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

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 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 ignition timing, or when to start powered descent guidance. None investigate ignition timing 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 3.3 and 3.4 by defining a linear or quadratic thrust acceleration profile which ensures satisfaction of the terminal constraints 3.5, 3.6, and (for constrained final attitude with quadratic thrust profile) 3.21. 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 3.11. 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 ignition timing 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 3.2.2.

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]

1.2 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 ignition timing, 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 ignition timing 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.

1.3 Organization

CHAPTER 2
PROBLEM STATEMENT

The guidance algorithm for powered descent and 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 $\hat{e}$.

It is desired to minimize propellant usage required by the E-Guidance law by improving ignition timing. 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.

CHAPTER 3
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.

3.1 Guidance Law

The guidance law under investigation is E-Guidance, first presented in Cherry 1964 [2]. E-Guidance was developed empirically by integrating the equations of motion and choosing basis functions for the thrust acceleration input $\boldsymbol{a}_T$ that provided the necessary degrees of freedom to satisfy the terminal constraints in Equations 3.5 and 3.6. With control over thrust acceleration (and therefore total acceleration), satisfying the initial conditions after integration of the equations of motion shows the need for two basis functions with vector coefficients. Cherry developed E-Guidance by considering first one guidance axis at a time, so the acceleration command took the form of Equation 3.1, where $p_1(t)$ and $p_2(t)$ are linearly independent functions of time.

$$\ddot{x} = c_1 p_1(t) + c_2 p_2(t) \tag{3.1}$$

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 below in Equation 3.13. The Cherry paper 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 ignition timing, nor does most of the academic literature since.

Below E-Guidance as a control law will be derived using optimal control theory. An extension that also considers the final orientation will also be presented without claim about its optimality, but its performance will be compared with E-Guidance. The rationale behind choice of time-to-go algorithm and subsequent ignition timing will follow.

3.1.1 Equations of Motion

Derivation of an optimal powered descent guidance law begins with a formulation of the State Equation 3.2

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{3.2}$$

The command input $\boldsymbol{u}$ for this problem is comprised of the thrust magnitude and direction. 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 powered descent guidance problem are as follows

$$\dot{\boldsymbol{r}} = \boldsymbol{V} \qquad \boldsymbol{r}(\boldsymbol{t_0}) = \boldsymbol{r_0} \tag{3.3}$$

$$\dot{\boldsymbol{V}} = \boldsymbol{g}(\boldsymbol{r}) + \boldsymbol{a_T} \qquad \boldsymbol{V}(\boldsymbol{t_0}) = \boldsymbol{V_0} \tag{3.4}$$

with terminal constraints at a fixed final time t_f

$$\boldsymbol{r}(t_f) = \boldsymbol{r}_f^* \tag{3.5}$$

$$\boldsymbol{V}(t_f) = \boldsymbol{V}_f^* \tag{3.6}$$

where $\boldsymbol{a}_T$ is the thrust acceleration vector. $\boldsymbol{a}_T$ is limited such that

$$0 < a_{min} \leq ||\boldsymbol{a}_T|| \leq a_{max} \tag{3.7}$$

and $\boldsymbol{g}(\boldsymbol{r})$ is gravitational acceleration

$$\boldsymbol{g} = -\frac{\mu \boldsymbol{r}}{(\boldsymbol{r}^T\boldsymbol{r})^{(3/2)}} \tag{3.8}$$

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

3.1.2 Performance Index

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

$$\dot{m} = -k||\boldsymbol{a}_T|| \tag{3.9}$$

A fuel optimal guidance law should therefore use the performance index

$$J = \int_{t_0}^{t_f} ||\boldsymbol{a}_T||dt \tag{3.10}$$

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

$$J = \frac{1}{2}\int_{t_0}^{t_f} (\boldsymbol{g} + \boldsymbol{a}_T)^T(\boldsymbol{g} + \boldsymbol{a}_T)dt \tag{3.11}$$

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 3.10, but it does provide a cost to large thrust accelerations and might be expected to give good fuel performance.

3.1.3 Guidance solution

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

$$H = \boldsymbol{p}_r^T\boldsymbol{V} + \boldsymbol{p}_V^T\boldsymbol{u} - \frac{1}{2}\boldsymbol{u}^T\boldsymbol{u} \tag{3.12}$$

$$\begin{aligned}
\dot{\boldsymbol{p}}_r = -\frac{\partial H}{\partial \boldsymbol{r}} = 0 &\implies \boldsymbol{p}_r = -\boldsymbol{c}_2 \\
\dot{\boldsymbol{p}}_V = -\frac{\partial H}{\partial \boldsymbol{V}} = -\boldsymbol{p}_r &\implies \boldsymbol{p}_V = \boldsymbol{c}_1 + \boldsymbol{c}_2 t \\
\frac{\partial H}{\partial \boldsymbol{u}} = 0 &\implies \boldsymbol{u} = \boldsymbol{p}_V = \boldsymbol{c}_1 + \boldsymbol{c}_2 t
\end{aligned}$$

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

$$\boldsymbol{u} = \boldsymbol{k}_1 + \boldsymbol{k}_2\tau \tag{3.13}$$

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

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

$$\int \dot{\boldsymbol{V}}(t)dt = \boldsymbol{k}_1(t - t_0) + \frac{1}{2}\boldsymbol{k}_2(t - t_0)^2 + \boldsymbol{V}(t_0) \tag{3.14}$$

$$\int \dot{\boldsymbol{r}}(t)dt = \frac{1}{2}\boldsymbol{k}_1(t - t_0)^2 + \frac{1}{6}\boldsymbol{k}_2(t - t_0)^3 + \boldsymbol{V}(t_0)(t - t_0) + \boldsymbol{r}(t_0) \tag{3.15}$$

Setting $t = t_f$ and letting $t_{go} = t_f - t_0$ satisfies the terminal constraints from Equations 3.5 and 3.6, resulting in 6 linear equations in 6 unknowns

$$\boldsymbol{k}_1 t_{go} + \frac{1}{2}\boldsymbol{k}_2 t_{go}^2 = \boldsymbol{V}_f^* - \boldsymbol{V}_0 \tag{3.16}$$

$$\frac{1}{2}\boldsymbol{k}_1 t_{go}^2 + \frac{1}{6}\boldsymbol{k}_2 t_{go}^3 = \boldsymbol{r}_f^* - \boldsymbol{r}_0 - \boldsymbol{V}_0 t_{go} \tag{3.17}$$

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

$$\begin{bmatrix} t_{go} & \frac{1}{2}t_{go}^2 \\ \frac{1}{2}t_{go}^2 & \frac{1}{6}t_{go}^3 \end{bmatrix} \begin{pmatrix} k_{1_x} \\ k_{2_x} \end{pmatrix} = \begin{pmatrix} V_{f_x}^* - V_{0_x} \\ r_{f_x}^* - (r_{0_x} + V_{0_x} t_{go}) \end{pmatrix} \tag{3.18}$$

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

$$E = \begin{bmatrix} -2/t_{go} & 6/t_{go}^2 \\ 6/t_{go}^2 & -12/t_{go}^3 \end{bmatrix} \tag{3.19}$$

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

$$\begin{pmatrix} k_{1_x} \\ k_{2_x} \end{pmatrix} = E \begin{pmatrix} V_{f_x}^* - V_{0_x} \\ r_{f_x}^* - (r_{0_x} + V_{0_x} t_{go}) \end{pmatrix} \tag{3.20}$$

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.

Of some interest is the addition of a final attitude constraint. For a vehicle whose attitude it determined by the thrust acceleration vector, this constraint can be implemented as a final thrust acceleration constraint as in Equation 3.21.

$$\boldsymbol{a}_T(t_f) = \boldsymbol{a}_{T_f}^* \tag{3.21}$$

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 $\boldsymbol{u} = \boldsymbol{c}_1 p_1(t) + \boldsymbol{c}_2 p_2(t) + \boldsymbol{c}_3 p_3(t)$. A tempting choice for the third basis function is $p_3(t) = t^2$ for simplicity, with the other two functions the same as E-Guidance.

After applying the substitution from Equation 3.13, 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 3.18 and considering one coordinate a time we get a system similar to E-Guidance in Equation 3.22

$$\begin{pmatrix} k_{1_x} \\ k_{2_x} \\ k_{3_x} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 18/t_{go}^2 & -24/t_{go}^3 & -6/t_{go} \\ -24/t_{go}^3 & 36/t_{go}^4 & 6/t_{go}^2 \end{bmatrix} \begin{pmatrix} V_{f_x}^* - V_{0_x} \\ r_{f_x}^* - (r_{0_x} + V_{0_x} t_{go}) \\ g_x + a_{f_x}^* \end{pmatrix} \tag{3.22}$$

3.1.4 Time-to-go

The E-Guidance solution 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}

procedure FPI(t_{go_1})
    $tol \leftarrow c$
    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}
end procedure

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 3.23 for a gravity turn time-to-go, $t_{go_{GT}}$, was presented first in Cherry 1964 [2]. After engine ignition and initiation of the 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 3.23, 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 3.1.5.

$$
\begin{aligned}
\gamma &= \frac{\pi}{2} - cos^{-1}\left(\frac{\boldsymbol{r}^T\boldsymbol{V}}{||\boldsymbol{r}||||\boldsymbol{V}||}\right) \\
g_m &= ||\boldsymbol{g}|| \\
r_m &= ||\boldsymbol{r}|| \\
V_m &= ||\boldsymbol{V}|| \\
a &= 1/g_m^2 \\
b &= \frac{sin(\gamma)V_m^2}{2(r_m - R_M)g_m^2} \\
c &= -\frac{V_m^2(1 + sin(\gamma)^2)}{4(r_m - R_M)g_m} + 1 \\
a_{GT} &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\
t_{go_{GT}} &= \frac{V_m}{2}\left(\frac{1 + sin(\gamma)}{a_{GT} + g_m} + \frac{1 - sin(\gamma)}{a_{GT} - g_m}\right)
\end{aligned}
\tag{3.23}
$$

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 3.5 and 3.6. However, if the magnitude of the terminal velocity target in Equation 3.6 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.

3.1.5 Ignition Timing

The central focus of this thesis is the use of ignition timing to improve the fuel performance of E-Guidance. This is accomplished by making some attempt to force E-Guidance to command a nearly fuel optimal thrust acceleration profile. From Leitmann's [8] and Meditch's [9] work it is expected that a "bang-bang" style thrust profile is desirable. With E-Guidance controlling the thrust acceleration vector $\boldsymbol{a}_T$, the ignition timing may be chosen to force a nearly optimal solution. By choosing a time-to-go which is short enough that the thrust magnitude commanded by E-Guidance saturates at the upper bound for most of the flight, nearly optimal fuel performance may be expected.

Since E-Guidance does not take the thrust acceleration limits of Equation 3.7 into account directly the time-to-go must do so or risk the required thrust command exceeding the capabilities of the landing vehicle. One way to do this is to make use of the gravity turn solution as discussed in Section 3.1.4, Equation 3.23. This equation provides one criterion for ignition through the specified thrust acceleration $\boldsymbol{a}_{GT}$. If the gravity turn time-to-go $t_{go_{GT}}$ is used and the engine ignited for E-Guidance powered descent the moment the magnitude of the specified thrust acceleration $\boldsymbol{a}_{GT}$ reaches or exceeds the known maximum thrust acceleration magnitude a_{min}, E-Guidance will command a large initial thrust acceleration which exceeds the rocket's limits and the thrust will be saturated for most of the flight. Using a padded $t_{go_{GT}}$ helps ensure that there is some margin for error.

Another important criterion is the downrange travel required by a gravity turn maneuver. The engine should be ignited if the horizontal distance remaining to the 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. Equation3.24 gives the downrange distance traveled during a gravity turn maneuver s_{GT}, where h is the current altitude.

$$s_{GT} = \frac{V_m^2}{2 * a_{GT})} * cos(\gamma) * \frac{V_m^2 + 2 * g_m * h}{V_n orm^2 + g_m * h} * \frac{R_M}{r_m} \tag{3.24}$$

3.2 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.

3.2.1 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 3.25.

$$
\begin{aligned}
\boldsymbol{\phi}^*_{n+\frac{1}{2}} &= \boldsymbol{\phi}_n + \frac{\Delta t}{2}\boldsymbol{f}(t_n, \boldsymbol{\phi}_n) \\
\boldsymbol{\phi}^{**}_{n+\frac{1}{2}} &= \boldsymbol{\phi}^n + \frac{\Delta t}{2}\boldsymbol{f}(t_{n+\frac{1}{2}}, \boldsymbol{\phi}^*_{n+\frac{1}{2}}) \\
\boldsymbol{\phi}^*_{n+1} &= \boldsymbol{\phi}^n + \Delta t \boldsymbol{f}(t_{n+\frac{1}{2}}, \boldsymbol{\phi}^{**}_{n+\frac{1}{2}}) \\
\boldsymbol{\phi}_{n+1} &= \boldsymbol{\phi}^n + \frac{\Delta t}{6}[\boldsymbol{f}(t_n, \boldsymbol{\phi}_n) + 2\boldsymbol{f}(t_{n+\frac{1}{2}}, \boldsymbol{\phi}^*_{n+\frac{1}{2}}) \\
&\quad + 2\boldsymbol{f}(t_{n+\frac{1}{2}}, \boldsymbol{\phi}^{**}_{n+\frac{1}{2}}) + \boldsymbol{f}(t_{n+1}, \boldsymbol{\phi}^*_{n+1})]
\end{aligned}
\tag{3.25}
$$

The EOM function computes the derivative of the state vector $\boldsymbol{\phi}$. It is the same as the modeled equations of motion in Equations 3.3 and 3.4 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 3.26 and 3.27.

$$\dot{\boldsymbol{r}} = \boldsymbol{V} \tag{3.26}$$

$$\dot{\boldsymbol{V}} = \boldsymbol{g}(\boldsymbol{r}) + \boldsymbol{a_T} + \boldsymbol{F}_{LD} \tag{3.27}$$

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 3.27.

3.2.2 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 advanced E-Guidance with attitude constraint), 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 satisfy the real performance constraint represented in Equation 3.7. 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. 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 3.18 and 3.22. 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 3.1.4. 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. As discussed, the gravity turn time-to-go is used with an applied factor of 1.2 for a safety margin.

3.2.3 Ignition trigger

Ignition time is determined by checking the two criterion discussed in Section 3.1.5: 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 3.2.2. 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 clear in Figure **??**. 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.

3.2.4 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 3.28, where α 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.

$$\boldsymbol{x}_{est} = \alpha \boldsymbol{x}_{prev} + (1 - \alpha)\boldsymbol{x} \tag{3.28}$$

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 distrubution, modeling a Kalman filter would likely provide too good an estimate effectively canceling the effects of simulated navigation error.

3.2.5 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]

3.2.5.1 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 ρ 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 3.30 and 3.29, where S is a reference area.

$$L = \frac{1}{2}\rho |V|^2 S C_L \tag{3.29}$$

$$D = \frac{1}{2}\rho |V|^2 S C_D \tag{3.30}$$

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 ignition timing phase by simply dividing the reference area by a factor of 2.

3.2.5.2 LIFT AND DRAG

The vehicle aerodynamic model is based on the work of Cerimele et al [1]. The vehicle is the CobraMRV, a "rigid, enclosed, elongated lifting body shape that provides a higher lift-to-drag ratio (L/D) than a typical entry capsule..." [1]. 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.

The work of Cerimele er al provides a model of the lift and drag coeffecients 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 ignition timing is being optimized, the vehicle is held at a constant angle of attack of $55°$ to achieve maximum drag.

3.2.5.3 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 3.31, 3.32, 3.33. 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.

$$\boldsymbol{body}_Y = \boldsymbol{a}_T/||\boldsymbol{a}_T|| \tag{3.31}$$

$$\boldsymbol{body}_P = \frac{\boldsymbol{r} \times \boldsymbol{body}_Y}{||\boldsymbol{r} \times \boldsymbol{body}_Y||} \tag{3.32}$$

$$\boldsymbol{body}_R = \boldsymbol{body}_Y \times \boldsymbol{body}_R \tag{3.33}$$

This calculation ensures zero side-slip, that the vehicle will always be pointed in the direction it is traveling. Figure **??** shows a small changing yaw angle throughout flight since it is computed relative to a fixed guidance frame centered on the landing site, and the vehicle's initial condition is rotated around the curvature of the planet.

3.2.6 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 3.2.4. Rocket parameter dispersion is modeled as a uniform distribution of 2% of the parameter. Equation 3.34 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.

$$T_{max} = T_{max_{nom}} * (1 + 0.02 * (1 - 2 * rand)) \tag{3.34}$$

Each rocket parameter is modeled in this fashion.

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.

CHAPTER 4

RESULTS AND DISCUSSION

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

CHAPTER 6

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 containing the original text from Prof. Peter Blomgren’s thesis template.

APPENDIX B

PLACEHOLDER REDUX

PLACEHOLDER REDUX