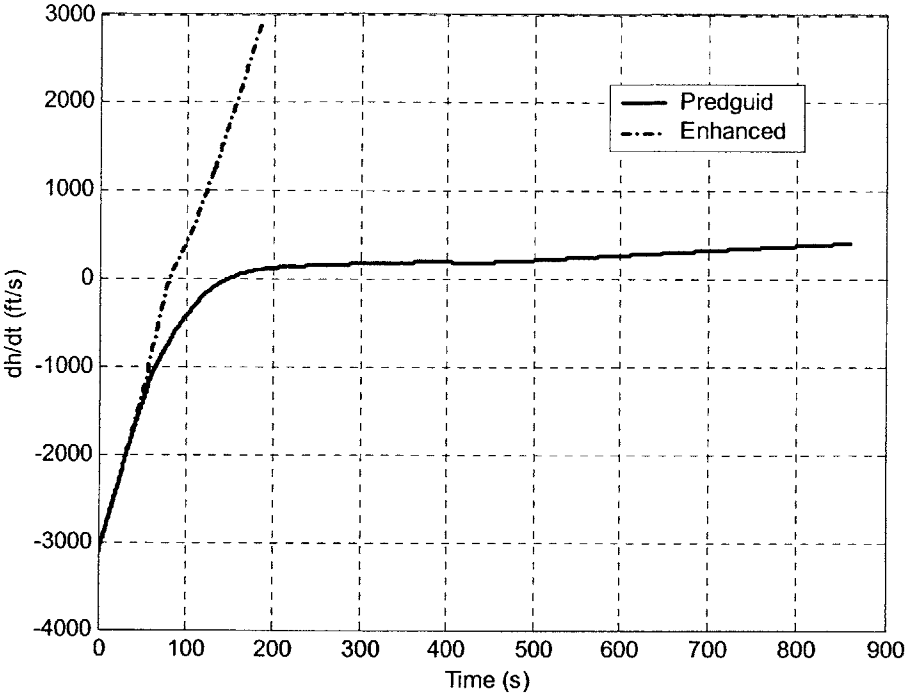


Figure 5.32: Altitude Rate vs. Time, $V_\infty = 3$ km/s, L/D = 0.75, $\gamma = 4.75^\circ$

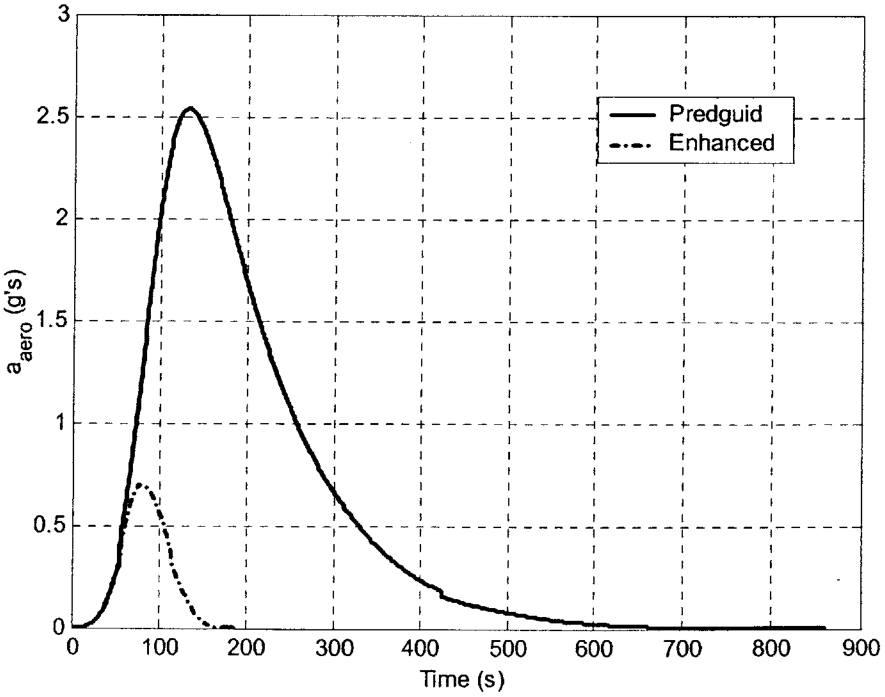


Figure 5.33: Acceleration vs. Time, $V_\infty = 3$ km/s, L/D = 0.75, $\gamma = 4.75^\circ$

This same trend continued throughout the higher hyperbolic excess velocities. Even the vehicles with the highest lift-to-drag ratios were still unable to capture on the shallow side of the corridor. The 'shallow' range extended to steeper and steeper flight path angles as V_∞ increased. Figure 5.34 and Figure 5.35 demonstrate a sample case with a fairly steep flight path angle but high velocity and low lift-to-drag ratio ($V_\infty = 14\ km/s$, $L/D = 0.25$, $\gamma = 6.5°$). Figure 5.36 and Figure 5.37 show a sample case with high velocity and high lift-to-drag ratio ($V_\infty = 18\ km/s$, $L/D = 1.5$, $\gamma = 6.25°$).

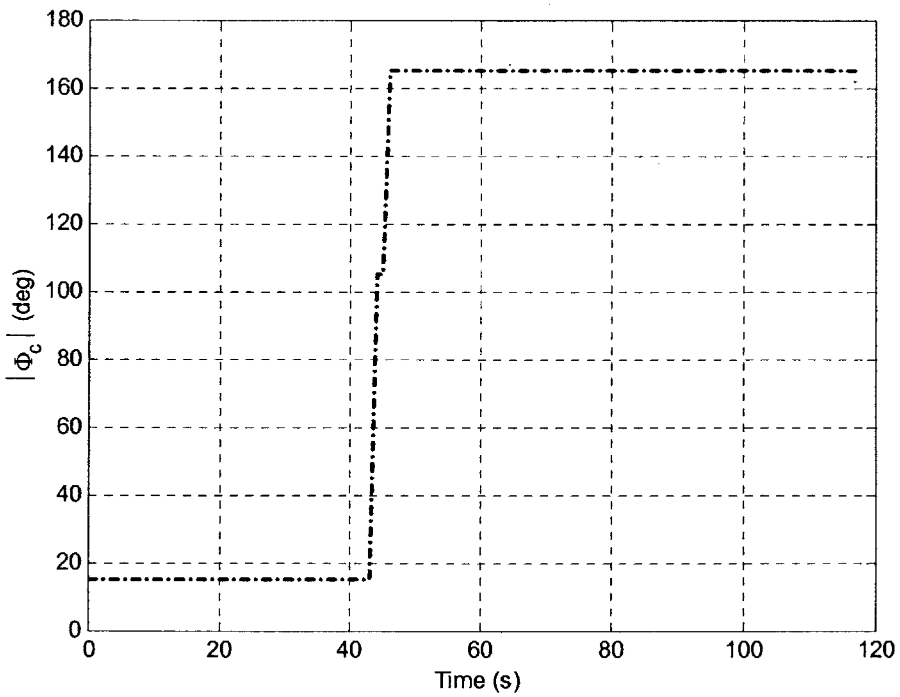


Figure 5.34: Commanded Bank Angle vs. Time, V_∞ = 14 km/s, L/D = 0.25, γ = 6.5°

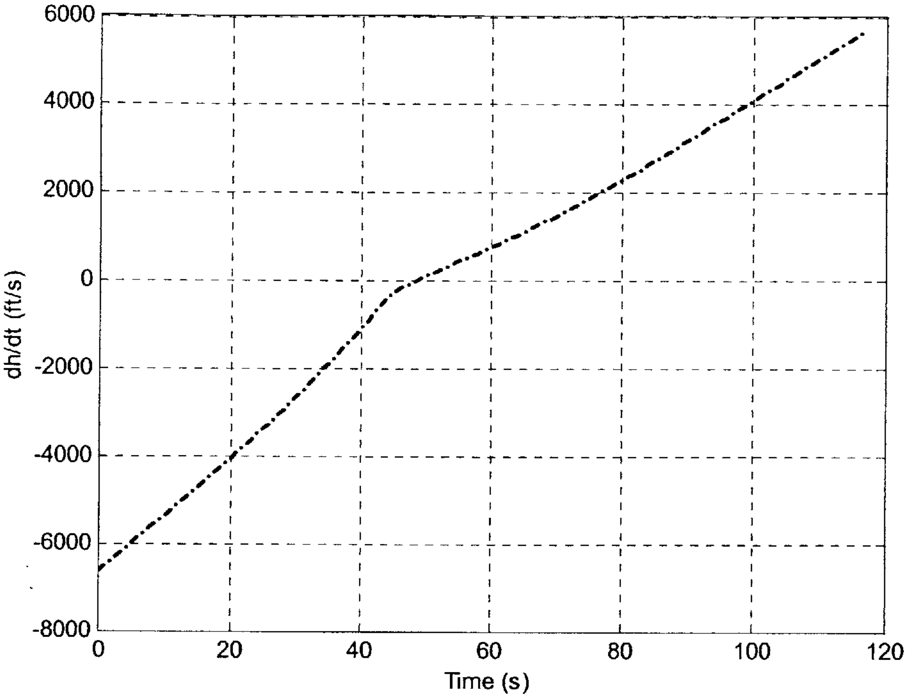


Figure 5.35: Altitude Rate vs. Time, V_∞ = 14 km/s, L/D = 0.25, γ = 6.5°

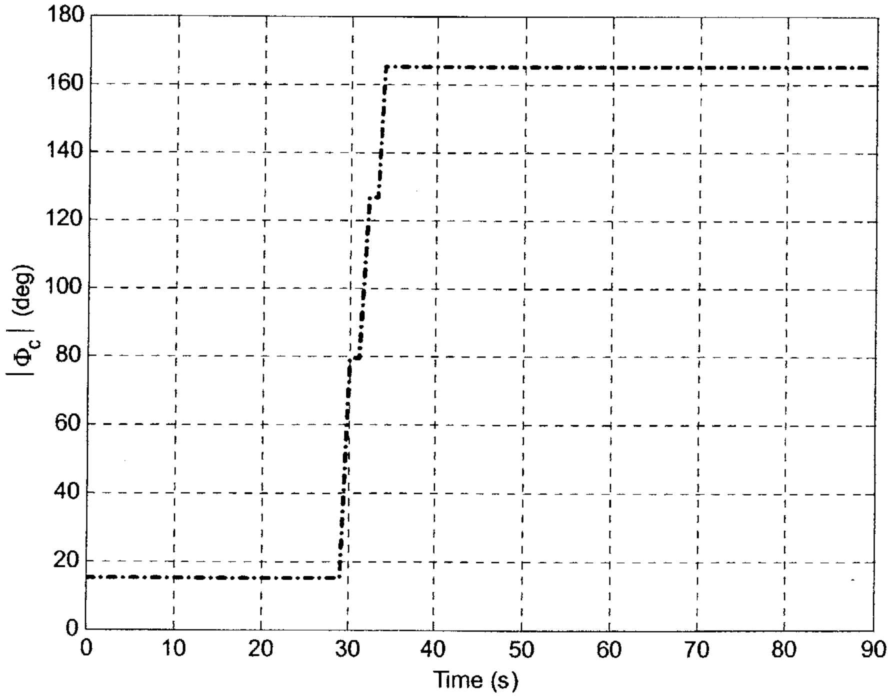


Figure 5.36: Commanded Bank Angle vs. Time, V_∞ = 18 km/s, L/D = 1.5, γ = 6.25°

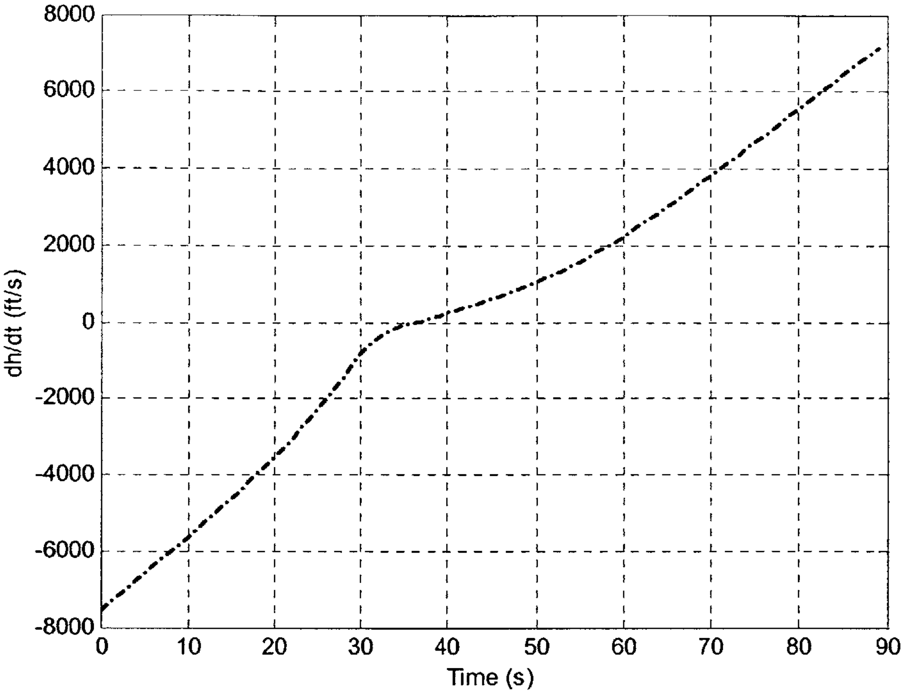


Figure 5.37: Altitude Rate vs. Time, V_∞ = 18 km/s, L/D = 1.5, γ = 6.25°

As you can see, the vehicle response is virtually identical for all of the cases shown. For shallow entries, flying full lift up set up orbital conditions from which the vehicle could not recover. The vehicle never transitioned out of the energy management phase and was unable to remain in the atmosphere despite flying full lift down.

5.3.4 Steep Flight Path Angles with Low Velocity

Figure 5.38 through Figure 5.40 show a sample case with a steep flight path angle beyond the capturable corridor of the enhanced algorithm ($V_\infty = 6\,km/s$, $L/D = 0.75$, $\gamma = 7.25°$).

The algorithm successfully targeted the apogee, but the acceleration loads were unacceptably high. The peak acceleration occurs after the vehicle has already transitioned to targeting. At these steep flight path angles, transition to targeting occurred too early before guidance was able to null the altitude rate and dissipate energy. Transition occurred early because the predicted trajectories indicated capture. This problem only occurs at low velocities which are more susceptible to capture. Once guidance switched to the targeting phase prematurely, the excessive acceleration problems that plagued PredGuid resurfaced.

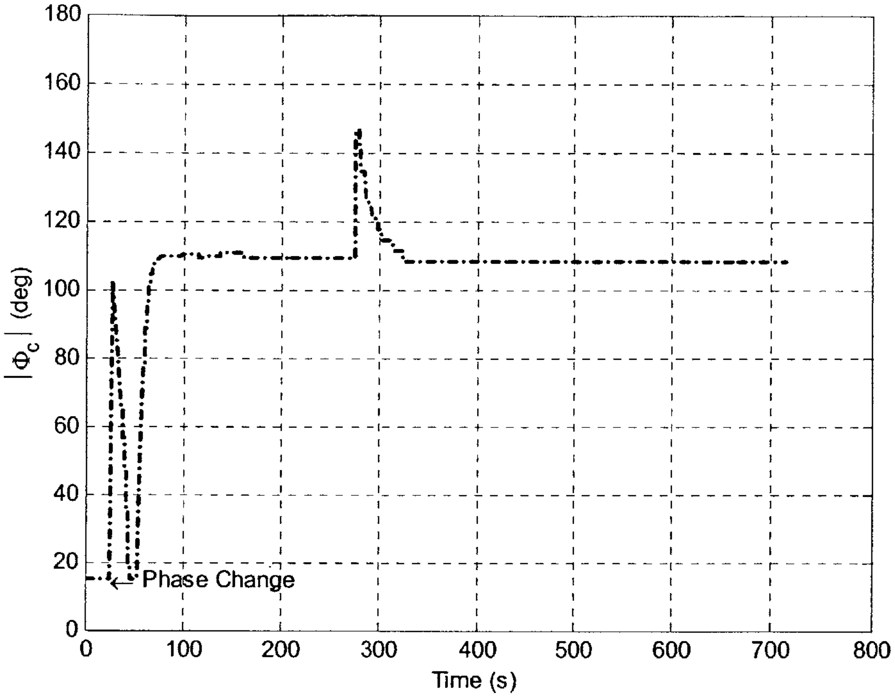


Figure 5.38: Commanded Bank Angle vs. Time, $V_\infty = 6$ km/s, L/D = 0.75, $\gamma = 7.25°$

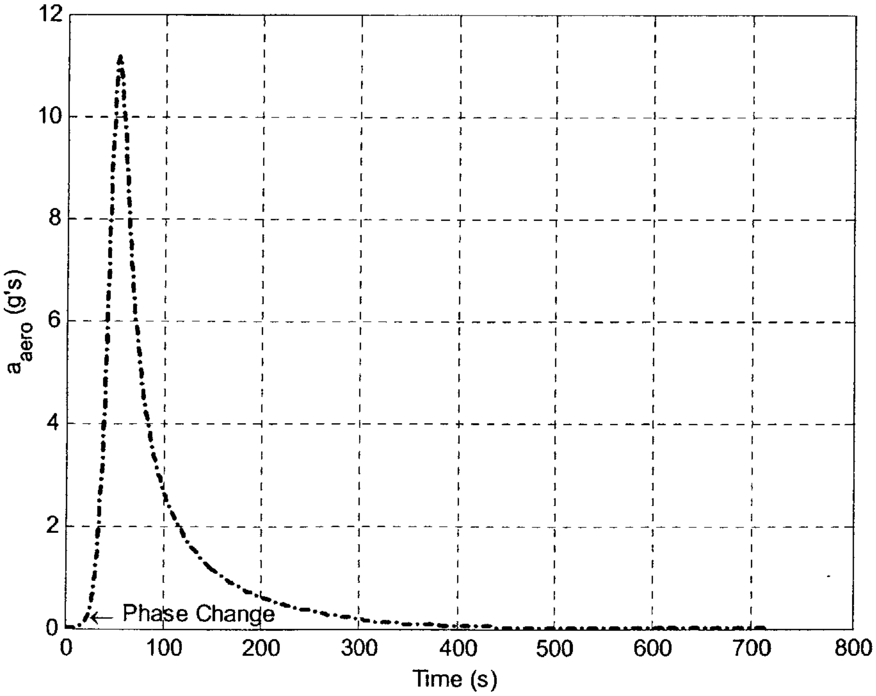


Figure 5.39: Acceleration vs. Time, $V_\infty = 6$ km/s, L/D = 0.75, $\gamma = 7.25°$

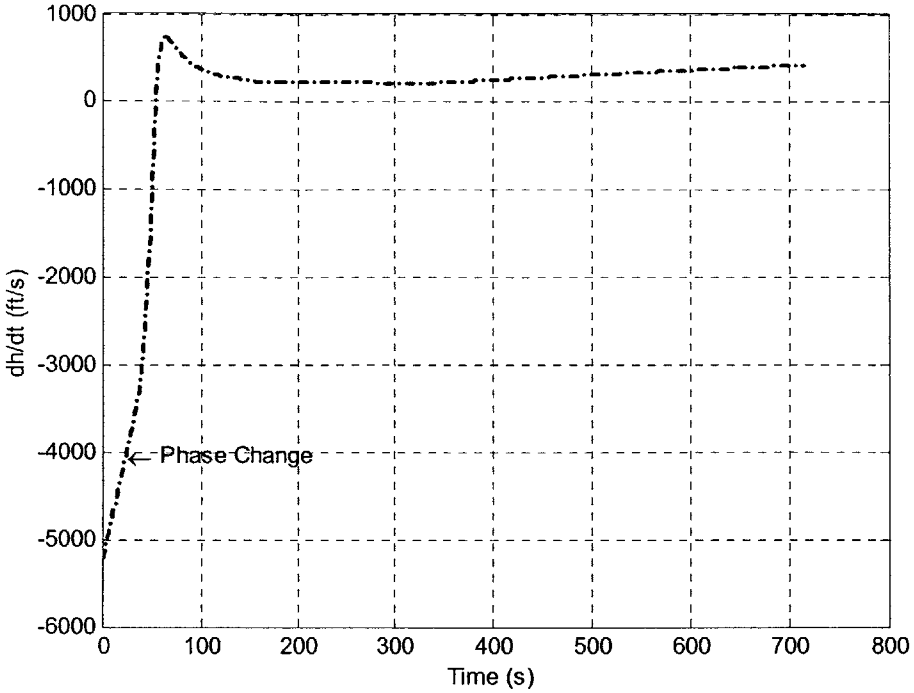


Figure 5.40: Altitude Rate vs. Time, V_∞ = 6 km/s, L/D = 0.75, γ = 7.25°

5.4 Algorithm Metrics

Improved corridor coverage should indicate a better algorithm. However, a discussion of improvement based solely on corridor coverage is incomplete for this thesis because only acceleration loads were considered in the corridor constraints. There are several other factors to consider when evaluating performance including heating rate, heating load, and ΔV to raise perigee. Each of these metrics as well as acceleration will be discussed in this section.

5.4.1 Acceleration

Figure 5.41 shows maximum vehicle acceleration as a function of flight path angle for the original and enhanced algorithms, respectively. Each dot represents a lift-to-drag ratio, V_∞, and flight path angle trio. As you can see from the graphs, the original algorithm yielded much higher maximum accelerations than the enhanced algorithm at the same flight path angles. These results are not surprising because the algorithm was designed to operate at a lower acceleration over a longer period of time. These figures also help visualize the expanded corridor coverage. If the acceleration limit were placed at a different level than 10 g's, the capturable corridor limits would change.

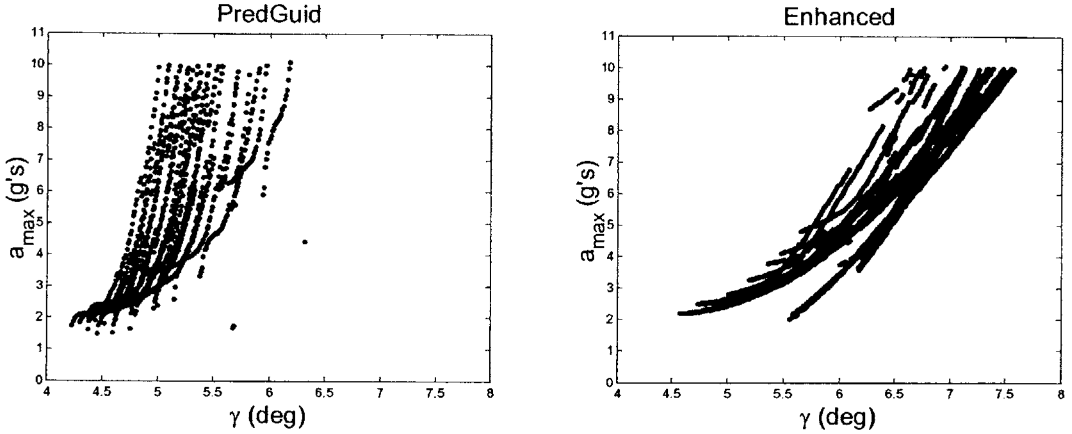


Figure 5.41: Maximum Acceleration vs. Flight Path Angle

5.4.2 Heating Rate

Maximum heating rate is another factor that must be considered. Again, at similar flight path angles, the enhanced algorithm had lower maximum heating rates as shown in Figure 5.42. The lower maximum heating rates are a product of the lower decelerations and shallower trajectories. The few high heating rates visible in the enhanced algorithm plot are high velocity trajectories. There are no figures to compare them against because the original algorithm does not work for these cases.

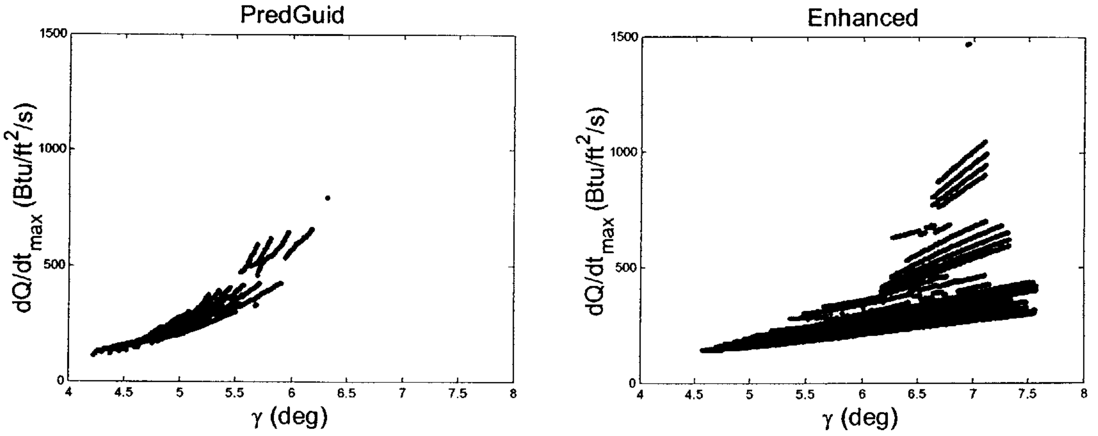


Figure 5.42: Maximum Heating Rate vs. Flight Path Angle

5.4.3 Heating Load

The enhanced algorithm typically resulted in longer periods of time spent in the atmosphere which would normally cause higher heating loads. However, as discussed above, the heating rates were lowered. The longer atmospheric flights do not appear to cause higher heating loads due to the decreased heating rates. Heating loads as a function of time are presented in Figure 5.43. The few high heating loads visible in the enhanced algorithm graph are for high velocity trajectories. As before, these cannot be compared to known data because the original algorithm failed to capture any portion of the corridor at those velocities.

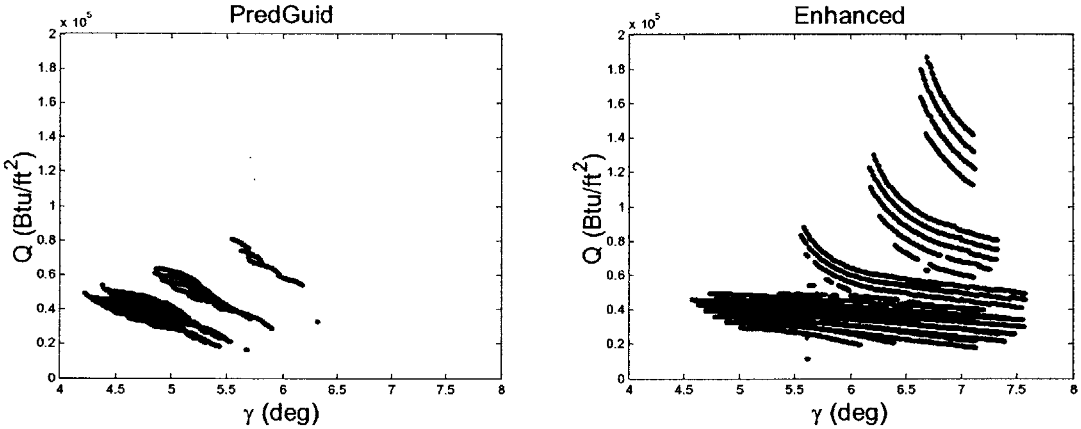


Figure 5.43: Heating Load vs. Flight Path Angle

5.4.4 Velocity Change to Raise Perigee

The final factor examined was the ΔV required to circularize the orbit after aerocapture. As seen in Figure 5.44, ΔV required to circularize held fairly stable for all flight path angles because both PredGuid and the enhanced algorithm used the same targeting algorithm. Even the flight path angles on the portion of the corridor captured by the enhanced algorithm still had approximately the same ΔV as the portion captured by the original algorithm. PredGuid is an excellent targeting algorithm and the energy management additions did not inhibit its function.

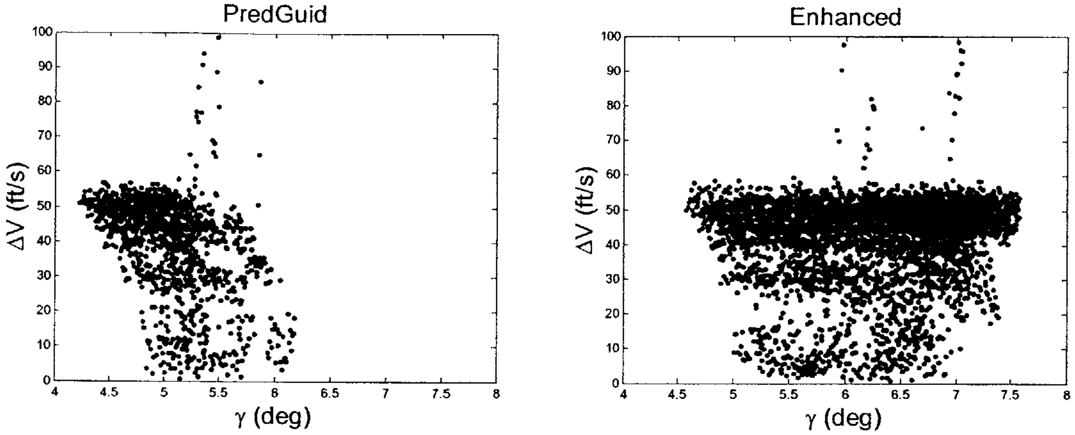


Figure 5.44: ΔV to Circularize vs. Flight Path Angle

5.5 Lateral Control

The updated lateral control portion of the guidance program worked fairly well. Figure 5.45 depicts the lateral corridor information for a nominal case which remained in the energy management phase for a significant period of time ($V_\infty = 14\,km/s$, $L/D = 0.75$, $\gamma = 6.9°$).

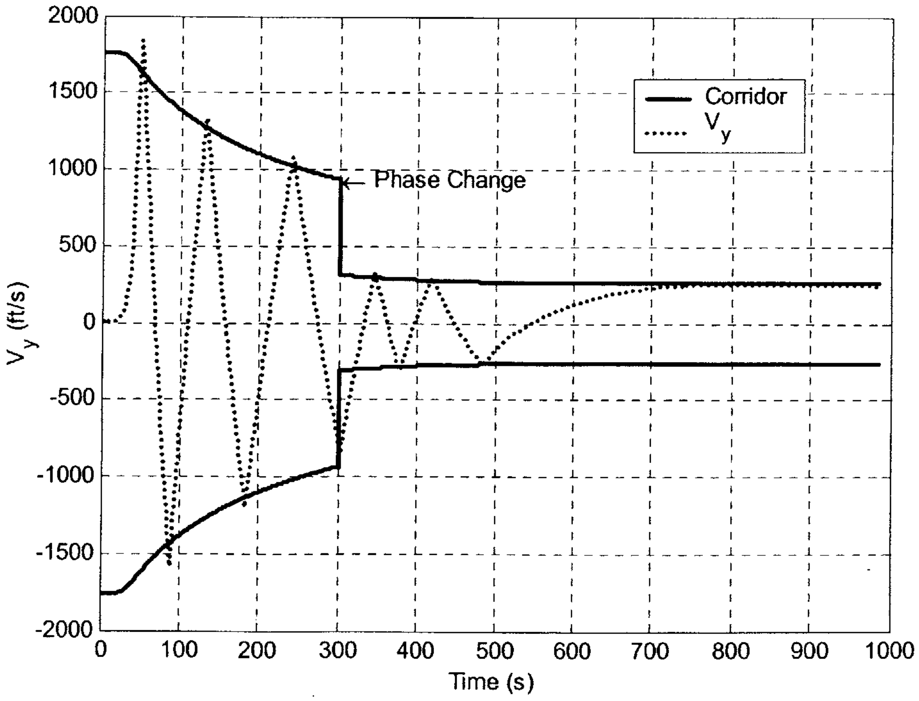


Figure 5.45: Out of Plane Velocity vs. Time, V_∞ = 14 km/s, L/D = 0.75, γ = 6.9°

As you can see from the above figure, the bank angle reversals are executed in a timely manner. The use of percentage of forward velocity allowed a gradual narrowing of the corridor early in the trajectory and prevented excessive reversals. However, the corridor did not narrow as quickly near the end of the trajectory because the vehicle is decelerating at a slower rate. For higher lift-to-drag ratio cases, bank reversals occurred more frequently because higher lift-to-drag ratios cause more out of plane lift. Figure 5.46 depicts corridor information for a higher lift-to-drag ratio case. Finally, a case was examined which did not remain in the energy management phase for an extended period ($V_\infty = 6\ km/s$, $L/D = 0.25$, $\gamma = 6.0°$). Figure 5.47 depicts these results. Even without the gradual narrowing of the lateral corridor during energy management, the guidance still managed to keep the lateral error within the corridor.

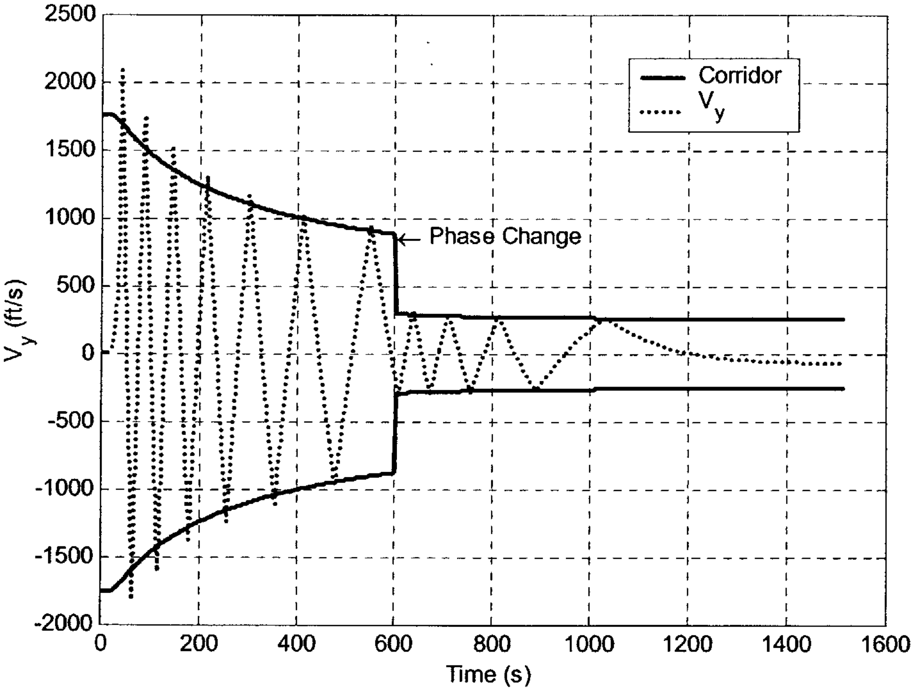


Figure 5.46: Out of Plane Velocity vs. Time, V_∞ = 14 km/s, L/D = 1.5, γ = 6.9°

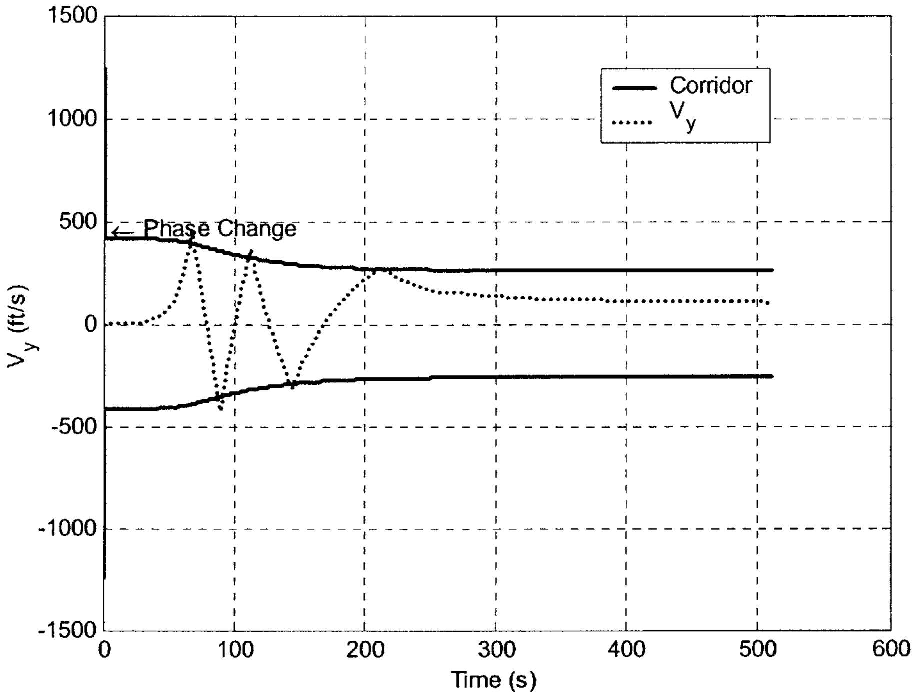


Figure 5.47: Out of Plane Velocity vs. Time, V_∞ = 6 km/s, L/D = 0.25, γ = 6.0°

Chapter 6
Conclusions

6.1 Summary and Conclusions

This thesis sought to enhance an existing numerical predictor-corrector aerocapture guidance algorithm. The enhancements were accomplished by implementing an energy management phase prior to targeting exit conditions and replacing heuristic features with more generic features. The purpose of the energy management phase is to deplete sufficient energy to allow targeting of the exit conditions in the targeting phase. The vehicle response during energy management can be modeled as a second-order spring/mass/damper system. The bank angle is determined by calculating the bank angle for zero altitude acceleration and then adding an altitude rate damper and a term to null the drag profile deviation. Phase change occurs when two conditions have been met: First, if at the given point in the trajectory, the vehicle could fly a constant bank angle of 110° for the remainder of the trajectory and have the resulting apogee be below or within a given tolerance above the target apogee. Second, the predicted final energy must indicate that the vehicle would be on an elliptical, not hyperbolic, trajectory.

Implementation of the energy management phase allowed removal of several heuristic features. Modeling of a separate lift down phase was replaced by using a lift down condition to determine phase change and biasing to the same lift down condition during targeting. Use of a heuristic sensitivity to calculate the first corrected bank angle was replaced by a simple 'smart guessing' algorithm based on whether the predicted apogee was above or below the target. Heuristic lateral corridor boundaries were replaced by boundaries based on percentage of forward velocity.

An analysis of the resulting flight path angle entry corridor revealed that the enhanced algorithm allowed 114% corridor coverage improvement the lowest lift-to-drag ratio, and 379% for the highest lift-to-drag ratio. The mean improvement was 261% over the PredGuid capturable corridor for all velocities and lift-to-drag ratios. Most of the gain occurred for higher flight path angles. However, results also indicated that at the same conditions, the enhanced algorithm yielded lower maximum accelerations. Comparable heating rates, heating loads, and ΔV to raise perigee were observed. The ΔV to raise perigee results strongly indicate that PredGuid is an excellent targeting algorithm and the enhancements did not inhibit the algorithm's targeting performance.

6.2 Recommendations for Future Work

There are several areas of this project which could be examined in the future.

Shallow Flight Path Angles: One of the most important areas for future study would be to examine how to make the enhanced algorithm work for the shallow end of the entry corridor. The original guidance algorithm was able to capture within this region but the enhanced algorithm could not. There should be a recourse which enables the algorithm to vary the initial guidance command rather than simply flying full lift up to null the altitude rate.

Steep Flight Path Angles with Low Velocity: This area was the only other 'hole' in the enhanced algorithm's corridor coverage. Guidance transitions between the energy management and targeting phases prematurely at these conditions. Another condition needs to be implemented into the phase check to prevent this premature transition.

Apogee Miss Criteria: The only true heuristic remaining in the program is the apogee error miss. Due to the extreme sensitivity of final apogee to bank angle at high velocities, placing a constraint to narrow the apogee miss as velocity decreased was unavoidable. This constraint caused reactionary spikes in almost every bank angle profile as the apogee miss suddenly dropped from 40 nm to 3 nm at a velocity of 26,600 ft/s. A more generic method, possibly based on percentage of velocity or energy decrease, to generate a more gradual narrowing of the apogee miss should be investigated.

Modeling Bank Changes: This thesis assumed instantaneous bank angle changes. For small changes in bank angle, this assumption is fairly accurate. However, as bank changes increase, the reversals and corrections can take a significant amount of time. Incorporation of actual vehicle roll rate would be more realistic. Modeling should include roll rate, roll acceleration and deceleration, and the ability to roll over or under through the shortest distance.

Lateral Corridor: The lateral control presented in this thesis works reasonably well. However, there is still significant out of plane velocity error present at the end of the trajectory. Another method should be investigated which allows additional narrowing of the lateral corridor towards the end of the trajectory.

Robustness analysis: This thesis assumed perfect navigation and control. A robustness analysis including density dispersions, lift-to-drag ratio estimate errors, control errors, etc. would provide a better metric of how well the algorithm works.

[This Page Intentionally Left Blank]

Appendix A
PredGuid

A.1 Aerocapture Executive

Aerocapture guidance uses a predictor-corrector method to determine the periodic bank angle commands that control the spacecraft to a target apogee. Bank angle commands are given to produce desired exit conditions. Bank angle commands generate an excessive out of plane velocity; as a result, periodic roll reversals are commanded to keep the spacecraft within the desired orbital plane.

The logical flow of the Aerocapture Executive algorithm is illustrated in Figure A.1.

Functional flow of the Aerocapture Executive is illustrated in Figure A.2

The data inputs to pred_guid.m are described in Table A-1.

The data outputs of pred_guid.m are described in Table A-2.

The constants used in pred_guid.m are described in Table A-3.

1. If this is the first guidance pass (guidance initialization flag is set to one), execute steps *a* through *c*. Otherwise, proceed directly to step *2*:
    a. Define the constants structure, c.
    b. Initialize the local guidance parameters structure, l.
    c. Execute the Guidance Initialization procedure to fill l.
2. If the spacecraft is in the sensible atmosphere (indicated by a user-defined constant g value), update coefficient of drag, density gain, and lift-to-drag ratio by executing the Aerodynamic Properties procedure. Otherwise, proceed directly to step *3*.

3. If the spacecraft is in the sensible atmosphere (indicated by a constant g value higher than the constant g value required to execute the Aerodynamic Properties procedure ensuring that the Aerodynamic Properties procedure filter has had sufficient time to provide good aerodynamic data), or if this is the first pass (guidance initialization flag is set to one), then execute steps *a* and *b*. Otherwise, proceed directly to step *4*.

   a. Set the guidance initialization flag to zero to ensure that the variables are not initialized again:

   $$F_{ig} = 0$$

   b. If the guidance pass number is zero, then get desired bank angle from the Predictor-Corrector Sequencer procedure. A guidance pass number equal to zero indicates either the first pass or that you have reached the limit of guidance passes without updating the bank angle. The aerocapture guidance algorithm is intended to be run at a frequency of 1 Hz to give a bank angle command once per second; however, bank angle updates need not be made every second. The guidance pass limit variable enables the user to determine the frequency at which to update the bank angle.

4. Execute the Lateral Guidance procedure to get the sign of the bank angle.

5. Execute the Bank Angle Command Incorporation procedure to get the commanded bank angle.

6. Increment the guidance pass number:

$$GP = GP + 1 \tag{A.1}$$

1. If you have reached the guidance pass limit (a user-defined number which determines the frequency at which to update the bank angle), reset the guidance pass number to zero:

$$GP = 0$$

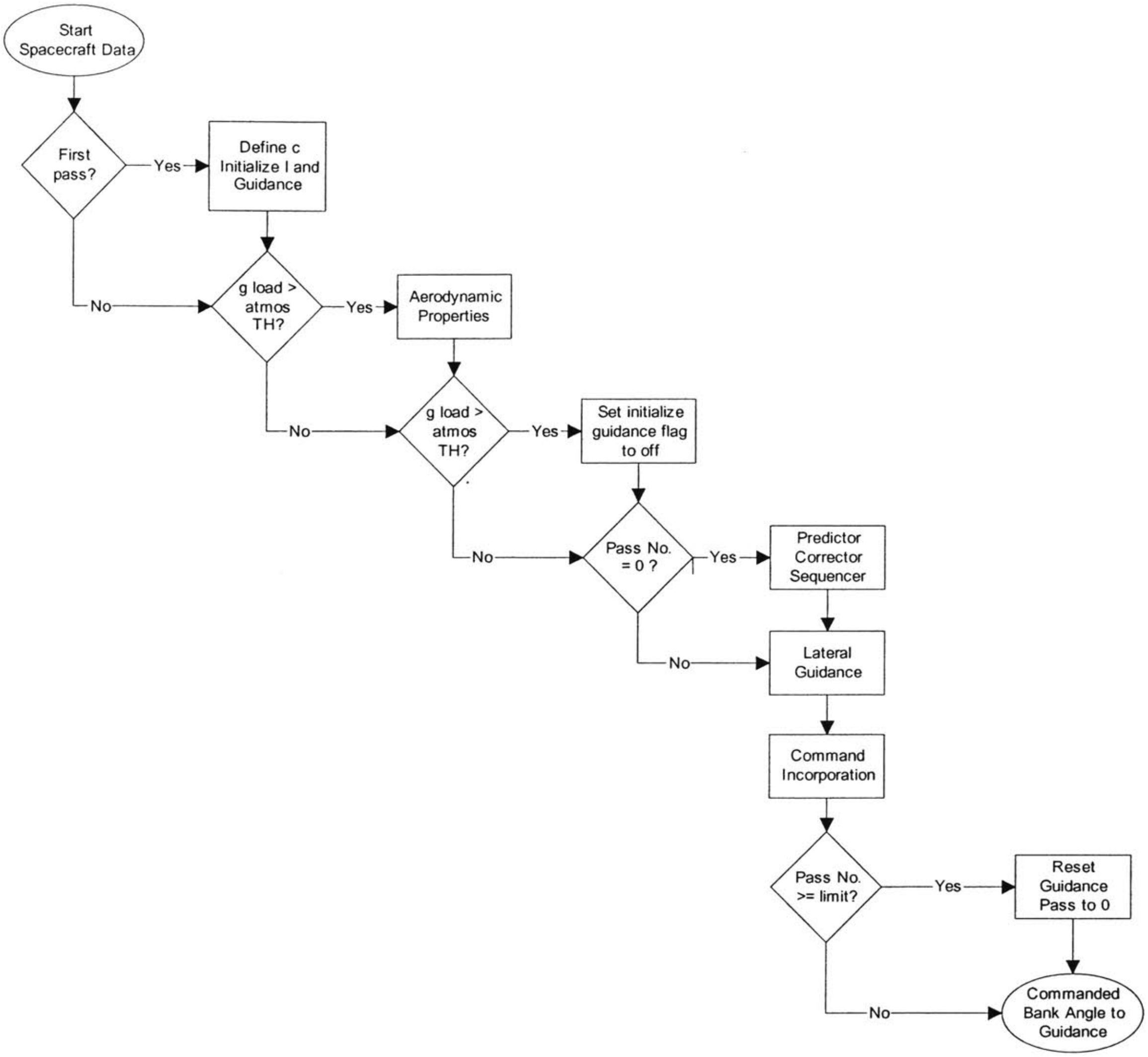


Figure A.1: Aerocapture Guidance Logical Flow: pred_guid.m

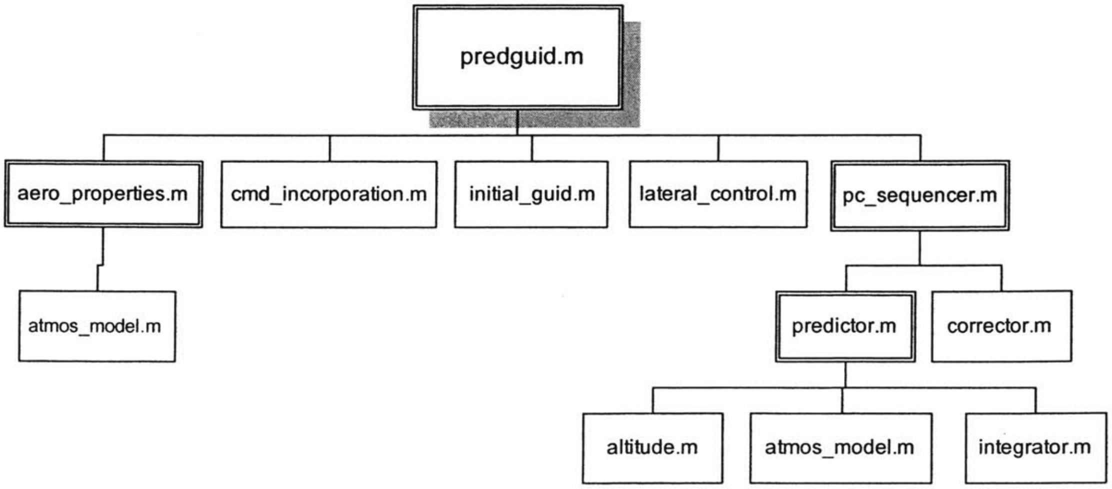


Figure A.2: Aerocapture Guidance Functional Flow: pred_guid.m

Table A-1: Inputs, pred_guid.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
i	A_NAV	$\bar{a}_{nav}$	Vehicle inertial aerodynamic acceleration vector	Ft/s^2	double[3]	Navigation
i	ALT_NAV	h_{nav}	Vehicle altitude	Ft	double	Derived
i	G_LOAD	g	Vehicle aerodynamic load factor	g's	double	Derived
i	IYD	$\bar{I}_{YD}$	Unit normal to the desired orbit plane	(-)	double[3]	Derived
i	Q_B_TO_I	Q_{BI}	Vehicle body-to-inertial attitude quaternion	(-)	double[4]	Navigation
i	R_NAV	$\bar{R}_{nav}$	Vehicle inertial position vector	ft	double[3]	Navigation
i	V_NAV	$\bar{V}_{nav}$	Vehicle inertial velocity vector	ft/s	double[3]	Navigation
i	V_NAV_MAG	$\lvert\bar{V}_{nav}\rvert$	Vehicle inertial velocity vector magnitude	ft/s	double	Derived
i	V_REL_MAG	$\lvert\bar{V}_{rel}\rvert$	Vehicle relative velocity vector magnitude	ft/s	double	Derived
i	V_REL_NAV	$\bar{V}_{rel}$	Vehicle relative velocity vector	ft/s	double[3]	Derived

Table A-2: Outputs, pred_guid.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Destination
	PHI_CMD	ϕ_{cmd}	Commanded bank angle	deg	double	Guidance System

Table A-3: Constants, pred_guid.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
c	AERO_PROPERTY_GLOAD	g_{aero}	Minimum g load required to update aerodynamic properties	G's	double	0.05
c	G_RUN_GUIDANCE	g_{guid}	Minimum g load required to run guidance	G's	double	0.075
c	GUID_PASS_LIM	C_{gpl}	Max number of guidance passes allowed without updating bank angle	(-)	double	10

A.2 Guidance Initialization

The initial_guid.m function initializes the local guidance parameters structure, l, on the first guidance pass.

The logical flow of the Guidance Initialization algorithm is illustrated in Figure A.3.

The data inputs to initial_guid.m are described in Table A-4.

The local guidance parameters used or updated in initial_guid.m are described in Table A-5.

The constants used in initial_guid.m are described in Table A-6.

1. Set the maximum and minimum cosine of bank angle values (corresponding to the minimum and maximum bank angles, respectively) equal to the user-defined constant values

$$\cos(\phi)\big|_{\min} = \cos(\phi_{\max} C_{D\rightarrow R})$$
$$\cos(\phi)\big|_{\max} = \cos(\phi_{\min} C_{D\rightarrow R})$$

2. Initialize the bank angle sign parameters by executing steps *a* through *c*:

   a. Calculate the sign of the bank angle using the currently velocity vector and unit normal to the desired plane

   $$sign(\phi) = sign(\overline{V}_{nav} \times \overline{I}_{YD}) \tag{A.2}$$

   b. Initialize the commanded bank angle using this sign of bank and the user-defined initial value

   $$\phi_{cmd} = sign(\phi)|\phi_{EI}| \tag{A.3}$$

   c. Initialize the previous sign of bank value to the current sign of bank value

   $$sign(\phi)\big|_{past} = sign(\phi)$$

3. Initialize the cosine of the desired bank angle to the user-defined initial value

   $$\cos(\phi)_{des} = \cos(\phi)_{des_{init}}$$

4. Initialize the roll reversal parameters by executing steps *a* through *c*:

   a. Calculate the corridor slope. Essentially we are creating an equation of a line where the corridor slope and the minimum corridor out of plane velocity define the slope and the y intercept, respectively. This is a user-defined 'acceptability' criterion to determine if, based on the out of plane velocity, the spacecraft is within the desired orbital plane.

   $$S_{corr} = \frac{C_{\max} - C_{\min}}{C_{V_{\max}} - C_{V_{\min}}} \tag{A.4}$$

   b. Initialize the lift down reversal flag to one (Feature not currently used)

   $$F_{LDR} = 1$$

   c. Initialize the over/under bank angle to the predicted angle (Feature not currently used)

   $$\phi_{OU} = \phi_{OU\,pred}$$

5. Initialize the lift down phase variables by executing steps *a* and *b*:

   a. Initialize the lift down modeling flag. A flag value of zero deactivates lift down modeling while a value of one activates it.

   $$F_{MLD} = F_{MLD_{init}}$$

b. Set the initial sine and cosine values of the lift down bank angle to the user-defined initial values

$$\cos(\phi_{LD}) = \cos\left(\phi_{LD_{init}} C_{D \to R}\right)$$
$$\sin(\phi_{LD}) = \sin\left(\phi_{LD_{init}} C_{D \to R}\right)$$

6. Initialize aerodynamic parameters to user-defined initial values:

a. Drag coefficient

$$C_{D_{est}} = C_{D_{est_{init}}}$$

b. Lift to drag ratio

$$\left.\frac{L}{D}\right)_{est} = \left.\frac{L}{D}\right)_{est_{init}}$$

c. Density gain

$$K_{\rho_{est}} = K_{\rho_{est_{init}}}$$

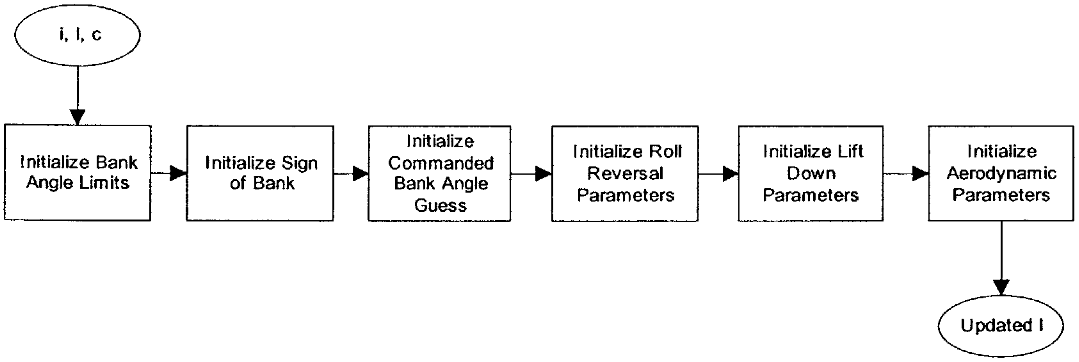

i, I, c
Initialize Bank Angle Limits
Initialize Sign of Bank
Initialize Commanded Bank Angle Guess
Initialize Roll Reversal Parameters
Initialize Lift Down Parameters
Initialize Aerodynamic Parameters
Updated I

Figure A.3: Guidance Initialization Logical Flow: initial_guid.m

Table A-4: Inputs, initial_guid.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
i	IYD	$\bar{I}_{YD}$	Unit normal to the desired orbit plane	-	double[3]	Derived
i	V_NAV	$\bar{V}_{nav}$	Vehicle inertial velocity vector	ft/s	double[3]	Navigation

Table A-5: Local Guidance Parameters, initial_guid.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Use
L	COS_PHI_MIN	$\cos(\phi)\vert_{\min}$	Cosine of the maximum bank angle	(-)	double	Initialize d
1	COS_PHI_MAX	$\cos(\phi)\vert_{\max}$	Cosine of the minimum bank angle	(-)	double	Initialize d
1	SIGN_OF_BANK	$sign(\phi)$	Sign of current bank angle guess	(-)	double	Initialize d
1	PHI_CMD	ϕ_{cmd}	Commanded bank angle	deg	double	Initialize d
1	SIGN_OF_ BANK_PAST	$sign(\phi)\vert_{past}$	Sign of previous bank angle guess	(-)	double	Initialize d
1	COS_PHI_DES	$\cos(\phi_{des})$	Cosine of the desired bank angle	(-)	double	Initialize d
1	CORRIDOR_ SLOPE	S_{corr}	Aerocapture corridor slope	(-)	double	Initialize d
1	LIFT_DOWN_ REVERSAL	F_{LDR}	Flag indicates need for a lift down reversal	(-)	double	Initialize d
1	DPHI_ OVER_UNDER	ϕ_{OU}	Bank angle which gives direction vehicle takes to reverse direction	deg	double	Initialize d

			for a sign change			
1	MODEL_ LIFT_DOWN	F_{MLD}	Flag to activate/deactivate lift down modeling	(-)	double	Initialized
1	CPHI_ LIFT_DOWN	$\cos(\phi_{LD})$	Cosine of the lift down bank angle	(-)	double	Initialized
1	SPHI_ LIFT_DOWN	$\sin(\phi_{LD})$	Sine of the lift down bank angle	(-)	double	Initialized
1.AERO	CD_EST	$C_{D_{est}}$	Drag coefficient estimate	(-)	double	Initialized
1.AERO	L_OVER_D_EST	$\left.\frac{L}{D}\right)_{est}$	Lift to drag ratio estimate	(-)	double	Initialized
1.AERO	K_RHO_EST	$K_{\rho_{est}}$	Density gain estimate	(-)	double	Initialized

Table A-6: Constants, initial_guid.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
c	BANK_MAX	$\phi_{\max}$	Maximum bank angle	deg	double	165.0
c	BANK_MIN	$\phi_{\min}$	Minimum bank angle	deg	double	15.0
c	DEG_TO_RAD	$C_{D\to R}$	Degrees to radians conversion factor	rad/deg	double	$\frac{\pi}{180.0}$
c	PHI_EI	ϕ_{EI}	Initial bank angle guess at entry interface	deg	double	90.0
c	COS_PHI_DES_INIT	$\cos(\phi_{des})_{init}$	Initial cosine of the desired bank angle	(-)	double	0.0
c	CORRIDOR_MAX	$C_{\max}$	Maximum lateral out	ft/s	double	2000.0

			of plane velocity			
c	CORRIDOR_MIN	$C_{\min}$	Minimum lateral out of plane velocity	ft/s	double	15.0
c	CORRIDOR_V_MAX	$C_{V_{\max}}$	Maximum forward velocity for corridor slope	ft/s	double	31000.0
c	CORRIDOR_V_MIN	$C_{V_{\min}}$	Minimum forward velocity for corridor slope	ft/s	double	26500.0
c	DPHI_OVER UNDER_PRED	$\phi_{OU_{pred}}$	Initial bank angle which gives direction to reverse	deg	double	150.0
c	MODEL_LIFT_DOWN	F_{LD}	Initial model lift down flag	(-)	double	0
c	PHI_LIFT_DOWN	ϕ_{LD}	Lift down bank angle	deg	double	115.0
c	CD_EST_INITIAL	$C_{D_{est_{init}}}$	Initial drag coefficient estimate	(-)	double	1.4286
c	L_OVER_D_INITIAL	$\left.\frac{L}{D}\right)_{est_{init}}$	Initial L/D coefficient estimate	(-)	double	0.25
c	K_RHO_EST_INITIAL	$K_{\rho_{est_{init}}}$	Initial density gain coefficient estimate	(-)	double	1.0

A.3 Aerodynamic Properties

The aero_properties.m function updates the aerodynamic properties.

The logical flow of the Aerodynamic Properties algorithm is illustrated in Figure A.4.

The data inputs to aero_properties.m are described in Table A-7.

The local guidance parameters used or updated in aero_properties.m are described in Table A-8.

The constants used in aero_properties.m are described in Table A-9.

1. Compute various unit vectors

$$\hat{x}_{VR} = \frac{\overline{V}_{nav}}{\left|\overline{V}_{nav}\right|} \tag{A.5}$$

$$\hat{y}_{BI} = Q_{BI}\hat{y}_{body} \tag{A.6}$$

$$\hat{z}_{DCL} = \frac{\hat{y}_{BI} \times \hat{x}_{VR}}{\left|\hat{y}_{BI} \times \hat{x}_{VR}\right|} \tag{A.7}$$

Notice that assuming zero sideslip, this is essentially a velocity frame: $\hat{x}$ is out the 'nose' in the velocity direction (it differs from the velocity direction by the angle of attack), $\hat{y}$ is out the 'right wing', and $\hat{z}$ completes the right hand coordinate system and is generally away from earth (positive up).

2. Compute aerodynamic accelerations. Drag is in the negative velocity direction and lift is perpendicular to the velocity direction (positive away from earth).

$$a_{drag} = -\overline{a}_{nsb}^{T}\hat{x}_{VR} \tag{A.8}$$

$$a_{lift} = \overline{a}_{nsb}^{T}\hat{z}_{DCL} \tag{A.9}$$

3. If the acceleration due to drag is greater than zero, execute steps *a* through *g*. Otherwise, exit the procedure.

   a. Convert velocity in the inertial frame to velocity in the body frame

$$\left.\overline{V}_{rel}\right|_{b} = Q_{BI}^{T}\overline{V}_{rel} \tag{A.10}$$

   b. Use geometry to calculate the angle of attack

$$\alpha = \tan^{-1}\left(\frac{V_{rel}(3)|_b}{V_{rel}(1)|_b}\right) \tag{A.11}$$

c. Compute the estimated drag coefficient

$$C_{d_{est}} = C_{d_0} + \left(C_{d_\alpha} + C_{d_{\alpha 2}}\alpha\right)\alpha \tag{A.12}$$

d. Compute the standard atmospheric density at the current altitude using the Atmospheric Model procedure.

e. Estimate the atmospheric density at the current altitude from the measured drag

$$\rho_{meas} = \frac{2m_{EI}a_{drag}}{C_{d_{est}}S_{ref}\left|\overline{V}_{rel}\right|^2} \tag{A.13}$$

f. Compute and filter the density bias estimate by executing steps *i* through *iii*:

i. Calculate the density bias estimate

$$K_\rho = \frac{\rho_{meas}}{\rho_{std}} \tag{A.14}$$

ii. Ensure the density bias estimate is within the minimum and maximum values by taking the median of $K_\rho, K_{\rho_{\min}}$, and $K_{\rho_{\max}}$.

iii. Filter the density bias to get the new estimate

$$K_{\rho_{est}} = \left(1 - K_{\rho_{fg}}\right)K_{\rho_{est}} + K_{\rho_{fg}}K_\rho \tag{A.15}$$

g. Compute and filter the lift to drag ratio estimate by executing steps *i* through *iii*:

i. Calculate the lift to drag ratio estimate

$$\frac{L}{D} = \frac{a_{lift}}{a_{drag}} \tag{A.16}$$

ii. Ensure the density bias estimate is within the minimum and maximum values by taking the median of $\frac{L}{D}, \left.\frac{L}{D}\right)_{\min}$, and $\left.\frac{L}{D}\right)_{\max}$.

iii. Filter the lift to drag ratio to get the new estimate

$$\left.\frac{L}{D}\right)_{est} = \left(1 - K_{L/D_{fg}}\right)\left.\frac{L}{D}\right)_{est} + K_{L/D_{fg}}\frac{L}{D} \tag{A.18}$$

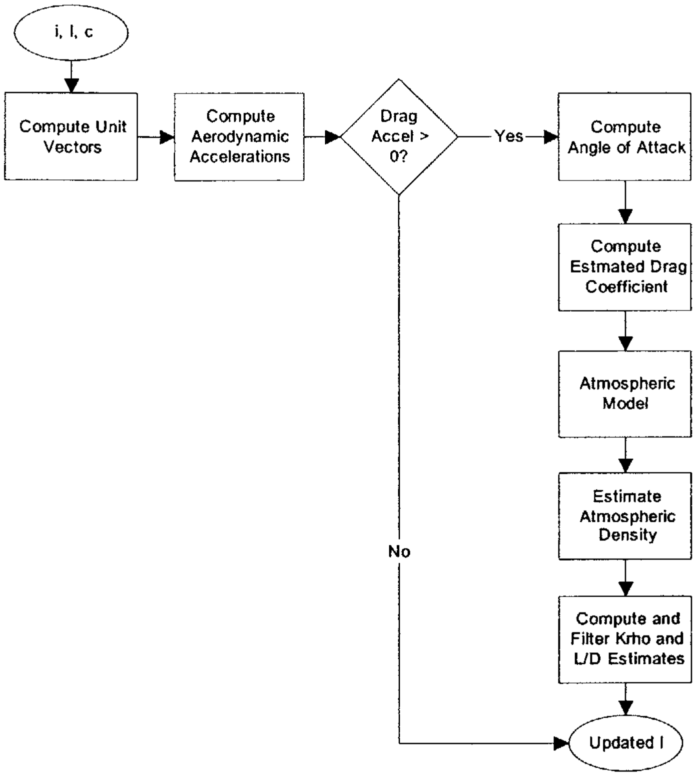

i, I, c
Compute Unit Vectors
Compute Aerodynamic Accelerations
Drag Accel > 0?
Yes
No
Compute Angle of Attack
Compute Estmated Drag Coefficient
Atmospheric Model
Estimate Atmospheric Density
Compute and Filter Krho and L/D Estimates
Updated I

Figure A.4: Aerodynamic Properties Logical Flow: aero_properties.m

Table A-7: Inputs, aero_properties.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
i	Q_B_TO_I	Q_{BI}	Vehicle body-to-inertial quaternion	(-)	double[4]	Navigation
i	A_NAV	a_{nav}	Vehicle inertial aerodynamic acceleration vector	ft/s^2	double[3]	Navigation
i	V_REL_MAG	$\left\|\overline{V}_{rel}\right\|$	Vehicle relative velocity vector magnitude	ft/s	double	Derived
i	V_REL_NAV	$\overline{V}_{rel}$	Vehicle relative velocity vector	ft/s	double[3]	Derived

Table A-8: Local Guidance Parameters, aero_properties.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Use
l.AERO	CD_EST	$C_{D_{est}}$	Drag coefficient estimate	(-)	double	Updated
l.AERO	L_OVER_D_EST	$\left.\frac{L}{D}\right)_{est}$	Lift to drag ratio estimate	(-)	double	Updated
l.AERO	K_RHO_EST	$K_{\rho_{est}}$	Density gain estimate	(-)	double	Updated

Table A-9: Constants, aero_properties.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
c	Y_BODY	$\hat{y}_B$	Unit vector out 'right wing' of vehicle	(-)	double	$\begin{bmatrix}0\\1\\0\end{bmatrix}$
c	RAD_TO_DEG	$C_{R\to D}$	Radians to degrees conversion	deg/rad	double	$\frac{180.0}{\pi}$
c	CD_ZERO	C_{d_0}	Parasite drag coefficient	(-)	double	1.4286
c	CD_ALPHA	C_{d_α}	Slope of the C_d vs. α graph	(-)	double	0
c	CD_ALPHA_SQ	$C_{d_{\alpha^2}}$	Slope of the C_d vs. α^2 graph	(-)	double	0
c	MASS_EI	m_{EI}	Vehicle mass at EI	slugs	double	15.4783
c	S_REF	S_{ref}	Ref area	ft^2	double	12.163
c	K_RHO_MIN	$K_{\rho_{min}}$	Min density bias	(-)	double	0.5

c	K_RHO_MAX	$K_{\rho_{\max}}$	Max density bias est.	(-)	double	2.0
c	K_RHO_FILTER_GAIN	$K_{\rho_{fg}}$	Filter gain for density bias est.	(-)	double	0.05
c	L_OVER_D_MIN	$\left.\frac{L}{D}\right)_{\min}$	Min L/D estimate	(-)	double	0.56
c	L_OVER_D_MAX	$\left.\frac{L}{D}\right)_{\max}$	Max L/D estimate	(-)	double	0.14
c	L_OVER_D_FILTER_GAIN	$K_{L/D_{fg}}$	Filter gain for L/D estimate	(-)	double	0.2

A.4 Predictor-Corrector Sequencer

The pc_sequencer.m function alternately runs the Predictor and Corrector functions to determine the desired bank angle.

The logical flow of the Predictor-Corrector Sequencer algorithm is illustrated in Figure A.5.

The data inputs to pc_sequencer.m are described in Table A-10.

The local guidance parameters used or updated in pc_sequencer.m are described in Table A-11.

The constants used in pc_sequencer.m are described in Table A-12.

1. The Predictor-Corrector Sequencer is executed at a specific (user-defined) frequency in the Aerocapture Executive. Each time it is executed, it begins counting acceptable tries from zero. The following list of counting and storage variables must be reinitialized before each run of the Predictor-Corrector Sequencer:

$$N_{high} = 0$$

$$N_{low} = 0$$

$$N_{capt} = 0$$

$$N_{good} = 0$$

$$\cos_{capt} = \infty$$

$$\cos(\phi_{try}) = \infty$$

$$\cos_{bracket} = \infty$$

$$\cos_{extrap} = \infty$$

$$a_{extrap} = \infty$$

$$a_{bracket} = \infty$$

$$\Delta a = 0$$

$$F_b = 0$$

$$\phi_{try_{last}} = 0$$

2. If velocity magnitude is less than the velocity to model lift down, terminate lift down modeling by resetting the model lift down flag to zero:

$$F_{MLD} = 0$$

3. Select the apogee criteria by determining if the velocity magnitude is greater than the velocity for the tighter apogee criteria. If the velocity is still 'fast' the program uses a looser requirement to hit the target apogee because at faster velocities the profile is more sensitive. As the velocity decreases, the apogee miss must be less in order to hone in on the target.

4. Calculate the desired bank angle by executing steps *a* through *g* a user-defined number of times:

   a. Calculate the bank angle to try and the cosine of the bank angle to try using the Corrector procedure.

   b. Calculated the predicted apogee using the Predictor procedure.

   c. Compute the apogee miss

$$\Delta a = a_{pred} - a_{t\arg et} \qquad \text{(A.19)}$$

d. If the apogee miss is less than the apogee correct criteria, then the try was acceptable. Set the cosine of the desired bank angle equal to the cosine of the bank angle tried and exit the procedure

$$\cos(\phi_{des}) = \cos(\phi)_{try}$$

e. If the cosine of the bank angle tried is greater than the maximum cosine of bank angle allowed, and the apogee miss is positive, (translated, the spacecraft is asking for a more shallow bank angle than it is capable of and still missed high) full lift down is required. Set the cosine of the desired bank angle equal to the maximum cosine of the bank angle and exit the procedure

$$\cos(\phi_{des}) = \cos(\phi)_{\max}$$

f. If the cosine of the bank angle tried is less than the minimum cosine of bank angle allowed, and the apogee miss is negative, (translated, the spacecraft is asking for a steeper bank than it is capable of and still missed low) full lift up is required. Set the cosine of the desired bank angle equal to the minimum cosine of the bank angle and exit the procedure

$$\cos(\phi_{des}) = \cos(\phi)_{\min}$$

g. Determine the nature of the solution (i.e. high, low, captured, etc.) and calculate the cosine of the desired bank angle by executing steps *i* through *vi*:

i. If the prediction captured (determined in the Predictor procedure), increment the number of captured solutions by one and set the capture cosine value equal to the cosine of the bank angle tried and proceed to step *iii*

$$N_{capt} = N_{capt} + 1 \qquad \text{(A.20)}$$

$$\cos_{capt} = \cos(\phi_{try})$$

ii. Otherwise, save the Predictor solution as a 'good' solution by executing steps *1* through *4* and then proceed to step *iii*:

1. Increment the number of good solutions

$$N_{good} = N_{good} + 1 \qquad \text{(A.21)}$$

2. Rotate the new good solution into the extrapolation variables

$$\cos_{extrap}(2) = \cos_{extrap}(1)$$
$$\cos_{extrap}(1) = \cos(\phi_{try})$$

$$a_{extrap}(2) = a_{extrap}(1)$$
$$a_{extrap}(1) = a_{pred}$$

3. If the predicted apogee is higher than the target apogee, the solution is high. Increment the number of high solutions using Eq. A.22, update the high bracket terms, and proceed to step *iii*

$$N_{high} = N_{high} + 1 \tag{A.22}$$
$$\cos_{bracket}(1) = \cos(\phi_{try})$$
$$a_{bracket}(1) = a_{pred}$$

4. Otherwise, the predicted apogee is lower than the target apogee, and the solution is low. Increment the number of low solutions using Eq. A.23, update the low bracket terms and proceed to step *iii*

$$N_{low} = N_{low} + 1 \tag{A.23}$$
$$\cos_{bracket}(2) = \cos(\phi_{try})$$
$$a_{bracket}(2) = a_{pred}$$

iii. If there is a high and a low (including captured) solution, we have bracketed the target apogee. Set the bracketed flag to 1:

$$F_b = 1$$

otherwise, proceed to step *iv*.

iv. If we have bracketed the target apogee execute the following steps and then proceed to step *v*

1. Calculate the delta bank angle from the last try

$$\Delta\phi = \left|\phi_{try} - \phi_{try_{last}}\right| \tag{A.24}$$

2. If the delta bank angle is less than the minimum allowable delta bank angle, set the cosine of the desired bank angle to the average of the two bank angles

$$\cos(\phi_{des}) = \cos\left(\frac{\left(\phi_{try} + \phi_{try_{last}}\right)C_{D\to R}}{2}\right) \tag{A.25}$$

otherwise, proceed directly to step *v*.

v. Set the previous bank angle tried to the current bank angle

$$\phi_{try_{last}} = \phi_{try}$$

vi. If we have reached the maximum number of runs to update the bank angle, limit the predictions by executing steps *1* and *2* and then exit the procedure

1. Correct the bank angle to try once more without prediction by running the Corrector procedure.
2. Set the cosine of the desired bank angle equal to the cosine of this most recently updated bank angle

$$\cos(\phi_{des}) = \cos(\phi)_{\min}$$

otherwise, exit the procedure.

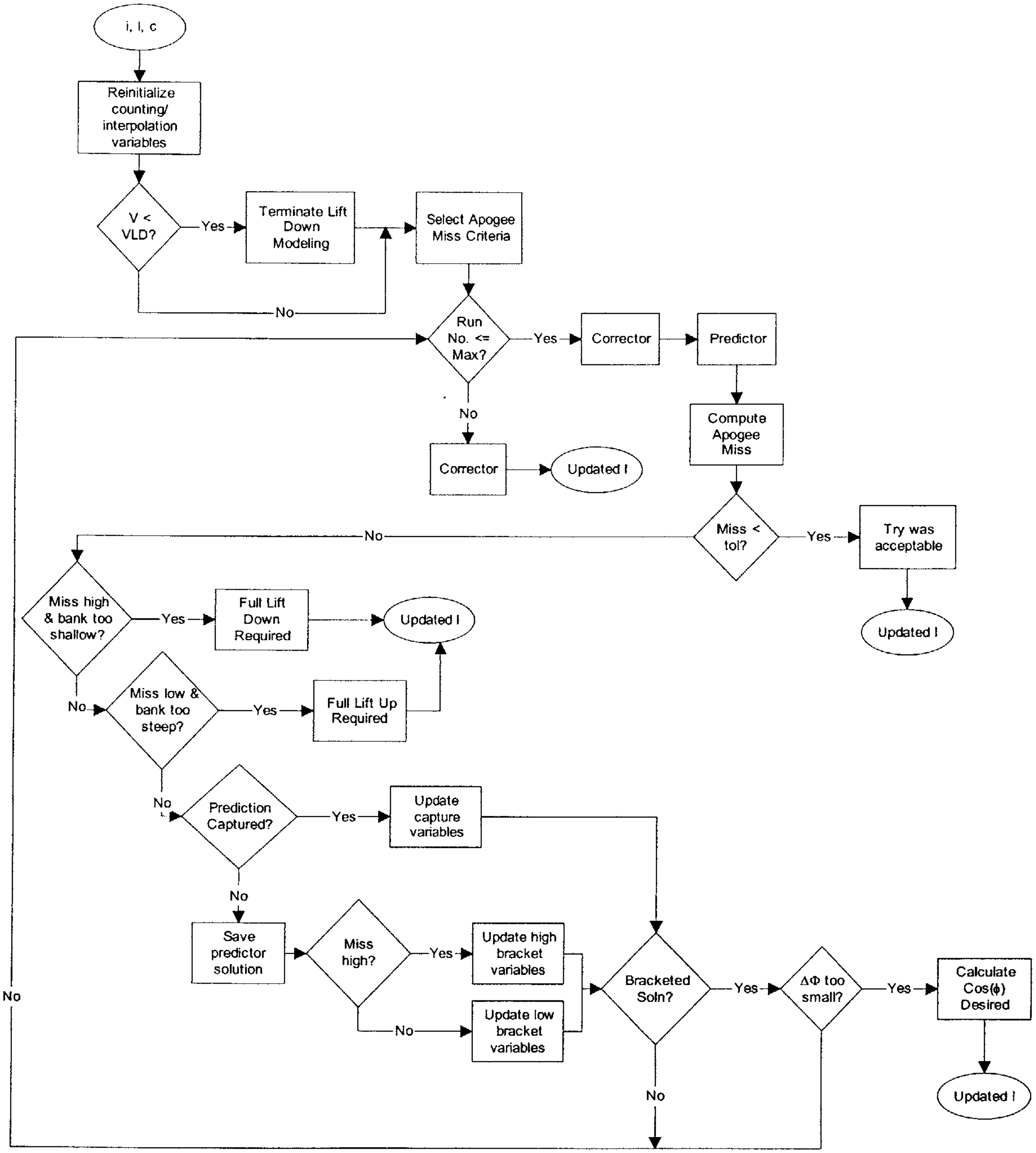


Figure A.5: Predictor-Corrector Sequencer Logical Flow: pc_sequencer.m

Table A-10: Inputs, pc_sequencer.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
i	V_NAV_MAG	$\lvert\bar{V}_{nav}\rvert$	Vehicle inertial velocity vector magnitude	ft/s	double[3]	Derive d

Table A-11: Local Guidance Parameters, pc_sequencer.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Use
1	COS_PHI_MIN	$\cos(\phi)\rvert_{\min}$	Cosine of maximum bank angle	(-)	double	Used
1	COS_PHI_MAX	$\cos(\phi)\rvert_{\max}$	Cosine of minimum bank angle	(-)	double	Used
1	COS_PHI_DES	$\cos(\phi)_{des}$	Cosine of the desired bank angle	(-)	double	Updated
1	MODEL_LIFT_DOWN	F_{MLD}	Flag which activates lift down modeling	(-)	double	Updated

Table A-12: Constants, pc_sequencer.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
c	DEG_TO_RAD	$C_{D\rightarrow R}$	Degrees to radians conversion factor	Rad/deg	double	$\frac{\pi}{180.0}$
c	VI_MODEL_LIFT_DOWN	V_{LD}	Velocity switch for lift down modeling	ft/s	double	27900.0
c	VI_LOOSE_APOGEE	V_{LA}	Switchover for apogee tolerances	ft/s	double	30000.0
c	APOGEE_EPSILON1	$a_{\varepsilon 1}$	Apogee tolerance limit for fast velocities	nm	double	25.0
c	APOGEE_EPSILON2	$a_{\varepsilon 2}$	Apogee tolerance limit for slower velocities	nm	double	1.0
c	APOGEE_TARGET	$a_{t\arg et}$	Target apogee	nm	double	216.0
c	MAX_NUMBER_RUNS		Maximum number PC sequences (or bank corrections) per run	(-)	double	4
c	INFINITY	∞	Infinity	(-)	double	99999999

A.5 Corrector

The corrector.m function generates a new bank angle to try in Predictor.

The logical flow of the Corrector algorithm is illustrated in Figure A.6.

The data inputs to corrector.m are described in Table A-13.

The data outputs of corrector.m are described in Table A-14.

The local guidance parameters used or updated in corrector.m are described in Table A-15.

The constants used in corrector.m are described in Table A-16.

1. Determine the method used to make the new bank angle guess:
    a. If this is the first run, use Method 1.
    b. Otherwise, if we have bracketed the solution and
        i. The number of low solutions is not zero, we have a high solution and a low solution, use Method 2.
        ii. There is a high solution and a capture, or more than one high solution, use Method 3.
    c. Otherwise, if we have not bracketed the solution and
        i. There is only one good solution, use Method 5.
        ii. There are at least two good solutions, use Method 6.
        iii. There are no good solutions yet, use Method 4.
2. Execute the applicable steps for each method to determine the new bank angle guess:
    a. Method 1: Run the last guess from the previous guidance cycle

       $$\cos(\phi_{try}) = \cos(\phi_{des})$$

    b. Method 2: Interpolate a high guess and a low guess to the target apogee
    c. Method 3: Interpolate a high guess and a capture or interpolate two high guesses

i. If there is only one high solution, calculate the new bank angle guess from the high solution and the captured solution

$$\cos(\phi_{try}) = \cos_{bracket}(1) + (\cos_{capt} - \cos_{bracket}(1))C_{LPC} \tag{A.26}$$

ii. Otherwise, interpolate a new bank angle to try from the two good solutions. If this new bank angle to try is greater than or equal to the current cosine capture value, calculate the new bank angle guess from the high solution and the captured solution

$$\cos(\phi_{try}) = \cos_{bracket}(1) + (\cos_{capt} - \cos_{bracket}(1))C_{LPC} \tag{A.27}$$

d. Method 4: March out of the capture region

$$\cos(\phi_{try}) = \cos_{capt} - C_{LIC} \tag{A.28}$$

e. Method 5: Extrapolate one good guess using a stored sensitivity

i. Calculate sensitivity if the velocity magnitude is less than the current break value (i)

$$S = C_{ss}(i)|\overline{V}_{nav}| + C_{si}(i) \tag{A.29}$$

ii. Calculate the cosine of the bank angle to try

$$\cos(\phi_{try}) = \cos(\phi_{try})_{last} + \frac{\Delta a}{S} \tag{A.30}$$

The sensitivity-bank angle loop can run up to 6 times (user defined). The final bank angle guess is either the sixth calculation or the (i-1)th calculation before the sensitivity break value (i) is less than the velocity magnitude.

f. Method 6: Extrapolate two high guesses or two low guesses to the target apogee

3. Ensure the cosine of the bank angle guess is within the minimum and maximum cosine of bank angle values by taking the median of $\cos(\phi_{try})$, $\cos(\phi)|_{\min}$, and $\cos(\phi)|_{\max}$.

4. Calculate the bank angle to try

$$\phi_{try} = \cos^{-1}(C_{R\to D}\cos(\phi_{try})) \tag{A.31}$$

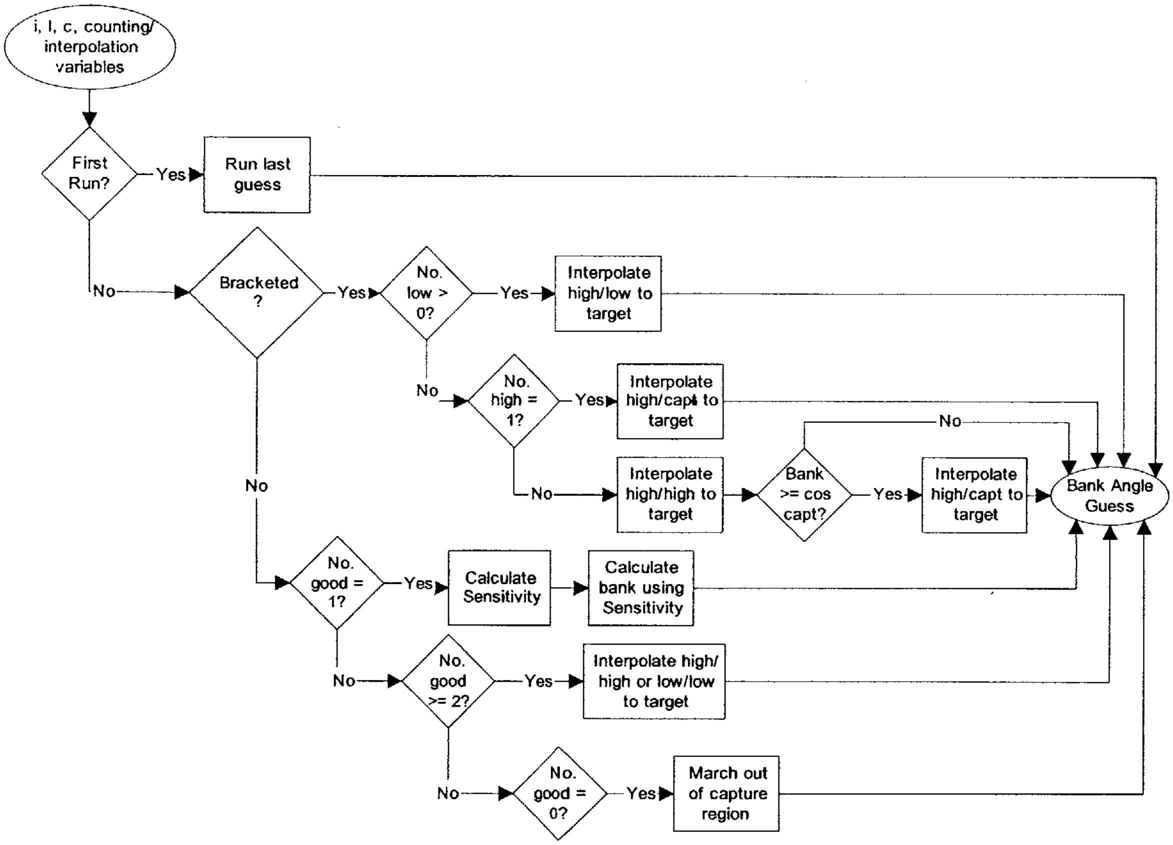

i, l, c, counting/ interpolation variables
First Run?
Yes
Run last guess
No
Bracketed ?
Yes
No. low > 0?
Yes
Interpolate high/low to target
No
No. high = 1?
Yes
Interpolate high/capt to target
No
Interpolate high/high to target
Bank >= cos capt?
Yes
Interpolate high/capt to target
No
Bank Angle Guess
No
No. good = 1?
Yes
Calculate Sensitivity
Calculate bank using Sensitivity
No
No. good >= 2?
Yes
Interpolate high/ high or low/low to target
No
No. good = 0?
Yes
March out of capture region

Figure A.6: Corrector Logical Flow: corrector.m

Table A-13: Inputs, corrector.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
i	V_NAV_MAG	$\left\|\bar{V}_{nav}\right\|$	Vehicle inertial velocity magnitude	ft/s	double[3]	Derived
	COS_PHI_TRY	$\cos(\phi_{try})$	Cosine of bank angle guess	(-)	double	pc_sequencer
	BRACKETED	F_b	Flag which indicates target bracketed	(-)	double	pc_sequencer
	COS_BRACKET	$\cos_{bracket}$	Cosine of angles that	(-)	double[2]	pc_sequencer

			bracket the target			
	APOGEE_BRACKET	$a_{bracket}$	Apogees which bracket the target	nm	double[2]	pc_sequencer
	COS_CAPT	$\cos_{capt}$	Cosine of the last bank that caused capture	(-)	double	pc_sequencer
	COS_EXTRAPOLATE	$\cos_{extrap}$	Cosine of angles for interpolation	(-)	double[2]	pc_sequencer
	APOGEE_EXTRAPOLATE	a_{extrap}	Apogees of solutions for interpolation	nm	double[2]	pc_sequencer
	NUMBER_GOOD	N_{good}	No. good solutions	(-)	double	pc_sequencer
	NUMBER_HIGH	N_{high}	No. high solutions	(-)	double	pc_sequencer
	NUMBER_LOW	N_{low}	No. low solutions	(-)	double	pc_sequencer
	DELTA_APOGEE	Δa	Apogee error	nm	double	pc_sequencer
	RUN_NUMBER	N_{run}	Run number	(-)	double	pc_sequencer

Table A-14: Outputs, Corrector.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Destination
	PHI_TRY	ϕ_{try}	Bank angle guess	deg	double	predictor
	COS_PHI_TRY	$\cos(\phi_{try})$	Cosine of the bank angle guess	(-)	double	predictor

Table A-15: Local Guidance Parameters, corrector.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Use
1	COS_PHI_MIN	$\cos(\phi)\|_{\min}$	Cosine of the maximum allowable bank angle	(-)	double	Used
1	COS_PHI_MAX	$\cos(\phi)\|_{\max}$	Cosine of the minimum allowable bank angle	(-)	double	Used
1	COS_PHI_DES	$\cos(\phi_{des})$	Cosine of the desired bank angle	(-)	double	Updated

Table A-16: Constants, corrector.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
C	RAD_TO_DEG	$C_{R\to D}$	Radians to degrees conversion factor	deg/rad	double	$\frac{180.0}{\pi}$
C	LIFT_PERCENT_CAPTURE	C_{LPC}			double	0.50
C	LIFT_INC_CAPTURE	C_{LIC}	Cosine amount used to march out of capture region		double	-0.10
C	SEN_BREAK_POINT	C_{sbp}	Vector determines which stored sensitivity equations to use		double[6]	$\begin{bmatrix}34000.0\\32000.0\\30000.0\\27500.0\\26000.0\\0.0\end{bmatrix}$
C	SEN_INTER	C_{ss}	y intercept of the stored sensitivity equation		double[6]	$\begin{bmatrix}-9066\\-7032\\-3489\\-1122\\-147\end{bmatrix}$
C	SEN_SLOPE	C_{si}	Slope of the stored sensitivity equation		double[5]	$\begin{bmatrix}0.0\\0.0\\0.0\\0.0\\0.0\\0.0\end{bmatrix}$

A.6 Predictor

The predictor.m function calculates the predicted apogee assuming the current bank angle guess is constant for the remainder of the trajectory. It also determines if the prediction captures.

The logical flow of the Predictor algorithm is illustrated in Figure A.7.

The data inputs to predictor.m are described in Table A-17.

The data outputs of predictor.m are described in Table A-18.

The local guidance parameters used or updated in predictor.m are described in Table A-19.

The constants used in predictor.m are described in Table A-20.

1. Initialize variables:

   a. Predictor state vector

$$\overline{R}_{nav} = \overline{R}_{pred}$$

$$\overline{V}_{nav} = \overline{V}_{pred}$$

$$\left|\overline{R}_{pred}\right|^2 = \overline{R}_{pred}^T \overline{R}_{pred} \tag{A.32}$$

$$\left|\overline{R}_{pred}\right| = \sqrt{\left|\overline{R}_{pred}\right|^2} \tag{A.33}$$

$$\left|\overline{V}_{pred}\right|^2 = \overline{V}_{pred}^T \overline{V}_{pred} \tag{A.34}$$

$$\left|\overline{V}_{pred}\right| = \sqrt{\left|\overline{V}_{pred}\right|^2} \tag{A.35}$$

   b. Calculate predicted altitude (m file function)

   c. Sine and cosine of the predicted bank angle

$$\cos(\phi_{pred}) = \cos(\phi_{try})$$

$$\sin(\phi_{pred}) = \sin(\phi_{try})$$

   d. Lift to drag ratio

$$\left.\frac{L}{D}\right)_{pred} = \left.\frac{L}{D}\right)_{est}$$

e. Prediction captured flag (a value of 1 indicates the prediction captured and a value of zero indicates it did not)

$$F_{PC} = 0$$

f. Sign of the sine of the lift down bank angle

$$\sin(\phi_{LD})\big|_{sign} = sign(\phi)\sin(\phi_{LD}) \tag{A.36}$$

2. Calculate two times the ballistic coefficient

$$2BC = \frac{2m_{EI}}{C_{D_{est}}S} \tag{A.37}$$

3. Execute the predictor loop for a user-defined number of time steps by performing steps *a* through *c*:

   a. Execute the Fourth Order Runge-Kutta Integration Loop four times by performing steps *i* through *vii*:

      i. Calculate the relative velocity

$$\overline{V}_{rel} = \overline{V}_{pred} - \omega_{\oplus}\left[\hat{e}_{pole} \times \overline{R}_{pred}\right] \tag{A.38}$$

$$\left|\overline{V}_{relpred}\right|^2 = \overline{V}_{relpred}^T \tag{A.39}$$

$$\left|\overline{V}_{relpred}\right| = \sqrt{\left|\overline{V}_{relpred}\right|^2} \tag{A.40}$$

      ii. Use the Atmospheric Model procedure to calculate the standard density at the current altitude.

      iii. Compute aerodynamic accelerations

$$a_D = \frac{K_\rho \rho_{std}\left|\overline{V}_{relpred}\right|^2}{BC^2} \tag{A.41}$$

$$a_L = \left.\frac{L}{D}\right)_{pred} a_D \tag{A.42}$$

      iv. Compute the lift vector

$$\bar{I}_{vel} = \frac{\overline{V}_{relpred}}{\left|\overline{V}_{relpred}\right|} \tag{A.43}$$

$$\overline{I}_{lat_c} = \overline{I}_{vel} \times \overline{R}_{pred} \tag{A.44}$$

$$\left|\overline{I}_{lat_c}\right|^2 = \overline{I}^T_{lat_c}\overline{I}_{lat_c} \tag{A.45}$$

$$\left|\overline{I}_{lat_c}\right| = \sqrt{\left|\overline{I}_{lat_c}\right|^2} \tag{A.46}$$

$$\overline{I}_{lat} = \frac{\overline{I}_{lat_c}}{\left|\overline{I}_{lat_c}\right|} \tag{A.47}$$

1. If lift down modeling is active, and the velocity is less than the velocity to model lift down, model lift down, calculate the lift vector using the model lift down cosine and sine values

$$\overline{I}_{lift} = \left(\overline{I}_{lat} \times \overline{I}_{vel}\right)\cos(\phi_{ld}) + \overline{I}_{lat} sign(\sin(\phi_{ld})) \tag{A.48}$$

2. Otherwise, compute direction for bank angle to try

$$\overline{I}_{lift} = \left(\overline{I}_{lat} \times \overline{I}_{vel}\right)\cos\left(\phi_{pred}\right) + \overline{I}_{lat}\sin\left(\phi_{pred}\right) \tag{A.49}$$

$$\overline{a}_{accel} = a_L\overline{I}_{lift} - a_D\overline{I}_{vel} \tag{A.50}$$

v. Compute gravity acceleration with J2

$$\overline{U}_{pred} = \frac{\overline{R}_{pred}}{\left|\overline{R}_{pred}\right|} \tag{A.51}$$

$$\overline{Z}_{pred} = \overline{U}^T_{pred}\overline{P}_{\oplus} \tag{A.52}$$

$$\overline{U}_{pred} = \overline{U}_{pred} + \left(\frac{3}{2}J_2\right)\left(\frac{R_{\oplus}}{\left|\overline{R}_{pred}\right|}\right)^2\left(\left(1 - 5\overline{Z}^2_{pred}\right)\overline{U}_{pred} + 2\overline{Z}_{pred}\overline{P}_{\oplus}\right) \tag{A.53}$$

$$\overline{a}_g = \frac{-\mu_{\oplus}}{\left|\overline{R}_{pred}\right|^2}\overline{U}_{pred} \tag{A.54}$$

vi. Compute total acceleration

$$\overline{a}_{pred} = \overline{a}_{accel} + \overline{a}_g \tag{A.55}$$

vii. Perform Runge-Kutta integration using the Integrator procedure

viii. Compute state parameters

$$\left|\overline{R}_{pred}\right|^2 = \overline{R}^T_{pred}\overline{R}_{pred} \tag{A.56}$$

$$\left|\overline{R}_{pred}\right| = \sqrt{\left|\overline{R}_{pred}\right|^2} \tag{A.57}$$

$$\left|\overline{V}_{pred}\right|^2 = \overline{V}_{pred}^T \overline{V}_{pred} \tag{A.58}$$

$$\left|\overline{V}_{pred}\right| = \sqrt{\left|\overline{V}_{pred}\right|^2} \tag{A.59}$$

b. Check for atmospheric exit by executing the following steps:

   i. Calculate centrifugal velocity and predicted flight path angle

$$\dot{\overline{R}}_{pred} = \frac{\overline{V}_{pred}^T \overline{R}_{pred}}{\left|\overline{R}_{pred}\right|} \tag{A.60}$$

$$\gamma_{pred} = \sin^{-1}\left(\frac{\dot{\overline{R}}_{pred}}{\left|\overline{V}_{pred}\right|}\right) \tag{A.61}$$

   ii. If the altitude is greater than the altitude at which we are still in the atmosphere, and the centrifugal velocity is positive, stop integrating and proceed to step 4.

c. Check for atmospheric capture by executing the following steps:

   i. Calculate centrifugal acceleration

$$\ddot{\overline{R}}_{pred} = \frac{\overline{a}_{pred}^T \overline{R}_{pred} + \left[\left|\overline{V}_{pred}\right| \cos\left(\gamma_{pred}\right)\right]^2}{\left|\overline{R}_{pred}\right|} \tag{A.62}$$

   ii. If centrifugal velocity and centrifugal acceleration are both negative, the prediction has captured

      1. Reset the capture flag to indicate the prediction captured

$$F_{capt} = 1$$

      2. Stop integrating and proceed to step 4.

4. If the prediction captured, set the predicted apogee to negative infinity and exit the procedure.

$$a_{pred} = -\infty$$

5. Otherwise, compute the predicted apogee and then exit the procedure.

$$p_{pred} = \frac{\left[\left|\overline{R}_{pred}\right|\left|\overline{V}_{pred}\right| \cos\left(\gamma_{pred}\right)\right]^2}{\mu_{\oplus}} \tag{A.63}$$

$$e_{pred} = \sqrt{\left(1 - p_{pred}\left(\frac{2}{\left|\overline{R}_{pred}\right|} - \frac{\left|\overline{V}_{pred}\right|^{2}}{\mu_{\oplus}}\right)\right)} \tag{A.64}$$

$$r_{a_{pred}} = \frac{p_{pred}}{\left(1 - e_{pred}\right) - R_{\oplus}} \tag{A.65}$$

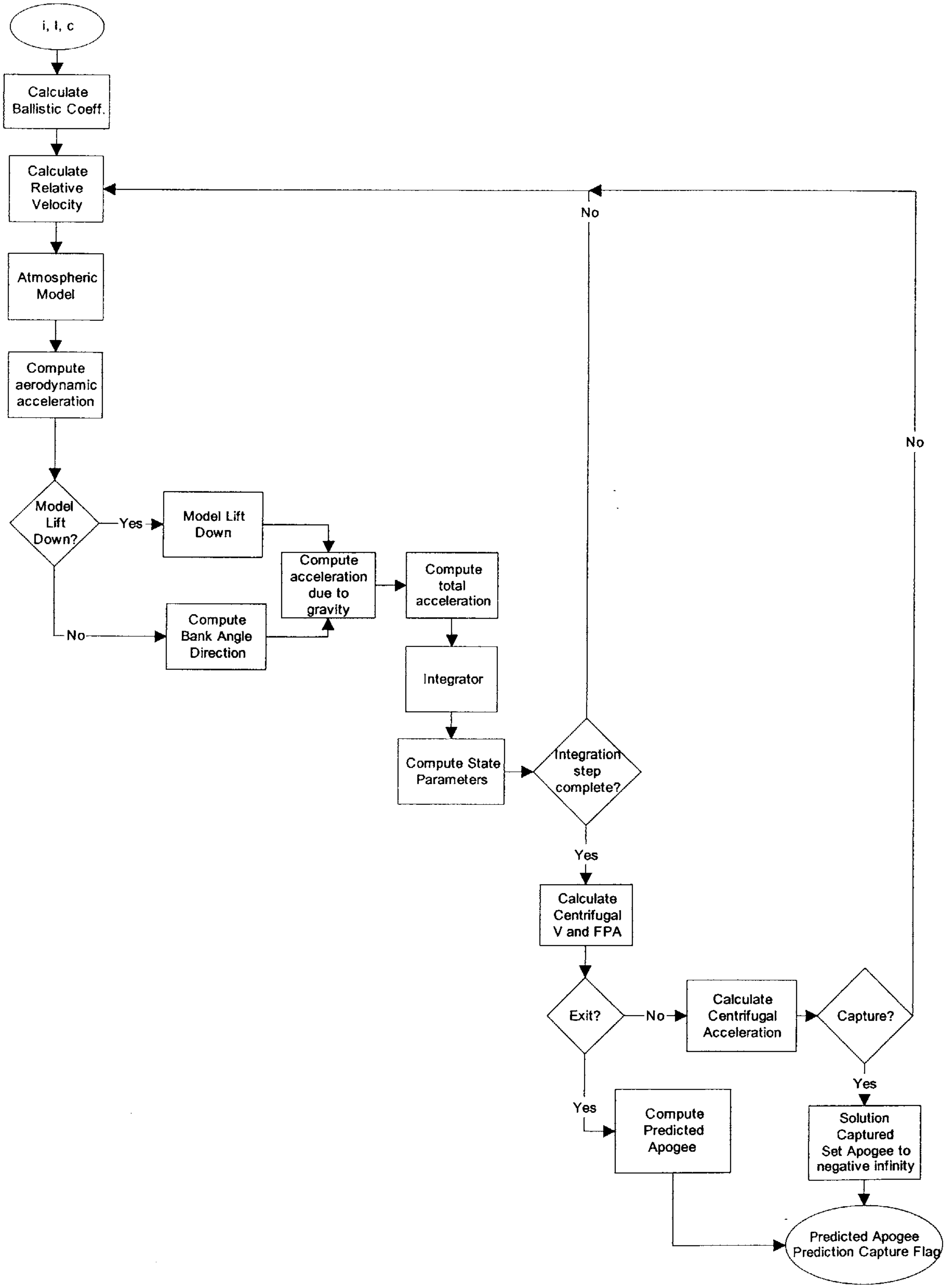

i, l, c
Calculate Ballistic Coeff.
Calculate Relative Velocity
Atmospheric Model
Compute aerodynamic acceleration
Model Lift Down?
Yes
Model Lift Down
No
Compute Bank Angle Direction
Compute acceleration due to gravity
Compute total acceleration
Integrator
Compute State Parameters
Integration step complete?
No
Yes
Calculate Centrifugal V and FPA
Exit?
No
Calculate Centrifugal Acceleration
Capture?
No
Yes
Compute Predicted Apogee
Yes
Solution Captured Set Apogee to negative infinity
Predicted Apogee Prediction Capture Flag

Figure A.7: Predictor Logical Flow: predictor.m

Table A-17: Inputs, predictor.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
i	R_NAV	$\overline{R}_{nav}$	Vehicle inertial position vector	ft	double[3]	Navigation
i	V_NAV	$\overline{V}_{nav}$	Vehicle inertial velocity vector	ft/s	double[3]	Navigation
	PHI_TRY	ϕ_{try}	Bank angle guess	deg	double	Corrector

Table A-18: Outputs, predictor.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Destination
	APOGEE_PREDICTED	a_{pred}	Predicted apogee	nm	double	pc_sequencer
	PRED_CAPTURE	F_{capt}	Flag variable indicates if prediction captured	(-)	double	pc_sequencer

Table A-19: Local Guidance Parameters, predictor.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Use
1	SIGN_OF_BANK	$sign(\phi)$	Sign of the current bank angle guess	(-)	double	Used
1	MODEL_LIFT_DOWN	F_{MLD}	Flag which activates lift down modeling	(-)	double	Used
1	CPHI_LIFT_DOWN	$\cos(\phi_{LD})$	Cosine of the lift down bank angle	(-)	double	Used

1	SPHI_LIFT_DOWN_SIGN	$\sin(\phi_{LD})$	Sign of the lift down bank angle	(-)	double	Used
1.AERO	CD_EST	$C_{D_{est}}$	Drag coefficient estimate	(-)	double	Used

Table A-20: Constants, predictor.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
c	MASS_EI	m_{EI}	Vehicle mass in EI units	slugs	double	15.4783
c	S_REF	S_{ref}	Reference area	Ft^2	double	12.163
c	DEG_TO_RAD	$C_{D\to R}$	Degrees to radians conversion	rad/deg	double	$\frac{\pi}{180.0}$
c	EARTH_RATE	$\omega_{\oplus}$	Angular rotation rate of the Earth	rad/sec	double	7.29211488e-5
c	EARTH_POLE	$\hat{e}_{pole}$	Unit vector through the Earth's pole	(-)	double[3]	$\begin{bmatrix}-0.0042772340\\-0.0000901672\\0.9999908485\end{bmatrix}$
c	VI_LIFT_DOWN		Switch for lift down modeling	Ft/s	double	27500.0
c	EARTH_J2	$J2_{\oplus}$	Earth J2 constant	(-)	double	1082.63e-6
c	EARTH_R	$r_{\oplus}$	Radius of Earth	ft	double	$\frac{6378137}{0.3048}$
c	EARTH_MU	$\mu_{\oplus}$	Gravitational parameter of Earth	ft^3/sec^2	double	$\frac{3.986005e14}{0.3048^3}$
c	ALT_EI	a_{alt}	Entry interface altitude	ft	double	400000.0

c	ALT_CAPT	a_{capt}	Altitude vehicle is considered captured	ft	double	200000.0
c	FT_TO_NM	$C_{FT \to NM}$	Conversion factor feet to nautical miles	Nm/ft	double	$\frac{1.0}{6076.116}$
c	INFINITY	∞	Infinity	(-)	double	99999999

A.7 Integrator

The integrator.m function uses a fourth order Runge-Kutta method to integrate the predicted state vectors forward one time step.

The logical flow of the Integrator algorithm is illustrated in Figure A.8.

The data inputs to integrator.m are described in Table A-21.

The data outputs from integrator.m are described in Table A-22.

The constants used in integrator.m are described in Table A-23.

1. If this is the first step of integration, execute the following steps:

   a. Set values for original position and velocity

   $$\overline{R}_{orig} = \overline{R}_{pred}$$

   $$\overline{V}_{orig} = \overline{V}_{pred}$$

   b. Set accumulated velocity and acceleration

   $$\overline{V}_{accum} = \overline{V}_{pred}$$

   $$\overline{a}_{accum} = \overline{a}_{pred}$$

   c. Perform first step of integration

   $$\overline{R}_{pred} = \overline{R}_{orig} + \frac{1}{2}\Delta t \overline{V}_{pred} \tag{A.66}$$

   $$\overline{V}_{pred} = \overline{V}_{orig} + \frac{1}{2}\Delta t \overline{a}_{pred} \tag{A.67}$$

2. If this is the second step of integration, execute the following steps:

a. Set accumulated velocity and acceleration

$$\overline{V}_{accum} = \overline{V}_{accum} + 2\overline{V}_{pred} \tag{A.68}$$

$$\overline{a}_{accum} = \overline{a}_{accum} + 2\overline{a}_{pred} \tag{A.69}$$

b. Perform second step of integration using Eq A.66 and Eq A.67

3. If this is the third step of integration, execute the following steps:

   a. Set accumulated velocity and acceleration using Eq. A.68 and A.69

   b. Perform third step of integration using Eq A.66 and Eq A.67

4. Otherwise, this is the fourth step, perform fourth step of integration

$$\overline{R}_{pred} = \overline{R}_{orig} + \frac{\Delta t}{6}\left(\overline{V}_{accum} + \overline{V}_{pred}\right) \tag{A.70}$$

$$\overline{V}_{pred} = \overline{V}_{orig} + \frac{\Delta t}{6}\left(\overline{a}_{accum} + \overline{a}_{pred}\right) \tag{A.71}$$

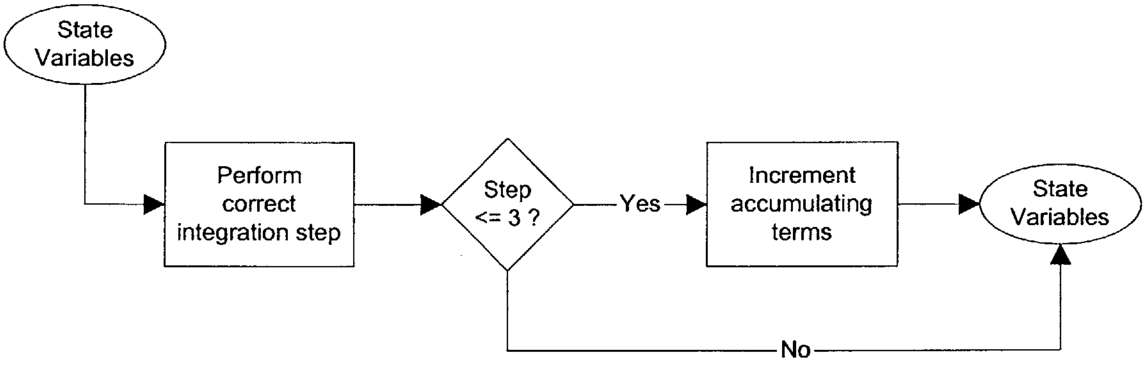

State Variables
Perform correct integration step
Step <= 3 ?
Yes
Increment accumulating terms
State Variables
No

Figure A.8: Integrator Logical Flow: integrator.m

Table A-21: Inputs, integrator.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
	INTEG_LOOP	F_{IL}	Count variable to determine which step of integration to perform	(-)	double[3]	predictor
	R_PRED	$\overline{R}_{pred}$	Predicted vehicle inertial position vector	ft	double	predictor
	V_PRED	$\overline{V}_{pred}$	Predicted vehicle inertial velocity vector	ft/s	double	predictor
	A_PRED	$\overline{a}_{pred}$	Predicted vehicle acceleration vector	ft/s^2	double	predictor

Table A-22: Outputs, integrator.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Destination
	R_PRED	$\overline{R}_{pred}$	Predicted vehicle inertial position vector	ft	double	predictor
	V_PRED	$\overline{V}_{pred}$	Predicted vehicle inertial velocity vector	ft/s	double	predictor

Table A-23: Constants, integrator.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
c	DELTA_T_PRED	Δt	Time step for integration	s	double	20

A.8 Lateral Guidance

The lateral_control.m function computes the sign for the commanded bank angle and determines the need for the roll reversals that keep the spacecraft in the desired orbital plane.

The logical flow of the Lateral Guidance algorithm is illustrated in Figure A.9.

The data inputs to lateral_control.m are described in Table A-24.

The local guidance parameters used or updated in lateral_control.m are described in Table A-25.

The constants used in lateral_control.m are described in Table A-26.

1. Compute the maximum lateral corridor out of plane velocity by executing the following steps:
   a. If the inertial velocity magnitude is less than or equal to the minimum corridor velocity, the corridor out of plane velocity is equal to the minimum corridor out of plane velocity

   $$C = C_{\min}$$

   b. Otherwise calculate the maximum corridor out of plane velocity based on the current forward velocity and the equation of the line and ensure it is within the minimum and maximum values

   $$C = C_{\min} - sign\left(\left(-\frac{L}{D}\right)_{est}\right) S_{corr}\left(\left|\overline{V}_{nav}\right| - V_{corr_{\min}}\right) \tag{A.72}$$

2. Compute the actual out of plane velocity based on the navigational velocity and the unit normal to the desired plane

   $$\dot{y} = \overline{V}_{nav}^{T} \overline{I}_{yd} \tag{A.73}$$

3. If absolute value of the out of plane velocity is greater than the maximum corridor out of plane velocity $|\dot{y}| > C$, the vehicle is outside of the corridor. The sign of bank is opposite that of the out of plane velocity sign to cause a roll reversal

   $$sign(\phi) = -sign(\dot{y}) \tag{A.74}$$

4. Otherwise, the sign of bank is equal to the current sign of bank global parameter value

$$sign(\phi) = sign(\phi) \tag{A.75}$$

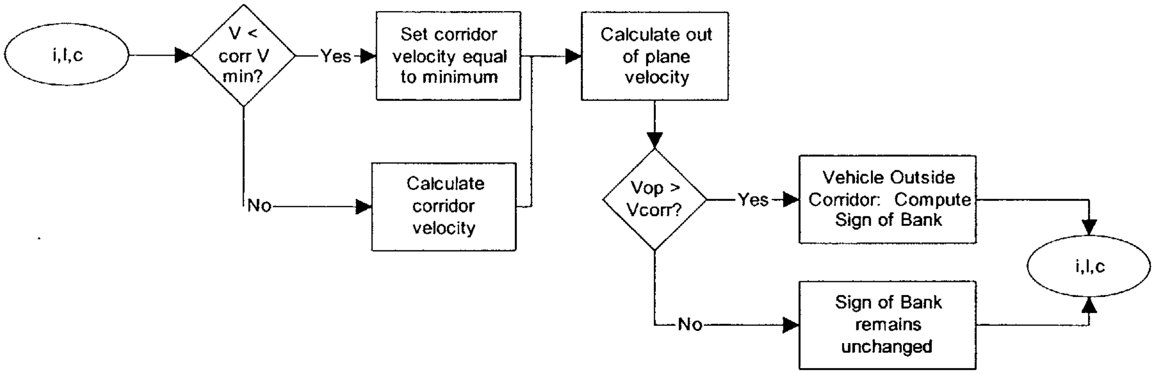

i,l,c
V < corr V min?
Yes
No
Set corridor velocity equal to minimum
Calculate corridor velocity
Calculate out of plane velocity
Vop > Vcorr?
Yes
No
Vehicle Outside Corridor: Compute Sign of Bank
Sign of Bank remains unchanged
i,l,c

Figure A.9: Lateral Guidance Logical Flow: lateral_control.m

Table A-24: Inputs, lateral_control.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
i	V_NAV_MAG	$\lvert\bar{V}_{nav}\rvert$	Vehicle inertial velocity vector magnitude	ft/s	double	Derived
i	V_NAV	$\bar{V}_{nav}$	Vehicle inertial velocity vector	ft/s	double[3]	Navigation
i	IYD	$\bar{I}_{YD}$	Unit normal to the desired orbit plane	(-)	double[3]	Derived

Table A-25: Local Guidance Parameters, lateral_control.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Use
1	CORRIDOR_SLOPE	S_{corr}	Aerocapture corridor slope (out of plane velocity over forward velocity)	(-)	double	Used
1	SIGN_OF_BANK	$sign(\phi)$	Sign of the current bank angle guess	(-)	double	Updated
1.AERO	L_OVER_D_EST	$\left.\frac{L}{D}\right)_{est}$	Lift to drag ratio estimate	(-)	double	Used

Table A-26: Constants, lateral_control.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
C	CORRIDOR_MAX	$C_{\max}$	Maximum lateral corridor out of plane velocity	ft/s	double	2000.0
C	CORRIDOR_MIN	$C_{\min}$	Minimum lateral corridor out of plane velocity	ft/s	Double:	15.0
C	CORRIDOR_V_MAX	$C_{V_{\max}}$	Maximum forward velocity to use corridor slope	ft/s	double	31000.0

A.9 Bank Angle Command Incorporation

The cmd_incorporation.m function updates the bank angle command.

The logical flow of the Bank Angle Command Incorporation algorithm is illustrated in Figure A.10.

The data inputs to cmd_incorporation.m are described in Table A-27.

The data outputs from cmd_incorporation.m are described in Table A-28.

The local guidance parameters used or updated in cmd_incorporation.m are described in Table A-29.

The constants used in cmd_incorporation.m are described in Table A-30.

1. Select the nominal bank angle command
   a. If inertial velocity is less than the bias switch velocity
   $$\cos(\phi_{nom}) = \cos(\phi_{bias}(1))$$
   b. Otherwise
   $$\cos(\phi_{nom}) = \cos(\phi_{bias}(2))$$
2. Update the bank command
   c. If inertial velocity is less than the command bias required velocity, bias the commanded bank angle
   $$\cos(\phi_{cmd}) = \cos(\phi_{des}) + K_{cb}(\cos(\phi_{des}) - \cos(\phi_{nom})) \tag{A.76}$$
   d. Otherwise, do not bias the commanded bank angle
   $$\cos(\phi_{cmd}) = \cos(\phi_{des}) \tag{A.77}$$
3. Limit the cosine of the bank angle

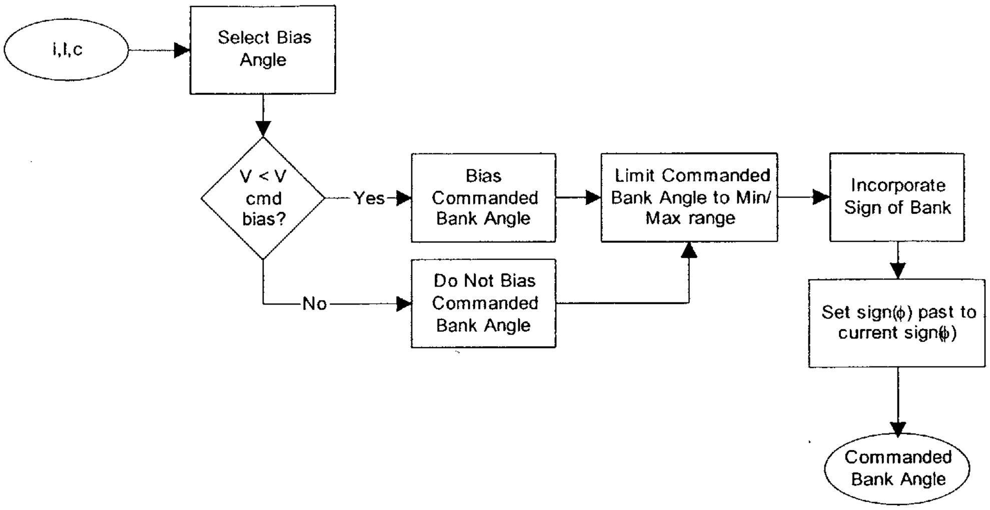

i,l,c
Select Bias Angle
V < V cmd bias?
Yes
No
Bias Commanded Bank Angle
Do Not Bias Commanded Bank Angle
Limit Commanded Bank Angle to Min/ Max range
Incorporate Sign of Bank
Set sign(ϕ) past to current sign(ϕ)
Commanded Bank Angle

Figure A.10: Command Incorporation Logical Flow: cmd_incorporation.m

Table A-27: Inputs, cmd_incorporation.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
i	V_NAV_MAG	$\left\|\overline{V}_{nav}\right\|$	Vehicle inertial velocity vector magnitude	ft/s	double	Derived

Table A-28: Outputs, cmd_incorporation.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Destination
	PHI_CMD	ϕ_{cmd}	Commanded bank angle	deg	double	pred_guid

Table A-29: Local Guidance Parameters, cmd_incorporation.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Use
l	COS_PHI_MIN	$\cos(\phi)\|_{\min}$	Cos max bank angle	(-)	double	Used

l	COS_PHI_MAX	$\cos(\phi)\|_{\max}$	Cos min bank angle	(-)	double	Used
l	SIGN_OF_BANK	$sign(\phi)$	Sign of current bank angle guess	(-)	double	Used
l	SIGN_OF_BANK_PAST	$sign(\phi)_{past}$	Sign of previous bank angle guess	(-)	double	Updated
l	COS_PHI_DES	$\cos(\phi_{des})$	Cosine of desired bank angle	(-)	double	Used

Table A-30: Constants, cmd_incorporation.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
c	RAD_TO_DEG	$C_{R\to D}$	Radians to degrees conversion factor	deg/rad	double	$\frac{180.0}{\pi}$
c	VI_BIAS_SWITCH		Velocity to switch between bias angles	ft/s	double	30800.0
c	COS_PHI_BIAS	$\cos(\phi)_{bias}$	Vector which stores two bias angles	(-)	double[2]	$\begin{bmatrix}0.0\\0.0\end{bmatrix}$
c	VI_COMMAND_BIAS		Minimum velocity to bias bank angle	ft/s	double	26500.0
c	K_COMMAND_BIAS	K_{cb}	Gain which determines how quickly the solution biases	(-)	double	2

A.10 Atmospheric Model

The atmos_model.m function computes the atmospheric density using a fourth order curve fit of scale height.

•

The logical flow of the Atmospheric Model algorithm is illustrated in Figure A.11.

The data inputs to atmos_model.m are described in Table A-31.

The data outputs from atmos_model.m are described in Table A-32.

The constants used in atmos_model.m are described in Table A-33.

1. Calculate the normalized altitude

$$H_{norm} = \frac{H}{H_{ref}} \tag{A.78}$$

2. If the current altitude is greater than the maximum atmospheric altitude, calculate the scale height using the maximum atmospheric altitude

$$HS_{norm} = \frac{HS_{alt\max}}{H_{ref}} \tag{A.79}$$

3. Otherwise, use a fourth order curve fit to get scale height

$$HS_{norm} = C_5^{HS} H_{norm}^4 + C_4^{HS} H_{norm}^3 + C_3^{HS} H_{norm}^2 + C_2^{HS} H_{norm} + C_1^{HS} \tag{A.80}$$

4. Calculate density

$$\rho_{std} = \rho_{ref} e^{(1-H_{norm})/HS_{norm}} \tag{A.81}$$

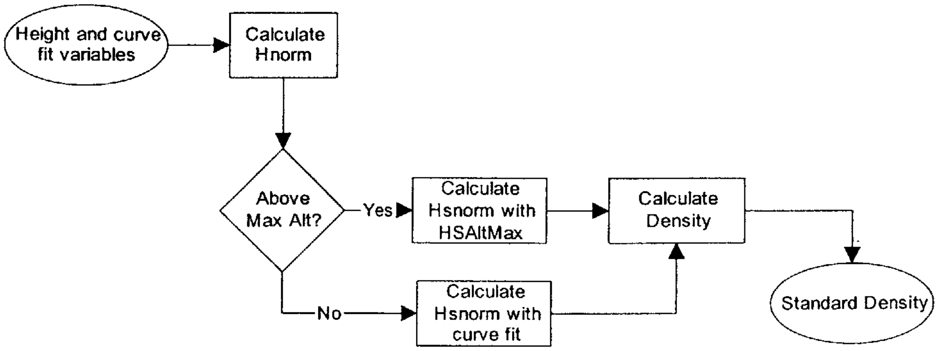

Height and curve fit variables
Calculate Hnorm
Above Max Alt?
Yes
Calculate Hsnorm with HSAltMax
Calculate Density
No
Calculate Hsnorm with curve fit
Standard Density

Figure A.11: Atmospheric Model Logical Flow: atmos_model.m

Table A-31: Inputs, atmos_model.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Source
	H	H	Altitude	ft	double	various

Table A-32: Outputs, atmos_model.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Destination
	RHO_STD	ρ_{std}	Standard density		double	various

Table A-33: Constants, atmos_model.m

Structure Name	Variable Name	Symbol	Description	Units	Data Type/ Precision	Value
c	H_REF	H_{ref}	Reference altitude	ft	double	242524.94
c	ATMOS_ALT_MAX	$a_{alt\max}$	Altitude of maximum sensible atmosphere	ft	double	400000.0
c	C_HS	C^{HS}	Fourth order curve fit of density	(-)	double[2]	$\begin{bmatrix} 9.154583e-1 \\ -2.574749e0 \\ 3.023997e0 \\ -1.595031e0 \\ 3.168398e-1 \end{bmatrix}$
c	HS_ALT_MAX	$H_{alt\max}$	Maximum scale height	ft	double	20000.0
c	RHO_REF	ρ_{ref}	Reference density	slugs/ft^3	double	9.9469294e-8

[This Page Intentionally Left Blank]

Appendix B
Simulation Verification

B.1 Dynamic Simulation Verification

The dynamic simulation written in Matlab Simulink was verified against an existing simulation. The existing simulation used the US Standard Atmosphere, 1976 and assumed spherical Earth. These conditions were applied to the Simulink simulation and a 90° constant bank angle input was used. Figure B.1 illustrates the verification setup

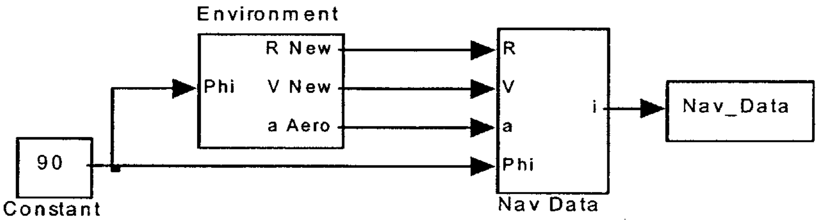


Figure B.1: Dynamic Simulation Verification Setup

Plotting both the trajectory generated by the above set up and the trajectory generated by the existing simulation, both using a 90° constant bank angle input, Figure B.2 through Figure B.9 were generated.

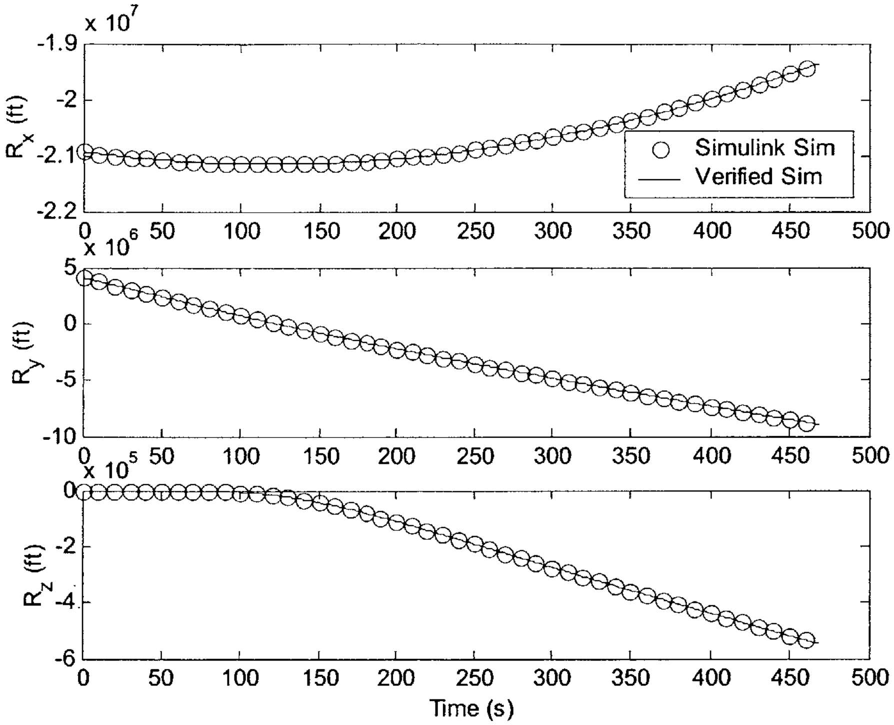


Figure B.2: Vehicle Position vs. Time

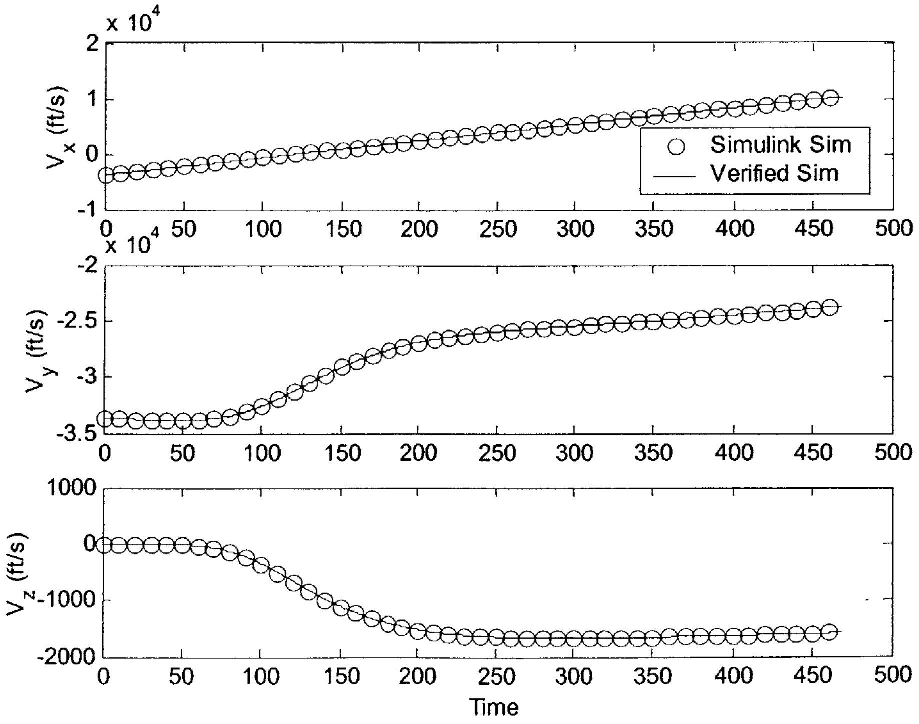


Figure B.3: Vehicle Velocity Components vs. Time

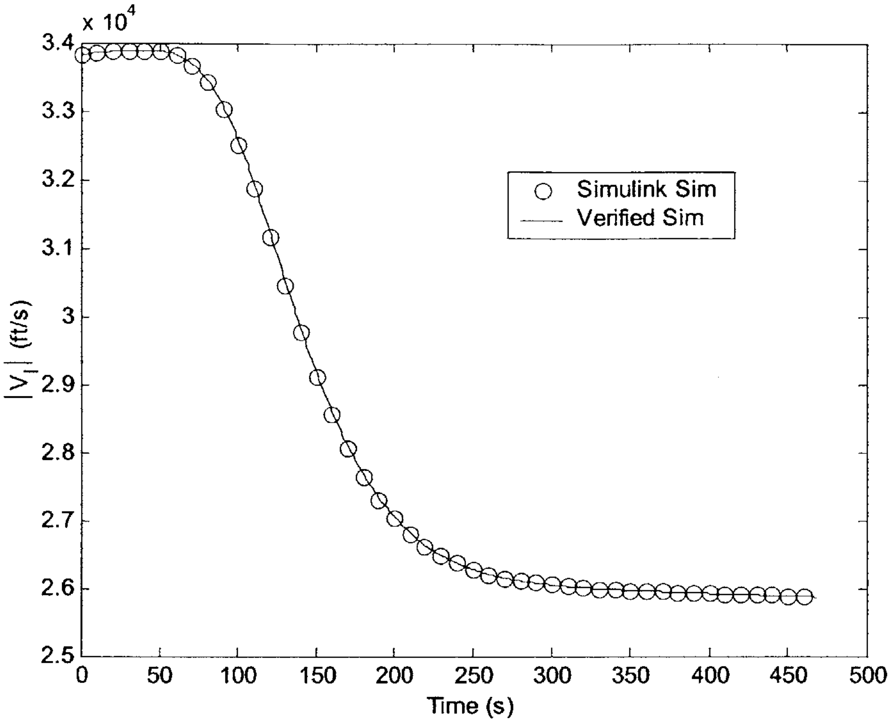


Figure B.4: Vehicle Velocity Magnitude vs. Time

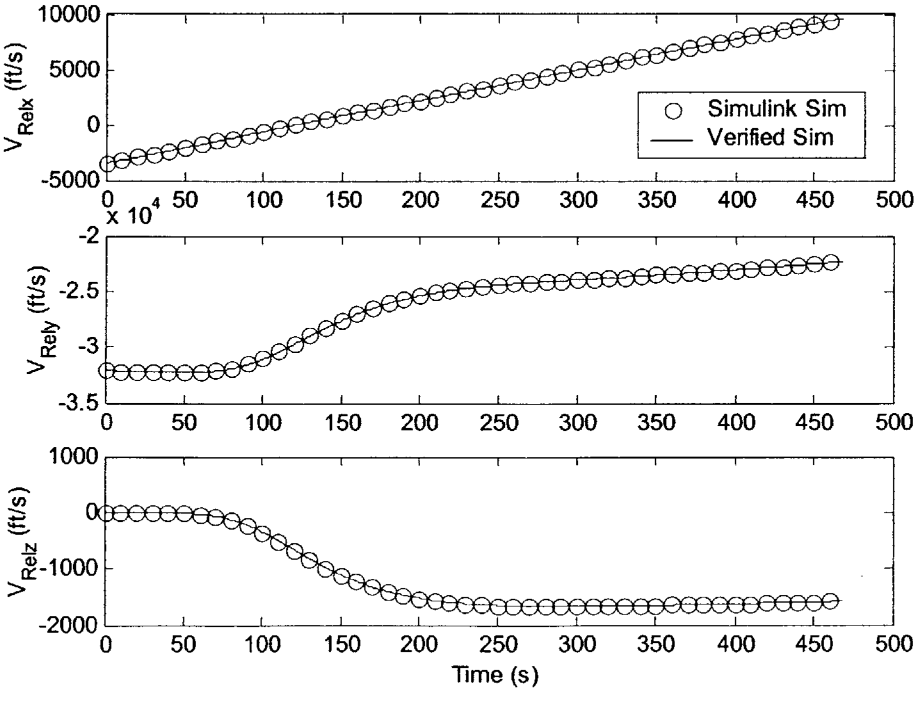


Figure B.5: Vehicle Relative Velocity Components vs. Time

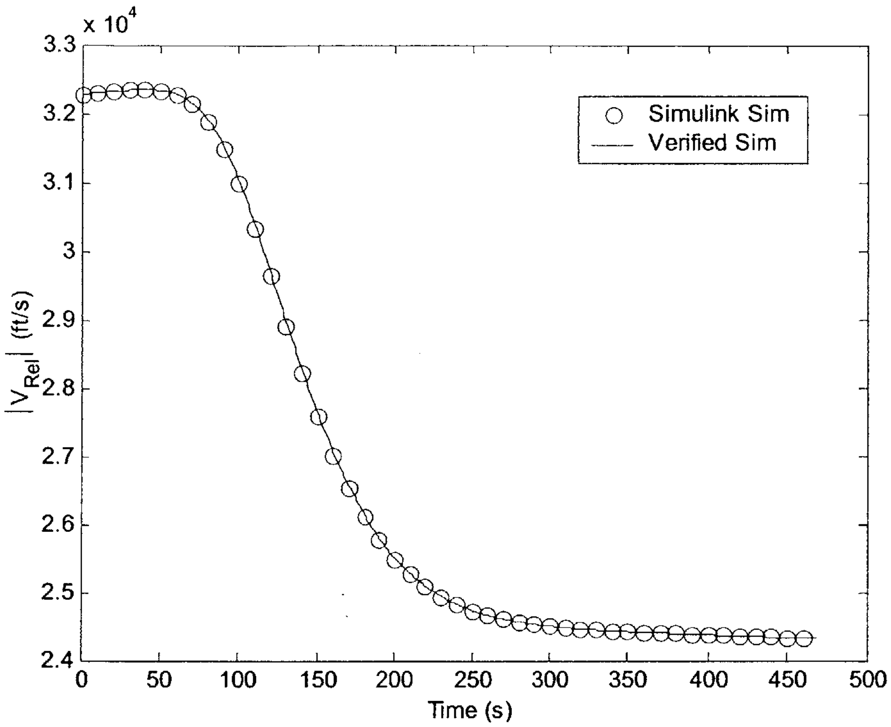


Figure B.6: Vehicle Relative Velocity Magnitude vs. Time

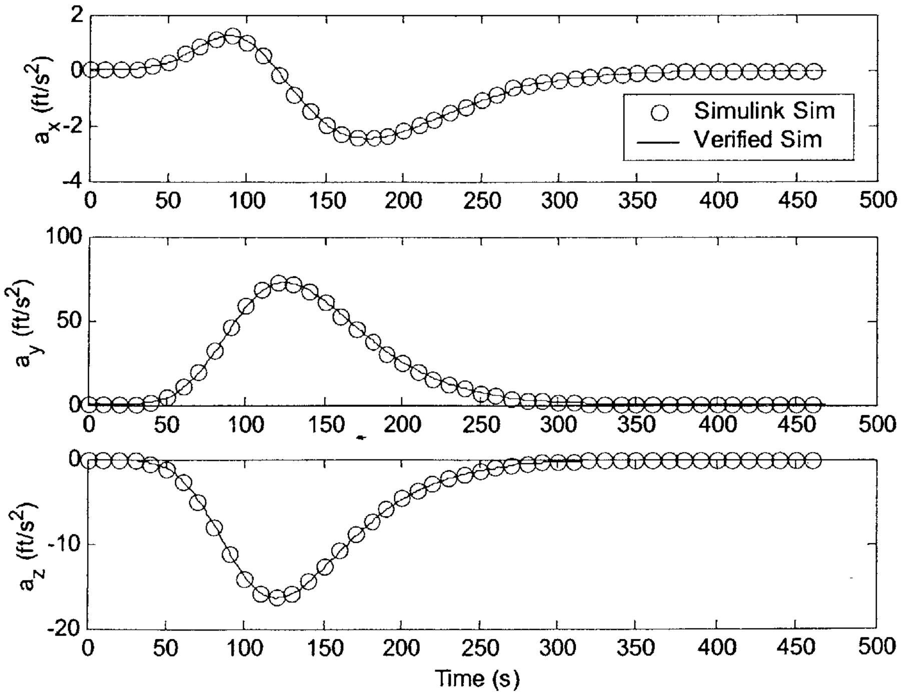


Figure B.7: Vehicle Acceleration Components vs. Time

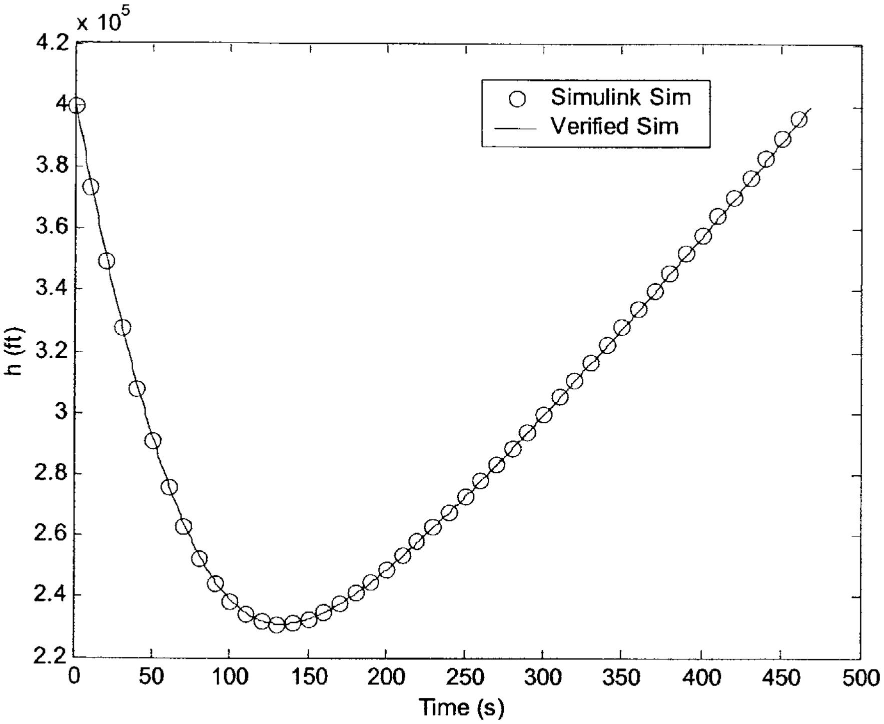


Figure B.8: Vehicle Altitude vs. Time

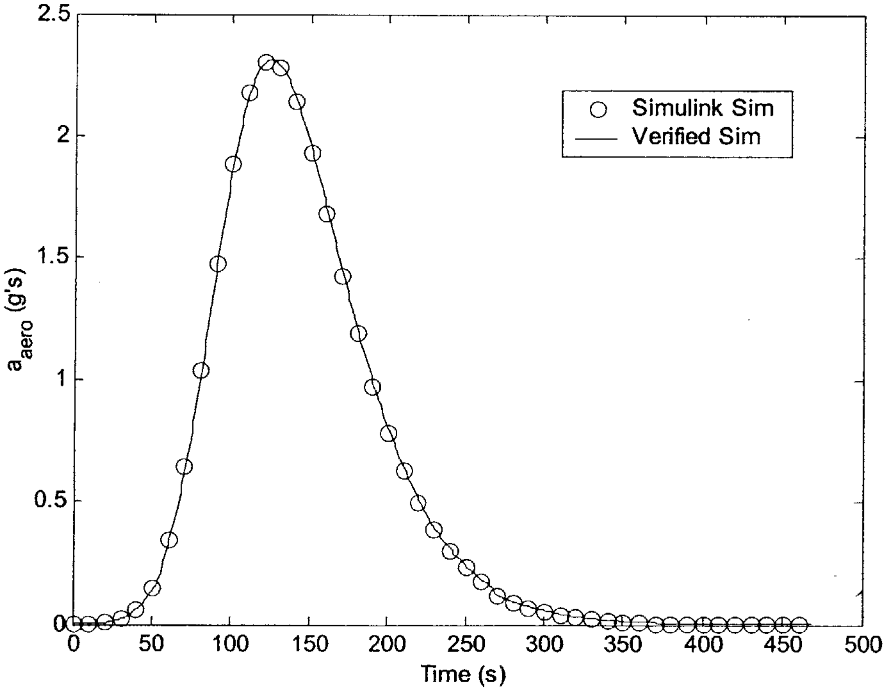


Figure B.9: Vehicle Acceleration vs. Time

The Simulink simulation was then updated to include J2 effects and plotted vs. the verified simulation in order to ensure the results were still reasonable. These plots are shown in Figure B.10 through Figure B.17.

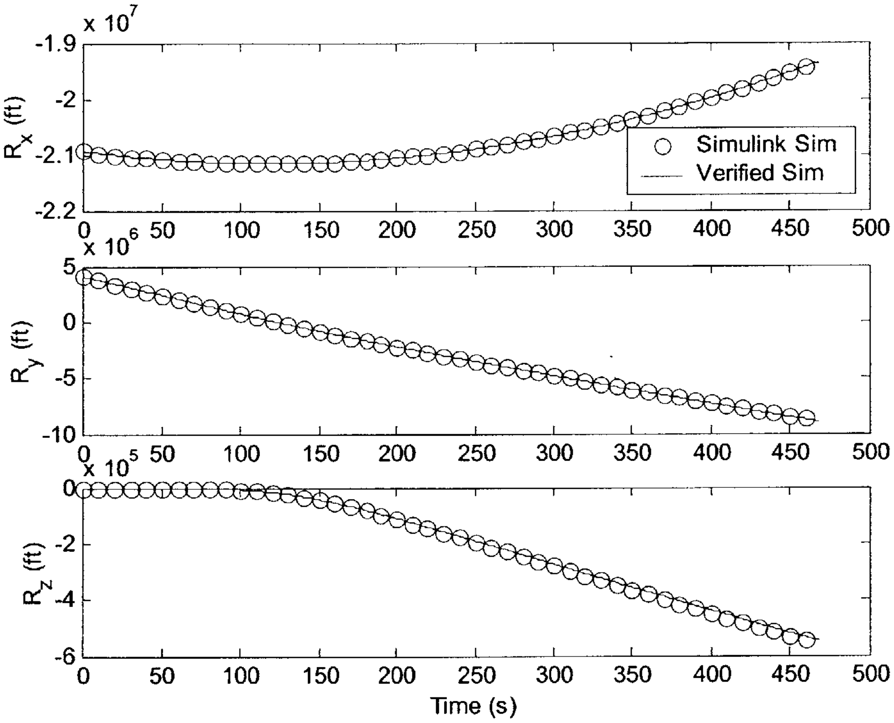


Figure B.10: Vehicle Position vs. Time, J2 Effects

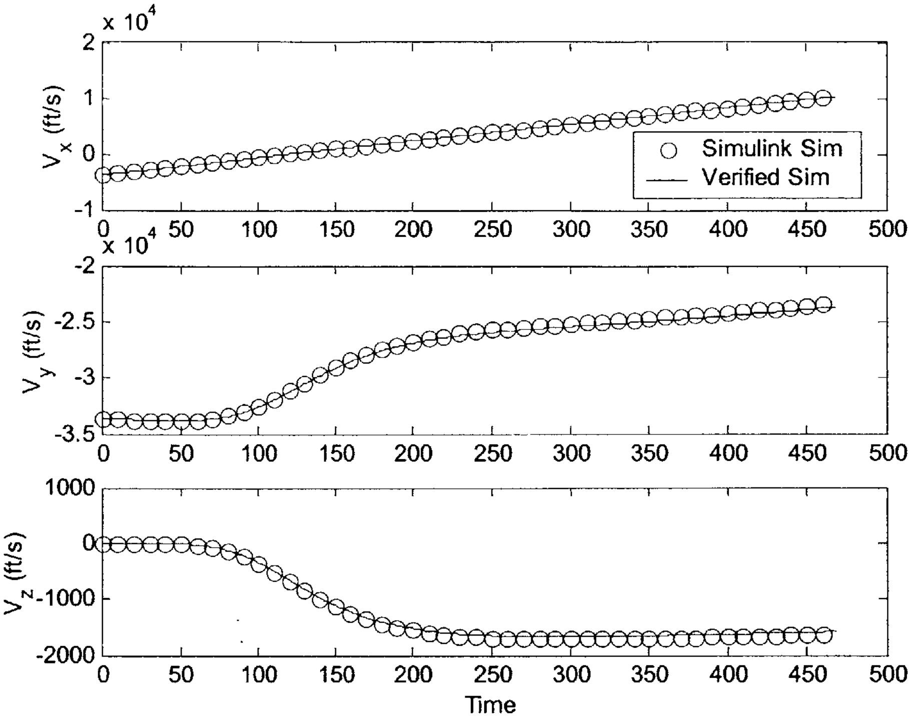


Figure B.11: Vehicle Velocity Components vs. Time, J2 Effects

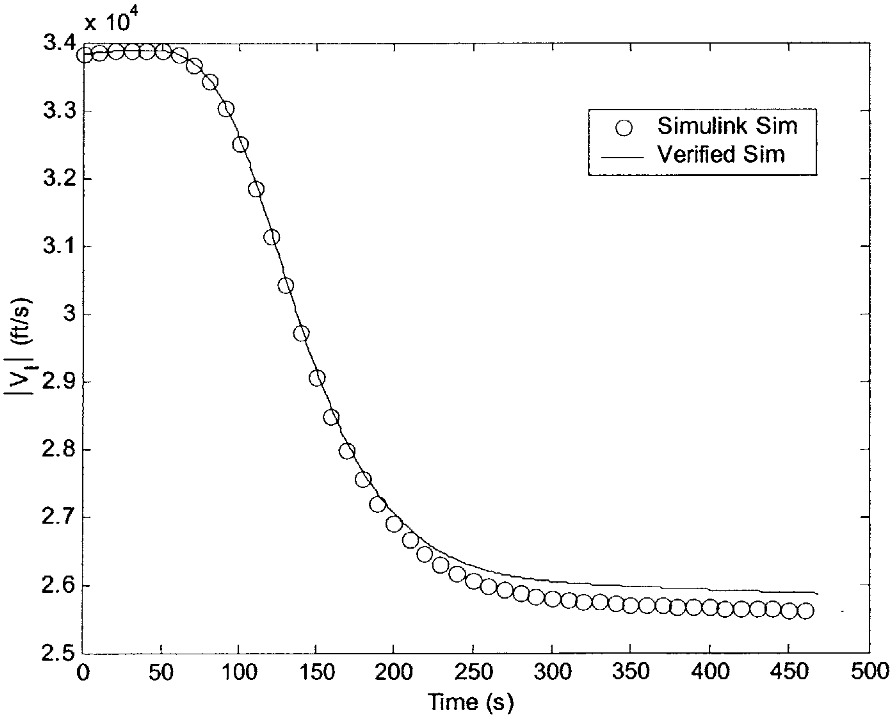


Figure B.12: Vehicle Velocity Magnitude vs. Time, J2 Effects

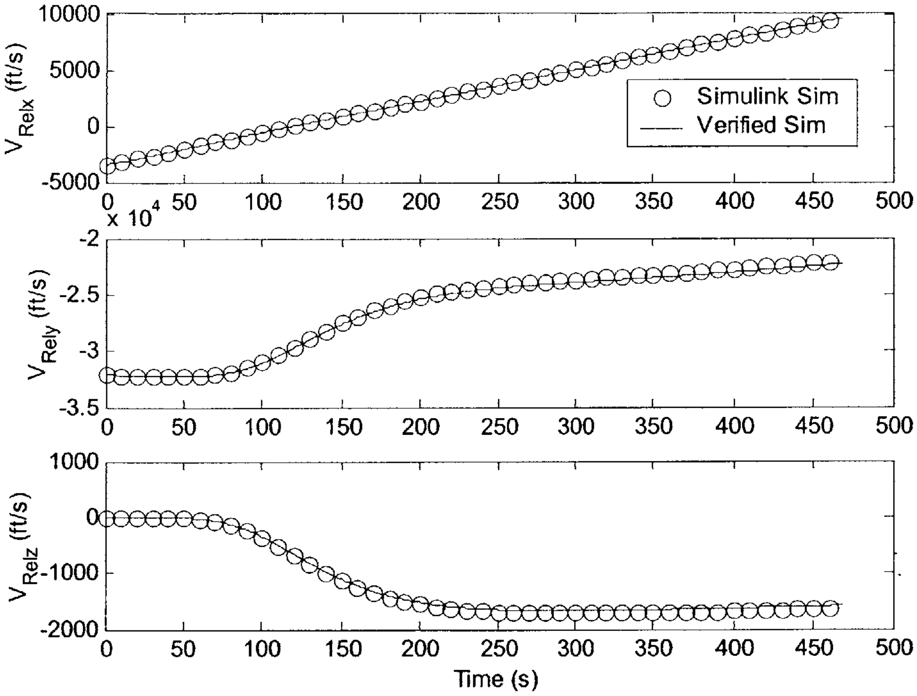


Figure B.13: Vehicle Relative Velocity Components vs. Time, J2 Effects

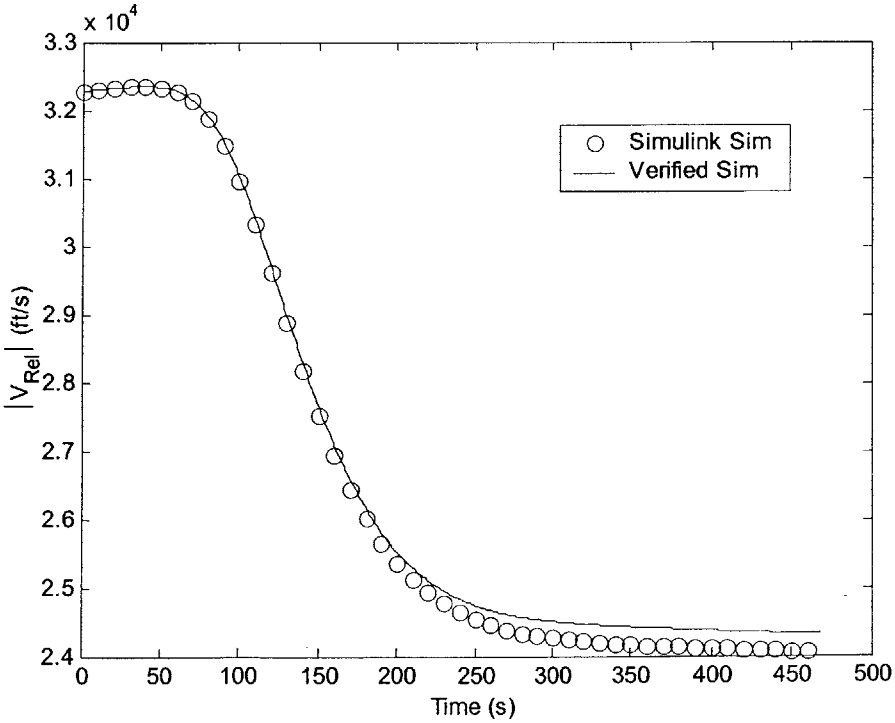


Figure B.14: Vehicle Relative Velocity Magnitude vs. Time, J2 Effects

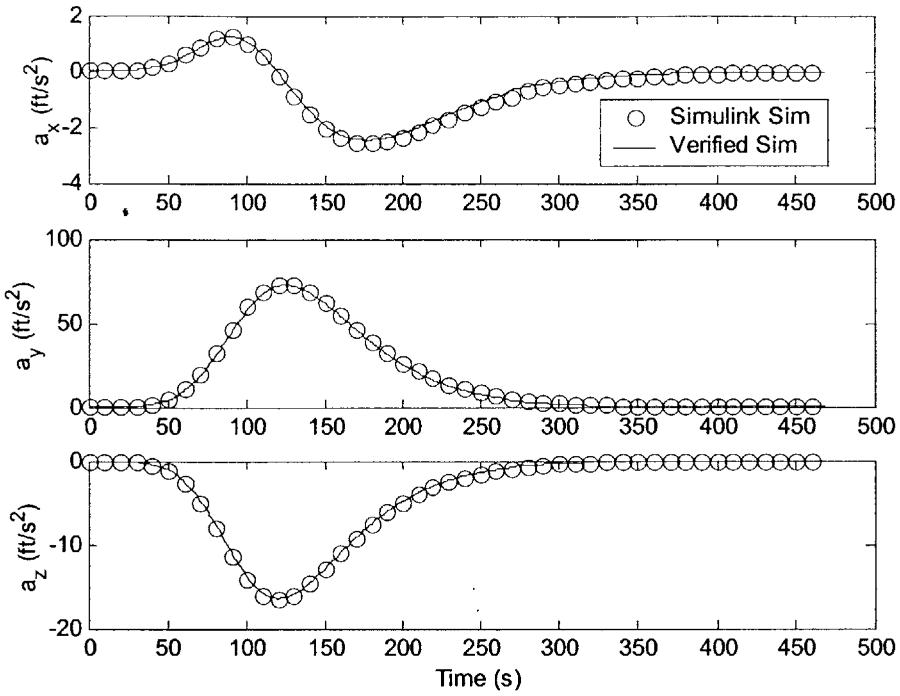


Figure B.15: Vehicle Acceleration Components vs. Time, J2 Effects

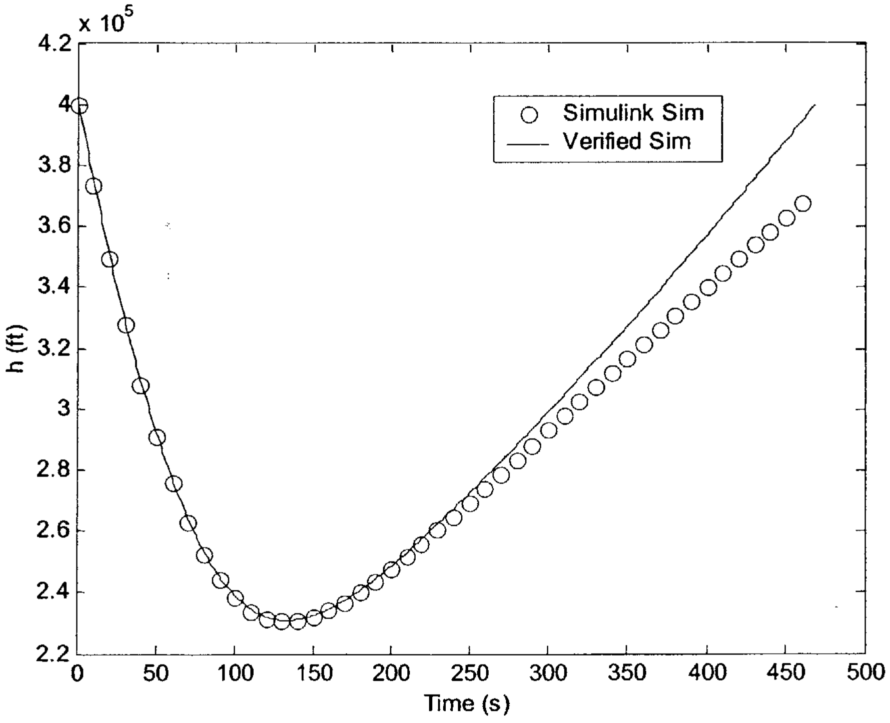


Figure B.16: Vehicle Altitude vs. Time, J2 Effects

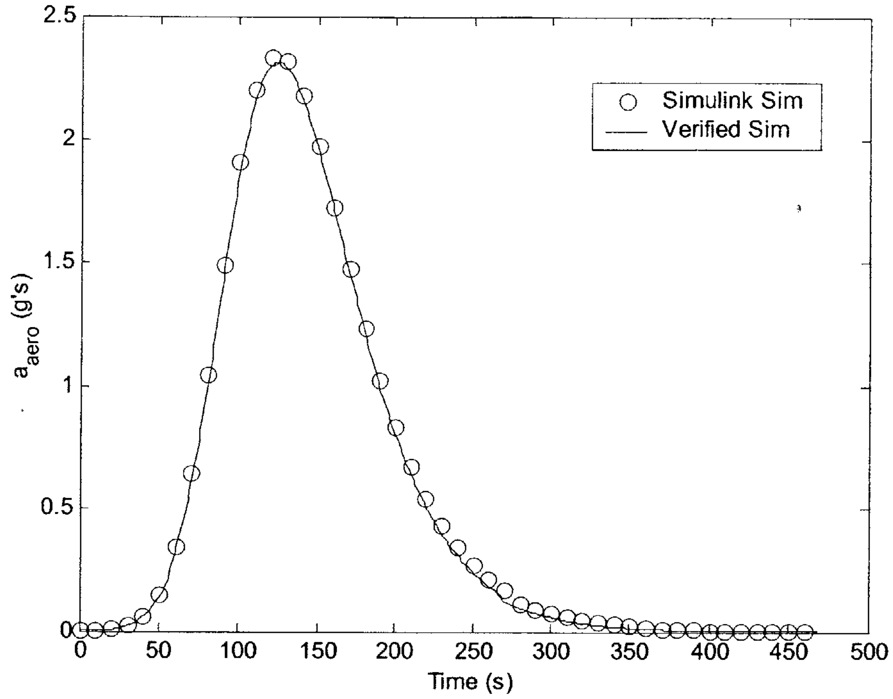


Figure B.17: Vehicle Acceleration vs. Time, J2 Effects

B.2 Guidance Verification

In order to verify the guidance portion of the simulation (PredGuid), an input file was used to run the Simulink simulation and the original pred_guid code in an open loop format. This setup is illustrated in Figure B.18. With the same inputs and no feedback, the two programs generated the same bank angle profile as shown in Figure B.19.

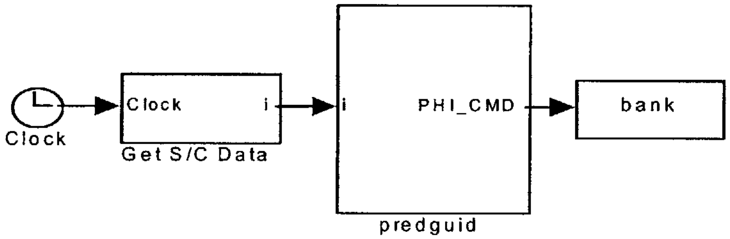


Figure B.18: Open Loop Guidance Verification Setup

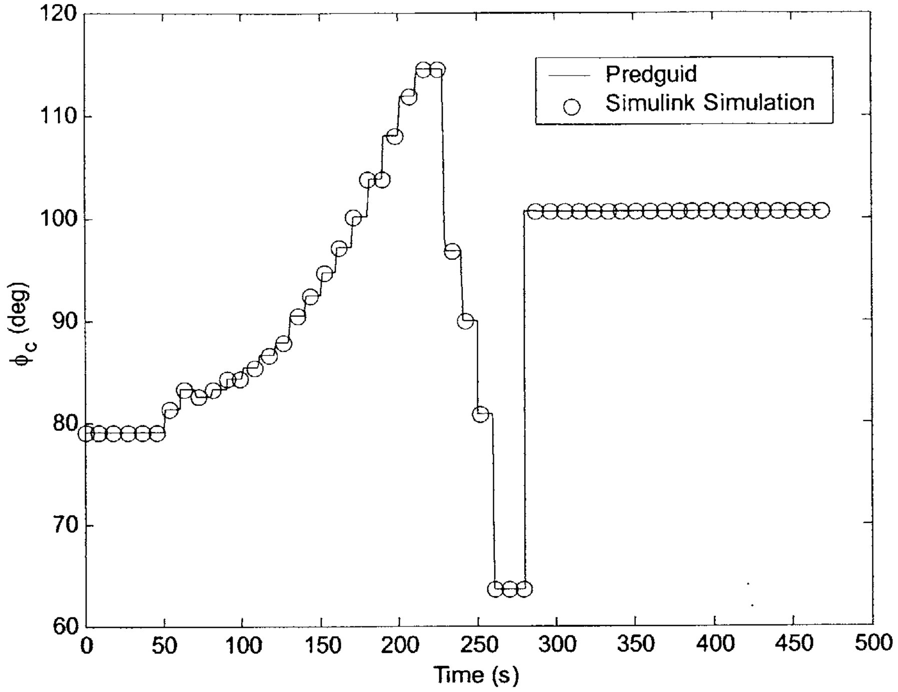


Figure B.19: Bank Angle Command vs. Time, Open Loop

B.3 Closed Loop Simulation

After independently verifying the dynamics and open loop guidance, the two components were run in a closed loop format. While there was no 'known' trajectory to verify against, several things would indicate the program was running properly. The trajectory should hover around 90° for two reasons: First, the initial conditions used in the closed loop simulation were for a trajectory that, with a constant bank angle of 90°, should hit the target apogee. Second, the command biasing angle was set to 90°. As a result, the commanded bank angle should settle at 90°. The closed loop setup is illustrated in Figure B.20. The resulting commanded bank angle profile is shown in Figure B.21.

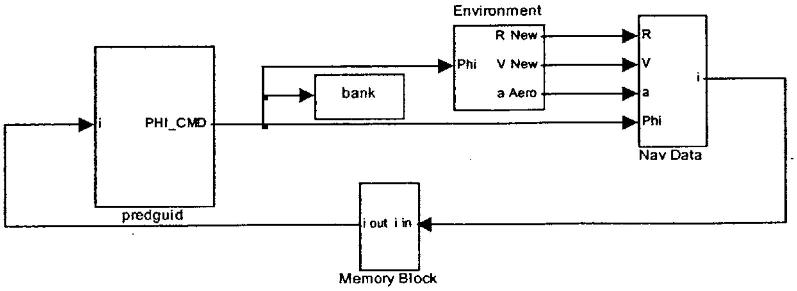


Figure B.20: Closed Loop Guidance Verification Setup

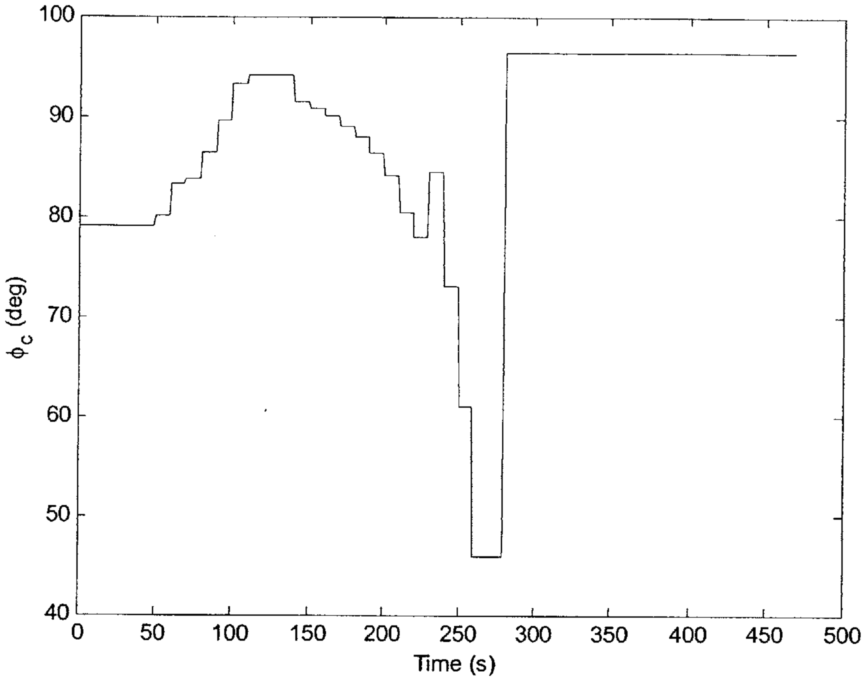


Figure B.21: Bank Angle Command vs. Time, Closed Loop

This profile was not as close to settling at 90° as hoped. The problem was determined to be that guidance used a different atmosphere model than the dynamics were using. In reality, guidance does not have perfect knowledge of the atmosphere and the values used by guidance would not be identical to the actual environment. However, for debugging and development purposes, assuming perfect knowledge of the atmosphere helps to determine if the guidance logic itself is flawed versus not being robust enough to handle atmospheric dispersions. As a result, the atmosphere model in the guidance was replaced with the US Standard Atmosphere, 1976. Figure B.22 shows that the results were much closer to what was expected.

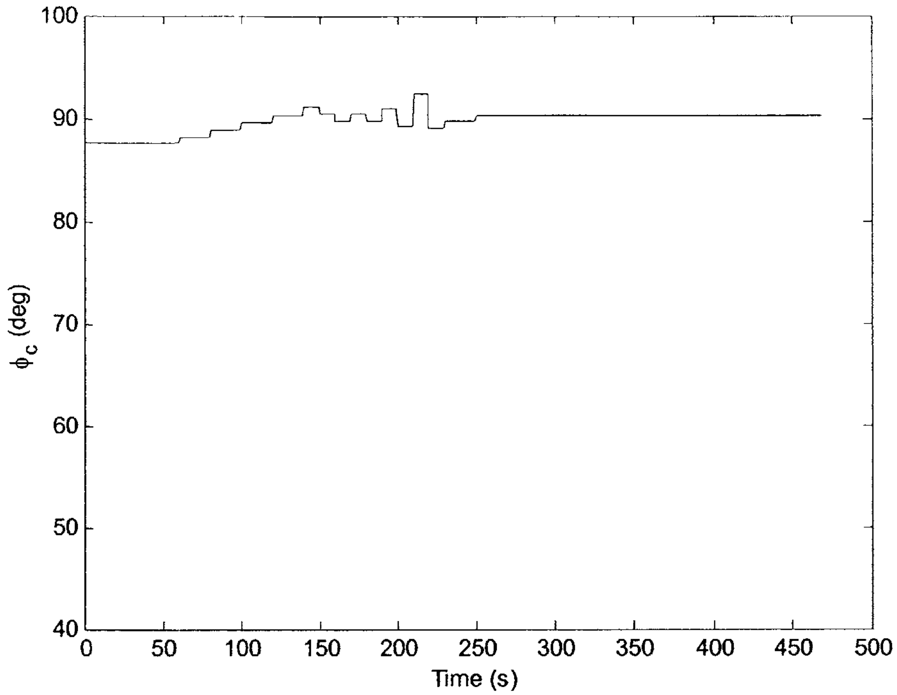


Figure B.22: Bank Angle Command vs. Time, Closed Loop, US 1976

B.4 Guidance Corrections

In the process of translating the original guidance code from Matlab to Simulink, several errors and problems were discovered in the code. First, the simulation was not executing bank reversals. This lack of reversals is obvious from Figure B.22. However, a plot of the lateral corridor and out-of-plane velocity (Figure B.23) further verified this observation. As the vehicle approached the lateral corridor limits, instead of reversing bank to stay within the limits, it continued to move out of plane.

Examination of the code revealed that the unit normal to the desired plane was being calculated incorrectly. Eq. B.1 shows the correct calculation and B.2 shows the original incorrect calculation.

$$\vec{I}_{yd} = \frac{\vec{v} \times \vec{r}}{|\vec{v} \times \vec{r}|} \tag{B.1}$$

$$\vec{I}_{yd} = \frac{\vec{r} \times \vec{v}}{|\vec{r} \times \vec{v}|} \tag{B.2}$$

Figure B.24 shows the lateral corridor and out of plane velocity after making this correction.

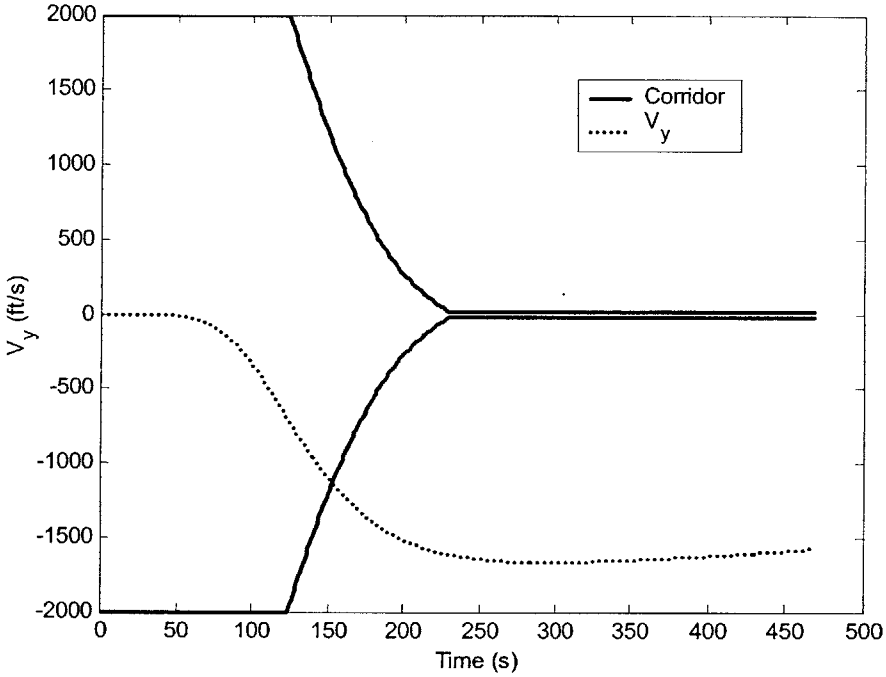


Figure B.23: Lateral Corridor and Out of Plane Velocity, No Bank Reversals

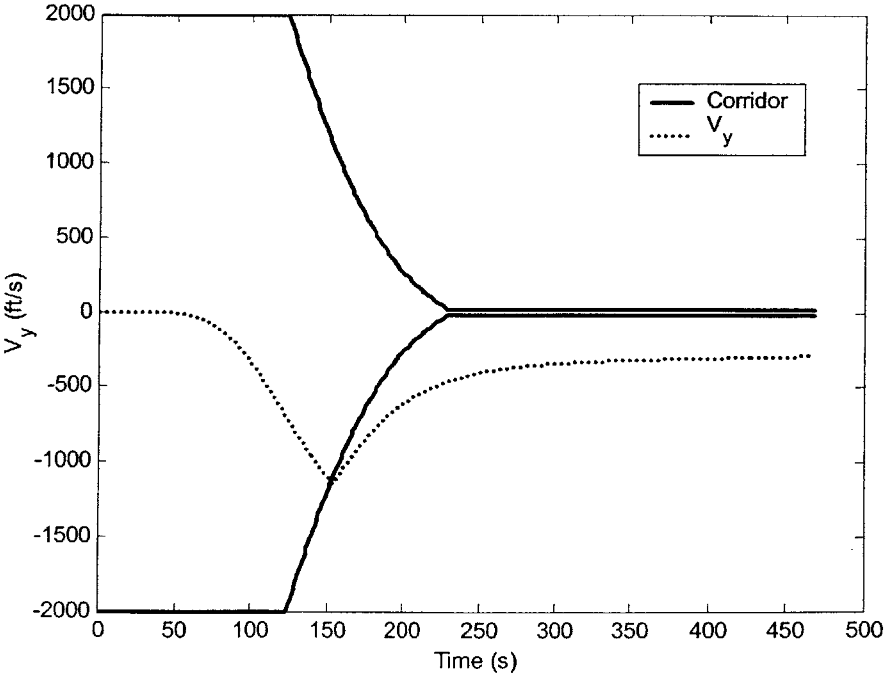


Figure B.24: Lateral Corridor and Out of Plane Velocity, Corrected

Further examination of the code revealed several problems with sign usage. When the code was originally written for the Aeroassist Flight Experiment (AFE), a negative value for lift-to-drag ratio was used. When translated to Matlab for another aerocapture proposal, the decision was made to use a more intuitive, positive lift-to-drag ratio. In the process of translating and switching signs, several instances of incorrect sign usage were overlooked. After correcting all of these instances, a closed loop simulation was run and produced the bank angle profile in Figure B.25.

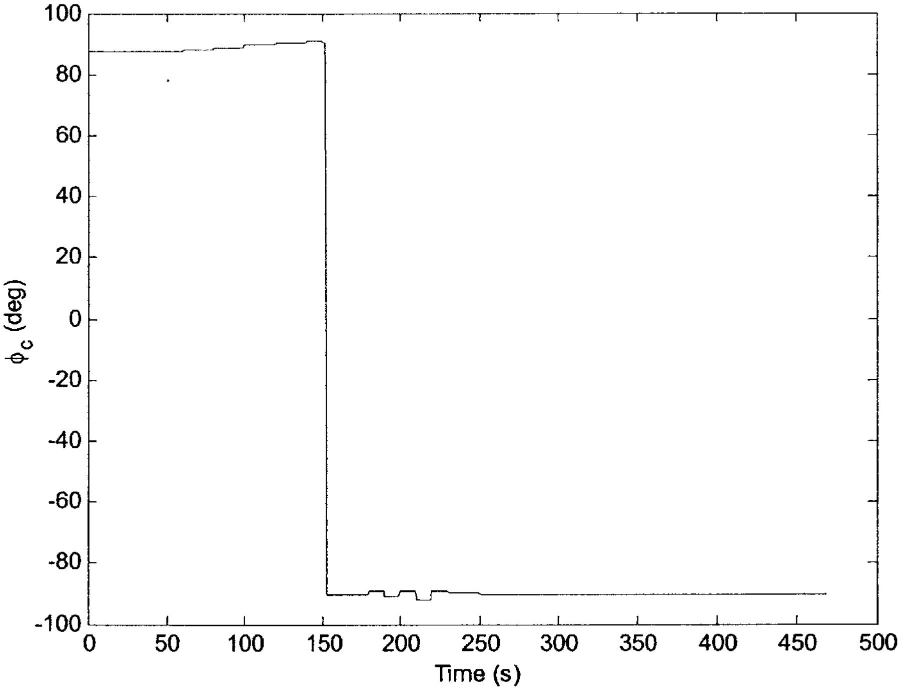


Figure B.25: Bank Angle Command vs. Time, Correct Guidance

[This Page Intentionally Left Blank]

References

[1] Gamble, J.D., Cerimele, C.J., Moore, T.E., and Higgins, J. "Atmospheric Guidance Concepts for an Aeroassist Flight Experiment," *Journal of the Astronautical Sciences*, Vol. 36, Nos. 1/2 , January-June 1988, pp 45-71.

[2] Masciarelli, J. P., Rousseau, S., Fraysse, H., Perot, E., "An Analytic Aerocapture Guidance Algorithm for the Mars Sample Return Orbiter," AIAA Paper 2000-4116, AIAA Atmospheric Flight Mechanics Conference, Denver, CO, 14-17 Aug 2000.

[3] Graves, C.A., Masciarelli, J.P., "An Analytical Assessment of Aerocapture Guidance and Navigation Flight Demonstration for Applicability to Other Planets," AIAA 2002-4622, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Monterey, CA, 5-8 Aug 2002.

[4] Hall, J.L. , "An Overview of the Aerocapture Flight Test Experiment (AFTE)" AIAA 2002-4621, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Monterey, CA, 5-8 Aug 2002.

[5] Ling, L., Baseggio, F.M., Fuhry, D.P., "Parametric Entry Corridors for Lunar/Mars Aerocapture Missions," NASA Technical Memorandum 102178, April 1991.

[6] Fuhry, D.P., *A Design Study of Onboard Navigation and Guidance During Aerocapture at Mars*, S.M. Thesis, Department of Aeronautics and Astronautics, MIT, May 1988.

[7] Mendeck, G.F., Carman, G.L., "Guidance Design for Mars Smart Landers Using the Entry Terminal Point Controller," AIAA 2002-4502, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Monterey, CA, 5-8 Aug 2002.

[8] Ro, T.U., Queen, E.M. "Study of Martian Aerocapture Terminal Point Guidance," AIAA Paper 98-4571, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Boston, MA, 10-12 Aug 1998.

[9] Vallado, D.A., *Fundamentals of Astrodynamics and Applications*, The McGraw-Hill Companies, Inc., New York, 1997.

[10] *U.S. Standard Atmosphere, 1976*, National Oceanic and Atmospheric Administration— S/T 76-1562, Washington, D.C., October 1976.