# AN EVALUATION OF APOLLO POWERED DESCENT GUIDANCE

A Thesis

Presented to the

Faculty of

San Diego State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Aerospace Engineering

with a Concentration in

Guidance, Navigation, and Controls

by

Lloyd David Strohl III

Spring 2018

**SAN DIEGO STATE UNIVERSITY**

The Undersigned Faculty Committee Approves the

Thesis of Lloyd David Strohl III:

An Evaluation of Apollo

Powered Descent Guidance

Ping Lu, Chair
Department of Aerospace Engineering

Ahmad Bani Younes
Department of Aerospace Engineering

Peiman Naseradinmousavi
Department of Mechanical Engineering

Approval Date

Copyright © 2018
by
Lloyd David Strohl III

# ABSTRACT OF THE THESIS

An Evaluation of Apollo
Powered Descent Guidance
by
Lloyd David Strohl III
Master of Science in Aerospace Engineering with a Concentration in Guidance, Navigation, and Controls
San Diego State University, 2018

This is my abstract which describes my thesis. It isn't done yet; this is a placeholder.

Many extraterrestrial missions require a powered descent phase. Because this phase is late in the mission its fuel efficiency has an outsized effect on payload capacity. This thesis presents a strategy for optimizing fuel use using well tested guidance algorithms and a unique strategy that reduces fuel consumption over conventional strategies by a significant margin and has wide applicability.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# GLOSSARY

## ACKNOWLEDGMENTS

I would like to thank Dr. Lu for his serendipitous arrival at SDSU and consequent advice and instruction. With his mentoring I have been able to launch an enjoyable and fulfilling career in GN&C, something I could not have achieved otherwise.

# CHAPTER 1
# INTRODUCTION

A manned Mars mission places a heavy penalty on propellant inefficiency. This is because every kilogram of fuel required for landing must be delivered as payload through every phase of the mission until that point, and due to the exponential nature of Tsiolkovsky's rocket equation increasing landing payload mass is extremely expensive. The problem of powered descent guidance is then one of fuel optimality.

## 1.1 Guidance and Control in Aerospace

There is a difference between guidance and control in engineering. Here I describe what that difference is.

## 1.2 Literature Review

The problem of powered descent guidance has been studied extensively throughout the last century, particularly since The Space Race of the 1960s and the Apollo program which spawned E-Guidance, presented first in Cherry 1964 [2]. These analyses have approached the problem in several different ways, but they generally share the requirement that the solution ensure soft landing in vacuum conditions. Most if not all of these analyses attempt to optimize fuel consumption due to the heavy penalty imposed on payload mass by inefficient propellant use when landing on an extraterrestrial body. Almost all approaches also share the assumption of a fixed final time, computed or chosen in various ways. Few of them address the problem of PDI, or when to start powered descent guidance. None investigate PDI with E-Guidance in the context of a manned mission landing in atmosphere. Some work has been done in mission phase planning but little optimization has been done for an initial trajectory with free ignition time.

Apollo Lunar Descent Guidance (E-Guidance) solves the Equations of Motion 2.10 and 2.11 by defining a linear or quadratic thrust acceleration profile which ensures satisfaction of the terminal constraints 2.12, 2.13, and (for constrained final attitude with quadratic thrust profile) 2.29. These two methods require choosing a fixed final time $t_f$, and Cherry proposes an algorithm similar to Algorithm 1. The final time $t_f$ is dependent upon the initial conditions assuming the powered descent guidance is active.

D'Souza did take consideration of an optimal time-to-go in his paper "An Optimal Guidance Law for Planetary Landing" [3]. D'Souza's solution minimizes a weighted function of the time-to-go and the performance index given in Equation 2.19. As discussed below, minimizing this performance index is not fuel optimal but by minimizing the time-to-go a better performance can be realized, as demonstrated in D'Souza's paper. However, this approach still only considers the problem after ignition, when the powered descent guidance phase has already begun.

Rea and Bishop examine the fuel optimal powered descent guidance problem in their paper, "Analytical Dimensional Reduction of a Fuel Optimal Powered Descent Subproblem" [11]. Rea and Bishop explore a time-to-go which optimizes their fuel optimal performance index, but it only considers the portion of flight after a deorbiting "braking" maneuver which ends when the vehicle altitude reaches some pre-designated altitude. They also discuss the approaches previously explored and the limitations of these approaches. These approaches do not explore PDI optimization either.

Meditch showed that the fuel optimal thrust profile for a vertical landing given lower and upper thrust bounds is a "bang-bang" style thrust, where the thrust switches between its upper and lower bounds with at most one switch between the bounds during landing. The paper, "On the Problem of Optimal Thrust Programming For a Lunar Soft Landing," [9] examined the one-dimensional fuel-optimal powered descent in a uniform gravitational field. This strategy may be directly applicable for the final touchdown phase, during which the E-Guidance solution must be stopped when time-to-go gets small as discussed in Section 3.2.

Leitmann also examined a set of two-dimensional rocket flight problems including the two-dimensional powered descent and landing problem, similar to the problem under examination. Leitmann also showed that the fuel optimal thrust profile is "bang-bang," with up to two switches, in his paper "Class of Variational Problems in Rocket Flight." [8]

### 1.3 Contribution

This paper seeks to improve upon the classic E-Guidance solution's fuel performance in the context of a manned Mars mission. The unique requirements of such a mission include high payload mass, aerodynamics of the landing vehicle, critical safety requirements, and the very high penalty for propellant inefficiency. The method by which the E-Guidance solution is improved is through PDI, pushing the solution closer to fuel optimality by making use of a longer glide slope during which energy is shed through drag effects and a thrust profile closer to fuel optimal. The solution presented is equally as computationally expensive as traditional E-Guidance and does not rely on high rate thrust magnitude switching. The guidance law has been studied extensively and flown on real spacecraft.

The method of PDI optimization is applicable beyond the implementation of E-Guidance. This strategy is valid for any multi-phase mission involving powered descent and could be employed in many other contexts. It has particular applicability for atmospheric landings due to the increased efficiency gained from aerodynamic effects, extending the glide and aerobraking phase.

## 1.4 Organization

## 1.5 Problem Statement

Introduce the soft landing problem in the Apollo context and the Mars context.

The guidance algorithm for powered descent and soft landing is formulated under the following assumptions:

1. Atmospheric forces can be neglected
2. Rotation of the planetary body is accounted for by the terminal conditions and the guidance frame formulation
3. The vehicle's engine is not vectored such that the thrust is in line with the vehicle's yaw axis at all times
4. The vehicle's control system is perfect with zero lag
5. The nozzle exit velocity $v_{ex}$ is a known constant
6. The thrust magnitude's upper bound is known
7. The vehicle's state can be reliably measured at all times, including measurement of local gravitational acceleration
8. The vehicle's thrust is throttleable between minimum and maximum values
9. Upon ignition, the vehicle can obtain commanded thrust within its upper and lower bounds instantaneously

Implicit in these assumptions is a specified landing condition which defines the guidance frame $\hat{\boldsymbol{e}}$.

It is desired to minimize propellant usage required by the E-Guidance law by improving PDI. It is essential that the solution is robust and reliable given the safety criticality of a manned mission.

The last assumption is only relevant when ignition is started; the E-Guidance solution does not demand instantaneous throttle responses. Taking into account ignition lag would be simple and not affect performance significantly, but it is not modeled here.

### 1.5.1 CobraMRV

The soft landing problem in atmosphere must be studied with a vehicle model. The vehicle model is taken from Cerimele et. al [1], the CobraMRV, a "rigid, enclosed, elongated lifting body shape that provides a higher lift-to-drag ratio (L/D) than a typical entry capsule...". It was designed as an atmospheric entry and powered descent and landing vehicle for manned Mars missions compatible with the current NASA Human Mars mission architectures. These missions require delivery of roughly 20 tonnes of cargo to the surface. This vehicle, rendered in Figure 1.1 taken from Cerimele et. al, is designed such that its thrust vector is fixed opposite the body yaw axis, i.e. its thrust is oriented down when it is flying horizontally relative to the surface.

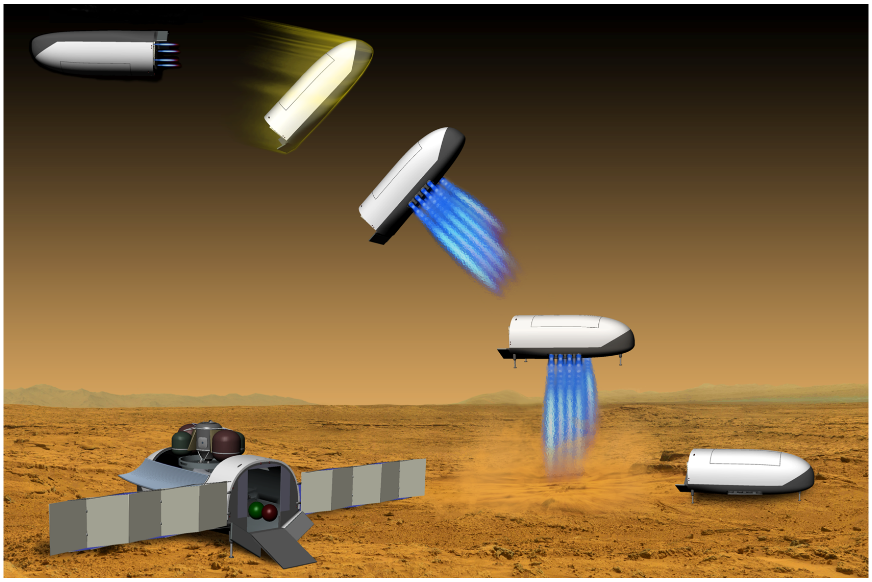

**Figure 1.1. CobraMRV Design. Source: C. Cerimele et al. A rigid mid lift-to-drag ratio approach to human mars entry, descent, and landing. AIAA Paper, 2017-1898, 2017.**

In their paper, Cerimele et. al [1] provide maximum thrust $T_{max}$ as well as mass, lift and drag characteristics, and other factors important for physical modeling and simulation to be described in more detail in Section 3. The maximum thrust magnitude $T_{max}$ is given as $800\ kN$, and the vehicle's mass as the start of powered descent initiation is $58,000kg$.

# CHAPTER 2
# METHODOLOGY

To investigate a guidance law with careful focus on the time-to-go approach, a law must be developed and implemented in a simulation framework. The Law's derivation is presented here in the context of optimal control, as is the time-to-go approach.

The simulation methodology is also described, including the numerical methods and the aerodynamic model.

## 2.1 Notation

This paper uses a set of standard mathematical notation in deriving algorithms and describing models. That notation is defined here.

Vectors consisting of three coordinates in the guidance frame are indicated by bold face type as in Equation 2.1. Also relevant in this equation is that vectors are defined as column matrices.

$$\boldsymbol{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{2.1}$$

Matrices are represented by capital letters as in Equation 2.2. All matrices in this paper are 3x3 matrices.

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \tag{2.2}$$

A column vector $\boldsymbol{V}$ may be converted to a row vector via a transpose operation.

$$\boldsymbol{V}^T = [V_1 \; V_2 \; V_3]$$

Vector multiplication arises mainly in three ways, all used throughout the guidance algorithm derivation. The first is the vector dot product, indicated as in Equation 2.3.

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b} = a_1 b_1 + a_2 b_2 + a_3 b_3 \tag{2.3}$$

Vectors consist of a magnitude and a direction. The magnitude is calculated by way of a vector norm operation as in Equation 2.4

$$||\boldsymbol{V}|| = \sqrt{\boldsymbol{V}^T\boldsymbol{V}} \tag{2.4}$$

The direction of a vector is itself a vector of magnitude 1 which points in the direction of the vector. A direction vector is indicated by a hat $\hat{\boldsymbol{g}}$ as in Equation 2.5. Equation 2.5 represents the direction of gravitational acceleration as a function of position $\boldsymbol{r}$.

$$\hat{\boldsymbol{g}} = \frac{\boldsymbol{g}(\boldsymbol{r})}{||\boldsymbol{g}(\boldsymbol{r})||} \tag{2.5}$$

The second multiplication operations is the cross product of two vectors, indicated by $\times$ as in Equation 2.6 for two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in frame $\hat{\boldsymbol{e}}$, where $|A|$ represents the determinant of matrix $A$.

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{vmatrix} \hat{\boldsymbol{e_1}} & \hat{\boldsymbol{e_2}} & \hat{\boldsymbol{e_3}} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \tag{2.6}$$

The third vector multiplication operation is left matrix multiplication as in Equation 2.7. As shown, left matrix multiplication of a 3x3 matrix $A$ with a 3x1 vector $\boldsymbol{x}$ results in a 3x1 vector $\boldsymbol{b}$.

$$A\boldsymbol{x} = \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{2.7}$$

Equation 2.7 can be interpreted as a system of linear equations in 3 unknowns: $x_1$, $x_2$, and $x_3$. The solution to this linear system requires evaluations of the inverse of the matrix $A$, represented by $A^{-1}$.

Time derivatives are represented with dot notation as in Equation 2.8 showing the derivative relationship between position and velocity vectors.

$$\frac{d\boldsymbol{r}}{dt} = \dot{\boldsymbol{r}} = \boldsymbol{V} \tag{2.8}$$

## 2.2 Coordinate Frame

The guidance algorithm is developed within the guidance coordinate frame $\hat{\boldsymbol{e}}$. This frame can be defined in several ways. Cherry [2] used a gravitational body-centered frame to develop E-Guidance, but any inertial frame can be appropriate. The simulation uses a body-centered frame such that Initial and Final conditions are defined by longitudes, latitudes, and radii from the center of Mars. This simulation neglects the body rotation, but it could be taken into account by defining a rotated final landing site and would not alter the problem qualitatively.

For convenience these Initial Conditions are presented in Table 3.1 after a coordinate transformation to a frame centered on the landing cite so that the final position constraint 2.12 is at the origin. This frame is a right-handed system such that the first coordinate is Mars East, the second is Mars North, and the third is Mars up (the direction opposite the local gravitational acceleration $\boldsymbol{g}$). This E-N-U frame is intuitive because mission conditions such as ground range and altitude are directly represented.

## 2.3 GUIDANCE LAW

The guidance law under investigation is Apollo Powered Descent Guidance (APDG), first presented in Cherry 1964 [2]. APDG is a version of E-Guidance, which was developed empirically by integrating the equations of motion and choosing basis functions for the thrust acceleration $\boldsymbol{a}_T$ that provided the necessary degrees of freedom to satisfy the terminal constraints in Equations 2.12 and 2.13. With control over thrust acceleration (and therefore total acceleration), satisfying the initial conditions after integration of the equations of motion requires two basis functions with vector coefficients. APDG extends this approach to controlled final attitude by adding an additional constraint on final thrust acceleration $\boldsymbol{a}^*_{T_f}$ as in Equation 2.29, and satisfying these three vector constraints requires a further third basis function.

The derivation of E-Guidance as developed by Cherry proceeds below, followed by its formulation as an optimal guidance problem. APDG is then derived as developed by Cherry in a similar fashion though no claim is made about its optimality. The relative performance of the two laws is examined in Chapter 4.

### 2.3.1 Equations of Motion

Derivation of a powered descent guidance law begins with a formulation of the State Equation 2.9

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.9}$$

Cherry derives E-Guidance by treating this set of differential equations as a boundary value problem with initial conditions and final conditions to be satisfied by some command $\boldsymbol{u}$. Aerodynamic effects are not considered for development of the law, though they will be simulated and investigated with regards to performance.

The state equations for the 3-dimensional boundary value problem are as follows

$$\dot{\boldsymbol{r}} = \boldsymbol{V} \qquad \boldsymbol{r}(\boldsymbol{t_0}) = \boldsymbol{r_0} \tag{2.10}$$

$$\dot{\boldsymbol{V}} = \boldsymbol{g}(\boldsymbol{r}) + \boldsymbol{a_T} \qquad \boldsymbol{V}(\boldsymbol{t_0}) = \boldsymbol{V_0} \tag{2.11}$$

with terminal constraints at a fixed final time $t_f$

$$\boldsymbol{r}(t_f) = \boldsymbol{r}_f^* \tag{2.12}$$

$$\boldsymbol{V}(t_f) = \boldsymbol{V}_f^* \tag{2.13}$$

where $\boldsymbol{a}_T$ is the thrust acceleration vector and $\boldsymbol{g}(\boldsymbol{r})$ is gravitational acceleration

$$\boldsymbol{g} = -\frac{\mu \boldsymbol{r}}{(\boldsymbol{r}^T\boldsymbol{r})^{(3/2)}} \tag{2.14}$$

where $\mu$ is the standard gravitational parameter for the body in question. For Mars, $\mu \approx 4.282 * 10^{13}$

Cherry recognized that a desired performance index would minimize the integral of the thrust acceleration magnitude as in Equation 2.18. He also recognized that the thrust acceleration $\boldsymbol{a}_T$ is limited such that

$$0 < a_{min} \leq ||\boldsymbol{a}_T|| \leq a_{max} \tag{2.15}$$

By choosing the command $\boldsymbol{u} = \boldsymbol{g(r)} + \boldsymbol{a}_T$ so that the nonlinearities of the gravitational acceleration are effectively removed from the problem, a boundary value problem can be constructed by directly integrating the equations of motion and applying the boundary values to form a set of linear equations to be solved.

### 2.3.2 E-Guidance

While the set of non-linear equations presented in Section 2.3.1 do uniquely define a guidance command $\boldsymbol{u}$, Cherry recognized that they represented a ”formidable bunch” of equations that he could not practically solve. Instead, he considered the infinite-dimensional generalized Fourier series that must represent the guidance command $\boldsymbol{u}$. By reducing the order of the Fourier series to equal the degrees of freedom demanded by the vector constraints of Equations 2.12 and 2.29, Cherry developed E-Guidance by considering first one guidance axis at a time, a ”divide and conquer” strategy, so the acceleration command took the form of Equation 2.16, where $p_1(t)$ and $p_2(t)$ are linearly independent functions of time.

$$\ddot{x} = c_1 p_1(t) + c_2 p_2(t) \tag{2.16}$$

Cherry sets $p_1(t) = 1$ and $p_2(t) = t$, mostly for simplicity while recognizing that it may be suboptimal, arriving at a form identical to the one presented in Equation 2.21.

Cherry also proposes a time-to-go algorithm similar to that in Algorithm 1. It does not, however, do much to consider the optimality of this algorithm or, more critically, to deal with the problem of powered descent initiation.

### 2.3.3 Optimality

Cherry's solution for E-Guidance matches the result from the solution to an optimal guidance problem with a particular performance index.

#### 2.3.3.1 Performance Index

Fuel consumption is related to the thrust acceleration vector by engine parameters represented by some positive constant k

$$\dot{m} = -k||\boldsymbol{a}_T|| \tag{2.17}$$

A fuel optimal guidance law should therefore use the performance index

$$J = \int_{t_0}^{t_f} ||\boldsymbol{a}_T||dt \tag{2.18}$$

Choosing to minimize the square of the total acceleration $\boldsymbol{a} = \boldsymbol{g} + \boldsymbol{a}_T$ gives a performance index

$$J = \frac{1}{2}\int_{t_0}^{t_f} (\boldsymbol{g} + \boldsymbol{a}_T)^T(\boldsymbol{g} + \boldsymbol{a}_T)dt \tag{2.19}$$

For a constant gravitational acceleration $\boldsymbol{g}$, this performance index attempts to minimize $||\boldsymbol{a}_T||^2$. It is not fuel optimal as in Equation 2.18, but it does provide a cost to large thrust accelerations and might be expected to give good fuel performance.

#### 2.3.3.2 E-Guidance As An Optimal Guidance Law

Choosing the guidance command $\boldsymbol{u} = \boldsymbol{g} + \boldsymbol{a}_T$ and applying optimal control theory results in the following

$$H = \boldsymbol{p}_r^T\boldsymbol{V} + \boldsymbol{p}_V^T\boldsymbol{u} - \frac{1}{2}\boldsymbol{u}^T\boldsymbol{u} \tag{2.20}$$

$$\dot{\boldsymbol{p}}_r = -\frac{\partial H}{\partial \boldsymbol{r}} = 0 \implies \boldsymbol{p}_r = -\boldsymbol{c}_2$$

$$\dot{\boldsymbol{p}}_V = -\frac{\partial H}{\partial \boldsymbol{V}} = -\boldsymbol{p}_r \implies \boldsymbol{p}_V = \boldsymbol{c}_1 + \boldsymbol{c}_2 t$$

$$\frac{\partial H}{\partial \boldsymbol{u}} = 0 \implies \boldsymbol{u} = \boldsymbol{p}_V = \boldsymbol{c}_1 + \boldsymbol{c}_2 t$$

For convenience, let $\tau = t_f - t$

$$\boldsymbol{u} = \boldsymbol{k}_1 + \boldsymbol{k}_2\tau \tag{2.21}$$

where $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ are constant vectors.

Integrating the equations of motion with $\dot{\boldsymbol{V}} = \boldsymbol{u}$ then gives

$$\int \dot{\boldsymbol{V}}(t)dt = \boldsymbol{k}_1(t - t_0) + \frac{1}{2}\boldsymbol{k}_2(t - t_0)^2 + \boldsymbol{V}(t_0) \tag{2.22}$$

$$\int \dot{\boldsymbol{r}}(t)dt = \frac{1}{2}\boldsymbol{k}_1(t - t_0)^2 + \frac{1}{6}\boldsymbol{k}_2(t - t_0)^3 + \boldsymbol{V}(t_0)(t - t_0) + \boldsymbol{r}(t_0) \tag{2.23}$$

Setting $t = t_f$ and letting $t_{go} = t_f - t_0$ satisfies the terminal constraints from Equations 2.12 and 2.13, resulting in 6 linear equations in 6 unknowns

$$\boldsymbol{k}_1 t_{go} + \frac{1}{2}\boldsymbol{k}_2 t_{go}^2 = \boldsymbol{V}_f^* - \boldsymbol{V}_0 \tag{2.24}$$

$$\frac{1}{2}\boldsymbol{k}_1 t_{go}^2 + \frac{1}{6}\boldsymbol{k}_2 t_{go}^3 = \boldsymbol{r}_f^* - \boldsymbol{r}_0 - \boldsymbol{V}_0 t_{go} \tag{2.25}$$

These equations can be separated into sets of two per vector component. Define an inertial guidance frame $\boldsymbol{e} = (\hat{x}, \hat{y}, \hat{z})^T$ such that guidance vector $\boldsymbol{u}$ is composed of components in $\boldsymbol{e}$, $\boldsymbol{u} = (u_x, u_y, u_z)^T$. For the equations in $\hat{x}$ we have

$$\begin{bmatrix} t_{go} & \frac{1}{2}t_{go}^2 \\ \frac{1}{2}t_{go}^2 & \frac{1}{6}t_{go}^3 \end{bmatrix} \begin{pmatrix} k_{1_x} \\ k_{2_x} \end{pmatrix} = \begin{pmatrix} V_{f_x}^* - V_{0_x} \\ r_{f_x}^* - (r_{0_x} + V_{0_x} t_{go}) \end{pmatrix} \tag{2.26}$$

Solving the two-equation system is accomplished by inverting the left-hand matrix, leading to a coefficient matrix $E$

$$E = \begin{bmatrix} -2/t_{go} & 6/t_{go}^2 \\ 6/t_{go}^2 & -12/t_{go}^3 \end{bmatrix} \tag{2.27}$$

The coefficients in $\hat{x}$ are then

$$\begin{pmatrix} k_{1_x} \\ k_{2_x} \end{pmatrix} = E \begin{pmatrix} V_{f_x}^* - V_{0_x} \\ r_{f_x}^* - (r_{0_x} + V_{0_x} t_{go}) \end{pmatrix} \tag{2.28}$$

It can be shown that the equations in $\hat{y}$ and $\hat{z}$ take the same form. This $2x2$ $E$ matrix is the origin of the name *E-Guidance*, the guidance law used in the Apollo lunar landing missions.

### 2.3.4 Apollo Powered Descent Guidance

APDG enforces a final attitude constraint as a matter of practicality. A real rocket landing would typically require that the vehicle be oriented in a specific direction upon landing. Without explicitly enforcing a final attitude, the vehicle could touch down at a large angle while still satisfying final velocity and position constraints. For a vehicle whose attitude is determined by the thrust acceleration vector, this constraint can be implemented as a final thrust acceleration constraint as in Equation 2.29.

$$\boldsymbol{a}_T(t_f) = \boldsymbol{a}^*_{T_f} \tag{2.29}$$

A logical choice for this final thrust acceleration constraint is a vector in the negative gravitational acceleration direction as in Equation 2.30 where c is some positive scalar, ensuring that the vehicle touches down vertically.

$$\boldsymbol{a}^*_{T_f} = -c\hat{\boldsymbol{g}} \tag{2.30}$$

This vector constraint cannot be satisfied with only two basis functions for the command $\boldsymbol{u}$, so a third linearly independent function must be introduced such that $\boldsymbol{u} = \boldsymbol{c}_1 p_1(t) + \boldsymbol{c}_2 p_2(t) + \boldsymbol{c}_3 p_3(t)$. Cherry's choice for the third basis function is $p_3(t) = t^2$ for simplicity, with the other two functions the same as in E-Guidance.

After applying the substitution from Equation 2.21, this choice gives a command $\boldsymbol{u} = \boldsymbol{k}_1 + \boldsymbol{k}_2\tau + \boldsymbol{c}_3\tau^2$. This form for the command input is also easily integrable and leads to a similar linear system of 9 equations in $\boldsymbol{k_1}$, $\boldsymbol{k_2}$, and $\boldsymbol{k_3}$. Using the same guidance frame $\hat{e}$ defined for Equation 2.26 and considering one coordinate a time we get a system similar to E-Guidance in Equation 2.31

$$\begin{pmatrix} k_{1_x} \\ k_{2_x} \\ k_{3_x} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 18/t_{go}^2 & -24/t_{go}^3 & -6/t_{go} \\ -24/t_{go}^3 & 36/t_{go}^4 & 6/t_{go}^2 \end{bmatrix} \begin{pmatrix} V^*_{f_x} - V_{0_x} \\ r^*_{f_x} - (r_{0_x} + V_{0_x} t_{go}) \\ g_x + a^*_{f_x} \end{pmatrix} \tag{2.31}$$

### 2.3.5 Time-to-go

APDG depends upon a reliable estimate of remaining time-to-go ($t_{go}$). The Apollo mission's guidance used an estimate that updated continuously using Newton's method, but it was intended to only operate until start of the terminal descent phase at which point guidance switched to a manual vertical descent operation. Updating the $t_{go}$ estimate continuously is attractive since it should be robust; if conditions have to change during the mission a closed-loop (continuously updating) solution will adjust and a new, realistic $t_{go}$ will feed into the guidance solution. This quality was important to the Apollo Guidance solution because it relied upon pilot inputs to define the landing location visually, which meant allowing for landing site redesignations mid-mission. If $t_{go}$ was not recomputed after site redesignation, the guidance law would command unrealizable thrust acceleration commands.

For the purposes of this study, live landing site redesignation was not considered. Without the possibility of landing site redesignation, an open-loop $t_{go}$ solution lends the guidance law more stability in that the performance is less dependent upon specific assumptions and conditions imposed by the $t_{go}$ algorithm. For instance, one closed-loop $t_{go}$ algorithm is implemented in Algorithm 1.

**Algorithm 1** Fixed-Point-Iteration $t_{go}$

$tol \leftarrow c$
**while** $|t_{go_0} - t_{go_1}| \geq tol$ **do**
    $t_{go_0} \leftarrow t_{go_1}$
    $\Delta V \leftarrow \sqrt{(\boldsymbol{V} - \boldsymbol{V_0} + \boldsymbol{g} \cdot t_{go})^T(\boldsymbol{V} - \boldsymbol{V_0} + \boldsymbol{g} \cdot t_{go_0})}$
    $t_{go_1} \leftarrow \frac{m_0}{\dot{m}}\left(e^{\frac{-\Delta V}{v_{ex}}} - 1\right)$ $\triangleright\ \dot{m} < 0$
**end while**
**return** $t_{go_1}$

Each guidance update uses the previous update's $t_{go}$ minus clock time as its initial guess $t_{go_0}$, and the max iterations may be limited to some reasonable number.

This algorithm requires an assumption about a fixed mass flow rate $\dot{m}$ which is not guaranteed by the guidance law. Adjustment of this mass flow rate estimate is very particular to the initial conditions of the mission, resulting in a necessarily conservative $t_{go}$ to account for initial condition dispersion.

One attractive open-loop option is the time to perform a gravity turn landing at maximum thrust. Equation 2.32 for a gravity turn time-to-go, $t_{go_{GT}}$, was presented first in Cherry 1964 [2]. After engine ignition and initiation of Powered Descent Guidance, the updated time-to-go is computed as $t_{go_{GT}}$ minus elapsed clock time. The algorithm for time-to-go using a gravity turn maneuver is given in Equation 2.32, where $a_{GT}$ is the thrust acceleration magnitude applied during the maneuver. This equation gives two roots for $a_{GT}$, a positive root and a negative root. The desired root is positive, making it a simple matter to find. This quantity will be important in section **??**.

$$
\begin{aligned}
\gamma &= \frac{\pi}{2} - cos^{-1}\left(\frac{\boldsymbol{r}^T\boldsymbol{V}}{||\boldsymbol{r}||||\boldsymbol{V}||}\right) \\
g_m &= ||\boldsymbol{g}|| \\
r_m &= ||\boldsymbol{r}|| \\
V_m &= ||\boldsymbol{V}|| \\
a &= 1/g_m^2 \\
b &= \frac{sin(\gamma)V_m^2}{2(r_m - R_M)g_m^2} \\
c &= -\frac{V_m^2(1+sin(\gamma)^2)}{4(r_m - R_M)g_m} + 1 \\
a_{GT} &= \frac{-b \pm \sqrt{b^2-4ac}}{2a} \\
t_{go_{GT}} &= \frac{V_m}{2}\left(\frac{1+sin(\gamma)}{a_{GT}+g_m} + \frac{1-sin(\gamma)}{a_{GT}-g_m}\right)
\end{aligned}
\tag{2.32}
$$

A gravity turn landing does not directly apply to the general powered descent guidance problem under investigation because it does not seek to satisfy the constraints given in Equations 2.12 and 2.13. However, if the magnitude of the terminal velocity target in Equation 2.13 is small, the required time to decelerate from an initial $\boldsymbol{V}_0$ under only the forces of thrust acceleration $\boldsymbol{a}_T$ and gravity $\boldsymbol{g}$ is, at minimum, the gravity turn solution $t_{go_{GT}}$. Since the terminal position $\boldsymbol{r}_f$ is specified, the vehicle necessarily needs more time than given by the gravity turn solution to satisfy it. Assuming a small trajectory error requiring a small diversion to landing site, a small constant factor $c_t \approxeq 1.2$ can be applied to the gravity turn time-to-go to allow for redirection. The choice of $c_t = 1.2$ will prove to be sufficiently conservative to survive initial condition dispersion, rocket parameter dispersion, and navigation error.

## 2.4 Adaptive Powered Descent Initiation

The central focus of this thesis is the use of Adaptive Powered Descent Initiation (PDI) to improve the fuel performance of APDG. This is accomplished by choosing an ignition time which results in superior propellant consumption. From Leitmann's [8] and Meditch's [9] work it is expected that a "bang-bang" style thrust profile is desirable. Lu shows in his paper "Propellant-Optimal Powered Descent Guidance" that for the three-dimensional problem, the propellant optimal thrust profile is indeed "bang-bang" style and may have up to two switches. He also shows that the profile is further constrained to a final thrust value $T(tf) = T_{max}$ in order to satisfy the final position and velocity constraints. With APDG controlling the thrust acceleration vector $\boldsymbol{a}_T$, a "bang-bang" style thrust profile will not be realized due to the continuous nature of the boundary-value problem presented, but the ignition time may be chosen to force a thrust profile closer to optimum. By choosing a time-to-go carefully, nearly optimal fuel performance may be expected.

Since APDG does not take the thrust acceleration limits of Equation 2.15 into account directly the time-to-go must do so or risk the required thrust command exceeding the capabilities of the landing vehicle. One approach is to make use of the gravity turn solution as discussed in Section 2.3.5, Equation 2.32. This equation provides one criterion for ignition through the specified thrust acceleration $\boldsymbol{a}_{GT}$ as in Algorithm 2. If the gravity turn time-to-go $t_{go_{GT}}$ is used and the engine ignited for APDG the moment the magnitude of the specified thrust acceleration $\boldsymbol{a}_{GT}$ reaches or exceeds the known maximum thrust acceleration magnitude $a_{max}$, APDG will command a large initial thrust acceleration which may exceed the rocket's limits and the thrust will be nearly or copmletely saturated at its upper limit. Using a padded $t_{go_{GT}}$ helps ensure that there is some margin for error.

**Algorithm 2** Thrust Acceleration Criterion

1: **if** $|\boldsymbol{a}_{GT}| \geq \frac{T_{max}}{m}$ **then**
2: $\quad ignite\ engine$
3: **end if**

Another important criterion is the downrange travel required by a gravity turn maneuver. The powered descent phase should be initiated if the horizontal distance remaining in the velocity direction to landing site is greater than or equal to the downrange distance traveled during a gravity turn maneuver. This will help ensure that the vehicle does not overshoot the landing site which would ultimately require far more fuel than a less optimal thrust profile. Equation2.33 gives the downrange distance traveled during a gravity turn maneuver $s_{GT}$, where $h$ is the current altitude. Algorithm 3 details this ignition logic.

$$s_{GT} = \frac{V_m^2}{2 * a_{GT})} * cos(\gamma) * \frac{V_m^2 + 2 * g_m * h}{V_n orm^2 + g_m * h} * \frac{R_M}{r_m} \tag{2.33}$$

**Algorithm 3** Range Criterion

1: **if** $|s_{GT}| \leq |\boldsymbol{r}_{xy} - \boldsymbol{r_f}_{xy}|$ **then**
2: $\quad ignite\ engine$
3: **end if**

Notably, Cerimele et. al [1] recognize in the design of the CobraMRV (described in Section **??**) that ”the selection of the altitude to transition to powered descent ... is a crucial performance variable.” They suggest a timing strategy similar to that described in this section.

# CHAPTER 3
# SIMULATION

The powered descent simulation (PD Sim) is developed using standard programming techniques and numerical methods. The code is modular to facilitate design and reflect the functions of a mission computer. The simulation has three degrees of freedom (3-DoF) with derived orientation for force modeling. The simulation takes a set of initial conditions $\boldsymbol{r}_0$ and $\boldsymbol{V}_0$, a set of final conditions $\boldsymbol{r}_f$ and $\boldsymbol{V}_f$, dispersed rocker parameters, and nominal rocket parameters for the guidance module to use in its calculations of throttle setting.

At the core of the simulation is numerical time integration. The method employed is a 4th order Runge-Kutta (RK4). The RK4 function is called at each time step to progress the simulation forward.

The separate modules called during simulation are the guidance computer which contains the guidance law and computes a commanded thrust acceleration vector, the navigation module which generates a state estimate from simulated instrument measurements for use by the guidance computer, and the aerodynamic module which contains a model of the landing vehicle to compute forces and orientation for use in integration.

The simulation stops when either the vehicle passes through zero altitude (crashes) or the time-to-go is less than a single time integration step. When the latter condition is met, the time integration step is reduced to the remaining time-to-go and one more iteration is performed.

The full simulation code is wrapped in a Monte-Carlo script. The Monte-Carlo script calls the PD Sim with simulation settings, vehicle parameters, initial condition, and navigation uncertainty. This allows each individual run to have dispersed conditions without modification of the code.

Further detail for each part of the simulation follows.

## 3.1 Numerical Integration

The numerical time integration is done via a standard Runge-Kutta formulation, presented here as in Ferziger and Perić [4]. The state vector $\boldsymbol{\phi}$ is passed to the equations of motion $\boldsymbol{f}(\boldsymbol{\phi}, t)$ several times as in Equation 3.1. $\Delta t$, the time integration step for the simulation, is fixed at 1 millisecond ($10^{-3}$ s). This is shown to be sufficiently small in Section 5.1.

$$\begin{aligned}
\boldsymbol{\phi}^*_{n+\frac{1}{2}} &= \boldsymbol{\phi}_n + \frac{\Delta t}{2}\boldsymbol{f}(t_n, \boldsymbol{\phi}_n) \\
\boldsymbol{\phi}^{**}_{n+\frac{1}{2}} &= \boldsymbol{\phi}^n + \frac{\Delta t}{2}\boldsymbol{f}(t_{n+\frac{1}{2}}, \boldsymbol{\phi}^*_{n+\frac{1}{2}}) \\
\boldsymbol{\phi}^*_{n+1} &= \boldsymbol{\phi}^n + \Delta t\boldsymbol{f}(t_{n+\frac{1}{2}}, \boldsymbol{\phi}^{**}_{n+\frac{1}{2}}) \\
\boldsymbol{\phi}_{n+1} &= \boldsymbol{\phi}^n + \frac{\Delta t}{6}[\boldsymbol{f}(t_n, \boldsymbol{\phi}_n) + 2\boldsymbol{f}(t_{n+\frac{1}{2}}, \boldsymbol{\phi}^*_{n+\frac{1}{2}}) \\
&\quad + 2\boldsymbol{f}(t_{n+\frac{1}{2}}, \boldsymbol{\phi}^{**}_{n+\frac{1}{2}}) + \boldsymbol{f}(t_{n+1}, \boldsymbol{\phi}^*_{n+1})]
\end{aligned} \tag{3.1}$$

The EOM function computes the derivative of the state vector $\boldsymbol{\phi}$. It is the same as the modeled equations of motion in Equations 2.10 and 2.11 but with an additional force term, $\boldsymbol{F}_{LD}$, to represent the aerodynamic forces of Lift and Drag. $\boldsymbol{\phi}$ is passed in as a $6x1$ column vector such that $\boldsymbol{\phi} = [\boldsymbol{r}^T, \boldsymbol{V}^T]^T$, and the derivatives are calculated as in Equations 3.2 and 3.3.

$$\dot{\boldsymbol{r}} = \boldsymbol{V} \tag{3.2}$$

$$\dot{\boldsymbol{V}} = \boldsymbol{g}(\boldsymbol{r}) + \boldsymbol{a_T} + \frac{\boldsymbol{F}_{LD}}{m} \tag{3.3}$$

For simplicity, the vehicle's thrust acceleration response to a guidance command is modeled as zero-lag, i.e. perfect control. It assumes that the vehicle instantly and perfectly responds to the commanded $\boldsymbol{a}_T$, so the command from the guidance computer, after being put through a thrust magnitude limiter, is the same as the term in Equation 3.3. This is done to separate the study of Guidance laws from the performance of Control laws, as discussed in Section 1.1.

## 3.2 Guidance Computer

The guidance computer takes as input the current state as provided by the navigation system, the terminal constraints (final position and velocity in the case of E-Guidance, with final acceleration as well in the case of APDG), nominal rocket max thrust, current vehicle mass, time-to-go, and selection of guidance law.

The output is a throttle setting as a fraction of the nominal max thrust and a direction for the thrust acceleration $\boldsymbol{a}_T$. These values are unlimited at the level of the guidance computer, i.e. the throttle setting can be larger than 1 requiring thrust magnitude greater than $T_{max}$.

In this implementation, the throttle is limited after being passed out of the guidance computer to reflect the real performance constraint represented in Equation 2.15. Since the guidance computer only receives nominal $T_{max}$ upon which to base its throttle setting, the resultant command can be off due to engine performance dispersion.

The guidance command is updated periodically at a rate lower than the simulation rate. Here the guidance is refreshed at 5 Hz. It must also stop updating when $t_{go}$ becomes small due to its presence in the denominator for several entries in the E matrices of Equations 2.26 and 2.31. For the last half second the thrust acceleration is held constant. This has very little impact on landing precision.

The time-to-go computation takes place in the guidance module. The computation method is as described in 2.3.5. A gravity turn estimate is used to initialize the simulation, but it is reset upon ignition based on the current conditions and then allowed to run open-loop until simulation stop. The safety factor applied depends upon conditions. In vacuum, a factor of $1.2$ is necessarily applied to ensure soft landing when using the ignition optimization strategy. With atmospheric conditions the additional factor is unnecessary, as will be demonstrated in section 4.0.6.

The guidance solution requires specification of the terminal constraints. In both E-Guidance and APDG, the terminal constraints for position and velocity are specified in the same frame as the initial conditions. For the initial conditions listed in Table 3.1, the final position constraint is the origin. The simulation operates in a Mars centered frame in order to simplify gravitational calculations and model a spherical body easily, so the final position constraint seen by the guidance computer is $\boldsymbol{r}_f = [0, 0, R_M]^T$, where $R_M$ is the radius of Mars. The final velocity is chosen as 1 m/s in the down direction. For APDG the final thrust acceleration is given as Equation 2.30, where c is chosen to be 2. This more strongly enforces the final attitude constraint than a smaller scalar without imposing too strong a requirement to the detriment of propellant performance.

## 3.3 Ignition trigger

Ignition time is determined by checking the two criterion discussed in Section 2.4: required thrust acceleration magnitude $\boldsymbol{a}_{GT}$ and downrange distanced traveled by a gravity turn maneuver $s_{GT}$. The vehicle is started in atmosphere on a trajectory roughly in line with the landing site. As the vehicle travels along its trajectory with its engine off, these two criterion are constantly checked. Once either one is satisfied, the engine is ignited and the time-to-go estimate is updated as described in Section 3.2. Implementation of the ignition switch is performed in the simulation script by a simple flag that is checked upon guidance updates.

It is important to be certain that the criteria are met at some point in the flight. This strategy ensures that the ignition trigger is flipped in time as is discussed in Section 4.0.1. Gravity turn required thrust acceleration magnitude monotonically increases throughout the unpowered trajectory, and downrange distance traveled in a gravity turn maneuver decreases faster than the vehicle's range from landing site.

## 3.4 Navigation Module

The navigation module is fairly simplistic. It has two functions.

The first function is to produce navigation error in the form of normally distributed noise. The function takes the actual vehicle state $\boldsymbol{r}$ and $\boldsymbol{V}$, adds noise with a distribution specified in the simulation settings of the Monte Carlo script, and creates $\boldsymbol{r}_{nav}$ and $\boldsymbol{V}_{nav}$ vectors. The noise distribution is chosen to reflect realistic navigation error as provided by an inertial navigation system per the trade study presented in Moesser 2010 [10]. Typical standard deviations are 1 m position error and 1/3 m/s velocity error.

The second function is to filter the noisy navigation measurements to produce a smoother estimate of vehicle state. It does this using a simple low-pass filter as in Equation 3.4, where $\alpha$ is a chosen filter constant (here $\alpha = 0.3$), $\boldsymbol{x}$ is the current nav output of state, $\boldsymbol{x}_{prev}$ is the previous filtered state estimate, and $\boldsymbol{x}_{est}$ is the current filtered state estimate.

$$\boldsymbol{x}_{est} = \alpha\boldsymbol{x}_{prev} + (1-\alpha)\boldsymbol{x} \tag{3.4}$$

A Kalman filter would likely be used on a more sophisticated navigation system for a real aerospace vehicle, but given the simplicity of the navigation error and perfect knowledge of the noise distribution, modeling a Kalman filter would likely provide too good an estimate effectively canceling the effects of simulated navigation error.

## 3.5 Aerodynamic Model

The Aerodynamic model is comprised of Mars atmosphere data from (insert atmosphere data reference), and the vehicle model was developed by Cerimele et. al [1]

### 3.5.1 Atmosphere Model

The atmosphere model is based on NASA's Mars-GRAM [6], an engineering-level atmospheric model of the Martian atmosphere. The model used for this simulation is an empirically developed equation to represent the Martian atmosphere data within the mission's envelope. Mars-GRAM 2010 is based on NASA Ames Mars General Circulation Model for altitudes under 80 km.

For this simulation, the Mars-GRAM data is used to calculate a density $\rho$ and a speed of sound $V_{sound}$, from which Mach number is calculated as $|V|/V_{sound}$ and lift and drag are calculated as in Equations 3.6 and 3.5, where S is a reference area.

$$L = \frac{1}{2}\rho|V|^2 SC_L \tag{3.5}$$

$$D = \frac{1}{2}\rho|V|^2 SC_D \tag{3.6}$$

A particular feature of the powered descent phase is the change in drag calculation while the engine is ignited. When the guidance system is operating and the rocket is firing, the drag is modeled as half the value computed when the vehicle is unpowered, i.e. during glide. This is due to the rocket plume's aerodynamics. This change is accomplished in the simulation during the PDI phase by simply dividing the reference area by a factor of 2.

### 3.5.2 Lift and Drag

The vehicle aerodynamic model is based on the work of Cerimele et al [1].

The work of Cerimele er al provides a model of the lift and drag coefficients $C_L$ and $C_D$ based on vehicle Mach number and angle of attack. Vehicle angle of attack is determined directly from the thrust acceleration vector $\boldsymbol{a}_T$ and the vehicle velocity vector $\boldsymbol{V}$.

During the unpowered phase of approach when PDI is being optimized, the vehicle is held at a constant angle of attack of $55°$ to achieve maximum drag.

### 3.5.3 Orientation

The vehicle's body Euler angles are computed in a 3-2-1 sequence from the body axes. Since the simulation assumes perfect control, the body axes are computed as Equations 3.7, 3.8, 3.9. Pitch, yaw, and roll are then calculated as the angles between the body axes and the planet-fixed guidance frame $\hat{e}$. Angle of attack is similarly calculated using the body axes and the velocity vector. The CobraMRV's configuration determines that the thrust acceleration vector be in the direction of the vehicle yaw axis.

$$\boldsymbol{body}_Y = \boldsymbol{a}_T / ||\boldsymbol{a}_T|| \tag{3.7}$$

$$\boldsymbol{body}_P = \frac{\boldsymbol{r} \times \boldsymbol{body}_Y}{||\boldsymbol{r} \times \boldsymbol{body}_Y||} \tag{3.8}$$

$$\boldsymbol{body}_R = \boldsymbol{body}_Y \times \boldsymbol{body}_R \tag{3.9}$$

## 3.6 Monte Carlo

The Monte-Carlo script calls the PD Sim with simulation settings, vehicle parameters, initial condition, and navigation uncertainty. This allows each individual run to have dispersed conditions without modification of the code. The Monte-Carlo script allows choice of initial condition, fixed or dynamic, as well as specification of rocket parameter dispersion, initial condition dispersion in position and velocity, and navigation noise distribution.

The Monte-Carlo script is designed to allow testing of various scenarios and includes initial condition selection among a set of cases. The cases simulated here are points along a trajectory that passes nearly over the landing site, simulating the trajectory resulting from a typical re-entry and aerobraking glide phase. These cases also include initial velocity information that coincides with the vehicle having traveled unpowered along its trajectory. In fixed initial condition mode, the user specifies which initial conditions to test, the initial states are dispersed, and the simulations are run from the nominally fixed starting conditions. In ignition optimization mode, the Monte-Carlo script automatically chooses the nominal starting condition as the first point along this trajectory. It then disperses the starting point according to dispersion specifications and performs the number of runs specified by the user.

The dispersion factors are chosen to reflect realistic parameter dispersion. Navigation error is modeled using a normal distribution with standard deviations in accordance with Moesser's [10] work as described in Section 3.4. Rocket parameter dispersion is modeled as a uniform distribution of 2% of the parameter. Equation 3.10 shows this calculation for the maximum thrust dispersion, where $T_{max_{nom}}$ is the nominal maximum thrust expected by the guidance computer and $rand$ is a uniformly distributed random number between zero and one.

$$T_{max} = T_{max_{nom}} * (1 + 0.02 * (1 - 2 * rand)) \tag{3.10}$$

Each rocket parameter is modeled in this fashion. Only the nominal rocket parameters and dispersed navigation values can be known to the guidance computer, and the module design allows this distinction to be made easily. The simulation script calls the guidance computer with nominal max thrust and the guidance law outputs a throttle setting based on that nominal value, but the simulation script applies thrust limits based on the real dispersed values.

The separation between nominal and dispersed rocket parameters is also important to the PDI strategy, ideally performed within the guidance computer. The PDI required acceleration criterion is implemented as in Algorithm 2, but the guidance computer only has the nominal maximum thrust available to it. Similarly, it only has the dispersed navigation values available for the range criterion in Algorithm 3.

Initial condition values are normally distributed about their nominal values. The dispersion is applied such that three standard deviations in position dispersion is 1000 m from the mean, and three standard deviations in velocity dispersion is 10 m/s from the mean.

The Monte-Carlo script maintains repeatable randomness for testing. Each of the individual runs starts with a seed to the simulation environment's random number generation algorithm such that re-running the simulation with the same seed will produce precisely identical results. This pseudo-random quality allows investigation of particular runs and helps the software development cycle immensely. The run seeds are stored directly with the run result data in a table so that if later evaluation shows an oddity that should be examined, the run can quickly be replicated.

## 3.7 Initial Condition

The initial conditions are listed in Table 3.1. They are provided in the guidance frame $\hat{e}$, which is oriented such that the origin is at the final landing site, $\hat{\boldsymbol{e}}_x$ is pointed "East", $\hat{\boldsymbol{e}}_y$ is pointed "North", and $\hat{\boldsymbol{e}}_z$ is pointed "Up". The simulation itself runs in a Mars-centered Mars-fixed frame.

**Table 3.1. Initial Conditions for Mars Landing.**

| State | Case | $\hat{e}_x$ | $\hat{e}_y$ | $\hat{e}_z$ | Unit |
|---|---|---|---|---|---|
| $\boldsymbol{r}_0$ | 6 | 6079 | -30720 | 8685 | m |
| $\boldsymbol{V}_0$ | 6 | -121.0 | 644.1 | -64.82 | m/s |
| $\boldsymbol{r}_0$ | 5 | 5013 | 25170 | 8054 | m |
| $\boldsymbol{V}_0$ | 5 | -121.1 | 616.5 | -78.57 | m/s |
| $\boldsymbol{r}_0$ | 4 | 3947 | -19860 | 7305 | m |
| $\boldsymbol{V}_0$ | 4 | -120.9 | 589.6 | -91.67 | m/s |
| $\boldsymbol{r}_0$ | 3 | 2887 | -14790 | 6444 | m |
| $\boldsymbol{V}_0$ | 3 | -120.1 | 563.3 | -103.9 | m/s |
| $\boldsymbol{r}_0$ | 2 | 2359 | -12,340 | 5973 | m |
| $\boldsymbol{V}_0$ | 2 | -119.7 | 550.2 | -109.7 | m/s |
| $\boldsymbol{r}_0$ | 1 | 1832 | -9949 | 5478 | m |
| $\boldsymbol{V}_0$ | 1 | -119.8 | 537.0 | -115.4 | m/s |

These conditions represent an entry trajectory that is slightly off-target for the landing site and requires both powered deceleration to avoid overshoot as well as redirection to correct the angle error. The trajectory traveled from the furthest starting condition (Case 6) unpowered in a vacuum is plotted in Figure 3.1 along with the starting positions for Cases 1 through 6, indicated by an X. Similarly, Figure 3.3 shows the trajectory in atmosphere with the vehicle held at optimum angle of attack. Figure 3.2 shows the trajectory error in terms of ground range to landing site as the vacuum trajectory passes over at around 48 seconds.

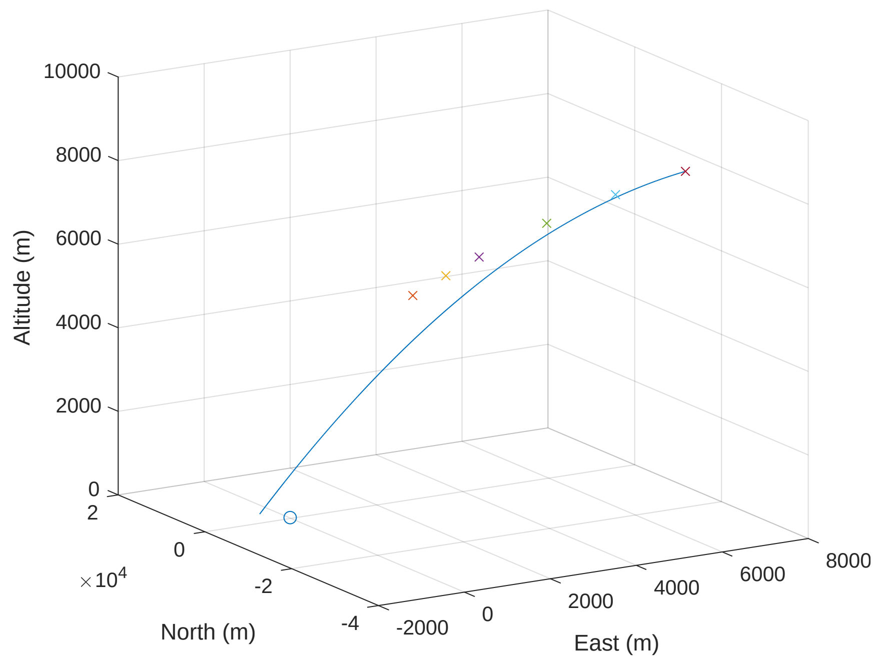


**Figure 3.1. Unpowered Initial Trajectory in Vacuum**

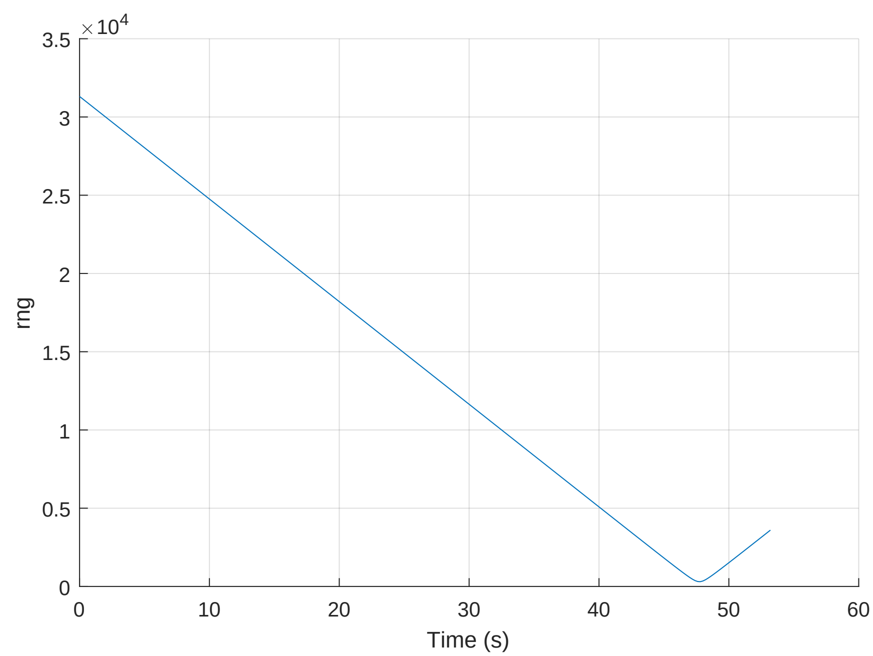


**Figure 3.2. Ground Range: Unpowered Initial Trajectory In Vacuum**

As is clear from Figures 3.3 and 3.4 by the placement of the initial conditions of the 6 cases, the initial conditions are chosen to represent the landing vehicle on aerodynamic approach making corrections toward the site. While employing the PDI strategy outlined in Section 2.4 the vehicle will travel along the atmospheric trajectory shown in Figures 3.3 and 3.4 until ignition.

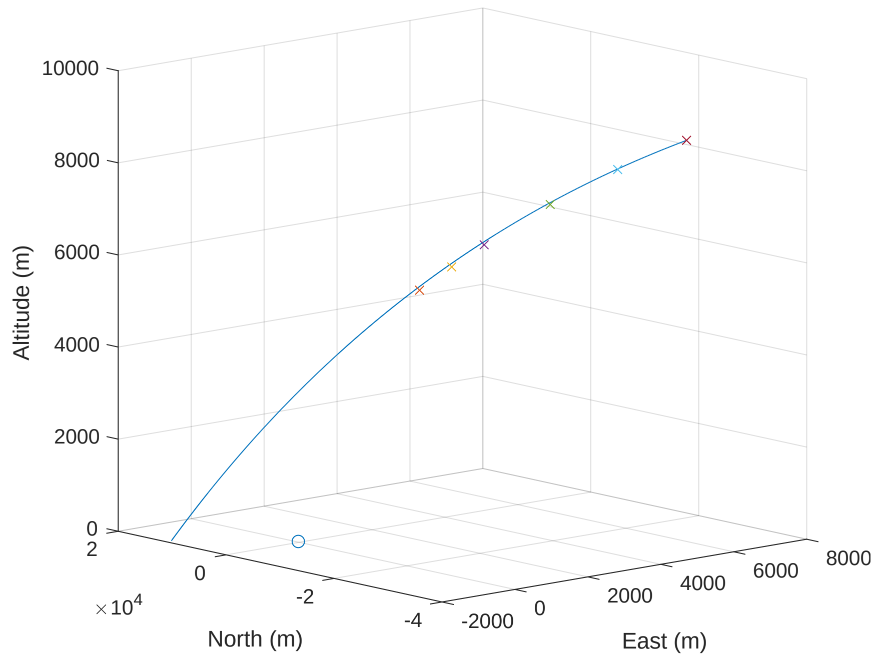


**Figure 3.3. Unpowered Initial Trajectory in Atmosphere**

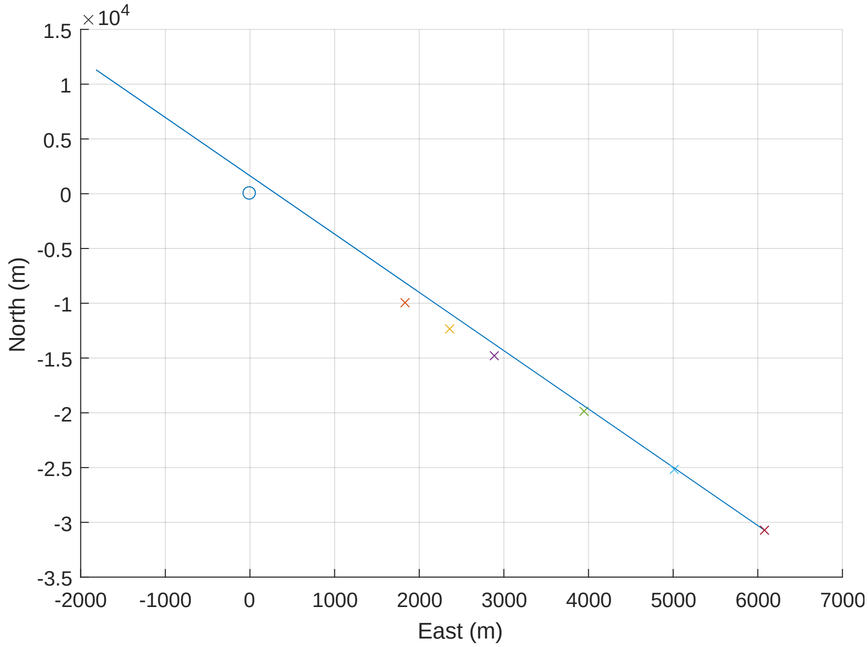


**Figure 3.4. Unpowered Initial Trajectory in Atmosphere From Above**

# CHAPTER 4
# RESULTS

In this chapter results are presented for several simulated conditions.

1. Nominal vs. dispersed parameters
2. Atmosphere vs. vacuum
3. Static initial conditions vs. dynamic PDI

The first condition, the effect of dispersion, is of interest on the topic of robustness. A closed-loop guidance law implementation is intended to ensure satisfaction of the problem constraints in practical conditions with uncertainty of state and performance limitations. A guidance system without periodic state feedback, i.e. an open-loop implementation, would be expected to quickly lose accuracy and ultimately fail in real application. Due to the criticality of safety in the intended mission, it is vital to show that this strategy performs well in nominal as well as dispersed conditions and to give a sense for the uncertainty in its performance.

The second condition is of particular interest because, as is clear in Section 2.3, neither APDG nor the PDI strategy take into account atmospheric effects and, at least in the case of the Apollo missions, have not been flown in the context of atmospheric landing. It is important to verify that the strategy can satisfy mission goals and still perform well in atmosphere, and the comparison with vacuum conditions serves both as a control and a fuel requirement baseline.

The third condition is the subject of this study. Whether dynamic PDI can reliably provide propellant efficiency improvement while maintaining safety and practicality is the topic of interest.

For dispersed cases the starting conditions are as listed in Table 3.1 with the addition of Case 7, which represents the active PDI strategy. This Case is flown from the initial condition specified by Case 6, the first point on the trajectory, but with the engine off until one of the PDI criteria defined in Equations 2 and 3 is met.

### 4.0.1 Adaptive PDI

The first result to investigate is the validity of the PDI criteria in Equations 2 and 3. For these criteria to serve as ignition criteria, they must reliably trigger in time to land the vehicle softly.

Figure 4.1 shows data from a vacuum case of APDG with PDI active during the unpowered phase, before PDI has been triggered. The required thrust acceleration magnitude $a_{GT}$for a gravity turn is shown monotonically increasing throughout the unpowered trajectory, and the required downrange travel $s_{GT}$ is monotonically decreasing. $a_{GT}$ and $s_{GT}$ reach their limits at almost the same time, but this would not be true of all trajectories.

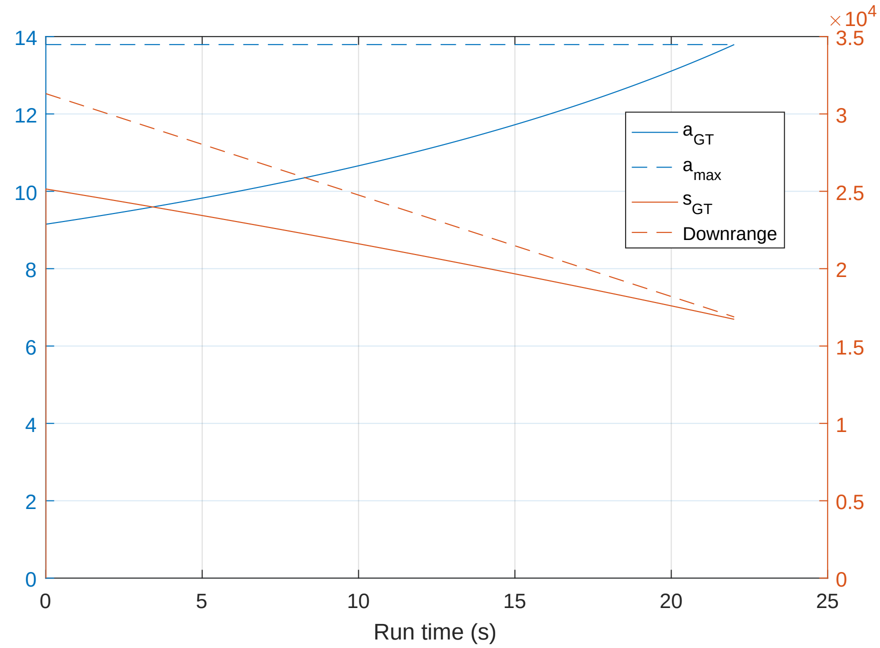


**Figure 4.1. PDI Criteria: Vacuum**

Figure 4.2 shows data from an atmospheric case of APDG with PDI active during the unpowered phase, before PDI has been triggered. Here the required thrust acceleration $a_{GT}$ is still monotonically increasing but not as quickly as it did in a vacuum, while $s_{GT}$ is decreasing similarly quickly. This behavior is examined in Section 4.0.4, where it is clear in Figures 4.10 and 4.12 that the vehicle is able to provide significant lift while decelerating through drag during the unpowered phase in atmosphere. Figure 4.2 clearly shows that the risk of overshoot, reflected in the downrange criterion, is much greater than exceeding the maximum thrust acceleration for this trajectory in atmosphere.

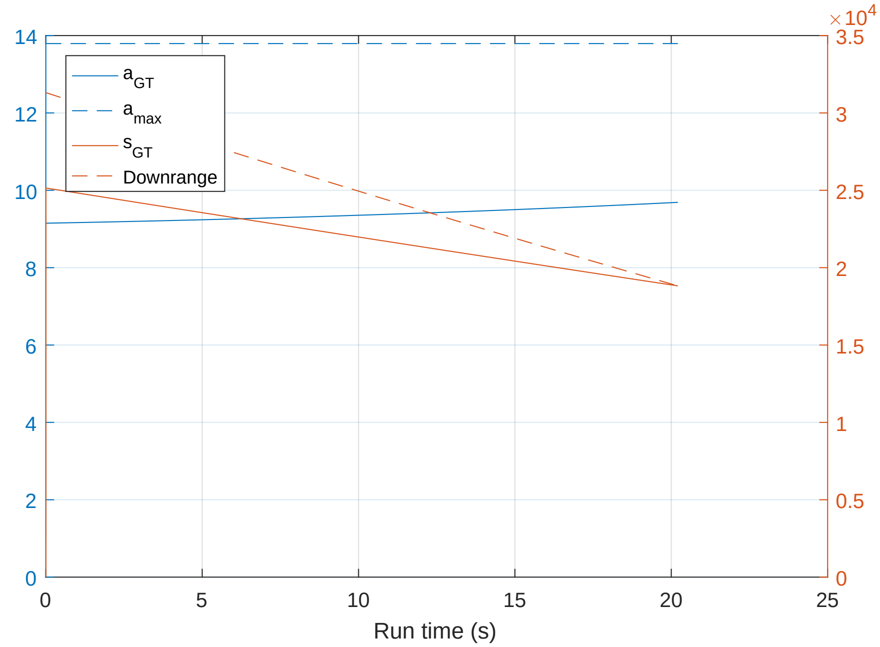


**Figure 4.2. PDI Criteria: Atmopshere**

### 4.0.2 Comparison of E-Guidance with APDG

Table and figures for the following condition: Both guidance laws Full dispersion (Rocket, IC, Nav) Atmospheric effects Dynamic Ignition Case Comparing guidance laws

The choice between E-Guidance of Equation 2.28 as flown on the Apollo missions and APDG of Equation 2.31 is examined below. Results are first presented as nominal performance in atmosphere in order to investigate the individual performance of each law without the effects of dispersion. Beyond this section, results will be presented using APDG.

Figures 4.3 through 4.8 show plotted quantities of a rocket in atmosphere employing the PDI strategy on approach to landing at the origin from Initial Condition Case 6 (Table 3.1). In Figure 4.3 the trajectory with the steeper approach upon landing is flown by the APDG guidance law.

The laws each result in similar trajectories as evidenced by Figures 4.3 through 4.8. However, the thrust and therefore speed profiles are notably different, with the APDG law starting at a lower magnitude of thrust in Figure 4.7 than the Simple law and their profiles roughly reversed. The pitch angle profiles in Figure 4.8 are very different, with the APDG law providing a final pitch angle much closer to the horizontal. Also clear in this plot is the guidance update stoppage in the last half second to a poorly conditioned system in Equation 2.31. This stoppage manifests as a constant pitch angle in Figure 4.8 because the thrust acceleration command is not updated, so the vehicle's orientation remains constant.

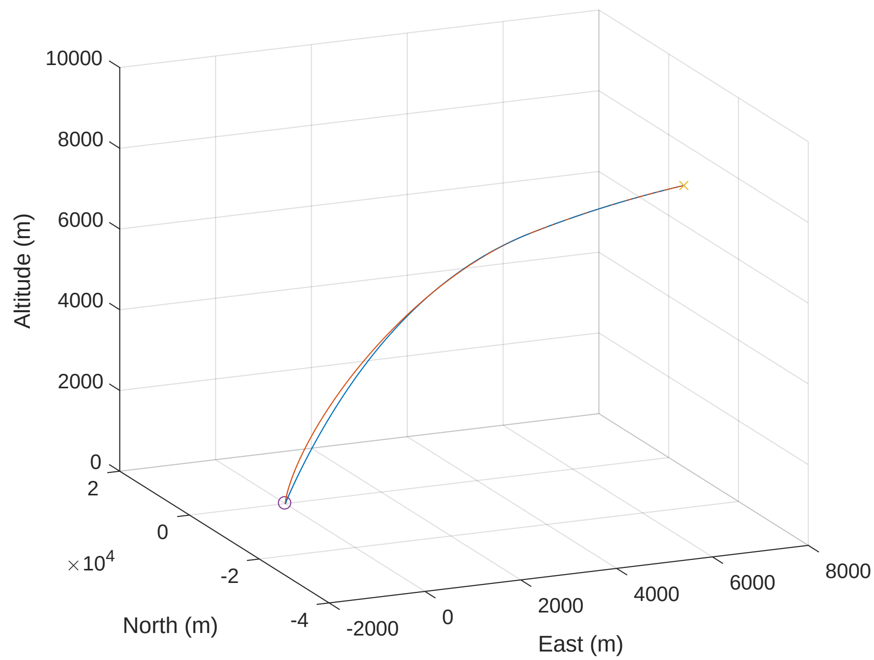


**Figure 4.3. Trajectory: E-Guidance vs. APDG**

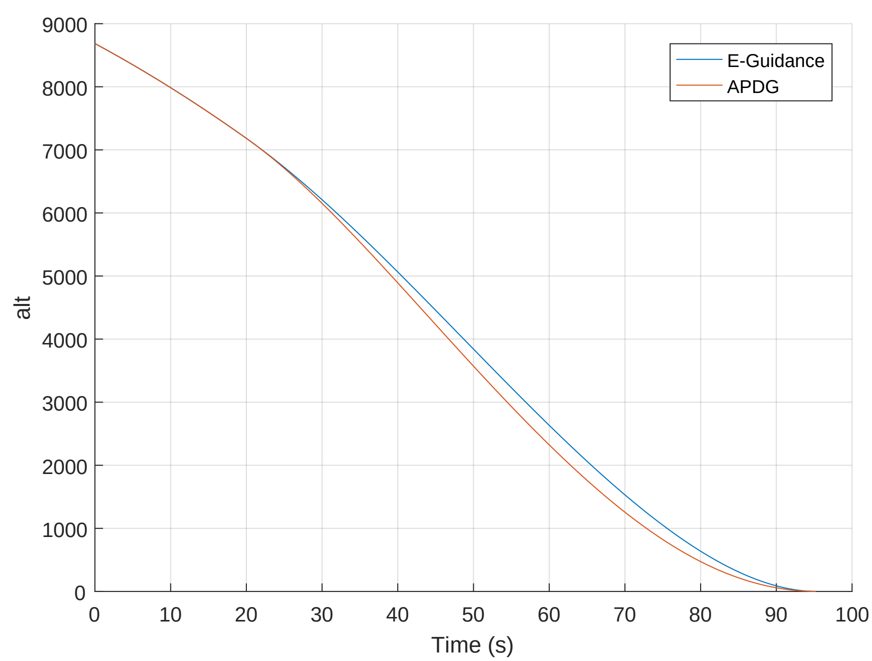


**Figure 4.4. Altitude: E-Guidance vs. APDG**

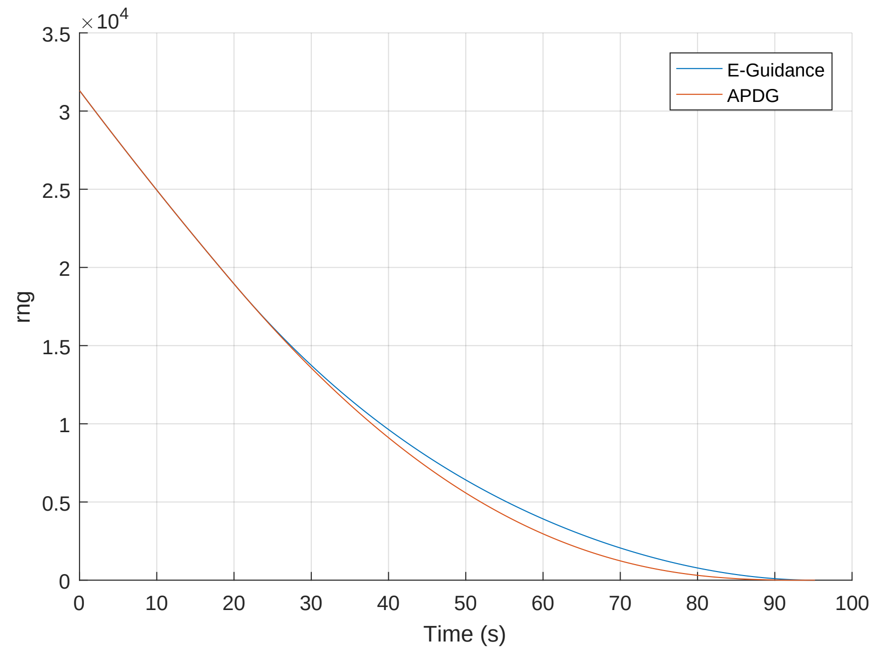


**Figure 4.5. Ground Range: E-Guidance vs. APDG**

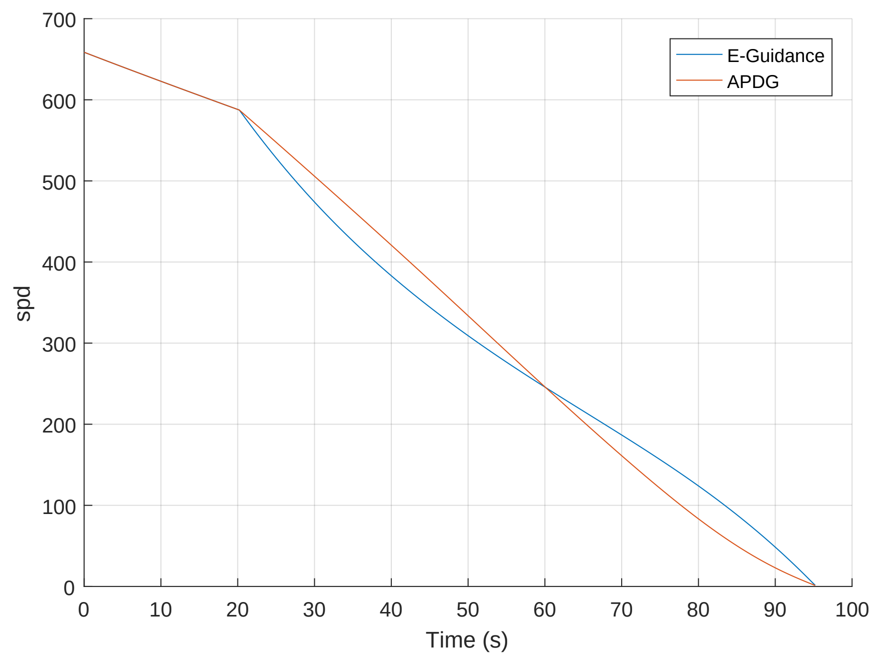


**Figure 4.6. Speed: E-Guidance vs. APDG**

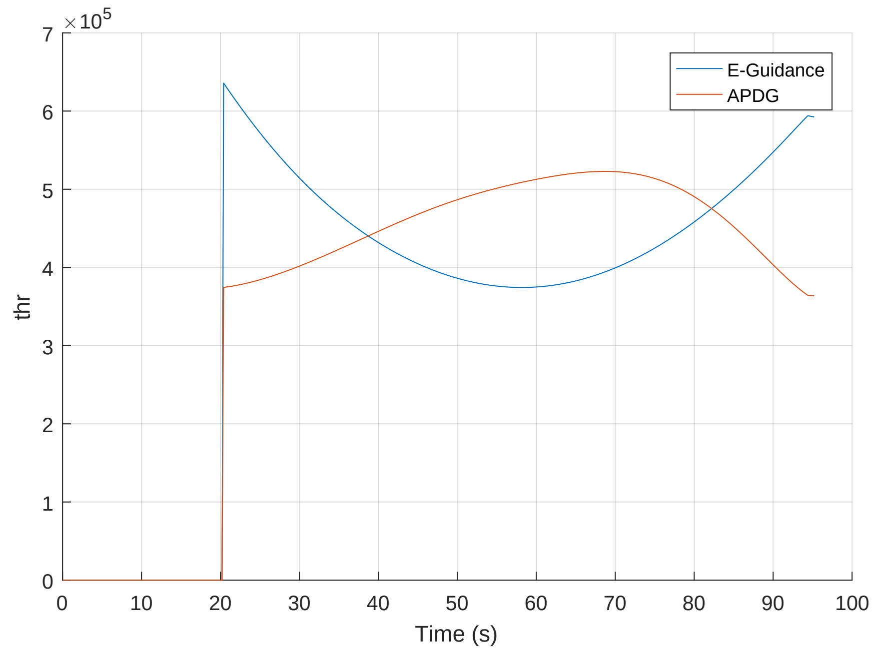


**Figure 4.7. Thrust Magnitude: E-Guidance vs. APDG**

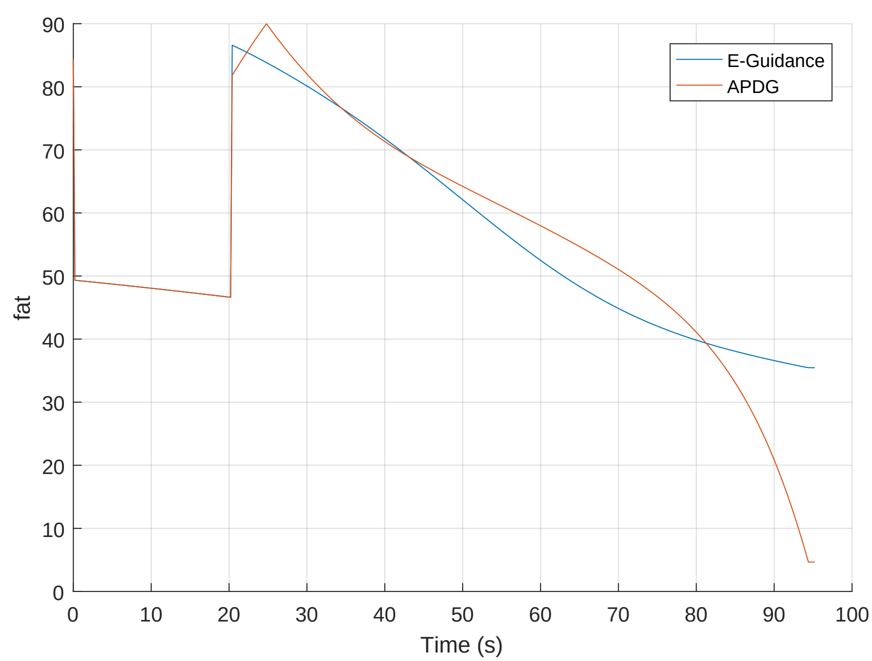


**Figure 4.8. Pitch: E-Guidance vs. APDG**

Including dispersion results in similar profiles which will be examined more closely in later sections, but a numerical comparison of the laws' performance is given in Table 4.1. Each law was in control for 1000 runs. Results are presented as mean values and standard deviations $\sigma$. Both laws perform similarly with the APDG law showing slightly higher speed on landing and using more fuel. This speed inaccuracy will be addressed in Section 4.0.3

**Table 4.1. Comparison of Performance of E-Guidance with APDG**

| Guidance Law | Runs | Fuel (kg) | Fuel $\sigma$ | Flight Time (s) | FT $\sigma$ | Range (m) | Range $\sigma$ | Speed (m/s) | Speed $\sigma$ |
|---|---|---|---|---|---|---|---|---|---|
| E-Guidance | 1000 | 9707.4 | 280.1 | 95.2 | 2.0 | 1.9 | 1.5 | 5.9 | 3.1 |
| APDG | 1000 | 9849.8 | 287.3 | 95.0 | 2.0 | 2.5 | 1.9 | 8.0 | 3.6 |

### 4.0.3 Navigation Error

Table and figures for the following condition: APDG Guidance Law Rocket and IC Dispersion Atmospheric effects Dynamic Ignition Case Comparing Navigation Error vs. No Nav Error

The final speeds shown in Table 4.1 are not very accurate given the terminal velocity constraint magnitude of $1\ m/s$. This constraint is important since it determines how softly the vehicle will land. However, this final speed (and range) inaccuracy is due almost entirely to the navigation dispersion model outlined in Section 3.4, as is clear from the results given in Table 4.2. Here the simulation was run with atmospheric effects, PDI, and full initial condition and rocket parameter dispersion. The cases differed only in whether navigation dispersion was active. The Nominal runs clearly show very accurate satisfaction of the terminal constraints. A higher fidelity model would include much lower navigation error near the landing site due to the availability of more accurate measurements through radar systems and short range devices and would result in similarly low inaccuracies. More sophisticated sensor models are not investigated here since they would only apply near the end of flight, resulting in little impact on fuel consumption.

**Table 4.2. Comparison of Performance With and Without Navigation Error**

| Nav | Runs | Fuel (kg) | Fuel $\sigma$ | Flight Time (s) | FT $\sigma$ | Range (m) | Range $\sigma$ | Speed (m/s) | Speed $\sigma$ |
|---|---|---|---|---|---|---|---|---|---|
| Dispersed | 1000 | 9849.8 | 287.3 | 95.0 | 2.0 | 2.5 | 1.9 | 8.0 | 3.6 |
| Nominal | 1000 | 9761.8 | 270.8 | 95.2 | 2.0 | 0.1 | 0.0 | 1.0 | 0.0 |

### 4.0.4 Vacuum vs. Atmosphere

Here APDG's performance is compared with and without aerodynamic effects. This is an important comparison to make because, as stated in Section 2.3.1, the guidance law is derived without consideration of aerodynamic effects. There is no guarantee it will perform well in atmosphere. Only nominal results will be presented here, as the dispersed results are presented in Sections 4.0.5 and 4.0.6 in the context of adaptive powered descent initiation (PDI) vs. predetermined ignition time.

Figures 4.9 through 4.14 show nominal profiles for the APDG law in both atmosphere and vacuum. These trajectories start from Case 6 as listed in Table 3.1. Figures 4.9 and 4.10 show that in atmosphere, the PDI strategy with the vehicle held at optimum angle of attack results in higher lift, meaning that the vehicle does not lose altitude as quickly as in vacuum, while Figure 4.12 shows it losing speed much more quickly during this phase. This suggests that the aerodynamic effects have an appreciable effect on the energy ultimately required for powered descent and soft landing.

Figures 4.13 and 4.14 suggest the same conclusion. Upon ignition, the thrust required in atmosphere is far lower than that required in vacuum, and the ship mass stays higher in atmosphere throughout most of the trajectory, particularly the ending mass, reflecting lower fuel consumption. The thrust profile in atmosphere suggests a fairly conservative ignition; since more energy is burned off through aerodynamic effects, less thrust is required than might be expected by the gravity turn solution which generates the operative time-to-go. This leaves appreciable margin for error, which is explored further in the dispersed results of Secion 4.0.6.

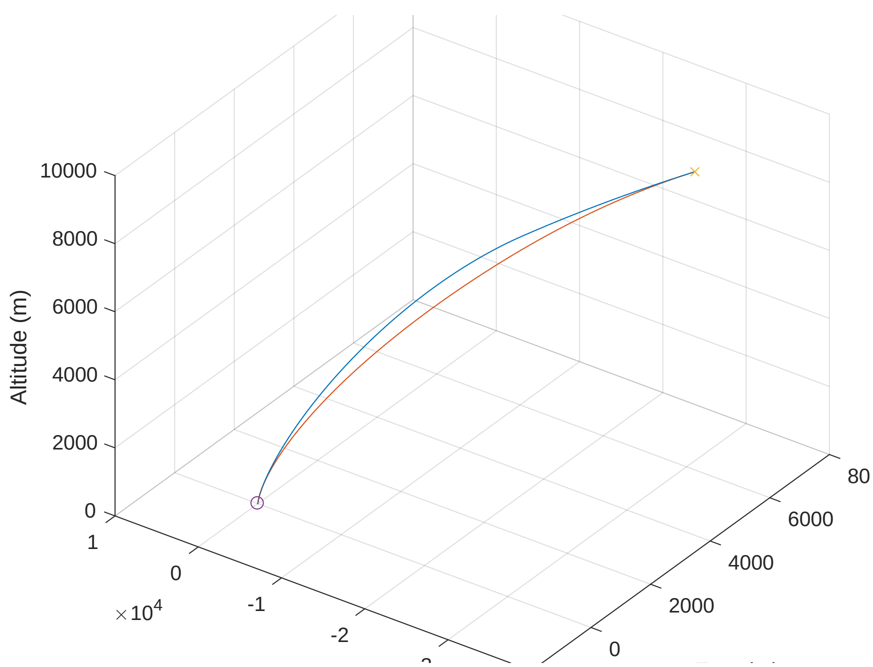


**Figure 4.9. Trajectory: Vacuum vs. Atmosphere**

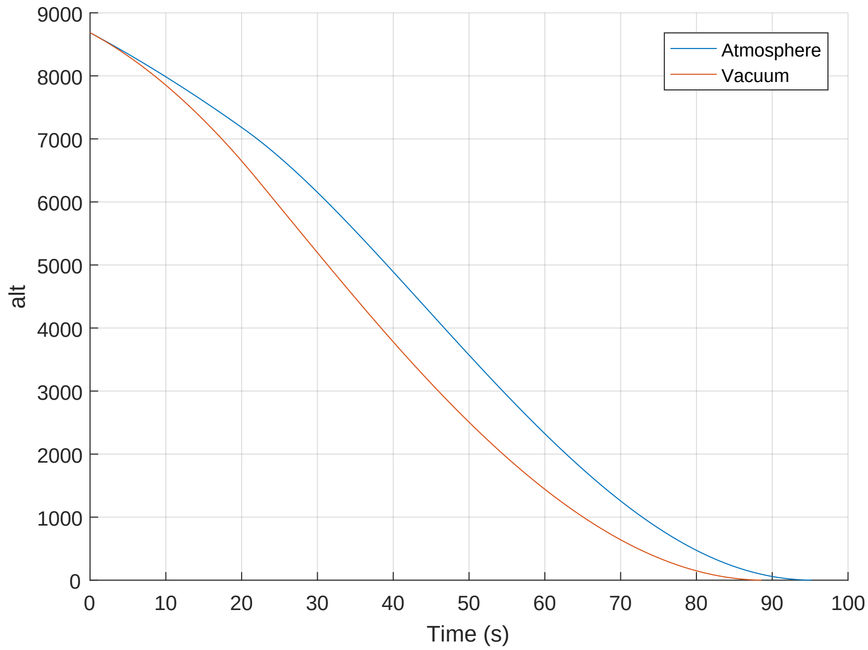


**Figure 4.10. Altitude: Vacuum vs. Atmosphere**

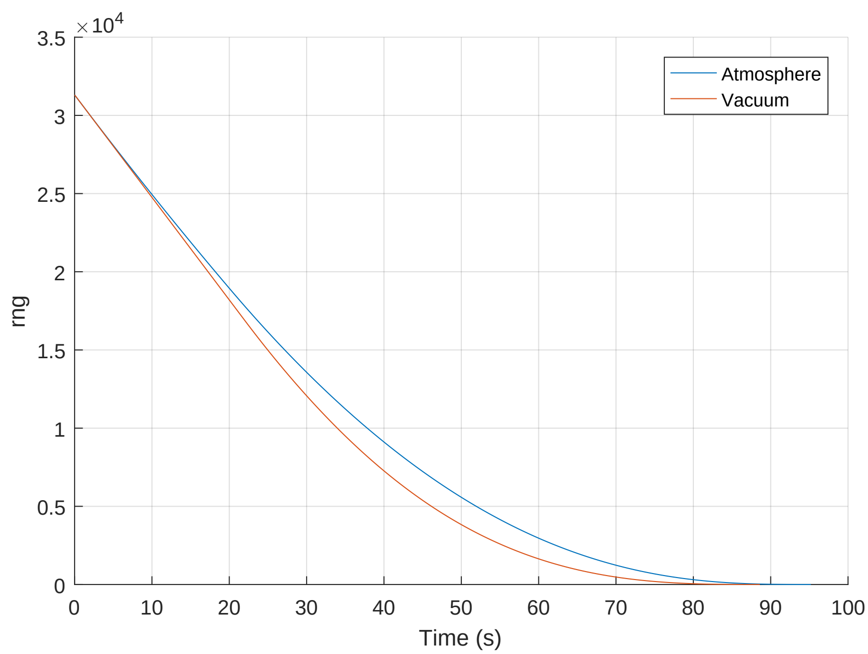


**Figure 4.11. Ground Range: Vacuum vs. Atmosphere**

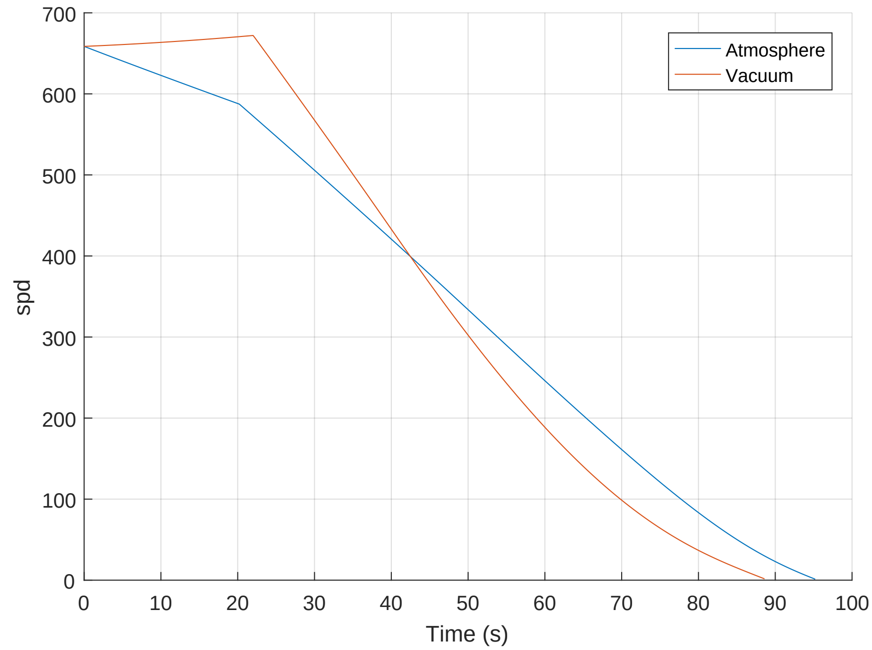


**Figure 4.12. Speed: Vacuum vs. Atmosphere**

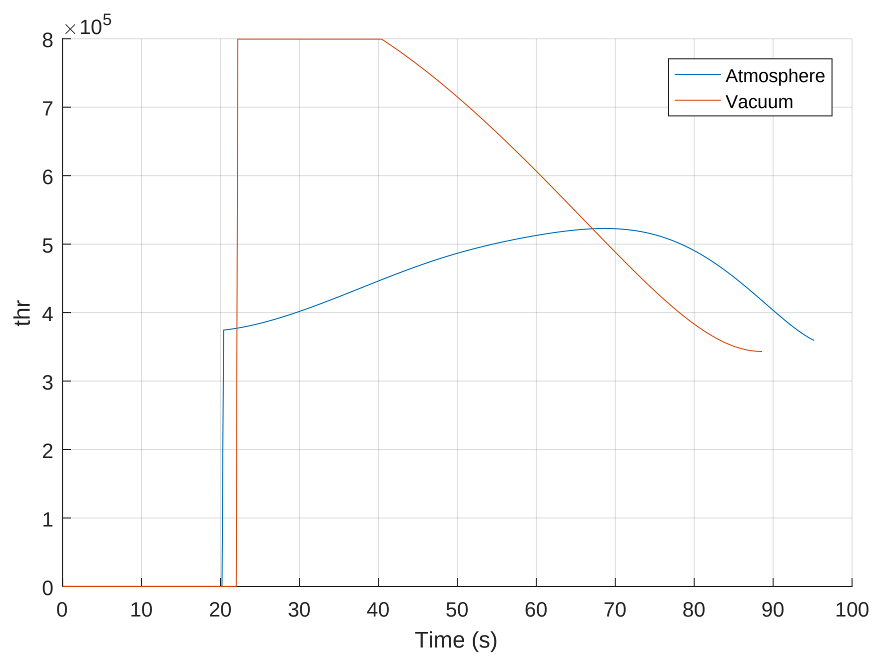


**Figure 4.13. Thrust Magnitude: Vacuum vs. Atmosphere**

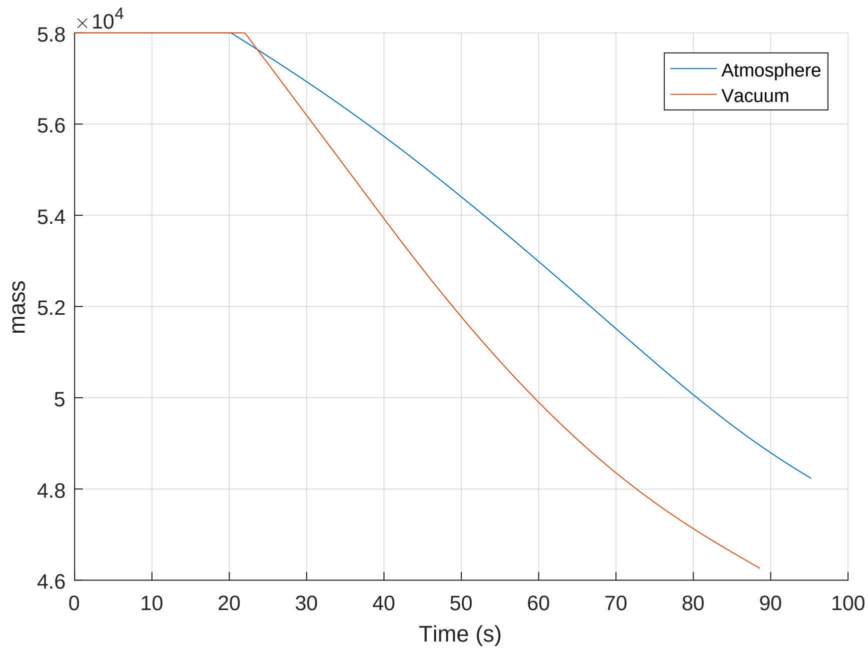


**Figure 4.14. Ship Mass: Vacuum vs. Atmosphere**

### 4.0.5 Vacuum Performance

Presented here are results for the vacuum condition, comparing the performance and robustness of APDG with adaptive PDI against specified ignition time in the form of predetermined initial conditions for a mission without aerodynamic effects. This is similar to the conditions experienced by the Apollo missions, though the simulation's parameters reflect a Martian gravitational field. As discussed in Section 3.2, there is a $1.2$ safety factor applied to the time-to-go as provided by Equation 2.32 upon ignition for all cases in vacuum.

Figures 4.15 through 4.19 show nominal results for all cases. This gives a baseline from which to evaluate the typical profiles of each Case and the PDI strategy.

Figure 4.15 shows significant landing site overshoot on Case 1. The plot of Ground Range vs. Time in Figure 4.18 tells a similar story, with the ground range nearly reaching zero before increasing at around 30 seconds before settling back toward zero. Figure 4.19 is the most telling plot, with the Thrust Magnitude profiles for each case plotted vs. time. Cases 4 through 6 do not saturate at any point in their trajectories, either at the lower or upper bounds of thrust. Cases 1 through 3 do however, with Case 1 saturating at the upper bound for a large portion of its trajectory.

Case 7 representing the PDI strategy also saturates its thrust for part of its trajectory after ignition but settles out toward the middle of the range by the end of its flight. Its other plots are unique relative to the other cases as well, with Speed in Figure 4.16 showing the characteristic increase in speed due to its ballistic trajectory until powered descent initiation unlike all other Cases. Its Altitude in Figure 4.17 starts in line with Case 6 but decreases more quickly and ends on a similar path as Case 4. Its Ground Range behaves similarly in Figure 4.18. Its trajectory in Figure 4.15 stands out in red, clearly following a much different path than any of the other Cases.

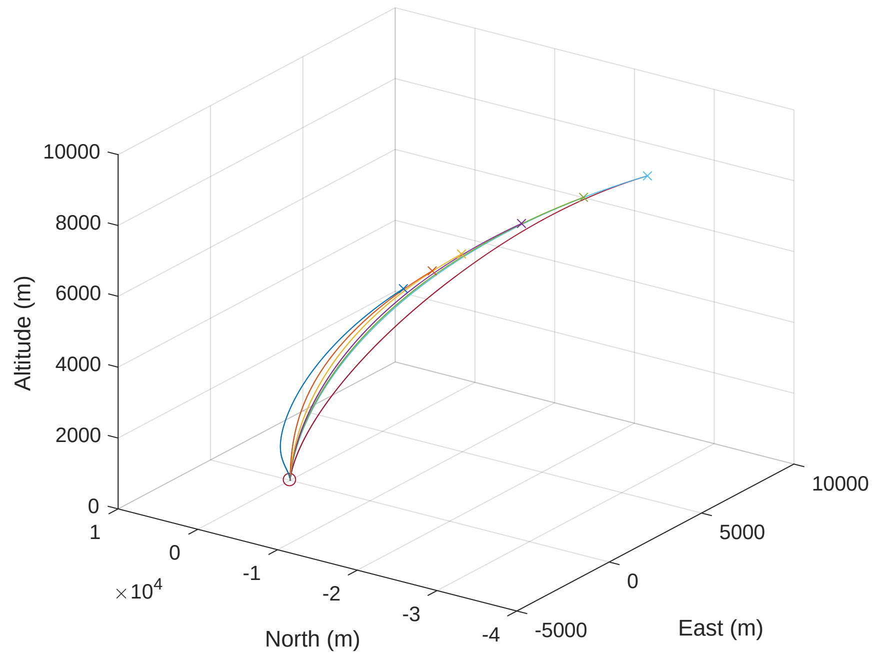


**Figure 4.15. Trajectory: APDG in Vacuum**

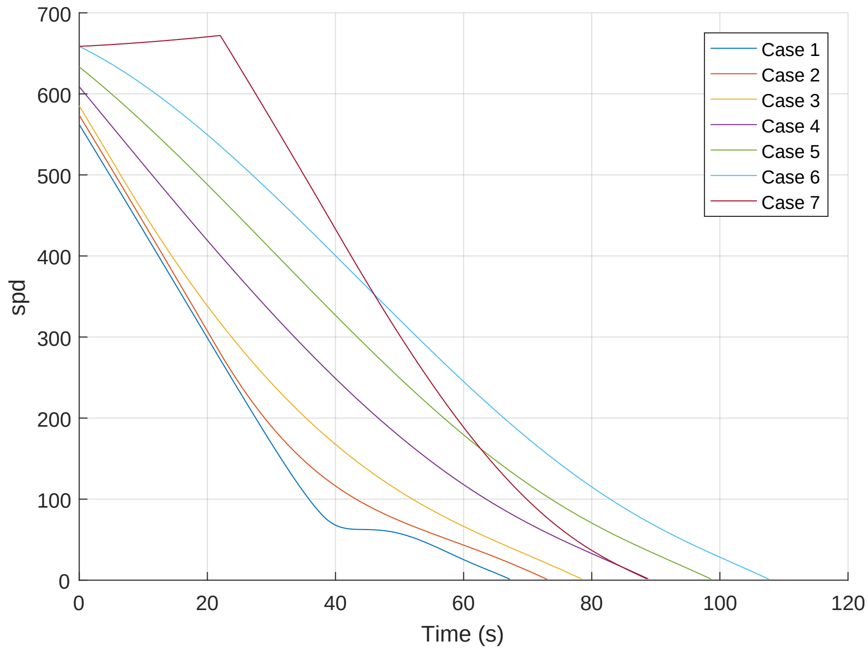


**Figure 4.16. Speed: APDG in Vacuum**

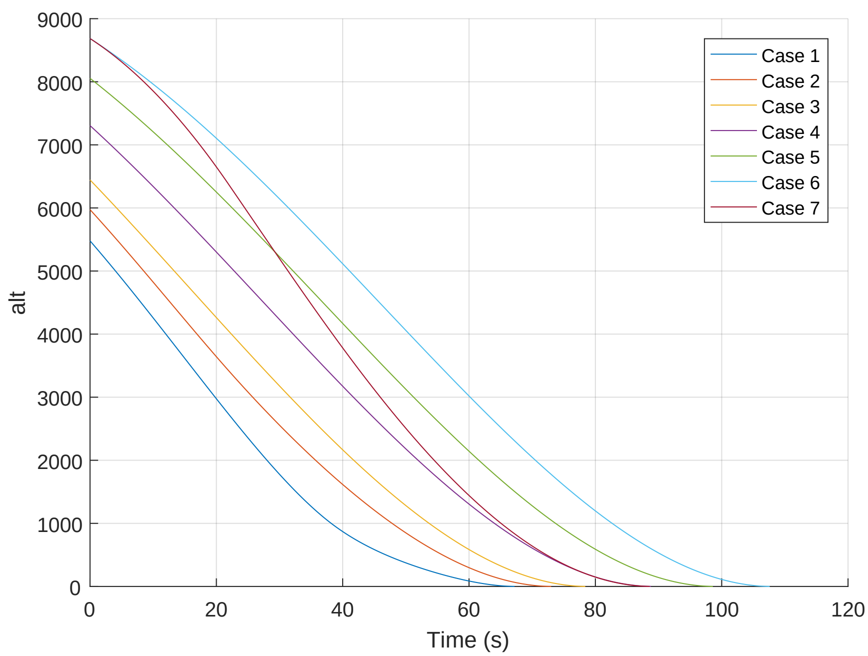


**Figure 4.17. Altitude: APDG in Vacuum**

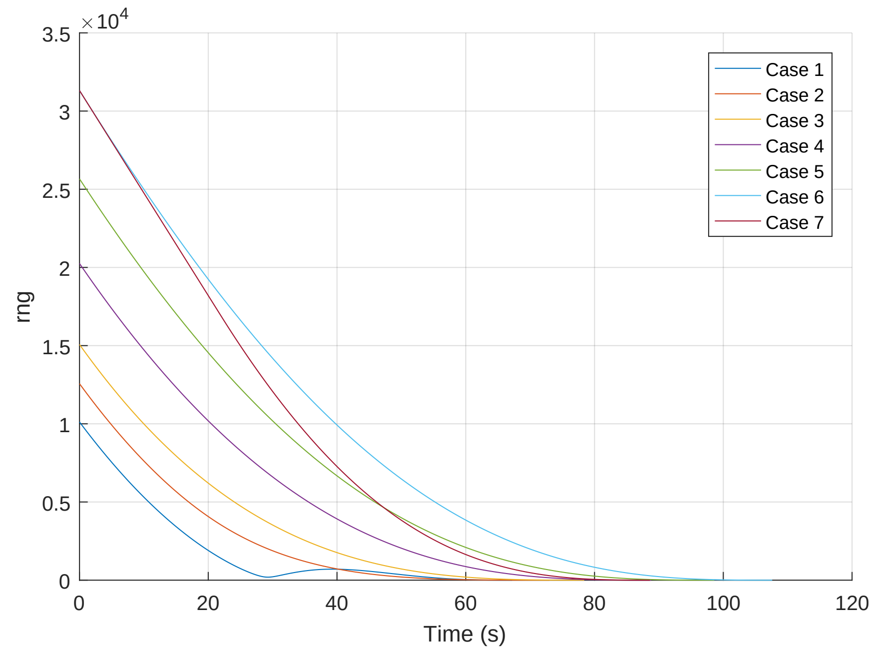


**Figure 4.18. Ground Range: APDG in Vacuum**

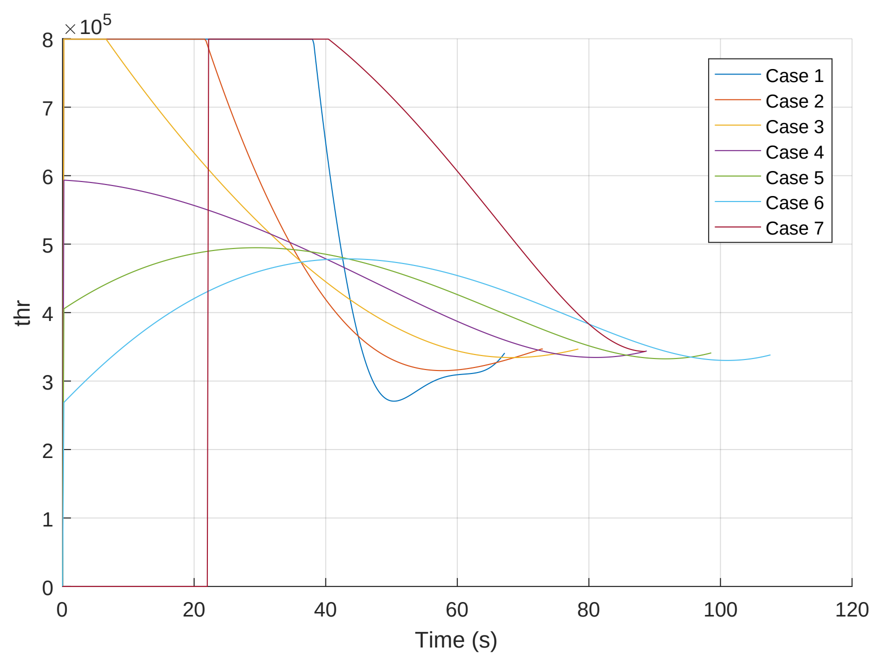


**Figure 4.19. Thrust Magnitude: APDG in Vacuum**

Introducing dispersion gives the results in Table 4.3. Most of the Cases perform similarly in terms of deviation from the mean aside from Case 1. Case 1 shows much higher fuel consumption deviation, flight time deviation, final range deviation, and final speed deviation.

These deviations are due to failed landings; fully 45 of the 1000 runs of Case 1 in vacuum failed to land softly. All other runs had final ranges below 16 m, and all 45 of these failed landings had final ranges of over 600 m. Similarly, the 45 failed landings had final speeds in excess of 90 m/s, compared to the 16 m/s for all other Case 1 runs. Since the standard deviations of the other successful cases are in line with the errors expected from Navigation uncertainty in Section 4.0.3, 16 m/s can be considered a bound on the error due to Navigation uncertainty, so these 45 landings are clearly cases where the guidance system failed.

None of the other cases had any such landing failures.

The lowest mean propellant consumption is given by Case 2, as well as the highest standard deviation in propellant consumption aside from Case 1.

**Table 4.3. Performance of PD Guidance In Vacuum**

| Case | Runs | Fuel (kg) | Fuel $\sigma$ | Flight Time (s) | FT $\sigma$ | Range (m) | Range $\sigma$ | Speed (m/s) | Speed $\sigma$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1000 | 11650.1 | 605.8 | 66.0 | 5.7 | 54.3 | 244.5 | 11.5 | 15.8 |
| 2 | 1000 | 11184.7 | 308.2 | 72.7 | 3.4 | 2.5 | 2.0 | 8.3 | 3.7 |
| 3 | 1000 | 11264.0 | 264.9 | 78.3 | 3.1 | 2.5 | 2.0 | 8.0 | 3.7 |
| 4 | 1000 | 11634.4 | 247.0 | 88.9 | 3.1 | 2.6 | 2.0 | 8.3 | 3.8 |
| 5 | 1000 | 12032.7 | 259.6 | 98.6 | 3.1 | 2.7 | 2.1 | 8.5 | 3.7 |
| 6 | 1000 | 12436.7 | 241.9 | 107.7 | 3.0 | 2.5 | 2.1 | 8.3 | 3.8 |
| 7 | 1000 | 11885.2 | 257.9 | 90.1 | 3.7 | 2.7 | 2.0 | 8.4 | 3.7 |

### 4.0.6 Atmospheric Performance

Presented here are results for the atmospheric condition, comparing the performance and robustness of APDG with adaptive PDI against specified ignition time in the form of predetermined initial conditions for a mission with aerodynamic effects. This is representative of a Martian mission, the intended purpose of the CobraMRV. As discussed in Section 3.2, there is no safety factor applied to the time-to-go as provided by Equation 2.32 upon ignition for any case in atmosphere.

Figures 4.20 through 4.24 show nominal trajectories from each of the 6 Cases outlined in Table 3.1 as well as Case 7 representing the adaptive PDI strategy from the starting condition of Case 6. Figures 4.21 and 4.23 show the behavior discussed in Section 4.0.4 in which PDI's unpowered glide retains altitude while shedding speed. Figure 4.23 also shows odd behavior in Case 6, explained easily by Figure 4.24 in which the thrust saturates on the lower bound. Figure 4.24 shows PDI operating in the middle of the thrust range throughout the trajectory after ignition.

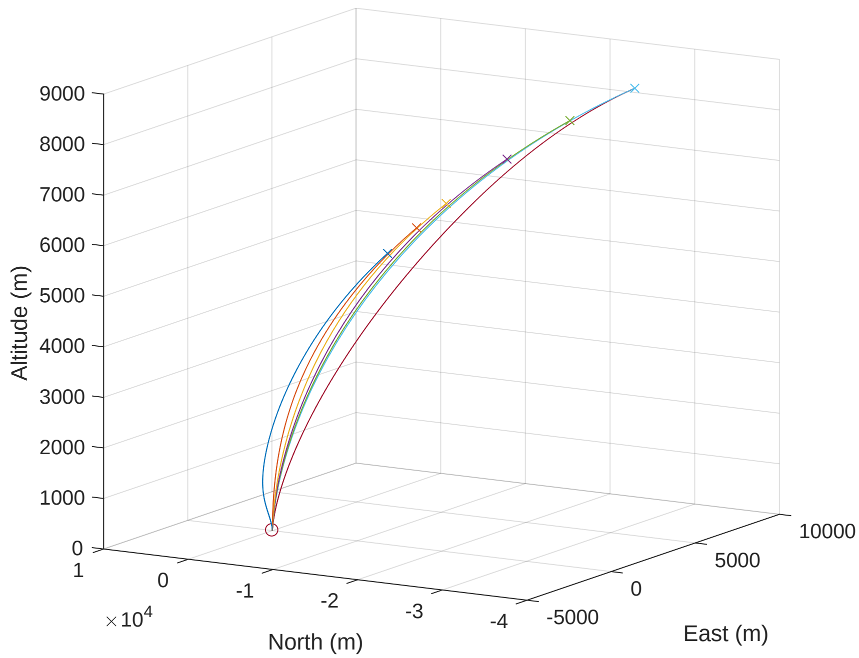


**Figure 4.20. Trajectory: APDG in Atmosphere**

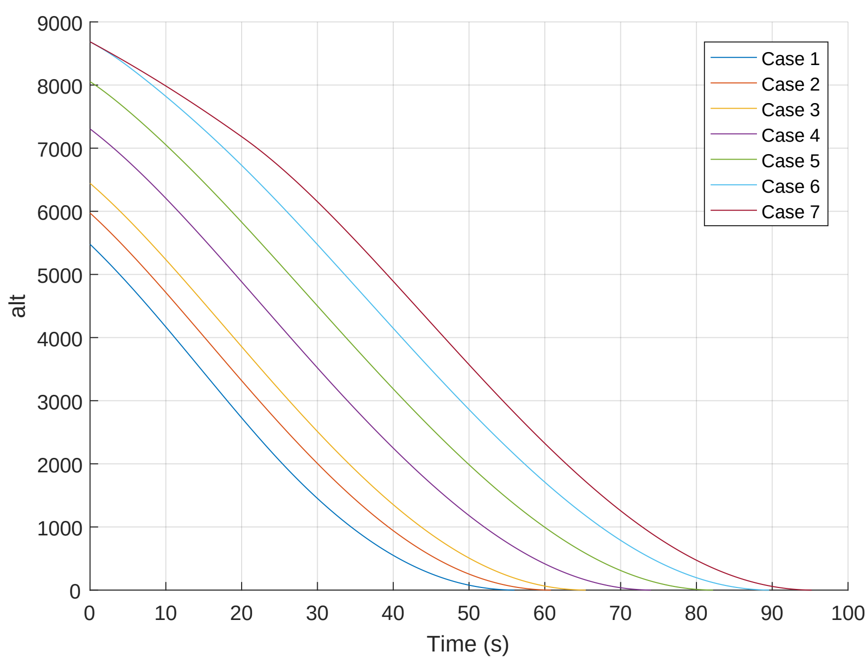


**Figure 4.21. Altitude: APDG in Atmosphere**

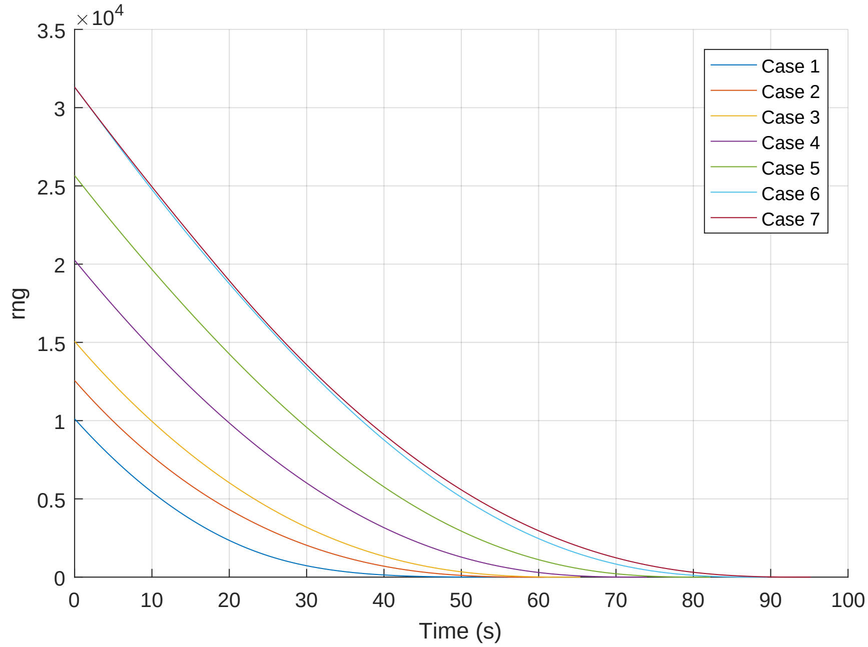


**Figure 4.22. Ground Range: APDG in Atmosphere**

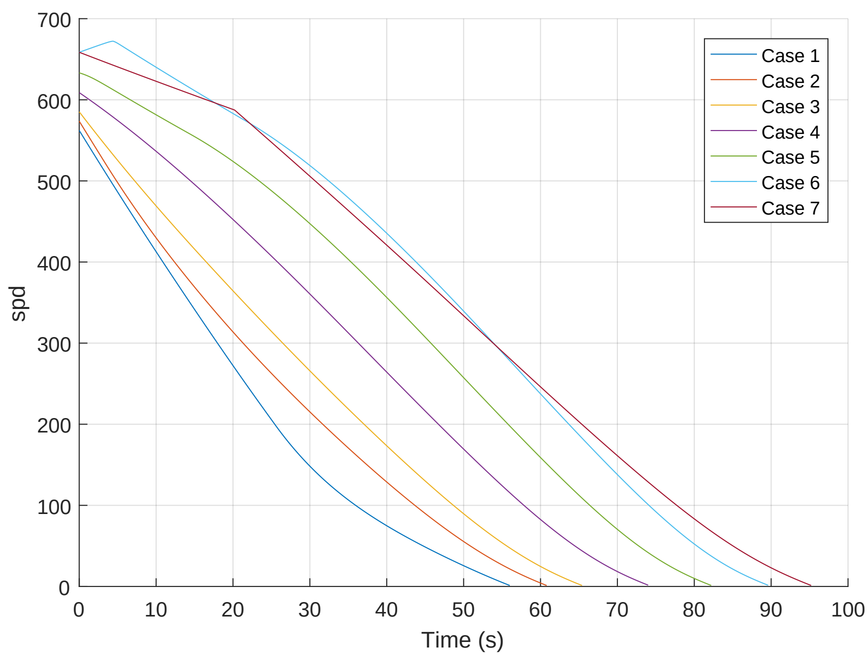


**Figure 4.23. Speed: APDG in Atmosphere**

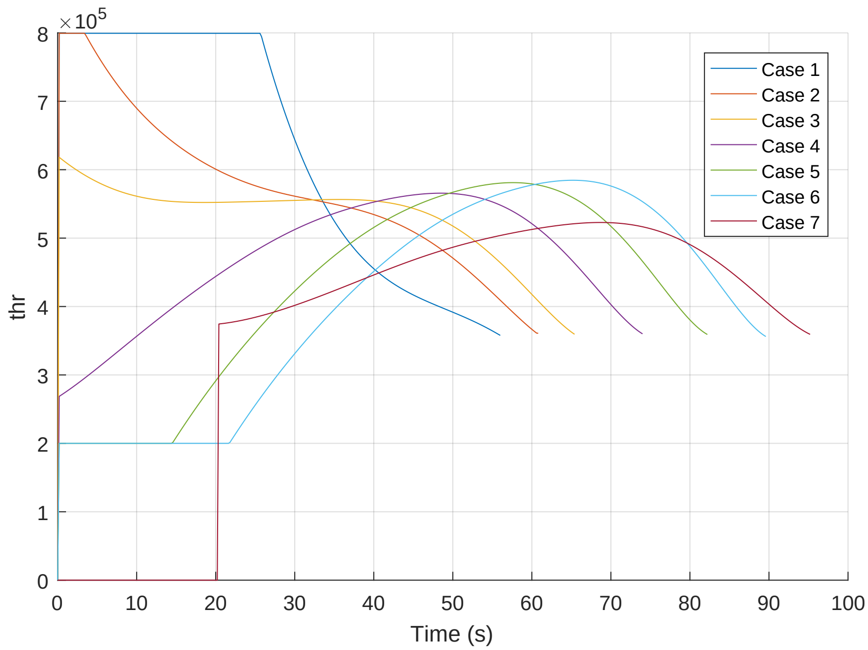


**Figure 4.24. Thrust Magnitude: APDG in Atmosphere**

Introducing dispersion produces Table 4.4. Again we see Case 1 showing failed landings. In atmosphere only 13 landings of 1000 failed, with range misses in excess of 152 m and speeds in excess of 32 m/s. These figures are lower than in vacuum.

PDI shows the best propellant performance in atmosphere, and no landing failures. Its final range and speed accuracy is well in line with the other Cases and the error introduced by Navigation uncertainty discussed in Section 4.0.3.

**Table 4.4. Performance of PD Guidance With Aerodynamic Effects**

| Case | Runs | Fuel (kg) | Fuel $\sigma$ | Flight Time (s) | FT $\sigma$ | Range (m) | Range $\sigma$ | Speed (m/s) | Speed $\sigma$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1000 | 10141.2 | 357.5 | 55.8 | 3.2 | 9.1 | 60.2 | 8.8 | 7.4 |
| 2 | 1000 | 9947.4 | 267.2 | 60.7 | 2.6 | 2.5 | 1.9 | 8.0 | 3.6 |
| 3 | 1000 | 9918.5 | 264.8 | 65.4 | 2.8 | 2.6 | 1.9 | 8.2 | 3.5 |
| 4 | 1000 | 9900.6 | 244.9 | 74.0 | 2.5 | 2.6 | 2.0 | 8.1 | 3.5 |
| 5 | 1000 | 10010.2 | 217.6 | 82.0 | 2.7 | 2.5 | 1.9 | 8.1 | 3.5 |
| 6 | 1000 | 10413.7 | 210.5 | 89.5 | 2.6 | 2.5 | 1.9 | 8.1 | 3.5 |
| 7 | 1000 | 9849.8 | 287.3 | 95.0 | 2.0 | 2.5 | 1.9 | 8.0 | 3.6 |

# CHAPTER 5
# DISCUSSION

The results in Chapter 4 have many implications on the guidance approach of APDG as well as the adaptive powered descent initiation strategy of Section 2.4. They will be discussed here, as well as the validity of the results themselves.

## 5.1 Simulation Error Analysis

As discussed in Section 3.1, the numerical integration technique employed is a Runge-Kutta $4^{th}$ order method. This method is particularly attractive in computationally expensive applications because the total accumulated error due to time integration reduces on the order $O(\Delta t^4)$ where $\Delta t$ is the time integration step size, while the computational expense increases on the order $O(\Delta t)$. This means that a larger step size can produce the same order of error as a simulation using a first order scheme, saving significant computational and engineering time resources. There is a limit to the error reduction in that any finite computation produces machine error in rounding and variable storage. This simulation was run on a system with a machine error $\epsilon \approx 2 * 10^{-16}$.

To trust the validity of a simulation's results, it must be shown that the simulation is running within the numerical integration scheme's asymptotic convergence. In essence, the error should demonstrate the expected order behavior.

A $4^{th}$ order scheme should show this error behavior. However, Figure 5.1 clearly shows a linear error behavior, with error reducing with order $O(\Delta t)$. Error of order $O(\Delta t)$ and $O(\Delta t^4)$ is shown for reference. This figure was generated by testing a particular moment in the simulation near the end of the trajectory without dispersion, with atmospheric effects and powered descent guidance. The baseline against which the error is measured is the finest resolution, i.e. the smallest time step, $\Delta t = 2^{-13} \approx 1.2 * 10^{-4}$. Running the same test with an explicit $1^{st}$ order Euler scheme produces a similar result.

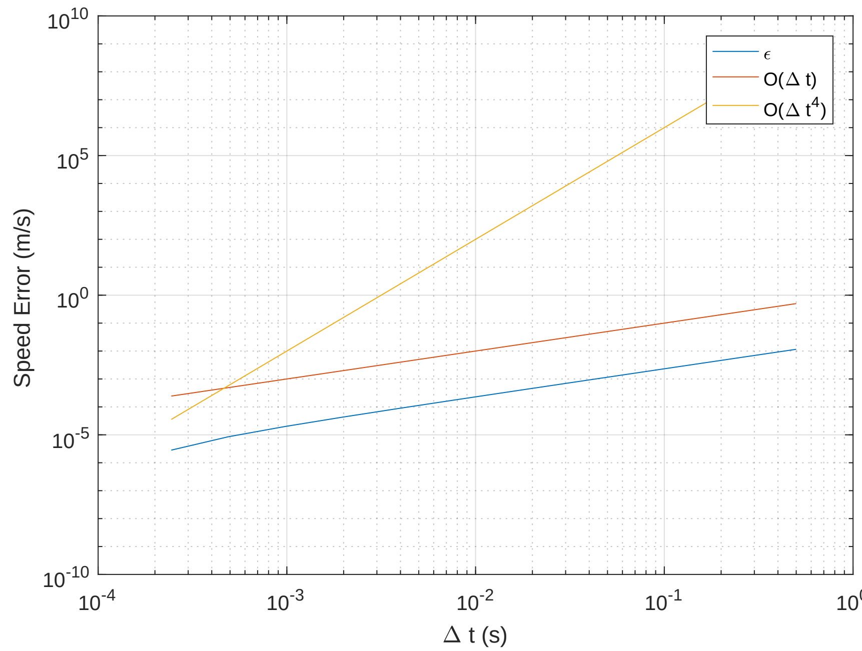


**Figure 5.1. INSERT FIGURE CAPTION**

The reason is that the derivative cannot be calculated at each intermediate step required for the RK4 scheme. Specifically, the terms $\boldsymbol{a}_T$ and $\frac{\boldsymbol{F}_{LD}}{m}$ representing the thrust acceleration and the aerodynamic effects in Equation 3.3 must be held constant for the intermediate derivatives. These terms are actually also functions of the state $\boldsymbol{x}$, but they are highly non-linear. The thrust acceleration term is often held actually constant between steps since the guidance update rate is lower than the time integration rate, but every 200 time steps this assumption fails and error accumulates. Similarly, the aerodynamic model is never actually constant between time steps.

Holding terms which are functions of the state $\boldsymbol{\phi}$ constant between time steps reduces the integration method to Explicit Euler, given in Equation 5.1.

$$\boldsymbol{\phi}_{n+1} = \boldsymbol{\phi}_n + \Delta t \boldsymbol{f}(t_n, \boldsymbol{\phi}_n) \tag{5.1}$$

If these terms are removed from the simulation, representing vacuum conditions with an unpowered trajectory, the error behavior is significantly different as demonstrated in Figure 5.2. Here we see that the magnitude of the total error is drastically lower, a factor of $10^-7$ times the atmospheric error, yet no longer decreasing with $\Delta t$. This is because all remaining error in the solution is due to machine error for any practical time step size. Testing larger time steps would lead to unrealistic guidance update rates, making the simulation invalid or impractical.

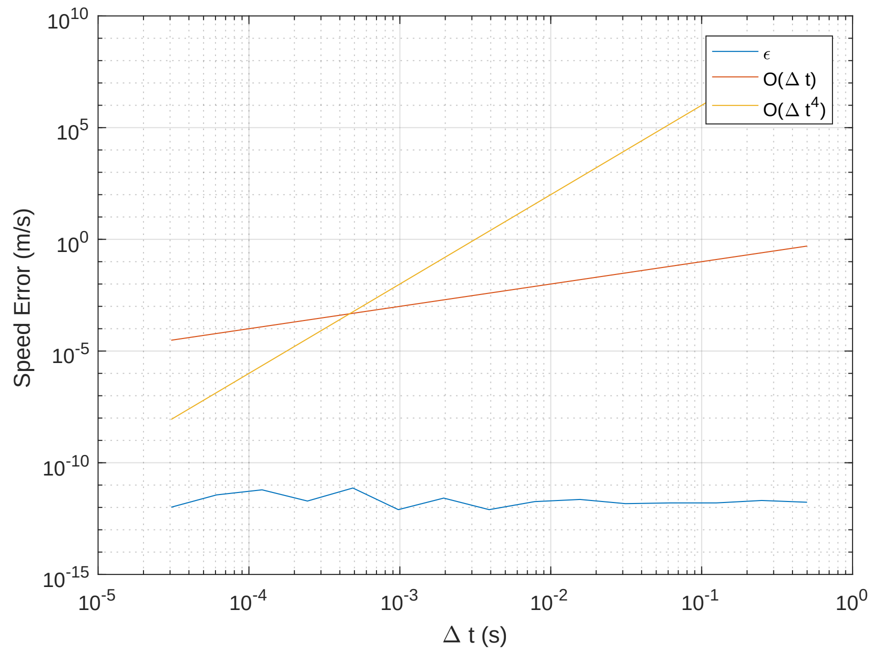


**Figure 5.2. INSERT FIGURE CAPTION**

However, since the simulation error is dominated by the atmospheric and thrust acceleration terms and has reached the asymptotic convergence range for these terms, the results can be taken as valid.

## 5.2 Apollo Powered Descent Guidance

The performance of APDG relative to E-Guidance is discussed here. The motivation is to show that, while not fuel optimal, it does produce reasonable results and can be expected to perform well while ensuring the practical mission consideration of final attitude constraints.

## 5.3 Adaptive Powered Descent Initiation

The performance of PDI is examined. Discuss its raw propellant impact and explain the results (not the absolute lowest possible in vacuum, but the lowest in atmosphere). Discuss the failure of pre-determined ignition conditions in Case 1 and 6 and how PDI deals with this. Do this by examining both Case 1 as the ”too late” condition and case 6 in atmosphere as the ”too early” condition. Talk about how PDI can be adjusted to account for dispersions, atmospheres, etc.

# CHAPTER 6

# CONCLUSIONS AND FUTURE WORK

## 6.1 PDI

Discuss its wider applicability to any powered descent problem in which engines have upper or lower bounds of thrust.

## 6.2 Future Work

Discuss refinements to PDI; tgo safety factor as function of conditions for mission planning, or directly optimal approach of some kind.

# CHAPTER 7

# REFERENCING

# BIBLIOGRAPHY

[1] C. Cerimele et al. A rigid mid lift-to-drag ratio approach to human mars entry, descent, and landing. *AIAA Paper*, 2017-1898, 2017.

[2] G. W. Cherry. A general, explicit, optimizing guidance law for rocket-propelled spaceflight. *AIAA Paper*, 64-638, 1964.

[3] C. S. D'Souza. An optimal guidance law for planetary landing. *AIAA Paper*, 97-3709, 1997.

[4] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, Berlin, 2002.

[5] S J. Citron, S E. Dunin, and H F. Meissinger. A terminal guidance technique for lunar landing. *AIAA Journal*, 2:503–509, 1964.

[6] Hilary L. Justh. Mars global reference atmospheric model. NASA, https://see.msfc.nasa.gov/model-Marsgram, accessed February 2018, 2010.

[7] A. R. Klumpp. Apollo lunar descent guidance. *Automatica*, 10:133–146, 1974.

[8] G. Leitmann. Class of variational problems in rocket flight. *Journal of the Aerospace Sciences*, 26:586–591, 1959.

[9] J. J. Meditch. On the problem of optimal thrust programming for a lunar soft landing. *IEEE Transactions on Automatic Control*, 9:477–484, 1964.

[10] T. J. Moesser. Guidance and navigation linear covariance analysis for lunar powered descent. Master's thesis, Utah State University, Logan, UT, 2010.

[11] J. R. Rea and R. H. Bishop. Analytical dimensional reduction of a fuel optimal powered descent subproblem. *AIAA Paper*, 2010-8026, 2010.

[5] [2] [7] [1]

# APPENDIX A

# PLACEHOLDER

# PLACEHOLDER

These appendices are placeholders.

# APPENDIX B

# PLACEHOLDER REDUX

# PLACEHOLDER REDUX