

Copyright
by
Christina Tvrdik Chomel
2007

The Dissertation Committee for Christina Tvrdik Chomel
certifies that this is the approved version of the following dissertation:

**Development of an Analytical Guidance Algorithm for
Lunar Descent**

Committee:

Robert H. Bishop, Supervisor

Maruthi R. Akella

Timothy P. Crain

Wallace T. Fowler

David G. Hull

**Development of an Analytical Guidance Algorithm for
Lunar Descent**

by

Christina Tvrdik Chomel, B.S., S.M.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2007

Dedicated to my children William and Caroline.

Acknowledgments

An endeavor such as this is not completed without the support of many people. First, I would like to thank my advisor, Dr. Robert Bishop, for his vision that lead me to this research and his guidance through the four years of rough starts and dead ends that finally lead me down the right path. I would like to thank NASA for my first year of financial support. I learned a great deal about spacecraft guidance and navigation during my work on the Mars Science Laboratory Entry, Descent, and Landing program. I would like to thank the National Science Foundation for their three subsequent years of aid. This support allowed me the freedom to work on a very interesting problem without the pressure of programmatic issues burdening my research path.

I do not have the words to express my gratitude to the people who have challenged me, supported me, laughed with me, and cried with me through this process. Without people like Renato, Patricia, Jeremy, Chad, Kyle, and the numerous others who have touched my life, this work would not have been completed. You all motivated me at each step of this process and helped me see that I could accomplish the goal at hand. I will always be grateful for the support my parents and brother have provided as I have taken on each new challenge. Your love has been important through more than just this process. Most importantly, I would like to thank my husband and best friend, Justin. Your love and support have always been unfailing and appreciated even when I could not find the words to express what you mean to me.

Development of an Analytical Guidance Algorithm for Lunar Descent

Publication No. _____

Christina Tvrdik Chomel, Ph.D.
The University of Texas at Austin, 2007

Supervisor: Robert H. Bishop

In recent years, NASA has indicated a desire to return humans to the moon. With NASA planning manned missions within the next couple of decades, the concept development for these lunar vehicles has begun. The guidance, navigation, and control (GN&C) computer programs that will perform the function of safely landing a spacecraft on the moon are part of that development. The lunar descent guidance algorithm takes the horizontally oriented spacecraft from orbital speeds hundreds of kilometers from the desired landing point to the landing point at an almost vertical orientation and very low speed. Existing lunar descent GN&C algorithms date back to the Apollo era with little work available for implementation since then. Though these algorithms met the criteria of the 1960's, they are cumbersome today.

At the basis of the lunar descent phase are two elements: the targeting, which generates a reference trajectory, and the real-time guidance, which forces the spacecraft to fly that trajectory. The Apollo algorithm utilizes a complex, iterative, numerical optimization scheme for developing the reference trajectory. The real-time guidance utilizes this reference trajectory in the form of

a quartic rather than a more general format to force the real-time trajectory errors to converge to zero; however, there exist no guarantees under any conditions for this convergence. The proposed algorithm implements a purely analytical targeting algorithm used to generate two-dimensional trajectories “on-the-fly” or to retarget the spacecraft to another landing site altogether. It is based on the analytical solutions to the equations for speed, downrange, and altitude as a function of flight path angle and assumes two constant thrust acceleration curves. The proposed real-time guidance algorithm has at its basis the three-dimensional non-linear equations of motion and a control law that is proven to converge under certain conditions through Lyapunov analysis to a reference trajectory formatted as a function of downrange, altitude, speed, and flight path angle. The two elements of the guidance algorithm are joined in Monte Carlo analysis to prove their robustness to initial state dispersions and mass and thrust errors. The robustness of the retargeting algorithm is also demonstrated.

Table of Contents

Acknowledgments	v
Abstract	vi
List of Tables	x
List of Figures	xi
List of Symbols	xvii
List of Nomenclature	xx
Chapter 1. Introduction	1
1.1 Current Lunar Landing Guidance and the Need for an Analytical Methodology	2
1.1.1 Apollo Lunar Targeting and Guidance: Braking and Approach Phase	4
1.1.2 Need for an Analytical Guidance Algorithm	9
1.2 Description of Work	11
Chapter 2. Mathematical Models of Spacecraft Motion	13
2.1 Translational Dynamics Model	13
2.2 LVLH Frame Dynamics Model	20
Chapter 3. Target Trajectory Design	33
3.1 Constant Thrust Acceleration Solution	33
3.2 Targeting Algorithm Design	46

Chapter 4. Guidance Algorithm	61
4.1 Algorithm Design and Lyapunov Analysis	61
4.2 Performance in a Test Setting For Gain Selection	73
4.3 Performance in a Nominal Setting	85
4.3.1 Response to NASA Target Trajectory	85
4.3.2 Response to Proposed Target Trajectory	88
Chapter 5. Algorithm Simulation	92
5.1 Simulation Description and Results	92
5.2 Retargeting	108
Chapter 6. Conclusions	134
Bibliography	137
Vita	139

List of Tables

4.1	Real-Time Guidance Gains	74
5.1	Lunar Planetary Constants	93
5.2	Monte Carlo Simulation Perturbation Values	94
5.3	Monte Carlo Simulation Final State Error Statistics for Case Using Same Target Trajectory for Each Run	94
5.4	Monte Carlo Simulation Final State Error Statistics for Case Using Run Specific Target Trajectory for Each Run	102
5.5	Monte Carlo Simulation Final State Error Statistics for North-east Retargeting Case	122
5.6	Monte Carlo Simulation Final State Error Statistics for South-east Retargeting Case	128

List of Figures

1.1	Illustration of Lunar Landing [10]	2
1.2	Lunar Descent Reference Trajectory [5]	4
2.1	Maneuver Reference Frame Defined	16
2.2	Maneuver Reference Frame Definitions	18
2.3	Local Vertical-Local Horizontal Reference Frame	21
2.4	State and Force Vectors Defined In Local Vertical-Local Horizontal Frame	22
2.5	Downrange and Crossrange Definitions	25
2.6	Illustration of Effect of Planet Curvature on LVLH Relative States	28
2.7	Spherical Geometry For Describing Downrange and Crossrange	29
3.1	Comparison of Full Integrated Solution to Approximated Solution with Varying Values of k : Speed	39
3.2	Comparison of Full Integrated Solution to Approximated Solution with Varying Values of k : Time	40
3.3	Comparison of Full Integrated Solution to Approximated Solution with Varying Values of k : Altitude	40
3.4	Comparison of Full Integrated Solution to Approximated Solution with Varying Values of k : Downrange	41
3.5	Comparison of Analytical Target Trajectory Solutions To Numerical Solutions: Speed	44
3.6	Comparison of Analytical Target Trajectory Solutions To Numerical Solutions: Time	44
3.7	Comparison of Analytical Target Trajectory Solutions To Numerical Solutions: Altitude	45
3.8	Comparison of Analytical Target Trajectory Solutions To Numerical Solutions: Downrange	45
3.9	Sample Trajectory Space Varying $\frac{T}{m}$ Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}$	51

3.10	Three-Dimensional View of Downrange Sample Space	52
3.11	Three-Dimensional View of Altitude Sample Space	52
3.12	Sample Trajectory Space Varying $\frac{T}{m}$ and v_0 Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}; v_0 = [1800 : -100 : 1400] \frac{m}{sec}$	53
3.13	Sample Trajectory Space Varying $\frac{T}{m}$ and v_2 Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}; v_2 = [101 : -20 : 1] \frac{m}{sec}$	54
3.14	Sample Trajectory Space Varying $\frac{T}{m}$ and θ_0 Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}; \theta_0 = [0.1^\circ : 0.5^\circ : 5.1^\circ]$	54
3.15	Sample Trajectory Space Varying $\frac{T}{m}$ and θ_f Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}; \theta_f = [49^\circ : 2^\circ : 89^\circ]$	55
3.16	Sample Trajectory Space Varying $\frac{T}{m}$ and g Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}; g = [1 : -\frac{1}{20} : \frac{3}{4}] \frac{\mu}{h_0^2}$	56
3.17	Comparison Between Proposed and Existing Trajectory: Flight Path Angle	58
3.18	Comparison Between Proposed and Existing Trajectory: Altitude	58
3.19	Comparison Between Proposed and Existing Trajectory: Speed	59
3.20	Comparison Between Proposed and Existing Trajectory: Time	59
3.21	Comparison Between Proposed and Existing Trajectory: Acceleration	60
4.1	Real-Time Guidance Response to Test Trajectory: Commanded Acceleration vs. Downrange	77
4.2	Real-Time Guidance Response to Test Trajectory: Commanded Thrust Angle vs. Downrange	77
4.3	Real-Time Guidance Response to Test Trajectory: Downrange vs. Time	78
4.4	Real-Time Guidance Response to Test Trajectory: Altitude vs. Downrange	78
4.5	Real-Time Guidance Response to Test Trajectory: Flight Path Angle vs. Downrange	79
4.6	Real-Time Guidance Response to Test Trajectory: Speed vs. Downrange	79
4.7	Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Downrange vs. Time	80

4.8	Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Altitude vs. Downrange	81
4.9	Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Flight Path Angle vs. Downrange	81
4.10	Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Speed vs. Downrange	82
4.11	Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Commanded Acceleration vs. Downrange	82
4.12	Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Commanded Thrust Angle vs. Downrange . . .	83
4.13	Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Crossrange vs. Downrange	83
4.14	Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Crossing Angle vs. Downrange	84
4.15	Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Commanded Thrust Roll Angle vs. Downrange	84
4.16	Real-Time Guidance Response to NASA Target Trajectory: Downrange vs. Time	85
4.17	Real-Time Guidance Response to NASA Target Trajectory: Altitude vs. Downrange	86
4.18	Real-Time Guidance Response to NASA Target Trajectory: Speed vs. Downrange	86
4.19	Real-Time Guidance Response to NASA Target Trajectory: Flight Path Angle vs. Downrange	87
4.20	Real-Time Guidance Response to NASA Target Trajectory: Acceleration vs. Downrange	87
4.21	Real-Time Guidance Response to Proposed Target Trajectory: Downrange vs. Time	89
4.22	Real-Time Guidance Response to Proposed Target Trajectory: Altitude vs. Downrange	89
4.23	Real-Time Guidance Response to Proposed Target Trajectory: Flight Path Angle vs. Downrange	90
4.24	Real-Time Guidance Response to Proposed Target Trajectory: Speed vs. Downrange	90
4.25	Real-Time Guidance Response to Proposed Target Trajectory: Commanded Acceleration vs. Downrange	91
4.26	Real-Time Guidance Response to Proposed Target Trajectory: Commanded Thrust Angle vs. Downrange	91

5.1	Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Downrange vs. Time	95
5.2	Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Altitude vs. Downrange	95
5.3	Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Speed vs. Downrange	96
5.4	Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Flight Path Angle vs. Downrange	96
5.5	Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Crossrange vs. Downrange	97
5.6	Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Crossing Angle vs. Downrange	97
5.7	Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Commanded Acceleration vs. Downrange	98
5.8	Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Commanded Thrust Angle vs. Downrange	98
5.9	Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Commanded Thrust Roll Angle vs. Downrange	99
5.10	Sample Trajectory Space With Monte Carlo Limits	101
5.11	Monte Carlo Simulation Results With Run Specific Target Trajectory: Downrange vs. Time	103
5.12	Monte Carlo Simulation Results With Run Specific Target Trajectory: Altitude vs. Downrange	103
5.13	Monte Carlo Simulation Results With Run Specific Target Trajectory: Speed vs. Downrange	104
5.14	Monte Carlo Simulation Results With Run Specific Target Trajectory: Flight Path Angle vs. Downrange	104
5.15	Monte Carlo Simulation Results With Run Specific Target Trajectory: Crossrange vs. Downrange	105
5.16	Monte Carlo Simulation Results With Run Specific Target Trajectory: Crossing Angle vs. Downrange	105
5.17	Monte Carlo Simulation Results With Run Specific Target Trajectory: Commanded Acceleration vs. Downrange	106
5.18	Monte Carlo Simulation Results With Run Specific Target Trajectory: Commanded Thrust Angle vs. Downrange	106
5.19	Monte Carlo Simulation Results With Run Specific Target Trajectory: Commanded Thrust Roll Angle vs. Downrange	107

5.20	Retargeting (Northeast Example): Latitude vs. Longitude . . .	110
5.21	Retargeting (Northeast Example): Downrange vs. Time	111
5.22	Retargeting (Northeast Example): Altitude vs. Time	111
5.23	Retargeting (Northeast Example): Crossrange vs. Time	112
5.24	Retargeting (Northeast Example): Speed vs. Time	112
5.25	Retargeting (Northeast Example): Flight Path Angle vs. Time	113
5.26	Retargeting (Northeast Example): Crossing Angle vs. Time .	113
5.27	Retargeting (Northeast Example): Thrust Acceleration vs. Time	114
5.28	Retargeting (Northeast Example): Thrust Angle vs. Time . .	114
5.29	Retargeting (Northeast Example): Thrust Roll Angle vs. Time	115
5.30	Retargeting (Southeast Example): Latitude vs. Longitude . .	116
5.31	Retargeting (Southeast Example): Downrange vs. Time	117
5.32	Retargeting (Southeast Example): Altitude vs. Time	117
5.33	Retargeting (Southeast Example): Crossrange vs. Time	118
5.34	Retargeting (Southeast Example): Speed vs. Time	118
5.35	Retargeting (Southeast Example): Flight Path Angle vs. Time	119
5.36	Retargeting (Southeast Example): Crossing Angle vs. Time .	119
5.37	Retargeting (Southeast Example): Thrust Acceleration vs. Time	120
5.38	Retargeting (Southeast Example): Thrust Angle vs. Time . .	120
5.39	Retargeting (Southeast Example): Thrust Roll Angle vs. Time	121
5.40	Retargeting Monte Carlo Results (Northeast Example): Latitude vs. Longitude	123
5.41	Retargeting Monte Carlo Results (Northeast Example): Downrange vs. Time	123
5.42	Retargeting Monte Carlo Results (Northeast Example): Altitude vs. Time	124
5.43	Retargeting Monte Carlo Results (Northeast Example): Crossrange vs. Time	124
5.44	Retargeting Monte Carlo Results (Northeast Example): Speed vs. Time	125
5.45	Retargeting Monte Carlo Results (Northeast Example): Flight Path Angle vs. Time	125
5.46	Retargeting Monte Carlo Results (Northeast Example): Crossing Angle vs. Time	126

5.47	Retargeting Monte Carlo Results (Northeast Example): Thrust Acceleration vs. Time	126
5.48	Retargeting Monte Carlo Results (Northeast Example): Thrust Angle vs. Time	127
5.49	Retargeting Monte Carlo Results (Northeast Example): Thrust Roll Angle vs. Time	127
5.50	Retargeting Monte Carlo Results (Southeast Example): Latitude vs. Longitude	129
5.51	Retargeting Monte Carlo Results (Southeast Example): Down-range vs. Time	129
5.52	Retargeting Monte Carlo Results (Southeast Example): Altitude vs. Time	130
5.53	Retargeting Monte Carlo Results (Southeast Example): Cross-range vs. Time	130
5.54	Retargeting Monte Carlo Results (Southeast Example): Speed vs. Time	131
5.55	Retargeting Monte Carlo Results (Southeast Example): Flight Path Angle vs. Time	131
5.56	Retargeting Monte Carlo Results (Southeast Example): Crossing Angle vs. Time	132
5.57	Retargeting Monte Carlo Results (Southeast Example): Thrust Acceleration vs. Time	132
5.58	Retargeting Monte Carlo Results (Southeast Example): Thrust Angle vs. Time	133
5.59	Retargeting Monte Carlo Results (Southeast Example): Thrust Roll Angle vs. Time	133

List of Symbols

Symbol	Description
A	spacecraft thrust acceleration magnitude limit for guidance algorithm
\mathbf{a}	spacecraft acceleration vector
α	angle of thrust vector relative to $-\mathbf{v}$ in local vertical plane
β	bearing from north of nominal target trajectory
β_h	proportioning gain on altitude guidance channel: $0 \leq \beta_h \leq 1$
β_d	proportioning gain on downrange guidance channel: $0 \leq \beta_d \leq 1$
β_c	proportioning gain on crossrange guidance channel: $0 \leq \beta_c \leq 1$
c	range from target location to current spacecraft location projected along lunar surface perpendicular to desired bearing
d	range from target location to current spacecraft location projected along lunar surface parallel to desired bearing
Δ	downrange angle
e	planetary eccentricity
$\hat{\mathbf{e}}_1^{PCI}$	planet centered inertial reference frame unit vector 1
$\hat{\mathbf{e}}_2^{PCI}$	planet centered inertial reference frame unit vector 2
$\hat{\mathbf{e}}_3^{PCI}$	planet centered inertial reference frame unit vector 3
$\hat{\mathbf{e}}_1^{PCPF}$	planet centered-planet fixed reference frame unit vector 1
$\hat{\mathbf{e}}_2^{PCPF}$	planet centered-planet fixed reference frame unit vector 2
$\hat{\mathbf{e}}_3^{PCPF}$	planet centered-planet fixed reference frame unit vector 3
$\hat{\mathbf{e}}_1^{ROT}$	crossrange unit vector in crossrange-downrange-up reference frame rotated through downrange angle
$\hat{\mathbf{e}}_2^{ROT}$	downrange unit vector in crossrange-downrange-up reference frame rotated through downrange angle
$\hat{\mathbf{e}}_3^{ROT}$	up unit vector in crossrange-downrange-up reference frame rotated through downrange angle
$\hat{\mathbf{e}}_1^L$	local vertical-local horizontal reference frame unit vector 1
$\hat{\mathbf{e}}_2^L$	local vertical-local horizontal reference frame unit vector 2
$\hat{\mathbf{e}}_3^L$	local vertical-local horizontal reference frame unit vector 3
$\hat{\mathbf{e}}_1^M$	maneuver reference frame unit vector 1

$\hat{\mathbf{e}}_2^M$	maneuver reference frame unit vector 2
$\hat{\mathbf{e}}_3^M$	maneuver reference frame unit vector 3
f	planetary flattening
\mathbf{g}	gravity vector
g	gravity magnitude
Γ	crossrange angle
γ_h	gain on altitude guidance channel tuning function: $\gamma_h \geq 0$
γ_d	gain on downrange guidance channel tuning function: $\gamma_d \geq 0$
γ_c	gain on crossrange guidance channel tuning function: $\gamma_c \geq 0$
h	spacecraft altitude above the lunar surface
θ	angle between local horizontal and velocity vector or flight path angle
k	approximate proportion of centrifugal acceleration to gravity assumed over target trajectory
k_h	altitude guidance channel tuning function
k_d	downrange guidance channel tuning function
k_c	crossrange guidance channel tuning function
λ	longitude of current spacecraft position
λ_t	longitude of target location
λ_{h_1}	altitude guidance channel gain on altitude error: $\lambda_{h_1} \geq 0$
λ_{d_1}	downrange guidance channel gain on downrange error: $\lambda_{d_1} \geq 0$
λ_{c_1}	crossrange guidance channel gain on crossrange error: $\lambda_{c_1} \geq 0$
λ_{h_2}	altitude guidance channel gain on altitude rate error: $\lambda_{h_2} \geq 0$
λ_{d_2}	downrange guidance channel gain on downrange rate error: $\lambda_{d_2} \geq 0$
λ_{c_2}	crossrange guidance channel gain on crossrange rate error: $\lambda_{c_2} \geq 0$
μ	planetary gravitational constant
m	spacecraft mass
n	number of target trajectory segments
p	$\frac{T}{g}$
\mathbf{r}	spacecraft position vector
r	spacecraft position vector magnitude
\mathbf{r}_t	target position vector
t	time
\mathbf{T}	thrust vector
T	thrust magnitude
τ	torsion
\mathbf{u}	control vector

\mathbf{v}	spacecraft velocity vector
v	spacecraft velocity vector magnitude or spacecraft speed
ϕ	rotation angle of thrust vector as well as spacecraft rotation about $\hat{\mathbf{e}}_1^M$ vector
φ	geodetic latitude of current spacecraft position
φ'	geocentric latitude of current spacecraft position
φ_t	geodetic latitude of target location
φ'_t	geocentric latitude of target location
\mathbf{x}	state vector
ψ	angle between local vertical and velocity vector in local horizontal plane or crossing angle
R_{eq}	planetary equatorial radius
R_{pol}	planetary polar radius
$\boldsymbol{\omega}_I^M$	angular velocity vector of maneuver reference frame in inertial reference frame
$\boldsymbol{\omega}_I^L$	angular velocity vector of local vertical-local horizontal reference frame in inertial reference frame
$\boldsymbol{\omega}_L^M$	angular velocity vector of maneuver reference frame in local vertical-local horizontal reference frame

List of Nomenclature

<u>Nomenclature</u>	<u>Description</u>
\dot{x}	time derivative of scalar x
\ddot{x}	second time derivative of scalar x
$\dot{\mathbf{x}}$	time derivative of vector x
$\ddot{\mathbf{x}}$	second time derivative of vector x
$\dot{\mathbf{X}}$	time derivative of matrix X
$\frac{d^A}{dt}\mathbf{x}$	time derivative of vector \mathbf{x} relative to the A reference frame
\mathbf{T}_A^B	direction cosine matrix defining rotation from A reference frame to B reference frame
\times	vector cross product
\odot	vector inner product
$\ \mathbf{x}\ $	vector magnitude

Chapter 1

Introduction

In recent years, the National Aeronautics and Space Administration (NASA) has indicated a desire to return humans to the moon. With NASA planning manned missions within the next couple of decades, the concept development for these lunar vehicles has begun. The development of the hardware is generally a timely endeavor as it passes from conceptualization through the build and integration phase to the final testing. However, during this development, the computer programs that will control the hardware must be considered in order to allow for their development, testing, and integration with the system.

One important computer program will focus on the guidance, navigation, and control (GN&C) algorithms that will safely take the vehicle and its occupants from the Earth to a desired landing point on the lunar surface. The GN&C computer program will be segmented into mission phases: the outbound mission that goes from the Earth to the moon and the return mission that takes the vehicle safely back to the Earth. The components of the outbound mission are segmented into the launch phase, the transition from Earth orbit to lunar orbit, and the descent phase. The return mission will be similar; launch from the moon into a transition to Earth orbit and a safe entry, descent, and landing on the Earth. While much work has been done on the trajectory planning between the Earth and moon as well as in developing new algorithms

for the Earth entry, descent, and landing (EDL) segment, little reference material beyond the Apollo algorithms described in [1] and [5] is available that discusses new algorithms for the lunar descent and landing phase (more will be discussed about the work of McInnes ([7], [6]) and Uchiyama, et. al. ([11])). Therefore, this author selected the lunar descent guidance algorithm as the research focus discussed herein.

1.1 Current Lunar Landing Guidance and the Need for an Analytical Methodology

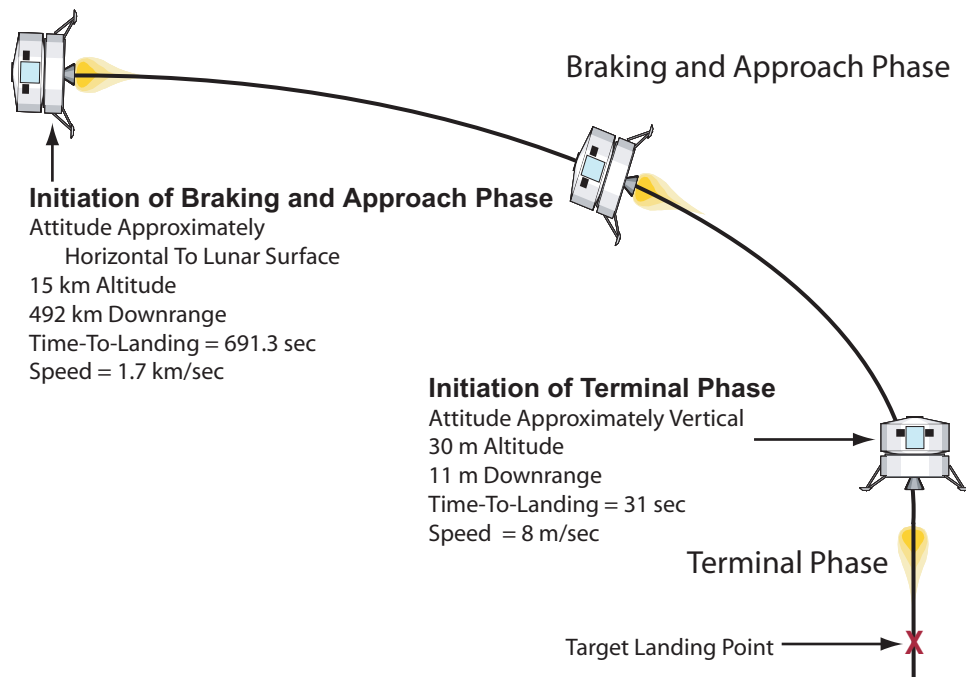


Figure 1.1: Illustration of Lunar Landing [10]

The Apollo guidance algorithms accomplished the mission of delivering men to the moon and safely returning them to Earth. They are therefore

generally being used as the basis for these future missions. The lunar landing guidance algorithm consists of two phases as illustrated in Fig. 1.1. The braking and approach phase uses variable thrust to reduce the velocity from orbital speeds, pitch the vehicle over, and throttle down while targeting a point directly over the landing site. In the terminal phase, the vehicle uses constant thrust to descend vertically to the landing site. The braking and approach phase guidance algorithm is the focus of this dissertation.

Before an in-depth discussion of the Apollo algorithms is begun, another area of exploration that is significantly different from the Apollo algorithm should be noted. This is in the development of an analytically based gravity-turn descent by McInnes as explained in [7]. The analysis makes constant thrust-to-mass ratio and gravity assumptions thereby allowing for an analytical solution for speed as a function of flight path angle. However, the assumptions made around the dynamic nature of the centrifugal acceleration from orbit to landing lead to a solution for speed that is defined by a series that is truncated. Additionally, McInnes does not develop the full analytical solutions for all the states necessary for defining the reference trajectory, i.e. translational states of altitude and range to the landing point, and does not present an algorithm for defining the trajectory or performing any in-flight retargeting. The control laws discussed in [6] and [11] describe non-linear algorithms for controlling to the reference trajectory. Both support their convergence to the reference trajectory with Lyapunov analysis but only in the two dimensions of the reference trajectory. Both fail to develop a full three-dimensional real-time guidance law that will guarantee convergence to the desired landing point.

1.1.1 Apollo Lunar Targeting and Guidance: Braking and Approach Phase

The Apollo lunar braking and approach phase guidance algorithm has two main elements. The first element is the targeting which generates a reference trajectory. The reference trajectory is the position and velocity of the spacecraft as a function of time-to-go to the landing site. The vehicle should ideally follow the reference trajectory from orbit to the initiation of the terminal phase. For the Apollo missions, the reference trajectory was designed pre-mission on the ground and then loaded aboard the vehicle. It is attached at its endpoint to a location on the lunar surface and is generated using a complex, iterative, numerical optimization scheme. Due to its complexity, the optimization algorithm cannot be readily implemented on board the spacecraft to update the reference trajectory in real-time.

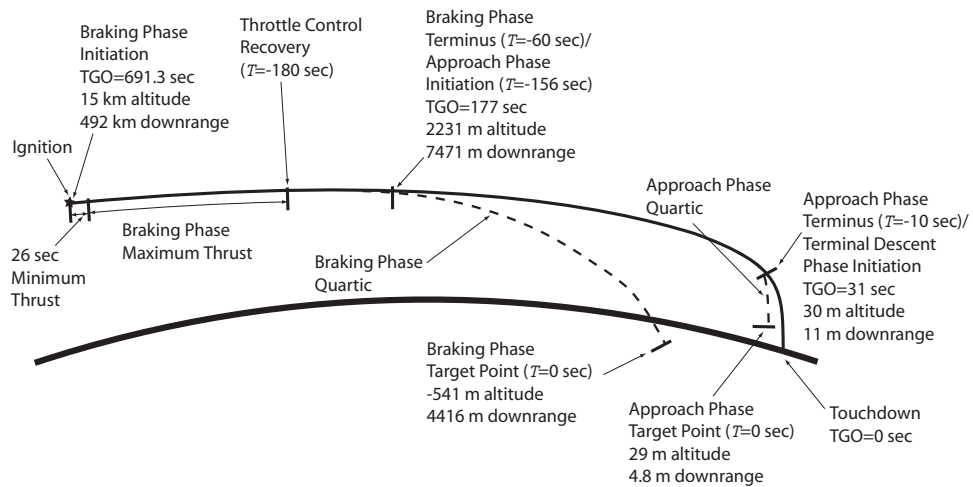


Figure 1.2: Lunar Descent Reference Trajectory [5]

A description of the Apollo targeting program given in [5] is illustrated in Fig. 1.2. The values for TGO in the figure are typical mission times from

the specified point to expected touchdown at the end of the terminal descent phase. At the basis of the targeting program are two phases: the braking phase (P63) and the approach phase (P64). The P63 targeting must satisfy a hardware induced throttling constraint such that the thrusters are operated at maximum thrust for most of their operation but are throttled down to stay within 11% and 65% of maximum thrust for the last two minutes of P63. The P64 reference trajectory is computed to provide lunar surface visibility until about 5 seconds before the phase end and allow for a smooth transition to the terminal phase.

The methodology for this targeting algorithm is such that the reference is built from the terminal phase up. The approach phase (P64) target position and velocity are selected and are referenced to time $T = 0$; this position is not the actual terminus point of this phase but a point approximately 10 seconds beyond the phase terminus (this was done to avoid issues with T approaching zero in the real-time guidance). The trajectory is then built backwards in time such that all time references are negative (this is the “ T ” in Fig. 1.2). The reference trajectory curve takes the form of a quartic. According to Klumpp [5], this methodology was chosen in order to define a reference trajectory that satisfied “a two-point boundary value problem with a total of five degrees of freedom for each of the three components. This number of degrees of freedom was required in order to constrain terminal thrust in P63 and to shape the trajectory design in P64”. The reference trajectory is defined by the quartic vector polynomial

$$\text{RRG} = \text{RTG} + \text{VTGT} + \text{ATG}\frac{T^2}{2} + \text{JTG}\frac{T^3}{6} + \text{STG}\frac{T^4}{24} \quad (1.1)$$

where RRG is the position vector on the reference trajectory at the current (negative) time T and RTG, VTG, ATG, JTG, and STG, are the targeted

position, velocity, acceleration, jerk, and snap. These states are all given in a guidance reference frame that has its origin at the specified landing point.

Because of the quartic form for the approach phase reference trajectory, fifteen constraints must be specified (five coefficients in three dimensions). First, the trajectory is confined to two dimensions, thereby reducing the unknown variables to ten. Then, according to Klumpp, if the approach phase target point, velocity, and time are selected along with a phase inception position and time and an additional midpoint position, velocity, and time, there exists a closed form solution for the remaining ten unknown variables that define RTG, VTG, ATG, JTG, and STG. The equations are given in [5] where the states for the approach phase target point, velocity, and time, the phase inception position, and time and the midpoint position, velocity, and time are called the constraint set for the problem.

One important comment that Klumpp makes is that, though there exists a closed form solution for the reference trajectory coefficients given a constraint set, the closed form solutions for a number of constraint sets are generated. The constraint set that produces an approach phase trajectory with an acceptable thrust profile and adequate visibility for the pilot to perform retargeting is selected. This implies that the solution for the approach phase is not automatic but requires the judgement of the trajectory designer.

With the design of the approach phase reference trajectory in place, the braking phase (P63) reference is designed. This phase has a reference trajectory of the same quartic form as the approach phase, again confined to be in a plane. This implies that ten constants must be specified in RTG, VTG, ATG, JTG, and STG. According to Klumpp, seven of the constants can be determined in closed-form, but the remaining three must be computed by iter-

atively running a simulation from the braking phase inception point through to the braking phase terminus. This simulation essentially builds the trajectory that must take the spacecraft from orbit through the 26 second minimum thrust segment shown in Fig. 1.2 through the approach phase inception point down to the braking phase target point (again note that the braking phase target point is beyond the braking phase terminus so that reference time does not approach zero once this reference trajectory is invoked in the real-time guidance). The simulation is run iteratively until a set of three convergence tests are passed (recall seven of the unknown states are computed from a closed form solution so only three states are sought in this iterative fashion). The equations that define the braking phase reference trajectory are given in [5].

One interesting item of note is that the braking phase reference trajectory is such that the majority of its commanded thrust is above the maximum thrust allowed. This meets the requirement that the spacecraft use maximum thrust during most of this phase. At some point, the reference thrust transitions to values that are in the allowable range of 11% and 65% of maximum thrust; this is the throttle control recovery point in Fig. 1.2

The second element of the Apollo lunar braking and approach phase guidance algorithm is the real-time guidance. Real-time guidance generates the thrust commands to fly the reference trajectory. A description of the Apollo guidance is given in [1] and [5]. The relevant elements of this algorithm are presented below.

The Apollo real-time guidance element forces the vehicle trajectory to converge to a reference trajectory defined by the quartic vector polynomial in Eq. 1.1. The goal of the real-time guidance element is to compute a com-

manded acceleration that is based on the current state error from the reference position. This commanded acceleration is passed to the control algorithm that generates the thrust command. The commanded acceleration vector, ACG, is given by taking the derivative of Eq. 1.1:

$$\begin{aligned}
\text{ACG} &= \text{ATG} + \text{JTGT} + \text{STG} \frac{T^2}{2} & (1.2) \\
&- \left(\text{VG} - \text{VTG} - \text{ATGT} - \text{JTG} \frac{T^2}{2} - \text{STG} \frac{T^3}{6} \right) \frac{K_V}{T} \\
&- \left(\text{RG} - \text{RTG} - \text{VTGT} - \text{ATG} \frac{T^2}{2} - \text{JTG} \frac{T^3}{6} - \text{STG} \frac{T^4}{24} \right) \frac{K_R}{T^2}
\end{aligned}$$

where VG and RG are the current velocity and position vectors and K_V and K_R are the non-dimensional scalar gains on the errors between the current and expected velocity and position states. In [5] Klumpp describes a basis for choosing the gains based on response time and damping, and settles on values of $K_V = -6$ and $K_R = 12$. These are substituted into Eq. 1.2 yielding

$$\text{ACG} = \text{ATG} + 6(\text{VTG} + \text{VG}) \frac{1}{T} + 12(\text{RTG} - \text{RG}) \frac{1}{T^2} \quad (1.3)$$

Equation 1.3 would be the final form for the guidance output except that system delays were significant for this algorithm and had to be accounted for. Therefore the predicted time-to-go, T_P , is the current time-to-go with a lead time added to it. The lead time accounts for the transport delay from the time the command is computed to the time it is executed by the system. The acceleration of the quartic at the predicted time T_P is the acceleration that must be commanded at T in order to realize the quartic at the future time. Though this algorithm that accounts for the transport delay is critical to the Apollo implementation, it is not relevant to this discussion. It does not change the overall guidance concept.

As discussed previously, the Apollo landing is comprised of the approach phase (P63) and the braking phase (P64). Both phases use this same guidance algorithm with differences in the reference constants. In addition, the braking phase allows for a landing site redesignation. The landing site redesignation algorithm allows the pilot to visually “steer” the vehicle. The pilot visually identifies the current landing site and then steers the vehicle onto a course coincident with another visually selected landing site. This procedure is iterative as the pilot commands incremental displacements or redesignations until the goal has been achieved. These incremental angles are in fixed one degree increments with respect to the current line of sight but can be oriented left, right, forward, or aft. The one degree incremental changes are fed into the guidance algorithm by computationally changing the landing site value used in the remainder of the guidance algorithm.

1.1.2 Need for an Analytical Guidance Algorithm

The Apollo approach and braking lunar guidance algorithm worked well for the intended missions in the 1960’s. However, key shortcomings must be addressed to support precision landings envisioned in the future. The goal of these future missions will be to reduce cost as safely as possible; reducing the complexity of developing the descent trajectories will certainly reduce pre-flight analysis cost. Apollo targeting is numerically complex and cumbersome. Though the approach phase element does have a closed form solution, it requires the user to interpret the data and select an appropriate reference trajectory. The fact that the design of the braking phase requires iterative simulation runs to select the reference trajectory is undesirable because no guarantee of convergence to a viable solution is guaranteed. Because of these

points, the Apollo design is not realistically implementable in real time and has the burden of requiring pre-flight analysis time to set an appropriate reference.

Additionally, the real-time guidance algorithm that removes the state errors requires that the reference trajectory be generated in the form of the set of quartic polynomials. It does not take in any generic trajectory and make it usable. Finally, the development of the real-time guidance algorithm does not provide any proof of guaranteed convergence to the reference trajectory under any conditions. In fact, the thrust limiting that is part of the design in the first segment of the braking phase means that the only control authority available is in changing the orientation of the thrust angle which provides no guarantee of convergence to the reference trajectory.

Any algorithm developed to replace the Apollo algorithms can have at its core the same element structure: a targeting element and a real-time guidance element. The targeting can then be separate from the real-time guidance so that new trajectories can be developed and either used or discarded based on some set of criteria. However, some requirements are necessary to make this algorithm more useful than what is already existing and proven to work.

The targeting algorithm must be analytical in nature so that 1.) reference trajectories are easily generated without intense pre-flight analysis and 2.) the reference trajectory can be easily and reliably updated at any point in a mission. This will allow for retargeting so that a new mission objective can be selected once the descent has started without the optical requirements necessary for Apollo; a new target location is simply selected and its coordinates are entered into the algorithm. In addition, the trajectory format required for the real-time guidance should be rather generic in nature so it

can easily be used for any trajectory (i.e., an Apollo-like algorithm trajectory can be inserted into this algorithm). This requires a whole new design that is completely different from the Apollo algorithm's dependence on a trajectory fitting the two quartic polynomials. An additional desire is that the real-time guidance have some proof of convergence given some set of conditions.

1.2 Description of Work

The proposed algorithm will begin with the equations of motion that are the necessary basis for any guidance algorithm development in Chapter 2. These equations are formulated in a reference frame that lends itself to straightforward development of the targeting and guidance algorithms. The equations of motion that define the downrange, altitude, crossrange, speed, flight path angle, and crossing angle progressions with time are given in their fullest form.

The targeting algorithm can be developed from these equations, as discussed in Chapter 3. The equations in Chapter 2 are used along with some assumptions that confine the motion to a single plane to further reduce the complexity of the equations of motion. Finally, some assumptions are made that force the thrust-to-mass ratio and gravity to be constant so that the equations can be solved analytically similar to that done in [7]. These analytical equations are derived and presented and a discussion of real-time implementation issues is included. A comparison of an analytically computed reference trajectory is provided and is compared to a similar trajectory proposed by NASA engineers.

The equations of motion in Chapter 2 are again utilized in their full 3-dimensional form to develop the real-time guidance algorithm described in

Chapter 4. This algorithm uses as its basis a 2-dimensional trajectory defined in terms of downrange, altitude, speed, and flight path as defined with respect to any landing site and bearing to that landing site. The analysis that proves its asymptotic convergence is presented along with a demonstration of the algorithm performance given a stressful test target trajectory. A demonstration of the algorithm's performance given the trajectory defined by NASA engineers is also provided along with the performance given the proposed reference trajectory defined in Chapter 3.

Chapter 5 demonstrates the algorithm performance given the off-nominal realistic environment that the vehicle is sure to encounter in the form of Monte Carlo analysis. Finally, Chapter 5 will illustrate the algorithm performance given retargeting. Chapter 6 contains a synopsis of the algorithm points and its performance along with some recommendations for future work.

Chapter 2

Mathematical Models of Spacecraft Motion

2.1 Translational Dynamics Model

The first step in developing a guidance algorithm is defining the translational equations that describe the motion of the spacecraft. The mathematical model has the general form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2.1)$$

where $\mathbf{x} = [\mathbf{r} \quad \mathbf{v}]^T \in \mathfrak{R}^6$, $\mathbf{u} \in \mathfrak{R}^3$ is a control vector, and

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{r}) + \mathbf{a}(\mathbf{x}, \mathbf{u}) \end{bmatrix} \quad (2.2)$$

where the vector function $\mathbf{f}(\cdot)$ is continuously differentiable. The spacecraft position and velocity vectors represented in lunar centered inertial coordinates are given by

$$\begin{aligned} \mathbf{r} &= [r_1 \quad r_2 \quad r_3]^T \\ \mathbf{v} &= [v_1 \quad v_2 \quad v_3]^T \end{aligned}$$

Lunar centered inertial coordinates define an inertial reference frame centered at the lunar center of mass and described by the unit vectors $\hat{\mathbf{e}}_1^{PCI}$, $\hat{\mathbf{e}}_2^{PCI}$, and $\hat{\mathbf{e}}_3^{PCI}$, where $\hat{\mathbf{e}}_1^{PCI}$; the exact orientation of these vectors is inconsequential to this discussion. The accelerations due to gravity and spacecraft motor thrust are given by the vectors \mathbf{g} and \mathbf{a} , respectively.

In order to simplify the development of the guidance law, the equations of motion can be developed in a maneuver reference frame defined relative to the lunar surface. It is important to understand the underlying assumptions. Additionally, in anticipation of defining the trajectory targeting scheme, the assumptions necessary to confine the motion of the maneuver frame to two dimensions are necessary.

One method to describe the motion of the maneuver reference frame to be two-dimensional is to force the torsion of the trajectory to be zero [2]. By definition the torsion is [8]:

$$\tau = \frac{(\mathbf{v} \times \dot{\mathbf{v}}) \odot \ddot{\mathbf{v}}}{\|\mathbf{v} \times \dot{\mathbf{v}}\|^2} \quad (2.3)$$

where “ \times ” denotes the vector cross product and “ \odot ” denotes the vector inner product. To understand the implications of zero torsion, we consider $\ddot{\mathbf{v}}$ more closely.

From Eq. 2.2, we have

$$\dot{\mathbf{v}} = \mathbf{g} + \mathbf{a} \quad (2.4)$$

Premultiplying Eq. 2.2 by the direction cosine matrix that describes the rotation from the inertial reference frame to the maneuver reference frame, \mathbf{T}_I^M , yields

$$\mathbf{T}_I^M (\dot{\mathbf{v}} - \mathbf{g}) = \mathbf{T}_I^M \mathbf{a} \quad (2.5)$$

Taking the time derivative of Eq. 2.5 it follows that

$$\dot{\mathbf{T}}_I^M (\dot{\mathbf{v}} - \mathbf{g}) + \mathbf{T}_I^M (\ddot{\mathbf{v}} - \dot{\mathbf{g}}) = \dot{\mathbf{T}}_I^M \mathbf{a} + \mathbf{T}_I^M \dot{\mathbf{a}} \quad (2.6)$$

If $\boldsymbol{\nu}$ is any 3-dimensional vector defined in inertial space, we have

$$\mathbf{T}_M^I \dot{\mathbf{T}}_I^M \boldsymbol{\nu} = -\boldsymbol{\omega}_I^M \times \boldsymbol{\nu}$$

where $\boldsymbol{\omega}_I^M$ is the rotational rate of the maneuver frame relative to the inertial frame and represented in the inertial frame. Equation 2.6 then reduces to

$$-\mathbf{T}_I^M (\boldsymbol{\omega}_I^M \times (\dot{\mathbf{v}} - \mathbf{g})) + \mathbf{T}_I^M (\ddot{\mathbf{v}} - \dot{\mathbf{g}}) = \mathbf{T}_I^M (-\boldsymbol{\omega}_I^M \times \mathbf{a}) + \mathbf{T}_I^M \dot{\mathbf{a}} \quad (2.7)$$

Premultiplying Eq. 2.7 by \mathbf{T}_M^I and rearranging yields

$$\ddot{\mathbf{v}} = \boldsymbol{\omega}_I^M \times (\dot{\mathbf{v}} - \mathbf{g}) + \dot{\mathbf{g}} + \dot{\mathbf{a}} - \boldsymbol{\omega}_I^M \times \mathbf{a} \quad (2.8)$$

Given the relationship [4]

$$\dot{\mathbf{a}} = \frac{d^M}{dt} \mathbf{a} + \boldsymbol{\omega}_I^M \times \mathbf{a}$$

it follows that Eq. 2.8 can be written as

$$\ddot{\mathbf{v}} = \boldsymbol{\omega}_I^M \times (\dot{\mathbf{v}} - \mathbf{g}) + \dot{\mathbf{g}} + \frac{d^M}{dt} \mathbf{a}$$

Now the details of $\boldsymbol{\omega}_I^M$ must be further examined.

From fundamental kinematics described in [4], we have

$$\dot{\hat{\mathbf{e}}}_1^M = \boldsymbol{\omega}_I^M \times \hat{\mathbf{e}}_1^M \quad (2.9)$$

where $\hat{\mathbf{e}}_1^M$ is the unit vector in Fig. 2.1 that describes the orientation of the 1-axis of the maneuver frame in inertial space. If Eq. 2.9 is multiplied through by $\hat{\mathbf{e}}_1^M$, then

$$\begin{aligned} \hat{\mathbf{e}}_1^M \times \dot{\hat{\mathbf{e}}}_1^M &= \hat{\mathbf{e}}_1^M \times (\boldsymbol{\omega}_I^M \times \hat{\mathbf{e}}_1^M) \\ &= (\hat{\mathbf{e}}_1^M \odot \hat{\mathbf{e}}_1^M) \boldsymbol{\omega}_I^M - (\hat{\mathbf{e}}_1^M \odot \boldsymbol{\omega}_I^M) \hat{\mathbf{e}}_1^M \end{aligned} \quad (2.10)$$

Define the roll angle, ϕ , as

$$\dot{\phi} \triangleq \boldsymbol{\omega}_I^M \odot \hat{\mathbf{e}}_1^M$$

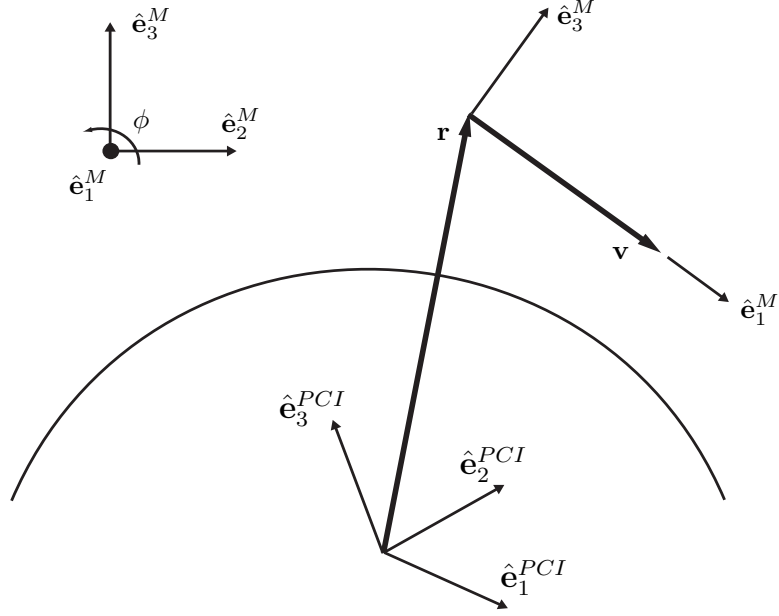


Figure 2.1: Maneuver Reference Frame Defined

From Eq. 2.10, it follows that

$$\boldsymbol{\omega}_I^M = \hat{\mathbf{e}}_1^M \times \dot{\hat{\mathbf{e}}}_1^M + \dot{\phi} \hat{\mathbf{e}}_1^M \quad (2.11)$$

The maneuver frame is defined such that

$$\hat{\mathbf{e}}_1^M = \frac{\mathbf{v}}{v}$$

where v is the magnitude of \mathbf{v} . The time derivative of $\hat{\mathbf{e}}_1^M$ is given by

$$\dot{\hat{\mathbf{e}}}_1^M = \frac{\dot{\mathbf{v}}}{v} - \frac{\mathbf{v} \odot \dot{\mathbf{v}}}{v^3} \mathbf{v} \quad (2.12)$$

Substituting Eq. 2.12 into Eq. 2.11 yields

$$\boldsymbol{\omega}_I^M = \frac{\mathbf{v} \times \dot{\mathbf{v}}}{v^2} + \dot{\phi} \hat{\mathbf{e}}_1^M$$

Using ω_I^M in Eq. 2.11 in Eq. 2.9 leads to

$$\ddot{\mathbf{v}} = \frac{\mathbf{v} \times \dot{\mathbf{v}}}{v^2} \times (\dot{\mathbf{v}} - \mathbf{g}) + \dot{\phi} \hat{\mathbf{e}}_1^M \times (\dot{\mathbf{v}} - \mathbf{g}) + \dot{\mathbf{g}} + \frac{d^M}{dt} \mathbf{a} \quad (2.13)$$

Equation 2.13 is substituted into the torsion equation (Eq. 2.3) yielding:

$$\tau = \frac{(\mathbf{v} \times \dot{\mathbf{v}})}{\|\mathbf{v} \times \dot{\mathbf{v}}\|^2} \odot \left(\frac{\mathbf{v} \times \dot{\mathbf{v}}}{v^2} \times (\dot{\mathbf{v}} - \mathbf{g}) + \dot{\phi} \hat{\mathbf{e}}_1^M \times (\dot{\mathbf{v}} - \mathbf{g}) + \dot{\mathbf{g}} + \frac{d^M}{dt} \mathbf{a} \right) \quad (2.14)$$

The first term in Eq. 2.14 is zero by the fact that the two terms $(\mathbf{v} \times \dot{\mathbf{v}})$ and $(\mathbf{v} \times \dot{\mathbf{v}} \times (\dot{\mathbf{v}} - \mathbf{g}))$ are perpendicular. If the assumption is made that $\dot{\phi}$ is zero, Eq. 2.14 reduces to

$$\tau = \frac{(\mathbf{v} \times \dot{\mathbf{v}})}{\|\mathbf{v} \times \dot{\mathbf{v}}\|^2} \odot \left(\dot{\mathbf{g}} + \frac{d^M}{dt} \mathbf{a} \right) \quad (2.15)$$

If the additional assumptions are made that the acceleration with respect to the maneuver frame is constant and that gravity is constant as was done in [2], then the torsion is zero. This yields a trajectory that remains in a two dimensional plane, i.e., the maneuver plane. For some applications, such as the aircraft tracking problem in [2], these assumptions lead to a fixed maneuver plane and are often valid. For the spacecraft problem, the assumptions are too restrictive.

The remainder of the maneuver frame unit vectors are defined as follows:

$$\hat{\mathbf{e}}_2^M = -\frac{\hat{\mathbf{e}}_1^M \times \mathbf{r}}{\|\hat{\mathbf{e}}_1^M \times \mathbf{r}\|} \quad (2.16)$$

$$\hat{\mathbf{e}}_3^M = \hat{\mathbf{e}}_1^M \times \hat{\mathbf{e}}_2^M \quad (2.17)$$

The acceleration due to thrust is defined by

$$\mathbf{a} = -\frac{T}{m} \cos \alpha \hat{\mathbf{e}}_1^M + \frac{T}{m} \sin \alpha (-\sin \phi \hat{\mathbf{e}}_2^M + \cos \phi \hat{\mathbf{e}}_3^M) \quad (2.18)$$

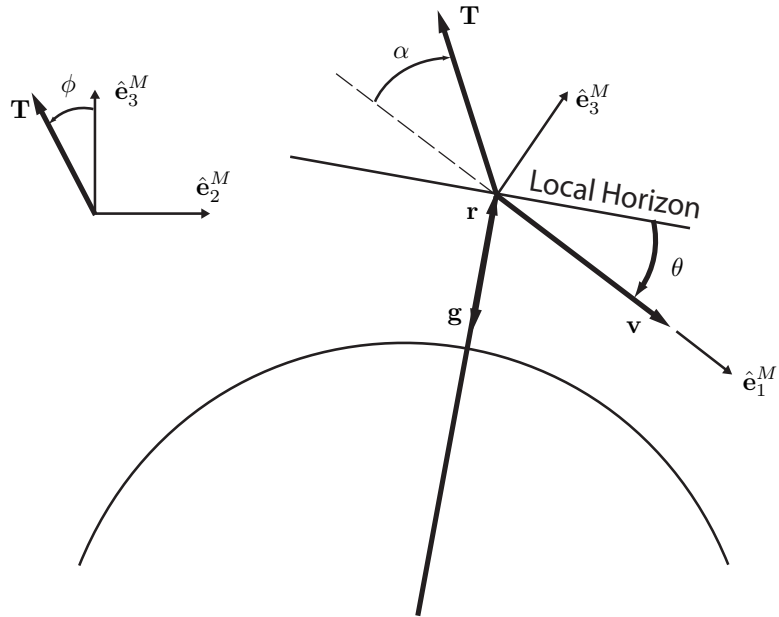


Figure 2.2: Maneuver Reference Frame Definitions

where m is the spacecraft mass, T is the thrust magnitude, and α is the angle at which the thrust vector is oriented with respect to the velocity vector Fig. 2.2. The assumption is made that the moon is perfectly spherical and homogeneous so that the acceleration due to gravity is defined by

$$\mathbf{g} = g \sin \theta \hat{\mathbf{e}}_1^M - g \cos \theta \hat{\mathbf{e}}_3^M \quad (2.19)$$

where g is the magnitude of \mathbf{g} and θ is the angle between local horizontal and

the velocity vector. The derivatives of \mathbf{a} and \mathbf{g} are given by

$$\begin{aligned}\frac{d^M}{dt}\mathbf{a} &= \left(-\frac{d}{dt}\left(\frac{T}{m}\right)\cos\alpha + \frac{T}{m}\dot{\alpha}\sin\alpha\right)\hat{\mathbf{e}}_1^M \\ &+ \left(\frac{d}{dt}\left(\frac{T}{m}\right)\sin\alpha + \frac{T}{m}\dot{\alpha}\cos\alpha\right)(-\sin\phi\hat{\mathbf{e}}_2^M + \cos\phi\hat{\mathbf{e}}_3^M) \\ &+ \left(\frac{T}{m}\sin\alpha\right)\dot{\phi}(-\cos\phi\hat{\mathbf{e}}_2^M - \sin\phi\hat{\mathbf{e}}_3^M)\end{aligned}\quad (2.20)$$

$$\begin{aligned}\dot{\mathbf{g}} &= \left(\dot{g}\sin\theta + g\dot{\theta}\cos\theta\right)\hat{\mathbf{e}}_1^M - \left(\dot{g}\cos\theta - g\dot{\theta}\sin\theta\right)\hat{\mathbf{e}}_3^M \\ &+ g\sin\theta\dot{\hat{\mathbf{e}}}_1^M - g\cos\theta\dot{\hat{\mathbf{e}}}_3^M\end{aligned}\quad (2.21)$$

Substituting Eqs. 2.20, 2.21, and 2.4 into Eq. 2.15, the torsion term takes the general form

$$\tau = f_1\left(\frac{T}{m}, \alpha, g, \theta, \frac{d}{dt}\left(\frac{T}{m}\right), \dot{\alpha}\right)\sin\phi + f_2\left(\frac{T}{m}, \alpha, g, \theta, \frac{d}{dt}\left(\frac{T}{m}\right), \dot{\alpha}\right)\dot{\phi}$$

where $f_1\left(\frac{T}{m}, \alpha, g, \theta, \frac{d}{dt}\left(\frac{T}{m}\right), \dot{\alpha}\right)$ and $f_2\left(\frac{T}{m}, \alpha, g, \theta, \frac{d}{dt}\left(\frac{T}{m}\right), \dot{\alpha}\right)$ are scalar functions that can be non-zero when $\sin\phi = 0$ or $\dot{\phi} = 0$, respectively. The torsion will go to zero when the roll angle, ϕ , and its derivative, $\dot{\phi}$, equal zero. This constrains the motion to the $\hat{\mathbf{e}}_1^M/\hat{\mathbf{e}}_3^M$ plane. It follows that the maneuver frame is oriented such that the center of the planet is contained in the maneuver plane and the gravity is along the position vector.

Note that this is an intuitive result - if the spacecraft motion is initiated in the $\hat{\mathbf{e}}_1^M/\hat{\mathbf{e}}_3^M$ plane and no acceleration is allowed in the $\hat{\mathbf{e}}_2^M$ direction, the spacecraft motion will remain in the $\hat{\mathbf{e}}_1^M/\hat{\mathbf{e}}_3^M$ plane. However, this proof is presented for two reasons. First, to motivate the selection of the maneuver frame unit vector definitions. Second, to prove that, if the statement is made that $\phi = \dot{\phi} = 0$, no acceleration is present in the $\hat{\mathbf{e}}_2^M$. This will become important when defining the reference trajectory in Chapter 3.

2.2 LVLH Frame Dynamics Model

The maneuver frame from Section 2.1 is utilized here to redefine the equations of motion in a more intuitive format. Note that these equations will describe the full three-dimensional motion of the spacecraft; the requirements that convert these equations to their two-dimensional form have already been discussed.

Combining Eqs. 2.18 and 2.19 with Eq. 2.4, the time derivative of the velocity vector is given by

$$\begin{aligned}
 \dot{\mathbf{v}} &= \mathbf{g} - \frac{T}{m} \cos \alpha \hat{\mathbf{e}}_1^M + \frac{T}{m} \sin \alpha (-\sin \phi \hat{\mathbf{e}}_2^M + \cos \phi \hat{\mathbf{e}}_3^M) \\
 &= \left(g \sin \theta - \frac{T}{m} \cos \alpha \right) \hat{\mathbf{e}}_1^M + \left(-\frac{T}{m} \sin \alpha \sin \phi \right) \hat{\mathbf{e}}_2^M \\
 &\quad + \left(\frac{T}{m} \sin \alpha \cos \phi - g \cos \theta \right) \hat{\mathbf{e}}_3^M
 \end{aligned} \tag{2.22}$$

where θ is the angular orientation of the velocity vector from local horizontal and is called the flight path angle. With

$$\mathbf{v} = v \hat{\mathbf{e}}_1^M$$

it follows that

$$\dot{\mathbf{v}} = \dot{v} \hat{\mathbf{e}}_1^M + v \dot{\hat{\mathbf{e}}}_1^M = \dot{v} \hat{\mathbf{e}}_1^M + v (\boldsymbol{\omega}_I^M \times \hat{\mathbf{e}}_1^M) \tag{2.23}$$

The angular velocity vector, $\boldsymbol{\omega}_I^M$, is defined as

$$\boldsymbol{\omega}_I^M = \boldsymbol{\omega}_I^L + \boldsymbol{\omega}_L^M \tag{2.24}$$

where $\boldsymbol{\omega}_I^L$ is the angular velocity of the local vertical-local horizontal (LVLH) frame with respect to the inertial frame shown in Fig. 2.3. The vector $\boldsymbol{\omega}_L^M$

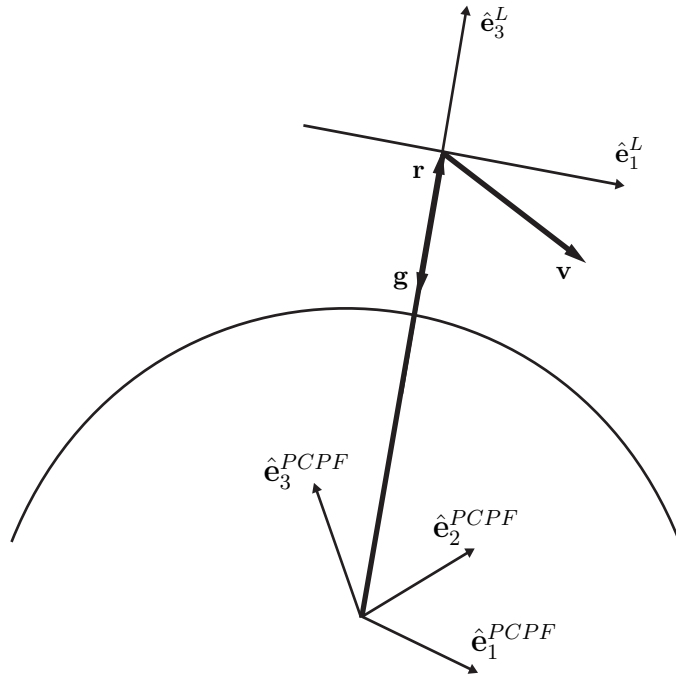


Figure 2.3: Local Vertical-Local Horizontal Reference Frame

represents the angular velocity of the maneuver reference frame with respect to the LVLH frame.

The LVLH reference frame is defined by the unit vectors $\hat{\mathbf{e}}_1^L$, $\hat{\mathbf{e}}_2^L$, and $\hat{\mathbf{e}}_3^L$ given by

$$\begin{aligned}\hat{\mathbf{e}}_3^L &= \frac{\mathbf{r}}{r} \\ \hat{\mathbf{e}}_2^L &= \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|} \\ \hat{\mathbf{e}}_1^L &= \hat{\mathbf{e}}_2^L \times \hat{\mathbf{e}}_3^L\end{aligned}$$

The LVLH reference frame is rotated to maintain an artificial horizon; the maneuver frame can be defined relative to the LVLH frame. Figure 2.4 shows

the states and forces with respect to the LVLH reference frame; it also shows the relationship of the maneuver frame to the LVLH unit vectors.

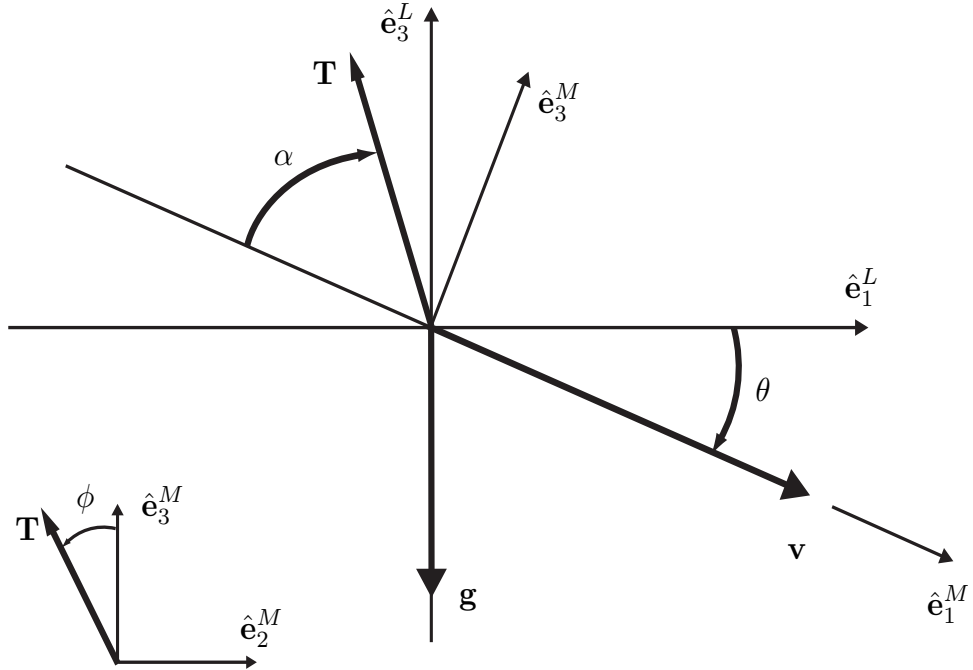


Figure 2.4: State and Force Vectors Defined In Local Vertical-Local Horizontal Frame

Given [4]

$$\dot{\hat{\mathbf{e}}}_3^L = \boldsymbol{\omega}_I^L \times \hat{\mathbf{e}}_3^L$$

it follows that

$$\begin{aligned} \hat{\mathbf{e}}_3^L \times \dot{\hat{\mathbf{e}}}_3^L &= \hat{\mathbf{e}}_3^L \times (\boldsymbol{\omega}_I^L \times \hat{\mathbf{e}}_3^L) \\ &= (\hat{\mathbf{e}}_3^L \odot \hat{\mathbf{e}}_3^L) \boldsymbol{\omega}_I^L - (\hat{\mathbf{e}}_3^L \odot \boldsymbol{\omega}_I^L) \hat{\mathbf{e}}_3^L \end{aligned}$$

Define

$$\dot{\psi} \triangleq \boldsymbol{\omega}_I^L \odot \hat{\mathbf{e}}_3^L$$

where ψ is the crossing angle that defines the rotation of the LVLH frame about its 3-axis. Then the angular velocity of the LVLH frame relative to the inertial frame is

$$\boldsymbol{\omega}_I^L = \hat{\mathbf{e}}_3^L \times \dot{\hat{\mathbf{e}}}_3^L + \dot{\psi} \hat{\mathbf{e}}_3^L \quad (2.25)$$

The time derivative of $\hat{\mathbf{e}}_3^L$ is

$$\dot{\hat{\mathbf{e}}}_3^L = \frac{\mathbf{v}}{r} - \frac{\mathbf{r} \odot \mathbf{v}}{r^3} \mathbf{r} \quad (2.26)$$

where $r = \|\mathbf{r}\|$. Substituting Eq. 2.26 into Eq. 2.25 yields

$$\boldsymbol{\omega}_I^L = \frac{\mathbf{r} \times \mathbf{v}}{r^2} + \dot{\psi} \hat{\mathbf{e}}_3^L \quad (2.27)$$

Based on the description of the maneuver frame relative to the LVLH frame, we have

$$\boldsymbol{\omega}_L^M = \dot{\theta} \hat{\mathbf{e}}_2^M \quad (2.28)$$

Therefore, combining Eqs. 2.27 and 2.28 with Eq. 2.24 yields

$$\boldsymbol{\omega}_I^M = \frac{\mathbf{r} \times \mathbf{v}}{r^2} + \dot{\psi} \hat{\mathbf{e}}_3^L + \dot{\theta} \hat{\mathbf{e}}_2^M \quad (2.29)$$

Note that

$$\mathbf{r} \times \mathbf{v} = rv \cos \theta \hat{\mathbf{e}}_2^M$$

and

$$\hat{\mathbf{e}}_3^L = \cos \theta \hat{\mathbf{e}}_3^M - \sin \theta \hat{\mathbf{e}}_1^M$$

Therefore, Eq. 2.29 reduces to

$$\boldsymbol{\omega}_I^M = -\dot{\psi} \sin \theta \hat{\mathbf{e}}_1^M + \left(\dot{\theta} + \frac{v}{r} \cos \theta \right) \hat{\mathbf{e}}_2^M - \dot{\psi} \cos \theta \hat{\mathbf{e}}_3^M \quad (2.30)$$

Substituting $\boldsymbol{\omega}_I^M$ in Eq. 2.30 into Eq. 2.23 yields

$$\dot{\mathbf{v}} = \dot{v}\hat{\mathbf{e}}_1^M - v\dot{\psi}\cos\theta\hat{\mathbf{e}}_2^M - v\left(\dot{\theta} + \frac{v}{r}\cos\theta\right)\hat{\mathbf{e}}_3^M$$

Then from Eq. 2.22, it follows that

$$\begin{aligned} \dot{v}\hat{\mathbf{e}}_1^M + v\dot{\psi}\cos\theta\hat{\mathbf{e}}_2^M - v\left(\dot{\theta} + \frac{v}{r}\cos\theta\right)\hat{\mathbf{e}}_3^M \\ = \left(g\sin\theta - \frac{T}{m}\cos\alpha\right)\hat{\mathbf{e}}_1^M - \frac{T}{m}\sin\alpha\sin\phi\hat{\mathbf{e}}_2^M \\ + \left(\frac{T}{m}\sin\alpha\cos\phi - g\cos\theta\right)\hat{\mathbf{e}}_3^M \end{aligned}$$

Therefore, the simplified LVLH frame equations for spacecraft speed, flight path angle and crossing angle are

$$\dot{v} = g\sin\theta - \frac{T}{m}\cos\alpha \quad (2.31)$$

$$v\dot{\theta} = \left(g - \frac{v^2}{r}\right)\cos\theta - \frac{T}{m}\sin\alpha\cos\phi \quad (2.32)$$

$$v\dot{\psi}\cos\theta = -\frac{T}{m}\sin\alpha\sin\phi \quad (2.33)$$

Equations 2.31, 2.32, and 2.33 can be utilized in two dimensions to design the reference trajectory and in three dimensions to design the guidance algorithm.

The position and velocity states provided to the guidance algorithm are generally in the planet-centered planet-fixed (PCPF) coordinates rather than the LVLH frame. The PCPF reference frame is defined by the unit vectors $\hat{\mathbf{e}}_1^{PCPF}$, $\hat{\mathbf{e}}_2^{PCPF}$, and $\hat{\mathbf{e}}_3^{PCPF}$, where $\hat{\mathbf{e}}_1^{PCPF}$; as with the planet centered inertial coordinates, the exact description of these is irrelevant here other than to say that the PCPF coordinates must be consistent with the PCI coordinates. The speed, flight path angle, and crossing angle must be extracted from the PCPF relative position and velocity vectors. In addition, the kinematics must be

discussed so that altitude of the spacecraft above the lunar surface, h , the distance along the lunar surface parallel with the spacecraft desired reference path to the target location simply termed downrange, d , and the distance along the lunar surface perpendicular to the spacecraft desired reference path to the target location simply called crossrange, c , shown in Fig. 2.5 can be computed.

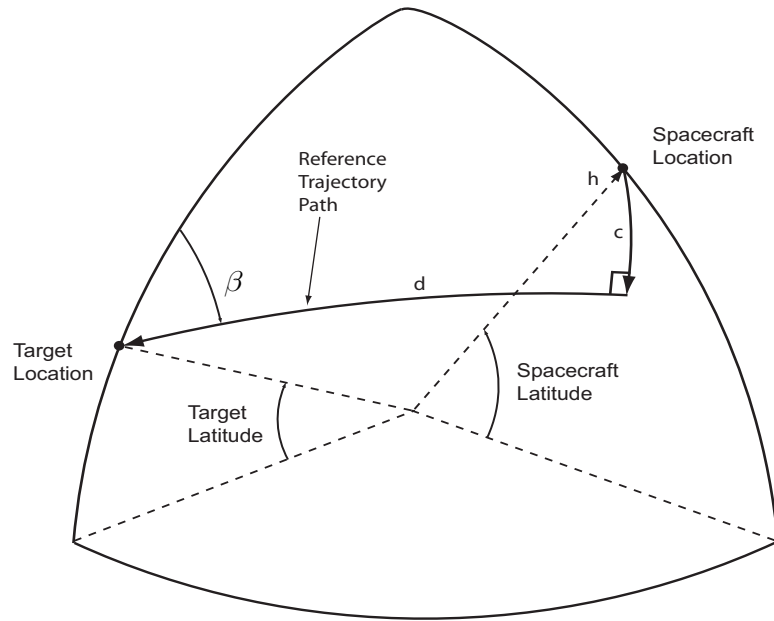


Figure 2.5: Downrange and Crossrange Definitions

In addition to the spacecraft state as a function of time, the target position, \mathbf{r}_t , represented in the PCPF reference frame, is the vector from the planet center to the location on the planet surface that is the desired landing site. The landing site is the termination of the reference trajectory. Note, however, that the reference trajectory will be defined in only two dimensions; therefore the bearing of the reference trajectory clockwise from North is specified as β .

The first definition is relatively straightforward; speed is simply the magnitude of the velocity:

$$v = \|\mathbf{v}\|$$

The flight path and crossing angles define the direction of the velocity vector. The velocity must be represented in a reference frame that is oriented along the downrange, crossrange, and altitude to give these angles meaning. Therefore, the set of rotations that take the velocity vector from the PCPF frame through some intermediate frames to a frame that has its 1-axis oriented parallel to the lunar surface and parallel to the desired bearing to the desired landing point will be explained.

The first step in defining that reference frame is to compute the geodetic latitude, longitude, and altitude of both the target point and current state. The geodetic latitude, φ , and geodetic altitude, h , are computed with the following algorithm:

Initialize:

$$e^2 = \frac{R_{eq}^2 - R_{pol}^2}{R_{eq}^2}$$

$$f = \frac{R_{eq} - R_{pol}}{R_{eq}}$$

$$\varphi = \arcsin\left(\frac{r_3}{\|\mathbf{r}\|}\right)$$

Iterate:

$$N = \frac{R_{eq}}{1 - f(2 - f)\sin^2\varphi}$$

$$h = \frac{\sqrt{r_1^2 + r_2^2}}{\cos\varphi} - N$$

$$\tan\varphi = \frac{r_3}{\sqrt{r_1^2 + r_2^2}} \frac{N + h}{N(1 - e^2) + h}$$

where R_{eq} is the planetary equatorial radius and R_{pol} is the planetary polar radius (note that the derivation of the dynamic equations of motion assumed a spherical planet; in the computation of the geodetic states, however, we take into account the knowledge that the planet is non-spherical in order to increase accuracy). The longitude, λ , is

$$\lambda = \arctan \frac{r_2}{r_1}$$

The rotation from the PCPF reference frame to an East-North-Up (ENU) reference frame centered at the target location is defined as a function φ_t and λ_t :

$$T_{PCPF}^{ENU} = \begin{bmatrix} -\sin \lambda_t & \cos \lambda_t & 0 \\ -\sin \varphi_t \cos \lambda_t & -\sin \varphi_t \sin \lambda_t & \cos \varphi_t \\ \cos \varphi_t \cos \lambda_t & \cos \varphi_t \sin \lambda_t & \sin \varphi_t \end{bmatrix}$$

The rotation from ENU to the crossrange-downrange-altitude (JRU) reference frame is defined using the bearing of the target trajectory from North:

$$T_{ENU}^{JRU} = \begin{bmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that this rotation from PCPF to JRU is defined at the target location (using the target latitude and longitude). If this rotation is used to represent the current spacecraft velocity in the JRU frame, the representation of flight path angle and crossing angle will be skewed. As illustrated in two dimensions in Fig. 2.6, the frame used to compute flight path and crossing angle must have an “up” vector parallel with the $\hat{\mathbf{e}}_3^L$. Therefore, the arc between the current and target locations must be used to account for this curvature. That rotation matrix has the form

$$T_{JRU}^{ROT} = \begin{bmatrix} \cos \Gamma & \sin \Delta \sin \Gamma & \cos \Delta \sin \Gamma \\ 0 & \cos \Delta & -\sin \Delta \\ -\sin \Gamma & \sin \Delta \cos \Gamma & \cos \Delta \cos \Gamma \end{bmatrix}$$

where Δ is the angle over which the downrange is computed and Γ is the angle over which the crossrange is computed; these will both be discussed shortly.

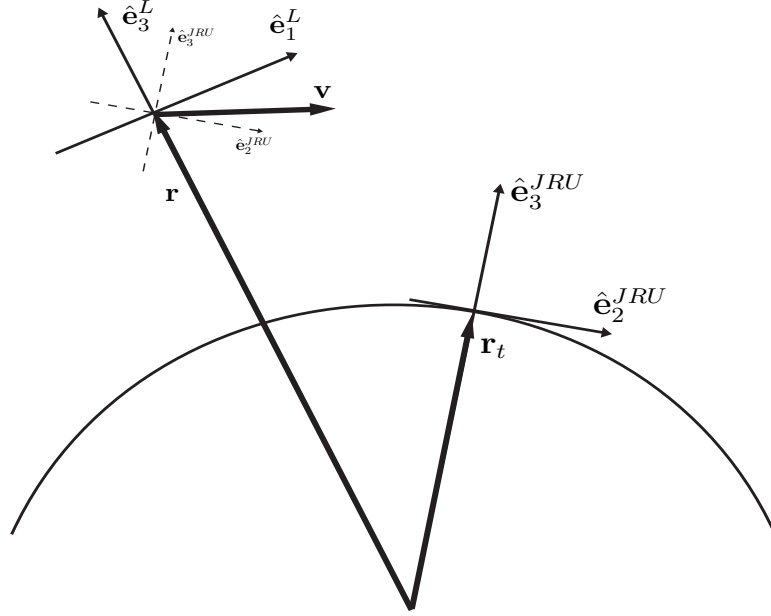


Figure 2.6: Illustration of Effect of Planet Curvature on LVLH Relative States

The rotation from PCPF to this JRU rotated reference frame takes the form

$$\begin{aligned} T_{PCPF}^{ROT} &= T_{JRU}^{ROT} T_{ENU}^{JRU} T_{PCPF}^{ENU} \\ &= \left[\hat{\mathbf{e}}_1^{ROT} \quad \hat{\mathbf{e}}_2^{ROT} \quad \hat{\mathbf{e}}_3^{ROT} \right]^T \end{aligned}$$

The flight path angle is

$$\theta = \arctan \left(\frac{-\mathbf{v} \odot \hat{\mathbf{e}}_3^{ROT}}{\sqrt{(\mathbf{v} \odot \hat{\mathbf{e}}_1^{ROT})^2 + (\mathbf{v} \odot \hat{\mathbf{e}}_2^{ROT})^2}} \right)$$

and the crossing angle is

$$\psi = \arctan \left(\frac{-\mathbf{v} \odot \hat{\mathbf{e}}_1^{ROT}}{\mathbf{v} \odot \hat{\mathbf{e}}_2^{ROT}} \right)$$

The downrange and crossrange are computed over the arc length between the spacecraft position and the target position. This is illustrated in Fig. 2.7 where φ' and φ'_t are the geocentric latitudes of the spacecraft and the target location, respectively:

$$\varphi' = \arctan \left(\frac{R_{pol}^2}{R_{eq}^2} \tan \varphi \right)$$

Additionally,

$$\Delta\lambda = \lambda - \lambda_t$$

The downrange and crossrange angles must be computed as functions of the known angles β , φ'_t , φ' , and $\Delta\lambda$. Therefore, some intermediate angles will be defined and used to solve for the Δ and Γ .

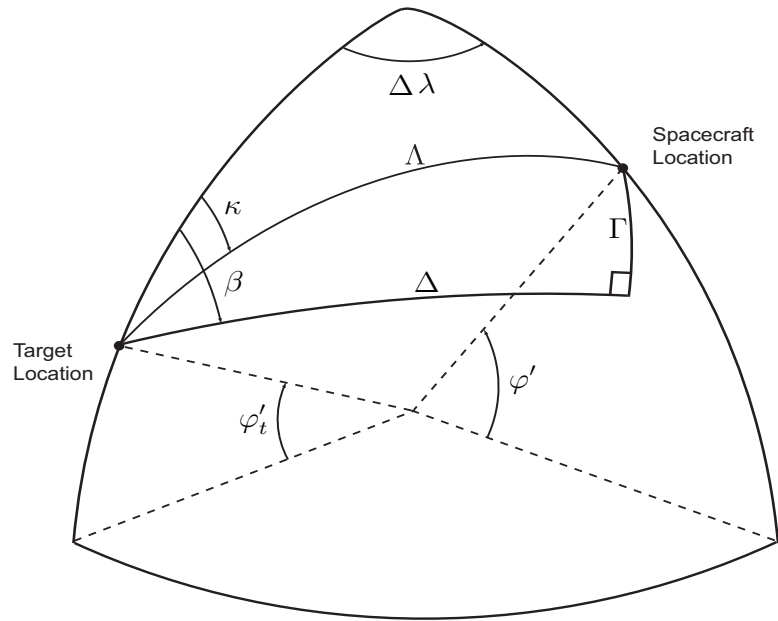


Figure 2.7: Spherical Geometry For Describing Downrange and Crossrange

For a right spherical triangle

$$\cos(\beta - \kappa) = \frac{\tan \Delta}{\tan \Lambda} \quad (2.34)$$

where

$$\cos(\beta - \kappa) = \cos \kappa \cos \beta + \sin \kappa \sin \beta \quad (2.35)$$

From the law of cosines for an oblique spherical triangle

$$\sin \varphi' = \sin \varphi'_t \cos \Lambda + \cos \varphi'_t \sin \Lambda \cos \kappa$$

or

$$\cos \kappa = \frac{\sin \varphi' - \sin \varphi'_t \cos \Lambda}{\cos \varphi'_t \sin \Lambda} \quad (2.36)$$

From the law of sines for an oblique spherical triangle

$$\sin \kappa = \frac{\sin \Delta \lambda \cos \varphi'}{\sin \Lambda} \quad (2.37)$$

Combining Eqs. 2.34, 2.35, 2.36, and 2.37 results in

$$\begin{aligned} \tan \Delta &= (\cos \kappa \cos \beta + \sin \kappa \sin \beta) \tan \Lambda \\ &= \frac{\sin \varphi' - \sin \varphi'_t \cos \Lambda}{\cos \varphi'_t \cos \Lambda} \cos \beta + \frac{\sin \Delta \lambda \cos \varphi'}{\cos \Lambda} \sin \beta \end{aligned} \quad (2.38)$$

Again, from the law of cosines for an oblique spherical triangle

$$\cos \Lambda = \sin \varphi'_t \sin \varphi' + \cos \varphi'_t \cos \varphi' \cos \Delta \lambda \quad (2.39)$$

which can be used to reduce Eq. 2.38 to

$$\tan \Delta = \frac{\sin \varphi' \cos \beta \cos \varphi'_t + \cos \varphi' (\sin \beta \sin \Delta \lambda - \sin \varphi'_t \cos \Delta \lambda \cos \beta)}{\cos \Lambda}$$

This is now in a form that can be inverted to compute Δ . The downrange is then given by

$$d = \|\mathbf{r}_t\| \Delta$$

The crossrange is

$$c = \|\mathbf{r}_t\| \Gamma$$

where, for a right spherical triangle,

$$\sin \Gamma = \sin \Lambda \sin(\beta - \kappa)$$

Various substitutions are made in a similar fashion to the derivation for Δ to compute Γ :

$$\begin{aligned} \sin \Gamma &= \frac{\sin \Delta \lambda \cos \varphi'}{\sin \kappa} (\sin \beta \cos \kappa - \cos \beta \sin \kappa) \\ &= \sin \beta \left(\frac{\sin \varphi' - \sin \varphi'_t \cos \Lambda}{\cos \varphi'_t} \right) - \cos \beta \sin \Delta \lambda \cos \varphi' \\ &= \sin \beta \sin \varphi' \cos \varphi'_t - \cos \varphi' (\sin \beta \cos \Delta \lambda \sin \varphi'_t + \sin \Delta \lambda \cos \beta) \end{aligned}$$

With the definitions for downrange and crossrange now available, their time derivatives are computed as:

$$\begin{aligned} \dot{d} &= \|\mathbf{r}_t\| \dot{\Delta} \\ \dot{c} &= \|\mathbf{r}_t\| \dot{\Gamma} \end{aligned}$$

The equations for the downrange and crossrange angular rates are

$$\begin{aligned} \dot{\Delta} &= \frac{v \cos \theta \cos \psi}{r} \\ \dot{\Gamma} &= \frac{v \cos \theta \sin \psi}{r} \end{aligned}$$

The time derivatives of altitude, downrange, and crossrange are therefore

$$\dot{h} = -v \sin \theta \quad (2.40)$$

$$\dot{d} = \frac{\|\mathbf{r}_t\| v \cos \theta \cos \psi}{\|\mathbf{r}_t + h\|} \quad (2.41)$$

$$\dot{c} = \frac{\|\mathbf{r}_t\| v \cos \theta \sin \psi}{\|\mathbf{r}_t + h\|} \quad (2.42)$$

This completes the definition of the equations of motion in an intuitive format that can be used for further algorithm development.

Chapter 3

Target Trajectory Design

3.1 Constant Thrust Acceleration Solution

From the previous chapter, the fundamental equations of motion that will be used to describe the spacecraft motion in three dimensions are from Eqs. 2.31, 2.32, 2.33, 2.40, 2.41, and 2.42:

$$\begin{aligned}\dot{v} &= g \sin \theta - \frac{T}{m} \cos \alpha \\ v\dot{\theta} &= \left(g - \frac{v^2}{h + h_0} \right) \cos \theta - \frac{T}{m} \sin \alpha \cos \phi \\ v\dot{\psi} \cos \theta &= -\frac{T}{m} \sin \alpha \sin \phi \\ \dot{h} &= -v \sin \theta \\ \dot{d} &= v \cos \theta \cos \psi \frac{h_0}{h + h_0} \\ \dot{c} &= v \cos \theta \sin \psi \frac{h_0}{h + h_0}\end{aligned}$$

where $h_0 = \|\mathbf{r}_t\|$ and is the reference radius of the spherical lunar model (note the r in Eq. 2.32 has been replaced with $h + h_0$ so that the equations of motion are explicit functions of the six states). The targeting phase of the guidance algorithm will define a target trajectory for the spacecraft to ideally fly; in order to simplify the algorithm and its development, the motion of the target trajectory will be defined to two dimensions. Therefore the initial states for the out of plane motion are initialized to zero ($c(0) = 0$, $\dot{c}(0) = 0$, $\psi(0) = 0$, and

$\dot{\psi}(0) = 0$) and the roll control states are held at zero ($\phi(t) = 0$ and $\dot{\phi}(t) = 0$). This reduces the equations of motion to their two-dimensional form:

$$\dot{v} = g \sin \theta - \frac{T}{m} \cos \alpha \quad (3.1)$$

$$v\dot{\theta} = \left(g - \frac{v^2}{h + h_0} \right) \cos \theta - \frac{T}{m} \sin \alpha \quad (3.2)$$

$$\dot{h} = -v \sin \theta \quad (3.3)$$

$$\dot{d} = v \cos \theta \frac{h_0}{h + h_0} \quad (3.4)$$

To allow for the development of analytical solutions to the above equations, some additional assumptions are made consistent with those in [3]. First, note that

$$\frac{dv}{dt} = \frac{dv}{d\theta} \frac{d\theta}{dt}$$

so

$$\frac{\frac{dv}{d\theta}}{v} = \frac{g \sin \theta - \frac{T}{m} \cos \alpha}{\left(g - \frac{v^2}{h+h_0} \right) \cos \theta - \frac{T}{m} \sin \alpha} \triangleq \Phi(\theta) \quad (3.5)$$

This equation can be solved analytically if the right hand side of the equation is reduced to a simple function of θ . This means that the terms $\frac{T}{m}$, α , g , and $\frac{v^2}{h+h_0}$ are constant. However, it is known that, even if T and α are held constant over the target trajectory, m will not be and so $\frac{T}{m}$ is varying. It will be shown later that this assumption is reasonable for generating the target trajectory and the errors will be removed by the real-time guidance algorithm. Similar logic is applied to the gravity terms; g is actually close to constant, but any real-time variation is absorbed by the guidance algorithm. The centrifugal acceleration term, $\frac{v^2}{h+h_0}$, is also varying significantly, but its errors can be captured in real-time by the guidance algorithm.

While constant values to use for mass and gravity are straightforward (for example using initial mass and gravity may be reasonable assumptions), developing some logic for computing the constant value to be used for the centrifugal acceleration term is helpful. To do this, first note that gravity with respect to a spherical source is defined as

$$g = \frac{\mu}{(h + h_0)^2}$$

where μ is the gravitational constant. Then assume that the initial speed of the spacecraft is some proportion of orbital speed:

$$v_{orbital} \approx \sqrt{\frac{\mu}{h + h_0}} \rightarrow v \approx \sqrt{\frac{k\mu}{h + h_0}}$$

Therefore, the approximation can be made that

$$g - \frac{v^2}{h + h_0} \approx (1 - k)g$$

where k is a constant value that essentially is the approximation of the relationship between the centrifugal acceleration over the entire trajectory to the constant approximated gravity over the trajectory.

One additional assumption is made to simplify the solution. The angle α is a control input that varies the orientation of the thrust vector with respect to the velocity vector. For the purposes of defining the reference trajectory, this control input is set to zero. Though it is critical in the guidance algorithm for orienting the thrust vector to properly fly the target trajectory, it is not critical to defining that same trajectory.

With the above assumptions, Eq. 3.5 reduces to

$$\frac{dv}{d\theta} = \frac{g \sin \theta - \frac{T}{m}}{(1 - k)g \cos \theta} \triangleq \Phi(\theta)$$

or

$$\frac{dv}{v} = \Phi(\theta)d\theta$$

which can be integrated as

$$\ln \frac{v}{v_0} = \int_{\theta_0}^{\theta} \Phi(\xi)d\xi$$

If the exponential of both sides is taken, the velocity becomes a function of flight path angle:

$$v(\theta) = v_0 e^{\int_{\theta_0}^{\theta} \Phi(\xi)d\xi}$$

The integration of $\Phi(\theta)$ yields

$$\begin{aligned} \int_{\theta_0}^{\theta} \Phi(\xi)d\xi &= \int_{\theta_0}^{\theta} \frac{\sin \xi}{(1-k)\cos \xi} d\xi - \int_{\theta_0}^{\theta} \frac{\frac{T}{m}}{(1-k)g \cos \xi} d\xi \\ &= \ln \left(\frac{\cos \theta_0}{\cos \theta} \right)^{\frac{1}{1-k}} + \ln \left(\frac{\tan \left(\frac{\pi}{4} + \frac{\theta_0}{2} \right)}{\tan \left(\frac{\pi}{4} + \frac{\theta}{2} \right)} \right)^{\frac{\frac{T}{m}}{(1-k)g}} \end{aligned}$$

Therefore,

$$v(\theta) = v_0 \left(\frac{\cos \theta_0}{\cos \theta} \right)^{\frac{1}{1-k}} \left(\frac{\tan \left(\frac{\pi}{4} + \frac{\theta_0}{2} \right)}{\tan \left(\frac{\pi}{4} + \frac{\theta}{2} \right)} \right)^{\frac{\frac{T}{m}}{(1-k)g}}$$

As the development of the equations continues the following identity will be useful:

$$\tan \left(\frac{\pi}{4} + \frac{\theta}{2} \right) = \frac{\cos \theta}{1 - \sin \theta}$$

The solution for speed now takes the form

$$\begin{aligned} v(\theta) &= v_0 \left(\frac{\cos \theta_0}{\cos \theta} \right)^{\frac{1}{1-k}} \left(\frac{\frac{\cos \theta_0}{1 - \sin \theta_0}}{\frac{\cos \theta}{1 - \sin \theta}} \right)^{\frac{\frac{T}{m}}{(1-k)g}} \\ &= v_0 \frac{(\cos \theta_0)^{\frac{g + \frac{T}{m}}{(1-k)g}} (1 - \sin \theta)^{\frac{\frac{T}{m}}{(1-k)g}}}{(1 - \sin \theta_0)^{\frac{\frac{T}{m}}{(1-k)g}} (\cos \theta)^{\frac{g + \frac{T}{m}}{(1-k)g}}} \end{aligned} \quad (3.6)$$

where $\theta \neq \pm \frac{\pi}{2}$ and $\theta_0 \neq \frac{\pi}{2}$

Time, downrange, and altitude are solved in a similar fashion. First, the assumption is made in Eq. 3.4 that $h \ll h_0$ so that $\frac{h_0}{h+h_0} \approx 1$. Then the differential equations are formulated as a function of flight path angle as

$$\begin{aligned}\frac{dt}{d\theta} &= \frac{v(\theta)}{(1-k)g \cos \theta} \\ \frac{dh}{d\theta} &= \frac{-v(\theta)^2 \sin \theta}{(1-k)g \cos \theta} \\ \frac{dd}{d\theta} &= \frac{v(\theta)^2 \cos \theta}{(1-k)g \cos \theta}\end{aligned}$$

or

$$\begin{aligned}\frac{dt}{d\theta} &= v_0 \frac{(\cos \theta_0)^{\frac{g+\frac{T}{m}}{(1-k)g}} (1 - \sin \theta)^{\frac{\frac{T}{m}}{(1-k)g}}}{(1 - \sin \theta_0)^{\frac{\frac{T}{m}}{(1-k)g}} (\cos \theta)^{\frac{g+\frac{T}{m}}{(1-k)g}}} \frac{1}{(1-k)g \cos \theta} \\ &= \frac{v_0}{(1-k)g} \frac{(\cos \theta_0)^{\frac{g+\frac{T}{m}}{(1-k)g}} (1 - \sin \theta)^{\frac{\frac{T}{m}}{(1-k)g}}}{(1 - \sin \theta_0)^{\frac{\frac{T}{m}}{(1-k)g}} (\cos \theta)^{\frac{g+\frac{T}{m}}{(1-k)g} + 1}} \\ \frac{dh}{d\theta} &= -v_0^2 \frac{(\cos \theta_0)^{2\frac{g+\frac{T}{m}}{(1-k)g}} (1 - \sin \theta)^{\frac{2\frac{T}{m}}{(1-k)g}} \sin \theta}{(1 - \sin \theta_0)^{2\frac{\frac{T}{m}}{(1-k)g}} (\cos \theta)^{2\frac{g+\frac{T}{m}}{(1-k)g}} (1-k)g \cos \theta} \\ &= -\frac{v_0^2}{(1-k)g} \frac{(\cos \theta_0)^{2\frac{g+\frac{T}{m}}{(1-k)g}} (1 - \sin \theta)^{\frac{2\frac{T}{m}}{(1-k)g}} \sin \theta}{(1 - \sin \theta_0)^{2\frac{\frac{T}{m}}{(1-k)g}} (\cos \theta)^{2\frac{g+\frac{T}{m}}{(1-k)g} + 1}} \\ \frac{dd}{d\theta} &= \frac{v_0^2}{(1-k)g} \frac{(\cos \theta_0)^{2\frac{g+\frac{T}{m}}{(1-k)g}} (1 - \sin \theta)^{\frac{2\frac{T}{m}}{(1-k)g}}}{(1 - \sin \theta_0)^{2\frac{\frac{T}{m}}{(1-k)g}} (\cos \theta)^{2\frac{g+\frac{T}{m}}{(1-k)g}}}\end{aligned}$$

The first notation simplification is to define $p \triangleq \frac{T}{m}g$ where $p \in \Re$ and

$p > 0$. Then define

$$\begin{aligned}\kappa_t &\triangleq \frac{v_0}{(1-k)g} \frac{(\cos \theta_0)^{\frac{g+\frac{T}{m}}{(1-k)g}}}{(1-\sin \theta_0)^{\frac{T}{m}g}} \\ \kappa_h &\triangleq \frac{v_0^2}{(1-k)g} \frac{(\cos \theta_0)^{2\frac{g+\frac{T}{m}}{(1-k)g}}}{(1-\sin \theta_0)^{2\frac{T}{m}g}} \\ \kappa_d &\triangleq \frac{v_0^2}{(1-k)g} \frac{(\cos \theta_0)^{2\frac{g+\frac{T}{m}}{(1-k)g}}}{(1-\sin \theta_0)^{2\frac{T}{m}g}}\end{aligned}$$

so that

$$\frac{dt}{d\theta} = \kappa_t \frac{(1-\sin \theta)^{\frac{p}{(1-k)}}}{(\cos \theta)^{\frac{1+p}{(1-k)}+1}} \quad (3.7)$$

$$\frac{dh}{d\theta} = -\kappa_h \frac{(1-\sin \theta)^{\frac{2p}{(1-k)}} \sin \theta}{(\cos \theta)^{\frac{2+2p}{(1-k)}+1}} \quad (3.8)$$

$$\frac{dd}{d\theta} = \kappa_d \frac{(1-\sin \theta)^{\frac{2p}{(1-k)}}}{(\cos \theta)^{\frac{2+2p}{(1-k)}}} \quad (3.9)$$

Recall k is the constant proportion of centrifugal acceleration to gravity acceleration term assumed for the solutions to the two-dimensional differential equations. This author could find no general solution to the above equations for any $k \in \mathfrak{R}$ so had to specify a logical value at this juncture. In order to integrate the equations analytically the value for $\frac{1}{1-k}$ must be an integer. This requires $k = \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \dots$. In an attempt to get a rough estimate of what an ideal k might be note that, if speed were a linear function of time, the integral of $\frac{v^2}{h+h_0}$ from orbital speed to zero (assuming r is constant and the trajectory initiates at orbital speed) is $\frac{1}{3} \frac{v_{orbital}^2}{h+h_0}$ or $k = \frac{1}{3}$. However, this value can not be used because it does not produce an integer value for $\frac{1}{1-k}$.

The impact of varying the constant k is shown in Figs. 3.1, 3.2, 3.3, and 3.4. Different values for k were introduced into Eqs. 3.6, 3.7, 3.8, and 3.9 and these were numerically integrated with constant values for g and $\frac{T}{m}$ where $g = 1.623 \frac{m}{sec^2}$ and $\frac{T}{m} = 4 \frac{N}{kg}$. The full numerically integrated solutions to Eqs. 3.1, 3.2, 3.3, and 3.4 are shown for comparison; no approximations are made about the centrifugal acceleration in these integrations while $\alpha = 0$ and g and $\frac{T}{m}$ are the same values used in the integration of Eqs. 3.6, 3.7, 3.8, and 3.9.

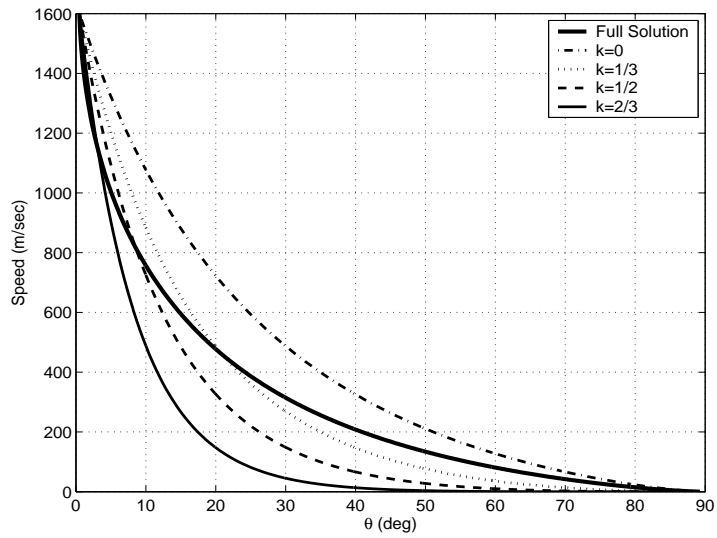


Figure 3.1: Comparison of Full Integrated Solution to Approximated Solution with Varying Values of k : Speed

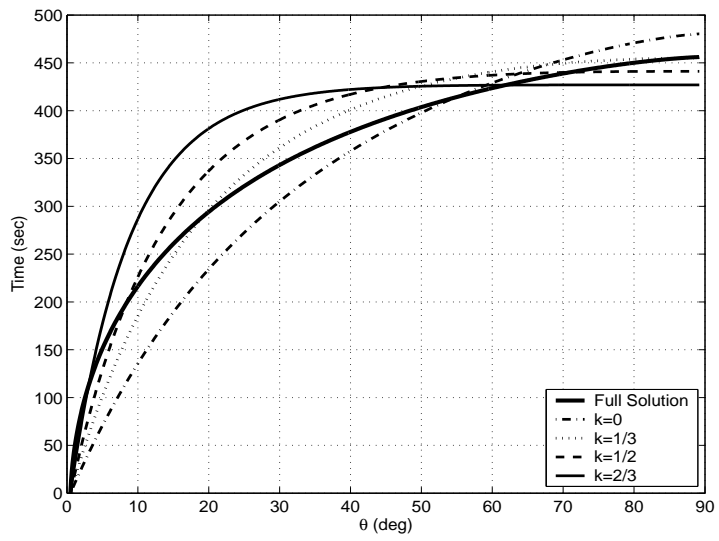


Figure 3.2: Comparison of Full Integrated Solution to Approximated Solution with Varying Values of k : Time

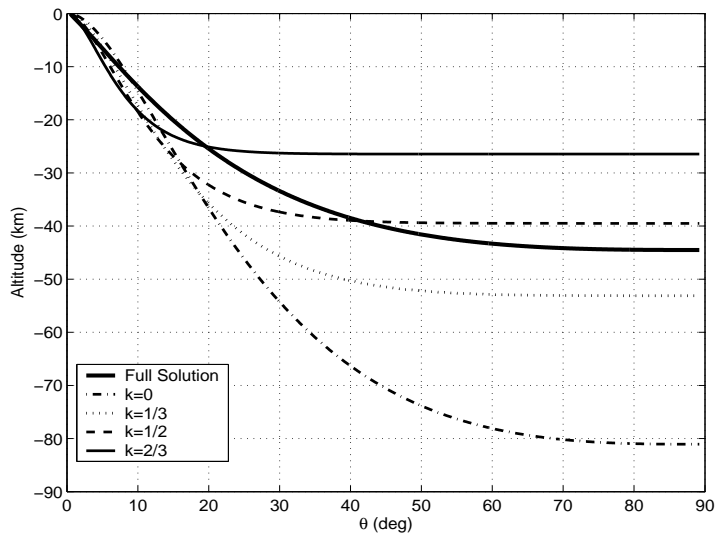


Figure 3.3: Comparison of Full Integrated Solution to Approximated Solution with Varying Values of k : Altitude

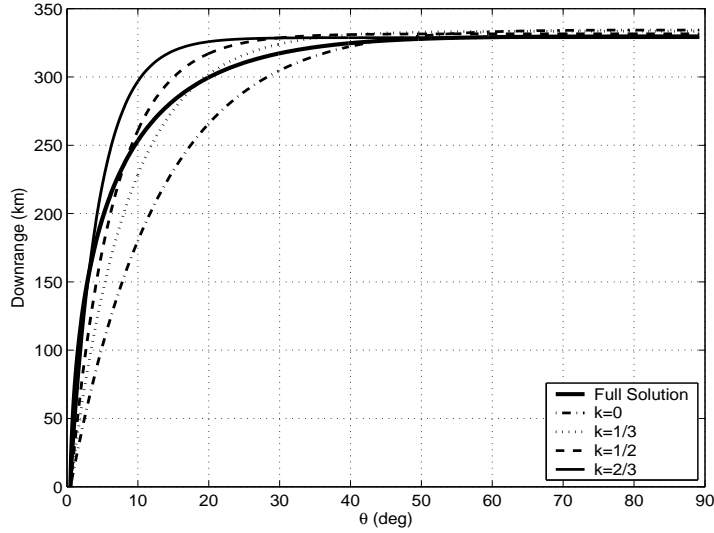


Figure 3.4: Comparison of Full Integrated Solution to Approximated Solution with Varying Values of k : Downrange

Note from the plots that varying k has the largest impact on the final altitude variation. The centrifugal acceleration essentially adjusts the rate of change of the flight path angle which impacts the orientation of the velocity vector. If $\dot{\theta}$ is small (i.e. k is large), θ will remain small and the trajectory will tend to be more horizontal than when $\dot{\theta}$ is large (i.e. k is small). Another way to look at this is that the centrifugal acceleration impacts the rate at which the spacecraft “falls”. This directly affects the altitude of the trajectory.

From the comparison of the various values for k , a value of $k = \frac{1}{2}$ appears to be a reasonable number from the set of allowable values for k that produce integers for $\frac{1}{1-k}$. In fact, $k = \frac{1}{2}$ appears to give a comparable solution to the case where $k = \frac{1}{3}$ (recall $k = \frac{1}{3}$ is the solution for the rough estimate discussed earlier). Recall that this value for k is not critical to the design of the target trajectory; rather it is a bias value that is used to better approximate

the real solution. The real-time guidance algorithm will absorb the errors in the centrifugal acceleration and adjust the control parameters accordingly.

The final form of the differential equations with $k = \frac{1}{2}$ is

$$\begin{aligned}\frac{dt}{d\theta} &= \kappa_t \frac{(1 - \sin \theta)^{2p}}{(\cos \theta)^{2p+3}} \\ \frac{dh}{d\theta} &= -\kappa_h \frac{(1 - \sin \theta)^{4p} \sin \theta}{(\cos \theta)^{4p+5}} \\ \frac{dd}{d\theta} &= \kappa_d \frac{(1 - \sin \theta)^{4p}}{(\cos \theta)^{4p+4}}\end{aligned}$$

The solution for the time equation is

$$\begin{aligned}t(\theta) = & -\kappa_t \left(\frac{(1 - \sin \theta)^{2p}}{(2p + 2)(\cos \theta)^{2p+2}} + \frac{(1 - \sin \theta)^{2p-1}}{p(2p + 2)(\cos \theta)^{2p}} \right. \\ & \left. + \frac{(1 - \sin \theta)^{2p-2}}{p(2p + 2)(2p - 2)(\cos \theta)^{2p-2}} \right) + \eta_t\end{aligned}\quad (3.10)$$

where

$$\eta_t = t_0 - t(\theta_0)$$

and t_0 is the initial time. The solution for altitude is

$$\begin{aligned}h(\theta) = & -\kappa_h \left(\frac{(1 - \sin \theta)^{4p}}{(4p + 4)(\cos \theta)^{4p+4}} - \frac{2p(1 - \sin \theta)^{4p}}{(4p + 4)(4p + 2)(\cos \theta)^{4p+2}} \right. \\ & - \frac{(2p + 3)(1 - \sin \theta)^{4p-1}}{2(4p + 4)(4p + 2)(\cos \theta)^{4p}} - \frac{2p(1 - \sin \theta)^{4p-3}}{(4p + 4)(4p - 2)(\cos \theta)^{4p-2}} \\ & + \frac{(4p^2 - 3)(1 - \sin \theta)^{4p-2}}{2(4p + 4)(4p + 2)(4p - 2)(\cos \theta)^{4p-2}} \\ & + \frac{(p + 2)(1 - \sin \theta)^{4p-5}}{(4p + 4)(4p - 4)(\cos \theta)^{4p-4}} - \frac{(p + 2)(1 - \sin \theta)^{4p-6}}{2(4p + 4)(4p - 4)(\cos \theta)^{4p-6}} \\ & \left. + \frac{(2 - 6p)(1 - \sin \theta)^{4p-4}}{(4p + 4)(4p - 4)(4p - 2)(\cos \theta)^{4p-4}} \right) + \eta_h\end{aligned}\quad (3.11)$$

where

$$\eta_h = h_0 - h(\theta_0)$$

and h_0 is the initial altitude. The solution for downrange is

$$\begin{aligned}
d(\theta) = \kappa_d \left(& -\frac{(1 - \sin \theta)^{4p}}{(4p + 3)(\cos \theta)^{4p+3}} + \frac{(2 \sin \theta - 1)(1 - \sin \theta)^{4p-1}}{(4p + 3)(4p + 1)(\cos \theta)^{4p+1}} \right. \\
& + \frac{(4p \sin \theta + 3 \sin \theta - 4p - 1)(1 - \sin \theta)^{4p-2}}{(4p + 3)(4p + 1)(4p - 1)(\cos \theta)^{4p-1}} \\
& \left. + \frac{(3 - 16p^2)(1 - \sin \theta)^{4p-3}}{(4p + 3)(4p - 3)(4p + 1)(4p - 1)(\cos \theta)^{4p-3}} \right) + \eta_d \quad (3.12)
\end{aligned}$$

where

$$\eta_d = d_0 - d(\theta_0)$$

and d_0 is the initial downrange.

Figures 3.5, 3.6, 3.7, and 3.8 show a comparison of the analytical solutions for speed, time, altitude, and downrange given $k = \frac{1}{2}$ to the numerically integrated solutions to 3.1, 3.2, 3.3, and 3.4 where $\alpha = 0$. The numerical solution assumes a constant gravity, g , and constant thrust to mass ratio, $\frac{T}{m}$, but does not assume a constant centrifugal acceleration. These figures illustrate the trajectory variations that the guidance will be required to remove.

Given this set of analytical equations for speed, time, downrange, and altitude as a function of flight path angle, an algorithm can be developed to create a nominal planar target trajectory for the spacecraft.

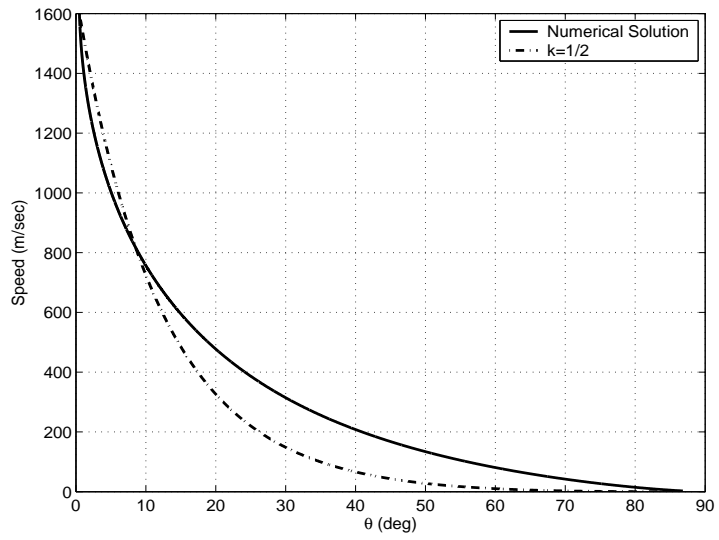


Figure 3.5: Comparison of Analytical Target Trajectory Solutions To Numerical Solutions: Speed

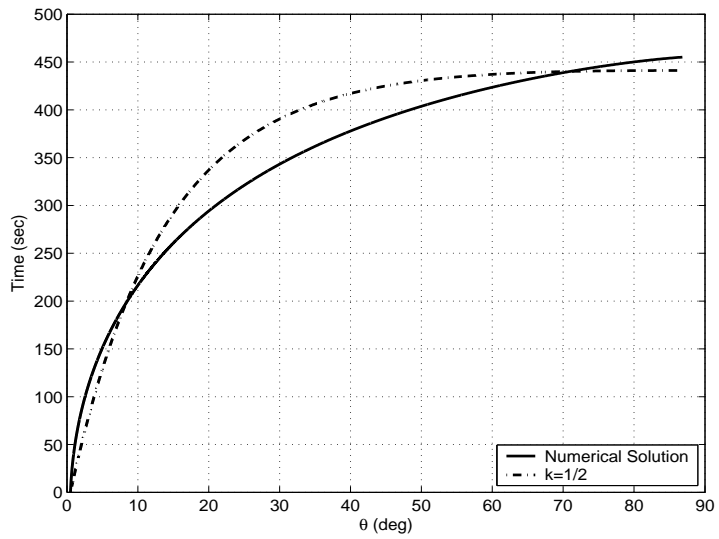


Figure 3.6: Comparison of Analytical Target Trajectory Solutions To Numerical Solutions: Time

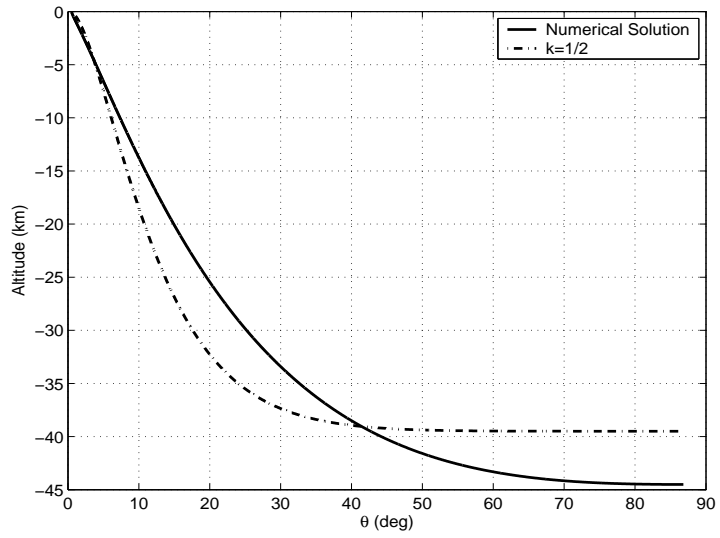


Figure 3.7: Comparison of Analytical Target Trajectory Solutions To Numerical Solutions: Altitude

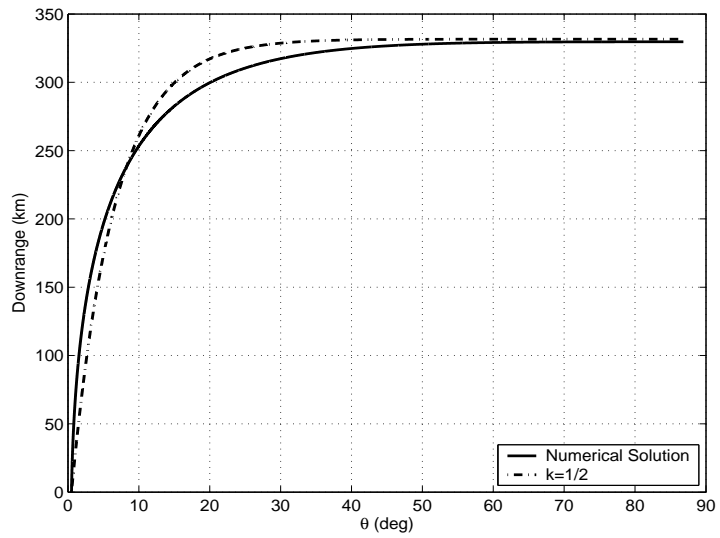


Figure 3.8: Comparison of Analytical Target Trajectory Solutions To Numerical Solutions: Downrange

3.2 Targeting Algorithm Design

With the equations of the previous section, a planar target trajectory can be generated. Note that this trajectory will be a function of downrange, altitude, speed, and flight path angle only. Therefore, it is not specific to any one point on the lunar surface or even to a bearing from North. It can be easily relocated or rotated as necessary to meet mission objectives.

The analytical solutions for the speed, altitude, and downrange equations of motion in Eqs. 3.6, 3.11, and 3.12 can be rewritten. The speed equation now assumes $k = \frac{1}{2}$ and is given by

$$v(\theta) = v_0 \frac{(\cos \theta_0)^{2p+2} (1 - \sin \theta)^{2p}}{(1 - \sin \theta_0)^{2p} (\cos \theta)^{2p+2}} \quad (3.13)$$

In rewriting downrange, let

$$\begin{aligned} D(\theta, p) = & -\frac{\cos \theta}{(4p+3)} + \frac{(2 \sin \theta - 1)(\cos \theta)^3}{(4p+3)(4p+1)(1 - \sin \theta)} \\ & + \frac{(4p \sin \theta + 3 \sin \theta - 4p - 1)(\cos \theta)^5}{(4p+3)(4p+1)(4p-1)(1 - \sin \theta)^2} \\ & + \frac{(3 - 16p^2)(\cos \theta)^7}{(4p+3)(4p-3)(4p+1)(4p-1)(1 - \sin \theta)^3} \end{aligned}$$

so that

$$d(\theta) = d_0 + \frac{2v(\theta)^2}{g} D(\theta, p) - \frac{2v_0^2}{g} D(\theta_0, p) \quad (3.14)$$

In rewriting altitude, let

$$\begin{aligned}
H(\theta, p) = & \frac{1}{(4p+4)} - \frac{2p(\cos \theta)^2}{(4p+4)(4p+2)} \\
& - \frac{(2p+3)(\cos \theta)^4}{2(4p+4)(4p+2)(1-\sin \theta)} - \frac{2p(\cos \theta)^6}{(4p+4)(4p-2)(1-\sin \theta)^3} \\
& + \frac{(4p^2-3)(\cos \theta)^6}{2(4p+4)(4p+2)(4p-2)(1-\sin \theta)^2} \\
& + \frac{(p+2)(\cos \theta)^8}{(4p+4)(4p-4)(1-\sin \theta)^5} - \frac{(p+2)(\cos \theta)^{10}}{2(4p+4)(4p-4)(1-\sin \theta)^6} \\
& + \frac{(2-6p)(\cos \theta)^8}{(4p+4)(4p-4)(4p-2)(1-\sin \theta)^4}
\end{aligned}$$

so that

$$h(\theta) = h_0 - \frac{2v(\theta)^2}{g}H(\theta, p) + \frac{2v_0^2}{g}H(\theta_0, p) \quad (3.15)$$

The target trajectory will be comprised of n segments where $\frac{T}{m}$ is constant over each segment and the combination of the segments results in a cumulative downrange and altitude. These segments will be joined by the continuity conditions that the speed and flight path angle be continuous over each junction. If the initial downrange and altitude of the entire trajectory are given by d_0 and h_0 and the final downrange and altitude of the entire trajectory are given by d_f and h_f then the total change in downrange and altitude is defined by

$$\Delta d = d_f - d_0 = \sum_{i=1}^n \Delta d_i \quad (3.16)$$

$$\Delta h = h_f - h_0 = \sum_{i=1}^n \Delta h_i \quad (3.17)$$

The final flight path angle and speed for the entire trajectory are also correlated to the segments by the fact that the final flight path angle, θ_f , is the same

as the n^{th} flight path angle, θ_n and the final speed, v_f is the same as the n^{th} speed, v_n . From Eqs. 3.13, 3.14, and 3.15 the individual segment's change in downrange and altitude are given by

$$\begin{aligned}\Delta d_i &= \frac{2v_i^2}{g}D(\theta_i, p_i) - \frac{2v_{i-1}^2}{g}D(\theta_{i-1}, p_i) \\ \Delta h_i &= -\frac{2v_i^2}{g}H(\theta_i, p_i) + \frac{2v_{i-1}^2}{g}H(\theta_{i-1}, p_i)\end{aligned}$$

where θ_i is the flight path angle at the end of the segment and θ_{i-1} is the flight path angle at the beginning of the segment. Additionally, the speed at the end of the segment is given by

$$v_i = v_{i-1} \frac{(\cos \theta_{i-1})^{2p_i+2} (1 - \sin \theta_i)^{2p_i}}{(1 - \sin \theta_{i-1})^{2p_i} (\cos \theta_i)^{2p_i+2}} \quad (3.18)$$

where v_{i-1} is the speed at the beginning of the segment and

$$p_i = \frac{\frac{T}{m_i}}{g_i}$$

where p_i is constant over the segment.

Typically, the goal is to specify the flight path angle and speed at the beginning and end of the total trajectory $(\theta_0, \theta_f, v_0, v_f)$. This would result in a desired total downrange and altitude given a set of values for the n values of $\frac{T}{m_i}$. With these definitions and n segments, there exist $n + 2$ equations (Eqs. 3.16 and 3.17 and $n \times$ Eq. 3.18) and $3n + 4$ variables $(\Delta d, \Delta h, \theta_0, v_0, n \times \theta_i, n \times v_i, n \times \frac{T}{m_i})$.

If a single segment is used to define the total trajectory, there exist three equations (Eqs. 3.16 and 3.17 where $n = 1$ and Eq. 3.18 where $i = 1$) with seven variables $(\theta_0, \theta_1, v_0, v_1, \Delta d, \Delta h, \frac{T}{m})$. Given the desire to specify six of those variables $(\theta_0, \theta_1, v_0, v_1, \Delta d, \Delta h)$, the problem is quickly over-constrained. This can be remedied by adding more segments to the solution.

If two segments are used with the same goals (define $\theta_0, \theta_f, v_0, v_f, \Delta d, \Delta h$), the problem becomes perfectly constrained. There exist four equations (Eqs. 3.16 and 3.17 where $n = 2$ and two from Eq. 3.18 with $i = 1$ and $i = 2$) with ten variables ($\theta_0, \theta_1, \theta_2, v_0, v_1, v_2, \Delta d, \Delta h, \frac{T}{m_1}, \frac{T}{m_2}$). The goal is to compute $\theta_1, v_1, \frac{T}{m_1}$, and $\frac{T}{m_2}$ given the other six specified variables ($\theta_0, \theta_2, v_0, v_2, \Delta d, \Delta h$).

Computing v_1 is straightforward if θ_1 and either $\frac{T}{m_1}$ or $\frac{T}{m_2}$ are known along with θ_0 and v_0 or θ_2 and v_2 :

$$\begin{aligned} v_1 &= v_0 \frac{(\cos \theta_0)^{2p_1+2} (1 - \sin \theta_1)^{2p_1}}{(1 - \sin \theta_0)^{2p_1} (\cos \theta_1)^{2p_1+2}} \\ &= v_2 \frac{(\cos \theta_2)^{2p_2+2} (1 - \sin \theta_1)^{2p_2}}{(1 - \sin \theta_2)^{2p_2} (\cos \theta_1)^{2p_2+2}} \end{aligned} \quad (3.19)$$

This equality can be exploited to compute θ_1 . First,

$$v_2 = v_0 \frac{(\cos \theta_0)^{2p_1+2} (1 - \sin \theta_1)^{2p_1-2p_2} (1 - \sin \theta_2)^{2p_2}}{(1 - \sin \theta_0)^{2p_1} (\cos \theta_1)^{2p_1-2p_2} (\cos \theta_2)^{2p_2+2}} \quad (3.20)$$

This equation can be manipulated into

$$\begin{aligned} \frac{v_2 (\cos \theta_2)^{2p_2+2} (1 - \sin \theta_0)^{2p_1}}{v_0 (1 - \sin \theta_2)^{2p_2} (\cos \theta_0)^{2p_1+2}} &= \frac{(1 - \sin \theta_1)^{2p_1-2p_2}}{(\cos \theta_1)^{2p_1-2p_2}} \\ &= \left(\tan \left(\frac{\pi}{4} - \frac{\theta_1}{2} \right) \right)^{2p_1-2p_2} \end{aligned}$$

so that θ_1 can be readily computed as

$$\theta_1 = -2 \left(\tan^{-1} \left(\frac{v_2 (\cos \theta_2)^{2p_2+2} (1 - \sin \theta_0)^{2p_1}}{v_0 (1 - \sin \theta_2)^{2p_2} (\cos \theta_0)^{2p_1+2}} \right)^{\frac{1}{2p_1-2p_2}} - \frac{\pi}{4} \right) \quad (3.21)$$

From Eqs. 3.16 and 3.17, the two-segment trajectory spans the following

downrange and altitude:

$$\begin{aligned}\Delta d &= \frac{2v_1^2}{g} (D(\theta_1, p_1) - D(\theta_1, p_2)) \\ &\quad - \frac{2v_0^2}{g} D(\theta_0, p_1) + \frac{2v_2^2}{g} D(\theta_2, p_2)\end{aligned}\quad (3.22)$$

$$\begin{aligned}\Delta h &= -\frac{2v_1^2}{g} (H(\theta_1, p_1) - H(\theta_1, p_2)) \\ &\quad + \frac{2v_0^2}{g} H(\theta_0, p_1) - \frac{2v_2^2}{g} H(\theta_2, p_2)\end{aligned}\quad (3.23)$$

where θ_1 is given in Eq. 3.21 and v_1 is given in Eq. 3.19. This makes Δd and Δh fundamentally functions of θ_0 , θ_2 , v_0 , v_2 , $\frac{T}{m_1}$, and $\frac{T}{m_2}$. The ideal is to now invert Eqs. 3.22 and 3.23 such that Δd and Δh can be specified and $\frac{T}{m_1}$ and $\frac{T}{m_2}$ can be solved for. However, upon examination of the equations, an analytical solution cannot be found for the inversion.

Given the restrictions on the inversion of Eqs. 3.22 and 3.23, some options exist for creating the targeting law. The first option is to specify all of the independent variables (θ_0 , θ_2 , v_0 , v_2 , $\frac{T}{m_1}$, and $\frac{T}{m_2}$) with reasonable values. The handover conditions from the orbital to descent phase must then be initiated close to the downrange and altitude values generated by inserting those values for θ_0 , θ_2 , v_0 , v_2 , $\frac{T}{m_1}$, and $\frac{T}{m_2}$ into Eqs. 3.22 and 3.23. This is by far the simplest solution, but may not be reasonable.

Another solution is to specify θ_0 , θ_2 , v_0 , and v_2 and search through reasonable values for $\frac{T}{m_1}$ and $\frac{T}{m_2}$ to generate a trajectory space. A desirable trajectory can then be selected from the options available. This is not automatic as there is no guarantee that the desired trajectory will be in the solution space. It does, however, offer the best option for allowing the orbital phase to descent phase handover to merge in an acceptable fashion.

A sample trajectory space is given in Fig. 3.9. The initial and final flight path angles were set at 0.1° and 89° and the initial and final speeds were set at $1688 \frac{m}{sec}$ and $8 \frac{m}{sec}$. The gravity used in this analysis is that at the lunar surface. The diamonds represent altitudes versus downranges that can be reached with Eqs. 3.22 and 3.23 by varying the values for $\frac{T}{m}$ values that span $0.1 \frac{N}{kg}$ to $10.1 \frac{N}{kg}$ in $0.25 \frac{N}{kg}$ increments for both $\frac{T}{m_1}$ and $\frac{T}{m_2}$. Note that the number of points is actually much larger than those shown; the downranges and altitudes generated span values that are thousand of kilometers from the planet surface and thereby well out of a reasonable range of values. Figure 3.9 appears to have sets of curves. This is the impact of varying the thrust acceleration. Each curve is created by varying $\frac{T}{m_1}$ and those curves are transformed by changing $\frac{T}{m_2}$. This is further illustrated in the 3-dimensional plots in Figs. 3.10 and 3.11.

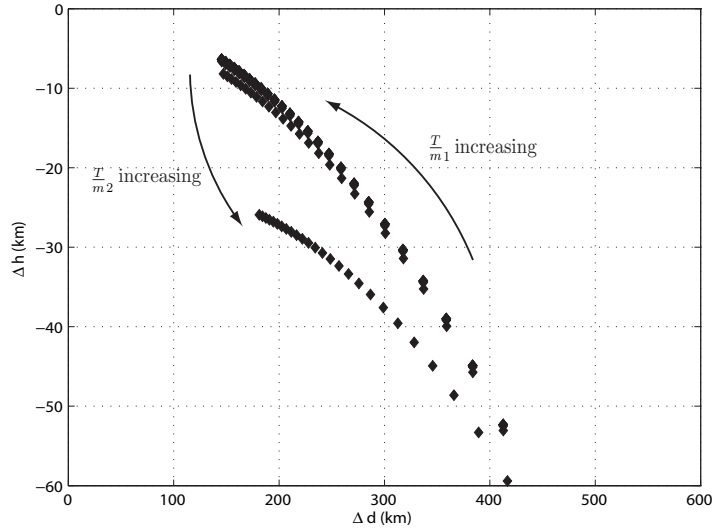


Figure 3.9: Sample Trajectory Space Varying $\frac{T}{m}$ Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}$

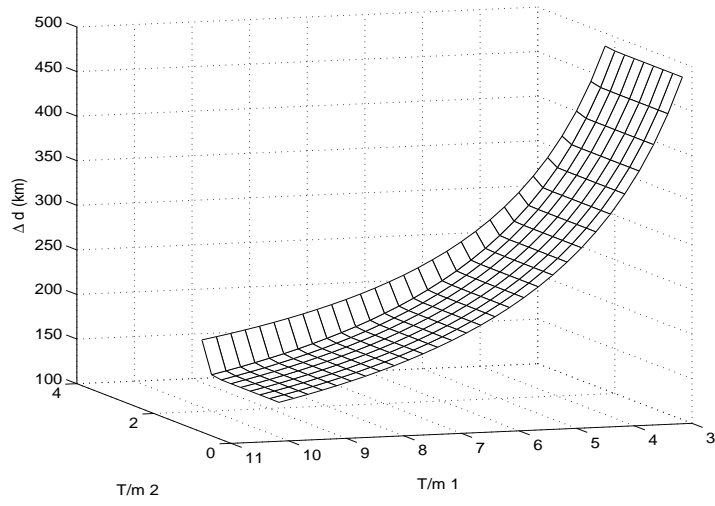


Figure 3.10: Three-Dimensional View of Downrange Sample Space

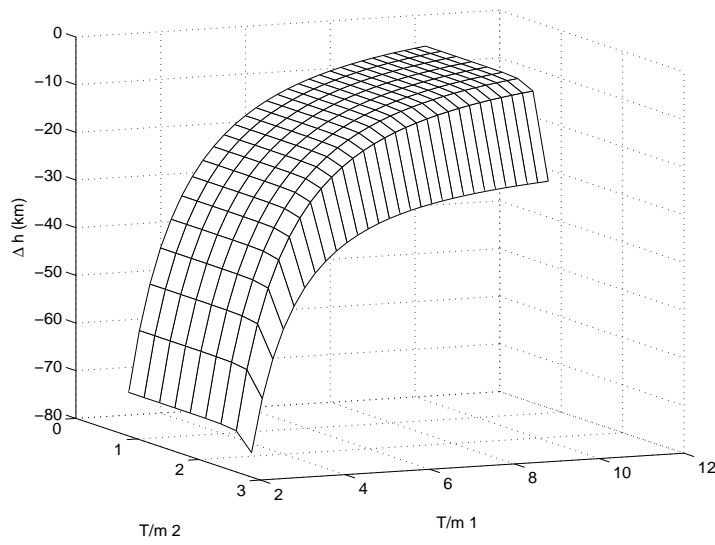


Figure 3.11: Three-Dimensional View of Altitude Sample Space

Depending on the trajectory design requirements, the trajectory space can be rather limited. Therefore an evaluation of the impact of changing the other variables is useful. The speeds are generally rather set; the initial speed is generally close to orbital speed and the final speed must be small for handover to the landing phase. However, a look into how their variance might impact the trajectory space is enlightening. In Fig. 3.12, the initial speed, v_0 , varies along with $\frac{T}{m}$ from $1800 \frac{m}{sec}$ to $1400 \frac{m}{sec}$ in $100 \frac{m}{sec}$ increments. In Fig. 3.13, the final speed, v_2 , varies along with $\frac{T}{m}$ from $101 \frac{m}{sec}$ to $1 \frac{m}{sec}$ in $20 \frac{m}{sec}$ increments. The flight path angles can also vary. In Fig. 3.14, the initial flight path angle varies from 0.1° to 5.1° in 0.5° increments. In Fig. 3.15, the final flight path angle varies from 49° to 89° in 2° increments. If all of these sample spaces are compared with Fig. 3.9, the impact of various variable variations is evident in the “blurring” of the very distinct curves of before.

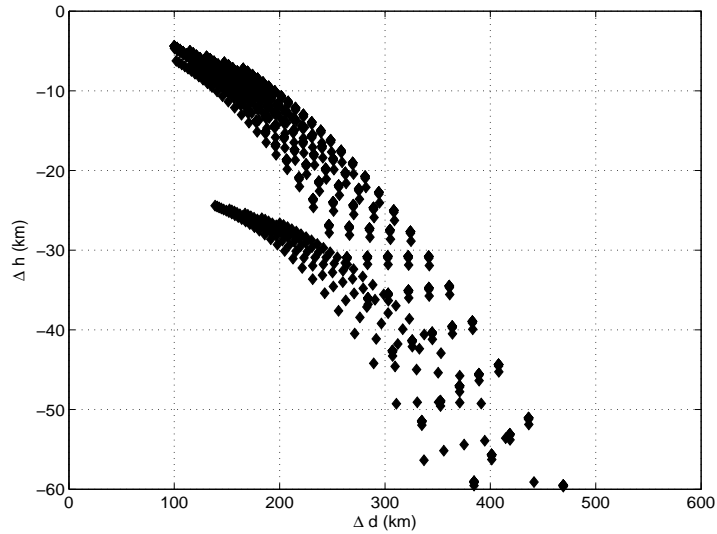


Figure 3.12: Sample Trajectory Space Varying $\frac{T}{m}$ and v_0 Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}$; $v_0 = [1800 : -100 : 1400] \frac{m}{sec}$

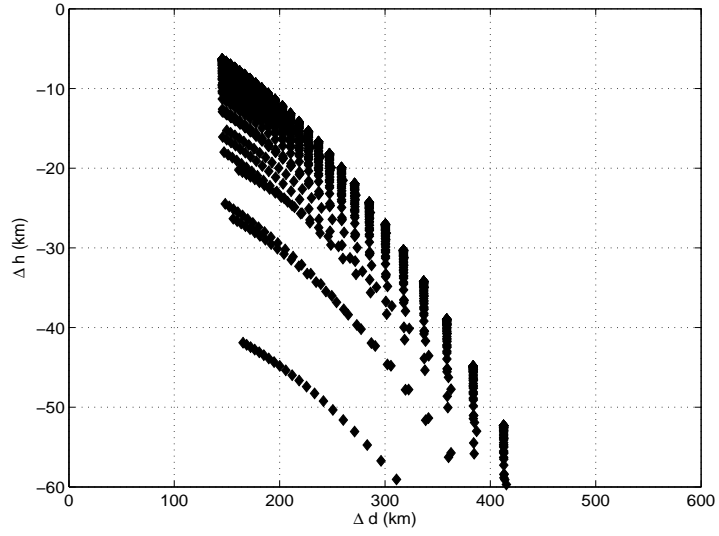


Figure 3.13: Sample Trajectory Space Varying $\frac{T}{m}$ and v_2 Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}$; $v_2 = [101 : -20 : 1] \frac{m}{sec}$

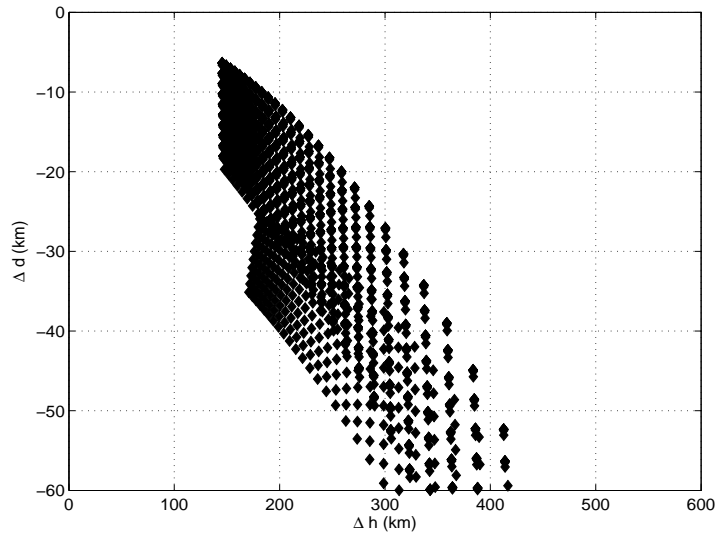


Figure 3.14: Sample Trajectory Space Varying $\frac{T}{m}$ and θ_0 Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}$; $\theta_0 = [0.1^\circ : 0.5^\circ : 5.1^\circ]$

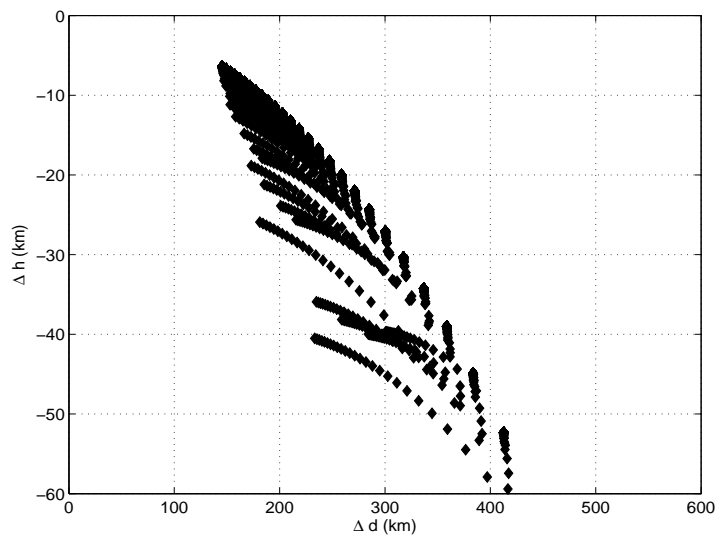


Figure 3.15: Sample Trajectory Space Varying $\frac{T}{m}$ and θ_f Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}$; $\theta_f = [49^\circ : 2^\circ : 89^\circ]$

The only variable remaining is the gravity term. Ideally this value is fixed to some appropriate value; the value used for the purposes of these figures is that at the lunar surface at the equator. However, it is instructive to assess the impact of its change on the trajectory space. Figure 3.16 illustrates that impact; the gravity was varied from $\frac{\mu}{h_0^2}$ to $\frac{3}{4} \frac{\mu}{h_0^2}$ in $\frac{1}{20} \frac{\mu}{h_0^2}$ increments.

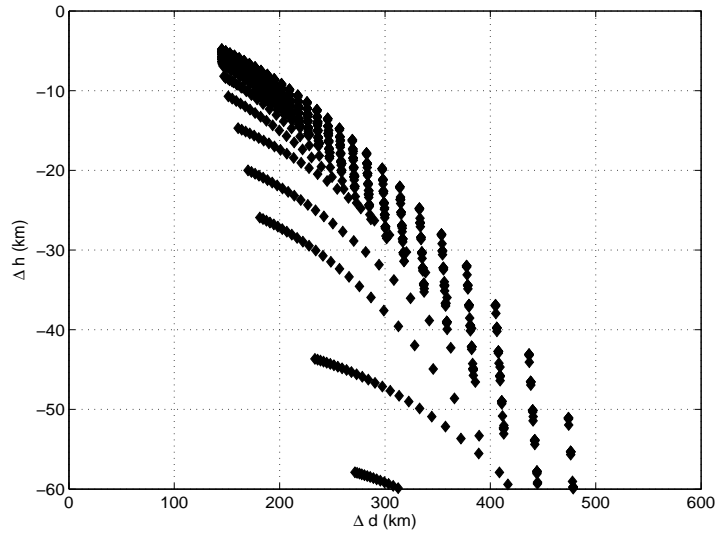


Figure 3.16: Sample Trajectory Space Varying $\frac{T}{m}$ and g Values: $\frac{T}{m_1}, \frac{T}{m_2} = [0.1 : 0.25 : 10.1] \frac{N}{kg}$; $g = [1 : -\frac{1}{20} : \frac{3}{4}] \frac{\mu}{h_0^2}$

Though changing the flight path angles, speeds, and gravity certainly have an impact on the trajectory space, that impact is not such that a whole segment of the space that was previously closed is opened. Therefore, the most desirable methodology is to simply vary $\frac{T}{m_1}$ and $\frac{T}{m_2}$ in order to generate the desired trajectory.

The most reliable methodology found was to generate a space of down-range versus altitude points given a set of realistic thrust distributions. The

points can then be sorted by the state that is most important to the user be that thrust values, downrange, or altitude (or even intermediary flight path angle or intermediary speed). The value that is closest to the state of importance can then be selected to define the desired trajectory.

A comparison against existing trajectory designs is always useful. A current design uses the existing Apollo era trajectory design algorithm and is given in [10]. The proposed design used the following parameters: $\theta_0 = 0.1^\circ$, $\theta_f = 77.4^\circ$, $v_0 = 1688 \frac{m}{sec}$, $v_f = 8 \frac{m}{sec}$, $\frac{T}{m_1} = 4.8 \frac{N}{kg}$, $\frac{T}{m_2} = 1.4 \frac{N}{kg}$, and $g = 1.136 \frac{m}{sec^2} = 0.7 \frac{\mu}{h_0^2}$. Figure 3.17 shows that the comparison of the flight path angle versus downrange for the two designs is quite close. Figure 3.18 illustrates that the proposed trajectory runs lower in altitude as compared to the existing trajectory. The slightly lower speed in Fig. 3.19 is likely the cause of the extended time in Fig. 3.20 (approximately 25 second increase in the timeline for the proposed trajectory). Fig. 3.21 shows that the acceleration for the proposed design is close to the average of the first segment and falls in the second segment as does the existing design.

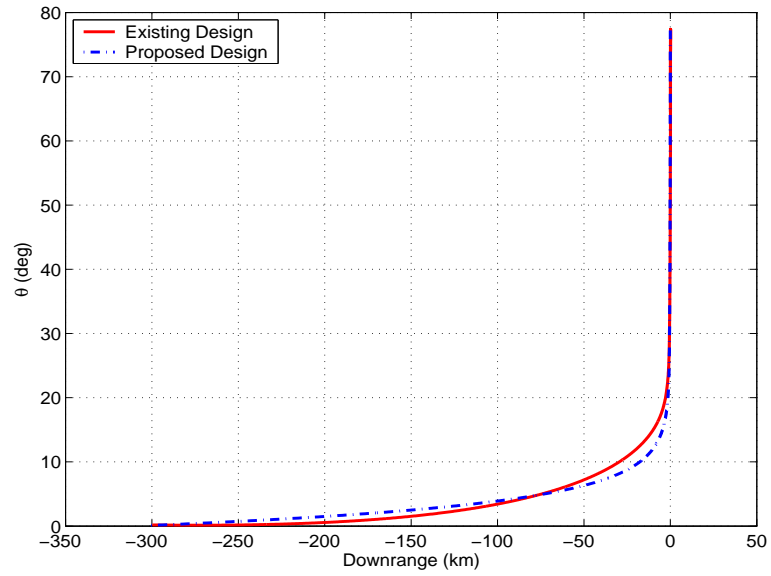


Figure 3.17: Comparison Between Proposed and Existing Trajectory: Flight Path Angle

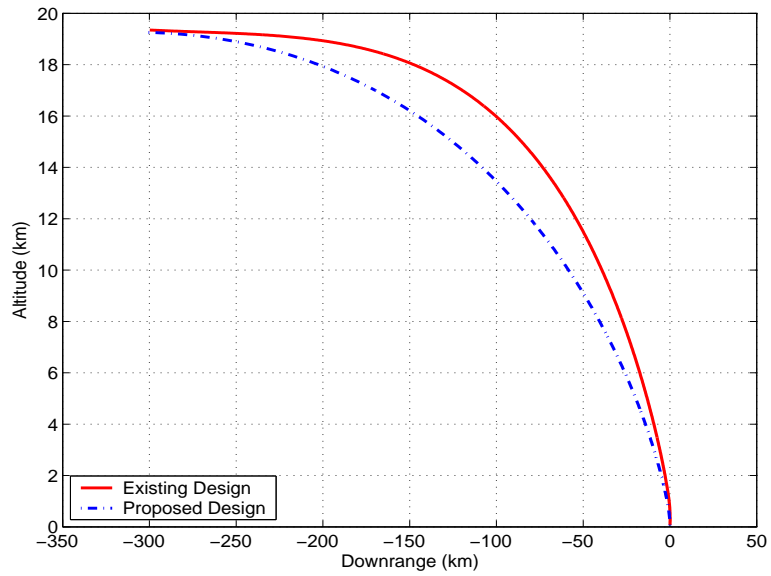


Figure 3.18: Comparison Between Proposed and Existing Trajectory: Altitude

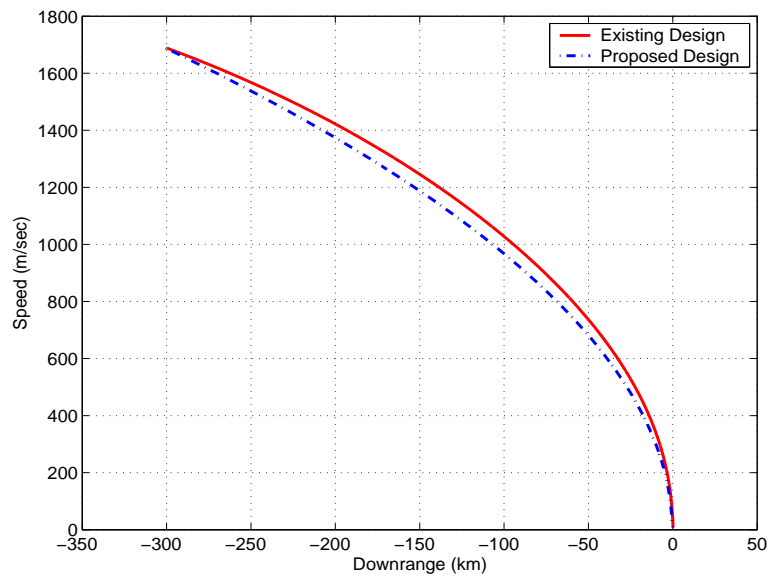


Figure 3.19: Comparison Between Proposed and Existing Trajectory: Speed

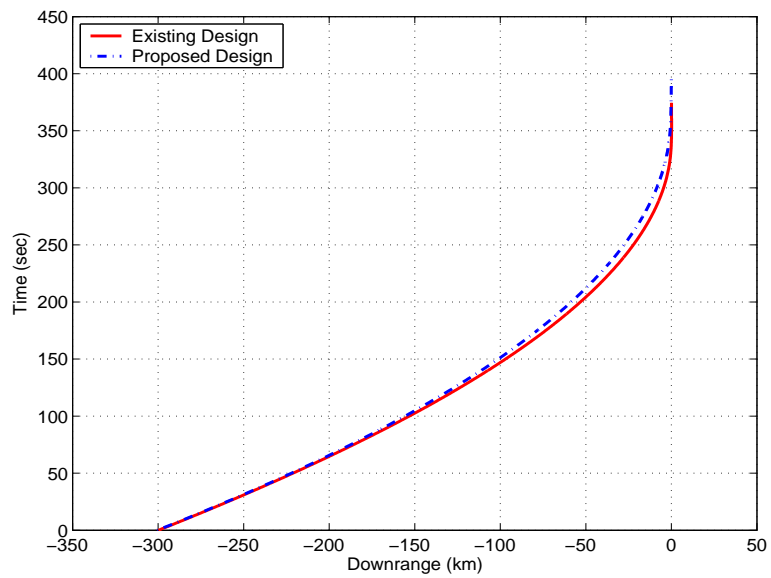


Figure 3.20: Comparison Between Proposed and Existing Trajectory: Time

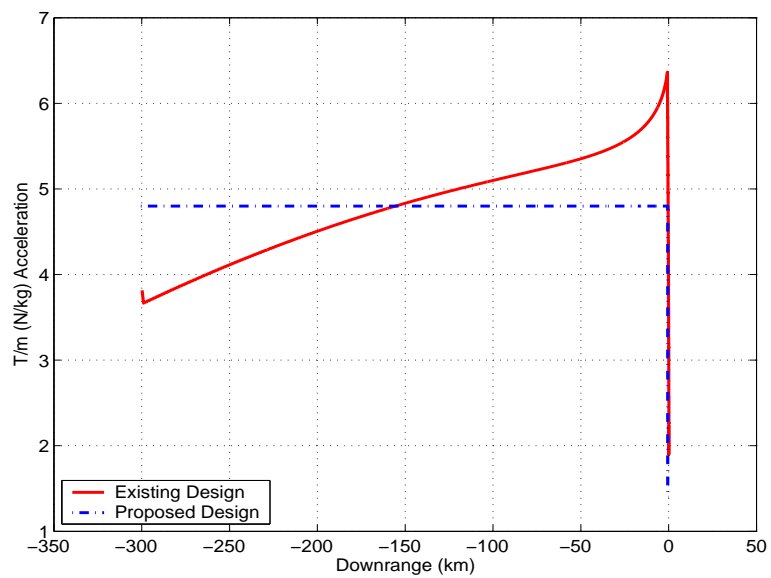


Figure 3.21: Comparison Between Proposed and Existing Trajectory: Acceleration

Chapter 4

Guidance Algorithm

4.1 Algorithm Design and Lyapunov Analysis

Once the target trajectory has been created, an algorithm that controls the spacecraft to that trajectory must be designed. The methodology chosen was to work with the non-linear equations of Chapter 2 rather than a linear model of those equations. One method by which to develop the guidance algorithm and to prove that the states converge to the target trajectory given that algorithm is to develop a Lyapunov candidate function that will be manipulated in a detailed proof. Through the formulations in this section, the goal will be to prove that the altitude, h , downrange, d , crossrange, c , speed, v , flight path angle, θ , and crossing angle, ψ , all converge to their respective target states denoted by h_{ref} , d_{ref} , c_{ref} , v_{ref} , θ_{ref} , and ψ_{ref} . The proof of the convergence of v_{ref} , θ_{ref} , and ψ_{ref} will be done by first proving that \dot{h} , \dot{d} , and \dot{c} converge to their target states.

The Lyapunov candidate function chosen for this application is

$$V = \frac{1}{2} \left[\dot{h}_{err}^2 + \dot{d}_{err}^2 + \dot{c}_{err}^2 + \frac{1}{\gamma_h} k_h^2 + \frac{1}{\gamma_d} k_d^2 + \frac{1}{\gamma_c} k_c^2 \right] \\ + \frac{\beta_h A}{\lambda_{h_1}} \ln \cosh \lambda_{h_1} h_{err} + \frac{\beta_d A}{\lambda_{d_1}} \ln \cosh \lambda_{d_1} d_{err} + \frac{\beta_c A}{\lambda_{c_1}} \ln \cosh \lambda_{c_1} c_{err}$$

where

$$\begin{aligned} h_{err} &= h - h_{ref} & d_{err} &= d - d_{ref} & c_{err} &= c - c_{ref} \\ \dot{h}_{err} &= \dot{h} - \dot{h}_{ref} & \dot{d}_{err} &= \dot{d} - \dot{d}_{ref} & \dot{c}_{err} &= \dot{c} - \dot{c}_{ref} \end{aligned}$$

and the variables k_h , k_d , and k_c are tuning functions of time. The value A is a positive value for the maximum expected acceleration. The gains λ_{h_1} , λ_{d_1} , λ_{c_1} , γ_h , γ_d , and γ_c are positive and $0 \leq \beta_h, \beta_d, \beta_c \leq 1$. Note that V is a positive function ($V \geq 0$) with these definitions (by definition $\cosh x \geq 1$ so $\ln \cosh x \geq 0$ for all x). This formulation of the Lyapunov candidate function was chosen because the $\ln \cosh x$ terms account for the fact that some acceleration limiting will be present. In the formulation of the controller, these terms will yield $\tanh x$ terms that are fundamentally limited to ± 1 . The tuning functions k_h , k_d , and k_c will add some complexity to the controller, but yield an algorithm that performs very well by balancing the proportional control with the derivative control in a manner that is straightforward to tune.

Taking the derivative of the Lyapunov function yields

$$\begin{aligned} \dot{V} &= \dot{h}_{err} \left(\ddot{h}_{err} + \beta_h A \tanh \lambda_{h_1} h_{err} \right) + \frac{1}{\gamma_h} k_h \dot{k}_h \\ &\quad + \dot{d}_{err} \left(\ddot{d}_{err} + \beta_d A \tanh \lambda_{d_1} d_{err} \right) + \frac{1}{\gamma_d} k_d \dot{k}_d \\ &\quad + \dot{c}_{err} \left(\ddot{c}_{err} + \beta_c A \tanh \lambda_{c_1} c_{err} \right) + \frac{1}{\gamma_c} k_c \dot{k}_c \end{aligned}$$

If the control law makes the following true (this will be shown to be the case later):

$$\ddot{h}_{err} + \beta_h A \tanh \lambda_{h_1} h_{err} = -(1 - \beta_h) A \tanh \lambda_{h_2} \left(\dot{h}_{err} + k_h h_{err} \right) \quad (4.1)$$

$$\ddot{d}_{err} + \beta_d A \tanh \lambda_{d_1} d_{err} = -(1 - \beta_d) A \tanh \lambda_{d_2} \left(\dot{d}_{err} + k_d d_{err} \right) \quad (4.2)$$

$$\ddot{c}_{err} + \beta_c A \tanh \lambda_{c_1} c_{err} = -(1 - \beta_c) A \tanh \lambda_{c_2} \left(\dot{c}_{err} + k_c c_{err} \right) \quad (4.3)$$

where λ_{h_2} , λ_{d_2} , and λ_{c_2} are additional positive control gains then

$$\begin{aligned}
\dot{V} = & - \left(\dot{h}_{err} + k_h h_{err} \right) (1 - \beta_h) A \tanh \lambda_{h_2} \left(\dot{h}_{err} + k_h h_{err} \right) \\
& + k_h \left(\frac{1}{\gamma_h} \dot{k}_h + h_{err} (1 - \beta_h) A \tanh \lambda_{h_2} \left(\dot{h}_{err} + k_h h_{err} \right) \right) \\
& - \left(\dot{d}_{err} + k_d d_{err} \right) (1 - \beta_d) A \tanh \lambda_{d_2} \left(\dot{d}_{err} + k_d d_{err} \right) \\
& + k_d \left(\frac{1}{\gamma_d} \dot{k}_d + d_{err} (1 - \beta_d) A \tanh \lambda_{d_2} \left(\dot{d}_{err} + k_d d_{err} \right) \right) \\
& - \left(\dot{c}_{err} + k_c c_{err} \right) (1 - \beta_c) A \tanh \lambda_{c_2} \left(\dot{c}_{err} + k_c c_{err} \right) \\
& + k_c \left(\frac{1}{\gamma_c} \dot{k}_c + c_{err} (1 - \beta_c) A \tanh \lambda_{c_2} \left(\dot{c}_{err} + k_c c_{err} \right) \right)
\end{aligned}$$

If the tuning function is implemented such that

$$\begin{aligned}
\dot{k}_h = & -\gamma_h h_{err} (1 - \beta_h) A \tanh \lambda_{h_2} \left(\dot{h}_{err} + k_h h_{err} \right) \\
& -\gamma_h h_{err} \tanh k_h h_{err}
\end{aligned} \tag{4.4}$$

$$\begin{aligned}
\dot{k}_d = & -\gamma_d d_{err} (1 - \beta_d) A \tanh \lambda_{d_2} \left(\dot{d}_{err} + k_d d_{err} \right) \\
& -\gamma_d d_{err} \tanh k_d d_{err}
\end{aligned} \tag{4.5}$$

$$\begin{aligned}
\dot{k}_c = & -\gamma_c c_{err} (1 - \beta_c) A \tanh \lambda_{c_2} \left(\dot{c}_{err} + k_c c_{err} \right) \\
& -\gamma_c c_{err} \tanh k_c c_{err}
\end{aligned} \tag{4.6}$$

then

$$\begin{aligned}
\dot{V} = & - \left(\dot{h}_{err} + k_h h_{err} \right) (1 - \beta_h) A \tanh \lambda_{h_2} \left(\dot{h}_{err} + k_h h_{err} \right) \\
& - k_h h_{err} \tanh k_h h_{err} \\
& - \left(\dot{d}_{err} + k_d d_{err} \right) (1 - \beta_d) A \tanh \lambda_{d_2} \left(\dot{d}_{err} + k_d d_{err} \right) \\
& - k_d d_{err} \tanh k_d d_{err} \\
& - \left(\dot{c}_{err} + k_c c_{err} \right) (1 - \beta_c) A \tanh \lambda_{c_2} \left(\dot{c}_{err} + k_c c_{err} \right) \\
& - k_c c_{err} \tanh k_c c_{err}
\end{aligned}$$

meaning that $\dot{V} \leq 0$ and V is a non-increasing function (any function $x \tanh x \geq 0$).

It has already been shown that the Lyapunov candidate function V is positive and non-increasing if the gains $\lambda_{h1}, \lambda_{d1}, \lambda_{c1}, \lambda_{h2}, \lambda_{d2}, \lambda_{c2}, \gamma_h, \gamma_d$, and γ_c are positive and $0 \leq \beta_h, \beta_d, \beta_c \leq 1$. Therefore, it is upper bounded and its limit as $t \rightarrow \infty$ exists and is finite:

$$\begin{aligned} \lim_{t \rightarrow \infty} V(t) &\triangleq V_\infty \\ V(t) &\in L_\infty \end{aligned}$$

If V is bounded, then the signals that it is comprised of must also be bounded. Because V contains the squared norms of $\dot{h}_{err}, \dot{d}_{err}, \dot{c}_{err}, k_h, k_d$, and k_d and the function $\ln \cosh x$ being bounded implies that x must be bounded,

$$h_{err}, d_{err}, c_{err}, \dot{h}_{err}, \dot{d}_{err}, \dot{c}_{err}, k_h, k_d, k_d \in L_\infty$$

Assuming again that Eqs. 4.1, 4.2, and 4.3 are true, two of the three terms in each equation are bounded (note $-1 \leq \tanh x \leq 1$), so the third term in each must also be bounded implying

$$\ddot{h}_{err}, \ddot{d}_{err}, \ddot{c}_{err} \in L_\infty$$

Also note that in Eqs. 4.4, 4.5, and 4.6, the signals on the right hand side of the equations are bounded so

$$\dot{k}_h, \dot{k}_d, \dot{k}_c \in L_\infty$$

Now that all of the signals in the system have been proven to be bounded, the next step is to prove that the error states asymptotically approach zero as a function of time. The first signals investigated will be \dot{h}_{err} ,

\dot{d}_{err} , and \dot{c}_{err} . Because the Lyapunov function is bounded and its limit exists and is finite

$$\lim_{t \rightarrow \infty} \int_0^t \dot{V} d\tau = V_\infty - V_0$$

Taking the derivative of \dot{V} yields

$$\begin{aligned} \ddot{V} = & - \left(\ddot{h}_{err} + \dot{k}_h h_{err} + k_h \dot{h}_{err} \right) (1 - \beta_h) A \tanh \lambda_{h_2} \left(\dot{h}_{err} + k_h h_{err} \right) \\ & - \left(\dot{h}_{err} + k_h h_{err} \right) (1 - \beta_h) A \frac{\lambda_{h_2} \left(\ddot{h}_{err} + \dot{k}_h h_{err} + k_h \dot{h}_{err} \right)}{\cosh^2 \lambda_{h_2} \left(\dot{h}_{err} + k_h h_{err} \right)} \\ & - \left(\dot{k}_h h_{err} + k_h \dot{h}_{err} \right) \tanh k_h h_{err} - k_h h_{err} \frac{\dot{k}_h h_{err} + k_h \dot{h}_{err}}{\cosh^2 k_h h_{err}} \\ & - \left(\ddot{d}_{err} + \dot{k}_d d_{err} + k_d \dot{d}_{err} \right) (1 - \beta_d) A \tanh \lambda_{d_2} \left(\dot{d}_{err} + k_d d_{err} \right) \\ & - \left(\dot{d}_{err} + k_d d_{err} \right) (1 - \beta_d) A \frac{\lambda_{d_2} \left(\ddot{d}_{err} + \dot{k}_d d_{err} + k_d \dot{d}_{err} \right)}{\cosh^2 \lambda_{d_2} \left(\dot{d}_{err} + k_d d_{err} \right)} \\ & - \left(\dot{k}_d d_{err} + k_d \dot{d}_{err} \right) \tanh k_d d_{err} - k_d d_{err} \frac{\dot{k}_d d_{err} + k_d \dot{d}_{err}}{\cosh^2 k_d d_{err}} \\ & - \left(\ddot{c}_{err} + \dot{k}_c c_{err} + k_c \dot{c}_{err} \right) (1 - \beta_c) A \tanh \lambda_{c_2} \left(\dot{c}_{err} + k_c c_{err} \right) \\ & - \left(\dot{c}_{err} + k_c c_{err} \right) (1 - \beta_c) A \frac{\lambda_{c_2} \left(\ddot{c}_{err} + \dot{k}_c c_{err} + k_c \dot{c}_{err} \right)}{\cosh^2 \lambda_{c_2} \left(\dot{c}_{err} + k_c c_{err} \right)} \\ & - \left(\dot{k}_c c_{err} + k_c \dot{c}_{err} \right) \tanh k_c c_{err} - k_c c_{err} \frac{\dot{k}_c c_{err} + k_c \dot{c}_{err}}{\cosh^2 k_c c_{err}} \end{aligned}$$

which is a bounded function because all of the signals on the right hand side of this equation are bounded (note $0 \leq \frac{1}{\cosh^2 x} \leq 1$). This means that \dot{V} is uniformly continuous ([9]). Because \dot{V} is uniformly continuous and its integral exists and is finite, by Barbalet's Lemma ([9])

$$\lim_{t \rightarrow \infty} \dot{V}(t) = 0$$

Therefore,

$$\begin{aligned}
\lim_{t \rightarrow \infty} \left(\dot{h}_{err} + k_h h_{err} \right) \tanh \lambda_{h_2} \left(\dot{h}_{err} + k_h h_{err} \right) &= 0 \\
\lim_{t \rightarrow \infty} k_h h_{err} \tanh k_h h_{err} &= 0 \\
\lim_{t \rightarrow \infty} \left(\dot{d}_{err} + k_d d_{err} \right) \tanh \lambda_{d_2} \left(\dot{d}_{err} + k_d d_{err} \right) &= 0 \\
\lim_{t \rightarrow \infty} k_d d_{err} \tanh k_d d_{err} &= 0 \\
\lim_{t \rightarrow \infty} \left(\dot{c}_{err} + k_c c_{err} \right) \tanh \lambda_{c_2} \left(\dot{c}_{err} + k_c c_{err} \right) &= 0 \\
\lim_{t \rightarrow \infty} k_c c_{err} \tanh k_c c_{err} &= 0
\end{aligned}$$

or

$$\begin{aligned}
\lim_{t \rightarrow \infty} \left(\dot{h}_{err} + k_h h_{err} \right) &= 0 \\
\lim_{t \rightarrow \infty} k_h h_{err} &= 0 \\
\lim_{t \rightarrow \infty} \left(\dot{d}_{err} + k_d d_{err} \right) &= 0 \\
\lim_{t \rightarrow \infty} k_d d_{err} &= 0 \\
\lim_{t \rightarrow \infty} \left(\dot{c}_{err} + k_c c_{err} \right) &= 0 \\
\lim_{t \rightarrow \infty} k_c c_{err} &= 0
\end{aligned}$$

which implies

$$\begin{aligned}
\lim_{t \rightarrow \infty} \dot{h}_{err}(t) &= 0 \\
\lim_{t \rightarrow \infty} \dot{d}_{err}(t) &= 0 \\
\lim_{t \rightarrow \infty} \dot{c}_{err}(t) &= 0
\end{aligned}$$

The next signals investigated will be \ddot{h}_{err} , \ddot{d}_{err} , and \ddot{c}_{err} . The derivatives

of Eqs. 4.1, 4.2, and 4.3 are

$$\begin{aligned}\ddot{h}_{err} &= -\beta_h A \frac{\lambda_{h_1} \dot{h}_{err}}{\cosh^2 \lambda_{h_1} h_{err}} - (1 - \beta_h) A \frac{\lambda_{h_2} (\ddot{h}_{err} + \dot{k}_h h_{err} + k_h \dot{h}_{err})}{\cosh^2 \lambda_{h_2} (\dot{h}_{err} + k_h h_{err})} \\ \ddot{d}_{err} &= -\beta_d A \frac{\lambda_{d_1} \dot{d}_{err}}{\cosh^2 \lambda_{d_1} d_{err}} - (1 - \beta_d) A \frac{\lambda_{d_2} (\ddot{d}_{err} + \dot{k}_d d_{err} + k_d \dot{d}_{err})}{\cosh^2 \lambda_{d_2} (\dot{d}_{err} + k_d d_{err})} \\ \ddot{c}_{err} &= -\beta_c A \frac{\lambda_{c_1} \dot{c}_{err}}{\cosh^2 \lambda_{c_1} c_{err}} - (1 - \beta_c) A \frac{\lambda_{c_2} (\ddot{c}_{err} + \dot{k}_c c_{err} + k_c \dot{c}_{err})}{\cosh^2 \lambda_{c_2} (\dot{c}_{err} + k_c c_{err})}\end{aligned}$$

The right hand sides of these equations are bounded so the left hand sides must be as well or

$$\ddot{h}_{err}, \ddot{d}_{err}, \ddot{c}_{err} \in L_\infty$$

The fact that the derivatives of \ddot{h}_{err} , \ddot{d}_{err} , and \ddot{c}_{err} are bounded means that \dot{h}_{err} , \dot{d}_{err} , and \dot{c}_{err} are uniformly continuous ([9]).

Now note that

$$\begin{aligned}\int_0^t \ddot{h}_{err}(\tau) d\tau &= \dot{h}_{err}(t) - \dot{h}_{err}(0) \\ \int_0^t \ddot{d}_{err}(\tau) d\tau &= \dot{d}_{err}(t) - \dot{d}_{err}(0) \\ \int_0^t \ddot{c}_{err}(\tau) d\tau &= \dot{c}_{err}(t) - \dot{c}_{err}(0)\end{aligned}$$

and recall that

$$\begin{aligned}\lim_{t \rightarrow \infty} \dot{h}_{err}(t) &= 0 \\ \lim_{t \rightarrow \infty} \dot{d}_{err}(t) &= 0 \\ \lim_{t \rightarrow \infty} \dot{c}_{err}(t) &= 0\end{aligned}$$

Therefore,

$$\begin{aligned}\lim_{t \rightarrow \infty} \int_0^t \ddot{h}_{err}(\tau) d\tau + \dot{h}_{err}(0) &= 0 \\ \lim_{t \rightarrow \infty} \int_0^t \ddot{d}_{err}(\tau) d\tau + \dot{d}_{err}(0) &= 0 \\ \lim_{t \rightarrow \infty} \int_0^t \ddot{c}_{err}(\tau) d\tau + \dot{c}_{err}(0) &= 0\end{aligned}$$

meaning that the integrals of $\ddot{h}_{err}(t)$, $\ddot{d}_{err}(t)$, and $\ddot{c}_{err}(t)$ exist and are finite with the valid assumption that $\dot{h}_{err}(0)$, $\dot{d}_{err}(0)$, and $\dot{c}_{err}(0)$ are finite. It has been proven that the signals $\ddot{h}_{err}(t)$, $\ddot{d}_{err}(t)$, and $\ddot{c}_{err}(t)$ are uniformly continuous so, with the fact that their integrals exist and are finite, by Barbalet's Lemma ([9]),

$$\begin{aligned}\lim_{t \rightarrow \infty} \ddot{h}_{err}(t) &= 0 \\ \lim_{t \rightarrow \infty} \ddot{d}_{err}(t) &= 0 \\ \lim_{t \rightarrow \infty} \ddot{c}_{err}(t) &= 0\end{aligned}$$

Now the final signals (h_{err} , d_{err} , and c_{err}) must be investigated. Again, from Eqs. 4.1, 4.2, and 4.3,

$$\begin{aligned}\beta_h A \tanh \lambda_{h_1} h_{err} &= -\ddot{h}_{err} - (1 - \beta_h) A \tanh \lambda_{h_2} (\dot{h}_{err} + k_h h_{err}) \\ \beta_d A \tanh \lambda_{d_1} d_{err} &= -\ddot{d}_{err} - (1 - \beta_d) A \tanh \lambda_{d_2} (\dot{d}_{err} + k_d d_{err}) \\ \beta_c A \tanh \lambda_{c_1} c_{err} &= -\ddot{c}_{err} - (1 - \beta_c) A \tanh \lambda_{c_2} (\dot{c}_{err} + k_c c_{err})\end{aligned}$$

All signals on the right hand side of these equations converge to zero so all three error signals on the left hand side converge to zero given an infinite amount of time. Therefore, because the function $\tanh x$ approaching zero implies that x

approaches zero,

$$\begin{aligned}\lim_{t \rightarrow \infty} h_{err}(t) &= 0 \\ \lim_{t \rightarrow \infty} d_{err}(t) &= 0 \\ \lim_{t \rightarrow \infty} c_{err}(t) &= 0\end{aligned}$$

or

$$h \rightarrow h_{ref} \quad d \rightarrow d_{ref} \quad c \rightarrow c_{ref}$$

The fact that

$$\begin{aligned}\lim_{t \rightarrow \infty} \dot{h}_{err}(t) &= 0 \\ \lim_{t \rightarrow \infty} \dot{d}_{err}(t) &= 0 \\ \lim_{t \rightarrow \infty} \dot{c}_{err}(t) &= 0\end{aligned}$$

implies that

$$\dot{h} \rightarrow \dot{h}_{ref} \quad \dot{d} \rightarrow \dot{d}_{ref} \quad \dot{c} \rightarrow \dot{c}_{ref}$$

or

$$\begin{aligned}v \sin \theta &\rightarrow v_{ref} \sin \theta_{ref} \\ \frac{h_0}{h + h_0} v \cos \theta \cos \psi &\rightarrow \frac{h_0}{h_{ref} + h_0} v_{ref} \cos \theta_{ref} \cos \psi_{ref} \\ \frac{h_0}{h + h_0} v \cos \theta \sin \psi &\rightarrow \frac{h_0}{h_{ref} + h_0} v_{ref} \cos \theta_{ref} \sin \psi_{ref}\end{aligned}$$

This implies that

$$v \rightarrow v_{ref} \quad \theta \rightarrow \theta_{ref} \quad \psi \rightarrow \psi_{ref}$$

Now that all states have been proven to converge to their references, Eqs. 4.1, 4.2, and 4.3 must be utilized to define the control law. Recall the fundamental equations of motion for the kinematic states are

$$\begin{aligned}\dot{h} &= -v \sin \theta \\ \dot{d} &= v \cos \theta \cos \psi \frac{h_0}{h + h_0} \\ \dot{c} &= v \cos \theta \sin \psi \frac{h_0}{h + h_0}\end{aligned}$$

Their time derivatives will be used in Eqs. 4.1, 4.2, and 4.3 to account for the expected dynamics of the spacecraft:

$$\ddot{h} = -\dot{v} \sin \theta - v \dot{\theta} \cos \theta \tag{4.7}$$

$$\begin{aligned}\ddot{d} &= \left(\dot{v} \cos \theta \cos \psi - v \dot{\theta} \sin \theta \cos \psi - v \dot{\psi} \cos \theta \sin \psi \right) \frac{h_0}{h + h_0} \\ &\quad - v \cos \theta \cos \psi \frac{h_0 \dot{h}}{(h + h_0)^2}\end{aligned} \tag{4.8}$$

$$\begin{aligned}\ddot{c} &= \left(\dot{v} \cos \theta \sin \psi - v \dot{\theta} \sin \theta \sin \psi + v \dot{\psi} \cos \theta \cos \psi \right) \frac{h_0}{h + h_0} \\ &\quad - v \cos \theta \sin \psi \frac{h_0 \dot{h}}{(h + h_0)^2}\end{aligned} \tag{4.9}$$

Rearranging terms in Eqs. 4.1, 4.2, and 4.3 and combining these with Eqs.

4.7, 4.8, and 4.9 yields the following:

$$\begin{aligned}
& -\dot{v} \sin \theta - v \dot{\theta} \cos \theta = \\
& -\beta_h A \tanh \lambda_{h_1} h_{err} - (1 - \beta_h) A \tanh \lambda_{h_2} \left(\dot{h}_{err} + k_h h_{err} \right) + \ddot{h}_{ref} \\
& \left(\dot{v} \cos \theta \cos \psi - v \dot{\theta} \sin \theta \cos \psi - v \dot{\psi} \cos \theta \sin \psi - \frac{v \cos \theta \cos \psi \dot{h}}{h + h_0} \right) \frac{h_0}{h + h_0} = \\
& -\beta_d A \tanh \lambda_{d_1} d_{err} - (1 - \beta_d) A \tanh \lambda_{d_2} \left(\dot{d}_{err} + k_d d_{err} \right) + \ddot{d}_{ref} \\
& \left(\dot{v} \cos \theta \sin \psi - v \dot{\theta} \sin \theta \sin \psi + v \dot{\psi} \cos \theta \cos \psi - \frac{v \cos \theta \sin \psi \dot{h}}{h + h_0} \right) \frac{h_0}{h + h_0} = \\
& -\beta_c A \tanh \lambda_{c_1} c_{err} - (1 - \beta_c) A \tanh \lambda_{c_2} \left(\dot{c}_{err} + k_c c_{err} \right) + \ddot{c}_{ref}
\end{aligned}$$

Recall that the fundamental equations of motion for the dynamic states are

$$\begin{aligned}
\dot{v} &= g \sin \theta - \frac{T}{m} \cos \alpha \\
v \dot{\theta} &= \left(g - \frac{v^2}{h + h_0} \right) \cos \theta - \frac{T}{m} \sin \alpha \cos \phi \\
v \dot{\psi} \cos \theta &= -\frac{T}{m} \sin \alpha \sin \phi
\end{aligned}$$

If these are substituted into the previous equations, the resulting equation that must be solved in order to compute the control variables is

$$\begin{bmatrix} \sin \theta & \cos \theta & 0 \\ -\cos \theta \cos \psi & \sin \theta \cos \psi & \sin \psi \\ -\cos \theta \sin \psi & \sin \theta \sin \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} \frac{T}{m} \cos \alpha \\ \frac{T}{m} \sin \alpha \cos \phi \\ \frac{T}{m} \sin \alpha \sin \phi \end{bmatrix} = \begin{bmatrix} H \\ D \\ C \end{bmatrix}$$

where

$$\begin{aligned}
H &= \ddot{h}_{ref} + g - \frac{v^2}{h + h_0} \cos^2 \theta \\
&\quad - \beta_h A \tanh \lambda_{h_1} h_{err} - (1 - \beta_h) A \tanh \lambda_{h_2} (\dot{h}_{err} + k_h h_{err}) \\
D &= \ddot{d}_{ref} \frac{h + h_0}{h_0} - \frac{2v^2}{h + h_0} \cos \theta \sin \theta \cos \psi \\
&\quad - \left(\beta_d A \tanh \lambda_{d_1} d_{err} + (1 - \beta_d) A \tanh \lambda_{d_2} (\dot{d}_{err} + k_d d_{err}) \right) \frac{h + h_0}{h_0} \\
C &= \ddot{c}_{ref} \frac{h + h_0}{h_0} - \frac{2v^2}{h + h_0} \cos \theta \sin \theta \sin \psi \\
&\quad - (\beta_c A \tanh \lambda_{c_1} c_{err} + (1 - \beta_c) A \tanh \lambda_{c_2} (\dot{c}_{err} + k_c c_{err})) \frac{h + h_0}{h_0}
\end{aligned}$$

This can be inverted so that if the control variables $\frac{T}{m}$, α , and ϕ are given by

$$\begin{bmatrix} \frac{T}{m} \cos \alpha \\ \frac{T}{m} \sin \alpha \cos \phi \\ \frac{T}{m} \sin \alpha \sin \phi \end{bmatrix} = \begin{bmatrix} \sin \theta & -\cos \theta \cos \psi & -\cos \theta \sin \psi \\ \cos \theta & \sin \theta \cos \psi & \sin \theta \sin \psi \\ 0 & \sin \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} H \\ D \\ C \end{bmatrix} \quad (4.10)$$

all states are guaranteed to converge to their reference trajectories in an infinite amount of time.

Equation 4.10 is the basis for the guidance algorithm along with the numerical integration of Eqs. 4.4, 4.5, and 4.6. Equation 4.10 can be rewritten in the form

$$\begin{bmatrix} \frac{T}{m} \cos \alpha \\ \frac{T}{m} \sin \alpha \cos \phi \\ \frac{T}{m} \sin \alpha \sin \phi \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}$$

The solution for the control variables is then

$$\begin{aligned}
\frac{T}{m} &= \sqrt{n_1^2 + n_2^2 + n_3^2} \\
\alpha &= \tan^{-1} \left(\frac{\sqrt{n_2^2 + n_3^2}}{n_1} \right) \\
\phi &= \tan^{-1} \left(\frac{n_3}{n_2} \right)
\end{aligned}$$

In order to make the control angle results more intuitive, some additional modifications are made to the angles. The major modification is to force the thrust angle, α , to be in an intuitive direction given the thrust roll angle, ϕ . Therefore, if $\phi > \frac{\pi}{2}$, $\alpha = -\alpha$ and $\phi = \phi - \pi$. Conversely, if $\phi < \frac{-\pi}{2}$, $\alpha = -\alpha$ and $\phi = \phi + \pi$. The second modification simply makes ϕ an always positive angle so that it, again, is more intuitive: if $\phi < 0$, $\phi = 2\pi + \phi$.

4.2 Performance in a Test Setting For Gain Selection

To adjust the gains in the guidance algorithm properly, a target trajectory was contrived to thoroughly test the guidance response in a nominal sense. In an attempt to create a situation such that the simulated environment is as close to the environment assumed by the targeting algorithm, the simulated "true" mass and gravity were held constant. The thrust levels were then varied in a manner that the spacecraft is unlikely to see in a realistic setting but would test the real-time guidance response to discrete changes in thrust level. The gains selected are given in Table 4.1.

The selection of the acceleration limit, A , is an integral component of the gain selection. The terms that address acceleration limiting were included in the formulation to account for the fact that the guidance will not be allowed an infinite amount of thrust. It would seem that the value for A should simply be the maximum thrust acceleration expected at any point in time; however, this is not the case. First, the thrust force value is the element that will likely be limited for the spacecraft rather than the thrust acceleration limit. Therefore, the thrust acceleration limit at any point in time is given by the maximum allowed thrust divided by the mass estimate at that time; because mass varies with time the acceleration limit actually varies with time. For

Table 4.1: Real-Time Guidance Gains

Gain	Value
λ_{h_1}	0.05
λ_{d_1}	0.05
λ_{c_1}	0.005
λ_{h_2}	1.0
λ_{d_2}	1.0
λ_{c_2}	1.0
β_h	0.45
β_d	0.45
β_c	0.45
γ_h	0.00001
γ_d	0.00001
γ_c	0.000001
$k_h(0)$	0.01
$k_d(0)$	0.01
$k_c(0)$	0.01

all of the results presented in this document, the thrust force magnitude was limited to 266.9 kN (value from [10]).

The next issue is that this thrust acceleration is divided among three different errors. At any one point, all three axes can require the maximum thrust for error removal thereby exceeding the allowed thrust. This means that the thrust force must be limited by dividing 266.9 kN by the $\sqrt{3}$ in order to keep the magnitude of the requested thrust below the limit; a thrust force limit of 154.1 kN was set (recall this is then divided by the estimated mass to obtain the thrust acceleration limit, A).

The final issue is that this still provides no guarantee that the thrust command will remain below the limit because of the nature of the selected

guidance algorithm. From Eq. 4.10

$$\left| \frac{T}{m} \right|^2 \leq H^2 + D^2 + C^2$$

where

$$\begin{aligned} H &= -g_{ref} + \frac{v_{ref}^2}{h_{ref} + h_0} \cos^2 \theta_{ref} + \frac{T}{m_{ref}} \sin \theta_{ref} + g - \frac{v^2}{h + h_0} \cos^2 \theta \\ &\quad - \beta_h A \tanh \lambda_{h_1} h_{err} - (1 - \beta_h) A \tanh \lambda_{h_2} (\dot{h}_{err} + k_h h_{err}) \\ D &= \frac{2v_{ref}^2 (h + h_0)}{(h_{ref} + h_0)^2} \cos \theta_{ref} \sin \theta_{ref} - \frac{T}{m_{ref}} \cos \theta_{ref} \frac{h + h_0}{h_{ref} + h_0} \\ &\quad - \frac{2v^2}{h + h_0} \cos \theta \sin \theta \cos \psi \\ &\quad - \left(\beta_d A \tanh \lambda_{d_1} d_{err} + (1 - \beta_d) A \tanh \lambda_{d_2} (\dot{d}_{err} + k_d d_{err}) \right) \frac{h + h_0}{h_0} \\ C &= -\frac{2v^2}{h + h_0} \cos \theta \sin \theta \sin \psi \\ &\quad - (\beta_c A \tanh \lambda_{c_1} c_{err} + (1 - \beta_c) A \tanh \lambda_{c_2} (\dot{c}_{err} + k_c c_{err})) \frac{h + h_0}{h_0} \end{aligned}$$

To simplify this analysis now realize that $h \ll h_0$ and $h_{ref} \ll h_0$ so that

$$\begin{aligned} H &= \Delta \text{grav}_h + \frac{T}{m_{ref}} \sin \theta_{ref} \\ &\quad - \beta_h A \tanh \lambda_{h_1} h_{err} - (1 - \beta_h) A \tanh \lambda_{h_2} (\dot{h}_{err} + k_h h_{err}) \\ D &= \Delta \text{grav}_d - \frac{T}{m_{ref}} \cos \theta_{ref} \\ &\quad - \beta_d A \tanh \lambda_{d_1} d_{err} - (1 - \beta_d) A \tanh \lambda_{d_2} (\dot{d}_{err} + k_d d_{err}) \\ C &= \Delta \text{grav}_c + \\ &\quad - \beta_c A \tanh \lambda_{c_1} c_{err} - (1 - \beta_c) A \tanh \lambda_{c_2} (\dot{c}_{err} + k_c c_{err}) \end{aligned}$$

Therefore

$$\left| \frac{T}{m} \right|^2 \leq \Delta \text{grav}^2 + \left(\frac{T}{m} \right)_{ref}^2 + 3A^2$$

The need for the division by the $\sqrt{3}$ is evident from this equation, but so is the fact that, if a difference exists between the modeled gravity and the real gravity (Δ_{grav} terms), the thrust acceleration command can be greater than the allowed acceleration. In addition, the reference thrust acceleration command complicates the issue. Because this equation only illustrates the maximum thrust acceleration command that is possible, the decision was made to limit the acceleration command that results from Eq. 4.10 rather than to modify the value for A any further. Therefore, a simple line of code exists within this guidance algorithm that limits the thrust command to its maximum allowable value before the command is issued; that maximum value is 266.9 kN (value from [10]) for all results presented from here on.

The acceleration profile for the chosen target trajectory is shown in Figure 4.1 along with the time history response of the actual commanded thrust. The response of the guidance algorithm appears to be critically damped. In the absence of any concrete requirements for the guidance, this was deemed a desirable response. The response of the thrust angle is shown in Fig. 4.2.

The downrange response as a function of time is shown in Fig. 4.3. The guidance does an excellent job of keeping the spacecraft on the trajectory. This is also seen in the altitude (Fig. 4.4), flight path angle (Fig. 4.5), and speed (Fig. 4.6).

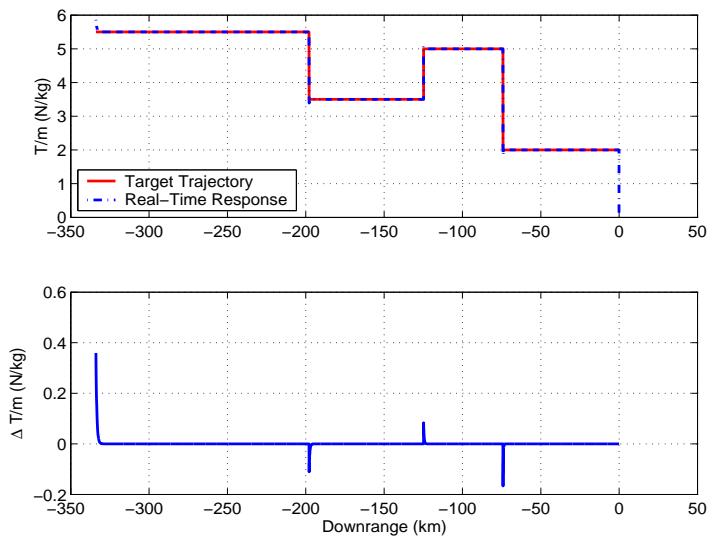


Figure 4.1: Real-Time Guidance Response to Test Trajectory: Commanded Acceleration vs. Downrange

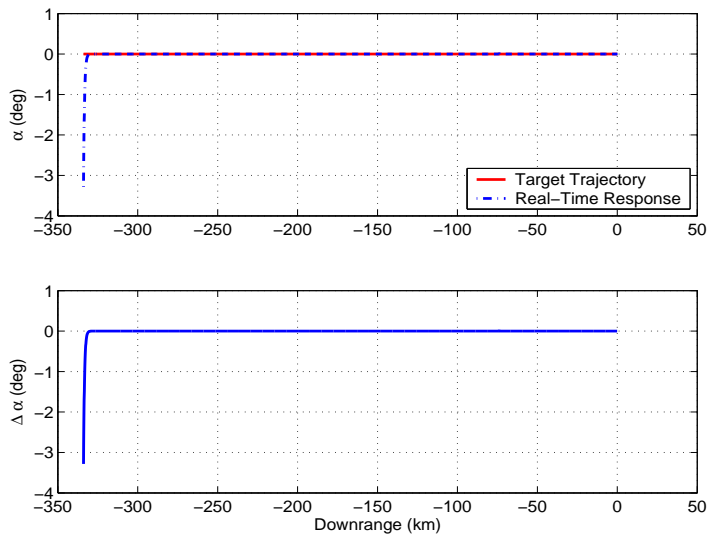


Figure 4.2: Real-Time Guidance Response to Test Trajectory: Commanded Thrust Angle vs. Downrange

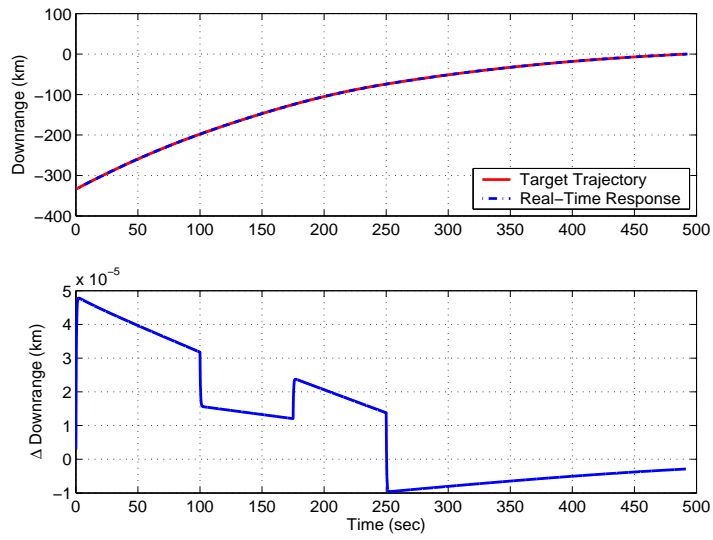


Figure 4.3: Real-Time Guidance Response to Test Trajectory: Downrange vs. Time

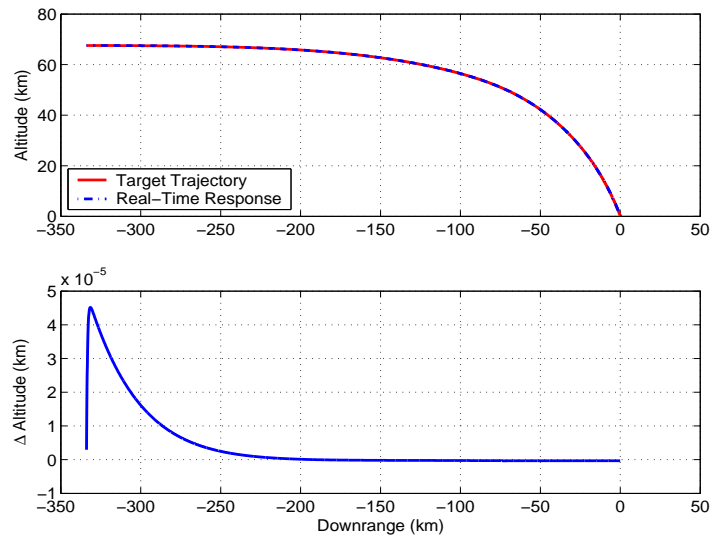


Figure 4.4: Real-Time Guidance Response to Test Trajectory: Altitude vs. Downrange

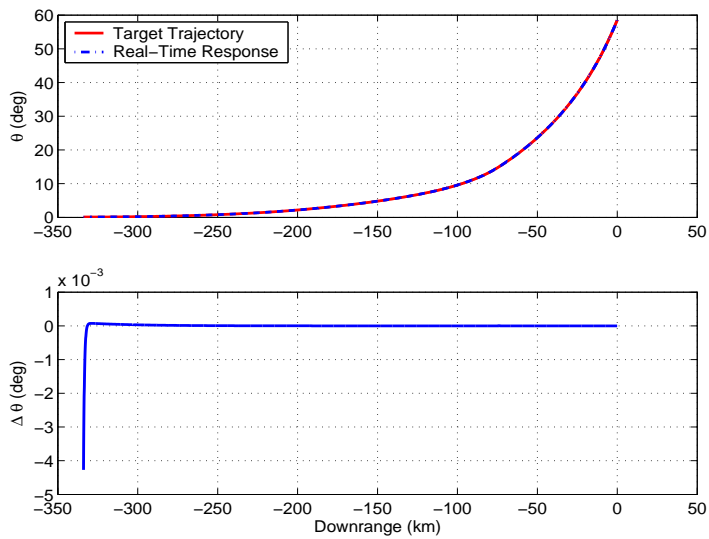


Figure 4.5: Real-Time Guidance Response to Test Trajectory: Flight Path Angle vs. Downrange

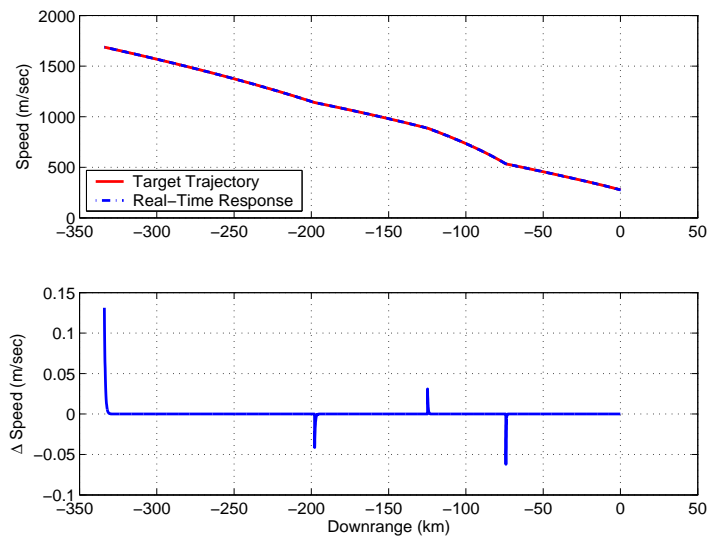


Figure 4.6: Real-Time Guidance Response to Test Trajectory: Speed vs. Downrange

Introducing initialization errors is a well accepted methodology for better understanding the time response of a system. Therefore, a random perturbation off the ideal was added to each state at its initialization while using the same test target trajectory. The response of the system is illustrated in Figs. 4.7 through 4.15. Figures 4.7 through 4.12 illustrate the in-plane response while Figs. 4.13 through 4.15 illustrate the out-of-plane response. These support the earlier statement that the selected gains result in a critically damped response that is acceptable in the absence of specific requirements.

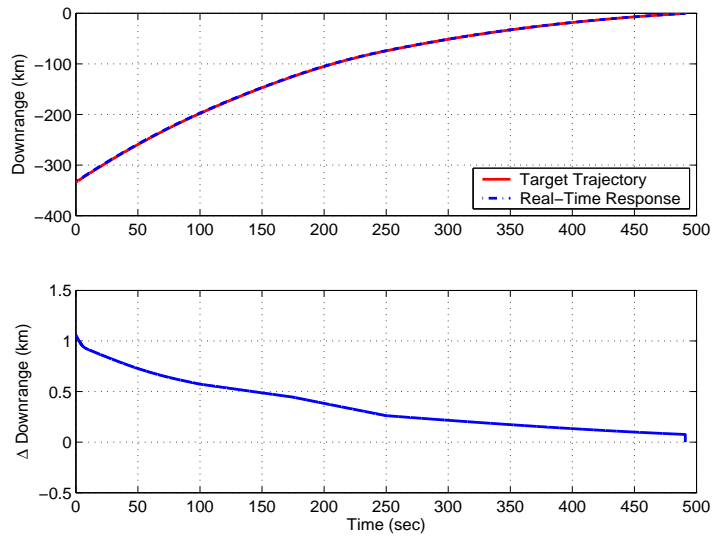


Figure 4.7: Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Downrange vs. Time

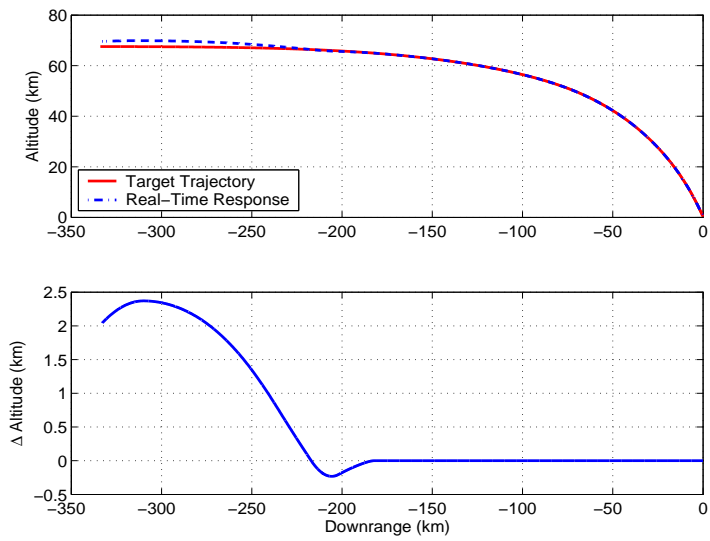


Figure 4.8: Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Altitude vs. Downrange

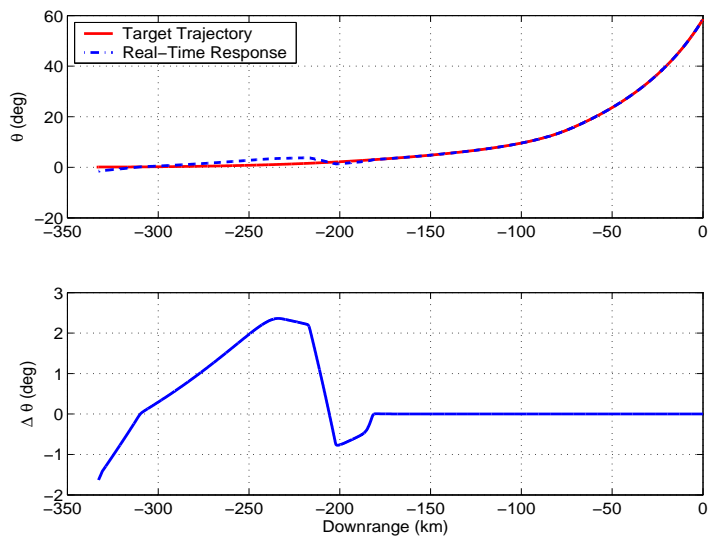


Figure 4.9: Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Flight Path Angle vs. Downrange

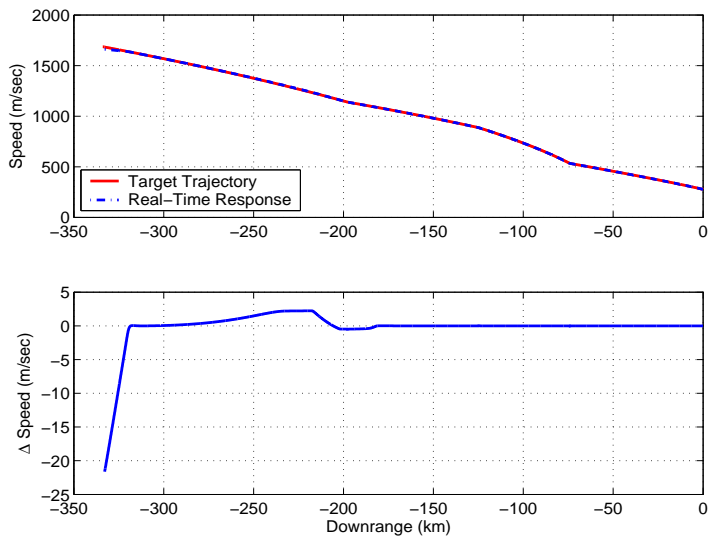


Figure 4.10: Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Speed vs. Downrange

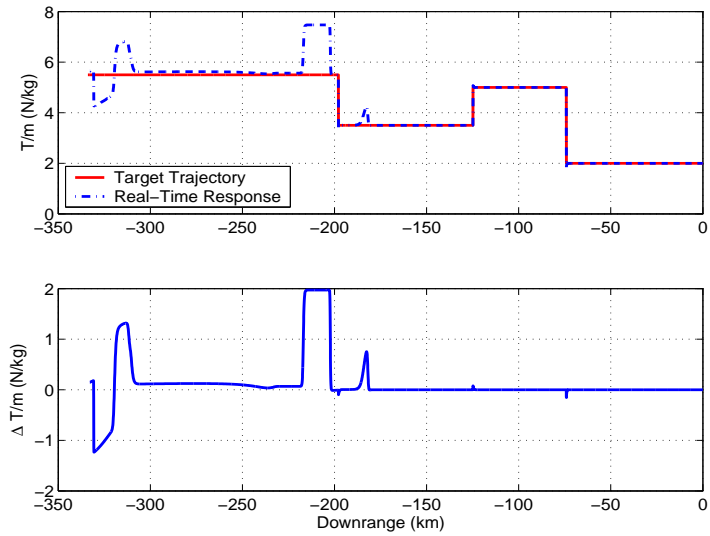


Figure 4.11: Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Commanded Acceleration vs. Downrange

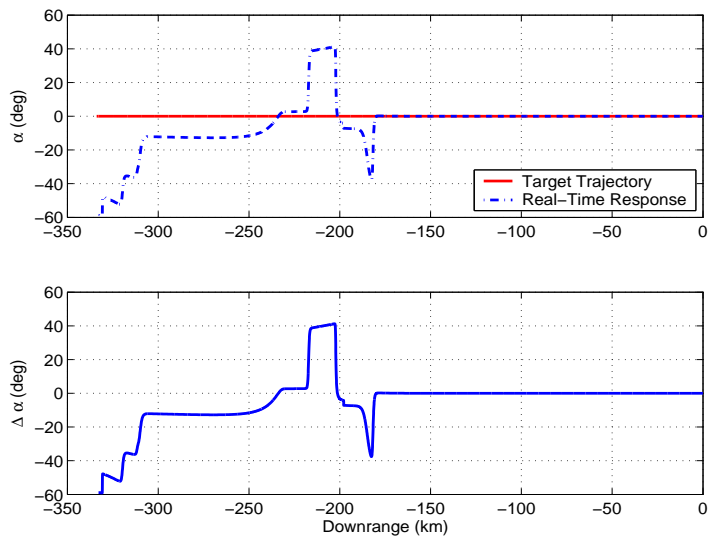


Figure 4.12: Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Commanded Thrust Angle vs. Downrange

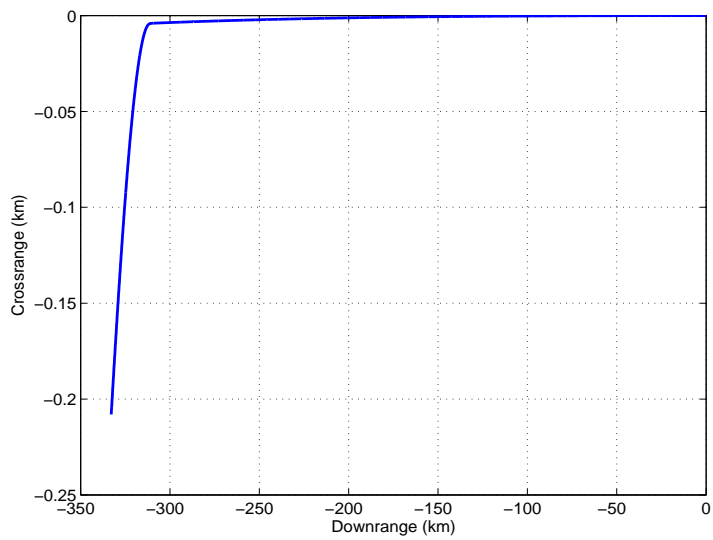


Figure 4.13: Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Crossrange vs. Downrange

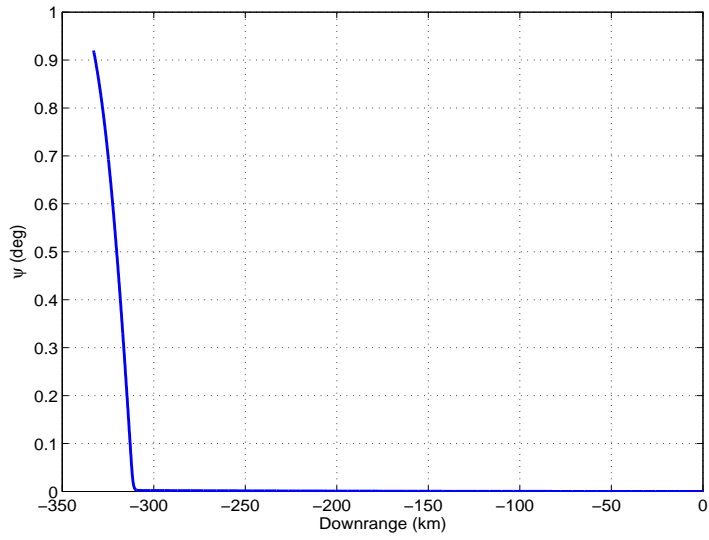


Figure 4.14: Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Crossing Angle vs. Downrange

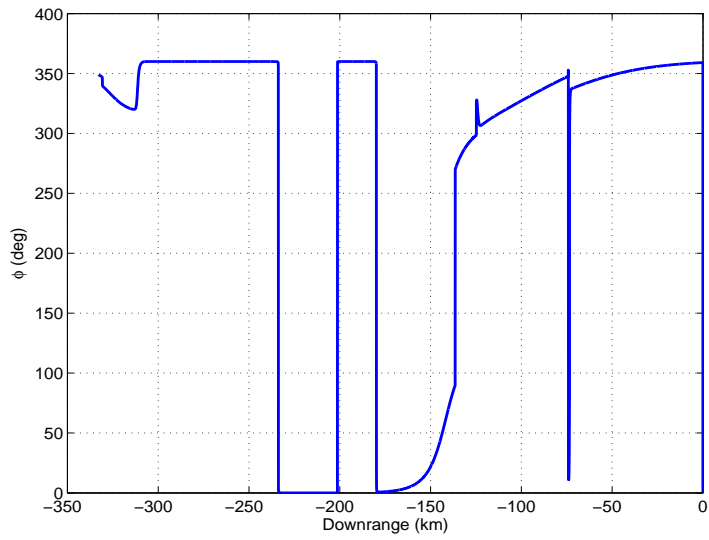


Figure 4.15: Real-Time Guidance Response to Test Trajectory with Initial State Offsets: Commanded Thrust Roll Angle vs. Downrange

4.3 Performance in a Nominal Setting

4.3.1 Response to NASA Target Trajectory

Given the target trajectory developed by existing NASA design methodologies discussed in the previous chapter, the real-time guidance algorithm should readily fly to that trajectory using approximately the same acceleration commands. This was tested to verify that theory. Figure 4.16 shows the good downrange comparison versus time. Similar results are seen in altitude, speed, and flight path angle (Figs. 4.17, 4.18, and 4.19)

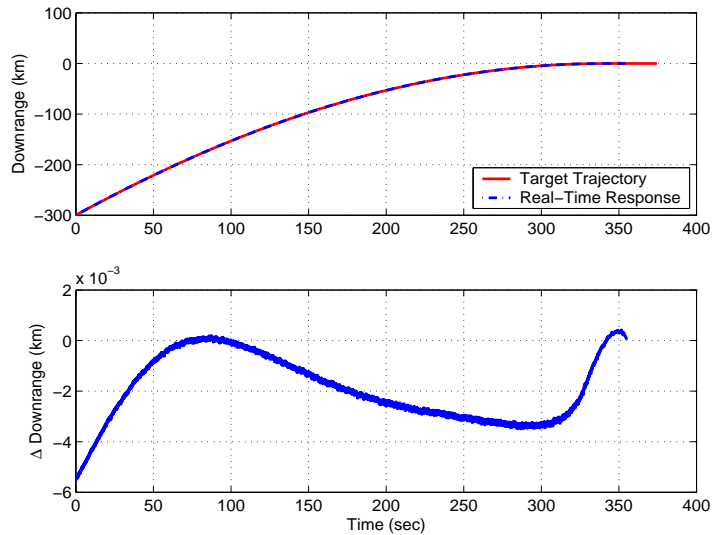


Figure 4.16: Real-Time Guidance Response to NASA Target Trajectory: Downrange vs. Time

Figure 4.20 illustrates the excellent comparison in the accelerations required to maintain the NASA target trajectory. This demonstrates that the guidance algorithm produces acceleration commands consistent with the expected target trajectory acceleration, regardless of how the trajectory was created.

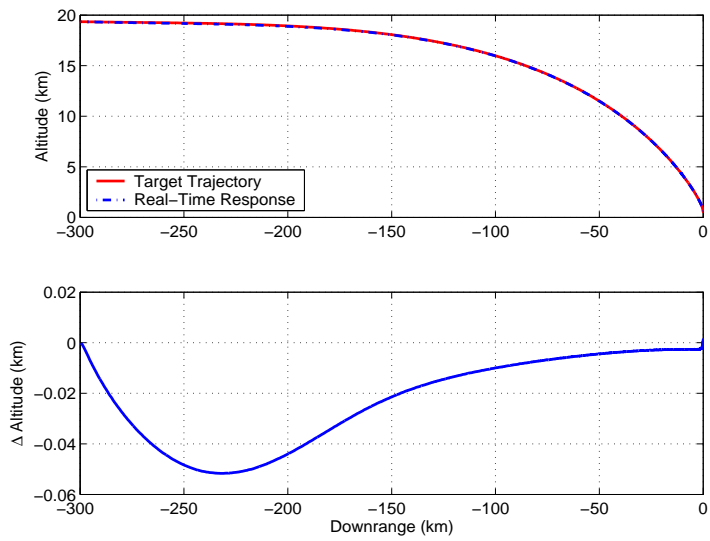


Figure 4.17: Real-Time Guidance Response to NASA Target Trajectory: Altitude vs. Downrange

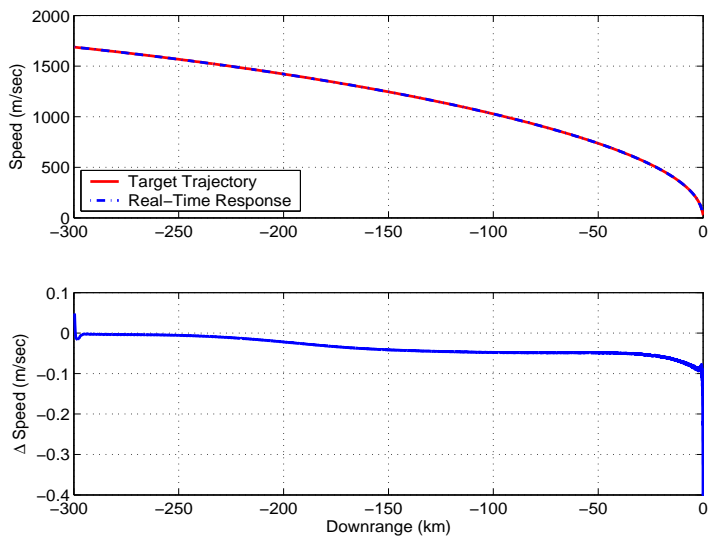


Figure 4.18: Real-Time Guidance Response to NASA Target Trajectory: Speed vs. Downrange

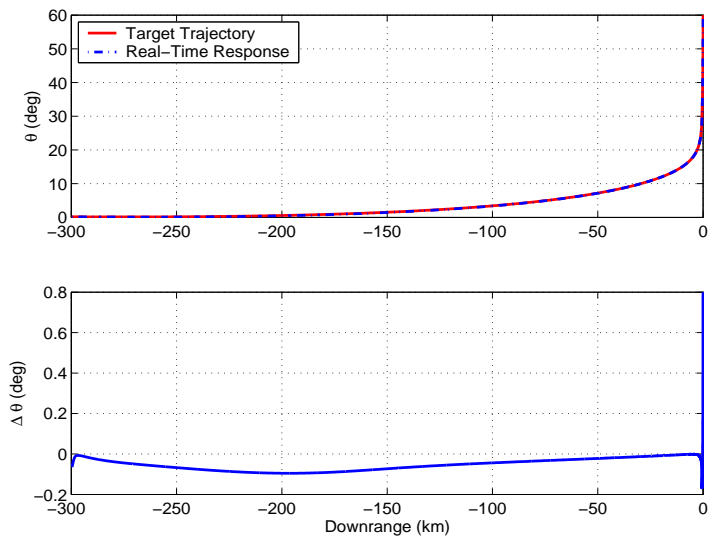


Figure 4.19: Real-Time Guidance Response to NASA Target Trajectory: Flight Path Angle vs. Downrange

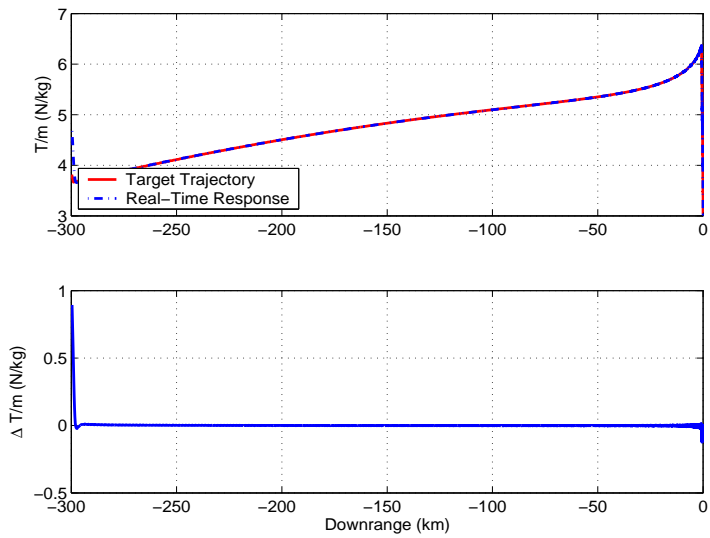


Figure 4.20: Real-Time Guidance Response to NASA Target Trajectory: Acceleration vs. Downrange

4.3.2 Response to Proposed Target Trajectory

In order to complete the nominal testing of the real-time guidance algorithm, the proposed target trajectory discussed in Chapter 3 was introduced. As illustrated in Figs. 4.21 through 4.26, the guidance is capable of flying this trajectory. Errors are the result of introducing realistic gravity and mass time histories as well as the difference between the centrifugal acceleration modeled in the targeting algorithm and the real centrifugal acceleration. Much of the compensation for the centrifugal acceleration can be seen in the thrust angle in Fig. 4.26 with some difference obvious in the thrust magnitude (Fig. 4.25). As expected, the guidance readily accounts for the errors. Note that the small “spikes” in the response most evident in the flight path angle (Fig. 4.23) and the acceleration command (Fig. 4.25) are a result of the change in acceleration command close to the end of the target trajectory (target trajectory’s constant acceleration switches from $4.8 \frac{N}{kg}$ to $1.4 \frac{N}{kg}$).

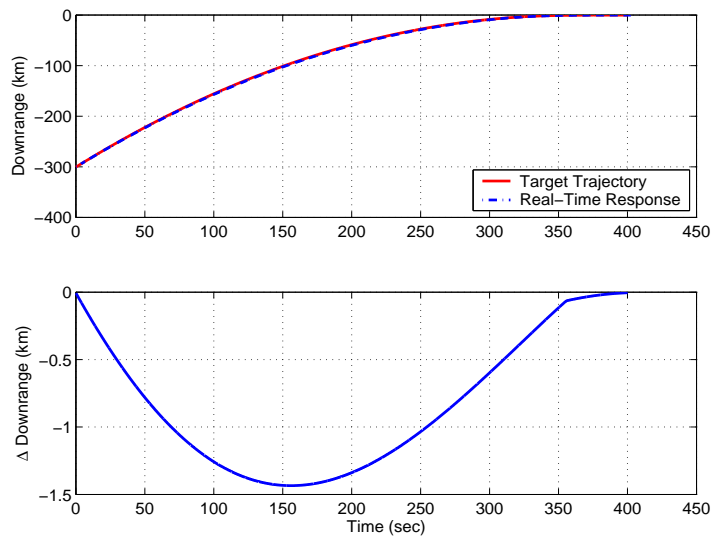


Figure 4.21: Real-Time Guidance Response to Proposed Target Trajectory: Downrange vs. Time

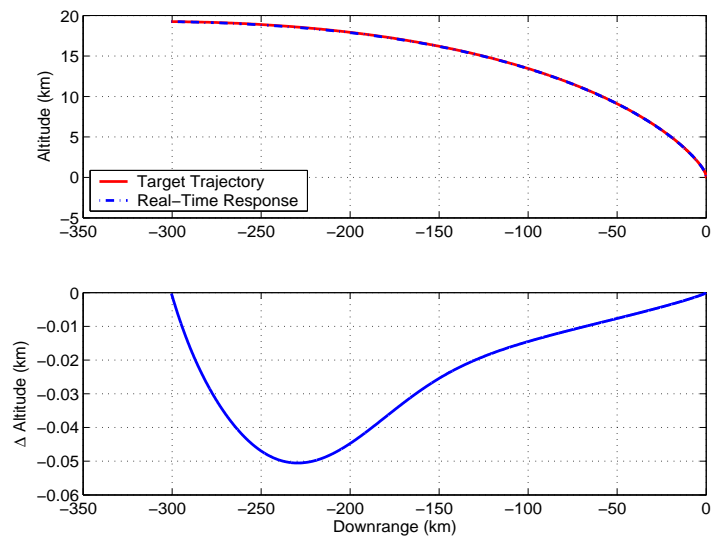


Figure 4.22: Real-Time Guidance Response to Proposed Target Trajectory: Altitude vs. Downrange

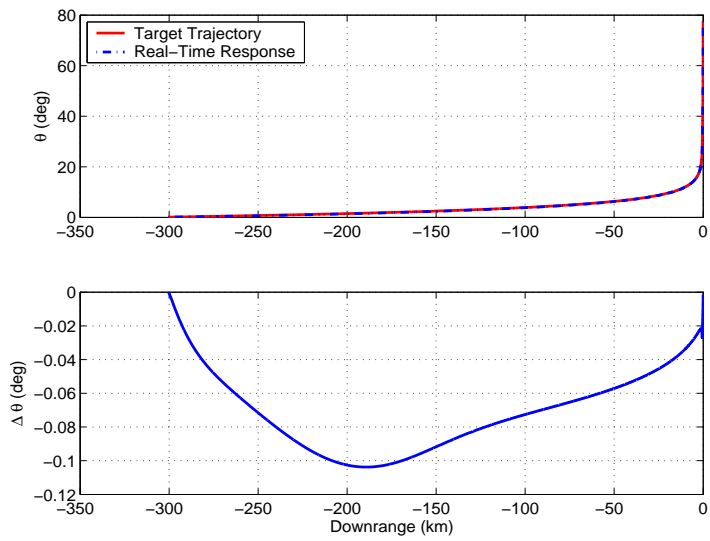


Figure 4.23: Real-Time Guidance Response to Proposed Target Trajectory: Flight Path Angle vs. Downrange

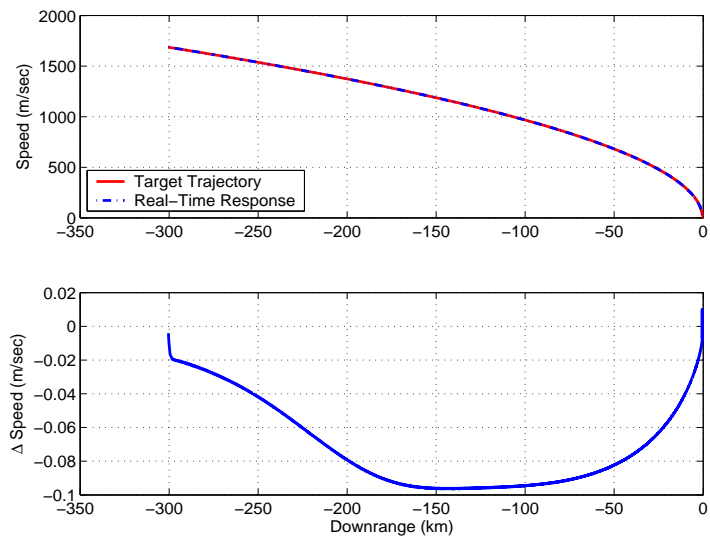


Figure 4.24: Real-Time Guidance Response to Proposed Target Trajectory: Speed vs. Downrange

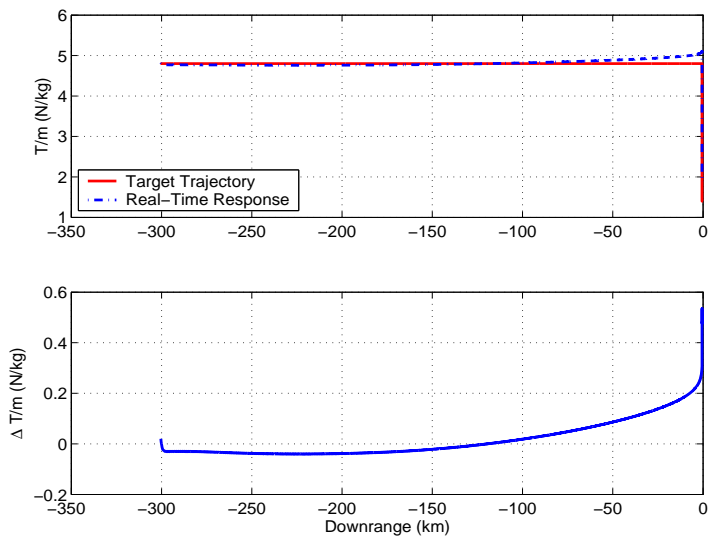


Figure 4.25: Real-Time Guidance Response to Proposed Target Trajectory: Commanded Acceleration vs. Downrange

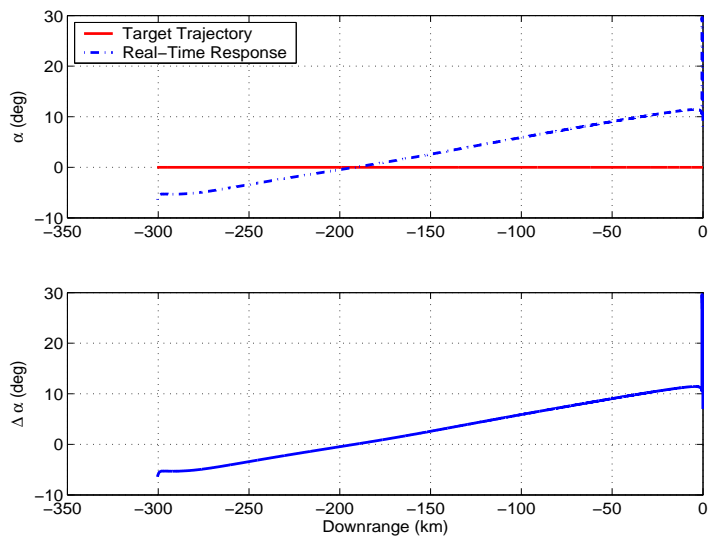


Figure 4.26: Real-Time Guidance Response to Proposed Target Trajectory: Commanded Thrust Angle vs. Downrange

Chapter 5

Algorithm Simulation

5.1 Simulation Description and Results

Any guidance law should work in the ideal sense or it would not be presentable. It must be tested given a realistic setting in order to verify its usability in a real application. Therefore, a simulation has been developed to do just that.

The simulation is a Fortran three degree-of-freedom (3-DOF) set of routines that simulate the translational dynamics of the vehicle. The full translational equations of motion are used as the basis with no approximations for a flat surface. The gravity model is that for an oblate spheroid and is non-constant. Planetary constants used for the moon are given in Table 5.1. Mass is modeled as linearly time-varying as a function of some computed mass flow rate; nominal mass is 31624.0 kg and nominal specific impulse is 459.7 seconds (values from [10]). The real-time guidance algorithm resides in this framework.

The simulation allows for Monte Carlo analysis given perturbations in initial position and velocity away from the initial target trajectory states, mass, mass flow rate, thrust, thrust angle. The errors used to test the algorithm are given in Table 5.2.

The Monte Carlo analysis can be performed in two different manners. The first is to generate a reference trajectory at the beginning based on ex-

Table 5.1: Lunar Planetary Constants

	Value
Equatorial Radius	$1.738x10^6 m$
Polar Radius	$1.735x10^6 m$
J_2	$202.7x10^{-6}$
μ	$4902.799x10^9 \frac{m^3}{sec^2}$
Rotational Rate	$4.2365x10^{-7} \frac{rad}{sec}$

pected states and simply use that for every perturbation of the states. The second, and arguably more desirable, manner is to generate a reference trajectory at the beginning of any simulated run based on the initial states. Both methodologies are presented so that the fact that the guidance algorithm can perform well given either setting can be illustrated. For each case, the perturbations of Table 5.2 are introduced and 100 individual runs are performed.

The first set of data in Figures 5.1 through 5.6 show the performance of the real-time guidance algorithm against the proposed target trajectory of Chapter 3; note that this target trajectory is used for each run in this case. The terminal states are summarized in Table 5.3 and illustrate that the real-time guidance readily forces the vehicle to the desired trajectory and results in what this author deems acceptable performance in the absence of requirements. The control states time histories are shown in Figs. 5.7, 5.8, and 5.9. The thrust acceleration limiting that changes as a function of time can be seen in Fig. 5.7. The thrust angle command shown in Fig. 5.8 stays nicely constrained within about $\pm 70^\circ$ with the exception of some transients at the end caused by the change in target thrust commands. The thrust roll angle in Fig. 5.9, while not particularly enlightening, is included for completeness.

Table 5.2: Monte Carlo Simulation Perturbation Values

Perturbation Source	Mean Value	Standard Deviation
Initial Downrange	0 <i>km</i>	5 <i>km</i>
Initial Altitude	0 <i>km</i>	2 <i>km</i>
Initial Crossrange	0 <i>km</i>	2 <i>km</i>
Initial Speed	0 $\frac{m}{sec}$	100 $\frac{m}{sec}$
Initial Flight Path Angle	0 <i>rad</i>	20 <i>mrad</i>
Initial Crossing Angle	0 <i>rad</i>	20 <i>mrad</i>
Initial Mass	0 <i>kg</i>	2000 <i>kg</i>
Mass Flow Rate	0 %	5 %
Thrust	0 %	5 %
Thrust Angle (α) Bias	0 <i>rad</i>	30 <i>mrad</i>
Roll Angle (ϕ) Bias	0 <i>rad</i>	50 <i>mrad</i>

Table 5.3: Monte Carlo Simulation Final State Error Statistics for Case Using Same Target Trajectory for Each Run

Error	Target Value	Mean Value	Standard Deviation
Time	370 <i>sec</i>	402.31 <i>sec</i>	3.92 <i>sec</i>
Downrange	0 <i>m</i>	-0.02 <i>m</i>	0.11 <i>m</i>
Crossrange	0 <i>m</i>	0.01 <i>m</i>	0.45 <i>m</i>
Speed	8 $\frac{m}{sec}$	8.04 $\frac{m}{sec}$	0.08 $\frac{m}{sec}$
Flight Path Angle	77.4°	77.42°	0.07°
Crossing Angle	0°	-0.01°	0.21°

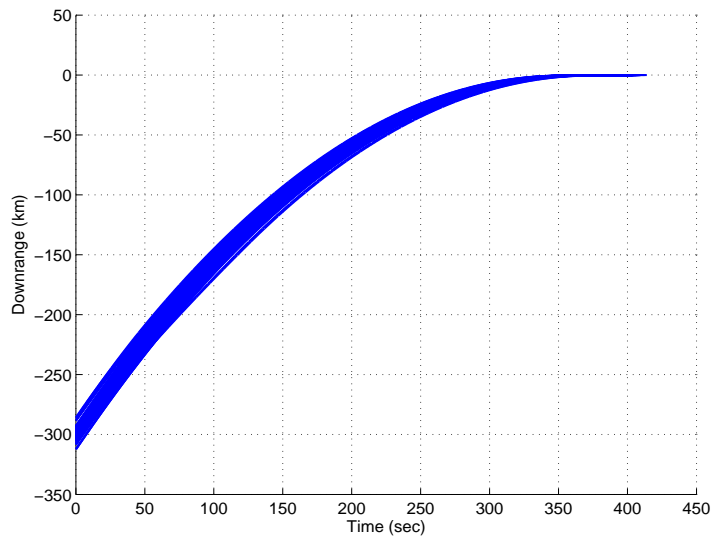


Figure 5.1: Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Downrange vs. Time

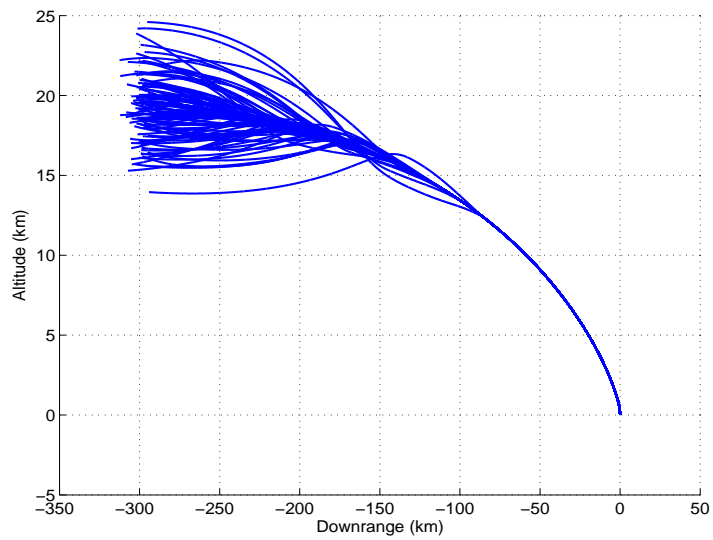


Figure 5.2: Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Altitude vs. Downrange

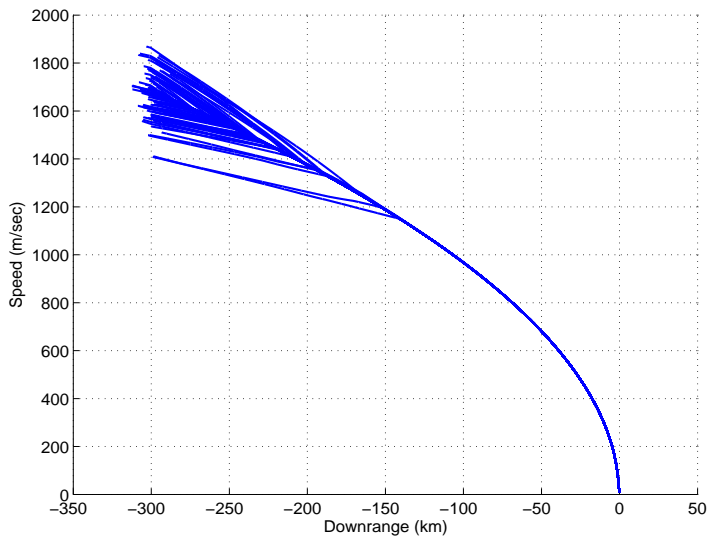


Figure 5.3: Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Speed vs. Downrange

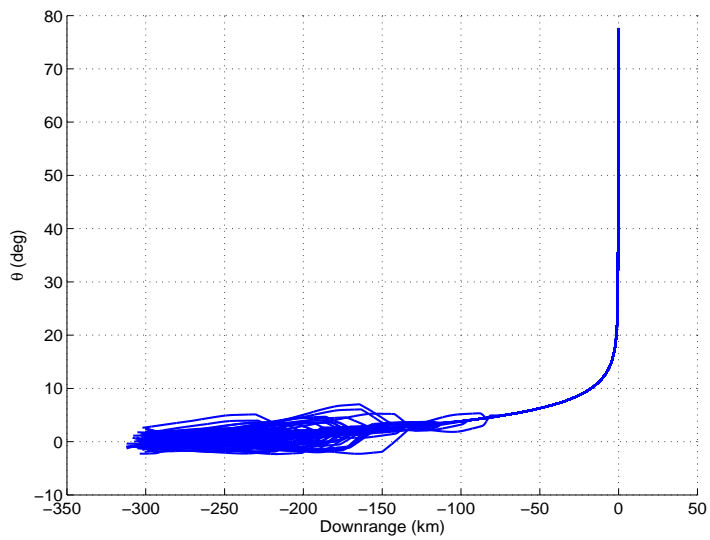


Figure 5.4: Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Flight Path Angle vs. Downrange

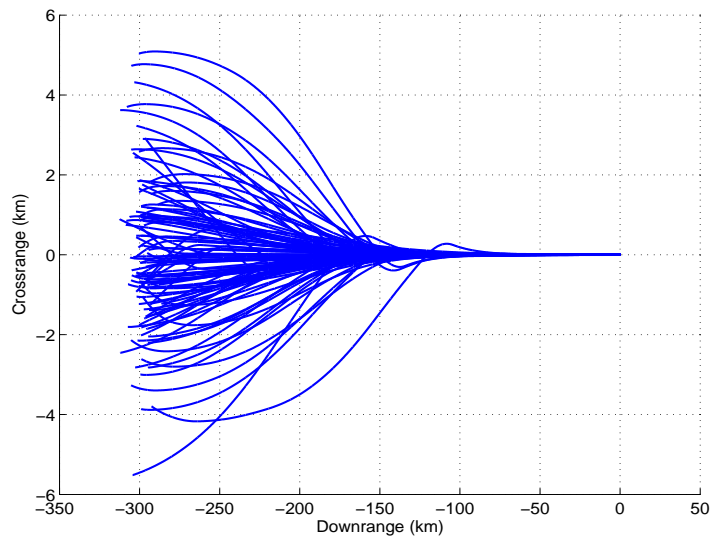


Figure 5.5: Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Crossrange vs. Downrange

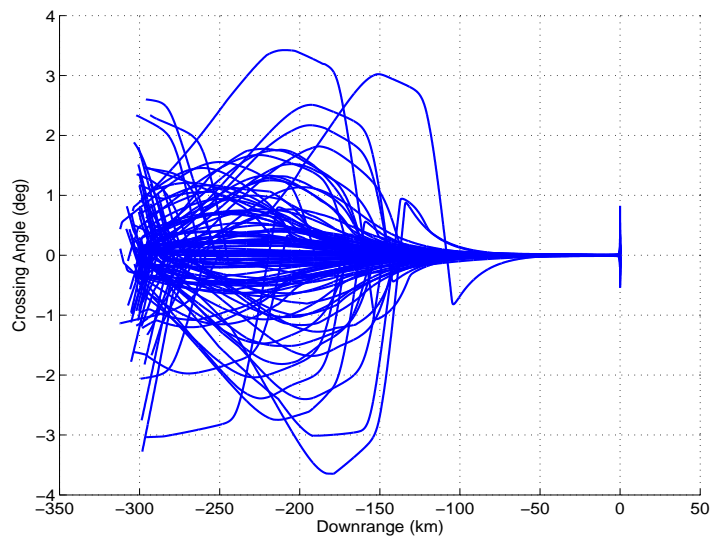


Figure 5.6: Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Crossing Angle vs. Downrange

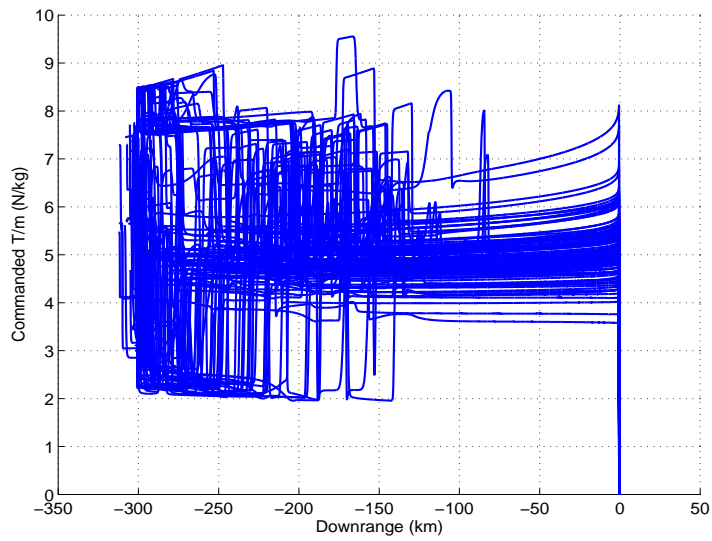


Figure 5.7: Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Commanded Acceleration vs. Downrange

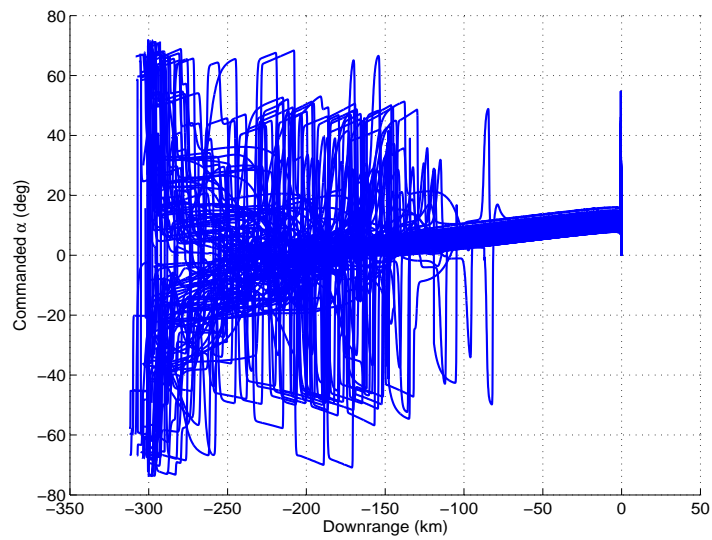


Figure 5.8: Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Commanded Thrust Angle vs. Downrange

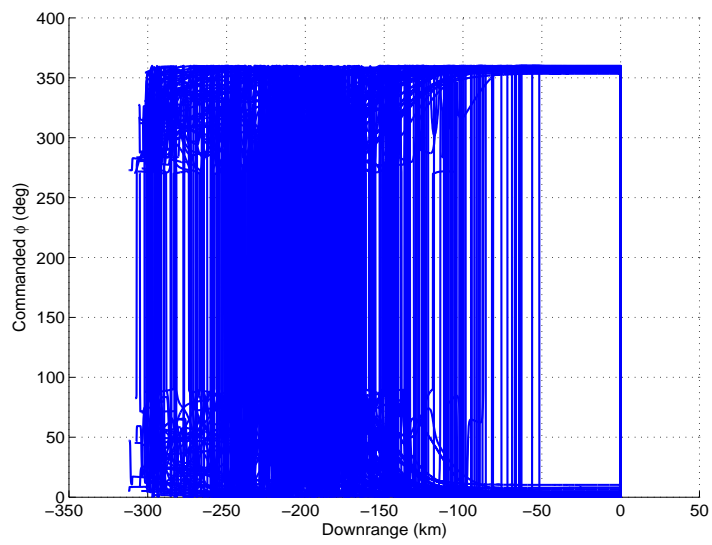


Figure 5.9: Monte Carlo Simulation Results With No Per Run Target Trajectory Change: Commanded Thrust Roll Angle vs. Downrange

The second manner of performing the guidance is to have the real-time guidance algorithm respond against target trajectories that are updated with each set of initial conditions. The algorithm for this the same that is explained in Chapter 3. The two segment target trajectory is used with equations 3.21, 3.19, 3.22, and 3.23 creating the altitude versus downrange solution space. For this analysis, the target trajectory that matched the altitude span closest to the current altitude was selected; this means the vehicle should not have to accelerate upward or dive downward to “catch” the reference while the downrange will eventually “catch up” with the reference with little impact on the response in the interim. Constraints were added that $\theta_0 \leq \theta_1 \leq \theta_2$ and that $v_2 \leq v_1 \leq v_0$ to keep the trajectories in a more desirable space. An additional requirement was levied that the initial flight path angle, θ_0 used to define the target trajectory was positive so as to not create target trajectories that initially increase in altitude; inducing additional upward acceleration was deemed undesirable.

Figure 5.10 is the same as Fig. 3.9; it shows the trajectory space available for the nominal states that are the basis for the Monte Carlo runs performed in this section. This is essentially the trajectory space available for each of the runs in this section (small variations will exist in the individual trajectory spaces generated for each run because the initial speed and flight path angle will be slightly different than the values used to generate the shown trajectory space). The blue box shows the area that is tested in the Monte Carlo runs and begins to give a feel for the capability of the algorithm given the vehicle parameters already described. For example, given the trajectory selection algorithm based on satisfying the altitude constraint, a vehicle at 20 km in altitude will select a trajectory at approximately 250 km in downrange,

regardless of the actual downrange which could be as large as 300 km.

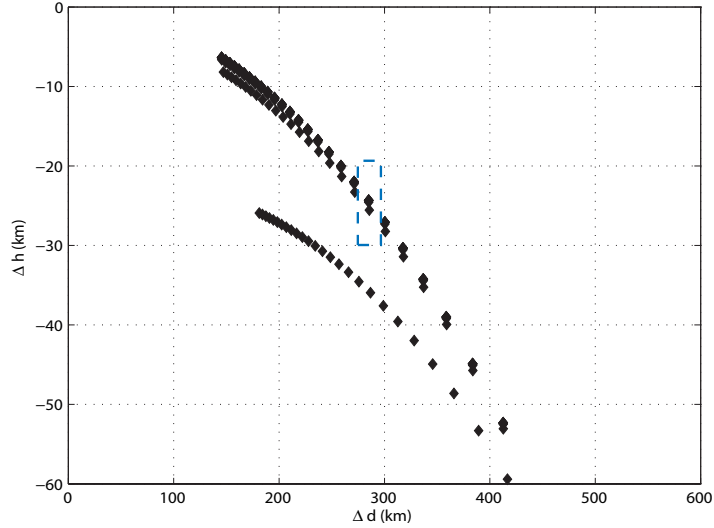


Figure 5.10: Sample Trajectory Space With Monte Carlo Limits

The guidance response for the second targeting methodology is shown in Figures 5.11 through 5.19. The terminal states are summarized in Table 5.4 and again illustrate that the real-time guidance performs well. Updating the target trajectory does not significantly change the performance for the test space shown here, but the definite advantages exist that the target trajectory is more “hand tailored” to the current situation.

Note the single case in all of the plots that appears to be an outlier. This is simply a situation where the target trajectory selected is different in character from all of the other runs; the targeting and real-time guidance algorithms still perform well with no problems. This run does illustrate the need for the user of this algorithm to think clearly about all states that should be constrained, because this run does have a long final trajectory time compared

to the other runs (about 100 seconds longer). If time is a major constraint (which it could easily be due to fuel constraints), then balancing the different trajectory options is important and would have to be considered in a specific algorithm.

Table 5.4: Monte Carlo Simulation Final State Error Statistics for Case Using Run Specific Target Trajectory for Each Run

Error	Target Value	Mean Value	Standard Deviation
Time	varies	400.59 <i>sec</i>	29.48 <i>sec</i>
Downrange	0 <i>m</i>	-0.02 <i>m</i>	0.05 <i>m</i>
Crossrange	0 <i>m</i>	0.01 <i>m</i>	0.32 <i>m</i>
Speed	8 $\frac{m}{sec}$	8.04 $\frac{m}{sec}$	0.38 $\frac{m}{sec}$
Flight Path Angle	89°	89.01°	0.02°
Crossing Angle	0°	-0.16°	2.23°

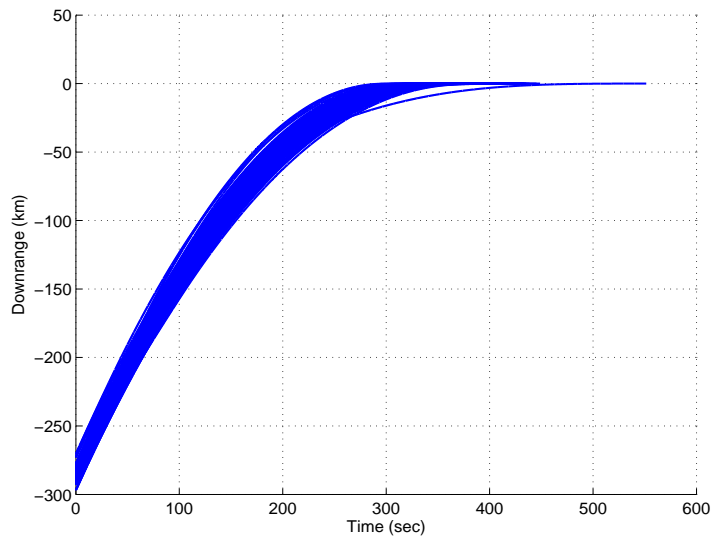


Figure 5.11: Monte Carlo Simulation Results With Run Specific Target Trajectory: Downrange vs. Time

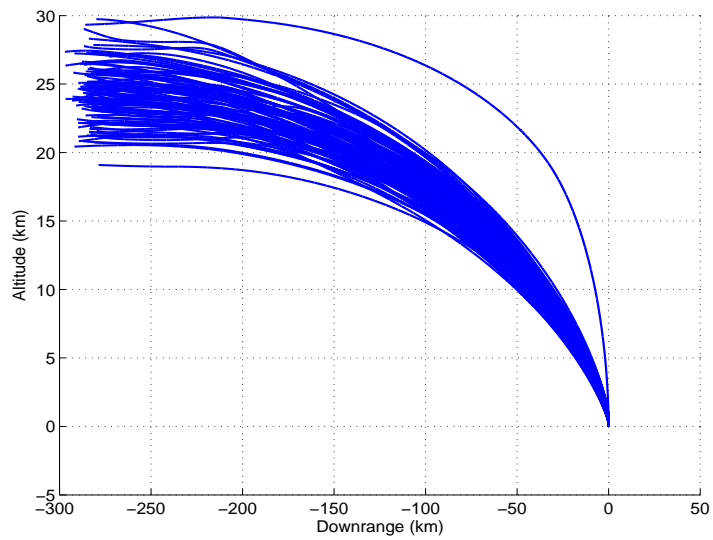


Figure 5.12: Monte Carlo Simulation Results With Run Specific Target Trajectory: Altitude vs. Downrange

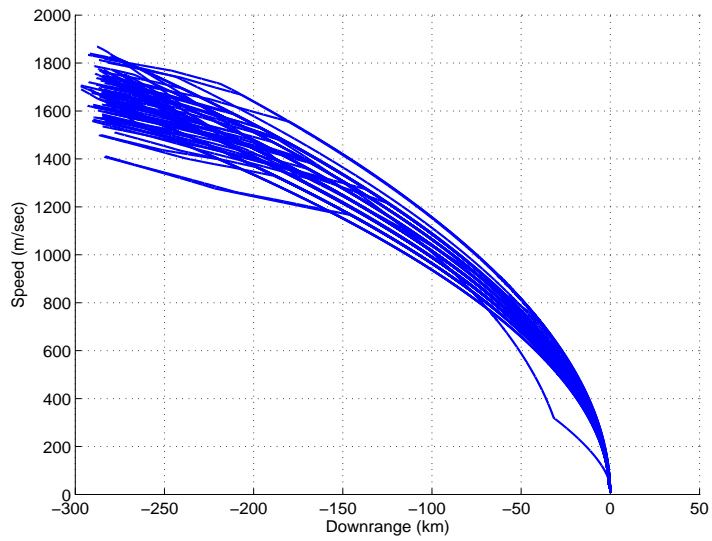


Figure 5.13: Monte Carlo Simulation Results With Run Specific Target Trajectory: Speed vs. Downrange

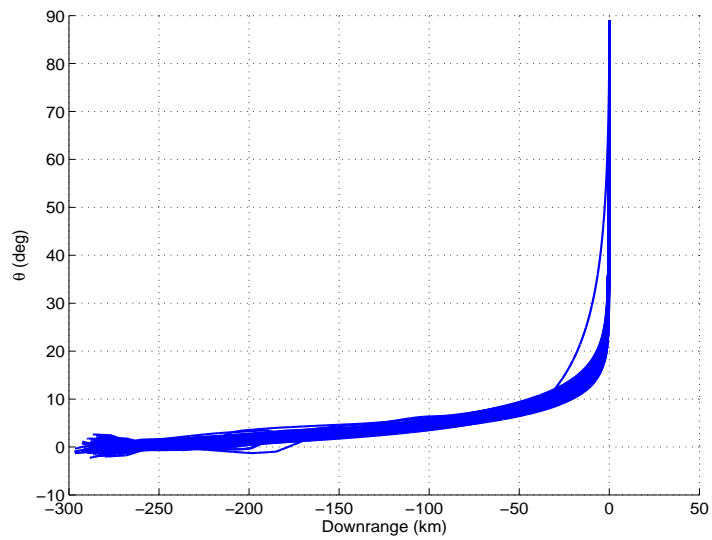


Figure 5.14: Monte Carlo Simulation Results With Run Specific Target Trajectory: Flight Path Angle vs. Downrange

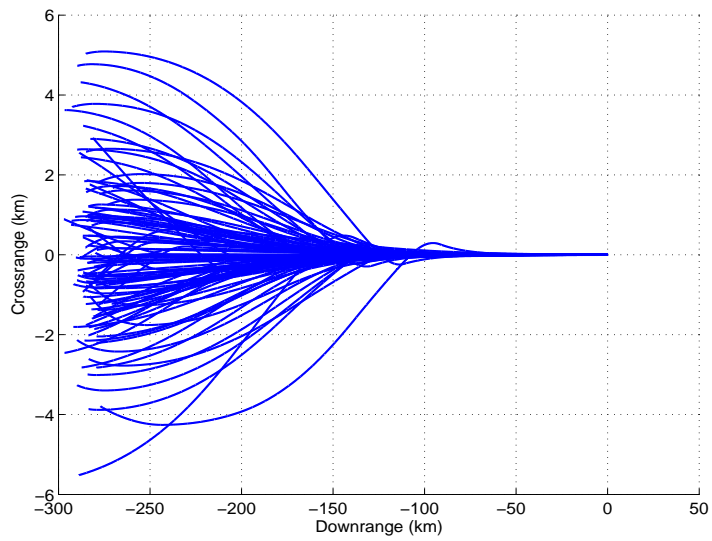


Figure 5.15: Monte Carlo Simulation Results With Run Specific Target Trajectory: Crossrange vs. Downrange

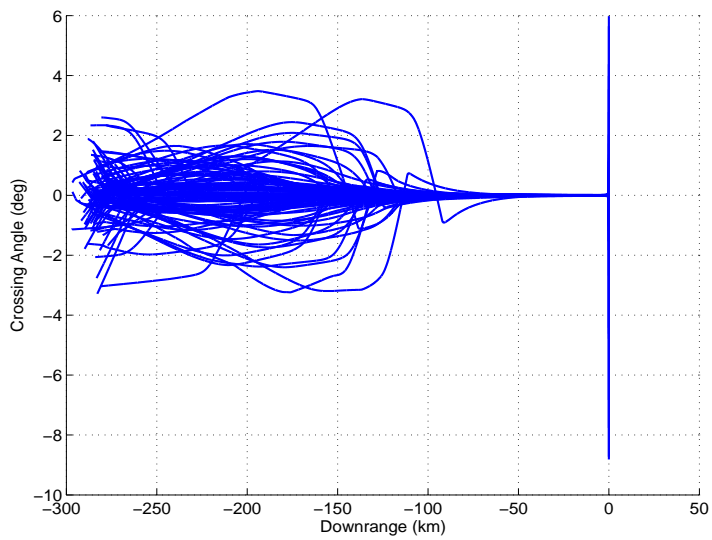


Figure 5.16: Monte Carlo Simulation Results With Run Specific Target Trajectory: Crossing Angle vs. Downrange

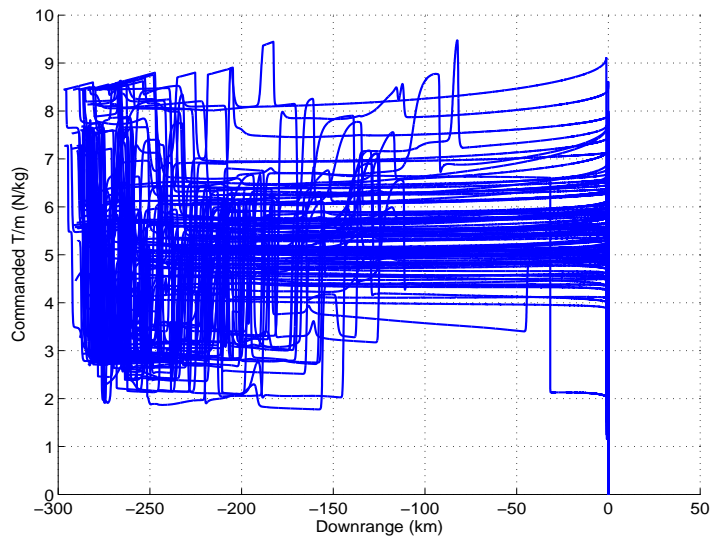


Figure 5.17: Monte Carlo Simulation Results With Run Specific Target Trajectory: Commanded Acceleration vs. Downrange

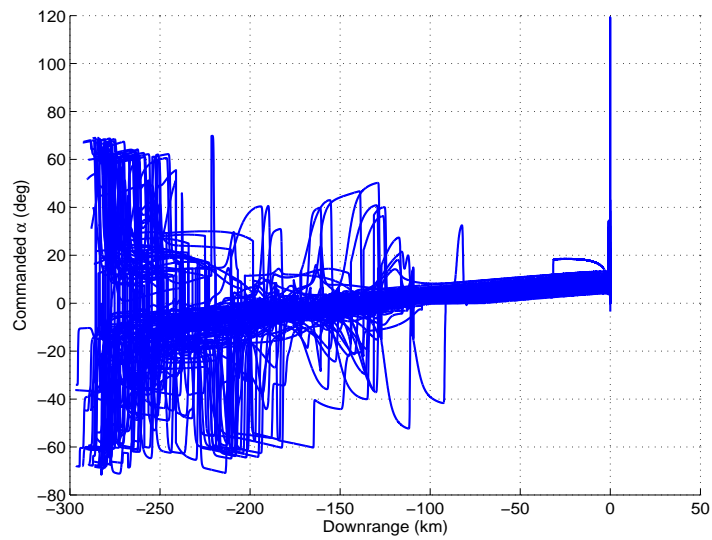


Figure 5.18: Monte Carlo Simulation Results With Run Specific Target Trajectory: Commanded Thrust Angle vs. Downrange

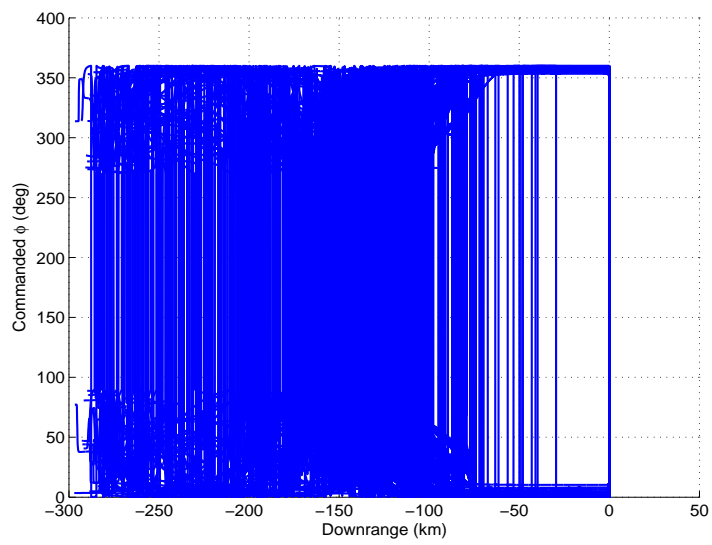


Figure 5.19: Monte Carlo Simulation Results With Run Specific Target Trajectory: Commanded Thrust Roll Angle vs. Downrange

5.2 Retargeting

At some point in a trajectory, the decision may be made that the current landing site is no longer desirable. This would require that the landing site be relocated and all of the guidance states translated to be with respect to that new landing site; a resultant change in kinematic states would thereby have to be handled by the real-time guidance. Additionally, a new target trajectory may be necessary, which will again require a response by the real-time guidance. These issues will be addressed in the following discussion.

The retargeting methodology chosen herein was to specify a new landing site location in PCPF coordinates at some desired time (another option might be to specify a location relative to the original location, but this is a simple matter of performing the appropriate geometry). This new landing site PCPF vector is converted to latitude and longitude as discussed in Section 2.2.2. From there, a new target trajectory bearing must be computed.

Recall that the trajectory bearing is a user specified input in the original formulation discussed in Chapter 2. However, this bearing will likely not work once the landing site location is moved. Therefore, the new bearing for the target trajectory is computed assuming that the shortest distance between the current vehicle location and the new desired landing location defines the current downrange. This means that the new bearing is computed using Eqs. 2.36, 2.37, and 2.39:

$$\begin{aligned}\cos \beta_{new} &= \frac{\sin \varphi' - \sin \varphi'_t \cos \Lambda}{\cos \varphi'_t \sin \Lambda} \\ \sin \beta_{new} &= \frac{\sin \Delta \lambda \cos \varphi'}{\sin \Lambda} \\ \cos \Lambda &= \sin \varphi'_t \sin \varphi' + \cos \varphi'_t \cos \varphi' \cos \Delta \lambda\end{aligned}$$

With this new bearing, the current states can be computed as before (described in Section 2.2) to get the appropriate altitude, downrange, crossrange, speed, flight path angle, and crossing angle.

With the vehicle states now defined with respect to the new landing site, the new target trajectory is defined by using those states as the basis. The altitude, downrange, flight path angle, and speed are fed into the algorithm described in Chapter 3 with the same constraints discussed in Section 5.1 ($\theta_0 \leq \theta_1 \leq \theta_2$, $v_2 \leq v_1 \leq v_0$, $\theta_0 > 0$). As with the initial targeting algorithm described in Section 5.1, matching altitude was used as the basis for selecting the target trajectory. With the new bearing defined, the vehicle states adjusted, and the new target trajectory selected, the real-time guidance algorithm continues as before with no changes.

Two examples of retargeting are presented. The first example moves the target location 1° North and 1° East and the second moves the target location 1° South and 1° East. These illustrate the impact of moving the target location both further and closer to the current vehicle location. In both cases, the retargeting occurs 50 seconds into the event; at this point, the vehicle is at approximately 23.3 km in altitude at a speed of approximately $1438 \frac{m}{sec}$ with a flight path angle of approximately 1.5° . Because these states are the same for both cases, the target trajectory selected was the same for both. It was created with the following parameters: $\theta_0 = 1.5^\circ$, $\theta_f = 89^\circ$, $v_0 = 1438 \frac{m}{sec}$, $v_f = 8 \frac{m}{sec}$, $\frac{T}{m_1} = 5.1 \frac{N}{kg}$, and $\frac{T}{m_2} = 1.6 \frac{N}{kg}$. It spans an altitude of 23.2 km and a downrange of -208.1 km.

Figure 5.20 illustrates the vehicle path in latitude and longitude for the example that retargets to the Northeast; both the original and retargeted landing sites can be seen. Figure 5.21 illustrates the downrange; note that

the downrange increases again at the point of the retargeting. This effect can be seen in Figs. 5.22 through 5.29 as the downrange “backs up” in all of the plots at the retargeting point. In all of these figures, both the original target trajectory and the retargeted target trajectory are shown. Notice that this new trajectory is hardly discernable from the original target trajectory indicating that a new target trajectory is not necessary if the vehicle altitude, speed, and flight path angle are close to the target altitude in the existing trajectory. Figures 5.22 and 5.23 illustrate the impact of retargeting on altitude and crossrange and Figs. 5.24, 5.25, and 5.26 illustrates the impact on speed, flight path angle, and crossing angle. The control states are shown in Figs. 5.27, 5.28, and 5.29.

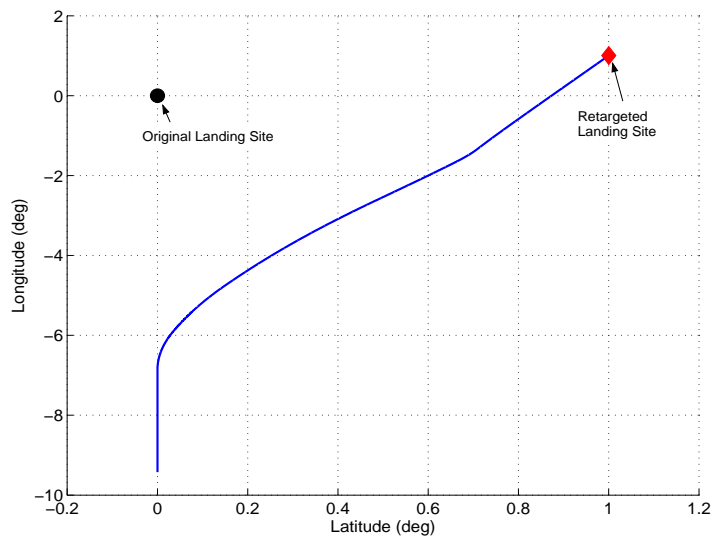


Figure 5.20: Retargeting (Northeast Example): Latitude vs. Longitude

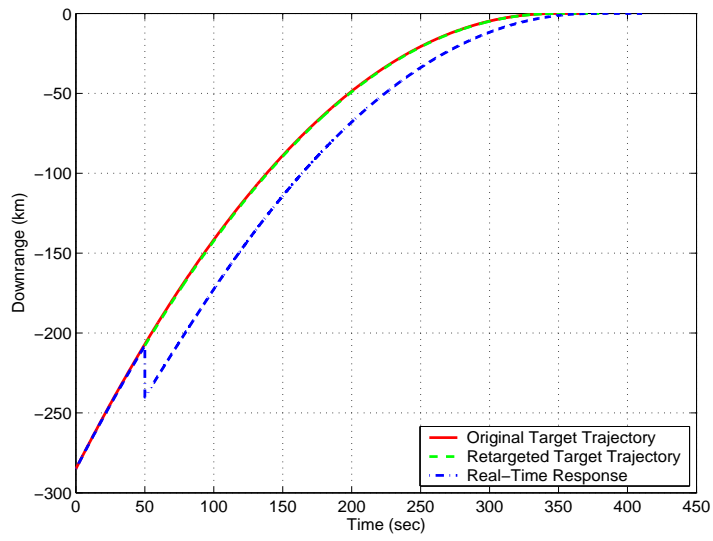


Figure 5.21: Retargeting (Northeast Example): Downrange vs. Time

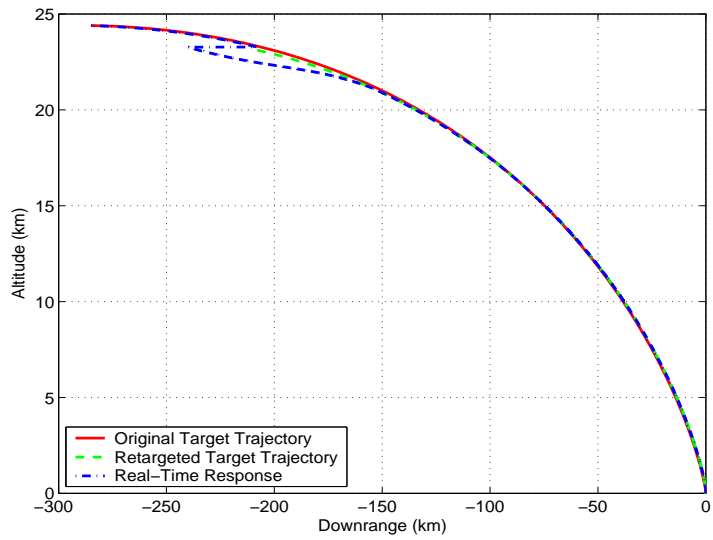


Figure 5.22: Retargeting (Northeast Example): Altitude vs. Time

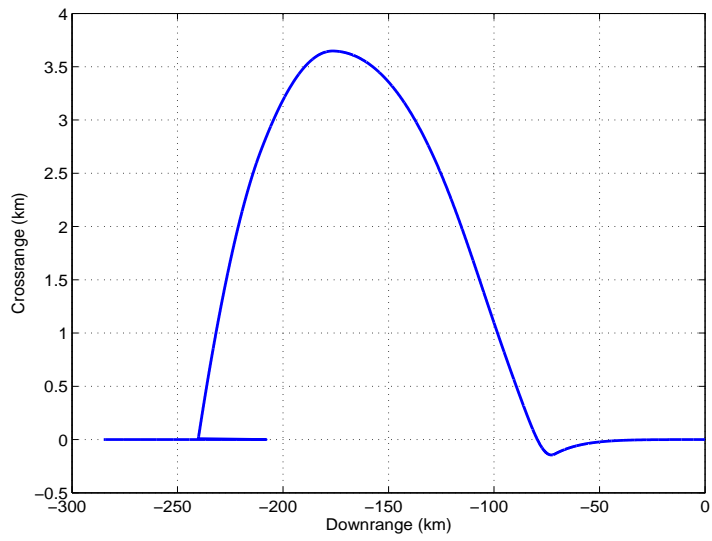


Figure 5.23: Retargeting (Northeast Example): Crossrange vs. Time

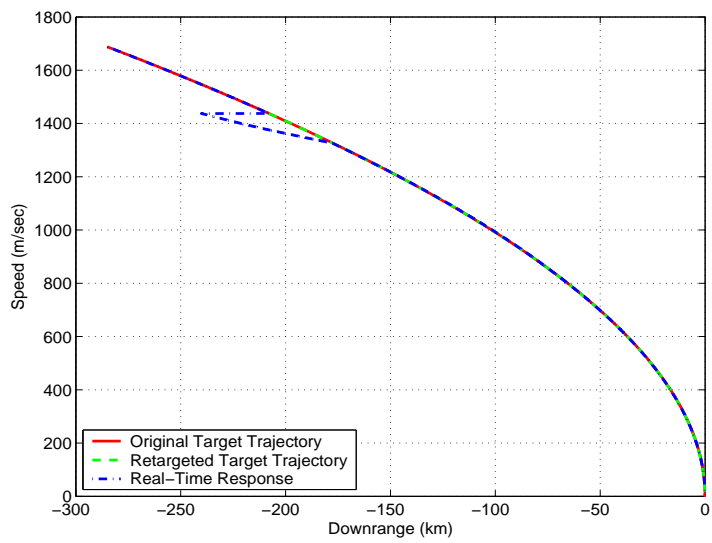


Figure 5.24: Retargeting (Northeast Example): Speed vs. Time

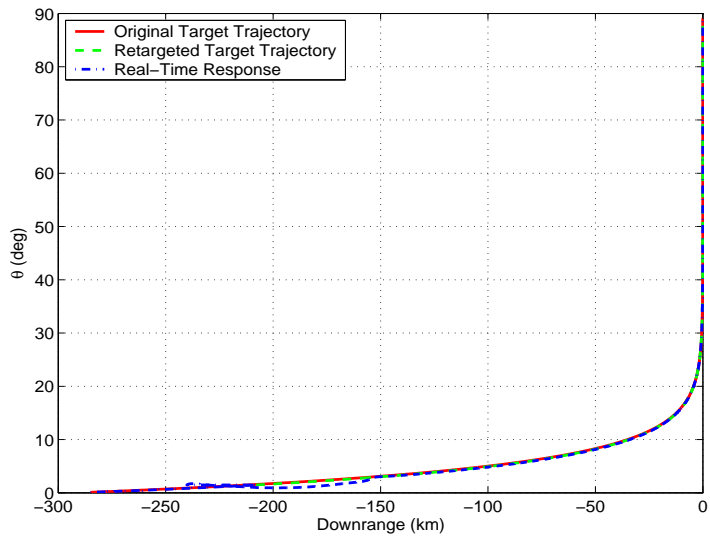


Figure 5.25: Retargeting (Northeast Example): Flight Path Angle vs. Time

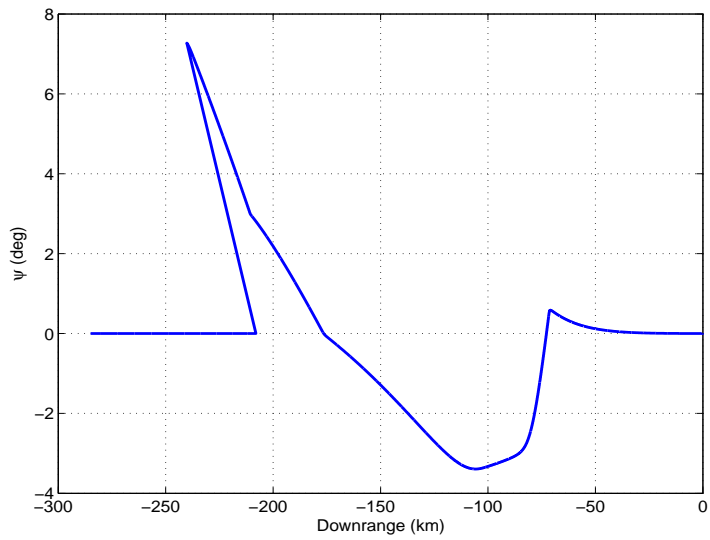


Figure 5.26: Retargeting (Northeast Example): Crossing Angle vs. Time

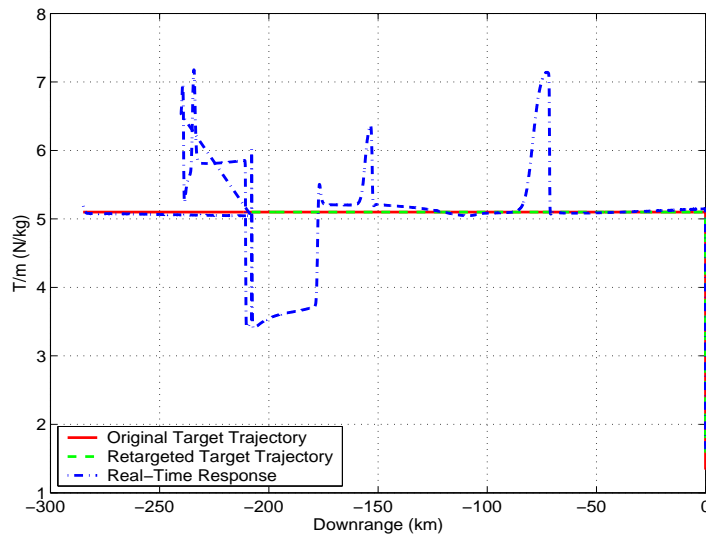


Figure 5.27: Retargeting (Northeast Example): Thrust Acceleration vs. Time

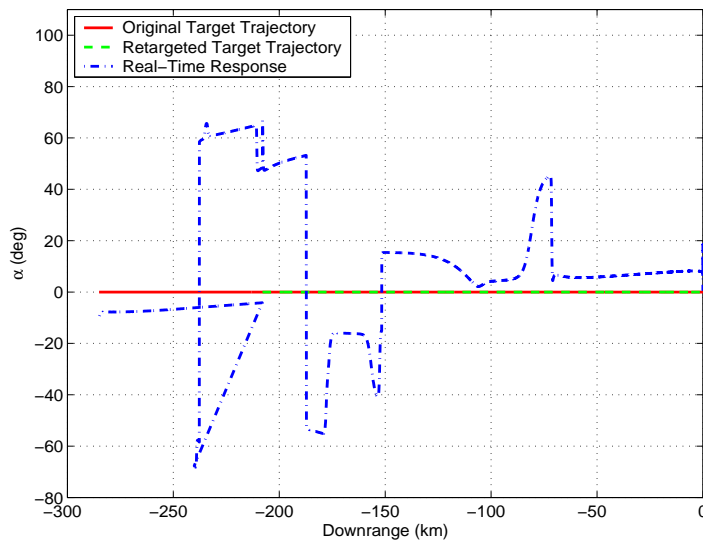


Figure 5.28: Retargeting (Northeast Example): Thrust Angle vs. Time

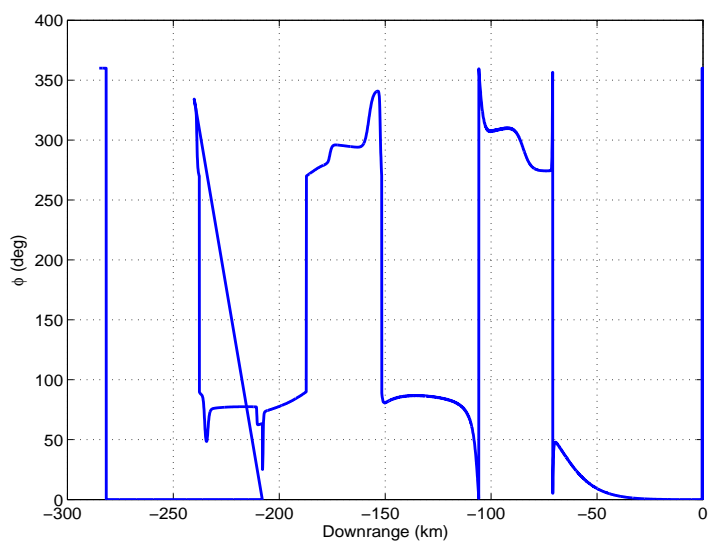


Figure 5.29: Retargeting (Northeast Example): Thrust Roll Angle vs. Time

The second retargeting example is shown in Figs. 5.30 through 5.39. Figure 5.30 illustrates the vehicle path in latitude and longitude for the example that retargets to the Southeast. Figures 5.31, 5.32 and 5.33 illustrate the impact of retargeting on downrange, altitude, and crossrange and Figs. 5.34, 5.35, and 5.36 illustrates the impact on speed, flight path angle, and crossing angle. The control states are shown in Figs. 5.37, 5.38, and 5.39.

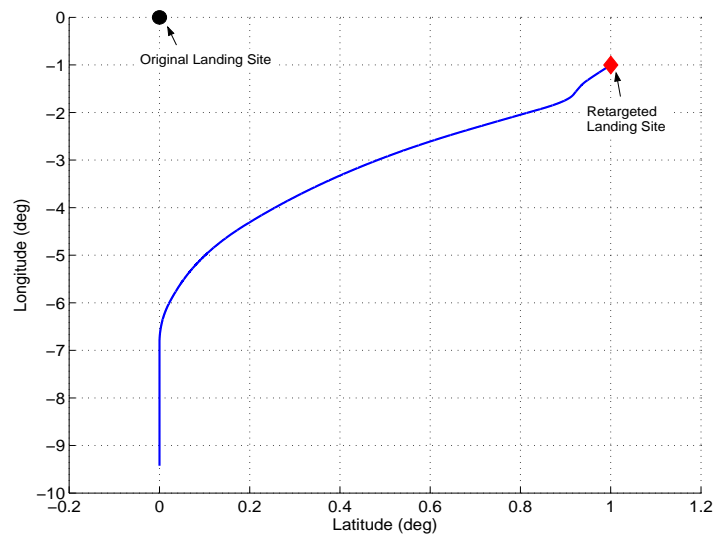


Figure 5.30: Retargeting (Southeast Example): Latitude vs. Longitude

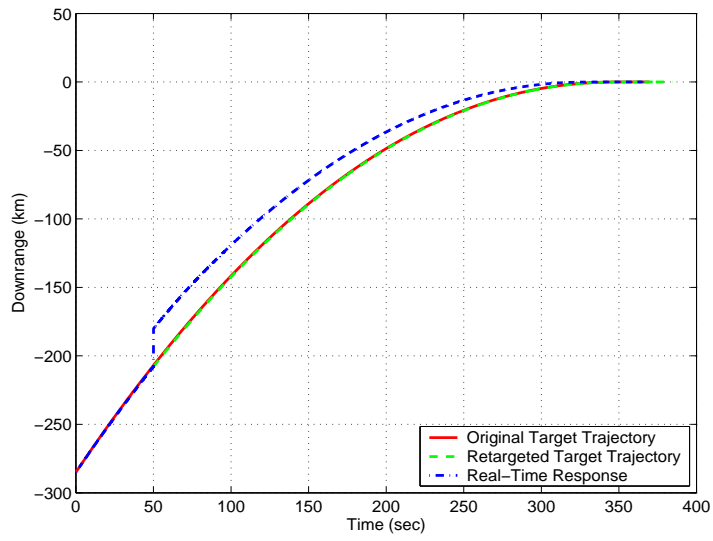


Figure 5.31: Retargeting (Southeast Example): Downrange vs. Time

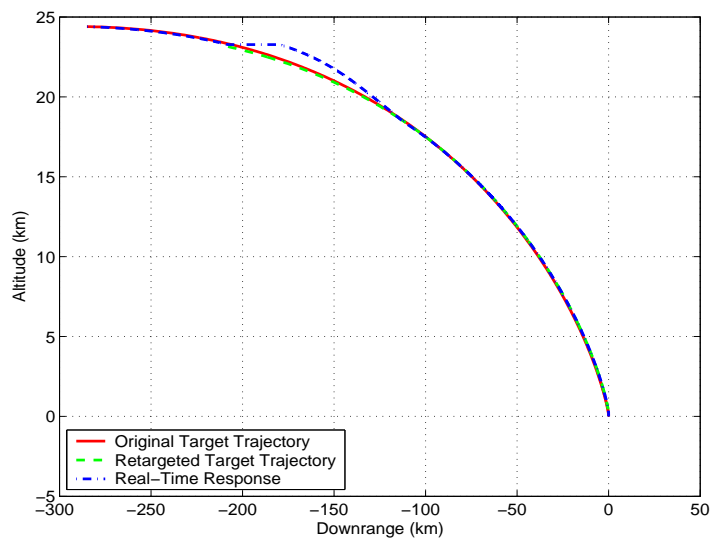


Figure 5.32: Retargeting (Southeast Example): Altitude vs. Time

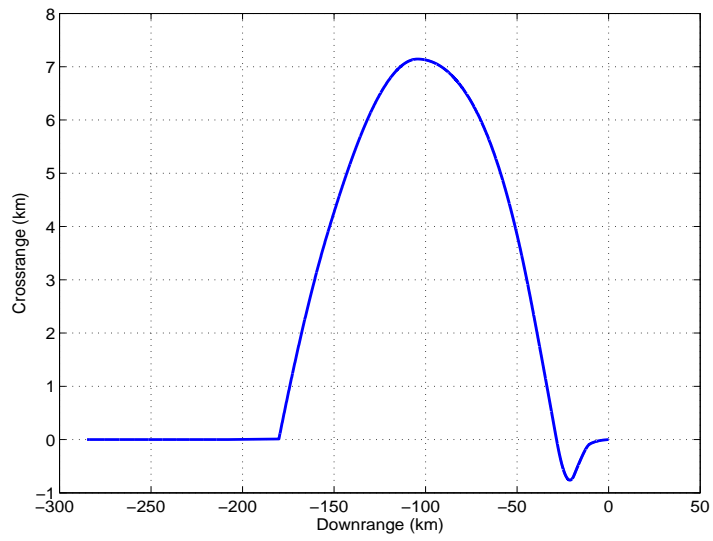


Figure 5.33: Retargeting (Southeast Example): Crossrange vs. Time

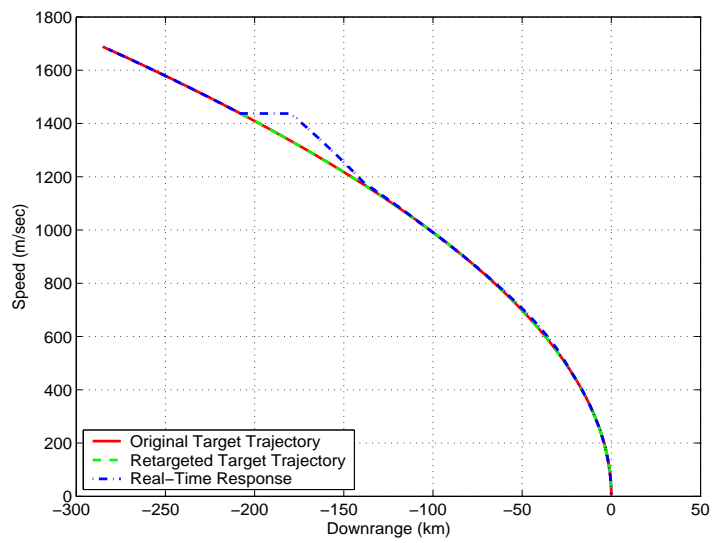


Figure 5.34: Retargeting (Southeast Example): Speed vs. Time

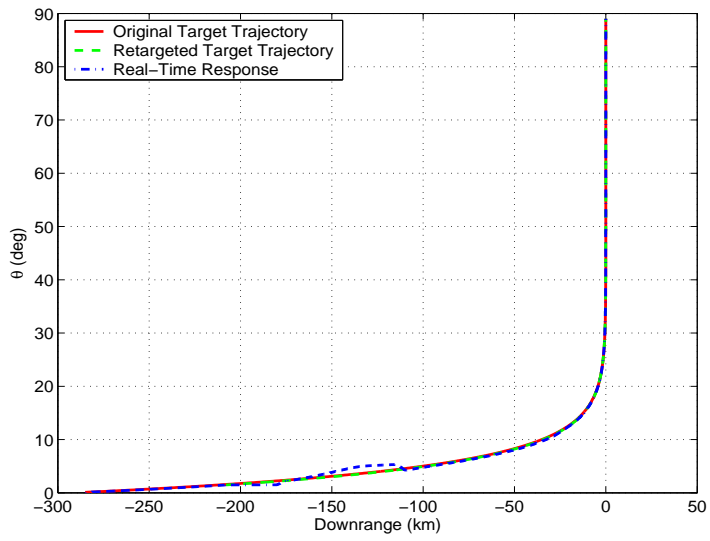


Figure 5.35: Retargeting (Southeast Example): Flight Path Angle vs. Time

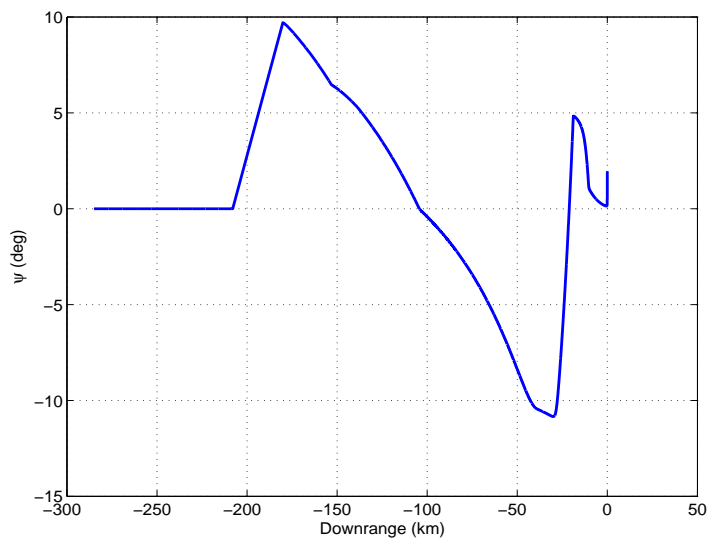


Figure 5.36: Retargeting (Southeast Example): Crossing Angle vs. Time

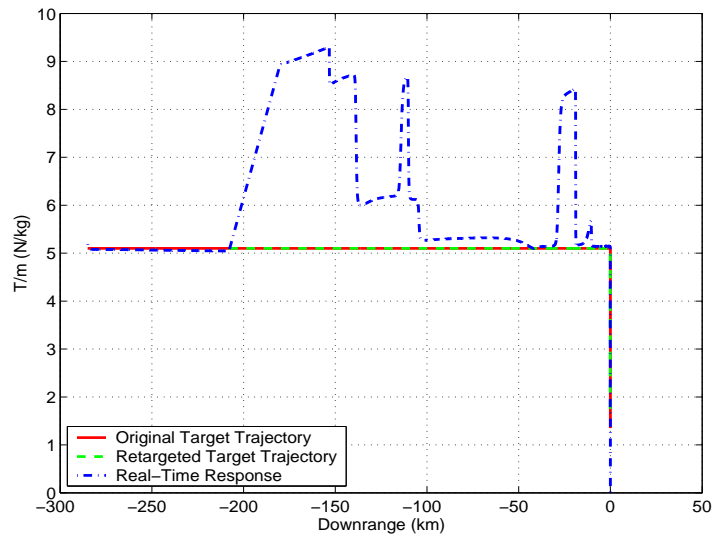


Figure 5.37: Retargeting (Southeast Example): Thrust Acceleration vs. Time

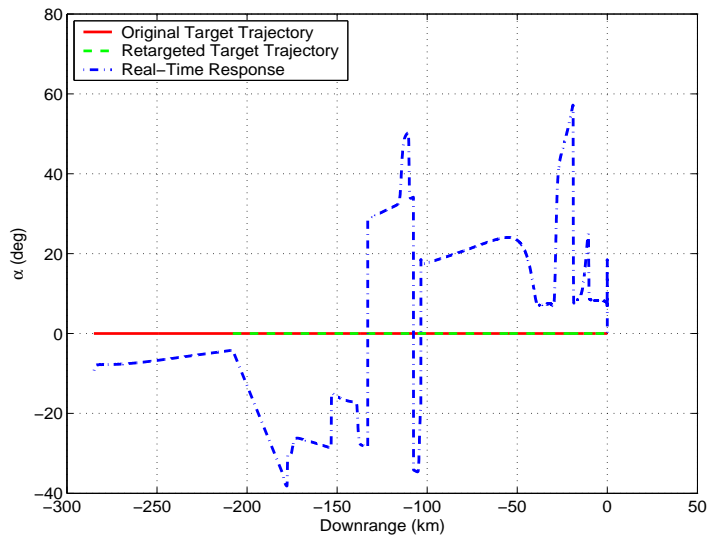


Figure 5.38: Retargeting (Southeast Example): Thrust Angle vs. Time

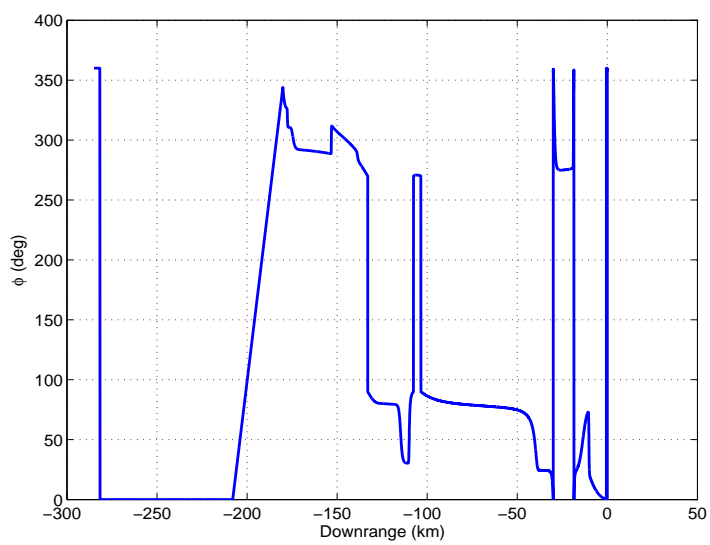


Figure 5.39: Retargeting (Southeast Example): Thrust Roll Angle vs. Time

Monte Carlo runs were performed against the two retargeting cases. Figures 5.40 through 5.49 illustrate the performance against the northeast case with the results summarized in Table 5.5. Figures 5.50 through 5.59 illustrate the performance against the southeast case with the results summarized in Table 5.6. As these figures illustrate, both the retargeting algorithm and the real-time guidance algorithm work well in concert together.

Table 5.5: Monte Carlo Simulation Final State Error Statistics for Northeast Retargeting Case

Error	Target Value	Mean Value	Standard Deviation
Time	varies	431.94 <i>sec</i>	23.61 <i>sec</i>
Downrange	0 <i>m</i>	-0.02 <i>m</i>	0.04 <i>m</i>
Crossrange	0 <i>m</i>	0.00 <i>m</i>	0.03 <i>m</i>
Speed	8 $\frac{m}{sec}$	8.01 $\frac{m}{sec}$.38 $\frac{m}{sec}$
Flight Path Angle	89°	89.01°	0.02°
Crossing Angle	0°	-0.05°	0.68°

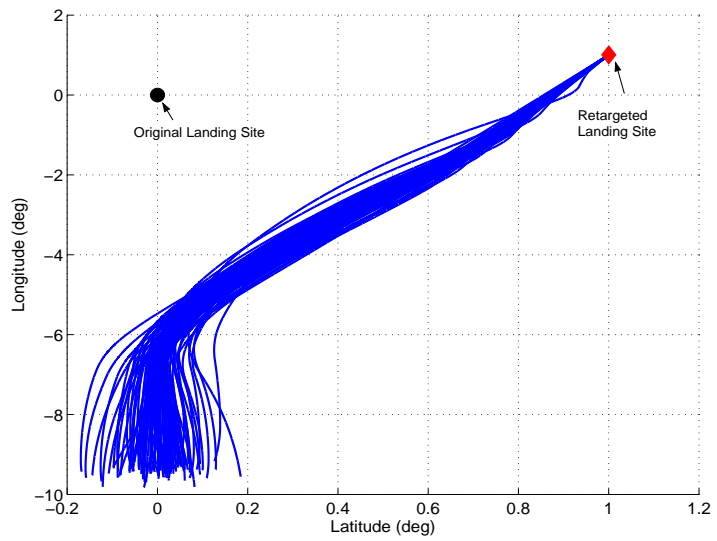


Figure 5.40: Retargeting Monte Carlo Results (Northeast Example): Latitude vs. Longitude

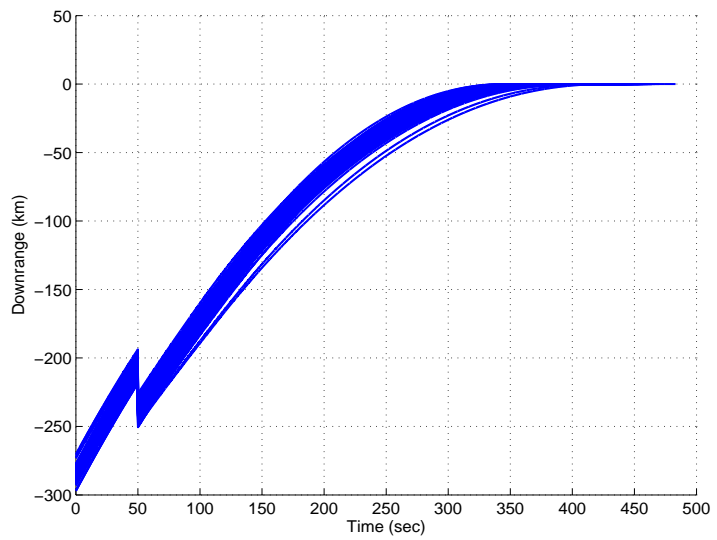


Figure 5.41: Retargeting Monte Carlo Results (Northeast Example): Downrange vs. Time

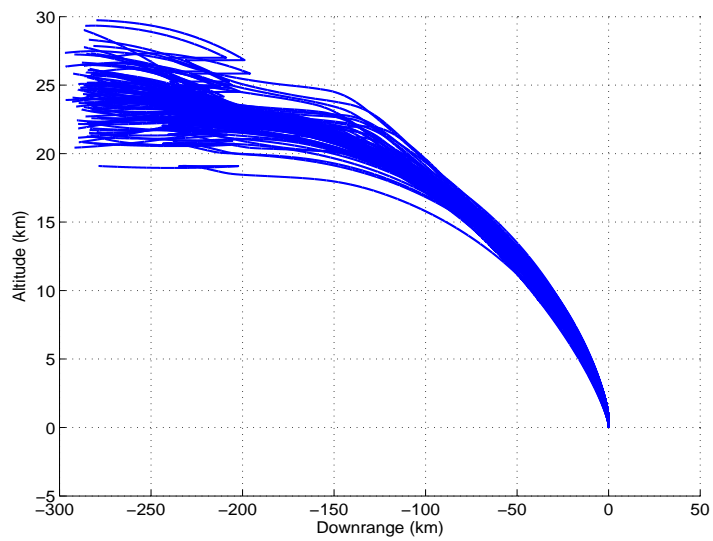


Figure 5.42: Retargeting Monte Carlo Results (Northeast Example): Altitude vs. Time

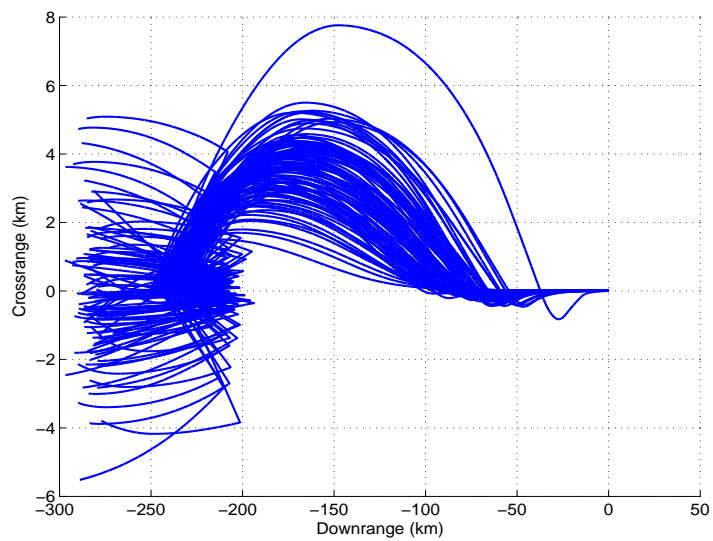


Figure 5.43: Retargeting Monte Carlo Results (Northeast Example): Crossrange vs. Time

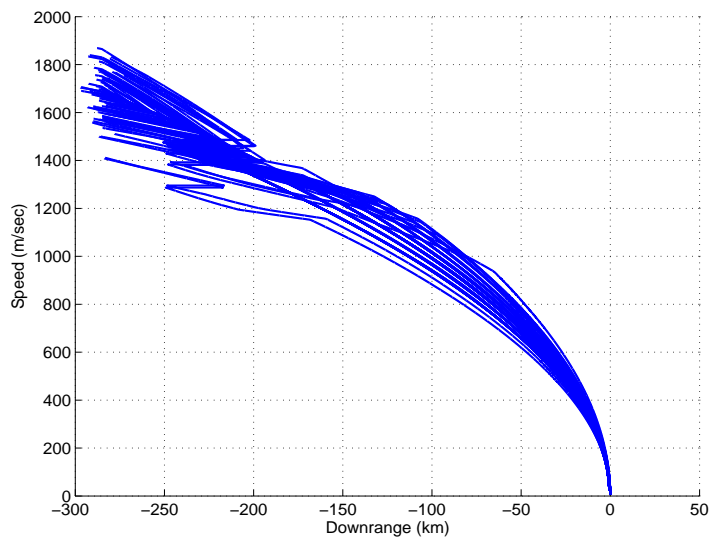


Figure 5.44: Retargeting Monte Carlo Results (Northeast Example): Speed vs. Time

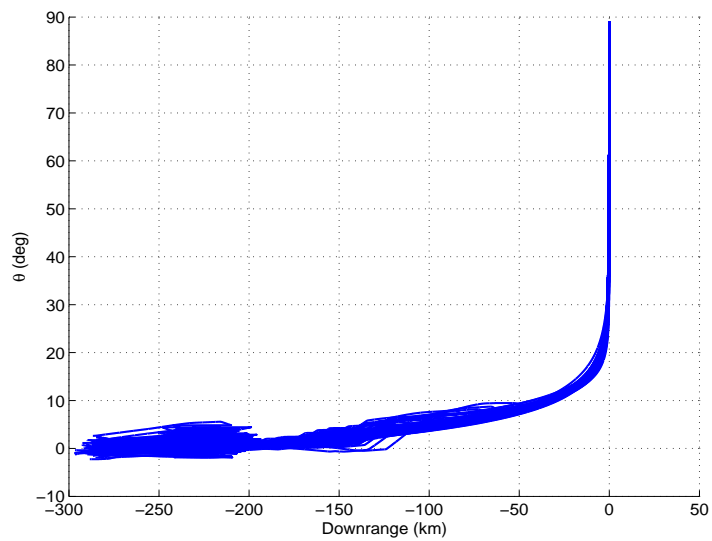


Figure 5.45: Retargeting Monte Carlo Results (Northeast Example): Flight Path Angle vs. Time

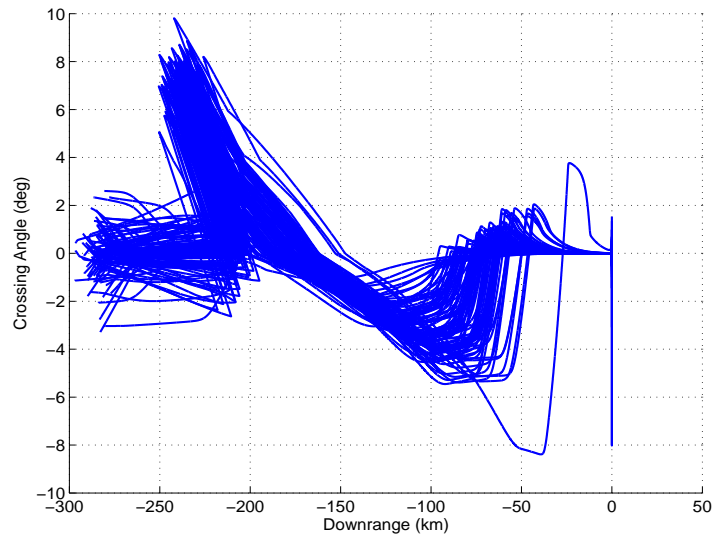


Figure 5.46: Retargeting Monte Carlo Results (Northeast Example): Crossing Angle vs. Time

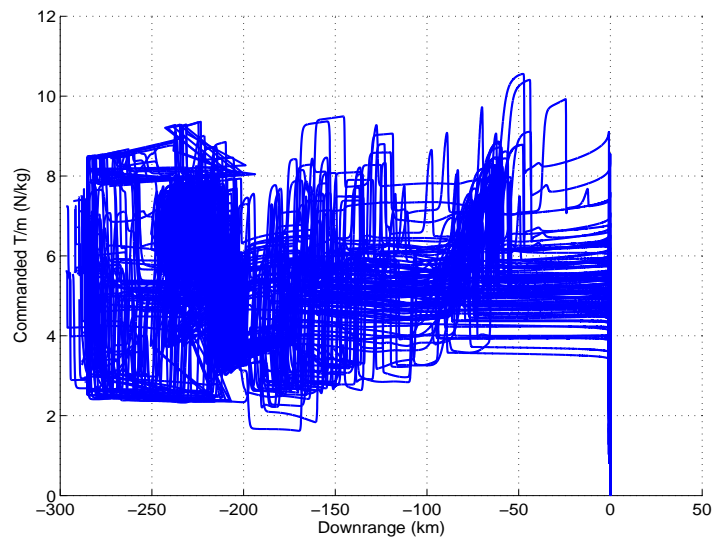


Figure 5.47: Retargeting Monte Carlo Results (Northeast Example): Thrust Acceleration vs. Time

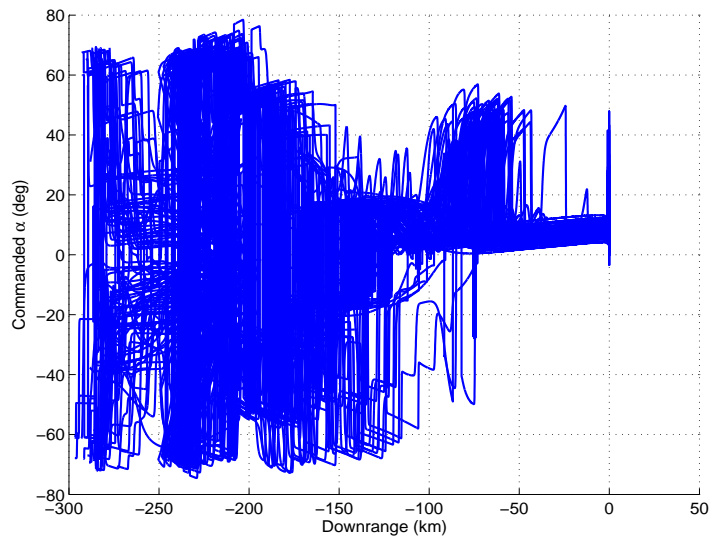


Figure 5.48: Retargeting Monte Carlo Results (Northeast Example): Thrust Angle vs. Time

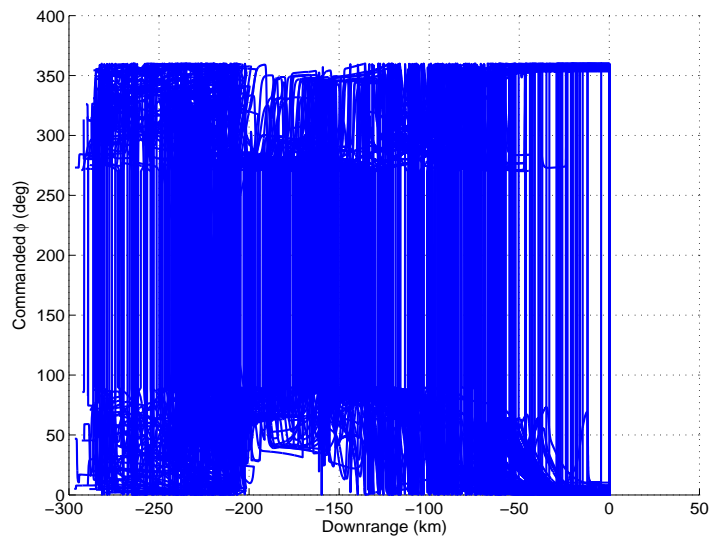


Figure 5.49: Retargeting Monte Carlo Results (Northeast Example): Thrust Roll Angle vs. Time

Table 5.6: Monte Carlo Simulation Final State Error Statistics for Southeast Retargeting Case

Error	Target Value	Mean Value	Standard Deviation
Time	varies	384.14 <i>sec</i>	23.10 <i>sec</i>
Downrange	0 <i>m</i>	-0.02 <i>m</i>	0.04 <i>m</i>
Crossrange	0 <i>m</i>	-0.40 <i>m</i>	1.37 <i>m</i>
Speed	8 $\frac{m}{sec}$	8.02 $\frac{m}{sec}$.36 $\frac{m}{sec}$
Flight Path Angle	89°	88.92°	0.47°
Crossing Angle	0°	6.32°	14.56°

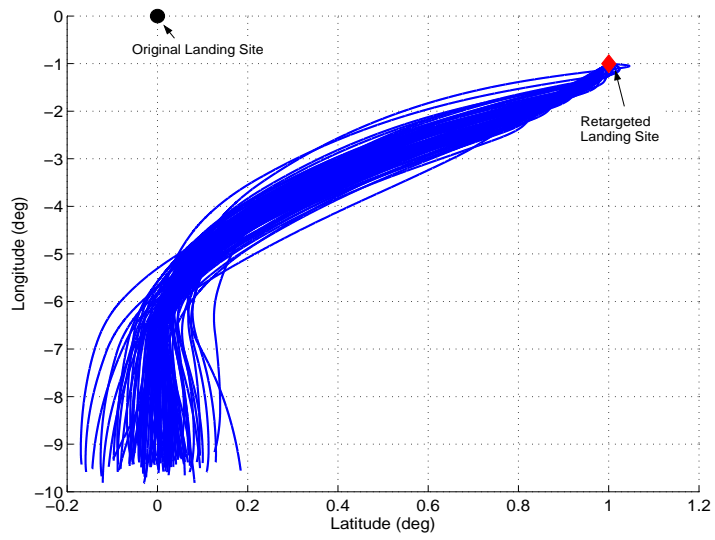


Figure 5.50: Retargeting Monte Carlo Results (Southeast Example): Latitude vs. Longitude

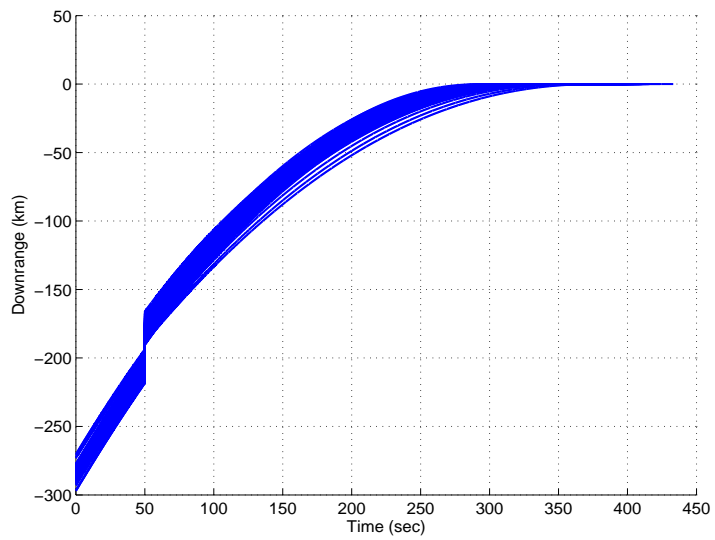


Figure 5.51: Retargeting Monte Carlo Results (Southeast Example): Downrange vs. Time

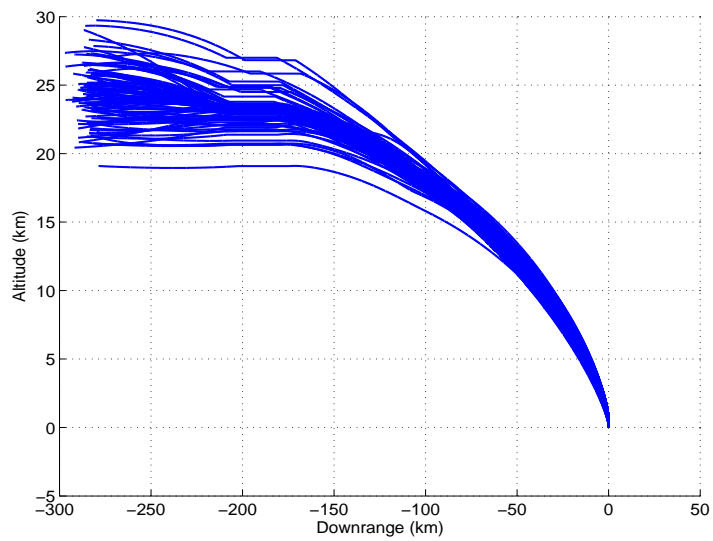


Figure 5.52: Retargeting Monte Carlo Results (Southeast Example): Altitude vs. Time

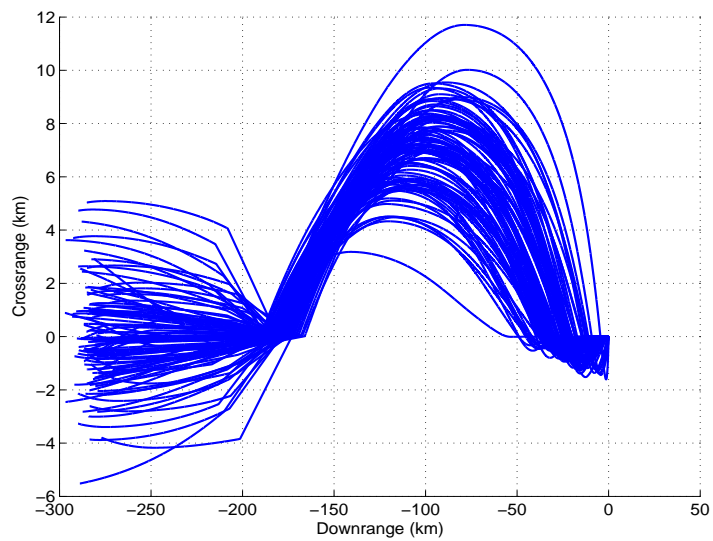


Figure 5.53: Retargeting Monte Carlo Results (Southeast Example): Crossrange vs. Time

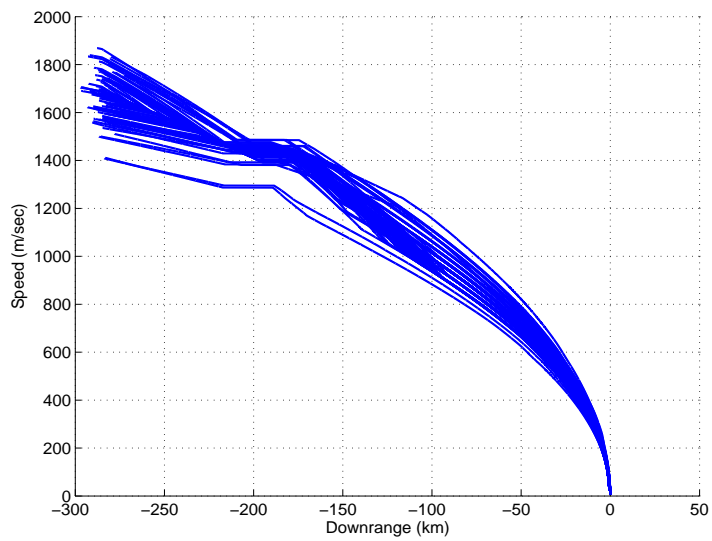


Figure 5.54: Retargeting Monte Carlo Results (Southeast Example): Speed vs. Time

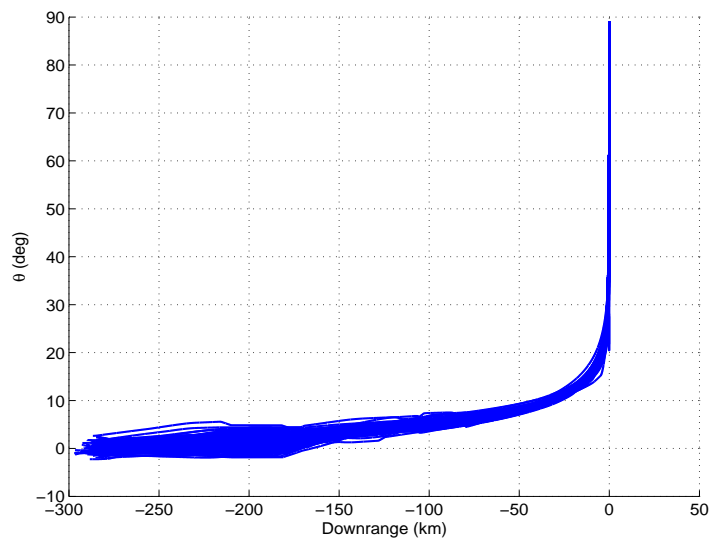


Figure 5.55: Retargeting Monte Carlo Results (Southeast Example): Flight Path Angle vs. Time

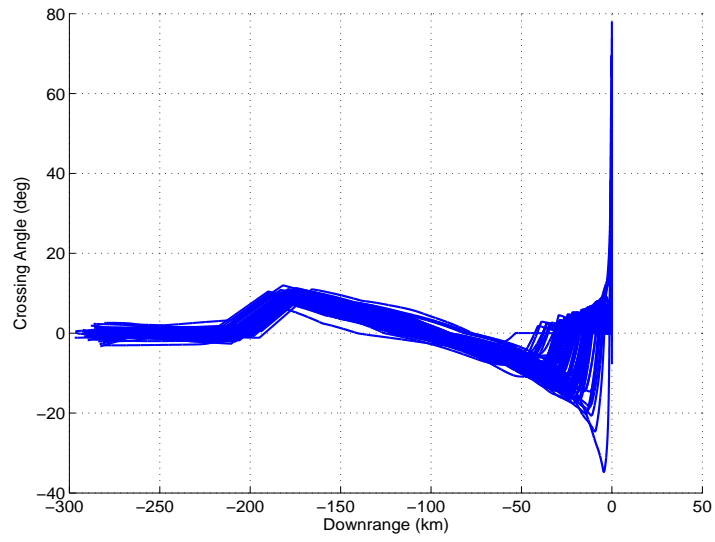


Figure 5.56: Retargeting Monte Carlo Results (Southeast Example): Crossing Angle vs. Time

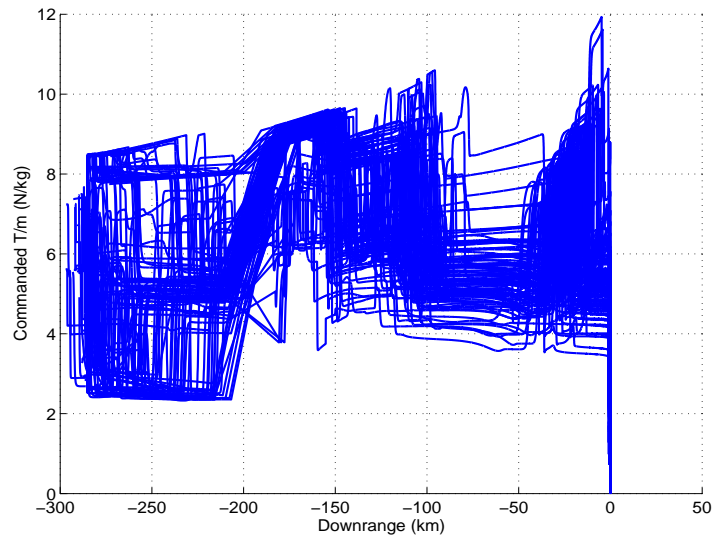


Figure 5.57: Retargeting Monte Carlo Results (Southeast Example): Thrust Acceleration vs. Time

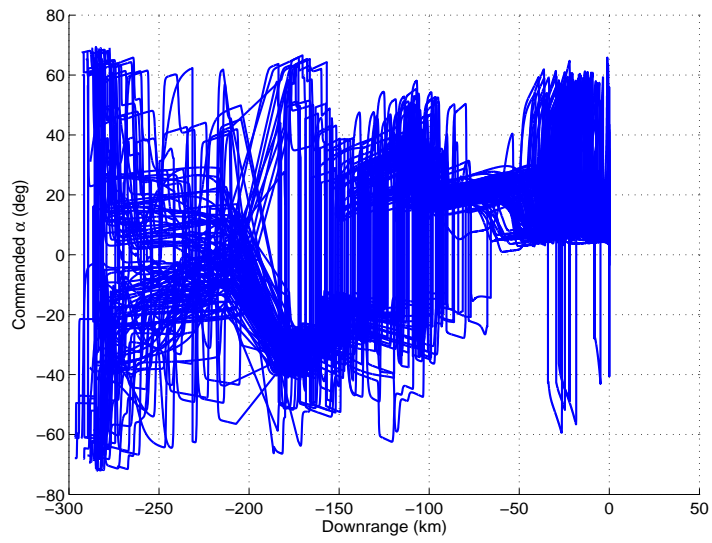


Figure 5.58: Retargeting Monte Carlo Results (Southeast Example): Thrust Angle vs. Time

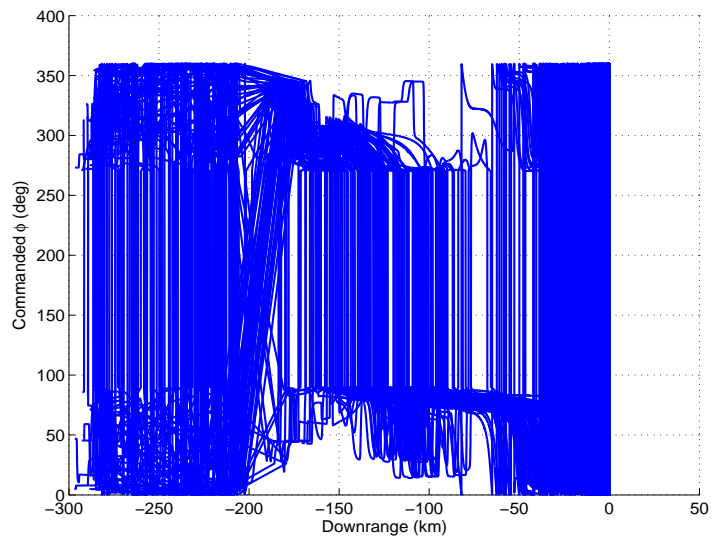


Figure 5.59: Retargeting Monte Carlo Results (Southeast Example): Thrust Roll Angle vs. Time

Chapter 6

Conclusions

The Apollo lunar descent guidance worked well for the goal of taking men safely to the moon. However, as the goal of lunar exploration has changed to encompass a desire to easily and cheaply explore many locations on the moon, new algorithms that allow for ease of change in landing sites is necessary. This document details a viable option for that algorithm.

The Apollo lunar descent guidance algorithm consists of two elements: the targeting algorithm and the real-time guidance algorithm. The proposed algorithm developed herein followed that format to allow for the separate use of the targeting and real-time guidance algorithms. The targeting algorithm can be used to develop an analytical reference trajectory that can then be used as the basis for the real-time guidance. Optionally, a reference trajectory from another development model can be formatted in terms of downrange, altitude, speed, and flight path angle and utilized in the real-time guidance algorithm.

Apollo lunar descent guidance targeting is a complex and iterative algorithm that can not be readily implemented in real-time. It has at its basis the need for a user to select a proper reference trajectory based on fuel consumption and viewing requirements. It also requires the use of a simulation in an iterative fashion that does not guarantee convergence to a solution. The proposed algorithm does not have these shortcomings. It is analytical at its basis and can be implemented automatically as demonstrated in Chapter 5 to

select an acceptable reference trajectory.

Apollo lunar descent real-time guidance requires the reference trajectory be formatted as a vector quartic polynomial. This means that any trajectory that is not developed in that format is unusable by the algorithm. The proposed real-time guidance algorithm does not have this shortcoming. The trajectory must be two-dimensional, but this is reasonable and is the same assumption made in the Apollo targeting algorithm. The primary advantage of the proposed algorithm is that the trajectory need only be formatted as a function of downrange, altitude, speed, and flight path angle and it can be easily referenced by the real-time guidance. Additionally, the Apollo real-time guidance has no guarantees of convergence to the reference trajectory. As proven in Chapter 4, the proposed algorithm is guaranteed to converge as long as the thrust magnitude stays below the maximum allowable thrust; this is the case for most trajectories.

During the development of these algorithms, this author has noted some areas that could be further explored. The first area is in the two segment development of the reference trajectory. The use of more segments should be explored if requirements that were not considered here are important. In particular, the requirement for lunar viewing that is inherent in the Apollo algorithm was not discussed because the proposed algorithm does not require lunar viewing for the retargeting as does Apollo. If, however, this lunar viewing requirement is still deemed important, an additional segment would allow for more degrees of freedom that could meet this requirement.

Another area not explored is the comparison of the reference trajectories generated by the proposed targeting scheme to optimal trajectories. In particular, NASA is very concerned with mass consumption. It might be in-

sightful for a comparison to be made with an optimal fuel usage trajectory to see if some improvements can be made to the proposed trajectory.

The final area that might be explored is in the reference trajectory generation process. This author chose to define a number of trajectories based on current speed and flight path angle and targeted speed and flight path angle and approximate gravity. The trajectory with the closest altitude to the current altitude was selected with disregard for the downrange. This worked well for the cases selected. However, data was presented to illustrate how the reference trajectory solution space changed if gravity, speed, and flight path were varied with thrust. Using this knowledge to select a reference trajectory might provide a more fuel efficient trajectory or one with a more desirable viewing angle and should be explored further.

Bibliography

- [1] Guidance system operations plan for manned lm earth orbital and lunar missions using program luminary 1d. Technical Report R-567, Rev. 8, MIT Charles Stark Draper Laboratory, March 1970.
- [2] Robert H. Bishop and A.C. Antoulas. Nonlinear approach to aircraft tracking problem. *Journal of Guidance, Control, and Dynamics*, 17(5):1124–1130, September - October 1994.
- [3] Robert H. Bishop, Dilmurat Azimov, and Olivier Dubois-Matra. Analysis of spiraling reentry vehicles. *Proceedings of Astrodynamics and Applications III, Princeton, NJ*, June 2006.
- [4] Thomas R. Kane and David A. Levinson. *Dynamics: Theory and Applications*. McGraw-Hill Series In Mechanical Engineering. McGraw-Hill, Inc., 1985.
- [5] Allan R. Klumpp. Apollo guidance, navigation, and control: Apollo lunar-descent guidance. Technical Report R-695, MIT Charles Stark Draper Laboratory, June 1971.
- [6] Colin R. McInnes. Direct adaptive control for gravity-turn descent. *Journal of Guidance, Control, and Dynamics*, 22(2):373–375, March-April 1999.
- [7] Colin R. McInnes. Gravity-turn descent from low circular orbit conditions. *Journal of Guidance, Control, and Dynamics*, 26(1):183–185,

January-February 2003.

- [8] Barrett O’Neill. *Elementary Differential Geometry*. Academic Press, Inc., 1966.
- [9] Shankar Sastry and Marc Bodson, editors. *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall Information and System Sciences Series. Prentice-Hall Inc., 1989.
- [10] Ron Sostaric. Lunar descent reference trajectory, version 2.0. Technical report, NASA/JSC, March 2006.
- [11] Kenji Uchiyama, Yuzo Shimada, and Shingo Nakajima. Tracking control to near-optimal trajectory for a lunar lander. In *Proceedings of the 23rd International Symposium on Space Technology and Science*, volume I, pages 977–982, Matsue, Japan, May 2002.

Vita

Christina Tvrdik Chomel was born Christina Ann Tvrdik in Austin, Texas on 19 March 1973 to Larry and Barbara Tvrdik. She married Justin Alexander Chomel in 1997 with whom she has two children, William Christopher and Caroline Ann. She received the Bachelor of Science degree with High Honors in Aerospace Engineering from The University of Texas at Austin in 1996. During her tenure at U.T., she worked as a cooperative education student at NASA/Johnson Space Center in Houston, Texas. She received the Master of Science degree in Aeronautics and Astronautics from the Massachusetts Institute of Technology in 1998 where she also worked as a Charles Stark Draper Laboratory Graduate Research Fellow. Christina then accepted a position at The Johns Hopkins University Applied Physics Laboratory in Laurel, Maryland where she analyzed guidance, navigation, and control system performance for the Ballistic Missile Defense Organization's Standard Missile-3 Flight Test Program. In 2003, Christina returned to The University of Texas at Austin to pursue her Doctor of Philosophy where she has been a National Science Foundation Graduate Research Fellow.

Permanent address: 3904 Tamarack Trail
Austin, Texas 78727

This dissertation was typeset with \LaTeX^\dagger by the author.

[†] \LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's \TeX Program.