University of Texas Rio Grande Valley

ScholarWorks @ UTRGV

5-2023

An Investigation Into Optimal Descent Trajectories for Multipurpose Long Range Space Vehicles Under Advanced Conditions

John M. Levis
The University of Texas Rio Grande Valley

AN INVESTIGATION INTO OPTIMAL DESCENT TRAJECTORIES FOR

MULTIPURPOSE LONG RANGE SPACE VEHICLES

UNDER ADVANCED CONDITIONS

A Thesis

by

JOHN M. LEVIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: Mathematics

The University of Texas Rio Grande Valley

May 2023

AN INVESTIGATION INTO OPTIMAL DESCENT TRAJECTORIES FOR

MULTIPURPOSE LONG RANGE SPACE VEHICLES

UNDER ADVANCED CONDITIONS

A Thesis
by
JOHN M. LEVIS

COMMITTEE MEMBERS

Dr. Andras Balogh
Co-chair of Committee

Dr. Josef Sifuentes
Co-chair of Committee

Dr. Anahit Galstyan
Committee Member

Dr. Cristina Villalobos
Committee Member

May 2023

ABSTRACT

Levis, John M., <u>An Investigation into Optimal Descent Trajectories for Multipurpose Long Range Space Vehicles Under Advanced Conditions</u>. Master of Science (MS), May, 2023, 105 pp., 7 tables, 20 figures, references, 24 titles.

In this work, we investigate the problem of fuel-optimal control of space vehicle descent trajectories. The main tool we use to establish optimality is Pontryagin's Maximum Principle. We present a variety of scenarios with increasing complexities, including drag, wind, and moving landing platforms in the context of differing atmospheric and gravitational conditions. Throughout the paper, we use a balance of analytical and numerical techniques. Finally, observations and conclusions drawn from the investigation form the basis for suggestions into additional areas of analysis.

DEDICATION

      I would like to dedicate this work to my family: to my three sons, Jack, Sam and Ben, whose own journey into their college years inspired me to return to school along with them; to their mother, Barb, whose patience and support throughout this effort was extremely encouraging; to my sister, Carolyn, whose own academic pursuits alongside a demanding professional career provided a template for me to follow; and to my parents, Mary and Ernie, for their dedication to the growth and support of their children.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION & FOUNDATIONS

1.1   Introduction

1.1.1   Origins of the Problem

Over the past decade there has been a renewed space-race reminiscent of the late 1950's

and 1960's. However, the driving objective of that era, a desire for military and political

superiority, has been replaced by the prospects of individual achievement and commercial gain.

Unlike the original space race, this new eagerness for interstellar travel is being driven by self-

made billionaires who, along with a desire to go where no one has gone before, bring a practical

reality to the economics of such ventures.

One of the primary areas of opportunity, and some will say necessity, for better

economics is the ability to be able to safely land space craft on a planet such as Mars (for

starters) with a full payload and then for that rocket to be able to relaunch itself for a return or

ongoing trip. Certainly, the engineering challenges for this kind of maneuver are plentiful.

These include being able to orient and reorient the craft for atmospheric and non-atmospheric

travel, dealing with the heat generated by atmospheric resistance, and many others. At the core,

however, is the need to be able to calculate and achieve descent trajectories that optimize the

craft's ability to safely, and economically, land and relaunch.

1.1.2   Outline of the Paper

This paper focuses on the derivation of mathematical solutions, both analytic and numeric, to the problem of finding optimal descent trajectories under a few advanced conditions. Underlying the research is the use of optimal control theory to determine the optimal use of the vehicle's engines as a means of controlling the descent to achieve the desired outcomes. In this first chapter, we provide an introduction to the topic of optimal control theory first by illustrating two elementary problems, and then by diving deeper into the mathematical theory.

In CHAPTER II, we begin applying that theory to the simplest of optimal descent scenarios, the lunar landing problem. In this simple scenario where the only naturally occurring force is that of gravity, we can see how the use of optimal control theory guides to the optimal way of using the control – in this case the thrust generated by the vehicle's engines – to guide the vehicle to a safe and soft landing while minimizing the fuel consumed in doing so.

CHAPTER III tackles the effects of atmospheric density by first modeling a case when the density is constant for the entire descent. It then examines the more realistic case where the density decreases the closer the vehicle gets to the surface. Under these conditions, we see how the behavior of the control varies with the presence of another naturally occurring force, drag. We also see how the relative strength of the forces of gravity and drag create different optimalities for the descent trajectories.

CHAPTER IV examines an even more advanced scenario where not only does the presence of an atmosphere induce drag on the descending vehicle, it also creates the possibility

of wind influencing the trajectory. We will also briefly extend the results of the other chapters in a scenario whether the desired landing platform is in motion as it would be in any situation involving a rotating celestial body.

Finally, in CHAPTER V, we summarize the results of the paper and outline two possible avenues for further investigation. Throughout the paper, analytic solutions are developed as far as practical at which point numerical solutions are used to illustrate and analyze the outcomes.

1.1.3 Recent Works

A review of related literature yields recent work in this area in two broad categories:

- Research papers focused on solutions to the optimal descent trajectories for vehicles on celestial bodies with no material atmosphere such as earth's moon. This aligns with NASA's renewed interest in lunar exploration through the Artemis program.
- Investigations into optimal descent trajectories for the reuse of launch vehicles by safely returning them to earth as efficiently as possible. SpaceX has been a very visible example of this.

(Brodkin, 2021) and (Macki, 2006), among others, explore the first category taking advantage of the simplifying assumptions that come with landing under conditions where the atmosphere is negligible. The models, as we will show, provide an illustrative example of the use of optimal control theory in solving such problems. However, the simplicity of the model limits its use beyond that scenario.

On the other hand, (Lu, 2018) and (Liu, 2018), among others, demonstrate a more robust treatment of the problem taking into account the additional complexities of atmospheric forces and a more sophisticated engineering of the space vehicle. In particular, Liu, outlines a model

that accounts for the both atmospheric drag and lift along with additional controls related to flight path and thrust angle. Both of these papers, along with others including (Prous, 2020), explore the optimality of both launch and landing which has applications in the reuse of rockets desired to place satellites into orbit.

Where this paper differs from others is in how it extends the modeling through the additional attention paid to ensuring the existence of optimal solutions and through additional scenarios that include conditions such as atmospheric wind and a moving landing surface.

1.2 Notations and Nomenclature

Throughout this paper, we will use a consistent set of notations and nomenclature to refer to mathematical models and their components. We will also adopt the convention that vectors will be represented in a bold font and the time derivative of variables will appear with a single dot above them.

1.2.1 Optical Control Foundations

Table 1-1: Variables Used in Optimal Control Equations

Variable/Parameter	Explanation
t	Time variable, often decomposed into discrete intervals
$\mathbf{x}(t)$	State variable describing the dynamics of the system
$\mathbf{u}(t)$	Control variable used to influence the behavior of the system
$\mathbf{f}(\mathbf{x}, \mathbf{u})$	Function used to model the motion of the system
$\lambda(t)$	Co-state variable introduced to mirror the state variable in confirming the controllability of a system
$H(\mathbf{x}, \mathbf{u}, \lambda)$	Hamiltonian equation used in Optimal Control Theory to govern the controlled system's dynamics
$C(\mathbf{x}, \mathbf{u})$	The "cost" function which is value being either minimized or maximized by the use of the control
$M(\mathbf{x}, \mathbf{u})$	The "variable" portion of the cost function

$N(\mathbf{x}_f)$	The "fixed" portion of the cost function associated with its final state

1.2.2 Optimal Descent Scenarios

Table 1-2: Variables Used in Modeling Descent Scenarios

Variable/ Parameter	Definition	Units	Specific Uses
t	Time variable	s	t_0 = initial time of descent t^* = optimal switching time t_f = time at landing
$h(t)$	Height of the vehicle above the landing surface	m	h_0 = initial height h_f = final height at landing
$r(t)$	Downrange position of the vehicle	m	r_0 = initial downrange position h_f = final downrange position at landing
$v(t)$ $v_h(t)$	Vertical velocity of the vehicle (positive is up)	m·s^{-1}	v_0, v_{h0} = initial vertical velocity v_f, v_{hf} = final velocity at landing
$v_r(t)$	Downrange velocity of the vehicle (positive is right)	m·s^{-1}	v_{r0} = initial downrange velocity v_{rf} = final downrange velocity at landing
$m(t)$	Mass of the vehicle including fuel	kg	m_0 = initial mass m_f = final mass at landing
$\mathbf{x}(t)$	State variable describing the dynamics of the system		$[h\ v\ m]^T$ for Scenarios 2 and 3 $[h\ r\ v_h\ v_r\ m]^T$ for Scenarios 4
$\mathbf{u}(t)$	Thrust of the vehicle engines (up and right are positive)	N	u, u_h = vertical engines thrust u_r = downrange engines thrust
k	Efficiency of thrust engines	s·m^{-1}	k, k_h = vertical engines efficiency k_h = downrange engines efficiency
$\rho(h)$	Atmospheric density	kg·m^3	ρ_0 = atmospheric density at surface
δ	Drag coefficient of vehicle	n/a	δ, δ_h = longitudinal profile of vehicle δ_r = vertical profile of vehicle
$w(h)$	Windspeed surrounding the vehicle (positive is right)	m·s^{-1}	w_0 = windspeed at the surface

1.3   Optimal Control Foundations

1.3.1   Uncontrolled and Controlled Systems

The fundamental idea behind Optimal Control Theory (OCT), and control theory in general, is the desire to influence the outcomes of a system through the use of controls imposed on the system in a certain manner at certain times during the execution of the system. The fields of systems where such "controls" could be applied include economics, biological, physical, electrical and many others. However, all of these applications share some foundational elements at the core of OCT.

We start with a basic system as modeled through a set of ordinary differential equations as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t)\big)$$

$$t \geq 0$$

$$\mathbf{x}(0) = \mathbf{x_0}$$

$$(1.1)$$

For simplicity, we will confine our discussion to real numbers although there is nothing preventing this from extending into complex numbers. As such, we have $\mathbf{f} \colon \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{x} \colon [0, \infty) \to \mathbb{R}^n$. To the degree it can be solved, the solution to (1.1) and the resulting behavior of the system can be thought of as "uncontrolled" by any influences outside the system. For example, a simple system involving dropping an object from an elevated

6

position can be modeled. The object will drop under the influence of gravity, drag and

potentially wind and will eventually land somewhere on the ground.

Now imagine that we wanted to control the object by affixing a small power source of

some kind designed to propel the object in some direction for some amount of time and at some

points in time during its fall. We might want to do this to ensure the object fell *where* we wanted

it to fall and/or *when* we wanted it to land. In other words, we want to *control* the system in

some way to meet some objective.

Mathematically that involves introducing a "control" variable, u, into the system in this

way

$$\dot{\mathbf{x}}(t) = \hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{u})$$

$$t \geq 0$$

$$\mathbf{x}(0) = \mathbf{x_0}$$

$$(1.2)$$

Note that **u** is potentially a function of some combination of t, $\mathbf{x}(t)$ and $\dot{\mathbf{x}}(t)$. That is, the

control can be triggered at specific points in time, at specific positions of the system, or at

moments when the system's natural rate of change reaches a certain level. Note, $\mathbf{u} \in \mathcal{U}$ which is

set of all permissible values for the control variable. We will discuss the characteristics and

constraints of $\mathcal{U}$ later in this paper.

In order to visualize a control in action, consider the system in $\mathbb{R}^2$

$$\dot{x}(t) = f(x(t), u(t))$$
$$t \geq 0$$
$$x(0) = x_0$$

$$(1.3)$$

where,

$$u(t) = \begin{cases} u_1(t), 0 \leq t \leq t_1 \\ u_2(t), t_1 \leq t \leq t_2 \\ u_3(t), t_2 \leq t \leq t_3 \end{cases}$$

$$(1.4)$$

For example, let us consider the simple case where in (1.3) where $f(x) = \frac{-g}{m} + u$ and the control, u, is as in (1.4). Here we are modeling our object of mass, m, under the influence of gravity, g, and our control. If we illustratively plot the trajectory of the controlled and uncontrolled system in Figure 1.1, we can see the different behavior the control imparts on the system.
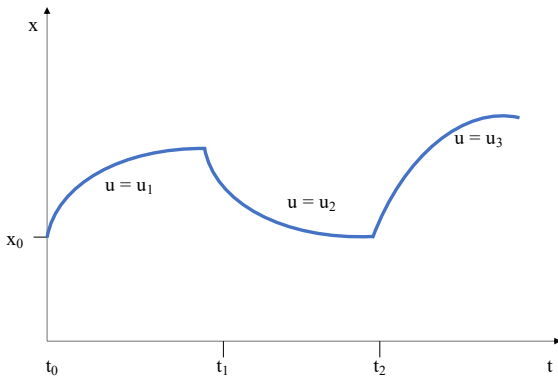
Figure 1.1: Discrete Controls

8

1.3.2 Properties for Controllability

As noted by (Macki, 2006), modelling a system in the form of control system as we did in (1.2) does not actually guarantee that the system is indeed controllable. In order to determine if a system is controllable, we must examine two key properties:

    i.   the initial state of the system relative to the desired outcome and

    ii.   the nature of the control

The first condition relates to whether or not one can control the system, that is, have it reach a desired outcome, based on its starting point. We define this as the controllable set of initial points as follows:

$$\mathcal{X}(t_f) = \{\mathbf{x}_0 \in \mathbb{R}^n \mid \exists\, \mathbf{u} \in \mathcal{U} \text{ so that } \mathbf{x}(t_f) = \mathbf{x}_f\}$$

$$(1.5)$$

where,

$\mathcal{X}$ is the set of all possible states of $\mathbf{x}$

$\mathcal{U}$ is the set of all possible controls, $\mathbf{u}$, that can be applied

t_f is an acceptable final time

$\mathbf{x}_f$ is the desired final state of the system

The second condition refers to the type of controls that can be applied to the system based, in turn, on the nature of the system and its scientific limits. Although there are many different controls that can be modelled, (Macki, 2006) describes three common ones that could apply in this scope of this paper:

a) *Bang-Bang* controls where the control is defined as either instantaneously on (at full power) or off (e.g., a light switch) which can be modelled as the switch between two values

b) *Piecewise Constant* where the control is constant for particular time interval (although the time interval and/or the magnitude of the constant can vary from interval to interval) (e.g., the volume control on a TV while watching a show among periodic background noise)

c) *Absolutely Continuous* where the magnitude of the control varies smoothly at all times within the duration of the system (e.g., steering of a car while driving)

Thinking about the controllability in these terms leads to the obvious conclusion that the two properties are not independent of each other. Mathematically, that is clear from (1.5). Intuitively, it becomes clear when thinking about our objective drop example from the previous section. If we drop the objective from too high an altitude that there is insufficient power in the control to overcome the force of gravity on it then we would not be able to land it where and when we want. We will refer to these properties of controllability as we explore the optimal descent trajectory problem in various scenarios later in this and subsequent chapters.

1.3.3   Optimally Controlling a System

The introduction of a single control scenario into a system is generally insufficient for most purposes. Instead, we want to consider a set of controls, $\mathcal{U}$, where $\mathbf{u}(\dot{x}, x, t) \in \mathcal{U}$. Each control in the set could potentially influence the system in a different way. In our simple example of the dropping object with an on-board propeller, the propeller could be instructed to fly the object in a spiral on its way down or land in a designated spot using the least amount of power.

This ability to control a system in multiple ways gives rise to controlling the system with some objective in mind. This leads to the second foundational element of OCT, the Cost Function. Different authors refer to it by other terms: (Evans, 1983) refers to it as the "Payoff"; (Brodkin, 2021), refers to it as the "Objective", and other terms are also used in the specific context of the application. For the rest of this paper, we will define the Cost Function as

$$C[\mathbf{u}] := \int_0^{t_f} M\big(\mathbf{x}(t), \mathbf{u}(t)\big) dt + N(\mathbf{x}(t_f))$$

(1.6)

where the Cost is a function of the control variable, u. The functions, $M(\mathbf{x}(t),\mathbf{u}(t))$ and $N(\mathbf{x}(t_f))$ are known as the variable cost and fixed cost of the system, respectively, that runs from $0 \leq t \leq t_f$. More specifically, M models the costs associated with the path the system takes to reach its end state, whereas, N models the costs associated with the particular end state reached regardless of the path taken. Of course, **x** solves the ODE under the influence of control, **u**, where $\mathbf{u} \in \mathcal{U}$.

We define the optimal control, $\mathbf{u}^* \in \mathcal{U}$, as the control that minimizes the cost function such that

$$C(\mathbf{u}^*) \leq C(\mathbf{u}) \ \forall \ \mathbf{u} \in \mathcal{U}$$

(1.7)

Given the desire to find $\mathbf{u}^*$ for a particular system, three key questions emerge:

1. Under what circumstances does an optimal control exist?
2. If it does exist, are we able to calculate it?
3. If we are able to calculate it, is it unique?

Before we explore those questions, let us examine the modeling of a couple of simple systems that will be very helpful later in the paper. We will not fully solve for the optimal control at this point but this will set the context for how optimal control problems are framed.

1.4 Examples of the Use of Optimal Control Theory

1.4.1 Example 1: The Rocket Car

Consider a car on a horizontal railway track with rocket engines on each end as depicted in Figure 1.2. The system can be modeled as follows:

$p(t)$ is the position of the center of the car at time t

$v(t)$ is its velocity at time t

$u(t)$ is the thrust produced from the engines at time t where $-1 \le u(t) \le 1$

m is the (constant) mass of the rocket car
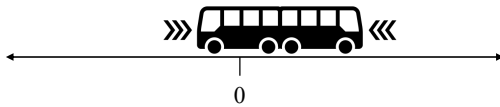
Figure 1.2: Rocket Car

Our objective is to determine the sequence of firing the rockets to move the car from $p(0) = p_0$ to $p(t_f) = 0$ in the least amount of time and when the car is stopped, that is, $v(t_f) = 0$. In order to illustrate this idea, we will make some simplifying assumptions. First, the rocket car's mass will be fixed in order to eliminate the need to model the impacts of fuel consumption. Second, we will also assume that the wheel assemblies and track are frictionless to avoid the

need to model drag on the car's motion. (When we examine the main problem of this paper, the optimal descent of a space vehicle, we will explore the effects of both of these).

In this case, our simplified rocket car system follows Newton's laws of motion

$$u(t) = m\dot{v}(t)$$

(1.8)

which we can convert this second order ODE to a system of first order ODEs as follows:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} p(t) \\ \dot{p}(t) \end{bmatrix}$$

$$\ddot{p}(t) = \frac{u(t)}{m}$$

We can derive the Cost Function by noting that we are trying to determine the control required to have the rocket pass by (but not necessarily stop at) $p = 0$ the first time

$$C(u) = \int_0^{t_f} 1 \, ds$$

(1.9)

and our task would be to find u^* that is of the bang-bang type described above (this will be proven in section 1.5 meaning that control switches from one value (± 1) to the other (∓ 1) only once during the optimal trajectory). This allows us to formally state the optimal control problem as follows:

$$\min_u \left(C(u) = \int_0^{t_f} 1 \, ds \right)$$

$$\text{such that } \dot{x} = \begin{bmatrix} x_2(t) \\ \frac{u(t)}{m} \end{bmatrix}$$

13

$$\text{where } u:[0, \infty) \rightarrow \{-1, 1\}$$

<div align="right">(1.10)</div>

With this in place, we can now visually explore the behavior of the system by returning to the variables introduced at the start of this section.

$$\dot{p}(t) = v(t)$$

$$m\ddot{p}(t) = u(t)$$

$$u(t) = \pm 1$$

<div align="right">(1.11)</div>

Since the mass is constant, we can set $m = 1$ to simplify the equations even further. We can also define some initial and terminal conditions as follows:

$$p(t_0) = p_0$$
$$\dot{p}(t_0) = v_0$$
$$p(t_f) = 0$$

Notice that we are not imposing the condition that $\dot{p}(t_f) = 0$ as for this simple illustration we are only interested in minimizing the time required for the car to get back to $p(t_f) = 0$. In the first case, let us assume without loss of generality that the control is set to 1 in which case the system in (1.11) becomes

$$\ddot{p}(t) = 1$$

To solve this system, we first note that from the second equation, $\dot{p}(t) = t + K$ with $K = v(t_0) = v_0$ we have $v(t) = t + v_0$. Inserting this into the first equations delivers $p(t) = \frac{1}{2}t^2 + v_0 t +$

K' with $K' = p(t_0) = p_0$ we arrive at $p(t) = \frac{1}{2}t^2 + v_0 t + p_0$. We can combine the expressions for

v(t) and p(t) by substituting for t to observe that

$$p(t) - p_0 = \frac{1}{2}(v^2(t) - v_0^2)$$

or

$$p(t) = \frac{1}{2}v^2(t) + \alpha$$

where,

$$\alpha = p_0 - \frac{1}{2}v_0^2$$

Now, since α is a constant which is directly related to the initial conditions, the result is a

set of parabolas in the v-p plane. Similarly, if we assume that the control is set to -1 over some

time interval $[t_1, t]$ we have

$$\dot{p}(t) = v(t)$$
$$\dot{v}(t) = -1$$

which yields the following similar outcome

$$p(t) = -\frac{1}{2}v^2(t) + \beta$$

where,

$$\beta = p(t_1) + \frac{1}{2}v^2(t_1)$$

Again, since β is a constant which is directly related to the conditions at which the

control, u, is set to -1, the result is another set of parabolas in the v-p plane. The geometric

15

interpretation of these results is shown in Figure 1.3. At t_0 the rocket car is set in motion by the

control value of u = 1. As it travels long the rail, it maps out a parabolic curve in v-p plane as

depicted by the green arrow labelled as such. When it intersects the inverted parabola that passes

through the origin, the control is flipped to u = -1. This occurs at time $t = t_1$. The rocket car then

moves along the rail following this new curve until it reaches the origin at $t = t_f$.
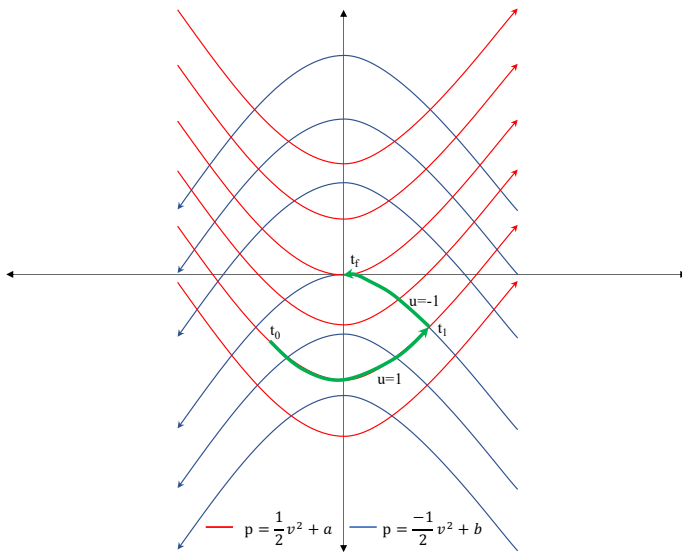
Figure 1.1: Geometric Illustration of a Bang-Bang Control

A critical observation we can draw from the illustration is that the initial conditions of

position and velocity of the rocket car, along with the possible values of the control dictate

whether there is an optimal solution for the system. For example, if the starting position,

velocity and control were as depicted in Figure 1.4, the rocket car's v-p trajectory would never

intersect a curve where a single switch to u = -1 would return it the origin. Instead, it would need

to make two switches, at $t = t_1$ and $t = t_2$ to accomplish it. However, if the order of the control

variable values were reversed such that at t_0 u = - 1, then an optimal path would exist by

16

switching u to 1 only once at the point marked t_2. In 1.5 we shall see, and in fact prove, that in problems of this nature involving both linear state and control variables, a single switch of a so-called "bang-bang" control produces an optimal result.
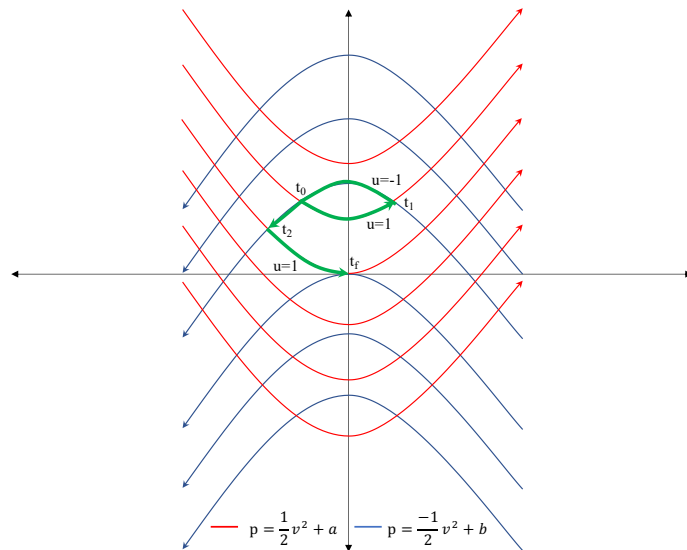
Figure 1.4: Geometric Illustration of a Sub-optimal Bang-Bang Control

The trajectory of the rocket car over the interval $t_0 \leq t \leq t_f$ is illustrated in Figure 1.5. After an initial velocity in the positive direction to the right, the engine fires at time t_1 exerting a force in the opposite (left) direction which will cause the vehicle to reverse direction and return back towards the origin. Note, that we are defining t_f as the time at which the rocket car first returns to the origin. We are not constraining that the velocity be zero at that point. This is another simplification of the model that we will address in 1.4.2 and beyond.
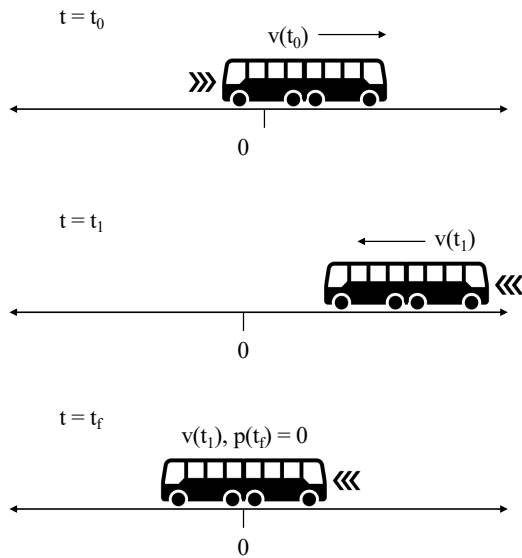
Figure 1.5: Rocket Car in Motion

What becomes clear from this simple example of the use of optimal control is that the system is very sensitive to both the initial conditions and the domain of the control variables. Understanding the impact of initial conditions and control limitations is a critical component of optimal control theory and something we will explore more fully in the vehicle descent scenarios we explore in the rest of this paper.

1.4.2   Example 2: Lunar Vehicle Landing

As a second example, let's consider a very simple version of the main problem of this paper, the soft landing of a space vehicle on the moon's surface while using the least amount of fuel. There are two forces affecting the vehicle: the force of gravity, G, in a downward direction and the upward thrust generated by the engine, T. The situation is depicted in Figure 1.6 and we can model the problem as follows

h(t) is the height of the vehicle at time t

v(t) is its velocity at time t

m(t) is its mass at time t

u(t) is the thrust applied at time t

Furthermore, we will set boundary, initial and terminal conditions as follows:

$0 \leq t \leq t_f$, where t_f is the time at which the vehicle lands

$h(0) = h_0$, the initial altitude of the vehicle above the surface

$v(0) = v_0$, the initial velocity of the vehicle

$m(0) = m_0$, the initial mass of the vehicle

$h(t_f) = 0$, the vehicle lands on the surface

$v(t_f) = 0$, the soft-landing requirement dictates zero velocity at t_f

$m(t_f) = m_{t_f}$, the mass of the vehicle at landing
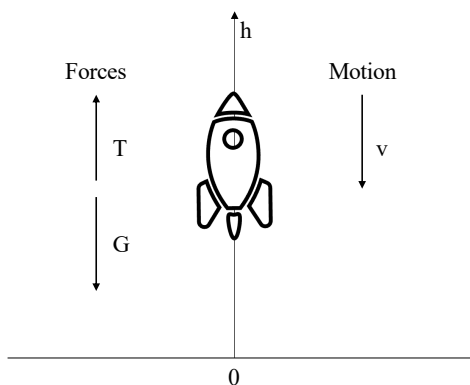
Figure 1.6: Lunar Landing Scenario

Note that in this case, unlike the rocket car, the use of fuel to generate thrust reduces the mass of the vehicle. Also, for simplicity, we will assume that thrust is unidirectional and limited as

$$u_{min} \leq u(t) \leq u_{max}$$

and applying Newton's second law in this case gives us

$$-gm(t) + u(t) = m(t)\dot{v}(t)$$

We then recognize that the control is really the thrust of the engines which in turn consumes the fuel on the vehicle. So, the rate of change of the mass can be described in terms of the control

$$\dot{m}(t) = -ku(t)$$

where k is the reciprocal of the speed at which the engine exhaust is expelled. Now, since we are trying to minimize the amount of fuel consumed, we can state the cost function as:

$$C(u) = -\int_0^{t_f} \dot{m}(s)\, ds = m(0) - m(t_f)$$

$$(1.12)$$

We note that because the vehicle is consuming fuel, $\dot{m}(t) \leq 0$ for the duration of the descent. Several interesting things emerge from (1.12):

- the actual calculation of the cost function, C(u), in this problem is straightforward once $m(t_f)$ is derived

- there are other potential ways to represent the cost function in this problem; for example, (Liu, 2018) shows how to minimize the "objective" function which is the negative of the

final mass of the vehicle and (Prous, 2020) attempts to minimize the square of the amount of thrust use

- regardless of how the cost function is framed, provided that it makes mathematical sense in the context of the problem, the challenge is deriving the control that optimizes it

- finally, minimizing C(u) is equivalent to minimizing the time it takes to land the vehicle

1.5 Conditions for Optimality

In both of these simple cases we were able to derive the cost function under the simple conditions laid out. In more realistic scenarios, the questions of existence and uniqueness become more important to address. We will address these in order.

1.5.1 Existence of an (Optimal) Control

Existence theorems for the solutions of the general case of optimal control problems is an active field of study, especially for non-linear systems. Indeed, before determining if an optimal control exists, we need to confirm that any control exists. The controllability question is often framed in this manner: given a system with an initial state, $\mathbf{x}_0$, and a desired outcome, $\mathbf{x}_f$, does there exist a control, $\mathbf{u}$, that can influence the system to attain $\mathbf{x}_f$ in a finite period of time? We first explored this in section 2.2 but now take a closer look especially as it relates to finding an *optimal* control.

There has been a great deal of study in investigating the conditions under which an optimal control both exists and can be efficiently determined. Although easier when the systems in question are linear, it turns out that the linearity of the system is not a direct driver of the existence of an easily derivable optimal control. What is more critical is the convexity of the set of controls, $\mathcal{U}$, and the constraints imposed on them. We recall that a set, A, is convex if

$$\forall\, a_1, a_2 \in A \text{ then } sa_1 + (1 - s)a_2 \in A \text{ where } s \in [0,1]$$

<div align="right">(1.13)</div>

In the context of our exploration, this implies that both $\mathcal{U}$ and the set of constraints on the controls must be convex if we are going to use calculation methods that depend on that convexity for efficiency. The reason for this is that a non-convex set of controls or constraints on controls has the potential to yield local extrema that may not be global extrema. That is, we may calculate a control which is only optimal for a portion of the system instead of the entire system. As we shall see, the convexity of the controls and in particular the control constraints come into question in the optimal descent trajectory problem and we will demonstrate an approach for addressing it.

1.5.2 Pontryagin Maximum Principle

One of the most powerful tools we have at our disposal to compute and confirm the existence of an optimal control, is the Pontrygain Maximum Principle (Pontryagin, Bolyanskii, Gamkrelidze, & Mishchenko, 1962). As we will see and show, the maximum principle can be applied in different ways to suit the desired conditions of the system. One of the things that the maximum principal makes clear is the usefulness of the Hamiltonian in determining optimal control solutions. However, the Hamiltonian as defined by Pontrygain is different from the Hamiltonian of classical mechanics which is defined by (Brittanica, 2022) "the Hamiltonian of a system specifies its total energy—i.e., the sum of its kinetic energy (that of motion) and its potential energy (that of position) ...". Instead, according to (Ferguson & Lim, 1998), the Hamiltonian used in optimal control problems "can be understood as an instantaneous increment

of the Lagrangian expression of the problem that is to be optimized over a certain time period."

As such we define the Hamiltonian as

$$H(x, \lambda, u) \ = \ f(x, u) \cdot \lambda \ + \ M(x, u)$$

In its most general version, the maximum principle states that the optimal state, $\mathbf{x}^*$, optimal

control, $\mathbf{u}^*$, and Lagrange multiplier, $\boldsymbol{\lambda}^*$, will minimize the Hamiltonian, H, of the system as

described in (1.3), over the time interval $0 \leq t \leq t_f$ such that

$$H(x^*(t), u^*(t), \lambda^*(t), t) \ \leq H(x(t), u(t), \lambda(t), t)$$

$$(1.14)$$

$$-\dot{\lambda}^T(t) \ = \ H_x(x^*(t), u^*(t), \lambda(t), t) = \lambda^T(t) f_x(x^*(t), u^*(t)) \ + \ M_x(x^*(t), u^*(t))$$

$$(1.15)$$

$$\lambda^T(t_f) = N_x(x(t_f))$$

$$(1.16)$$

and if the final state of the system, x(t$_f$) is not fixed, then

$$N(x(t_f)) + H(t_f) \ = \ 0$$

$$(1.17)$$

Recall from 1.3.3 that $M(x(t), x(t))$ and $N(x(t_f))$ represent the variable and fixed costs of

the system, respectively. The other important element introduced is the appearance of the

Lagrange multiplier, λ, which is often referred to as the "costate". We can see from (1.14) that

the costate variable also solves H. According to the Maximum Principle, equations (1.14) -

(1.17) represent necessary conditions for optimality. These conditions become sufficient when

the cost function and the constraints of the system meet certain convexity conditions. We will

discuss this in the context of the lunar landing problem in section 2.1.1.

As noted earlier, the Maximum Principle, can be applied in a number of different situations according to the nature of the problem. In many cases, that is what (Evans, 1983) refers to as a "fixed time, free end state" problem where we are trying to minimize the time to advance the system from a given state. However, in this paper, we are concerned with the category of a "free time, fixed end state" problem. Said differently, we are less concerned about how much time it takes the system to reach the desired end state because we are trying to optimize the amount of fuel spent in reaching the desired state.

In this case, we have the following conditions for achieving optimality over $0 \leq t \leq t_f$

$$\dot{x}^*(t) = \nabla_\lambda H(x^*(t), u^*(t), \lambda^*(t))$$

$$\dot{\lambda}^*(t) = -\nabla_x H(x^*(t), u^*(t), \lambda^*(t))$$

$$H(x^*(t), u^*(t), \lambda^*(t)) = \min_u \{ H(x^*(t), u(t), \lambda^*(t)) \}$$

$$H(x^*(t), u^*(t), \lambda^*(t)) = 0$$

(1.18)

We should note that in this case, t_f represents the first time the system reaches its desired state. In our case, that is the instant the vehicle touches down in the desired location on the surface with zero velocity and in the proper orientation. Furthermore, we shall see that we can represent H in a more straightforward manner tied directly back to our system as follows

$$H(x, \lambda, u) = f(x, u) \cdot \lambda + M(x, u)$$

(1.19)

One additional note. Despite the fact that we define the Cost function as something we would strive to minimize through an optimal control, the Pontrygain Maximum Principle applies equally well in this context as maximizing a function is the mathematical equivalent of minimizing the negative of that function.

CHAPTER II

SCENARIO ONE: THE LUNAR LANDING PROBLEM

2.1   Framing the Scenario

In 1.4.2 we introduced the basics of the lunar landing problem. In this chapter, we will formalize the treatment of the problem, first by applying the maximum principle to confirm the form and existence of an optimal control, then to develop analytical and numeric solutions to the problem, and finally to examine the behavior of the optimally controlled system through a series of simulations and sensitivity analyses.
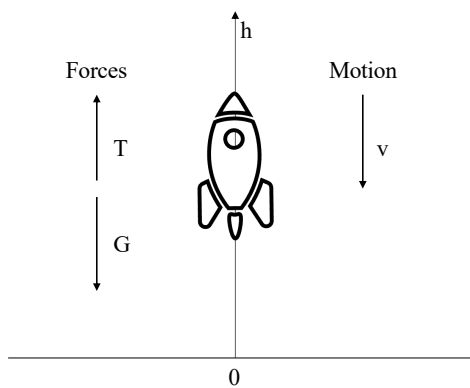
Figure 2.1: Lunar Landing Scenario

2.1.1   Applying the Maximum Principle to the Lunar Landing Problem

To illustrate how this works in practice, let us return to the lunar landing example from 1.4.2 using the variables from section 1.3.1. Recall that the equations of motion with control can be expressed as

$$f(\mathbf{x}, u) = \begin{bmatrix} \dot{h} \\ \dot{v} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v \\ \dfrac{u}{m} - g \\ -ku \end{bmatrix}$$

(2.1)

We can then define the control Hamiltonian as

$$\begin{aligned}
H(x, \lambda, u) &= v\lambda_1 + \left(\frac{u}{m} - g\right)\lambda_2 - ku\lambda_3 - u \\
&= v\lambda_1 + \frac{u}{m}\lambda_2 - g\lambda_2 - (k\lambda_3 + 1)u \\
&= v\lambda_1 - g\lambda_2 + (-k\lambda_3 + \frac{1}{m}\lambda_2 - 1)u
\end{aligned}$$

(2.2)

So, we can calculate

$$H_x = \begin{bmatrix} H_h \\ H_v \\ H_m \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda_1 \\ -\dfrac{u}{m^2}\lambda_2 \end{bmatrix}$$

(2.3)

and

$$\dot{\lambda}(t) = -H_x = -\begin{bmatrix} H_h \\ H_v \\ H_m \end{bmatrix} = \begin{bmatrix} 0 \\ -\lambda_1 \\ \dfrac{u}{m^2}\lambda_2 \end{bmatrix}$$

(2.4)

Now by applying the maximum principle, we know that the control, u^*, that maximizes H optimizes the cost function and therefore minimizes the fuel consumed. We can easily test for the maximum of H over u from

$$H_u = -k\lambda_3 + \frac{1}{m}\lambda_2 - 1 = 0$$

(2.5)

However, since $\mathcal{H}$ is linear in u, $\mathcal{H}$ is maximized at the endpoints of values of u, specifically

$$u = \begin{cases} u_{max} \text{ if } -k\lambda_3 + \dfrac{1}{m}\lambda_2 - 1 > 0 \\ u_{min} \text{ if } -k\lambda_3 + \dfrac{1}{m}\lambda_2 - 1 < 0 \end{cases}$$

(2.6)

It is at this point that we need to address the necessary condition for optimality in the use of the maximum principle. That condition is the convexity of the dynamics of the problem and the set of controls, $\mathcal{U}$, and the constraints imposed on it. We recall that a set, A, is convex if

$$\forall \, a_1, a_2 \in A \text{ then } sa_1 + (1-s)a_2 \in A \text{ where } s \in [0,1]$$

(2.7)

In the context of our exploration, this implies that $\mathcal{U} = \{u \mid u_{min} \le u \le u_{max}\}$ is convex. The reason for this is that if $\mathcal{U}$ is non-convex there is the potential that the calculations will yield local extrema (or possibly no extrema) that may not be global extrema. That is, we may calculate a control which is only optimal for a portion of the system instead of the entire system.

In our case, $\mathcal{U}$ as defined as not convex. This is seen as follows. Choose u_1 and u_2 as any two elements of $\mathcal{U}$ where $u_1 > u_2$. Consider $su_1 + (1-s)u_2$ where $0 \le s \le 1$. For $\mathcal{U}$ to be convex, then

$$u_{min} \le su_1 + (1-s)u_2 \le u_{max}$$

Now, let $\varepsilon = u_{min}$ so that

$$su_1 + (1-s)u_2 \ge \varepsilon$$

$$s \ge \frac{\varepsilon - u_2}{u_1 - u_2}$$

(2.8)

This says that as long as we select $s < \frac{\varepsilon - u_2}{u_1 - u_2}$ then we can make $\mathcal{U}$ non-convex. However, if $\varepsilon = u_{min} = 0$, then it is not possible to select such an s.

The physical interpretation of this non-convexity issue can be thought of as the fact that from an engineering perspective, there are concerns that it is not feasible for a rocket engine to be completely turned off and back on again during flight. Mathematically, this can be addressed in a number of ways. For example, (Acikmese & Ploen, 2013) introduce a slack variable to take the place of the thrust control which convexifies $\mathcal{U}$ at the cost of introducing another constraint. Alternatively, ignoring the engineering concerns in favor of a more straightforward approach used by (Evans, 1983) et. al., and one that we will adopt, is the assumption that engines are simple bang-bang controls such that $u \in \{u_{min}, u_{max}\}$ so that $u_{min} = 0$. It is easy to see that inserting these values into (2.8) confirms that $\mathcal{U}$ is convex.

The other challenge with convexity comes from the nature of the equations involving the controls. Convexification of these is core of the work by (Liu, 2018) and (Prous, 2020). However, because we are interested in exploring the just the descent phase of the vehicle's trajectory, we are going to make use of Taylor approximations to create the necessary conditions for the Maximum Principle to apply.

With these adjustments, we recognize that the control is now a switching function meaning the engines are turned on and off in some pattern to achieve optimality. Now, a review of (2.5) and (2.6) suggests that because H_u is independent of u, then there is only one switch during the optimal descent. This is something we will prove later in the solution.

To begin our analysis of the problem, we can assume that initially the thrusters are turned off and are turned on at some point. Intuitively, if we were to consider the opposite and have the

thrusters initially turned on then they would need to operate for the entire duration of the descent to counter gravity only turning off at the very last instant; this actually maximizes the amount of fuel consumed. So, with the engines initially turned off, we set $t = t^*$, to the time when they are turned on so that $v(t_f) = 0$

$$u^*(t) = \begin{cases} 0, \text{if } 0 \leq t < t^* \\ u_{max}, \text{if } t^* \leq t \leq t_f \end{cases}$$

(2.9)

For brevity, we will refer to u_{max} as u for the rest of this and subsequent chapters. So, if we consider the initial period where $0 \leq t < t^*$ then $u(t) = 0$ in (2.1) which gives

$$f(\mathbf{x}, u) = \begin{bmatrix} \dot{h} \\ \dot{v} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v \\ -g \\ 0 \end{bmatrix}$$

Which can easily be solved using the boundary, initial and terminal conditions as

$$h(t) = h_0 + tv_0 - \frac{1}{2}gt^2$$
$$v(t) = v_0 - gt$$
$$m(t) = m_0$$

(2.10)

Now, if we consider $t^* \leq t \leq t_f$ then we have

$$f(x, u) = \begin{bmatrix} \dot{h} \\ \dot{v} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v \\ -g + \dfrac{u}{m} \\ -ku \end{bmatrix}$$

Which can be solved in the order of m(t), then v(t), then h(t) from the initial conditions derived from (2.10) with $t = t^*$

$$m(t) \quad = \int(-ku)\, dt = -kut + c$$
$$= m_0 - ku(t - t^*)$$

29

$$v(t) \quad = \int \left(-g + \frac{u}{m}\right)dt = \int \left(-g + \frac{u}{m_0 - ku(t-t^*)}\right)dt$$

$$= -gt - \frac{1}{k}\ln\left(m_0 - ku(t - t^*)\right) + c$$

$$= v_0 - gt - \frac{1}{k}\ln\left(1 - \frac{ku}{m_0}(t - t^*)\right)$$

$$h(t) \quad = \int v\,dt = v_0 t - g\int t\,dt - \frac{1}{k}\int \left(\ln\left(1 - \frac{ku}{m_0}(t - t^*)\right)\right)dt$$

$$= v_0 t - g\int t\,dt - \frac{1}{k}\int \left(\ln\left(1 - \frac{ku}{m_0}(t - t^*)\right)\right)dt$$

$$= v_0 t - \frac{1}{2}gt^2 - \frac{1}{k}\left(t - t^* - \frac{m_0}{ku}\right)\ln\left(1 - \frac{ku}{m_0}(t - t^*)\right) + \frac{1}{k}t + c$$

$$= h_0 + v_0 t - \frac{1}{2}gt^2 + \frac{1}{k}(t - t^*) - \frac{1}{k}\left(t - t^* - \frac{m_0}{ku}\right)\ln\left(1 - \frac{ku}{m_0}(t - t^*)\right)$$

$$(2.11)$$

Now, if we write $h(t) = f(v(t))$ from both (2.10) and (2.11) we have

$$h(t) = h_0 - \frac{1}{2g}(v(t)^2 - v_0^2) \text{ for } 0 \le t \le t^*$$

and

$$h(t) \approx \alpha(v(t)^2) + \beta(v(t)) + \chi(\ln(t)) \text{ for } t^* \le t \le t_f$$

where α, β and χ represent functions derived from substituting $v(t)$ into $h(t)$. We can then note that over both time intervals $h(t)$ they are parabolic in shape with respect to $v(t)$. So, we can see that by switching the engines on at $t = t^*$ we shift from one parabola (the uncontrolled system) to the other (the controlled system) in manner described in Example 1: The Rocket Car and illustrated in Figure 1.2.

In order to confirm that our initial assumption of the optimal control, we examine the original system in terms of the costate variables, λi. Recall that,

$$\lambda(t) = -H_x = -\begin{bmatrix} H_h \\ H_v \\ H_m \end{bmatrix} = \begin{bmatrix} 0 \\ -\lambda_1 \\ \frac{u}{m^2}\lambda_2 \end{bmatrix}$$

and set initial conditions $\lambda_1(0) = \lambda_{1_0}$, $\lambda_2(0) = \lambda_{2_0}$, and $\lambda_3(0) = \lambda_{3_0}$ which together yields

$$\lambda_1(t) = \lambda_{1_0} \text{ for } (0 \le t \le t_f)$$

$$\lambda_2(t) = \lambda_{2_0} - t\lambda_{1_0} \text{ for } (0 \le t \le t_f)$$

$$\lambda_3(t) = \begin{cases} \lambda_{3_0} \text{ for } (0 \le t \le t^*) \\ \lambda_{3_0} + \int_{t^*}^{t_f} \dfrac{\lambda_{2_0} - s\lambda_{1_0}}{(m_0 + k(t^* - s))^2} \, ds \text{ for } (t^* \le t \le t_f) \end{cases}$$

Now, recall from (2.5) that

$$H_u(t) = -k\lambda_3 + \frac{1}{m}\lambda_2 - 1 \tag{2.12}$$

so

$$\dot{H}_u(t) = -k\dot{\lambda}_3 + \frac{m\dot{\lambda}_2 - \dot{m}\lambda_2}{m^2} = -k\frac{u}{m^2}\lambda_2 + \frac{\lambda_1}{m} + \frac{ku\lambda_2}{m^2} = \frac{\lambda_1}{m} = \frac{\lambda_{1_0}}{m}$$

This means that $H_u(t)$ is either increasing or decreasing throughout the descent. So, we can assume that it is increasing throughout. Now, we can evaluate $H_u(t^*)$ as

$$H_u(t^*) = -k\lambda_{3_0} + \frac{1}{m}(\lambda_{2_0} - t^*\lambda_{1_0}) - 1$$

and we can choose λ_{1_0}, λ_{2_0}, and λ_{3_0} so that $H_u(t^*) = 0$ which means that

$$H_u(t) \begin{cases} < 0, 0 \le t \le t^* \\ > 0, t^* \le t \le t_f \end{cases}$$

which corresponds to and confirms that the optimal control has only one switch at $t = t^*$.

2.1.2 Understanding the Impact of the Optimal Control

Now that we have determined that the optimal descent trajectory involves one switch of the control thrusters from off to on, we would naturally want to understand two critical aspects of the solution:

1. At what time, t^*, should the control be switched?

2. What is the set of initial conditions, h_0, v_0, and m_0 that allow for an optimal descent?

To determine the allowable set of initial conditions, we can compare the trajectory of a free-falling vehicle with that of a fully controlled vehicle. In the first case, $t^* = t_f$ in and in the second, $t^* = 0$ in. The former substitution characterizes the vehicle in free-fall through the entire duration of its descent. The latter provides for the trajectory under complete control with a soft landing. In this case,

$$v(t_f) = v_0 - gt_f - \frac{1}{k}\ln\left(1 - \frac{kut_f}{m_0}\right) = 0$$

(2.13)

$$h(t_f) = h_0 + v_0t_f - \frac{1}{2}gt_f^2 + \frac{1}{k}t_f - \frac{1}{k}\left(t_f - \frac{m_0}{ku}\right)\ln\left(1 - \frac{kut_f}{m_0}\right) = 0$$

(2.14)

Inserting (2.13) into (2.14) allows us to arrive at a quadratic expression for the initial height

$$h_0 = -\frac{1}{2}gt_f^2 + \frac{1}{k}\left(1 - \frac{gm_0}{u}\right)t_f - \left(\frac{v_0m_0}{ku}\right)$$

(2.15)

Both (2.13) and (2.15) can be further simplified by approximating the value of the logarithmic term with a Taylor expansion. Doing this allows us to develop an analytical foundation for approximating the switching time for the engines. We will replace this with

numerical techniques during our simulations later in this chapter. For now, the we can approximate as follows:

$$\ln\left(1 - \frac{kut_f}{m_0}\right) \approx -\frac{ku}{m_0}t_f - \frac{(ku)^2}{2m_0^2}t_f^2$$

(2.16)

By inserting this estimate into (2.13) we can derive an expression for the initial condition for height and velocity as

$$h_0 = \frac{1}{2}\alpha t_f^2 \text{ where } \alpha = \frac{u}{m_0} - g$$

$$v_0 = \beta t_f^2 - \alpha t_f \text{ where } \beta = \frac{-ku^2}{2m_0^2}$$

There are couple of observations from these results. The first is that the thrust-to-mass ratio of the engines must exceed the acceleration due to gravity, that is $\frac{u}{m_0} - g > 0$. Intuitively this makes sense as if the engines are not strong enough to overcome gravity, a soft landing is not possible. Secondly, if the initial conditions are trivial, in that the vehicle is stationary on the surface, then t_f defaults to 0 which again makes (uninteresting) sense.

Combining these two results for h_0 and v_0 gives us

$$0 = v_0 + \sqrt{2\alpha h_0} - \frac{2\beta}{\alpha}h_0 = S(h_0, v_0, m_0, u)$$

(2.17)

where we define $S(h_0, v_0, m_0)$ as the so-called switching function which can be interpreted as the function whose roots form the set of all possible initial conditions (h_0, v_0, m_0) for achieving a soft landing based on the limits of the control (u_{min}, u_{max}). Geometrically, the intersection of the curves S for values of $u = 0$ (fully free fall) and $u = 1$ (fully controlled) represents the point at which the control is switched on. This was previously illustrated in Example 1: The Rocket Car.

33

At this point we turn to programs developed in MATLAB to model the scenario. (See Appendix A for details of the code used). For ease of illustration, we are using the following values: $h_0 = 1000$m, $v_0 = 0$m/s, $m_0 = 100$kg, $g = 1.6$m/s^2, $u = 180$ kg·m/s^2, $k = .01$s/m. Note that the value of g is a close approximation to the acceleration due to gravity at the surface of the moon. The basic logic flow used in the MATLAB programming follows four main steps:

1. Establish the values for the various parameters, initial and terminal conditions

2. Calculate the trajectory for initial descent with the control engine turned off and the trajectory required for a soft landing with the engine turned on using ODE45

3. Determine the point of intersection of the two trajectories using an algorithm from (Schwarz, 2017) and assemble the combined optimal descent trajectory

4. Plot the results

(2.18)

In Figure 2.1 we can imagine the vehicle traveling down and to the left on the free-fall curve from its initial position at (0, 1000) until it intersects with the switching function at which point the engines are turned on and the vehicle's trajectory changes under the influence of the control down and to the right until it reaches the soft-landing position at (0,0). The engines are fired at 26.1s with the vehicle at 454m above the surface traveling at a velocity of 41.8m/s downwards. The total duration of the descent is 60.6s and the mass of the vehicle at soft landing is 38.8kg. Figure 2.2 shows the impact of the controlled engine burn on the mass of the vehicle. The actual height and velocity trajectories of the vehicle are depicted in Figure 2.3 where we can easily see how the acceleration of the vehicle changing to positive causing the descent trajectory curve to invert allowing both h(t) and v(t) to approach 0 at the end of the descent.
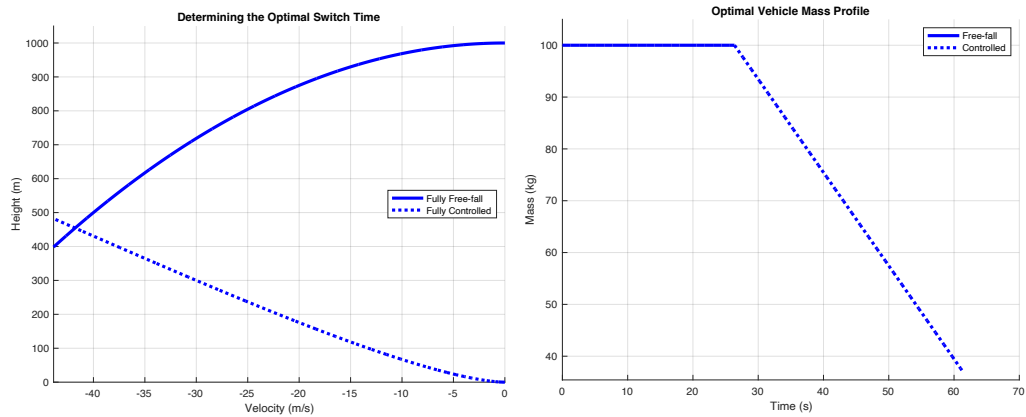
Figure 2.2: Optimal Switching Time and Mass Profile
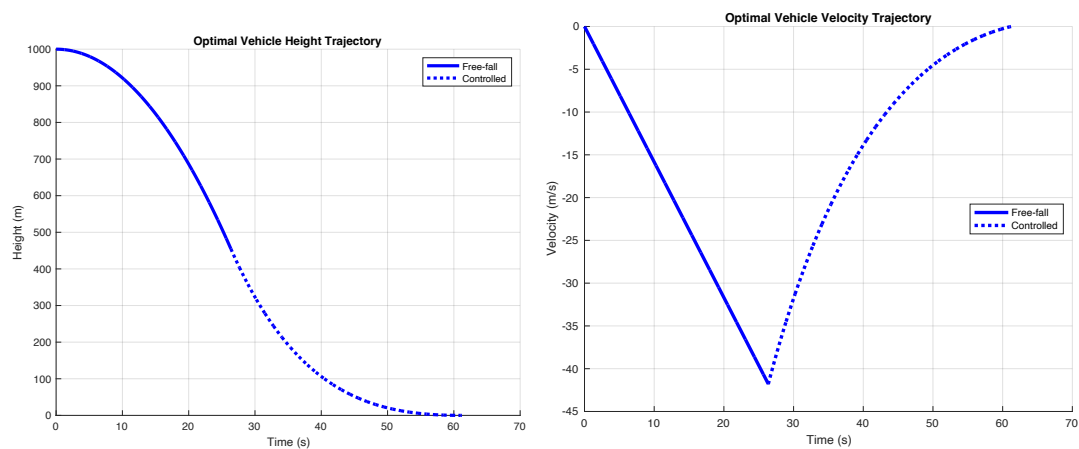
Figure 2.3: Optimal Descent Trajectories

2.2 Observations, Sensitivities and Conclusions

2.2.1 Observations

The results of section provide a complete review of the key elements of the use of optimal control theory applied to the problem of fuel-optimal descent trajectories for space vehicle landings. We saw that these kinds of problems are grounded in governing equations, in

this case Newtonian laws of motion. The introduction of the control variable and the application of the Maximum Principle gave rise to the introduction of the control Hamiltonian and the co-state variables. This allowed us to speculate and then prove that the vehicle's descent can be fuel-optimized through a single switching of the control engines from off to on during the descent. Finally, we were able to show that the optimal time for this switch is a function of the initial conditions as well as the maximum magnitude of the engine.

2.2.2 Sensitivities

Based on these observations, we can explore the sensitivity of the system to controllability. However, there are two things we must confirm in order to ensure that the solutions we are deriving are real.

The first thing we need to confirm is that the logarithmic term that appears throughout the equations beginning in

(2.11) only produces real results. That is $m_0 - kut > 0$. This is accomplished through the recognition that there is an upper bound on the amount of fuel that can be consumed. If we let the m_e represent the mass of the vehicle without fuel, then $m_0 - m_e \geq kut$. Which means that the vehicle will run out of fuel before a mathematical problem is created.

The second requirement is that $\alpha = \frac{u}{m_0} - g$ that appears in (2.17) is positive. This term represents the difference between the maximum force of the engines and that of gravity. It is easy to see that if the engines are not strong enough to overcome gravity's pull, the vehicle will never be able to achieve a soft landing. Interestingly, we shall see in Chapter 3 that the presence of an atmosphere requires less powerful engines as drag forces are able to assist in the slowing of the vehicle.

With these resolved, the first sensitivity we will explore is that which is associated with

the Taylor approximation in (2.16) which was derived from the standard Taylor approximation

for ln(1-x) at x = 0 where, in this case, $x = \frac{ku}{m_0}t$. So, the question we have is, for how long does

the order 2 Taylor approximation in (2.16) remain acceptable? To simplify our examination, let

$\mu = \frac{ku}{m_0}$ which is constant for the duration of the engine burn. So, this gives us that

$$\ln\left(1 - \frac{kut}{m_0}\right) = \ln(1 - \mu t) \approx -\mu t - \frac{(\mu t)^2}{2} - \frac{(\mu t)^3}{3} - \cdots = -\sum_{i=1}^{\infty} \frac{(\mu t)^i}{i}$$

If we assume, for example, that when fully fueled, half of the mass of the vehicle is fuel

then $\mu t \leq \frac{1}{2}$, and the maximum error derived from the second-degree approximation is

$$\left|\ln\left(\frac{1}{2}\right) + \frac{1}{2} + \frac{1}{2 \cdot 2^2}\right| = 0.06815$$

and for the third-degree approximation the maximum error is

$$\left|\ln\left(\frac{1}{2}\right) + \frac{1}{2} + \frac{1}{2 \cdot 2^2} + \frac{1}{3 \cdot 2^3}\right| = 0.02648$$

Although an understanding of this error size is helpful in determine the relative accuracy

of the approximations to the analytical solution, the use of MATLAB as a means for more

precisely calculating the optimal descent trajectory.

The second area of sensitivity exploration involves the effect of the initial conditions,

specifically, at what height (h_0) and with what initial velocity (v_0), should a vehicle of a given

initial mass (m_0), engine power (u), and engine efficiency (k) begin its descent. Table 2-1, Table

2-2 and Table 2-3 show the results of MATLAB simulations on the amount of fuel consumed in

the optimal descent trajectory for each pair initial and given conditions.

Table 2-1: Initial Height and Velocity Simulation Results

Fuel Used (kg)		Initial Height (m)				
		900	950	1000	1050	1100
Initial Velocity (m/s)	0	64.5	63.6	62.7	61.9	61.1
	-5	64.6	63.7	62.8	62.0	61.2
	-10	65.0	64.1	63.2	62.3	61.5
	-15	65.6	64.7	63.7	62.8	62.0
	-20	66.5	65.4	64.4	63.5	62.6

Table 2-2: Initial Height and Maximum Thrust Simulation Results

Fuel Used (kg)		Initial Height (m)				
		900	950	1000	1050	1100
Max Thrust (N)	160	15.3	15.0	14.8	14.6	14.4
	170	46.9	46.2	45.5	44.9	44.3
	180	64.5	63.6	62.7	61.9	61.1
	190	77.0	76.0	75.0	74.1	73.2
	200	87.0	85.9	84.8	83.8	82.9

Table 2-3: Initial Height and Engine Exhaust Speed Simulation Results

Fuel Used (kg)		Initial Height (m)				
		900	950	1000	1050	1100
Exhaust Speed (m/s)	90	67.7	66.7	65.8	64.9	64.1
	95	66.1	65.1	64.2	63.4	62.6
	100	64.5	63.6	62.7	61.9	61.1
	105	63.0	62.1	61.3	60.5	59.8
	110	61.6	60.7	59.9	59.2	58.5

These results lead us to some further observations:

- Increasing the initial (downward) velocity of the vehicle does not reduce the amount of fuel consumed and is thus, suboptimal; this is because the engines need to fire sooner in order to slow the vehicle to a soft landing

- The data also confirms the findings by (Brodkin, 2021) that under the negligible atmosphere conditions in this scenario the minimum fuel problem is the same as the minimum time problem for controlled descent

- The characteristics of the engines – both its maximum thrust (u) and the efficiency if the engines (k) – play an important role in the optimal descent trajectory; for example, experimentation in MATLAB illustrates that a 20% increase in the maximum thrust of the engines and a 20% increase in the efficiency of the engine can reduce fuel consumption by 3%; whether or not this is a good economic tradeoff is obviously dependent on the relative costs associated with such fuel consumption and engineering improvements

The insights we gain from this sensitivity analysis will guide us as we look more closely into optimal descent trajectories in scenarios involving drag in Chapter 3 and then drag and wind in Chapter 4.

2.2.3 Conclusions

Although the lunar landing scenario with its overall simplicity derived from the lack of a measurable atmosphere does have applicability, it is also instructive in giving us clues to understanding the more complex scenarios in the chapter that follow. More specifically, we will be looking for:

1. The format of the control Hamiltonian and in particular, its first derivative with respect to the control variable(s); this will give us an important clue as to the nature of the operation of the control(s)

2. Whether the dynamics of the scenario will provide the necessary convexity to ensure that the Maximum Principle will deliver and optimal control solution

3. The degree to which Taylor approximation will play a role in the derivation of a switching function to allow us to model the operation of the control(s)

4. How the presence of a measurable atmosphere (in various formats) impacts our ability to understand how initial conditions impact the optimal descent trajectories.

CHAPTER III

SCENARIO TWO: ATMOSPHERIC DRAG

3.1   Framing the Scenario

In

CHAPTER II we were able to illustrate the application of optimal control theory

techniques to explore the optimal descent trajectory of a vehicle on the surface of the moon.

Analysis of the lunar landing scenario benefits greatly from the mathematical simplicity that

follow from the material absence of an atmosphere. In this and subsequent chapters we will

leverage these to explore additional more realistic – and more complicated – scenarios.

The first additional scenario we will explore is the introduction of an atmosphere material

enough to exert drag on the vehicle as it descends. As the drag force is in the opposite direction

of motion, we can continue to view this scenario in one (vertical) dimension.   The decent will be

controlled in the same manner as in the lunar landing through an engine that can exert vertical

thrust to oppose gravity. In this scenario, however, the engine is aided by the drag force which

increases as the downward speed of the vehicle increases. Our objective remains to minimize the

amount of fuel consumed to enable a soft landing.

3.2   Modeling the System

The system in this scenario will be modeled using the same set of variables as in Chapter 2 with the addition of the variables and parameters required to represent drag. Based on this, the applicable equations of motion are:

$$\dot{h}(t) = v(t)$$

$$\dot{v}(t) = \frac{1}{m(t)}(T(t) + D(t)) - G(h)$$

$$\dot{m}(t) = -\frac{T(t)}{G(0)I_{sp}}$$

<div align="right">(3.1)</div>

where:

h(t) is the height of the vehicle above the landing surface

v(t) is the velocity of the vehicle

G(h) is the gravitational force acting on the vehicle

m(t) is the mass of the vehicle

T(t) is the thrust (force) magnitude

D(v) is the atmospheric drag force on the vehicle

I_{sp} is specific impulse, or thrust efficiency, of the engines

Further examination of the drag force acting on the vehicle as noted by (NASA, 2021) yields:

$$D(v) = \frac{1}{2}\rho(h)v^2 A_D C_D$$

<div align="right">(3.2)</div>

Our first observation is that the Drag force, D, varies quadratically with the speed of the vehicle which, in turn, varies with time. Secondly, in both (3.1) and (3.2), $\rho(h)$ is the atmospheric density and varies with altitude along with other factors such as temperature. We will explore this nature of how the modeling of atmospheric density affects the optimal descent

in section 3.4. Finally, C_D is the coefficient of drag and A_D is the cross-sectional area of the vehicle against which the drag force is applied.

In terms of initial and final conditions of the problem. It is clear that for a safe landing we require that the vehicle's velocity be zero at the end of the descent. Furthermore, we also want the vehicle to land in a specific location on the landing surface directly below the initial point consistent with a one-dimensional problem. Lastly, we want the amount of fuel consumed in the landing exercise to be as small as possible. Figure 3.1 illustrates the forces, motion and orientation of the scenario.
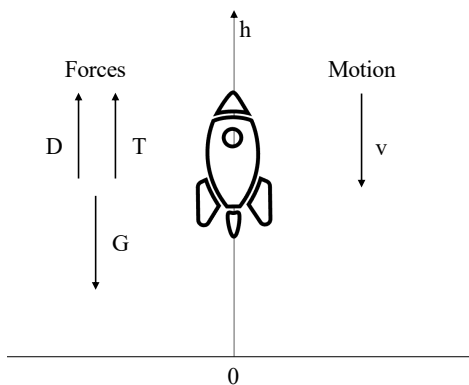
Figure 3.1: Atmospheric Drag Scenario

3.3   Discerning the Appropriate (Optimal) Control

As we pointed out in sections 1.3.3, to construct an optimal control problem we require two additional elements: the cost function we are trying to optimize and the controls we are using to do so. Starting with the cost function, we noted that we are trying to minimize the amount of fuel consumed in the descent trajectory. This is equivalent to wanting to maximize the mass of the vehicle at landing. So, for convenience, we will choose the latter perspective with the control

at our disposal, as in 2.1.2, being the magnitude of the thrust, T. Recognizing that there are physical and engineering limits to these controls, we will define upper and lower bounds for each of them as follows:

$$T_{min} \leq T \leq T_{max}$$

Finally, we will complete the model with a few additional assumptions that will allow for a more focused analysis of the outcomes. Specifically:

- The force of gravity will be assumed to be constant over the entire descent (g)

- We can then combine G_0 and I_{sp} into a single parameter, $\frac{1}{k}$, which is reciprocal of the velocity at which exhaust is expelled by the engines (in the opposite direction of the thrust)

- The coefficient of drag, C_D, which in reality is a function of the Mach number of the vehicle and angle of descent will also be assumed to be constant

- Similarity, the cross-sectional areas of the vehicle affected by the drag force will also be assumed to be constant

Taking these latter two into account, we will redefine (1.9) and (1.11) as follows:

$$D = \rho \delta v^2, \text{where } \delta = \frac{1}{2} A_D C_D$$

(3.3)

So, taking all of this into account, the mathematical model we will use to explore optimal descent trajectories for safely landing a vehicle on a moon or planet is

$$f(\mathbf{x}, u) = \begin{bmatrix} \dot{h} \\ \dot{v} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v \\ \dfrac{1}{m}(u + \rho\delta v^2) - g \\ uk \end{bmatrix}$$

$$\tag{3.4}$$

where we want to determine u^* such that

$$C(u^*) \leq C(u)$$

subject to

$$u_{min} \leq u \leq u_{max}$$

where we want to minimize

$$C(u) = \int_0^{t_f} \dot{m}\, ds = m(0) - m(t_f)$$

$$\tag{3.5}$$

so that

$$0 \leq t \leq t_f$$

$$h(0) = h_0 \text{ and } h(t_f) = 0$$

$$v(0) = v_0 \text{ and } v(t_f) = 0$$

$$m(0) = m_0 \text{ and } m(t_f) = m_f$$

$$u(t_f) = 0$$

$$\tag{3.6}$$

In this case, we are trying to minimize the fuel consumed or, alternatively, maximize the remaining fuel in the vehicle. Assuming that the vehicle does not lose mass in any other way during the descent, that gives us a cost function as shown in (3.5). This allows us to introduce the costate variables λ_1, λ_2, and λ_3 as follows to define the control problem Hamiltonian as

$$H(x, \lambda, u) = v\lambda_1 + \left(\frac{1}{m}(u + \rho\delta v^2) - g\right)\lambda_2 - uk\lambda_3 - u$$

$$= v\lambda_1 + \frac{\lambda_2}{m}(u + \rho\delta v^2) - g\lambda_2 - (k\lambda_3 + 1)u$$

$$= v\lambda_1 - g\lambda_2 + \frac{\lambda_2}{m}\rho\delta v^2 + (-k\lambda_3 + \frac{1}{m}\lambda_2 - 1)u$$

(3.7)

We note that the Hamiltonian in this scenario defaults to that in the Lunar Landing problem

from Chapter 2 when $\rho = 0$ which is the case when the atmosphere is negligible.

From (3.7) we can proceed as follows

$$H_h = \frac{k\delta v^2}{m}\left(\frac{d\rho}{dh}\right)\lambda_2$$

$$H_v = \lambda_1 + \frac{2}{m}\rho\delta v\lambda_2$$

$$H_m = -\frac{1}{m^2}(u + \rho\delta v^2)\lambda_2$$

(3.8)

and

$$\dot{\lambda}_1 = -H_h = -\frac{v^2\delta}{m}\left(\frac{d\rho}{dh}\right)$$

$$\dot{\lambda}_2 = -H_v = -\lambda_1 - \frac{2}{m}\rho\delta v$$

$$\dot{\lambda}_3 = -H_m = \frac{1}{m^2}(u + \rho\delta v^2)\lambda_2$$

(3.9)

and

$$H_u = -k\lambda_3 + \frac{1}{m}\lambda_2 - 1$$

(3.10)

If we compare (3.8), (3.9) and (3.10) to (1.18), (1.19) and (2.3) , (2.4) we note two

important observations. Firstly, the presence of an atmosphere changes the dynamics affecting

the Hamiltonian and the costate variables. Secondly, the Hamiltonian remains linear in u. This is an important result because it allows us to speculate that the optimal control for the system is again a switching function. As such, we can again choose to start from an assumption for u* based on (2.2) as follows

$$u^*(t) = \begin{cases} 0 \text{ if } 0 \leq t \leq t^* \\ u \text{ if } t^* \leq t \leq t_f \end{cases}$$

(3.11)

3.4   The Impact of Atmospheric Density

As we have seen, the presence of a non-negligible atmosphere introduces non-linearity into our system.   The v^2 term in the representation of the drag forces changes the nature of the equations of motion (3.4), the Hamiltonian (3.7), some of its partial derivatives (3.8), and the co-state variables (3.9). The key characteristic of the atmosphere in this scenario is the density of the atmosphere as the vehicle passes through it. This is also referred to as atmospheric pressure. As noted by (NASA, 2021) under most conditions, ρ is primarily a function of elevation from the surface. So, the degree to which ρ impacts the solution depends on the thickness of the atmosphere and the height above the surface at which we begin the descent. There are several different ways of modeling atmospheric density. In this paper, we will consider two. One where the density is constant and the other where the density is a known function of the altitude above the surface.

3.4.1   Case 1 – Constant Atmospheric Density

We will begin by assuming the conditions are such that ρ is constant. That is,

$$\rho(h) = \rho(0) = \rho_0$$

<div align="right">(3.12)</div>

Thus, in this case (3.8) and (3.10) become

$$H_h = 0$$

$$H_v = \lambda_1 + \frac{2}{m}\rho_0\delta v$$

$$H_m = -\frac{k}{m^2}(u + \rho_0\delta v^2)\lambda_2$$

<div align="right">(3.13)</div>

and

$$\dot{\lambda}_1 = -H_h = 0$$

$$\dot{\lambda}_2 = -H_v = -\lambda_1 - \frac{2}{m}\rho_0\delta v$$

$$\dot{\lambda}_3 = -H_m = \frac{k}{m^2}(u + \rho_0\delta v^2)\lambda_2$$

<div align="right">(3.14)</div>

and as we noted above, H_u is unchanged from (3.10).

One can easily see that if $\rho_0 = 0$ which is the case on the moon, then all of the equations in this scenario simplify to those in Section 2.1.1 and 2.1.2. As such, we can proceed with attempting to solve for the optimal control and the dynamics of the system in a similar manner.

For the initial period of $0 \le t \le t^*$, $u(t) = 0$ and becomes

$$f(\mathbf{x}, u) = \begin{bmatrix} v \\ \dfrac{1}{m_0}\rho_0\delta v^2 - g \\ 0 \end{bmatrix}$$

<div align="right">(3.15)</div>

Now, we can easily see that $m(t) = m_0$ as it was in the lunar landing. However, the solutions for $v(t)$ and $h(t)$ require additional work. Starting with $v(t)$ we have

$$\dot{v} = \frac{1}{m_0} \rho_0 \delta v^2 - g$$

$$(3.16)$$

For simplicity, set $a := \frac{1}{m_0} \rho_0 \delta$ so that (3.16) becomes

$$\dot{v} = av^2 - g$$

$$(3.17)$$

which reveals three possibilities:

1. If $av^2 - g > 0$, then the vehicle must be accelerating upwards and against the force of gravity; since the engines are off, then this is not possible.

2. If $av^2 - g = 0$, then the vehicle has reached so-called terminal velocity which we will designate as v_t and remains constant until the thrusters are applied.

3. If $av^2 - g < 0$ for this entire time interval we can recognize (3.17) as a Riccati Equation with constant coefficients that we can then rearrange to

$$\int \frac{dv}{av^2 - g} = \int dt$$

$$(3.18)$$

Now, let $b^2 := \frac{a}{g} > 0$ so we can simplify the integrand on the left-hand side

$$\frac{1}{av^2 - g} = \frac{\frac{1}{g}}{b^2 v^2 - 1} = \frac{\frac{1}{g}}{(bv - 1)(bv + 1)} = \frac{\frac{1}{2g}}{(bv - 1)} - \frac{\frac{1}{2g}}{(bv + 1)}$$

and so (3.18) becomes

$$\int \frac{1}{(bv - 1)} dv - \int \frac{1}{(bv + 1)} dv = 2g \int dt$$

and after integrating both sides with some slight rearranging

$$\ln\left(\frac{(bv-1)}{(bv+1)}\right) = -2\tanh^{-1}(bv) = 2bgt + C$$

which simplifies to

$$\tanh^{-1}(bv) = -bgt - C'$$

(3.19)

and since $v(0) = v_0$ we have $C' = -\tanh^{-1}(bv_0)$

Now we can again rearrange and take the hyperbolic tangent of both sides of (3.19) which results in

$$v = -\frac{1}{b}\tanh(bgt - C') = -\frac{1}{b}\tanh(bgt - \tanh^{-1}(bv_0))$$

(3.20)

By replacing all of the temporary parameters we arrive at

$$v(t) = -\sqrt{\frac{gm_0}{\rho_0\delta}}\tanh\left(\sqrt{\frac{\rho_0\delta}{gm_0}}gt - \tanh^{-1}\left(\sqrt{\frac{\rho_0\delta}{gm_0}}v_0\right)\right)$$

(3.21)

With an expression for v(t) we are able to calculate an expression for h(t) from (3.15) where,

$$h(t) = -\frac{1}{b}\int \tanh(bgt - \tanh^{-1}(bv_0))\, dt$$

$$= -\frac{1}{b^2}\ln(\cosh(bgt - \tanh^{-1}(bv_0))) + K$$

Initially, $h(0) = h_0$ so,

$$K = h_0 + \frac{1}{b^2}\ln(\cosh(\tanh^{-1}(bv_0))) = h_0 + \frac{1}{b^2}\ln\left(\frac{1}{\sqrt{1-(bv_0)^2}}\right)$$

which delivers

$$h(t) = -\frac{1}{b^2}\ln(\cosh(bgt - \tanh^{-1}(bv_0))) + h_0 + \frac{1}{b^2}\ln\left(\frac{1}{\sqrt{1 - (bv_0)^2}}\right)$$

(3.22)

Now let us attempt to reconcile this result with that in Chapter 2 by considering the

situation as $\rho_0 \to 0 \Rightarrow a \to 0 \Rightarrow b \to 0$. That is

$$\lim_{b\to 0} v = \lim_{b\to 0}\left(-\frac{1}{b}\tanh(bgt - C')\right)$$

(3.23)

Using the Taylor expansion of tanh at 0

$$\tanh(bgt - C') = (bgt - C') - \frac{1}{3}(bgt - C')^3 + \frac{2}{15}(bgt - C')^5 - + \cdots$$

which implies

$$-\frac{1}{b}\tanh(bgt - C') = \left(-gt + \frac{1}{b}C'\right) + \frac{1}{3b}(bgt - C')^3 - \frac{2}{15b}(bgt - C')^5 - + \cdots$$

Recall from $C' = \tanh^{-1}(bv_0)$ which also has a Taylor series at 0

$$C' = \tanh^{-1}(bv_0) = (bv_0) + \frac{(bv_0)^3}{3} + \frac{(bv_0)^5}{5} + \cdots$$

and

$$-\frac{1}{b}C' = -v_0 - \frac{(bv_0)^3}{3b} - \frac{(bv_0)^5}{b} + \cdots$$

which we can see tends to -v_0 as b tends to 0. Thus,

$$\lim_{b\to 0} v(t) = -gt - v_0$$

This is the same result we derived in the scenario where the atmosphere was negligible.

In a similar manner we can attempt to show that the same simplification applies to h(t).

Although it is possible to do this from (3.22), we can use the fact that (3.21) is both continuous and uniformly convergent to confirm that

$$\int \lim_{b \to 0} v(t)\, dt = \lim_{b \to 0} \int v(t)\, dt$$

thus,

$$\lim_{b \to 0} h = h_0 + tv_0 - \frac{1}{2}gt^2$$

which corresponds to (2.3).

We now turn to the period of time where $t^* \leq t \leq t_f$ and (3.16) becomes

$$f(\mathbf{x}, \mathbf{u}) \;=\; \begin{bmatrix} v \\ \frac{1}{m}(u + \rho_0 \delta v^2) - g \\ -ku \end{bmatrix}$$

As before, we will begin with finding m(t) which turns out to be straight forward in that it is the same as in (2.5).

$$m(t) = m_0 - ku(t - t^*) \tag{3.24}$$

Solving for the velocity is significantly more difficult as we have

$$\dot{v} = \frac{1}{m}(u + \rho_0 \delta v^2) - g = \frac{\rho_0 \delta v^2}{m} + \frac{u}{m} - g$$

Now, let $R(t) := \frac{\rho_0 \delta}{m}$ and $P(t) := \frac{u}{m} - g$ so we have another Riccati equation in form

$$\dot{v} = R(t)v^2 + P(t)$$

This is a simpler version of the general Riccati equation $\dot{v} = R(t)v^2 + Q(t)v + P(t)$

where $Q(t) = 0$. Solutions to this category are offered in a variety of texts including (Al Bastami,

Belic, & Petrovic, 2010) in which a substitution of $v = -\frac{\dot{z}}{Rz}$ yields

$$\ddot{z} - \frac{\dot{R}}{R}\dot{z} + PRz = 0$$

(3.25)

Noting that $\dot{R} = \frac{\rho_0 \delta ku}{(m_0 - ku(t-t^*))^2}$ and substituting the values for R and P from above

transforms (3.25) into

$$\ddot{z} - \frac{\rho_0 \delta ku}{m^2}\frac{m}{\rho_0 \delta}\dot{z} + \frac{\rho_0 \delta}{m}(\frac{u}{m} - g)z = 0$$

$$\ddot{z} - \frac{ku}{m}\dot{z} + \frac{\rho_0 \delta}{m}(\frac{u}{m} - g)z = 0$$

(3.26)

Although it is possible to advance (3.26) further analytically, at this point we will move

to numeric solutions using MATLAB following the same logic flow described in (2.18) that was

used in the Lunar Landing Scenario by adapting the equation for the vertical acceleration solved

by ODE45. In order to develop a broader and more realistic picture of the impact of atmospheric

density, the analytical results of this section were examined numerically for three different

settings: Mars, Earth and an imaginary planet, Prueba, with comparable characteristics of

gravitational acceleration and atmospheric density.

Table 3-1 contains the values used in the simulations for all three planets along with the

colors used to plot the results for ease of recognition. The initial conditions were kept constant

from the simulations used in Chapter 2, specifically $h_0 = 1000m$, $v_0 = 0m/s$, $m_0 = 100kg$, $u = 180$

kg·m/s^2, and k = .01s/m. The value of the unitless drag coefficient, d, was set to 0.1 in line with

the NASA estimates for a tapered cylinder as a representative shape of the vehicle.

Table 3-1: Planetary Parameters for Atmospheric Drag Simulations

Planet	Density (kg/m^3)	Gravity (m/s^2)	Graph Color
Mars	0.020	3.721	red
Earth	1.293	9.807	blue
Prueba	0.870	15.00	green

The output from the simulations on all three planetary models are plotted using the same

four graph frames used in

CHAPTER II. The first observation we can make from these results is that the

atmosphere helps. The drag imparted by the atmosphere assists in slowing the vehicle down,

which in turn means that the engines don't need to come on as high in the descent as when there

is no atmosphere.   This means that the engines do not need to run as long in order to bring the

vehicle to a soft landing resulting in less fuel consumed.
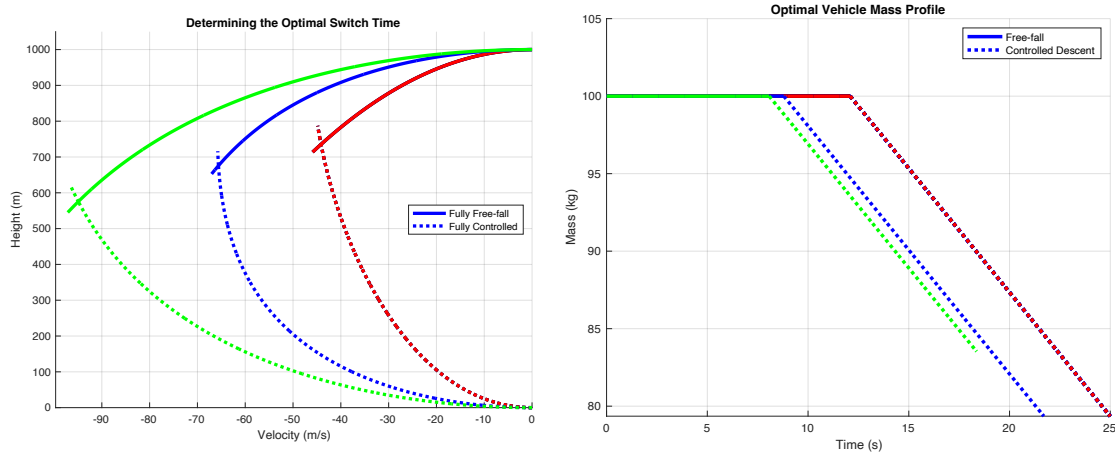
Figure 3.2: Optimal Switching Times and Mass Profiles with Atmospheric Drag
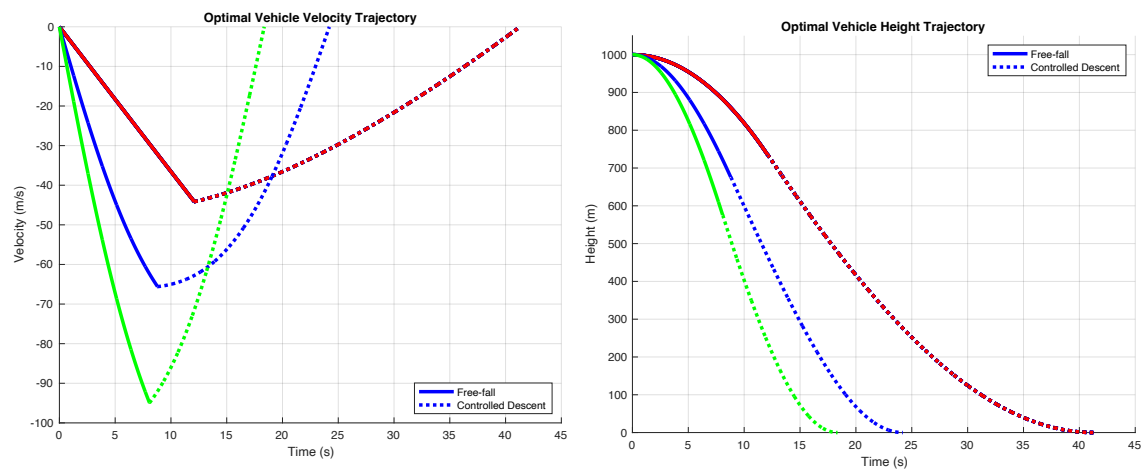
Figure 3.3: Optimal Descent Trajectories with Atmospheric Drag

A closer look at the switching heights and velocities in Figure 3.2 gives us further insight. We can clearly see that the switching time occurs much higher in the descent to Mars than for Earth or Prueba. This is a due primarily to the relative strength of the two natural forces affecting the vehicle: gravity and drag. On Mars, although the gravity is approximately a third of that on Earth, the atmospheric density is less that 2% of Earth's. As a result, the drag force is of

relatively little help to the engines compared to Earth. The same comparison can be made between Earth and Prueba. In this situation, we see that Prueba actually creates conditions for the latest switching time. This is a result of stronger gravity which accelerates the vehicle more rapidly than Earth's gravity. However, the faster the free-fall, the more it is affected by the atmospheric density as drag force varies directly with square of the velocity. In this case, the two forces actually work together to the benefit of the fuel optimization objective resulting the longest time before switching and hence, the shortest time with the engines burning.

Figure 3.3 rounds out the depiction of the comparison of the descent trajectories and reveal another critical observation. Even though the engines switch on a higher altitude when approaching Mars, they must run longer to counter-act the force of gravity (with relatively little help from drag). This longer burn time contributes to more fuel consumed. The mass profile of the vehicle on each of the three descents clearly shows the fact that an optimal descent to Prueba will require less fuel than either Earth or Mars.

On the other hand, we can see from Figure 3.3 that this optimality could come at a cost. At the instant its engines are turned on, the vehicle descending to Mars is traveling downwards at nearly 95 m/s, more than twice the speed if it were to be descending to Mars. The bang-bang nature of this kind of engine control would likely put undesired stress on the vehicle's frame. It is likely, that in this case, the optimal fuel descent may not be the optimal cost descent if the vehicle becomes damaged along the way.

Finally, the overall descent time, most easily seen in Figure 3.3, is almost twice as long under the conditions found on Mars compared to Earth. The longer the vehicle is in motion, the more susceptible it is to other potential forces such as wind as well as other difficult to control

conditions near the desired landing sight. We will explore the former in Chapter 4 and the latter in Chapter 5.

3.4.2   Case 2: Atmospheric Density Varies with Altitude

As we have noted, there are a number of factors affecting atmospheric density as noted by (NASA, 2021) including, among others, altitude, temperature and pressure. Indeed, for earth's atmosphere, the density does not vary uniformly with any variable because of the effects of the others. Furthermore, the atmosphere itself behaves differently as it moves from the lowest elevations, known as the troposphere to higher elevations of the lower and then upper stratospheres.

For the purpose of this paper, we will concentrate on the effect of elevation and use the modelling approximation from (Anglim, 2016) which in the variables from this paper are

$$\rho(h) = \rho_0 e^{-\frac{(h(t))}{h_0}}$$

$$(3.27)$$

and

$$\frac{d\rho}{dh} = -\frac{\rho_0}{h_0} e^{-\frac{h}{h_0}}$$

$$(3.28)$$

Substituting (3.27) and (3.28) into equations (3.8), (3.9), and (3.10) produces

$$H_h = \frac{-k\delta v^2}{m} \frac{\rho_0}{h_0} e^{-\frac{h}{h_0}}$$

$$H_v = \lambda_1 + \frac{2}{m} \delta v \lambda_2 \rho_0 e^{-\frac{h}{h_0}}$$

$$H_m = -\frac{1}{m^2}(u + \delta v^2 \rho_0 e^{-\frac{h}{h_0}})\lambda_2$$

$$(3.29)$$

and

$$\dot{\lambda}_1 = -H_h = \frac{kdv^2}{m}\frac{\rho_0}{h_0}e^{-\frac{h}{h_0}}$$

$$\dot{\lambda}_2 = -H_v = -\lambda_1 - \frac{2}{m}\delta v\lambda_2\rho_0 e^{-\frac{h}{h_0}}$$

$$\dot{\lambda}_3 = -H_m = \frac{1}{m^2}(u + \delta v^2\rho_0 e^{-\frac{h}{h_0}})\lambda_2$$

(3.30)

and

$$H_u = -k\lambda_3 + \frac{1}{m}\lambda_2 - 1$$

(3.31)

It is clear from equations (3.29) and (3.30) that solving for the velocity and movement of the vehicle will be more difficult than in Case 1. However, we observe that the Hamiltonian's derivative over u shown in (3.31) remains the same as in Case 1 and indeed the same as in the Lunar Landing scenario explored in Chapter 2. This allows us to examine the optimal descent trajectory starting from the same assumption of a single switching time at t^*.

So, restarting from the basic equations of motion from (3.4), we have

$$f(\mathbf{x}, u) = \begin{bmatrix} \dot{h} \\ \dot{v} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m}(u + \rho\delta v^2) - g \\ -uk \end{bmatrix}$$

(3.32)

now, the equation for the change in mass is unaffected by the formula for ρ so,

$$m(t) = \begin{cases} m_0, 0 \le t \le t^* \\ m(t), t^* \le t \le t_f \end{cases}$$

However, the equation for the velocity of the descending vehicle is affected by inserting (3.27) into (3.32) yielding

59

$$\dot{v} = \frac{1}{m}\left(u + \rho_0 e^{-\frac{h}{h_0}}\delta v^2\right) - g$$

<div align="right">(3.33)</div>

as noted in with the basic premise that

$$m(t) = \begin{cases} m_0, 0 \le t \le t^* \\ m(t), t^* \le t \le t_f \end{cases}$$

Unfortunately, solving the equations of motion analytically under the switching control is no long practical even for the period of $0 \le t \le t^*$ when the vehicle's mass is constant. So, as we did in the second part of the first case, we will use numerical techniques to solve for $v(t)$ and $h(t)$.

Using the same basic approach in MATLAB as described in (2.18) with the noticeable adjustment of the revised system of equations in (3.32), we explored the impact on all three planets. However, the most interesting result occurs when comparing the descent profiles on the same planet for the different representations of the atmospheric density. The velocity profile and height profile results are shown for Mars in Figure 3.4 where the red line reflects the descent when the atmospheric density is constant, that is, $\rho = \rho_0$, and the magenta line reflects a variable atmospheric density as modeled in (3.30) where

$$\rho(h) = \rho_0 e^{-\frac{(h(t))}{h_0}}.$$
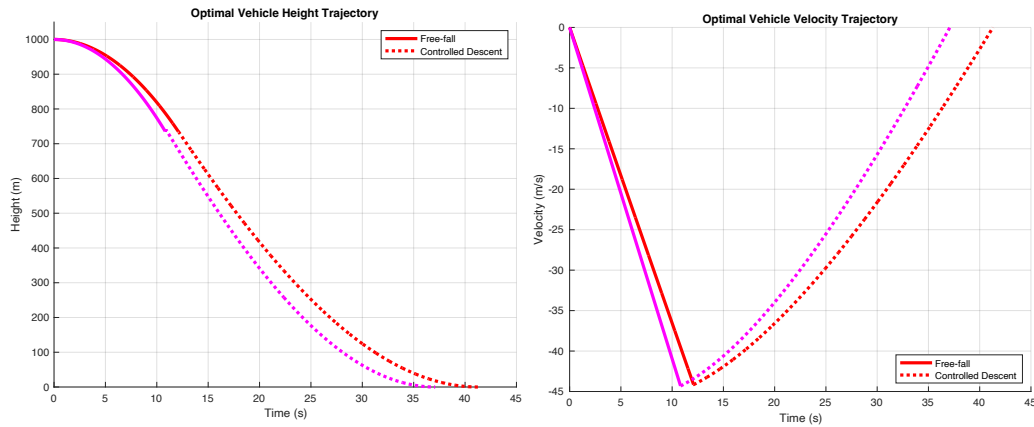
Figure 3.4: The Effect of Variable Atmospheric Density on Optimal Descent Trajectories

Although the impact is subtle, it is clear that the lower atmospheric density at higher altitudes enables the free-falling vehicle to acquire more speed which, as we have mentioned, increases the drag force. This slightly higher velocity is enough to provide even more aid to the engines, especially at lower altitudes, when the logarithmic modeling of the variable atmospheric density quickly approaches the fixed value of ρ_0. Overall, this results in a slightly shorter burn and less fuel consumed.

3.5   Conclusions

In this chapter we explored the more advanced and useful scenario of optimal descent trajectories through the introduction of an atmosphere surrounding the planet on which we would like to softly land the vehicle. In doing so, we have been able to draw some insightful conclusions:

1. The introduction of the drag force caused by the presence of an atmosphere does not change the nature of the control variable that we can use to model and determine the optimal descent. The bang-bang control of having the engines initially turned off and

then turned on at a specific time continues to deliver an optimal descent in terms of fuel consumed

2. On the other hand, the presence of the drag force, and in particular its direct dependence on the square of the velocity of the vehicle introduces non-linearity into the system of equations used to model the scenario. As a result, we begin to see Ricatti equations appear in the solutions. Although some of these are possible to solve, the additional complexity of a variable atmospheric density along with a desire to rapidly simulate the system as a means of determining the optimal descent, lead us to the need to develop numerical solutions for the system

3. Through the results of the numerical solutions, we are clearly able to see the impact of the atmosphere on the optimal descent. We observed that the drag force is the first line of defense against gravity when attempting to slow a free-falling vehicle to ensure a soft landing. Furthermore, the combined impact of gravity and drag actually create the ideal conditions for optimal fuel consumption on descent.

Our objective in this scenario was to understand the impact of an atmosphere on optimal fuel descent trajectories. The modeling of the atmosphere – both as a fixed value and as a logarithmic changing value – provided helpful insight into understanding how optimal descent trajectories will vary from planet to planet. Of course, there a number of other factors that can affect atmospheric density including the composition of the atmosphere, its thickness relative to the planet, and temperature to name just a few. These additional conditions provide ample opportunity for further study of this problem.

CHAPTER IV

SCENARIO THREE: ATMOSPHERIC WIND

4.1 Framing the Scenario

In this scenario, we will build on the work in CHAPTER III and add another force to be reckoned with, wind. This phenomenon is to be expected in any atmosphere. Wind speed, direction and the force it imparts on objects are a result of a number of factors, such as relative differences in atmospheric temperatures and densities. Experience tells us that windspeed and direction appear almost random with gusts of wind affecting objects over very short distances. The direction and firmness of a flag flying from the top of a building on one side of a street might be completely different than that of a flag from the top of a similarly sized building on the other side of the street.

For the purposes of this paper, we model wind as an entirely horizontal force acting on the descending vehicle. As such, we need to consider three important elements of the descent trajectory. The first is the concept of airspeed. As explained by NASA's (Glenn Research Center, 2021), in aviation, airspeed refers to the speed of an airborne craft relative to the air around it. This is different from the craft's groundspeed which the speed of the craft relative to a stationary point on the ground beneath it. Both of these are different from windspeed, which reflects the speed of the air relative to a stationary point on the ground.
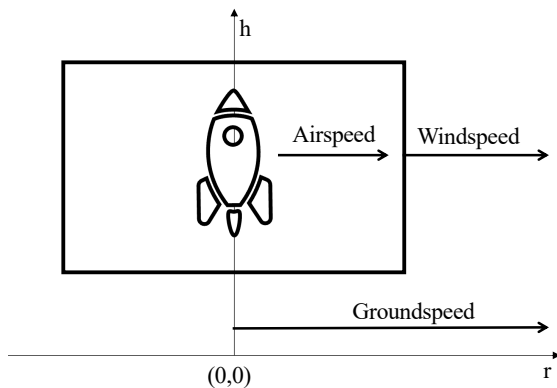
Mathematically, the three speeds are related through a simple formula:

Airspeed = Groundspeed − Windspeed

$$(4.1)$$

Figure 4.1: Aeronautical Speed Terminology

It is important to note that use of the word speed in all of this is aeronautical convention. Indeed, all three are velocities and direction matters. As we shall see, this relationship between the three impacts how we state the equations of motion on the vehicle's descent.

The second key element is that windspeeds generally vary with altitude with winds at higher altitudes being typically higher than at lower points in an atmosphere. There are to primary reasons for this. First, as we have discussed in CHAPTER III, atmospheric density declines with altitude so the forces responsible for creating wind are being imparted on less mass hence higher acceleration and velocities. Second, there is more resistance to air movement at lower altitudes such as trees, buildings, etc. This change in horizontal windspeed due to altitude is referred to as vertical windshear and is an important factor in both meteorology and aeronautics.

The third factor we need to consider is that any attempt to influence the airspeed of an airborne craft will create drag on the vehicle in the opposite direction of the airspeed. This is easy to grasp when we think about trying to walk or cycle into the wind. However, it is still true

if we are moving in the same direction of the wind. If we are walking faster than the wind, the drag will tend to slow us down. If we walk slower than the wind, the drag will cause us to speed up. There is no drag if we are moving at exactly the same speed as the wind because in that case, as noted in the formula above, our airspeed is actually zero.

We are going to introduce one more factor to increase the realism of this scenario and that is the flexibility for guiding the vehicle to land at a point on the ground somewhere downrange from the point directly below the initial position.
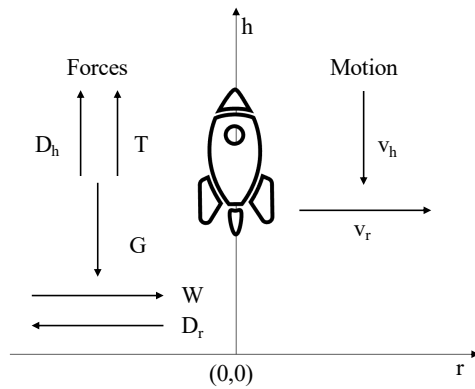
Figure 4.2: Atmospheric Drag and Wind Scenario

In order to create as realistic scenario as possible, we will layer the impact of wind on the second case we explored in CHAPTER III, being the situation where atmospheric density varies with height. The examination of the impact of wind will be done in two cases. The first will be the case where the windspeed varies linearly with height and will be followed by a second case where the windspeed will vary logarithmically with height.

We are going to introduce one more factor to increase the realism of this scenario and that is the flexibility for guiding the vehicle to land at a point on the ground somewhere downrange from the point directly below the initial position. This will provide for the ability to

65

account for, among other things, the rotation of the planet during a descent. An illustration of the system is shown in Figure 4.1.

4.2   Modeling the System

The introduction of a horizontal force makes this a two-dimensional problem. If we restrict our model to consider the two orthogonal paths of motion, vertical and horizontal, as sufficiently independent, we can use the same set of variables from the two previous scenarios to model the vertical dimension and then introduce two new variables for the horizontal motion of the vehicle. We will refer to the horizontal, or downrange, dimension of the problem as range, represented with r. Referring back to the development in (3.1), we can model the system as follows:

$$\dot{h}(t) = v_h$$
$$\dot{r}(t) = v_r$$
$$\dot{v}_h(t) = \frac{1}{m(t)}(T_h(t) + D_h(v_h)) - G(h)$$
$$\dot{v}_r(t) = \frac{1}{m(t)}(T_r(t) + D_r(v_r, w(h))) + W(h)$$
$$\dot{m}(t) = -\frac{T_h + T_r}{G(0)I_{sp}}$$

(4.2)

where:

h(t) is the height of the vehicle above the landing surface

r(t) is the horizontal distance of the vehicle downrange

$v_h(t)$ is the vertical velocity of the vehicle where $v_h(t) \leq 0$

$v_r(t)$ is the horizontal velocity of the vehicle

$G(h)$ is the acceleration due to gravity of the vehicle

$m(t)$ is the mass of the vehicle

$w(h)$ is the wind (atmospheric) velocity surrounding the vehicle

$T_h(t)$ is the thrust (force) of the vertical engines

$T_r(t)$ is the thrust (force) of the horizontal engines

$D_h(v_h)$ is the atmospheric drag force on the vehicle as it descends

$D_r(v_r)$ is the atmospheric drag force as it moves downrange

I_{sp} is specific impulse, or thrust efficiency, of the engines

The initial, terminal, boundary conditions in this scenario are:

$$0 \le t \le t_f$$

$$h(0) = h_0 \text{ and } h(t_f) = 0$$

$$r(0) = 0 \text{ and } r(t_f) = r_f$$

$$v_h(0) = v_{h0} \text{ and } v_h(t_f) = 0$$

$$v_r(0) = w(h_0) \text{ and } v_r(t_f) = 0$$

$$m(0) = m_0 \text{ and } m(t_f) = m_f$$

$$(4.3)$$

Notice that in scenario, the initial downrange velocity of the vehicle is windspeed at the initial height, h_0. We shall show how the different models for $w(h)$ will affect this initial condition.

4.3   Discerning the Appropriate (Optimal) Control

In the Atmospheric Drag scenario discussed in CHAPTER III, we used the same control, the vertical engine thrust, that was first introduced in

CHAPTER II. We found that by switching that control from off to on at the specified time, t^*, during the flight, we were able to minimize the amount of fuel consumed on safe descent and thus the control was optimal. In this new scenario involving wind, it is clear that a second control will be required to govern the thrust produced from a horizontal engine (similar to the rocket car example in 1.4.1).

The presence of horizontal forces requires us to determine appropriate equations of motion in the lateral or downrange dimension. From Newton's law, we have that

$$\dot{v}_r = \frac{-u_r}{m} + \frac{1}{m}\rho\delta_r(v_r - w)^2 + \dot{w}$$

$$(4.4)$$

The first term on the right-hand side represents the acceleration caused by the force of the horizontal engines similar to what we have seen in previous equations for $\dot{v}$ with the except that the force is expected to act in the opposite direction of the wind and as a result shown as negative. The middle term mathematically describes the impact of drag on the vehicle whenever the vehicles groundspeed differs that the windspeed. That is, when the airspeed is not zero. We have introduced the horizontal analogy to the drag coefficient of the vehicle in the form of δ_r. This is very likely a different value from the parameter we introduced in 3.2 (which we now rename as δ_h) because δ_r is based on the vertical profile of the descending vehicle. Finally, the last term represents the time rate of change in windspeed.

There is another important modification to our model in this scenario. The presence of a second horizontal engine acting independently from the vertical engine means there is another consumer of fuel for which we must take account. So, we need to change the way we describe the way in which the mass of the vehicle changes to $\dot{m} = -(k_h u_h + k_r u_r)$ which introduces k_h and k_r as measures of efficiency of the two separate engines.

Based on this, we can begin to examine the requirements for optimal control. Under these conditions we have

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{h} \\ \dot{r} \\ \dot{v}_h \\ \dot{v}_r \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v_h \\ v_r \\ \frac{1}{m}(u_h + \rho\delta_h v_h^2) - g \\ \frac{1}{m}(-u_r + \rho\delta_r(v_r - w)^2) + \dot{w} \\ -(k_h u_h + k_r u_r) \end{bmatrix}$$

(4.5)

and we have two controls, $\mathbf{u} = \begin{bmatrix} u_h \\ u_r \end{bmatrix}$ where u_h is the vertical thrust control and u_r is the horizontal thrust control. The total thrust at any one point in time, u, is the sum of the two so that $u = u_h + u_r$.

We can now proceed with defining the control theory Hamiltonian in this scenario

$$H(x, \lambda, u) = \lambda_1 \dot{h} + \lambda_2 \dot{r} + \lambda_3 \dot{v}_h + \lambda_4 \dot{v}_r + \lambda_5 \dot{m} + u$$

which we can make further explicit through substitution from (4.5) to arrive at

$$H(x, \lambda, u) = \lambda_1 v_h + \lambda_2 v_r + \lambda_3 \left(\frac{1}{m}(u_h + \rho\delta_h v_h^2) - g\right)$$

$$+ \lambda_4 \left(\frac{1}{m}(-u_r + \rho\delta_r(v_r - w)^2) + \dot{w}\right) - \lambda_5(k_h u_h + k_r u_r) - (u_h + u_r)$$

(4.6)

$$H_h = \frac{k\delta_h v_h^2}{m}\left(\frac{d\rho}{dh}\right)\lambda_3 + \left(\delta_r \frac{d(\rho(v_r - w)^2)}{dh} + \frac{d\dot{w}}{dh}\right)\lambda_4$$

$$H_r = 0$$

$$H_{v_h} = \lambda_1 + \frac{2}{m}\rho\delta_h v_h \lambda_3$$

$$H_{v_r} = 2\rho\delta_r(v_r - w)\lambda_4$$

$$H_m = -\frac{1}{m^2}\left(u_h + \rho\delta v_h^2\right)\lambda_3 + \frac{1}{m^2}\left(u_r - \rho\delta_r(v_r - w)^2\right)\lambda_4$$

(4.7)

and

$$\dot{\lambda}_1 = -H_h = -\frac{k\delta_h v_h^2}{m}\left(\frac{d\rho}{dh}\right)\lambda_3 - \left(\delta_r \frac{d(\rho(v_r - w)^2)}{dh} + \frac{d\dot{w}}{dh}\right)\lambda_4$$

$$\dot{\lambda}_2 = -H_r = 0$$

$$\dot{\lambda}_3 = -H_{v_h} = -\lambda_1 - \frac{2}{m}\rho\delta_h v_h \lambda_3$$

$$\dot{\lambda}_4 = -H_{v_r} = -2\rho\delta_r(v_r - w)\lambda_4$$

$$\dot{\lambda}_5 = -H_m = \frac{1}{m^2}\left(u_h + \rho\delta v_h^2\right)\lambda_3 - \frac{1}{m^2}\left(u_r - \rho\delta_r(v_r - w)^2\right)\lambda_4$$

(4.8)

and

$$H_{u_h} = \frac{\lambda_3}{m} - 1 - \lambda_5 k_h$$

$$H_{u_r} = -\frac{\lambda_4}{m} - 1 - \lambda_5 k_r$$

(4.9)

An important observation from equations (4.7) and (4.8) is that only the H_h partial derivative and consequently the $\dot{\lambda}_1$ co-state variable equation are directly dependent on h. We will address the impact on it each of the two cases below where we have specific representations for w(h). The other key result from (4.9) is that the Hamiltonian is linear in **u** in both

dimensions. As we have shown, that means that there is at most one switch for each control for optimality. This combination of two controls with two different values produces four different possibilities for engine thrust as show in Figure 4.3.
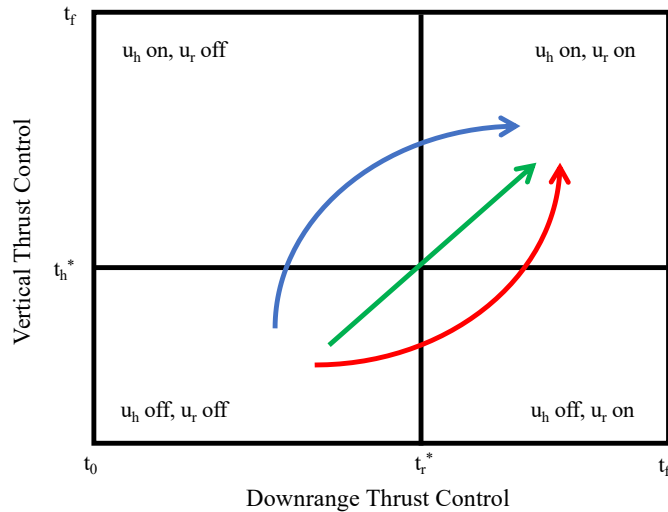
Figure 4.3: Possible Engines Ignition Sequences

This can be simplified by observing that there are only two possible paths for the switching controls from the initial condition where $u_h = 0 = u_r$. Either the vertical engines turn on first at time $t = t_h^*$ and then the horizontal engines at $t = t_r^*$ (blue path) or the two engines turn on in the opposite order (red path). It is possible, although highly unlikely, that both engines could turn on at exactly the same instant ($t_h^* = t_r^*$) (green path). However, regardless of the path taken, both engines can only switch once for optimality; once on, neither can turn off until landing. This gives us the optimal control values for $\mathbf{u}^*$ as

$$u_h^*(t) = \begin{cases} 0 & \text{if } 0 \le t \le t_h^* \\ u & \text{if } t_h^* \le t \le t_f \end{cases}$$

$$u_r^*(t) = \begin{cases} 0 & \text{if } 0 \le t \le t_r^* \\ u & \text{if } t_r^* \le t \le t_f \end{cases}$$

$$(4.10)$$

It is worth confirming that the cost function associated with the consumption of fuel

remains the same as it was when we first introduced it in (1.12) as

$$C(u) = -\int_0^{t_f} \dot{m}(s)\, ds = m(0) - m(t_f)$$

This means that finding the optimal trajectory for descent under these conditions remains a

problem of finding the fastest controlled trajectory resulting in a safe landing. The actual

sequence of the engines igniting, that is whether the vertical engines or the downrange engines

fire first, does not impact the final value of the cost function, although it will have an impact on

the actual trajectory of the vehicle.

We can make one last observation from the equations derived in this scenario. The

introduction of a horizontal wind has not changed the nature of the vertical movement of the

system. In some ways, the problem can be thought of as a combination of the horizontal rocket

car from 1.4.1 and the vertical lunar landing from 1.4.2, albeit with the additional drag and wind

forces we have introduced in CHAPTER III and this chapter, respectively. We are now ready to

determine solutions for the optimal descent trajectories for the two different models of wind.

4.3.1   Case 1 – Linearly Increasing Windspeed

Qualitatively, this case assumes that the windspeed increases in a simple linear fashion

with height, h. As such, we can define a function for the windspeed as

$$w(h) = w_0(1 + hz)$$

$$(4.11)$$

Where w_0 is the windspeed at the surface and z is a constant that represents rate of change in windspeed due to the conditions outlined earlier in the chapter. In this model, z has units of s^{-1}. We can then calculate both the height and time derivatives for the wind function as follows

$$\frac{dw}{dh} = w_0 z$$

$$\dot{w} = \frac{dw}{dh}\frac{dh}{dt} = w_0 z\dot{h} = w_0 z v_h$$

$$(4.12)$$

Recall that $v_h \leq 0$ for the entire descent and so we can see that the wind decreases from its maximum at $t = 0$ when $w(h_0) = w_0 + h_0 z$. We can also determine more explicit expressions for the h partial derivative of the Hamiltonian as well as the time derivate of the first state variable. Starting with

$$H_h = \frac{k\delta_h v_h^2}{m}\left(\frac{d\rho}{dh}\right)\lambda_3 + \left(\delta_r \frac{d(\rho(v_r - w)^2)}{dh} + \frac{d\dot{w}}{dh}\right)\lambda_4$$

$$(4.13)$$

we note from (3.28) that $\frac{d\rho}{dh} = -\frac{\rho_0}{h_0}e^{-\frac{h}{h_0}}$, $\frac{d\dot{w}}{dh} = 0$, and substituting from (4.12) results in

$$\frac{d(\rho(v_r - w)^2)}{dh} = \frac{d\rho}{dh}(v_r - w)^2 + \rho\frac{d(v_r - w)^2}{dh}$$

$$= -\frac{\rho_0}{h_0}e^{-\frac{h}{h_0}}(v_r - w)^2 - 2\rho_0 e^{-\frac{h}{h_0}}(v_r - w)z$$

Thus, (4.13) and $\dot{\lambda}_1$ become

$$H_h = \frac{k\delta_h v_h^2}{m}\left(\frac{d\rho}{dh}\right)\lambda_3 + \left(\delta_r \frac{d(\rho(v_r - w)^2)}{dh} + \frac{d\dot{w}}{dh}\right)\lambda_4$$

$$\dot{\lambda}_1 = -H_h$$

As previously mentioned, the rest of the partial derivatives of the Hamiltonian and time-derivatives of the co-state variables remain unchanged from (4.7), (4.8), and (4.9) by the explicit function for windspeed. Given these results, we are able to procced as we have done before in examining each of the time intervals of descent to understand how the equations of motion in (4.5) evolve during each of the time intervals.

First, let us consider the period of time, $0 \leq t \leq \min(t_h^*, t_r^*)$ when both of the engines are off. In this interval, $u_h = 0 = u_r$ and the downrange velocity remains equal to the windspeed which causes the drag term to vanish. Essentially, the vehicle will be "floating on the wind" during its descent until the downrange engines are ignited. By inserting these values (4.5) becomes

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{h} \\ \dot{r} \\ \dot{v}_h \\ \dot{v}_r \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v_h \\ v_r \\ \frac{1}{m}\left(\rho\delta_h v_h^2\right) - g \\ \dot{w} \\ 0 \end{bmatrix}$$

(4.14)

Next, we can consider the period when $\max(t_h^*, t_r^*) \leq t \leq t_f$ which is when both engines are turned on. During this interval, $f(\mathbf{x},\mathbf{u})$ becomes

74

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{h} \\ \dot{r} \\ \dot{v}_h \\ \dot{v}_r \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v_h \\ v_r \\ \dfrac{1}{m}(u_h + \rho\delta_h v_h^2) - g \\ \dfrac{1}{m}(-u_r + \rho\delta_r(v_r - w)^2) + \dot{w} \\ -(k_h u_h + k_r u_r) \end{bmatrix}$$

(4.15)

Now, we turn our attention to the period when only one of the engines is on. The obvious question we ask ourselves at this point is, which engine comes on first – the horizontal engine controlling for wind drift or the vertical engine controlling for gravity? If the former, then in (4.5), $u_h = 0$, and if the latter, $u_r = 0$.

As we saw in CHAPTER III, attempting to further solve the equations in (4.14) and (4.15) using analytical techniques would become quite challenging. So, we will proceed to numerical solutions leveraging MATLAB as we have done previously.

4.3.2 Case 2 – Exponentially Increasing Windspeed

An even more realistic model for the way in which windspeed increases with height above the surface is to acknowledge that one of the most significant contributing factors to that phenomenon mentioned above, namely the decreasing atmospheric density at higher altitudes. There are two common models that consider this and have been shown to be fairly accurate experimentally: the power or exponential wind law and the logarithmic wind profile law. For the power law (Bañuelos-Ruedas, Angeles-Camacho, & Rios-Marcuello, 2011), we have

$$\frac{w(h)}{w_\varepsilon} = \left(\frac{h}{h_\varepsilon}\right)^z$$

(4.16)

where w_ε is the know speed at some known reference height, h_ε, and z is the Hellman exponent which is a measure of the relative friction of the topography. A common value used for z for

open land is 0.143 (Masters, 2013) which is what we will use in our simulations as it is

reasonable to assume that most vehicle landings will want to be done in as open an area as

possible for safety, security and other reasons.

The second commonly used representation (especially in Europe), is the logarithmic wind

profile law (Perterson & Hennessey, Jr., 1977) which represents the same ratio as

$$\frac{w(h)}{w_\varepsilon} = \frac{\ln\left(\frac{h}{z}\right)}{\ln\left(\frac{h_\varepsilon}{z}\right)}$$

(4.17)

where in this law, z is referred to as the roughness coefficient length and again is a function of

the local topology around h_ε. The flatter and less obstructed terrain, the lower the value.

Although there are many published comparisons between these models, and indeed other

models adopted in different contexts, for our purposes, we will use the exponential power law as

expressed in (4.16). One thing we will need to address is that this law (as with logarithmic law)

relies on a ratio of the two heights. Throughout this paper so far, we have set the surface height

to be 0 and, in this chapter, designated w_0 to be the known windspeed at the surface. Obviously,

we cannot set $h_\varepsilon = 0$ in (4.16) so we will set it to be very small relative to the entire vertical

descent so that we can equate w_ε with w_0. So, from (4.16) we can calculate expressions for w(h)

and $\frac{dw}{dh}$ as follows

$$w(h) = w_0 \left(\frac{h}{h_\varepsilon}\right)^z$$

$$\frac{dw}{dh} = zw_0 \frac{h^{z-1}}{h_\varepsilon{}^z}$$

(4.18)

One observation we can make from the first equation in (4.18) is that if the surface is

perfectly unobstructed, we could let z = 0 in which case w is constantly equal to w_0 throughout

the entire duration of the descent. Secondly, $w(h_0) = w_0 \left(\frac{h_0}{h_\varepsilon}\right)^z$ which means that our choice of

the near-surface reference height, h_ε, could have a significant impact on the initial conditions.

We will address that in the simulations. Lastly, we can recognize that in order to derive an

expression for $\dot{w}$ at any point during the descent, we need to compute the following:

$$\dot{w} = \frac{dw}{dt} = \frac{dw}{dh}\frac{dh}{dt} = zw_0 \frac{h^{z-1}}{h_\varepsilon^z}\dot{h} = zw_0 \frac{h^{z-1}}{h_\varepsilon^z}v_h$$

(4.19)

Now, inserting both equations (4.18) and (4.19) into our general formulas for $\mathcal{H}_h$ and $\dot{\lambda}_1$

in (4.7) and (4.8), respectively, delivers

$$H_h = \frac{k\delta_h v_h^2}{m}\left(\frac{d\rho}{dh}\right)\lambda_3 + \left(\delta_r \frac{d(\rho(v_r - w)^2)}{dh} + \frac{d\dot{w}}{dh}\right)\lambda_4$$

$$\dot{\lambda}_1 = -H_h$$

Again, since all of the required conditions to apply the bang-bang single-switch optimal control

formulas continue you to be met, we can proceed with analyzing the equations of motion in the

time intervals of interest to us.

Starting with the period of time, $0 \leq t \leq \min(t_h^*, t_r^*)$ when both of the engines are off.

In this interval, $u_h = 0 = u_r$ and (4.5) becomes

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{h} \\ \dot{r} \\ \dot{v}_h \\ \dot{v}_r \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v_h \\ v_r \\ \frac{1}{m}(\rho\delta_h v_h^2) - g \\ \dot{w} \\ 0 \end{bmatrix}$$

(4.20)

Similarly, we can consider the period when $\max(t_h^*, t_r^*) \leq t \leq t_f$ which is when both engines are turned on. During this interval, $f(\mathbf{x}, \mathbf{u})$ becomes

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{h} \\ \dot{r} \\ \dot{v}_h \\ \dot{v}_r \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v_h \\ v_r \\ \dfrac{1}{m}\left(u_h + \rho\delta_h v_h^2\right) - g \\ \dfrac{1}{m}\left(-u_r + \rho\delta_r(v_r - w)^2\right) + \dot{w} \\ -(k_h u_h + k_r u_r) \end{bmatrix}$$

(4.21)

As we move to the numerical solutions for (4.20) and (4.21) we will insert the formulas from (4.18) to provide a complete model to be solved in MATLAB.

4.3.3 Numerical Simulations Using Linear Model for Wind

As the conditions under which we have explored optimal descent have advanced through this paper, so have the number of variables and parameters governing the systems used to model them. At this point, there are a great many simulations that we could explore. However, we will confine our exploration to a particular scenario that illustrates the impact of atmospheric wind alongside a desire to safely land the vehicle a position downrange from the initial position at the start of the descent. In order to provide a similar frame of reference to previous scenarios, we will set some of the parameters to the same or similar values. We will also present the results using the same basic format as done previously. We will select Mars as the target planet Table 4-1 captures the values used in MATLAB.

The second dimension requires enhancement to the logic flow initially outlined in (2.18). More specifically, the enhanced logic follows these steps:

1. Establish the values for the various parameters, initial and terminal conditions

2. Calculate the trajectory for initial descent with both control engines turned off and the soft-landing trajectory with the vertical engine turned on using ODE45

3. Determine the point of intersection of the two trajectories using an algorithm from (Schwarz, 2017) and assemble the adjusted descent trajectory resulting from the vertical engine ignition

4. Calculate the trajectory for the soft-landing trajectory with the downrange engine ignited using ODE45

5. Determine the point of intersection of the two trajectories using an algorithm from (Schwarz, 2017) and assemble the combined optimal trajectory and the optimal descent trajectory for the remainder of the descent to a soft landing

6. Plot the results

Table 4-1: MATLAB Simulation Parameters

Vertical Parameters		Downrange Parameters	
Initial Height	1000m	Initial Range	0m
Initial Velocity	0 m/s	Initial Velocity	4 m/s
Maximum Thrust	160N	Maximum Thrust	40N
Engine Exhaust Speed	100 m/s	Engine Exhaust Speed	100 m/s
Gravity	3.7 m/s^2	Surface Windspeed	2 m/s
Drag Coefficient	0.1	Drag Coefficient	1.0
Final Height	0m	Final Range	70m
Final Velocity	0m/s	Final Velocity	0m/s
Other Parameters			
Initial Mass	100kg	Atmospheric Density	.02 kg/m^3
Linear Windspeed	.001m^{-1}	Hellman Exponent	0.143

Some quick analysis on these parameters provides additional insights into the scenario being simulated. More specifically, the desired landing position relative to the initial height and wind profile (both the surface windspeed and the linear model for wind), yield an optimal firing sequence of vertical engine first and the downrange engine second (the blue path depicted in Figure 4.3). Furthermore, the relative engine strength and efficiency combine with the other parameters to show that downrange thrusters ignite to slow the downrange velocity of the vehicle to provide for a soft landing at the desired position. These are visible in Figure 4.4 below.
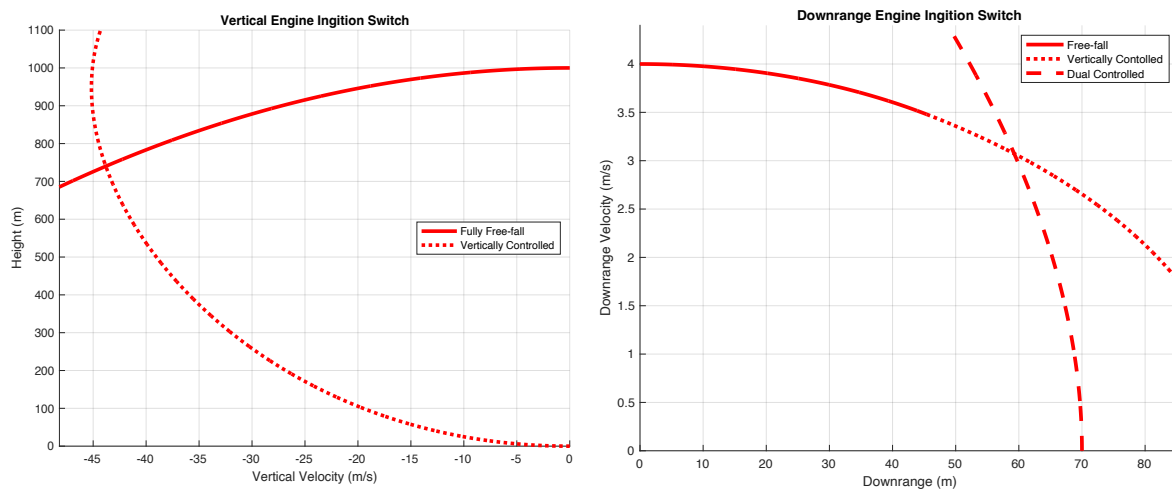
Figure 4.4: Engine Ignition Sequencing for Linear Wind Simulation

Of course, the more interesting observations of this simulation come from understanding how the optimal descent is characterized by consumption of fuel and the trajectory of the vehicle. This is captured in Figure 4.5 where, although subtle, we see the impact of the second engine firing on the mass of the vehicle by the increasing steepness of the line measuring the mass of the vehicle throughout the dual burn. Furthermore, we can see the distinctive change in the descent trajectory once both engines are ignited as the second engine works to bring the vehicle to a soft

landing at the desired downrange position while the vertical engine continues to softly land the vehicle on the surface.
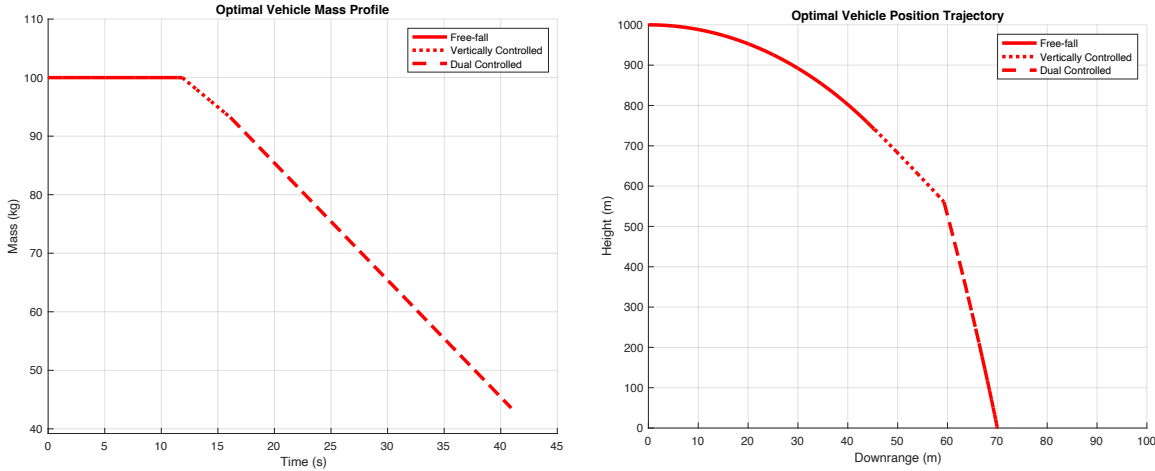
Figure 4.5: Optimal Fuel Consumption and Trajectory for Linear Wind Simulation

4.3.4 Numerical Simulations Using Power Law Model of Wind

In the course of refining the MATLAB programming to simulate this scenario using the power law model for wind, two initial insights were gained. Firstly, because of the compounding effect of exponential growth in windspeed with this model, the initial windspeed and indeed for most of the early part of the descent is much faster than in the case of linear wind speed growth. As a result, the downrange engine needs to be much more powerful in order to slow the vehicle's downrange velocity fast enough for a safe landing. For example, the simulation results illustrated in Figure 4.6 and Figure 4.7, represent a model in which the maximum downrange thrust is 160N, making it equally powerful as the vertical engine.

The second major insight is that the relative timing of the engine ignitions and general behavior of the system under the optimal control of the thrusters is very similar to that of the linear wind model depicted above. The shape of the curves in Figure 4.6 and Figure 4.7 are

81

largely identical to those in Figure 4.4 and Figure 4.5, respectively. The only material differences are derived from the increase in the power of the downrange thrust engine which consumes more fuel and the selection of a different downrange landing point of 110m (as seen in Figure 4.7).
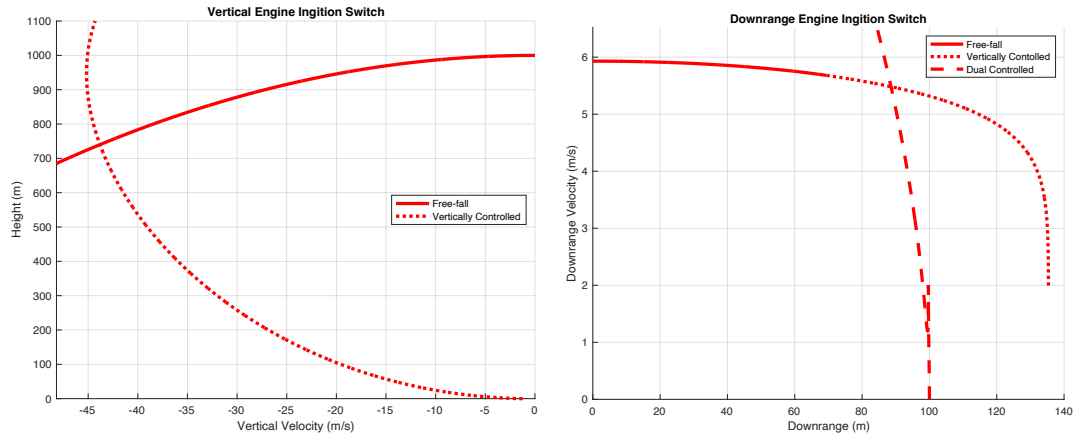
Figure 4.6 Engine Ignition Sequencing for Power Wind Simulation
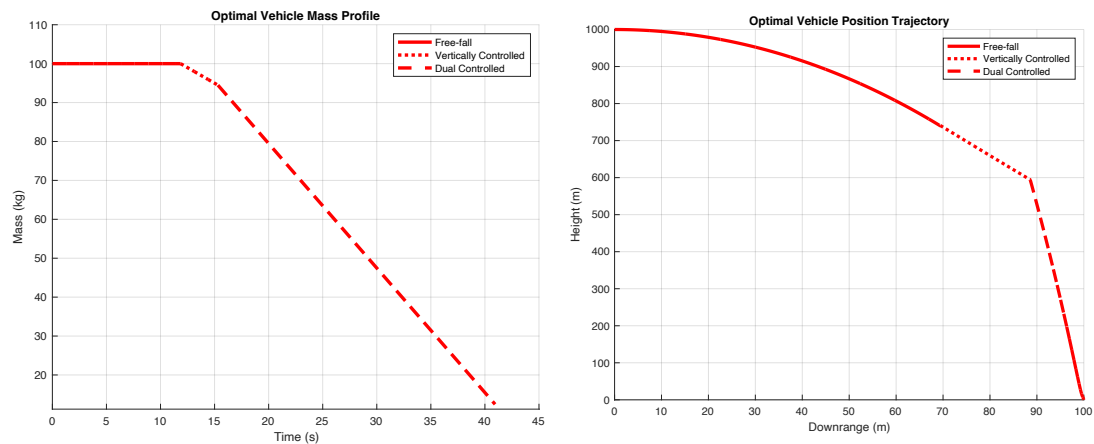
Figure 4.7 Optimal Fuel Consumption and Trajectory for Linear Wind Simulation

4.4   Conclusions

The objective of this scenario was to explore the impact on our optimal control model for descent trajectories when a second, downrange, dimension is added. This was done though the

introduction of simple model for wind which directly affected the downrange velocity of the vehicle along with the desire to select a different downrange landing position for the vehicle. The main results of the analytical and numerical exploration of this scenario as follows:

- The system can be modeled in such a way as to meet the necessary and sufficiency conditions for the Pontrygain Maximum Principle using two simplified bang-bang controls

- The addition of a second engine control gives rise to four possible control states: both engines off, one on and the other off, vice versa, and both on; furthermore, additional analysis is required to understand which engine ignites first to achieve optimal descent and under what conditions

- The manner in which wind is modeled does not materially change the behavior of the controls for optimal descent, provided that the wind is modeled in a manner that the wind is unidirectional; however, some of the parameters governing the system (e.g., engine thrust) may require adjustment in order for the model to numerically simulate optimal control

- This technique, although capable of producing optimal results, has a narrow range of applicability; for example, introducing more flexibility into the modeling of downrange acceleration would permit the accommodation of a wider range of initial and/or terminal conditions.

This last point lays the foundation for a discussion of how to extend the work and findings of this paper which will be outlined in the next and final chapter.

CHAPTER V

RESULTS AND FUTURE EXPLORATION

5.1   Summary of Results

This paper set out to explore the use of optimal control theory as a technique for determining the optimal descent trajectories that minimize the consumption of fuel for vehicles under the advanced conditions of varying atmospheric density, wind and landing positions. We began by laying a foundation of elementary optimal control theory including the properties and conditions under which solutions can be discerned and the use of the Pontrygain Maximum Principal to confirm the optimality of those solutions.

We then applied these principles to the problem of optimal descent trajectories, initially through the use of analytical techniques to derive direct optimal solutions to the applicable equations of motion under the relatively simple conditions of the classic lunar landing problem. We showed that the conditions require to apply the maximum principle were met through both the behavior of the Hamiltonian's partial derivatives and the convexity of the possible set of control values. In order to better visualize the analytical solutions, MATLAB was used to create a variety of simulations. These simulations helped us better understand the effects of a few of the variables on the optimal trajectories.

The next area of exploration was the introduction of an atmosphere to create a more realistic condition for future landings. The presence of an atmosphere changed the basic

equations of motion for the system as a drag force counteracts the force of gravity. This drag force was modeled in two ways: the first was assuming a fixed atmospheric density at all altitudes and the second varies the density with attitude. Under these advanced conditions we were again able to show the maximum principle applied. Once again, the equations were partially solved analytically. Although the Ricatti equations that emerged could have been solved further, it became more desirable to turn to MATLAB-generated numeric solutions. Those solutions showed that the presence of an atmosphere assisted in the conservation of fuel as the drag force worked in conjunction with the engines to slow the vehicle. Furthermore, the combination of the force of gravity and the density of the atmosphere showed to be a critical factor in the optimal descent trajectories.

The final advanced condition explored, atmospheric wind, required the introduction of a second dimension for the model and for the control equations. As with the investigation into the effects of atmosphere, we use two cases to attempt to better understand the impact of wind on optimal descent trajectories. Despite the introduction of a second dimension, the problem was framed in a manner that satisfied the conditions required by the maximum principal. Under the conditions, the models were directly solved numerically using MATLAB. The nuance in calculating these solutions was the recognition that each of the two separate control engines could switch once each but potentially in a different sequence depending on the relative strength of gravity and windspeed. With the introduction of a second dimension of downrange travel, we were also able to investigate the impact of a potentially moving landing surface on the optimal descent trajectories as a further advanced condition corresponding to the more realistic situation of a rotating planet surface.

5.2   Areas for Future Exploration

5.2.1   Deeper Sensitivity Analytics

In

CHAPTER II we explored the sensitivity of our optimal descent trajectory calculations to changes in a few of the initial conditions such as initial height, velocity, engine thrust limits, engine efficiency. This was accomplished by holding two these variables constant while varying the other two. Although this provided some helpful insight into the behavior of the system and how these variables impacted the quest for minimizing the amount of fuel consumed on descent, there is a room for much more analysis to be conducted.

One line of further analysis would be to increase the number of variables considered for sensitivity. For example, the mass of the vehicle could be one of the variables that could be varied, especially with an additional constraint on what portion of the vehicle's mass is fuel at the outset of the descent. Furthermore, we could look further into the relationship between gravity and atmospheric density to determine if there is a relationship between those two that could predictably provide insight into the initial height and velocity of the vehicle for optimal descent.

A second area of further analysis would be to attempt to consider all of the potential variables together as a multivariate regression through a large series of simulations. This would provide for an understanding of the complex relationships between all of the variables, some of which are ultimately fixed on a mission (e.g., gravity of planets) and some of which are variable in almost all situations (e.g., height at the start of descent). Following this line would be

extremely instructive in that as the more we understand about the interplay between these variables on earth prior to launching a vehicle for a multi-destination journey, the more fuel optimal the mission would be.

5.2.2 Sophisticated Wind Models

In CHAPTER IV we got a glimpse of how the presence of a predicable wind impacts the optimal descent trajectory of a vehicle. Our model was able to use this predictability to establish a frame of reference that allowed us to continue to leverage the relative simplicity of a single switch bang-bang control. Intuitively, however, we know that wind is anything but predictable. In reality, onboard sensors might be the only way to accurately understand the impact of atmospheric wind during descent in real time. That said, we could endeavor to gain greater insight into what we might expect through more sophisticated models for wind.

Two possible approaches could involve modeling wind as a sinusoidal function or modeling it as a stochastic function. As an example of the former, we could combine the notion of wind increasing with altitude that we saw used in CHAPTER IV to vary the amplitude of the alternating directions caused by a sinusoidal function to better simulate wind gusts. Alternatively, we could inject a degree of randomness into the model through the use of a stochastic function that could also be designed to drive the direction and speed of the atmospheric wind within an envelope of parameters such as gravity which are more predictable. It is clear, however, that any greater sophistication of wind modeling would also require a more sophisticated treatment of optimal control theory.

5.2.3 Variable Controls

As we mentioned in the previous section, our rather simplistic treatment of our optimal control variables as a single-switch bang-bang control throughout this paper reaches its limit of usefulness once we begin contemplate even more advanced conditions. The remedy is the change the model of the control from a pair of orthogonal engines, to a model where one of the controls is the thrust of the engines and the other is the angle at which the exhaust is expelled. Variations on these approaches are utilized by (Liu, 2018), (Lu, 2018), (Prous, 2020), (Anglim, 2016) and others.

By modeling the controls in this manner, a number of complications are introduced:

1. The basic equations of motion in the system become non-linear in that the direction of exhaust needs to be modeled through the use of sine and cosine functions

2. This complexity caries into the Hamiltonian where its partial derivatives all become more difficult to calculate and, most importantly, $\mathcal{H}_u$ is no longer linear which immediately eliminates the possibility that the optimal control is single-switch or bang-bang

3. The desire to have the thrust controls vary continuously in both amplitude and direction also introduces the need to deal with the fact that $\mathcal{U}$, the set of all possible values of the control variables, $\mathbf{u}$, is no longer convex. This creates a challenge in the use of the Pontrygain Maximum Principle and needs to be addressed. The authors cited above use different approaches for doing so.

Combining all three of these advances into a single examination appears to create an opportunity for original research which would benefit the next great area of human endeavor: going where no one has gone before.

REFERENCES

Acikmese, B., & Ploen, S. R. (2013). Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem. *IEEE transactions on control systems technology 21.6.*

Agrachev, A. A. (2001). Introduction to Optimal Control Theory. Trieste.

Anglim, K. S. (2016). *MINIMUM-FUEL OPTIMAL TRAJECTORY FOR REUSABLE FIRST-STAGE ROCKET LANDING USING PARTICLE SWARM OPTIMIZATION.*

Bañuelos-Ruedas, F., Angeles-Camacho, C., & Rios-Marcuello, S. (2011). Methodologies Used in the Extrapolation of Wind Speed Data at Different Heights and Its Impact in the Wind Energy Resource Assessment in a Region. *InTech Open*.

Boscain, U., & Piccoli, B. (n.d.). An Introduction to Optimal Control.

Brittanica. (2022).

Brodkin, P. L. (2021). An Investigation of Optimal Powered Descent.

Evans, L. C. (1983). An Introduction to Mathematical Optimal Control Theory Version 0.2.

Ferguson, B. S., & Lim, G. (1998). *Introduction to Dynamic Economic Problems.* Manchester: Manchester University Press.

Glenn Research Center. (2021). *Relative Velocities Grounf Reference*. Retrieved from NASA.gov: https://www.grc.nasa.gov/www/k-12/airplane/move.html#:~:text=Airspeed%20is%20the%20vector%20difference,speed%20and%20the%20wind%20speed.&text=On%20a%20perfectly%20still%20day,less%20than%20the%20ground%20speed.

Liu, X. (2018). Fuel-Optimal Rocket Landing with Aerodynamic Controls.

Lu, P. (2018). Propellant-Optimal Powered Descent Guidance. *Journal of Guidance, Control, and Dynamics, Vol. 41, No. 4*, 813–826.

Macki. (2006). *An Introduction to Optimal Control Theory.*

Masters, G. M. (2013). *Renewable and Efficient Electric Power Systems.* Hoboken, NJ: John Wiley & Sons.

NASA. (2021). *https://www.grc.nasa.gov/www/k-12/rocket/dragco.html#:~:text=The%20drag%20coefficient%20Cd%20is,times%20the%20reference%20area%20A.&text=The%20drag%20coefficient%20then%20expresses,dynamic%20pressure%20times%20the%20area.*

Perterson, E. W., & Hennessey, Jr., J. P. (1977). On the Use of Power Laws for Estimates of Wind Power Potential. *Journal of Applied Meteorology*, 390-394.

Pontryagin, L. S., Bolyanskii, W. G., Gamkrelidze, R. V., & Mishchenko, E. (1962). *The Mathematical Theory of Optimal Processes.* Wiley.

Prous, G. Z. (2020). Guidance and Control for Launch and Vertical Descend of Reusable Launchers using Model Predictive Control and Convex Optimisation.

Schwarz, D. M. (2017, Version 2.0). Intersections.m.

Speyer, J. L., & Jacobson, D. H. (2010). *Primer on Optimal Control Theory.* Philadelphia: Society for Industrial and Applied Mathematics.

Swiss Federal Office of Energy. (n.d.). *Wind Profile*. Retrieved from The Swiss Wind Power Data Website: https://wind-data.ch/

Velden, C. S., & Sears, J. (2014). Computing Deep-Tropospheric Vertical Wind Shear Analyses for Tropical Cyclone Applications: Does the Methodology Matter? *American Meteorlogical Society*, 1169-1180.

APPENDIX

MATLAB CODE

Chapter II - Scenario One: Lunar Landing

```
% CHAPTER 2: LUNAR LANDING (NO ATMOPSHERE) USING NUMERIC SOLUTIONS

% INITIAL & TERMINAL CONDITIONS

clear;
h0 = 1000; % Initial height of the vehicle in meters (m)
v0 = 0; % Initial velocity of the vehicle in m/s
m0 = 100; % Initial mass of the vehicle in kg
k = 1/100; % Reciprocal of the speed of expelled exhaust from engines in s/m
g = 1.6; % Acceleration due to gravity in m/s^2
ux = 180; % Control variable upper bound in kgm/s^2
un = 0; % Control variable lower bound in kgm/s^2
a = ux/m0 - g; % Simplifying variable alpha
b = -(k/2)*((ux/m0)^2); % Simplifying variable beta
tm = sqrt(2*h0/a); % Maximum time allotted for initial calculations=
numtimesteps = 9999; % Number of time steps used in calcualtions
timearray = linspace(0,tm,numtimesteps); % Consistent time dimension for all
psl = .95; psu = 1.05; pst = numtimesteps/101; % Limits for plotting


% DESCENT TRAJECTORY CALCULATIONS

% Fully Free-fall Trajectory
u = un;
mdot = @(t,m) -k*u;
vdot = @(t,v) u/(m0-k*u*t) - g;
% Note, hdot = @(t,h) is embedded in x(2)
Fxu = @(t,x) [mdot(t,x(1)); vdot(t,x(2)); x(2)];
[tffall, xffall] = ode45(Fxu, timearray, [m0;v0;h0]);

% Fully Controlled Trajectory
u = ux;
mdot = @(t,m) -k*u;
vdot = @(t,v) (u/(m0-k*u*t) - g);
% Note, hdot = @(t,h) is embedded in x(2)
Fxu = @(t,x) [mdot(t,x(1)); vdot(t,x(2)); x(2)];
[tfctrl, xfctrl] = ode45(Fxu, timearray, [m0,0,0]);

% Calculate the Optimal Switching and Landing Times
[vs,hs] = intersections(-xfctrl(:,2),xfctrl(:,3),xffall(:,2),xffall(:,3));
ts = (vs-v0)/(-g); % Time at which the control thruster is turned on
ffall = [tffall xffall];
fctrl = [tfctrl xfctrl(:,1) -xfctrl(:,2) xfctrl(:,3)];
tsfi = find(ffall(:,3) > vs, 1, 'last' );
tsci = find(fctrl(:,3) < vs, 1 );
tfi = tsfi + tsci;

% Assemble the Optimal Trajectory
optraj = [ffall(1:tsfi,:) ; flip(fctrl(1:tsci,:))];
```

```matlab
optraj(:,1) = transpose(0:1:tfi-1)/pst;
for i = tsfi+1:tfi
    optraj(i,2) = m0 - k*ux*(i-tsfi)/pst;
end

% PLOT RESULTS

% Free-fall and Fully Controlled Trajectories
figure(1); clf(1); hold on; grid on;
ylim([0 h0*psu]);
xlim([vs*psu 0]);
plot(xffall(:,2),xffall(:,3),'b-','linewidth',3);
plot(-xfctrl(:,2),xfctrl(:,3),'b:','linewidth',3);
title('Determining the Optimal Switch Time');
xlabel('Velocity (m/s)');
ylabel('Height (m)');
legend('Fully Free-fall', 'Fully Controlled','Location','east')
figure(1); hold off;

% Mass Profile of the Vehicle
figure(2); clf(2); hold on; grid on;
ylim([optraj(end,2)*psl m0*psu]);
plot(optraj(1:tsfi,1),optraj(1:tsfi,2),'b-','linewidth',3);
plot(optraj(tsfi:end,1),optraj(tsfi:end,2),'b:','linewidth',3);
title('Optimal Vehicle Mass Profile');
legend('Free-fall', 'Controlled','Location','northeast')
ylabel('Mass (kg)');
xlabel('Time (s)');
figure(2); hold off;

% Optimal Velocity Trajectory
figure(3); clf(3); hold on; grid on;
plot(optraj(1:tsfi,1),optraj(1:tsfi,3),'b-','linewidth',3);
plot(optraj(tsfi:end,1),optraj(tsfi:end,3),'b:','linewidth',3);
%plot(tfall,xfall(:,2),'b-','linewidth',3);
%plot(tctrl,xctrl(:,2),'r-','linewidth',3);
title('Optimal Vehicle Velocity Trajectory');
legend('Free-fall', 'Controlled','Location','east')
ylabel('Velocity (m/s)');
xlabel('Time (s)');
figure(3); hold off;

% Optimal Height Trajectory
figure(4); clf(4); hold on; grid on;
plot(optraj(1:tsfi,1),optraj(1:tsfi,4),'b-','linewidth',3);
plot(optraj(tsfi:end,1),optraj(tsfi:end,4),'b:','linewidth',3);
title('Optimal Vehicle Height Trajectory');
legend('Free-fall', 'Controlled','Location','northeast')
ylabel('Height (m)');
xlabel('Time (s)');
figure(4); hold off;
```

Chapter III - Scenario Two: Atmospheric Drag

Scenario Two – Case 1: Fixed Atmospheric Density

```
% CHAPTER 3: ATMOSPHERIC DRAG WITH FIXED ATMOSPHERIC DENSITY
%             USING NUMERIC SOLUTIONS

% INITIAL & TERMINAL CONDITIONS

clear;
h0 = 1000; % Initial height of the vehicle in meters (m)
v0 = 0; % Initial velocity of the vehicle in m/s
m0 = 100; % Initial mass of the vehicle in kg
k = .01; % Reciprocal of the speed of expelled exhaust from engines in s/m
g = 3.7; % Acceleration due to gravity in m/s^2 using Mars as reference
%g = 9.8; % Acceleration due to gravity in m/s^2 using Earth as reference
%g = 15.0; % Acceleration due to gravity in m/s^2 using imaginary planet
ux = 160; % Control variable upper bound in kgm/s^2
un = 0; % Control variable lower bound in kgm/s^2
p0 = .02; % Atmospheric density in kg/m^3 using Mars as reference
%p0 = 1.293; % Atmospheric density in kg/m^3 using Earth as reference
%p0 = .87; % Atmospheric density in kg/m^3 using imaginary planet
d = .1; % Assuming a long cylindrical tapered vehicle shape
tm = 100; % Maximum time allotted for initial calculations
numtimesteps = 9999; % Number of time steps used in calcualtions
timearray = linspace(0,tm,numtimesteps); % Consistent time dimension for all
psl = .95; psu = 1.05; pst = numtimesteps/101; % Limits for plotting

% DESCENT TRAJECTORY CALCULATIONS WITH FIXED AIR PRESSURE

% Fully Free-fall Trajectory
u = un;
mdot = @(t,m) -k*u;
vdot = @(t,v) (1/(m0-k*u*t))*(p0*d*v^2 + u) - g;
% Note, hdot = @(t,h) is embedded in x(2)
Fxu = @(t,x) [mdot(t,x(1)); vdot(t,x(2)); x(2)];
[tffall0, xffall0] = ode45(Fxu, timearray, [m0;v0;h0]);
%[tffall0,xffall0] = FreeFallFixedP(m0,v0,h0,p0,k,un,g,d,tm);

% Fully Controlled Trajectory
u = ux;
mdot = @(t,m) -k*u;
vdot = @(t,v) (1/(m0-k*u*t))*(p0*d*v^2 + u) - g;
% Note, hdot = @(t,h) is embedded in x(2)
Fxu = @(t,x) [mdot(t,x(1)); vdot(t,x(2)); x(2)];
[tfctrl0, xfctrl0] = ode45(Fxu, timearray, [m0;0;0]);
%[tfctrl0,xfctrl0] = CtrlFallFixedP(m0,0,0,p0,k,ux,g,d,tm,0,-1);

% Determine the OPTIMAL Switching TIme
[vs,hs] = intersections(xfctrl0(:,2),-
xfctrl0(:,3),xffall0(:,2),xffall0(:,3));
ffall0 = [tffall0 xffall0];
fctrl0 = [tfctrl0 xfctrl0(:,1) xfctrl0(:,2) -xfctrl0(:,3)];
tsfi = find(ffall0(:,3) > vs, 1, 'last' );
```

```matlab
tsci = find(fctrl0(:,3) < vs, 1 );
tfi = tsfi + tsci;

% Assemble Optimal Trajectory
optraj0 = [ffall0(1:tsfi,:) ; flip(fctrl0(1:tsci,:))];
optraj0 (:,1) = transpose(0:1:tfi-1)/pst;
for i = tsfi+1:tfi
    optraj0(i,2) = m0 - k*ux*(i-tsfi)/pst;
end

% PLOT RESULTS

% Free-fall and Fully Controlled Trajectories
figure(1); hold on; grid on;
ylim([0 h0*psu]);
xlim([vs*psu 0]);
plot(xffall0(1:round(tsfi*1.04),2),xffall0(1:round(tsfi*1.04),3),'g-
','linewidth',3);
plot(xfctrl0(1:round(tsci*1.04),2),-
xfctrl0(1:round(tsci*1.04),3),'g:','linewidth',3);
title('Determining the Optimal Switch Time');
xlabel('Velocity (m/s)');
ylabel('Height (m)');
legend('Fully Free-fall', 'Fully Controlled','Location','east')

% Optimal Mass Profile of the Vehicle
figure(2); hold on; grid on;
ylim([optraj0(end,2)*psl m0*psu]);
plot(optraj0(1:tsfi,1),optraj0(1:tsfi,2),'g-','linewidth',3);
plot(optraj0(tsfi:end,1),optraj0(tsfi:end,2),'g:','linewidth',3);
title('Optimal Vehicle Mass Profile');
legend('Free-fall', 'Controlled Descent', 'Location','northeast');
ylabel('Mass (kg)');
xlabel('Time (s)');
figure(2); hold off;

% Optimal Velocity Trajectory
figure(3); hold on; grid on;
%ylim([optraj0(end,3)*0.95 optraj0(1,3)*1.05]);
plot(optraj0(1:tsfi,1),optraj0(1:tsfi,3),'g-','linewidth',3);
plot(optraj0(tsfi:end,1),optraj0(tsfi:end,3),'g:','linewidth',3);
title('Optimal Vehicle Velocity Trajectory');
legend('Free-fall', 'Controlled Descent', 'Location','southeast');
ylabel('Velocity (m/s)');
xlabel('Time (s)');
figure(3); hold off;

% Optimal Height Trajectory
figure(4); hold on; grid on;
ylim([0 h0*psu]);
plot(optraj0(1:tsfi,1),optraj0(1:tsfi,4),'g-','linewidth',3);
plot(optraj0(tsfi:end,1),optraj0(tsfi:end,4),'g:','linewidth',3);
title('Optimal Vehicle Height Trajectory');
legend('Free-fall', 'Controlled Descent', 'Location','northeast');
ylabel('Height (m)');
```

```
xlabel('Time (s)');
figure(4); hold off;
```


Scenario Two – Case 2: Variable Atmospheric Density

```
% CHAPTER 3: ATMOSPHERIC DRAG WITH VARIABLE ATMOSPHERIC DENSITY
%            USING NUMERIC SOLUTIONS

% INITIAL & TERMINAL CONDITIONS

clear;
h0 = 1000; % Initial height of the vehicle in meters (m)
v0 = 0; % Initial velocity of the vehicle in m/s
m0 = 100; % Initial mass of the vehicle in kg
k = .01; % Reciprocal of the speed of expelled exhaust from engines in s/m
g = 3.721; % Acceleration due to gravity in m/s^2 using Mars as reference
%g = 9.8; % Acceleration due to gravity in m/s^2 using Earth as reference
%g = 15.0; % Acceleration due to gravity in m/s^2 on an imaginary planet
ux = 160; % Control variable upper bound in kgm/s^2
un = 0; % Control variable lower bound in kgm/s^2
p0 = .02; % Atmospheric density in kg/m^3 using Mars as reference
%p0 = 1.293; % Atmospheric density in kg/m^3 using Earth as reference
%p0 = .87; % Atmospheric density in kg/m^3 on an imaginary
d = .1; % Assuming a long cylindrical tapered vehicle shape
tm = 100; % Maximum time allotted for initial calculations
numtimesteps = 999; % Number of time steps used in calcualtions
timearray = linspace(0,tm,numtimesteps); % Consistent time dimension for all
psl = .95; psu = 1.05; pst = 11;%numtimesteps/101; % Limits for plotting

% DESCENT TRAJECTORY CALCULATIONS

% Fully Free-fall Trajectory
u = un;
mdot = @(t,m) -k*u;
vdot = @(t,v,h) (1/(m0-k*u*t))*(p0*(exp(-h/h0))*d*v^2 + u) - g;
% Note, hdot = @(t,h) is embedded in x(2)
Fxu = @(t,x) [mdot(t,x(1)); vdot(t,x(2),x(3)); x(2)];
[tffall0, xffall0] = ode45(Fxu, timearray, [m0;v0;h0]);

% Fully Controlled Trajectory
u = ux;
mdot = @(t,m) -k*u;
vdot = @(t,v,h) (1/(m0-k*u*t))*(p0*(exp(-h/h0))*d*v^2 + u) - g;
% Note, hdot = @(t,h) is embedded in x(2)
Fxu = @(t,x) [mdot(t,x(1)); vdot(t,x(2),x(3)); x(2)];
[tfctrl0, xfctrl0] = ode45(Fxu, timearray, [m0;0;0]);


% Determine the OPTIMAL Switching TIme
[vs,hs] = intersections(xfctrl0(1:30*pst,2),-
xfctrl0(1:30*pst,3),xffall0(:,2),xffall0(:,3));
ffall0 = [tffall0 xffall0];
fctrl0 = [tfctrl0 xfctrl0(:,1) xfctrl0(:,2) -xfctrl0(:,3)];
```

```matlab
tsfi = find(ffall0(:,3) < vs, 1, 'first' );
tsci = find(fctrl0(:,3) < vs, 1);
tfi = tsfi + tsci;

% Assemble Optimal Trajectory
optraj0 = [ffall0(1:tsfi,:) ; flip(fctrl0(1:tsci,:))];
optraj0 (:,1) = transpose(0:1:tfi-1)/pst;
for i = tsfi+1:tfi
    optraj0(i,2) = m0 - k*ux*(i-tsfi)/pst;
end

% PLOT RESULTS

% Free-fall and Fully Controlled Trajectories
figure(1); hold on; grid on;
ylim([0 h0*psu]);
xlim([vs*psu 0]);
plot(xffall0(:,2),xffall0(:,3),'k-','linewidth',3);
plot(xfctrl0(:,2),-xfctrl0(:,3),'g-','linewidth',3);
title('Determining the Optimal Switch Time');
xlabel('Velocity (m/s)');
ylabel('Height (m)');
legend('Fully Free-fall', 'Fully Controlled','Location','east')

% Optimal Mass Profile of the Vehicle
figure(2); hold on; grid on;
ylim([optraj0(end,2)*psl m0*psu]);
plot(optraj0(1:tsfi,1),optraj0(1:tsfi,2),'k-','linewidth',3);
plot(optraj0(tsfi:end,1),optraj0(tsfi:end,2),'g-','linewidth',3);
title('Optimal Vehicle Mass Profile');
legend('Free-fall', 'Controlled Descent', 'Location','northeast');
ylabel('Mass (kg)');
xlabel('Time (s)');
figure(2); hold off;

% Optimal Velocity Trajectory
figure(3); hold on; grid on;
%ylim([optraj0(end,3)*0.95 optraj0(1,3)*1.05]);
plot(optraj0(1:tsfi,1),optraj0(1:tsfi,3),'k-','linewidth',3);
plot(optraj0(tsfi:end,1),optraj0(tsfi:end,3),'g-','linewidth',3);
title('Optimal Vehicle Velocity Trajectory');
legend('Free-fall', 'Controlled Descent', 'Location','southeast');
ylabel('Velocity (m/s)');
xlabel('Time (s)');
figure(3); hold off;

% Optimal Height Trajectory
figure(4); hold on; grid on;
ylim([0 h0*psu]);
plot(optraj0(1:tsfi,1),optraj0(1:tsfi,4),'k-','linewidth',3);
plot(optraj0(tsfi:end,1),optraj0(tsfi:end,4),'g-','linewidth',3);
title('Optimal Vehicle Height Trajectory');
legend('Free-fall', 'Controlled Descent', 'Location','northeast');
ylabel('Height (m)');
xlabel('Time (s)'); figure(4); hold off;
```

CHAPTER IV: Scenario Three – Atmospheric Wind

Scenario Three – Case 1: Linear Growth in Windspeed

```
% CHAPTER 4: ATMOSPHERIC DRAG AND LINEAR WIND USING NUMERIC SOLUTIONS

% INITIAL & TERMINAL CONDITIONS

clear;
h0 = 1000; % Initial height of the vehicle in meters (m)
vh0 = 0; % Initial vertical velocity of the vehicle in m/s
r0 = 0; % Initial downrange position of the vehicle in m
rf = 70; % Final downrange position of the vehicle in m
m0 = 100; % Initial mass of the vehicle in kg
g = 3.7; % Acceleration due to gravity in m/s^2 using Mars as reference
%g = 9.8; % Acceleration due to gravity in m/s^2 using Earth as reference
%g = 15.0; % Acceleration due to gravity in m/s^2 using imaginary planet
kh = .01; % Reciprocal of the speed of expelled exhaust from vertical engines
in s/m
kr = .01; % Reciprocal of the speed of expelled exhaust from horizontal
engines in s/m
uxh = 160; % Vertical control variable upper bound in kgm/s^2
unh = 0; % Vertical control variable lower bound in kgm/s^2
uxr = 40; % Horizontal control variable upper bound in kgm/s^2
unr = 0; % Horizontal control variable lower bound in kgm/s^2
p0 = .02; % Atmospheric density in kg/m^3 using Mars as reference
%p0 = 1.293; % Atmospheric density in kg/m^3 using Earth as reference
%p0 = .87; % Atmospheric density in kg/m^3 using imaginary planet
w0 = 2; % Windspeed in m/s as measured at the surface
dh = 0.1; % Assuming a long cylindrical tapered vehicle shape
dr = 1.0; % Assuming a long cylindrical tapered vehicle shape
tm = 100; % Maximum time allotted for initial calculations
numtimesteps = 9999; % Number of time steps used in calcualtions
timearray = linspace(0,tm,numtimesteps); % Consistent time dimension for all
psl = .90; psu = 1.10; pst = 100; %numtimesteps/101; % Limits for plotting


% WIND FUNCTION CALCULATIONS

z = .001; % Linear growth in windspeed in 1/m
%z = .143; % Hellan exponent value for open surfaces
vr0 = w0*(1 + z*h0); % Initial downrange velocity of the vehicle in meters
(m/s)
h = linspace(0,h0,1000);
w = w0*(1 + z*h);


% TRAJECTORY CALCULATIONS

% Calculate Free-fall Trajectory
Fxu = @(t,x) [x(3);
              x(4);
              (1/x(5))*(unh + p0*(exp(-x(1)/h0))*dh*x(3)^2) - g;
              (1/x(5))*(-unr + p0*(exp(-x(1)/h0))*dr*(x(4) - ...
                  w0*(1+z*x(1)))^2) + w0*z*x(3);
```

```matlab
                -(kh*unh + kr*unr)];
[tffall, xffall] = ode45(Fxu, timearray, [h0;r0;vh0;vr0;m0]);
ffall = [tffall xffall];

% Determine r, vh and vr at free-fall impact (h = 0)
ffi = find(xffall(:,1) < 0, 1);
rff = xffall(ffi,2);
vhff = xffall(ffi,3);
vrff = xffall(ffi,4);

% Calculate Vertical Controlled Trajectory
Fxu = @(t,x) [x(3);
              x(4);
              (1/x(5))*(uxh + p0*(exp(-x(1)/h0))*dh*x(3)^2) - g;
              (1/x(5))*(-unr + p0*(exp(-x(1)/h0))*dr*(x(4) - ...
                  w0*(1+z*x(1)))^2) + w0*z*x(3);
              -(kh*uxh + kr*unr)];
[tvctrl, xvctrl] = ode45(Fxu, timearray, [0;rff;0;vrff;m0]);

% Determine Switching Time of Vertical Control
[vhs,hs] = intersections(xvctrl(1:30*pst,3),-
xvctrl(1:30*pst,1),xffall(:,3),xffall(:,1));
vctrl = [tvctrl -xvctrl(:,1) xvctrl(:,2) xvctrl(:,3) xvctrl(:,4)
xvctrl(:,5)];
tsvi = find(ffall(:,4) < vhs, 1, 'first' );
tcvi = find(vctrl(:,4) < vhs, 1);
tfvi = tsvi + tcvi;

% Assemble First Two Portions of Optimal Trajectory
optraj(1:tfvi,:) = ffall(1:tfvi,:);
optraj(tsvi:tfvi,2) = flip(vctrl(1:tcvi+1,2));
optraj(tsvi:tfvi,4) = flip(vctrl(1:tcvi+1,4));
optraj(:,1) = transpose(0:1:tfvi-1)/pst;
for i = tsvi:tfvi
    optraj(i,6) = m0 - kh*uxh*(i-tsvi)/pst;
end

% Calculate New Controlled Trajectory
Fxu = @(t,x) [x(3);
              x(4);
              (1/x(5))*(uxh + p0*(exp(-x(1)/h0))*dh*x(3)^2) - g;
              (1/x(5))*(-unr + p0*(exp(-x(1)/h0))*dr*(x(4) - ...
                  w0*(1+z*x(1)))^2) + w0*z*x(3);
              -(kh*uxh + kr*unr)];
[tnctrl, xnctrl] = ode45(Fxu, timearray,
[hs;xffall(tsvi,2);vhs;xffall(tsvi,4);m0]);
nctrl = [tnctrl xnctrl];

% Calculate Downrange Controlled Trajectory
Fxu = @(t,x) [x(3);
              x(4);
              (1/x(5))*(unh + p0*(exp(-x(1)/h0))*dh*x(3)^2) - g;
              (1/x(5))*(-uxr + p0*(exp(-x(1)/h0))*dr*(x(4) - ...
                  w0*(1+z*x(1)))^2) + w0*z*x(3);
              -(kh*unh + kr*uxr)];
```

```matlab
[trctrl, xrctrl] = ode45(Fxu, timearray, [0;rf;0;0;m0]);
rctrl = [trctrl xrctrl(:,1) -xrctrl(:,2) xrctrl(:,3) -xrctrl(:,4)
xrctrl(:,5)];

% Determine switching time of downrange control
[vrs,rs] = intersections(-xrctrl(:,4),xrctrl(:,2),xnctrl(:,4),xnctrl(:,2));
if length(vrs) > 1
    vrs = vrs(1);
    rs = rs(1);
end
tsri = find(nctrl(:,5) < vrs, 1, 'first' );
tcri = find(rctrl(:,5) < vrs, 1,'last');
tfri = tsri + tcri;

% Assemble Last Portion of Optimal Trajectory
optraj(tsvi+tsri:tfvi,3) =
flip(interp1(1:tcri,xrctrl(1:tcri,2),linspace(1,tcri,tcvi-tsri+1)));
optraj(tsvi+tsri:tfvi,5) = flip(interp1(1:tcri,-
xrctrl(1:tcri,4),linspace(1,tcri,tcvi-tsri+1)));
for i = tsvi+tsri:tfvi
    optraj(i,6) = optraj(tsvi+tsri,6) - (kh*uxh+kr*uxr)*(i-tsvi-tsri)/pst;
end
mf = optraj(end,6);


% PLOT RESULTS

% Vertical Engine Ignition Switching Time
figure(1); clf(1); hold on; grid on;
xlim([vhs*psu 0]);
ylim([0 h0*psu]);
plot(xffall(1:tfvi,3), xffall(1:tfvi,1), 'r-', LineWidth=3);
plot(xvctrl(1:tfvi,3), -xvctrl(1:tfvi,1), 'r:', LineWidth=3);
title('Vertical Engine Ingition Switch');
xlabel('Vertical Velocity (m/s)');
ylabel('Height (m)');
legend('Free-fall', 'Vertically Controlled','Location','east');
figure(1); hold off;

% Downrange Engine Ignition Switching Time
figure(2); clf(2); hold on; grid on;
ylim([0 vr0*psu]);
xlim([0 rff*psu]);
plot(ffall(1:tsvi,3), ffall(1:tsvi,5), 'r-',LineWidth=3);
plot(xnctrl(1:tfvi,2), xnctrl(1:tfvi,4), 'r:', LineWidth=3);
plot(xrctrl(1:tfri,2), -xrctrl(1:tfri,4), 'r--', LineWidth=3);
title('Downrange Engine Ingition Switch');
ylabel('Downrange Velocity (m/s)');
xlabel('Downrange (m)');
legend('Free-fall','Vertically Contolled', 'Dual
Controlled','Location','northeast');
figure(2); hold off;

% Optimal Mass Profile of the Vehicle
figure(3); clf(3); hold on; grid on;
```

```matlab
xlim([0 tfvi*psu/pst]);
ylim([mf*psl m0*psu]);
plot(optraj(1:tsvi,1),optraj(1:tsvi,6),'r-','linewidth',3);
plot(optraj(tsvi:tsvi+tsri,1),optraj(tsvi:tsvi+tsri,6),'r:','linewidth',3);
plot(optraj(tsvi+tsri:tfvi,1),optraj(tsvi+tsri:tfvi,6),'r--','linewidth',3);
title('Optimal Vehicle Mass Profile');
legend('Free-fall', 'Vertically Controlled', 'Dual Controlled',
'Location','northeast');
ylabel('Mass (kg)');
xlabel('Time (s)');
figure(3); hold off;

% Optimal Vehicle Position Trajectory
figure(4);  hold on; grid on;
xlim([0 100]);
ylim([0 h0]);
plot(optraj(1:tsvi,3), optraj(1:tsvi,2),'r-','linewidth',3);
plot(optraj(tsvi:tsvi+tsri,3), optraj(tsvi:tsvi+tsri,2),'r:','linewidth',3);
plot(optraj(tsvi+tsri:tfvi,3), optraj(tsvi+tsri:tfvi,2),'r--','linewidth',3);
title('Optimal Vehicle Position Trajectory');
legend('Free-fall', 'Vertically Controlled', 'Dual Controlled',
'Location','northeast');
ylabel('Height (m)');
xlabel('Downrange (m)');
figure(4); hold off;
```

Scenario Three – Case 1: Linear Growth in Windspeed

```
% CHAPTER 4: ATMOSPHERIC DRAG AND POWER LAW WIND USING NUMERIC SOLUTIONS

% INITIAL & TERMINAL CONDITIONS

clear;
h0 = 1000; % Initial height of the vehicle in meters (m)
vh0 = 0; % Initial vertical velocity of the vehicle in m/s
r0 = 0; % Initial downrange position of the vehicle in m
rf = 100; % Final downrange position of the vehicle in m
m0 = 100; % Initial mass of the vehicle in kg
g = 3.7; % Acceleration due to gravity in m/s^2 using Mars as reference
%g = 9.8; % Acceleration due to gravity in m/s^2 using Earth as reference
%g = 15.0; % Acceleration due to gravity in m/s^2 using imaginary planet
kh = .01; % Reciprocal of the speed of expelled exhaust from vertical engines
in s/m
kr = .01; % Reciprocal of the speed of expelled exhaust from horizontal
engines in s/m
uxh = 160; % Vertical control variable upper bound in kgm/s^2
unh = 0; % Vertical control variable lower bound in kgm/s^2
uxr = 160; % Horizontal control variable upper bound in kgm/s^2
unr = 0; % Horizontal control variable lower bound in kgm/s^2
p0 = .02; % Atmospheric density in kg/m^3 using Mars as reference
%p0 = 1.293; % Atmospheric density in kg/m^3 using Earth as reference
%p0 = .87; % Atmospheric density in kg/m^3 using imaginary planet
w0 = 2; % Windspeed in m/s as measured at the surface
dh = 0.1; % Assuming a long cylindrical tapered vehicle shape
dr = 1.0; % Assuming a long cylindrical tapered vehicle shape
tm = 100; % Maximum time allotted for initial calculations
numtimesteps = 9999; % Number of time steps used in calcualtions
timearray = linspace(0,tm,numtimesteps); % Consistent time dimension for all
psl = .90; psu = 1.10; pst = 100; %numtimesteps/101; % Limits for plotting


% WIND FUNCTION CALCULATIONS

%z = .001; % Linear growth in windspeed in 1/m
he = .5; %Reference height for windspeed at the surface
z = .143; % Hellan exponent value for open surfaces
vr0 = w0*(h0/he)^z; % Initial downrange velocity of the vehicle in meters
(m/s)
h = linspace(he,h0,1000);
w = w0*(h/he).^z;


% TRAJECTORY CALCULATIONS

% Calculate Free-fall Trajectory
Fxu = @(t,x) [x(3);
              x(4);
              (1/x(5))*(unh + p0*(exp(-x(1)/h0))*dh*x(3)^2) - g;
              (1/x(5))*(-unr + p0*(exp(-x(1)/h0))*dr*(x(4) - ...
                  w0*(x(1)/he)^z)^2) + w0*z*(x(1)^(z-1)/he^z)*x(3);
```

101

```matlab
                    -(kh*unh + kr*unr)];
[tffall, xffall] = ode45(Fxu, timearray, [h0;r0;vh0;vr0;m0]);
ffall = [tffall xffall];

% Determine r, vh and vr at free-fall impact (h = 0)
ffi = find(xffall(:,1) < 1, 1);
rff = xffall(ffi,2);
vhff = xffall(ffi,3);
vrff = xffall(ffi,4);

% Calculate Vertical Controlled Trajectory
Fxu = @(t,x) [x(3);
                x(4);
                (1/x(5))*(uxh + p0*(exp(-x(1)/h0))*dh*x(3)^2) - g;
                (1/x(5))*(-unr + p0*(exp(-x(1)/h0))*dr*(x(4) - ...
                w0*(x(1)/he)^z)^2) + w0*z*(x(1)^(z-1)/he^z)*x(3);
                -(kh*uxh + kr*unr)];
[tvctrl, xvctrl] = ode45(Fxu, timearray, [he;rff;0;vrff;m0]);

% Determine Switching Time of Vertical Control
[vhs,hs] = intersections(xvctrl(:,3),-xvctrl(:,1),xffall(:,3),xffall(:,1));
vctrl = [tvctrl -xvctrl(:,1) xvctrl(:,2) xvctrl(:,3) xvctrl(:,4)
xvctrl(:,5)];
tsvi = find(ffall(:,4) < vhs, 1, 'first' );
tcvi = find(vctrl(:,4) < vhs, 1);
tfvi = tsvi + tcvi;

% Assemble First Two Portions of Optimal Trajectory
optraj(1:ffi,:) = ffall(1:ffi,:);
optraj(tsvi:tfvi,2) = flip(vctrl(1:tcvi+1,2));
optraj(tsvi:tfvi,4) = flip(vctrl(1:tcvi+1,4));
optraj(:,1) = transpose(0:1:tfvi-1)/pst;
for i = tsvi:tfvi
    optraj(i,6) = m0 - kh*uxh*(i-tsvi)/pst;
end

% Calculate New Controlled Trajectory
Fxu = @(t,x) [x(3);
                x(4);
                (1/x(5))*(uxh + p0*(exp(-x(1)/h0))*dh*x(3)^2) - g;
                (1/x(5))*(-unr + p0*(exp(-x(1)/h0))*dr*(x(4) - ...
                    w0*(x(1)/he)^z)^2) + w0*z*(x(1)^(z-1)/he^z)*x(3);
                -(kh*uxh + kr*unr)];
[tnctrl, xnctrl] = ode45(Fxu, timearray,
[hs;xffall(tsvi,2);vhs;xffall(tsvi,4);m0]);
nctrl = [tnctrl xnctrl];

% Calculate Downrange Controlled Trajectory
Fxu = @(t,x) [x(3);
                x(4);
                (1/x(5))*(unh + p0*(exp(-x(1)/h0))*dh*x(3)^2) - g;
                (1/x(5))*(-uxr + p0*(exp(-x(1)/h0))*dr*(x(4) - ...
                    w0*(x(1)/he)^z)^2) + w0*z*(x(1)^(z-1)/he^z)*x(3);
                -(kh*unh + kr*uxr)];
[trctrl, xrctrl] = ode45(Fxu, timearray, [he;rf;0;0;m0]);
```

102

```matlab
xrctrl = real(xrctrl);
rctrl = [trctrl xrctrl(:,1) -xrctrl(:,2) xrctrl(:,3) -xrctrl(:,4)
xrctrl(:,5)];

% Determine switching time of downrange control
[vrs,rs] = intersections(-xrctrl(:,4),xrctrl(:,2),xnctrl(:,4),xnctrl(:,2));
if length(vrs) > 1
    vrs = vrs(1);
    rs = rs(1);
end
tsri = find(nctrl(:,5) < vrs, 1, 'first' );
tcri = find(rctrl(:,5) < vrs, 1, 'last');
tfri = tsri + tcri;

% Assemble Last Portion of Optimal Trajectory
optraj(tsvi+tsri:tfvi,3) =
flip(interp1(1:tcri,xrctrl(1:tcri,2),linspace(1,tcri,tcvi-tsri+1)));
optraj(tsvi+tsri:tfvi,5) = flip(interp1(1:tcri,-
xrctrl(1:tcri,4),linspace(1,tcri,tcvi-tsri+1)));
for i = tsvi+tsri:tfvi
    optraj(i,6) = optraj(tsvi+tsri,6) - (kh*uxh+kr*uxr)*(i-tsvi-tsri)/pst;
end
mf = optraj(end,6);


% PLOT RESULTS

% Vertical Engine Ignition Switching Time
figure(1); clf(1); hold on; grid on;
xlim([vhs*psu 0]);
ylim([0 h0*psu]);
plot(xffall(1:ffi,3), xffall(1:ffi,1), 'r-', LineWidth=3);
plot(xvctrl(1:tfvi,3), -xvctrl(1:tfvi,1), 'r:', LineWidth=3);
title('Vertical Engine Ingition Switch');
xlabel('Vertical Velocity (m/s)');
ylabel('Height (m)');
legend('Free-fall', 'Vertically Controlled','Location','east');

% Downrange Engine Ignition Switching Time
figure(2); clf(2); hold on; grid on;
ylim([0 vr0*psu]);
xlim([0 rff*psu]);
plot(ffall(1:tsvi,3), ffall(1:tsvi,5), 'r-', LineWidth=3);
plot(xnctrl(1:end,2), xnctrl(1:end,4), 'r:', LineWidth=3);
plot(xrctrl(1:end,2), -xrctrl(1:end,4), 'r--', LineWidth=3);
title('Downrange Engine Ingition Switch');
ylabel('Downrange Velocity (m/s)');
xlabel('Downrange (m)');
legend('Free-fall','Vertically Contolled', 'Dual
Controlled','Location','northeast');

% Optimal Mass Profile of the Vehicle
figure(3); clf(3); hold on; grid on;
xlim([0 tfvi*psu/pst]);
ylim([mf*psl m0*psu]);
```

```matlab
plot(optraj(1:tsvi,1),optraj(1:tsvi,6),'r-','linewidth',3);
plot(optraj(tsvi:tsvi+tsri,1),optraj(tsvi:tsvi+tsri,6),'r:','linewidth',3);
plot(optraj(tsvi+tsri:tfvi,1),optraj(tsvi+tsri:tfvi,6),'r--','linewidth',3);
title('Optimal Vehicle Mass Profile');
legend('Free-fall', 'Vertically Controlled', 'Dual Controlled',
'Location','northeast');
ylabel('Mass (kg)');
xlabel('Time (s)');
figure(3); hold off;

% Optimal Vehicle Position Trajectory
figure(4); clf(4); hold on; grid on;
xlim([0 100]);
ylim([0 h0]);
plot(optraj(1:tsvi,3), optraj(1:tsvi,2),'r-','linewidth',3);
plot(optraj(tsvi:tsvi+tsri,3), optraj(tsvi:tsvi+tsri,2),'r:','linewidth',3);
plot(optraj(tsvi+tsri:tfvi,3), optraj(tsvi+tsri:tfvi,2),'r--','linewidth',3);
title('Optimal Vehicle Position Trajectory');
legend('Free-fall', 'Vertically Controlled', 'Dual Controlled',
'Location','northeast');
ylabel('Height (m)');
xlabel('Downrange (m)');
figure(4); hold off;
```

BIOGRAPHICAL SKETCH

John M. Levis completed his Bachelor of Mathematics degree majoring in Applied Mathematics & Engineering Electives from the University of Waterloo from 1983-1988. After graduation, John embarked on a career in Management Consulting with Deloitte that is well into its fourth decade. During that time, he has specialized in developing growth and innovation strategies for a variety of companies in the telecommunications and technology industries across the country and around the world. Having been admitted to the partnership in 1997, John has held several leadership positions within the firm overseeing areas such as strategy, innovation, and Americas region operations.

In January of 2021, while he continued his career at Deloitte, John returned to school at the University of Texas – Rio Grande Valley where he earned a Master of Science in the School of Mathematical and Statistical Studies in May 2023. In his spare time, John is a member of the Board of the American Heart Association where he is the past-chair of the Leadership Succession Committee and a member of the Gala Executive Leadership Team. John resides in the Dallas area, has three college-age sons, and can be reached at jlevis314159@hotmail.com.