THE EFFECT OF NAVIGATION STATE SELECTION ON FUEL DISPERSIONS FOR POWERED LUNAR DESCENT

by

Jessica Thornley Anderson

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Aerospace Engineering

Approved:

Dr. David K. Geller
Major Professor

Dr. R. Rees Fullmer
Committee Member

Dr. Stephen A. Whitmore
Committee Member

Dr. Mark R. McLellan
Vice President for Research and
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2013
Abstract

The Effect of Navigation State Selection on Fuel Dispensions for Powered Lunar Descent

by

Jessica Thornley Anderson, Master of Science

Utah State University, 2013

Major Professor: Dr. David K. Geller
Department: Mechanical and Aerospace Engineering

A Monte Carlo simulation is developed to study the performance of the closed-loop guidance, navigation, and control (GN&C) system of a lunar lander in powered descent. The simulation includes six-degrees-of-freedom dynamics, an extended Kalman filter, guidance based upon modified Apollo methods, an attitude control system, and several different types of sensors. Sensors included in the sensor suite include accelerometers, gyroscopes, a star camera, an altimeter, a velocimeter, and a radio navigation system. The simulation is used to examine the effects of sensor errors and the number of navigation states on total fuel use.
Public Abstract

The Effect of Navigation State Selection on Fuel Dispensions for Powered Lunar Descent

by

Jessica Thornley Anderson, Master of Science

Utah State University, 2013

Major Professor: Dr. David K. Geller
Department: Mechanical and Aerospace Engineering

A Monte Carlo simulation is developed to study the performance of the closed-loop guidance, navigation, and control (GN&C) system of a lunar lander in powered descent. The simulation includes six-degrees-of-freedom dynamics, an extended Kalman filter, guidance based upon modified Apollo methods, an attitude control system, and several different types of sensors. Sensors included in the sensor suite include accelerometers, gyroscopes, a star camera, an altimeter, a velocimeter, and a radio navigation system. The simulation is used to examine the effects of sensor errors and the number of navigation states on total fuel use.
Acknowledgments

I would like to express my deep thanks to a number of people without whom I never could have completed this thesis or my studies at Utah State University. First, I’d like to thank Dr. David Geller, my major professor, for his support, expertise, and willingness to go above and beyond to make sure I succeeded in my research. From his initial suggestions of possible thesis topics, to helping me track down the last few bugs in my computer code, he has offered invaluable assistance at every step of the way.

Thanks also need to go to the other wonderful faculty members who have helped me during my time at Utah State University. Also, I’d like to thank the other members of the GN & C group, who have been willing to answer questions and help with problems. I also

need to acknowledge the members of the USU section of the Society of Women Engineers, especially our SWE counselor, Nina Glaftti, our SWE advisor, Dr. Christine Hailey, and all of my “SWE-isters” who helped me immensely.

I want to thank my parents, Tad and Melanie Thornley, and the rest of my wonderful family for their support, encouragement, and love. Also, my thesis wouldn’t be complete without mentioning my son, Graeme, who was the best reason I can imagine for not finishing this thesis when I originally intended. Finally, thanks to my wonderful husband, Phil, for helping me edit my thesis, putting up with a slightly stressed-out wife for several years, and for never complaining if I didn’t make dinner until 10 o’clock (or at all).

Jessica Thornley Anderson
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Public Abstract</td>
<td>iv</td>
</tr>
<tr>
<td>List of Tables</td>
<td>viii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Lunar Descent Guidance</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Apollo Guidance</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Optimal Rocket Problem</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Modified Apollo Guidance</td>
<td>7</td>
</tr>
<tr>
<td>2.3.1 Guidance During Intermediate Orbit</td>
<td>9</td>
</tr>
<tr>
<td>2.3.2 Braking Phase Guidance Equation</td>
<td>10</td>
</tr>
<tr>
<td>2.3.3 Approach Phase Guidance Equation</td>
<td>12</td>
</tr>
<tr>
<td>2.3.4 Guidance During Terminal Descent</td>
<td>13</td>
</tr>
<tr>
<td>2.3.5 Guidance Targets</td>
<td>14</td>
</tr>
<tr>
<td>3 Lunar Descent Navigation</td>
<td>16</td>
</tr>
<tr>
<td>3.1 Apollo Navigation</td>
<td>16</td>
</tr>
<tr>
<td>3.2 Inertial Navigation</td>
<td>16</td>
</tr>
<tr>
<td>3.3 Standard Kalman Filter</td>
<td>18</td>
</tr>
<tr>
<td>3.4 Extended Kalman Filter</td>
<td>22</td>
</tr>
<tr>
<td>3.5 Kalman Filter for Lunar Descent</td>
<td>24</td>
</tr>
<tr>
<td>3.5.1 States Included in the Navigation Filter</td>
<td>24</td>
</tr>
<tr>
<td>3.5.2 Navigation State Dynamics</td>
<td>27</td>
</tr>
<tr>
<td>3.5.3 Navigation State Propagation</td>
<td>28</td>
</tr>
<tr>
<td>3.5.4 Measurement Models</td>
<td>31</td>
</tr>
<tr>
<td>3.5.5 Navigation State Update</td>
<td>35</td>
</tr>
<tr>
<td>4 Simulation Models</td>
<td>36</td>
</tr>
<tr>
<td>4.1 Simulation Inputs</td>
<td>36</td>
</tr>
<tr>
<td>4.2 Simulation Outputs and Metrics</td>
<td>43</td>
</tr>
<tr>
<td>4.3 Dynamics</td>
<td>48</td>
</tr>
<tr>
<td>4.4 Actuators</td>
<td>50</td>
</tr>
<tr>
<td>4.5 Sensors</td>
<td>51</td>
</tr>
<tr>
<td>4.6 Attitude Control</td>
<td>54</td>
</tr>
</tbody>
</table>
5 Results .................................................................................................................. 58
  5.1 Full Navigation State Data ................................................................. 59
    5.1.1 Case 1 - Inertial Only ................................................................. 59
    5.1.2 Case 2 - Inertial and Radio Navigation ........................................ 59
  5.2 Reduced Navigation State Data ............................................................ 75
    5.2.1 Case 3 - Inertial Only ................................................................. 75
    5.2.2 Case 4 - Inertial and Radio Navigation ........................................ 76
  5.3 Fuel Use ............................................................................................. 93
  5.4 Results Summary Tables ................................................................. 96

6 Conclusions ................................................................................................. 100

References ......................................................................................................... 102
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>26</td>
</tr>
<tr>
<td>4.1</td>
<td>38</td>
</tr>
<tr>
<td>4.2</td>
<td>38</td>
</tr>
<tr>
<td>4.3</td>
<td>39</td>
</tr>
<tr>
<td>4.4</td>
<td>39</td>
</tr>
<tr>
<td>4.5</td>
<td>40</td>
</tr>
<tr>
<td>4.6</td>
<td>41</td>
</tr>
<tr>
<td>4.7</td>
<td>42</td>
</tr>
<tr>
<td>4.8</td>
<td>43</td>
</tr>
<tr>
<td>4.9</td>
<td>44</td>
</tr>
<tr>
<td>4.10</td>
<td>52</td>
</tr>
<tr>
<td>4.12</td>
<td>53</td>
</tr>
<tr>
<td>5.1</td>
<td>58</td>
</tr>
<tr>
<td>5.2</td>
<td>97</td>
</tr>
<tr>
<td>5.3</td>
<td>98</td>
</tr>
<tr>
<td>5.4</td>
<td>99</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Lunar-descent-guidance phases and targets [1].</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Optimal rocket trajectory problem.</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>Lunar descent mission scenario.</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>Phases of powered lunar descent.</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>Moon-centered inertial and Moon-fixed reference frames.</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Landing site reference frame.</td>
<td>19</td>
</tr>
<tr>
<td>3.3</td>
<td>Body-fixed and IMU-fixed reference frames.</td>
<td>20</td>
</tr>
<tr>
<td>3.4</td>
<td>Local vertical-local horizontal frame.</td>
<td>21</td>
</tr>
<tr>
<td>3.5</td>
<td>Navigation algorithm overview.</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Powered lunar descent Monte Carlo simulation overview.</td>
<td>37</td>
</tr>
<tr>
<td>5.1</td>
<td>LVLH errors, full navigation state, inertial measurements only, Best scenario.</td>
<td>60</td>
</tr>
<tr>
<td>5.2</td>
<td>LVLH errors, full navigation state, inertial measurements only, Typical scenario.</td>
<td>61</td>
</tr>
<tr>
<td>5.3</td>
<td>LVLH errors, full navigation state, inertial measurements only, Worst scenario.</td>
<td>62</td>
</tr>
<tr>
<td>5.4</td>
<td>Final position and velocity, full navigation state, inertial measurements only, Best scenario.</td>
<td>63</td>
</tr>
<tr>
<td>5.5</td>
<td>Final position and velocity, full navigation state, inertial measurements only, Typical scenario.</td>
<td>64</td>
</tr>
<tr>
<td>5.6</td>
<td>Final position and velocity, full navigation state, inertial measurements only, Worst scenario.</td>
<td>65</td>
</tr>
<tr>
<td>5.7</td>
<td>Crosstrack versus downrange errors, full navigation state, inertial measurements only, Best scenario.</td>
<td>66</td>
</tr>
<tr>
<td>5.8</td>
<td>Crosstrack versus downrange errors, full navigation state, inertial measurements only, Typical scenario.</td>
<td>66</td>
</tr>
</tbody>
</table>
5.9 Crosstrack versus downrange errors, full navigation state, inertial measurements only, Worst scenario. .............................................................. 67
5.10 LVLH errors, full navigation state, inertial + radio navigation measurements, Best scenario. ................................................................. 68
5.11 LVLH errors, full navigation state, inertial + radio navigation measurements, Typical scenario. .............................................................. 69
5.12 LVLH errors, full navigation state, inertial + radio navigation measurements, Worst scenario. ................................................................. 70
5.13 Final position and velocity, full navigation state, inertial + radio navigation measurements, Best scenario. ............................................ 71
5.14 Final position and velocity, full navigation state, inertial + radio navigation measurements, Typical scenario. ........................................... 72
5.15 Final position and velocity, full navigation state, inertial + radio navigation measurements, Worst scenario. ............................................ 73
5.16 Crosstrack versus downrange errors, full navigation state, inertial + radio navigation measurements, Best scenario. ............................................ 74
5.17 Crosstrack versus downrange errors, full navigation state, inertial + radio navigation measurements, Typical scenario. ............................................ 74
5.18 Crosstrack versus downrange errors, full navigation state, inertial + radio navigation measurements, Worst scenario. ............................................ 75
5.19 LVLH errors, reduced navigation state, inertial measurements only, Best Scenario. ................................................................. 77
5.20 LVLH errors, reduced navigation state, inertial measurements only, Typical scenario. ................................................................. 78
5.21 LVLH errors, reduced navigation state, inertial measurements only, Worst scenario. ................................................................. 79
5.22 Final position and velocity, reduced navigation state, inertial measurements only, Best scenario. ................................................................. 80
5.23 Final position and velocity, reduced navigation state, inertial measurements only, Typical scenario. ................................................................. 81
5.24 Final position and velocity, reduced navigation state, inertial measurements only, Worst scenario. ................................................................. 82
5.25 Crosstrack versus downrange errors, reduced navigation state, inertial measurements only, Best scenario. ............................................ 83
5.26 Crosstrack versus downrange errors, reduced navigation state, inertial measurements only, Typical scenario. ............................................ 83
5.27 Crosstrack versus downrange errors, reduced navigation state, inertial measurements only, Worst scenario. ............................................ 84
5.28 LVLH errors, reduced navigation state, inertial + radio navigation measurements, Best scenario. ............................................ 85
5.29 LVLH errors, reduced navigation state, inertial + radio navigation measurements, Typical scenario. ............................................ 86
5.30 LVLH errors, reduced navigation state, inertial + radio navigation measurements, Worst scenario. ............................................ 87
5.31 Final position and velocity, reduced navigation state, inertial + radio navigation measurements, Best scenario. ............................................ 88
5.32 Final position and velocity, reduced navigation state, inertial + radio navigation measurements, Typical scenario. ............................................ 89
5.33 Final position and velocity, reduced navigation state, inertial + radio navigation measurements, Worst scenario. ............................................ 90
5.34 Crosstrack versus downrange errors, reduced navigation state, inertial + radio navigation measurements, Best scenario. ............................................ 91
5.35 Crosstrack versus downrange errors, reduced navigation state, inertial + radio navigation measurements, Typical scenario. ............................................ 91
5.36 Crosstrack versus downrange errors, reduced navigation state, inertial + radio navigation measurements, Worst scenario. ............................................ 92
5.37 ΔV used, all cases, Best errors; Clockwise from upper left: full navigation state with inertial measurements only, full navigation state with inertial and radio navigation measurements, reduced navigation state with inertial and radio navigation measurements, and reduced navigation state with inertial measurements only. ............................................ 94
5.38 ΔV used, all cases, Typical errors; Clockwise from upper left: full navigation state with inertial measurements only, full navigation state with inertial and radio navigation measurements, reduced navigation state with inertial and radio navigation measurements, and reduced navigation state with inertial measurements only. ............................................ 95
5.39 $\Delta V$ used, all cases, Worst errors; Clockwise from upper left: full navigation state with inertial measurements only, full navigation state with inertial and radio navigation measurements, reduced navigation state with inertial and radio navigation measurements, and reduced navigation state with inertial measurements only.
Chapter 1

Introduction

Despite the fact that it has been more than forty years since Neil Armstrong and Buzz Aldrin first stepped off the Eagle lander onto the lunar surface, returning to the moon remains a challenge and will require a great deal of technological and scientific innovation. There have been several different plans to return to the moon, most notably NASA’s now-canceled Constellation program. Now, after the space shuttle has completed its final missions, NASA has an opportunity to pursue missions that will take us out of low earth orbit once more [2]. It is highly likely that humans will one day return to the moon, although when that will happen remains somewhat unclear.

President Obama has outlined a plan for human space flight which includes missions beyond low Earth orbit, including missions to an asteroid and to Mars. As part of a long term plan to explore more of our solar system than ever before, manned missions to the moon can provide valuable experience in human survival beyond the earth. Missions to the moon can help verify heavy lift vehicles and environmental systems that can help in the journey to Mars. Also, the Apollo missions were only able to explore a small portion of the lunar surface, while future missions could provide full surface access and a long term presence on the moon.

One of the important components of a manned lunar mission is a method of safely landing on the moon’s surface and then returning to orbit. The Apollo missions accomplished this by the use of lunar module (LM) which landed on the moon, then returned to orbit to dock with the command module (CM) which remained in orbit for the duration of the mission. It is likely that future missions would use a similar approach and so a lunar lander would need to be developed.
A major consideration in designing a lunar lander is the amount of fuel which will be required to achieve mission objectives. Carrying too little fuel on board could be catastrophic, endangering astronauts and mission success, while carrying too much fuel adds unnecessary cost to the mission. Therefore, it is necessary to make accurate estimates of the amount of fuel which would be necessary to carry on board the lunar lander.

Monte Carlo analysis provides a good way to determine how much fuel is necessary to include on board the lander. It can provide a fuel estimate that takes real world conditions, such as thrust variations, vehicle state estimation errors, and sensor errors, into account. This research includes the development of a Monte Carlo simulation, which, in addition to being able to estimate fuel use, can also be used to develop a nominal trajectory for a lunar lander.

The thesis of this research is that a feedback control and guidance system with a reduced navigation state will perform nearly as well as an optimal full-state feedback system. The performance of these systems will be compared based on navigation state errors and fuel performance. This will be accomplished by building a Monte Carlo Simulation in Matlab which includes guidance, navigation, and attitude control and 6 degrees-of-freedom dynamics. The Monte Carlo simulation models the powered descent of a lunar lander vehicle from an arbitrary parking orbit about the moon to touchdown on the lunar surface. The simulation also has the capability to employ either a minimal sensor suite with accelerometers, gyroscopes, a star tracker, an altimeter and a velocimeter or a more inclusive sensor suite which also includes a radio navigation system.
Chapter 2

Lunar Descent Guidance

A closed loop guidance algorithm for landing on the moon needs to meet two major requirements. It needs to be able to provide commands to guide a vehicle from an initial position and velocity to a final desired position and velocity. In addition, it needs to be simple enough that it can run in real time on board the spacecraft. These requirements need to also be balanced with other desirable traits of a guidance system, such as minimizing fuel use [3]. Solving for a truly optimal trajectory is a complicated and unwieldy problem, so most guidance methods sacrifice some performance for the sake of utility.

2.1 Apollo Guidance

A summary of the methods used for guidance, navigation, and control during the Apollo missions is provided by Klumpp [1]. During the Apollo missions, descent of the Lunar Module (LM) was separated into three guidance phases: the braking phase (P63), the approach phase (P64), and terminal descent (P66). A diagram of these phases can be seen in Fig. 2.1. Both the braking phase and the approach phase employed a targeted guidance algorithm, where the current position, velocity, and acceleration were compared with a target position, velocity, and acceleration. The guidance algorithm would then compute a unit thrust vector and a throttle value. Targets were selected to provide constant visibility of the lunar surface, so that the LM pilot could select a new landing site, if necessary, during descent. Both of these phases made use of the Apollo explicit guidance equation, Eq. 2.1.

\[
a_{cmd} = 12 \frac{(r_t - r_0)}{T^2} + 6 \frac{(v_t + v_0)}{T} + a_t
\]

The braking phase began when the LM was within a 492-km slant range of the landing site. The targets for this phase were determined iteratively preflight, with the constraint
that the throttle remained either at the maximum allowable level (93%), or between 11-65%. This range was selected to avoid fuel cavitation problems.

The approach phase began with the vehicle at an altitude of approximately 2.2 km and a ground range of about 7.5 km from the landing site. At this point, the vehicle would have a forward velocity of approximately 129 m/s and an altitude rate of about -44 m/s. At the end of the approach phase, the vehicle would have a very small forward velocity, and would be almost directly over the desired landing location.

Finally, terminal descent began when the vehicle was 11 m, ground range, from the landing site and at an altitude of 30 m. During this phase, all non-vertical velocity was nulled, and the vertical velocity was reduced in order to land safely. There was no control over position during this phase of descent, only velocity. The vertical velocity of the LM was controlled manually by the LM pilot by means of a switch which reduced or increased the velocity in increments of 0.3 m/s.

2.2 Optimal Rocket Problem

A simple problem which has applications to powered lunar descent is the optimization
of a rocket’s trajectory (for maximum range, minimum time or other desired performance indices) for a rocket with constant thrust in a constant gravity field. This problem is addressed in both Lawden [4] and Bryson and Ho [5]. A diagram of the problem is shown in Fig. 2.2.

![Diagram of optimal rocket trajectory problem.](image)

Fig. 2.2: Optimal rocket trajectory problem.

The equations of motion for a rocket experiencing planar motion are given in Eq. 2.2-2.5. The acceleration experienced by the rocket is $a(t)$. The direction of the thrust, $\beta$, is the control that will be changed to maximize or minimize the performance index.

\[
\begin{align*}
\dot{r}_x &= v_x \\
\dot{r}_y &= v_y \\
\dot{v}_x &= a(t) \cos \beta \\
\dot{v}_y &= a(t) \sin \beta - g
\end{align*}
\]

The Hamiltonian for this problem is Eq. 2.6.
\[ H = \lambda_r v_x + \lambda_r v_y + \lambda v_x (a \cos \beta) + \lambda (a \sin \beta - g) \]  \hfill (2.6)

Along an optimal trajectory, the derivative of the Hamiltonian with respect to the control is zero, so the following equation can be written as shown in Eq. 2.7.

\[ \frac{\partial H}{\partial \beta} = -\lambda v_x (a \sin \beta) + \lambda v_y (a \cos \beta) = 0 \]  \hfill (2.7)

This leads to form of the optimal control shown in Eq. 2.8.

\[ \tan \beta = \frac{\lambda v_y}{\lambda v_x} \]  \hfill (2.8)

Differential equations for the multipliers can be found by taking the derivative of the Hamiltonian, \( H \), with respect to the state, \( x \). These differential equations are Eq. 2.9-2.13.

\[ \dot{\lambda}_i = -\frac{\partial H}{\partial x_i} \]  \hfill (2.9)

\[ \dot{\lambda}_r v_x = 0 \]  \hfill (2.10)
\[ \dot{\lambda}_r v_y = 0 \]  \hfill (2.11)
\[ \dot{\lambda}_v v_x = -\lambda_r v_x \]  \hfill (2.12)
\[ \dot{\lambda}_v v_y = -\lambda_r v_y \]  \hfill (2.13)

The general solution to these equations is given in Eq. 2.14-2.17.
\[ \lambda_{r_x} = c_1 \]  
\[ \lambda_{r_y} = c_2 \]  
\[ \lambda_{v_x} = c_3 - c_1 t \]  
\[ \lambda_{v_y} = c_4 - c_2 t \]  

(2.14)  
(2.15)  
(2.16)  
(2.17)

When Eq. 2.16 and Eq. 2.17 are substituted into Eq. 2.8, the optimal control has a form known as the bilinear tangent law, shown in Eq. 2.18.

\[ \tan \beta = \frac{c_4 - c_3 t}{c_2 - c_1 t} \]  

(2.18)

The coefficients in Eq. 2.18 are determined by taking the derivative of the performance index (the quantity to be maximized or minimized) with respect to the state. For many optimization problems, the horizontal position, \( x \), is unconstrained. This leads to the condition that \( c_1 \) is zero, and then the optimal control law is given by the linear tangent law, Eq. 2.19.

\[ \tan \beta = d_2 - d_1 t \]  

(2.19)

### 2.3 Modified Apollo Guidance

A number of different guidance methods have been proposed for use in the next lunar lander. Like the guidance method used during the Apollo missions, many of these methods have a nominal path described by a low (between 2 and 5) degree polynomial. One such method was proposed by Ronald Sostaric and Jeremy Rea [6]. This method was used in a linear covariance analysis by Travis Moesser [7]. The lunar lander guidance generally begins once the vehicle is in an orbit about the moon. Obviously, a separate guidance algorithm would be necessary to take the vehicle from the earth to this lunar orbit [8].

The guidance algorithm used in the Monte Carlo simulation is based primarily on
methods proposed by Sostaric and Rea [6]. This algorithm begins after a deorbit burn takes the lunar lander from a parking orbit about the moon to an intermediate elliptical orbit with a perilune altitude of about 15 kilometers. After leaving the intermediate orbit, powered lunar descent begins. This mission scenario can be seen in Fig. 2.3.

![Diagram of lunar mission scenario](image)

**Fig. 2.3:** Lunar descent mission scenario.

Powered lunar descent is performed in three distinct phases. First there is the braking phase, which is the longest phase of powered descent, and which begins when the vehicle is within a 321 km slant range of the landing site. The primary purpose of the braking phase is to decrease the horizontal velocity of the lander. This is also the phase in which most of the altitude change occurs. Then there is the approach phase, when the lander continues to slow down and begins to pitch over so that it can land vertically. The approach phase begins 335 seconds after the start of the braking phase. The final stage is terminal descent, in which nearly all horizontal movement of the lander has ceased, and the lander descends vertically to come to a soft landing on the surface of the moon. Terminal descent begins 60 seconds after the beginning of the approach phase. These phases are the similar to those used in the guidance algorithm from the Apollo missions, and can be seen in Fig. 2.4. They have the same names and purposes as the guidance phases during the Apollo missions, but
they use different targets and take slightly less time.

![Phases of powered lunar descent.](image)

**Fig. 2.4: Phases of powered lunar descent.**

### 2.3.1 Guidance During Intermediate Orbit

Between the beginning of the simulation and the beginning of powered lunar descent, the lander performs a deorbit insertion burn to leave the initial parking orbit and enter an intermediate transfer orbit. The nominal initial orbit plane contains the desired landing site, but perturbations are added to the initial position and velocity for each data run, so the true initial orbital plane will generally not contain the landing site. However, any required plane change will occur during powered descent, so the intermediate orbit is chosen to lie in the same plane as the initial orbit. The intermediate orbit will be an elliptical orbit, with an apogee radius equal to the magnitude of the initial position and a perigee radius of 15.5 km.

The deorbit insertion burn is modeled as an impulsive maneuver, and the magnitude and direction of this burn are determined using Lambert’s method. Lambert’s method is an algorithm which calculates the path between two known position vectors given the transfer time between them [9], as shown in Eq. 2.20.

\[
[v_1, v_2] \leftarrow \text{Lambert}(\bar{r}_1, \bar{r}_2, \Delta t)
\] (2.20)

The initial position vector, \(\bar{r}_1\), is simply the position vector of the lander at the beginning of the simulation. The final position vector, \(\bar{r}_2\), is in the direction of the landing site, with a
length equal to the desired perigee radius, $r_{p,des}$, as shown in Eq. 2.21. The desired transfer time, $\Delta t$, is calculated as shown in Eq. 2.22 through Eq. 2.25. The semi-major axis of the desired transfer ellipse is $a_{des}$.

\[
\bar{r}_2 = r_{p,des} \frac{r_{LS}}{\bar{r}_{LS}}
\]  

(2.21)

\[
r_p = r_{p,des}
\]  

(2.22)

\[
r_a = |\bar{r}_1|
\]  

(2.23)

\[
a_{des} = \frac{(r_a + r_p)}{2}
\]  

(2.24)

\[
\Delta t = \frac{1}{2} \left( 2\pi \sqrt{\frac{a_{des}^3}{\mu_{moon}}} \right)
\]  

(2.25)

The outputs of Lambert’s algorithm, $\bar{v}_1$ and $\bar{v}_2$, are the necessary initial velocity for the desired path and the velocity at $\bar{r}_2$ on the desired path. The deorbit insertion burn can be calculated using $\bar{v}_1$ and $\bar{v}_{initial}$, the velocity of the lander at the beginning of the simulation, as shown in Eq. 2.26.

\[
\Delta v_{deorbit} = \bar{v}_1 - \bar{v}_{initial}
\]  

(2.26)

2.3.2 Braking Phase Guidance Equation

During the braking phase of descent, the lander follows a linear acceleration profile, as shown in Eq. 2.27. This differs from the braking phase during the Apollo missions, which forced the lander to follow a quadratic acceleration profile. The form of the guidance law during the braking phase, shown in Eq. 2.27, is meant to approximate the linear tangent law in Eq. 2.19, which is the optimal solution from optimal control theory. Note that the
time, \( t \), in this equation is the time lapsed since the current time. For the simulation, \( t \) is always equal to zero.

\[
\vec{a}_{cmd} = \vec{a}_0 + \vec{a}_1 t
\]  

(2.27)

A closed-form equation for calculating the coefficients, \( \vec{a}_0 \) and \( \vec{a}_1 \), is found by first integrating the desired acceleration profile between the current guidance time, \( t_0 \), and the guidance time at the end of the braking phase, \( t \), as shown in Eq. 2.28 and Eq. 2.29. The terms \( \vec{r}_0 \) and \( \vec{v}_0 \) represent the position and velocity at the current time. The desired position and velocity at the end of the braking phase are represented by \( \vec{r}_t \) and \( \vec{v}_t \), respectively.

\[
\vec{v}_t = \int_{t_0}^{t} \vec{a}_{cmd} = \vec{a}_0 (t - t_0) + \frac{1}{2} \vec{a}_1 (t - t_0)^2 + \vec{v}_0
\]

(2.28)

\[
\vec{r}_t = \int_{t_0}^{t} \vec{v}_t = \int_{t_0}^{t} \int_{t_0}^{t} \vec{a}_{cmd} = \frac{1}{2} \vec{a}_0 (t - t_0)^2 + \frac{1}{6} \vec{a}_1 (t - t_0)^3 + \vec{v}_0 (t - t_0) + \vec{r}_0
\]

(2.29)

An important substitution is given in Eq. 2.30. The term \( t_{go} \) is the time remaining until the end of the current guidance phase.

\[
t_{go} = t - t_0
\]

(2.30)

Making this substitution in Eq. 2.28 and Eq. 2.29 and rearranging terms results in Eq. 2.31 and Eq. 2.32.

\[
\vec{v} - \vec{v}_0 = \vec{a}_0 t_{go} + \frac{1}{2} \vec{a}_1 t_{go}^2
\]

(2.31)

\[
\vec{r}_t - \vec{r}_0 - \vec{v}_0 t_{go} = \frac{1}{2} \vec{a}_0 t_{go}^2 + \frac{1}{6} \vec{a}_1 t_{go}^3
\]

(2.32)

Equation 2.31 and Equation 2.32 can be written in matrix form, as shown in Eq. 2.33.
The matrix containing the \( t_{go} \) terms can be inverted and moved to the other side of the equation. Finally, the coefficients in Eq. 2.27 can be calculated using the Eq. 2.34.

$$\begin{bmatrix} \bar{v}_t - \bar{v}_0 \\ \bar{r}_t - \bar{r}_0 - \bar{v}_0 t_{go} \end{bmatrix} = \begin{bmatrix} t_{go} [I_{3x3}] & \frac{1}{2} t_{go}^2 [I_{3x3}] \\ \frac{1}{2} t_{go}^2 [I_{3x3}] & \frac{1}{6} t_{go}^3 [I_{3x3}] \end{bmatrix} \begin{bmatrix} \bar{c}_0 \\ \bar{c}_1 \end{bmatrix} \tag{2.33}$$

$$\begin{bmatrix} \bar{c}_0 \\ \bar{c}_1 \end{bmatrix} = \begin{bmatrix} -\frac{2}{t_{go}} [I_{3x3}] & \frac{6}{t_{go}^2} [I_{3x3}] \\ \frac{6}{t_{go}^2} [I_{3x3}] & -\frac{12}{t_{go}^3} [I_{3x3}] \end{bmatrix} \begin{bmatrix} \bar{v}_t - \bar{v}_0 \\ \bar{r}_t - \bar{r}_0 - t_{go} \cdot \bar{v}_0 \end{bmatrix} \tag{2.34}$$

2.3.3 Approach Phase Guidance Equation

During the approach phase of descent, the lander follows a quadratic acceleration profile. The guidance law for this portion of the trajectory is identical to that which was used during the Apollo missions. The guidance law for the approach phase is shown in Eq. 2.35. Again, \( t \) is the time lapsed from the current time.

$$\bar{a}_{cmd} = \bar{c}_0 + \bar{c}_1 t + \bar{c}_2 t^2 \tag{2.35}$$

As in the braking phase, the coefficients of Eq. 2.35 are determined by first integrating the acceleration profile between \( t_0 \) and \( t \), as shown in Eq. 2.36 and Eq. 2.37. The terms \( \bar{r}_t \) and \( \bar{v}_t \) now refer to the targets at the end of the approach phase, and \( \bar{a}_t \) is the acceleration target.

$$\bar{v}_t = \int_{t_0}^{t} \bar{c}_{cmd} = \bar{c}_0 (t - t_0) + \frac{1}{2} \bar{c}_1 (t - t_0)^2 + \frac{1}{3} \bar{c}_2 (t - t_0)^3 + \bar{v}_0 \tag{2.36}$$

$$\bar{r}_t = \int_{t_0}^{t} \bar{v}_t = \int_{t_0}^{t} \int_{t_0}^{t} \bar{a}_{cmd} \tag{2.37}$$

$$= \frac{1}{2} \bar{c}_0 (t - t_0)^2 + \frac{1}{6} \bar{c}_1 (t - t_0)^3 + \frac{1}{12} \bar{c}_2 (t - t_0)^4 + \bar{v}_0 (t - t_0) + \bar{r}_0 \tag{2.38}$$
The substitution from Eq. 2.30 is made, and Eq. 2.36 and Eq. 2.37 are rearranged to form Eq. 2.39 and Eq. 2.40.

\[
\vec{v}_t - \vec{v}_0 = \bar{c}_0 t_{go} + \frac{1}{2} \bar{c}_1 t_{go}^2 + \frac{1}{6} \bar{c}_2 t_{go}^3 \tag{2.39}
\]

\[
\vec{r}_t - \vec{r}_0 - \vec{v}_0 t_{go} = \frac{1}{2} \bar{c}_0 t_{go}^2 + \frac{1}{6} \bar{c}_1 t_{go}^3 + \frac{1}{12} \bar{c}_2 t_{go}^4 \tag{2.40}
\]

In matrix form, Eq. 2.35, Eq. 2.39, and Eq. 2.40 can be written as Eq. 2.41.

\[
\begin{bmatrix}
\vec{a}_t \\
\vec{v}_t - \vec{v}_0 \\
\vec{r}_t - \vec{r}_0 - \vec{v}_0 t_{go}
\end{bmatrix}
= \begin{bmatrix}
[I_{3x3}] & t_{go} [I_{3x3}] & t_{go}^2 [I_{3x3}]
end{bmatrix}
\begin{bmatrix}
\bar{c}_0 \\
\bar{c}_1 \\
\bar{c}_2
\end{bmatrix} \tag{2.41}
\]

Again, the \( t_{go} \) matrix is inverted, and the coefficients in Eq. 2.35 can be determined using Eq. 2.42.

\[
\begin{bmatrix}
\bar{c}_0 \\
\bar{c}_1 \\
\bar{c}_2
\end{bmatrix}
= \begin{bmatrix}
[I_{3x3}] & -\frac{6}{t_{go}} [I_{3x3}] & \frac{12}{t_{go}^2} [I_{3x3}]
end{bmatrix}
\begin{bmatrix}
\vec{a}_t \\
\vec{v}_t - \vec{v}_0 \\
\vec{r}_t - \vec{r}_0 - t_{go} \vec{v}_0
\end{bmatrix} \tag{2.42}
\]

### 2.3.4 Guidance During Terminal Descent

In the terminal descent phase, the vehicle is commanded to maintain a constant acceleration in the down direction. Movement in the north and east directions is nulled to zero. The acceleration in the down direction should be selected so that the vehicle slows to about 1 m/s before it touches down. This is slow enough to insure that the landing does not injure crew members or damage equipment in the lander.

A feedback guidance rule is used for each direction [10] (North, East and Down), as shown in Eq. 2.43 through Eq. 2.45. Control gains are represented by \( K \).
\[ a_N = -K_{vN}(v_{0,N} - v_{t,N}) - K_{rN}(r_{0,N} - r_{t,N}) \]  \hfill (2.43)

\[ a_E = -K_{vE}(v_{0,E} - v_{t,E}) - K_{rE}(r_{0,E} - r_{t,E}) \]  \hfill (2.44)

\[ a_D = a_{t,D} - K_{vD}(v_{0,D} - v_{t,D}) \]  \hfill (2.45)

\[ 2.3.5 \ \text{Guidance Targets} \]

The equations for calculating the guidance law coefficients for the braking phase and the approach phase (Eq. 2.34 and Eq. 2.42) include the term \( t_{go} \) in their denominators. As the vehicle approaches the target, \( t_{go} \) gets close to zero, which causes the coefficients to become very large. There are two ways to solve this problem. The first option is to stop recalculating the coefficients once a specified time has passed, called a fine count. The fine count is usually the last few seconds of the current guidance phase. The second option is to choose targets which are not actually intended to be reached. The guidance phase is terminated before the vehicle reaches the target, and so \( t_{go} \) never actually reaches zero. In this case, the desired state of the vehicle at the end of the guidance phase lies along the path specified by the chosen target.

For this simulation, a fine count was utilized. A fine count of 2 seconds was used for the braking and terminal descent phases and a fine count of 4 seconds was used for the approach phase. In cases where error levels are higher, the fine count may need to be even longer, as this guidance method can command very large accelerations if the trajectory deviates greatly from the nominal. The targets for each guidance phase are the desired state of the vehicle at the end of that guidance phase. Depending on the phase, these targets can include position, velocity, and acceleration.

The initial conditions of the vehicle at the beginning of each guidance phase is summarized in Table 2.1.

At the end of the approach phase, the lander should be directly over the landing site.
with near-zero forward or lateral velocity and a constant vertical acceleration. This will prepare the vehicle to begin terminal descent. The final desired position for terminal descent is simply the landing site. At the end of terminal descent, the vehicle should be moving downward at about 1 m/s. This is slow enough for a soft landing.

The targets selected for this simulation are summarized in Table 2.2. The targets are given in the landing site reference frame. These targets were included in the paper by Ronald Sostaric and Jeremy Rea [6], and were determined originally through trial and error using a Monte Carlo simulation.

<table>
<thead>
<tr>
<th>Table 2.1: Vehicle conditions at start of guidance phases.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Braking Phase</td>
</tr>
<tr>
<td>$t_{go}$</td>
</tr>
<tr>
<td>$h$</td>
</tr>
<tr>
<td>$\dot{h}$</td>
</tr>
<tr>
<td>Pitch</td>
</tr>
<tr>
<td>Throttle</td>
</tr>
<tr>
<td>$r_{v/LS}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2.2: Guidance targets, by phase.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Braking Phase</td>
</tr>
<tr>
<td>$N$</td>
</tr>
<tr>
<td>$r_t$</td>
</tr>
<tr>
<td>$D$</td>
</tr>
<tr>
<td>$N$</td>
</tr>
<tr>
<td>$v_t$</td>
</tr>
<tr>
<td>$D$</td>
</tr>
<tr>
<td>$N$</td>
</tr>
<tr>
<td>$a_t$</td>
</tr>
<tr>
<td>$D$</td>
</tr>
</tbody>
</table>
Chapter 3

Lunar Descent Navigation

In a perfect world, it would be possible to know the position and velocity of a spacecraft exactly, and that position and velocity could simply be propagated using a deterministic model in order to find the state of the spacecraft at a later time. In reality, the position and velocity of a vehicle are never known exactly, and cannot usually be measured directly. Also, any sensors used to make measurements on a system have errors and do not always provide information on all of the components of the system’s state. In addition, it is impossible to create a model which perfectly describes a system and takes all possible disturbances into account. This makes a method for estimating the vehicle state a necessity.

3.1 Apollo Navigation

During the Apollo missions, the lunar module (LM) employed a radar system and an inertial measurement unit (IMU) for navigation. The radar could operate at altitudes of less than 10 km, and velocities less than 610 m/s. The IMU operated during all powered portions of the flight, but was turned off during coasting phases to prevent errors in the estimated vehicle state from being propagated. The measurements from these sensors were processed in a type of optimum linear estimator known as a Kalman filter, as described in Kriegsman [11]. This filter was used to estimate the position and velocity of the LM. Accurate estimation of the vehicle state was necessary to use fuel efficiently, maintain landing site visibility, and to slow the vehicle so that it was possible to land without harm to the vehicle or its occupants.

3.2 Inertial Navigation

In order to perform inertial navigation (or even to be able to adequately describe the
lunar landing problem), it is necessary to define a number of different coordinate frames. Appropriate transformation matrices can be used to transform from one reference frame to another, although alternative methods, such as quaternions, can also be used [12]. An inertial frame is necessary in order to apply Newton’s second law to a problem. A non-rotating frame attached to the center of the moon, known as the Moon-centered inertial (MCI) is not an inertial frame in the strictest sense, but it can be considered to be a inertial frame in the region near the moon, just as an Earth-centered inertial frame can be used in the vicinity of Earth [12, 13]. The z-axis of the MCI frame is defined as the direction of the lunar north pole at the J2000 epoch. The MCI x-axis is in the direction of the lunar prime meridian at the J2000 epoch, and the MCI y-axis is orthogonal to the x- and z-axes.

A second important frame is the Moon-fixed (MF) frame, which is centered at the lunar center of mass and rotates with the moon. The z-axis of the MF frame is in the same direction as the MCI z-axis, but the MF x-axis and the MF y-axis rotate at the same rate as the moon, with the x-axis always pointing in the direction of the prime meridian. A diagram showing the MCI and MF frames can be seen in Fig. 3.1.

In addition, a reference frame centered at a potential landing site and fixed at that point on the lunar surface can also be useful. The Landing Site (LS) frame can be seen in Fig. 3.2. The LS frame is a North-East-Down (NED) coordinate frame, with the x-axis pointing east, the y-axis pointing north, and the z-axis pointing down.

It is also necessary to define a body-fixed frame, which is located at the center of mass for the vehicle and includes roll ($X_B$), pitch ($Y_B$), and yaw ($Z_B$) axes. The roll axis points to the front of the vehicle, the pitch axis points to the right-hand side of the vehicle, and the yaw axis points down [12]. A vehicle’s IMU does not necessarily lie at its center of mass, so a separate frame may also be defined for the IMU case. A diagram of these frames can be seen in Fig. 3.3.

In addition, a frame known as the local-vertical local horizontal frame (LVLH) can also prove useful for orbital mechanics problems. The LVLH frame is fixed on the vehicle, with the local vertical axis pointing away from the moon (or whatever body the vehicle is
orbiting), with the cross track perpendicular to the orbital plane, and the local horizontal completing the orthogonal triad [9]. This frame is shown in Fig. 3.3.

3.3 Standard Kalman Filter

A common tool used for estimating the state of a spacecraft is a Kalman filter. A Kalman filter is an algorithm that processes measurements to estimate the state of the vehicle, including its position and velocity [14]. Kalman filters are considered optimal filters, and they can combine data from many different measurements, taking into account the accuracy of those measurements. In addition to estimating the position and velocity of a vehicle, a Kalman filter can also estimate sensor errors, such as biases. One major benefit of the Kalman filter is that the estimates of the quantities of interest typically improves as more measurements become available.

There are three major assumptions made in the development of a standard Kalman filter. First, the system of interest is assumed to be linear or linearizable. Although few systems are truly linear, this is still a valid assumption because a system can often be
modeled as linear with an acceptable amount of error, and systems can also be linearized about a nominal point or trajectory. Secondly, any noise in the system is modeled as white noise or as colored noise driven by white noise. This assumption can be justified by the fact that systems generally only respond to inputs within a certain range of frequencies, so if noise appears white over this band, it will look white to the system. Finally, any noise in the system is also assumed to have a Gaussian probability distribution function. This is valid because noise often comes from multiple sources, and the sum of the probability densities of multiple random variables (regardless of their individual probability densities) tends to be very nearly Gaussian [14].

The equations used in a Kalman filter can be found in Crassidis and Junkins [15]. There are formulations of the Kalman filter for both discrete and continuous systems and incorporating both continuous and discrete measurements. The following formulation is for a Kalman filter with continuous system dynamics and discrete measurements. The system has the model shown in Eq. 3.1. The system state, which includes all variables which are going to be estimated, is \( \bar{x}(t) \), and the state covariance matrix is \( P(t) \). The control vector is \( \bar{u}(t) \), and \( \bar{w}(t) \) is white process noise with strength \( Q \).

\[
\dot{\bar{x}}(t) = F(t)\bar{x}(t) + B(t)\bar{u}(t) + G(t)\bar{w}(t)
\]  

(3.1)
The measurement model is shown in Eq. 3.2. The modifier ($\tilde{\cdot}$) indicates a measured quantity. This model includes white sensor noise, $\tilde{\nu}_k$, with zero mean and covariance $R_k$.

$$\tilde{y}_k = H_k \bar{x}_k + \tilde{\nu}_k$$  \hspace{1cm} (3.2)

It is important to note that the white noise in the dynamic model and the white noise in the measurement model are assumed to be uncorrelated.

The Kalman filter must first be initialized with a state and covariance. Then, the state and covariance can be propagated using the continuous differential equations in Eq. 3.3 and Eq. 3.4. The modifier ($\hat{\cdot}$) indicates an estimated quantity.

$$\dot{\hat{x}} = F(t)\hat{x}(t) + B(t)\bar{u}(t)$$  \hspace{1cm} (3.3)

$$\dot{\hat{P}}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t)$$  \hspace{1cm} (3.4)
Each time a new measurement is available, the state and covariance can be updated using Eq. 3.5 and Eq. 3.6.

\[
\hat{x}_k^+ = \hat{x}_k^- + K_k [\tilde{y}_k - H_k \hat{x}_k^-] \\
(3.5)
\]

\[
P_k^+ = [I - K_k H_k] P_k^- \\
(3.6)
\]

The Kalman gain, \(K\), is derived to minimize the covariance of the state vector. It can be calculated using Eq. 3.7.

\[
K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \\
(3.7)
\]

The above equations are only applicable to linear systems. In order to apply a Kalman filter to a system which is nonlinear, it is necessary to use an extended Kalman filter (EKF). This filter is not necessarily optimal, but has been applied to nonlinear systems very successfully [14,15].
3.4 Extended Kalman Filter

The navigation filter used in the Monte Carlo Simulation is an extended Kalman filter (EKF) employing continuous dynamics and discrete measurements. In an EKF, the system dynamics follow the general form of Eq. 3.8.

\[
\dot{x} = f(\dot{x}, t) + G(t)\dot{w}(t) \tag{3.8}
\]

To update the estimate of the navigation state, it is necessary to develop a model for each type of sensor measurement to be processed in the EKF. These models are similar to the sensor models used to calculate simulated measurements using the truth state, and follow the general form of Eq. 3.9. This model includes a white noise process, $\nu_k$.

\[
\bar{y}_k = h(\bar{x}_k) + \nu_k \tag{3.9}
\]

The white noise process, $\nu_k$, is assumed to be zero mean, with the covariance shown in Eq. 3.10.

\[
[\nu_k \nu^T_j] = R_k \delta_{kj} \tag{3.10}
\]

In order to begin state estimation using an EKF, it is necessary to initialize the filter with an initial state and an initial covariance matrix, $\bar{x}_0$ and $P_0$. Then, at each time step, the state and covariance are propagated by integrating Eq. 3.11 and Eq. 3.12. The matrix $F$ is the partial derivative of the dynamics with respect to the state, as shown in Eq. 3.13. The matrix $Q$ is the strength of the process noise in the system dynamics, $\dot{w}$.

\[
\dot{\hat{x}}(t) = f(\hat{x}, t) \tag{3.11}
\]

\[
\dot{P}(t) = F(\hat{x}(t), t)P(t) + P(t)F^T(\hat{x}(t), t) + G(t)Q(t)G^T(t) \tag{3.12}
\]
\[ F(\hat{x}(t), t) = \delta \frac{\delta f}{\delta \hat{x}} \hat{x}(t) \]  

(3.13)

Note that Eq. 3.11 is the same as the model for the state dynamics in Eq. 3.8, but without any white noise. Equation 3.12 is the differential Ricatti equation. It is possible to approximate Eq. 3.12 using the discrete Ricatti equation, as shown in Eq. 3.14 and Eq. 3.15. This method is valid as long as the navigation time step is small, and will be used in this simulation in order to shorten the computation time. The matrix \( \Phi \) is the state transition matrix for \( F \), as shown in Eq. 3.16.

\[
P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \int_{t_i}^{t_{i+1}} \Phi_k G Q G^T \Phi_k^T d\tau
\]

(3.14)

\[
P_{k+1}^- \approx \Phi_k P_k^+ \Phi_k^T + G Q G^T \Delta t
\]

(3.15)

\[
\Phi = e^{F \Delta t}
\]

(3.16)

When a measurement is available at a given time step, the state estimate is updated before it is propagated to the next time step. The equations used to update the state and covariance estimates are Eq. 3.17 and Eq. 3.18. The equation to update the state includes both a predicted measurement vector, \( \hat{y}_k \), which is an estimate of what the measurement should be based on the current navigation state, and the actual measurement vector, \( \tilde{y}_k \), which is the output of one or more sensors. The form of the covariance update equation shown in Eq. 3.18 is known as the Joseph formulation, and provides better numerical stability than the standard formulation.

\[
\hat{x}_n^+ = \hat{x}_n^- + K_k [\tilde{y}_k - \hat{y}_k]
\]

(3.17)

\[
P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T
\]

(3.18)
The estimated measurement, $\hat{y}_k$, and the measurement partial derivative matrix, $H_k$, are given by Eq. 3.19 and Eq. 3.20, respectively.

$$\hat{y}_k = h(\hat{x}_k)$$ \hspace{1cm} (3.19)

$$H_k = \frac{\delta h}{\delta \bar{x}}|_{\hat{x}_k^-}$$ \hspace{1cm} (3.20)

Also, the Kalman gain, $K_k$, is 3.21.

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}$$ \hspace{1cm} (3.21)

### 3.5 Kalman Filter for Lunar Descent

The navigation algorithm for this research is summarized in the flow chart shown in Fig. 3.5. The navigation algorithm is an extended Kalman filter which updates the state estimate if there is a measurement available, and, between measurements, propagates the current state estimate using the state dynamics.

#### 3.5.1 States Included in the Navigation Filter

It was desired to compare the performance (namely, the navigation errors and estimated fuel use) of two extended Kalman filters (EKF) with different navigation state vectors. The values that were included in each state vector are shown in Table 3.1. The reduced state filter uses a navigation state which only includes the position and velocity of the lunar lander and the position of the landing site. The full state filter uses a more inclusive navigation state, which incorporates sensor biases, misalignment errors, and scale factor errors. Including sensor errors in the navigation state allows the EKF to improve the estimates of these errors as measurements are taken, which was expected to cause an improvement in the estimate of the vehicle’s position and velocity. This may lead to a better estimate of fuel use.
Fig. 3.5: Navigation algorithm overview.
## Table 3.1: Navigation states.

<table>
<thead>
<tr>
<th>Navigation Filter State</th>
<th>Reduced State</th>
<th>Full State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Position</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Velocity</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Landing Site Position</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Accelerometer Bias</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Scale Factor</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Misalignment</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Altimeter Bias</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Scale Factor</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Velocimeter Bias</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Scale Factor</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Misalignment</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Radio Navigation Bias</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Range Scale Factor</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Radio Navigation Bias</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Range Rate Scale Factor</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>9</td>
<td>33</td>
</tr>
</tbody>
</table>

The reduced navigation state is shown in Eq. 3.22 and the full navigation state is shown in Eq. 3.23. The terms $\vec{E}_{\text{accel}}$, $\vec{E}_{\text{alt}}$, $\vec{E}_{\text{vel}}$, $\vec{E}_{\text{range}}$, and $\vec{E}_{\text{rrate}}$ are vectors of sensor errors, as shown.
\[ \bar{x}_{\text{reduced}} = \begin{bmatrix} \bar{r} \\ \bar{v} \\ \bar{r}_{LS} \end{bmatrix} \] (3.22)

\[ \bar{x}_{\text{full}} = \begin{bmatrix} \bar{x}_{\text{reduced}} \\ \bar{b}_{\text{accel}} \\ \bar{s}_{\text{accel}} \\ \bar{\varepsilon}_{\text{accel}} \\ \bar{b}_{\text{alt}} \\ \bar{s}_{\text{alt}} \\ \bar{\varepsilon}_{\text{alt}} \\ \bar{b}_{\text{vel}} \\ \bar{s}_{\text{vel}} \\ \bar{\varepsilon}_{\text{vel}} \\ \bar{b}_{\text{range}} \\ \bar{s}_{\text{range}} \\ \bar{\varepsilon}_{\text{range}} \end{bmatrix} = \begin{bmatrix} \bar{x}_{\text{reduced}} \\ \bar{b}_{\text{accel}} \\ \bar{s}_{\text{accel}} \\ \bar{\varepsilon}_{\text{accel}} \\ \bar{b}_{\text{alt}} \\ \bar{s}_{\text{alt}} \\ \bar{\varepsilon}_{\text{alt}} \\ \bar{b}_{\text{vel}} \\ \bar{s}_{\text{vel}} \\ \bar{\varepsilon}_{\text{vel}} \\ \bar{b}_{\text{range}} \\ \bar{s}_{\text{range}} \\ \bar{\varepsilon}_{\text{range}} \end{bmatrix} \] (3.23)

### 3.5.2 Navigation State Dynamics

The dynamic model used in the EKF for the vehicle position and velocity is Eq. 3.24. The acceleration due to the oblateness of the moon is \( \bar{a}_{\text{J2}} \), and the equation used to calculate it is Eq. 3.25, where \( R_{\text{eq}} \) is the lunar equatorial radius, \( J_2 \) is a coefficient equal to \( 2.033542 \times 10^{-4} \), and \( n \) is the vector \( [0 0 1]^T \). The non-gravitational acceleration, \( \bar{a}_{\text{nongrav}} \), includes thrust and any other accelerations not due to gravity.

\[
\begin{bmatrix}
\dot{\bar{r}} \\
\dot{\bar{v}}
\end{bmatrix} = \begin{bmatrix}
\bar{v} \\
-\mu_{\text{moon}} \frac{\bar{r}}{|\bar{r}|^3} + \bar{a}_{\text{J2}} + \bar{a}_{\text{nongrav}} + \bar{w}_a
\end{bmatrix}
\] (3.24)

\[
\bar{a}_{\text{J2}} = -\frac{\mu J_2 R_{\text{eq}}}{2 |\bar{r}|^5} \left\{ 6 (\bar{r} \cdot \bar{n}) \bar{n} + 3\bar{r} - 15 \left( \frac{\dot{\bar{r}} \cdot \bar{n}}{2} \right) \right\}
\] (3.25)
The term $\tilde{w}_a$ is white noise on the acceleration of the vehicle. This white noise is assumed to be zero-mean.

Assuming that the moon is rotating at a constant rate, and that the landing site is not changing, the dynamics for the landing site are shown in Eq. 3.26.

$$[\dot{r}_{LS}] = [\tilde{\omega}_{moon} \times \tilde{r}_{LS}] \quad (3.26)$$

All of the sensor errors, including bias, scale factor, and misalignment errors, are assumed to be constant, so their derivatives are zero, as shown in Eq. 3.27 through 3.31. These derivatives are only employed when a large navigation state vector is selected; otherwise, only the dynamics of the vehicle and the landing site are propagated in the navigation filter.

$$\dot{\tilde{E}}_{accel} = \tilde{0}_{2x1} \quad (3.27)$$
$$\dot{\tilde{E}}_{alt} = \tilde{0}_{2x1} \quad (3.28)$$
$$\dot{\tilde{E}}_{vel} = \tilde{0}_{2x1} \quad (3.29)$$
$$\dot{\tilde{E}}_{range} = \tilde{0}_{2x1} \quad (3.30)$$
$$\dot{\tilde{E}}_{rrate} = \tilde{0}_{2x1} \quad (3.31)$$

### 3.5.3 Navigation State Propagation

In order to propagate the navigation state estimate, a set of differential equations is integrated. These differential equations are very similar to the dynamics models given in Eq. 3.24 through Eq. 3.31.

Equation 3.32 is used to propagate the vehicle position and velocity. The term $\tilde{a}_{accel}$ is the output of the accelerometer, as shown in Eq. 3.34. When the reduced navigation state is used, the accelerometer errors ($\tilde{\varepsilon}_{accel}$, $\tilde{s}_{accel}$, and $\tilde{b}_{accel}$) included in Eq. 3.32 are assumed to be zero because a better estimate is not available. When the full navigation state is used, the current estimates of these errors are used, as shown.
\[
\begin{bmatrix}
\dot{\hat{r}} \\
\dot{\hat{v}}
\end{bmatrix} =
\begin{bmatrix}
\dot{\mathbf{v}} \\
-\mu_{\text{moon}} \frac{\hat{r}}{|\hat{r}|^3} + \hat{a}_{J2} + (I_{3\times3} - \mathbf{\hat{e}}_{\text{accel}} \times)^{-1} (I_{3\times3} + \mathbf{\hat{s}}_{\text{accel}})^{-1} (\mathbf{\hat{a}}_{\text{accel}} - \mathbf{\hat{b}}_{\text{accel}})
\end{bmatrix} 
\]  
(3.32)

\[
\hat{a}_{J2} = -\frac{\mu J_2 R_{eq}}{2 |\hat{r}|^5} \left\{ 6 (\hat{r} \cdot \hat{n}) \hat{n} + 3\hat{r} - 15 \left( (i_r \cdot \hat{n})^2 \hat{r} \right) \right\}
\]  
(3.33)

Unlike other measurements used in the EKF, the accelerometer measurement is processed during the state propagation phase. In order to utilize accelerometer measurements in the state update phase of a Kalman filter, it would be necessary to create a model of the non-gravitational accelerations acting on the vehicle [16]. An accurate model of these accelerations can be difficult, if not impossible, to develop. The traditional approach to avoid this problem is to integrate the non-gravitational, compensated acceleration measurements directly in the propagation step of the navigation process, if using a Kalman filter, in a process known as dead-reckoning. The gravitational acceleration acting on the vehicle, as determined from an appropriate model, can be added to the non-gravitational accelerations in order to create a complete model of the accelerations acting on the vehicle [12]. The total acceleration is then integrated to obtain the vehicle position and velocity. Compensated gyroscope measurements can also be processed in a similar manner in order to determine the attitude of the vehicle.

The model for the accelerometer measurement is given in Eq. 3.34.

\[
\mathbf{\tilde{a}}_{\text{accel}} = (I_{3\times3} + \text{diag}(\mathbf{\hat{s}}_{\text{accel}}))(I_{3\times3} - \mathbf{\hat{e}}_{\text{accel}} \times)\mathbf{\tilde{a}}_{\text{nongrav}} + \mathbf{\hat{b}}_{\text{accel}} + \mathbf{\tilde{v}}_{\text{accel}}
\]  
(3.34)

In order to propagate the position of the landing site, Eq. 3.35 is used.

\[
\begin{bmatrix}
\dot{\hat{r}}_{LS} \\
\dot{\hat{v}}_{LS}
\end{bmatrix} = \begin{bmatrix}
\bar{\omega}_{\text{moon}} \times \hat{r}_{LS} \\
\end{bmatrix}
\]  
(3.35)

When the large navigation state is selected, the additional navigation states are propagated using Eq. 3.36 through Eq. 3.39.
\[ \dot{b}_{alt} = \dot{s}_{alt} = 0 \] (3.36)

\[ \dot{b}_{vel} = \dot{s}_{vel} = \dot{s}_{vel} = 0 \] (3.37)

\[ \dot{b}_{range} = \dot{s}_{range} = 0 \] (3.38)

\[ \dot{b}_{rrate} = \dot{s}_{rrate} = 0 \] (3.39)

In order to propagate the state covariance matrix, Eq. 3.40 is used. The state transition matrix is \( \Phi_k \), and is defined in Eq. 3.41. The simulation time step is \( dt_{\text{sim}} \). The strength, \( Q \), of the process noise in the dynamics model is also necessary to propagate the state covariance.

\[ P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + GQG^T dt_{\text{sim}} \] (3.40)

\[ \Phi_k = e^{F dt_{\text{sim}}} \] (3.41)

The matrices \( F \) and \( G \) are the partial derivatives of the state dynamics with respect to the state and the state dynamics with respect to the process noise, respectively. Both are evaluated at the current estimate of the state, \( \hat{x} \). The matrix \( F \) for the reduced navigation state is shown in Eq. 3.42 and for the full navigation state in Eq. 3.45.

\[ F_{\text{reduced}} = \frac{\delta f_{\text{reduced}}}{\delta \hat{x}_{\text{reduced}}} \bigg|_{\hat{x}(t)} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} \\ \delta \hat{\theta} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \delta \hat{\phi}_{SL} \end{bmatrix} \] (3.42)

\[ \frac{\delta \dot{v}}{\delta \hat{r}} = -\frac{\mu}{|\hat{r}|^3} \left( I_{3\times3} - 3\hat{r}\hat{r}^T \right) - \frac{5\mu J_2 R_{eq}}{2|\hat{r}|^6} \left\{ 6 (\hat{r}^T \hat{n}) \hat{n} + 3\hat{r} - 15 \left( \hat{r}^T \hat{n} \right)^2 \hat{r} \right\} \hat{r} \] (3.43)

\[ -\frac{\mu J_2 R_{eq}}{2|\hat{r}|^6} \left\{ 6\hat{n}\hat{n}^T + 3I_{3\times3} - 15 \left[ \left( \hat{r}^T \hat{n} \right)^2 I + 2 \left( \hat{r}^T \hat{n} \right) \hat{n} \hat{n}^T \left( I_{3\times3} - \hat{r}\hat{r}^T \right) \right] \right\} \]
\[ \frac{\delta \hat{r}_{LS}}{\delta \hat{r}_{LS}} = [\tilde{\omega}_{\text{moon}} \times] \quad (3.44) \]

\[ F_{\text{full}} = \frac{\delta \vec{f}_{\text{full}}}{\delta \vec{x}_{\text{full}}} \bigg|_{\vec{z}(t)} = \begin{bmatrix} F_{\text{reduced}} & \frac{\delta \hat{\xi}_{\text{reduced}}}{\delta E_{\text{accel}}} & 0_{9 \times 15} \\ 0_{24 \times 9} & 0_{24 \times 24} \end{bmatrix} \quad (3.45) \]

\[ \frac{\delta \hat{x}_{\text{reduced}}}{\delta E_{\text{accel}}} = \begin{bmatrix} 0_{3 \times 9} \\ \frac{\delta \hat{\phi}}{\delta \phi_{\text{accel}}} \\ 0_{3 \times 9} \end{bmatrix} \quad (3.46) \]

\[ \frac{\delta \hat{\phi}}{\delta E_{\text{accel}}} = \begin{bmatrix} \frac{\delta \hat{\phi}}{\delta \phi_{\text{accel}}} \\ \frac{\delta \hat{\phi}}{\delta \phi_{\text{accel}}} \\ \frac{\delta \hat{\phi}}{\delta \phi_{\text{accel}}} \end{bmatrix} \]

\[ = \begin{bmatrix} I_{3 \times 3}, (I_{3 \times 3} - \text{diag}(s)) [\bar{a}_{\text{thr}} \times], \text{diag} [(I_{3 \times 3} - \bar{\phi}_{\text{accel}} \times) \bar{a}_{\text{thr}}] \end{bmatrix} \quad (3.47) \]

\[ \text{The matrix } G \text{ is given in Eq. 3.49 and Eq. 3.50.} \]

\[ G_{\text{reduced}} = \begin{bmatrix} 0_{9 \times 3} \\ I_{3 \times 3} \\ 0_{9 \times 3} \end{bmatrix} \quad (3.49) \]

\[ G_{\text{full}} = \begin{bmatrix} G_{\text{reduced}} \\ 0_{24 \times 3} \end{bmatrix} \quad (3.50) \]

### 3.5.4 Measurement Models

One necessary part of navigation is creating models for the various sensors used in the navigation process. Good examples of the types of sensor models necessary for a powered lunar descent problem, including accelerometers, gyroscopes, altimeters, and velocimeters, can be found in Christensen [17] and Geller [18].

Equation 3.51 is the model for the altimeter, and it includes a terrain altitude term, \( h_{\text{terrain}} \), which is used to represent the variability of the lunar terrain, and is modeled as a
first order Markov process. Equation 3.52 is the model for the velocimeter. The models for the range and range rate measurements from the radio navigation system are Eq. 3.53 and Eq. 3.54.

\[ y_{alt,k} = (1 + s_{alt}) (|\hat{\vec{r}}_k| - h_{\text{terrain},k} - R_{\text{moon}}) + b_{alt} + \nu_{alt,k} \quad (3.51) \]

\[ \hat{y}_{vel,k} = (I_{3 \times 3} + \text{diag}(\hat{s}_{vel}))(I_{3 \times 3} - \hat{\vec{v}}_{vel} \times)(\hat{\vec{v}}_k - \vec{\omega}_{\text{moon}} \times \hat{\vec{r}}_k) + \hat{b}_{vel} + \hat{\nu}_{vel,k} \quad (3.52) \]

\[ y_{range,k} = (1 + s_{range})(|\hat{\vec{r}}_k - \hat{\vec{r}}_{LS,k}|) + b_{range} + \nu_{range,k} \quad (3.53) \]

\[ y_{rrate,k} = (1 + s_{rrate}) \left( (\hat{\vec{v}}_k - \hat{\vec{v}}_{LS,k}) \cdot \left( \frac{\hat{\vec{r}}_k - \hat{\vec{r}}_{LS,k}}{|\hat{\vec{r}}_k - \hat{\vec{r}}_{LS,k}|} \right) \right) + b_{rrate} + \nu_{rrate,k} \quad (3.54) \]

The expressions from Eq. 3.51 through Eq. 3.54 are altered slightly in order to predict the measurements at each time step using the current estimate of the state, as seen in Eq. 3.55-3.58.

\[ \hat{y}_{alt,k} = (1 + \hat{s}_{alt}) (|\hat{\vec{r}}_k| - R_{\text{moon}}) + \hat{b}_{alt} \quad (3.55) \]

\[ \hat{y}_{vel,k} = (I_{3 \times 3} + \text{diag}(\hat{s}_{vel}))(I_{3 \times 3} - \hat{\vec{v}}_{vel} \times)(\hat{\vec{v}}_k - \vec{\omega}_{\text{moon}} \times \hat{\vec{r}}_k) + \hat{b}_{vel} \quad (3.56) \]

\[ \hat{y}_{range,k} = (1 + \hat{s}_{range})(|\hat{\vec{r}}_k - \hat{\vec{r}}_{LS,k}|) + \hat{b}_{range} \quad (3.57) \]

\[ \hat{y}_{rrate,k} = (1 + \hat{s}_{rrate}) \left( (\hat{\vec{v}}_k - \hat{\vec{v}}_{LS,k}) \cdot \left( \frac{\hat{\vec{r}}_k - \hat{\vec{r}}_{LS,k}}{|\hat{\vec{r}}_k - \hat{\vec{r}}_{LS,k}|} \right) \right) + \hat{b}_{rrate} \quad (3.58) \]
All of the measurements to be processed in a given time step can be combined into the the predicted and measured measurement vectors, \( \hat{y}_k \) and \( \tilde{y}_k \), as shown in Eq. 3.59 and Eq. 3.60.

\[
\hat{y}_k = \begin{bmatrix}
\hat{y}_{k,alt} \\
\hat{y}_{k,vel} \\
\hat{y}_{k,range} \\
\hat{y}_{k,rrate}
\end{bmatrix}
\] (3.59)

\[
\tilde{y}_k = \begin{bmatrix}
\tilde{y}_{k,alt} \\
\tilde{y}_{k,vel} \\
\tilde{y}_{k,range} \\
\tilde{y}_{k,rrate}
\end{bmatrix}
\] (3.60)

The partial derivatives of each of the measurements with respect to the navigation state are necessary in order to update the state when a measurement is available. These derivatives are evaluated at the current estimate of the state and combined into a matrix called \( H \). The \( H \) matrix for the reduced navigation state is given in Eq. 3.61 and for the full navigation state in Eq. 3.70. When the reduced navigation state is used, the current best estimate of all sensor errors used in the \( H \) matrix is zero. When the full navigation state is selected, the current estimate of the sensor errors is used to calculate \( H \), as shown.

\[
H_{\text{reduced}} = \left. \frac{\delta \tilde{h}}{\delta \tilde{x}_{\text{reduced}}} \right|_{\tilde{x}_{\text{reduced},k}} = \begin{bmatrix}
\frac{\delta h_{\text{alt}}}{\delta \tilde{r}} & 0_{1\times 3} \\
\frac{\delta h_{\text{vel}}}{\delta \tilde{r}} & \frac{\delta h_{\text{vel}}}{\delta \tilde{v}} \\
\frac{\delta h_{\text{range}}}{\delta \tilde{r}} & 0_{1\times 3} \\
\frac{\delta h_{\text{rate}}}{\delta \tilde{r}} & \frac{\delta h_{\text{rate}}}{\delta \tilde{\omega}} \\
\frac{\delta h_{\text{rate}}}{\delta \tilde{\omega}} & \frac{\delta h_{\text{rate}}}{\delta \tilde{\omega}}
\end{bmatrix}
\] (3.61)

\[
\frac{\delta h_{\text{alt}}}{\delta \tilde{r}} = (1 + \hat{\tilde{s}}_{\text{alt}}) \tilde{r}^T \\
\frac{\delta h_{\text{vel}}}{\delta \tilde{r}} = -\left( I_{3\times 3} + \text{diag}(\hat{\tilde{s}}_{\text{vel}}) \right) \left( I_{3\times 3} - \hat{\tilde{\omega}}_v \times \hat{\tilde{\omega}} \times \right)
\] (3.62)
\[
\delta h_{vel} = (I_{3x3} + \text{diag}(\hat{s}_{vel})) (I_{3x3} - \hat{v}_{vel} \times)
\]

\[
\frac{\delta h}{\delta \vec{v}} = (1 + s_{range}) \left( \frac{\hat{p} - \hat{r}_{LS}}{|\hat{p} - \hat{r}_{LS}|} \right) T
\]

\[
\frac{\delta h_{range}}{\delta \hat{r}_{LS}} = -(1 + s_{range}) \left( \frac{\hat{p} - \hat{r}_{LS}}{|\hat{p} - \hat{r}_{LS}|} \right) T
\]

\[
\frac{\delta h_{rate}}{\delta \vec{v}} = (1 + s_{rate}) \left( \frac{\hat{p} - \hat{r}_{LS}}{|\hat{p} - \hat{r}_{LS}|} \right)
\]

\[
\frac{\delta h_{rate}}{\delta \hat{r}_{LS}} = (1 + s_{rate}) \left[ \left( \frac{\hat{v} - \vec{\omega} \times \hat{r}_{LS}}{|\hat{r} - \hat{r}_{LS}|} \right) T \left( I_{3x3} - \left( \frac{\hat{r}_{LS}}{|\hat{r} - \hat{r}_{LS}|} \right) \left( \frac{\hat{r}_{LS}}{|\hat{r} - \hat{r}_{LS}|} \right)^T \right) \right]
\]

\[
H_{full} = \left. \frac{\delta h}{\delta \hat{x}_{full,k}} \right|_{\hat{x}_{full,k}} = \left[ \begin{array}{ccccccc}
0_{1x9} & \frac{\delta h_{alt}}{\delta E_{alt}} & 0_{1x9} & 0_{1x2} & 0_{1x2} \\
0_{3x9} & 0_{3x2} & \frac{\delta h_{vel}}{\delta E_{vel}} & 0_{3x2} & 0_{3x2} \\
0_{1x9} & 0_{1x2} & 0_{1x2} & 0_{1x2} & 0_{1x2} \\
0_{1x9} & 0_{1x2} & 0_{1x2} & 0_{1x2} & \frac{\delta h_{range}}{\delta E_{range}} & 0_{1x2} \\
\end{array} \right]
\]

\[
\frac{\delta h_{alt}}{\delta E_{alt}} = \left[ \begin{array}{cc}
\frac{\delta h_{alt}}{\delta s_{alt}}, & \frac{\delta h_{alt}}{\delta s_{alt}}
\end{array} \right] = \left[ \begin{array}{c}
1, \ (|\vec{r}| - R_{moon})
\end{array} \right]
\]
\[
\frac{\delta \hat{h}_{vel}}{\delta E_{vel}} = \begin{bmatrix}
\delta h_{vel} \\
\delta b_{vel} \\
\delta s_{vel} \\
\delta \varepsilon_{vel}
\end{bmatrix} = [ I_{3x3}, \; \text{diag}((I_{3x3} - \hat{\hat{\varepsilon}}_{vel})(\hat{\hat{\omega}} - \hat{\hat{\omega}} \times \hat{\hat{r}})), \; ...]
\]

\[
\frac{\delta h_{range}}{\delta E_{range}} = \begin{bmatrix}
\frac{\delta h_{range}}{\delta \hat{h}_{range}} \\
\frac{\delta h_{range}}{\delta \hat{b}_{range}} \\
\frac{\delta h_{range}}{\delta \hat{s}_{range}} \\
\frac{\delta h_{range}}{\delta \hat{\varepsilon}_{range}}
\end{bmatrix} = [ 1, \; |\hat{r} - \hat{r}_{LS}| ]
\]

\[
\frac{\delta h_{rate}}{\delta E_{rate}} = \begin{bmatrix}
\frac{\delta h_{rate}}{\delta \hat{h}_{rate}} \\
\frac{\delta h_{rate}}{\delta \hat{b}_{rate}} \\
\frac{\delta h_{rate}}{\delta \hat{s}_{rate}} \\
\frac{\delta h_{rate}}{\delta \hat{\varepsilon}_{rate}}
\end{bmatrix} = [ 1, \; (\hat{\hat{\omega}} - \hat{\hat{\omega}} \times \hat{\hat{r}}_{LS}) \cdot \frac{(|\hat{r} - \hat{r}_{LS}|^2)}{|\hat{r} - \hat{r}_{LS}|} ]
\]

### 3.5.5 Navigation State Update

Once the \( H \) matrix, the actual measurement, \( \tilde{y}_k \), and the predicted measurement, \( \hat{y}_k \) are available, it is possible to update the current estimate of the navigation state using Eq. 3.76. The state estimate before the update is \( \hat{x}_k^- \) and the state estimate after the update is \( \hat{x}_k^+ \). The matrix \( K \) is known as the Kalman gain and is calculated as shown in Eq. 3.77. The matrix \( R \) is the covariance of the white noise on the measurement vector.

\[
\hat{x}_n^+ = \hat{x}_n^- + K_k [\hat{y}_k - \hat{y}_k]
\]

\[
K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}
\]

The covariance matrix is also updated every time there is a measurement available. It is updated using Eq. 3.78. The covariance matrix before the update is \( P_k^- \) and the covariance after the update is \( P_k^+ \).

\[
P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T
\]
Chapter 4
Simulation Models

To examine the fuel dispersion for each navigation state selection, a Monte Carlo simulation was developed. Given guidance targets, initial orbital parameters (before the deorbit burn), a desired landing site, the number of runs to perform, and the appropriate dynamics models, the simulation generates a nominal trajectory, as well as 1-sigma error bounds on that trajectory. This nominal trajectory is equivalent to a data run with all errors turned off or the average trajectory when a large number of data runs are performed. It also generates the average and standard deviation of the navigation errors and of the amount of fuel used.

The Monte Carlo Simulation includes a guidance algorithm based on the methods developed by Sostaric [6]. It also includes an Extended Kalman filter to estimate the vehicle state throughout the simulation. Simulated measurements to be processed in the navigation subroutine are generated using a “truth” model of the vehicle dynamics as well as an analytic model for each sensor. In addition, the simulation also uses a proportional-derivative controller to perform attitude control. An overview of the simulation can be seen in Fig. 4.1.

4.1 Simulation Inputs

The necessary inputs for the simulation are summarized in Table 4.1.
Fig. 4.1: Powered lunar descent Monte Carlo simulation overview.
Table 4.1: Simulation inputs.

<table>
<thead>
<tr>
<th>Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial orbit</td>
</tr>
<tr>
<td>Landing site position</td>
</tr>
<tr>
<td>Initial covariances on navigation states</td>
</tr>
<tr>
<td>Vehicle mass properties</td>
</tr>
<tr>
<td>Desired number of runs</td>
</tr>
<tr>
<td>Desired time step</td>
</tr>
<tr>
<td>Sensor specifications</td>
</tr>
<tr>
<td>Actuator specifications</td>
</tr>
<tr>
<td>Guidance targets</td>
</tr>
<tr>
<td>Moon gravity model</td>
</tr>
<tr>
<td>Lunar Constants</td>
</tr>
</tbody>
</table>

A summary of the relevant mass properties for the lander is given in Table 4.2.

Table 4.2: Vehicle mass properties.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Mass</td>
<td>33240 kg</td>
</tr>
<tr>
<td>Dry Mass</td>
<td>13454 kg</td>
</tr>
<tr>
<td>$I_{xx,initial}$</td>
<td>255891 kg $\cdot m^2$</td>
</tr>
<tr>
<td>$I_{yy,initial}$</td>
<td>289049 kg $\cdot m^2$</td>
</tr>
<tr>
<td>$I_{zz,initial}$</td>
<td>306849 kg $\cdot m^2$</td>
</tr>
<tr>
<td>$I_{xx,final}$</td>
<td>131850 kg $\cdot m^2$</td>
</tr>
<tr>
<td>$I_{yy,final}$</td>
<td>182867 kg $\cdot m^2$</td>
</tr>
<tr>
<td>$I_{zz,final}$</td>
<td>200679 kg $\cdot m^2$</td>
</tr>
</tbody>
</table>

The nominal initial orbit selected for the simulation is a 100 km nearly circular, polar orbit. The orbit was selected so that the vehicle would be over the lunar north pole at an
altitude of 100 km at the beginning of the simulation. The initial orbital elements are given in Table 4.3.

<table>
<thead>
<tr>
<th>Orbital Element</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-major Axis ($a$)</td>
<td>$1837.4 \times 10^3 m$</td>
</tr>
<tr>
<td>Eccentricity ($e$)</td>
<td>0</td>
</tr>
<tr>
<td>Inclination ($i$)</td>
<td>$90^\circ$</td>
</tr>
<tr>
<td>Right Ascension of the Ascending Node ($\Omega$)</td>
<td>0</td>
</tr>
<tr>
<td>Argument of Perilune ($\omega$)</td>
<td>0</td>
</tr>
<tr>
<td>True Anomaly ($\nu$)</td>
<td>$90^\circ$</td>
</tr>
</tbody>
</table>

The landing site selected for this mission is the Shackleton crater, a crater near the South pole of the moon. It was important to select a landing site that was not exactly at the South pole in order to avoid singularities in the dynamics equations. The altitude, latitude and longitude of the Shackleton crater are given in Table 4.4.

<table>
<thead>
<tr>
<th>Position of landing site</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latitude</td>
</tr>
<tr>
<td>Longitude</td>
</tr>
<tr>
<td>Altitude</td>
</tr>
</tbody>
</table>

The initial covariances on all navigation states vary based on whether the best, worst, or typical scenario is selected. The covariances on each of the individual navigation states was set to the same value as the random noise on that state. The values of the random noise for each state were chosen to match the values in [7]. Table 4.5 and Table 4.6 give the values for each scenario.
Table 4.5: Initial variances of navigation states, reduced navigation state.

<table>
<thead>
<tr>
<th></th>
<th>State</th>
<th>Best</th>
<th>Typical</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vehicle Position</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Downrange</td>
<td></td>
<td>(15 m)$^2$</td>
<td>(150 m)$^2$</td>
<td>(750 m)$^2$</td>
</tr>
<tr>
<td>Cross-track</td>
<td></td>
<td>(2 m)$^2$</td>
<td>(20 m)$^2$</td>
<td>(100 m)$^2$</td>
</tr>
<tr>
<td>Altitude</td>
<td></td>
<td>(0.5 m)$^2$</td>
<td>(5 m)$^2$</td>
<td>(25 m)$^2$</td>
</tr>
<tr>
<td><strong>Vehicle Velocity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Downrange</td>
<td></td>
<td>(0.00047 m/s)$^2$</td>
<td>(0.0047 m/s)$^2$</td>
<td>(0.0235 m/s)$^2$</td>
</tr>
<tr>
<td>Cross-Track</td>
<td></td>
<td>(0.002 m/s)$^2$</td>
<td>(0.02 m/s)$^2$</td>
<td>(0.1 m/s)$^2$</td>
</tr>
<tr>
<td>Altitude</td>
<td></td>
<td>(0.015 m/s)$^2$</td>
<td>(0.15 m/s)$^2$</td>
<td>(0.75 m/s)$^2$</td>
</tr>
<tr>
<td><strong>Landing Site</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position</td>
<td></td>
<td>(1 m)$^2$</td>
<td>(10 m)$^2$</td>
<td>(50 m)$^2$</td>
</tr>
</tbody>
</table>
Table 4.6: Initial variances of navigation states, full navigation state.

<table>
<thead>
<tr>
<th>State</th>
<th>Best</th>
<th>Typical</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accelerometer</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias (per axis)</td>
<td>((0.3 \mu g)^2)</td>
<td>((3 \mu g)^2)</td>
<td>((15 \mu g)^2)</td>
</tr>
<tr>
<td>Scale Factor (per axis)</td>
<td>((0.66 \times 10^{-6})^2)</td>
<td>((6.6 \times 10^{-6})^2)</td>
<td>((33 \times 10^{-6})^2)</td>
</tr>
<tr>
<td>Misalignment (per axis)</td>
<td>((0.2 arcsec)^2)</td>
<td>((2 arcsec)^2)</td>
<td>((10 arcsec)^2)</td>
</tr>
<tr>
<td><strong>Altimeter</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>((0.005 m)^2)</td>
<td>((0.05 m)^2)</td>
<td>((0.25 m)^2)</td>
</tr>
<tr>
<td>Scale Factor</td>
<td>((1 \times 10^{-5})^2)</td>
<td>((1 \times 10^{-4})^2)</td>
<td>((5 \times 10^{-4})^2)</td>
</tr>
<tr>
<td><strong>Velocimeter</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias (per axis)</td>
<td>((0.001 m/s)^2)</td>
<td>((0.01 m/s)^2)</td>
<td>((0.05 m/s)^2)</td>
</tr>
<tr>
<td>Scale Factor (per axis)</td>
<td>((1 \times 10^{-5})^2)</td>
<td>((1 \times 10^{-4})^2)</td>
<td>((5 \times 10^{-4})^2)</td>
</tr>
<tr>
<td>Misalignment (per axis)</td>
<td>((0.5 arcsec)^2)</td>
<td>((5 arcsec)^2)</td>
<td>((25 arcsec)^2)</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>((1.05 m)^2)</td>
<td>((1.05 m)^2)</td>
<td>((5.25 m)^2)</td>
</tr>
<tr>
<td>Scale Factor</td>
<td>((1 \times 10^{-5})^2)</td>
<td>((1 \times 10^{-4})^2)</td>
<td>((5 \times 10^{-4})^2)</td>
</tr>
<tr>
<td><strong>Range Rate</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>((0.005 mm/s)^2)</td>
<td>((0.05 mm/s)^2)</td>
<td>((0.25 mm/s)^2)</td>
</tr>
<tr>
<td>Scale Factor</td>
<td>((1 \times 10^{-7})^2)</td>
<td>((1 \times 10^{-6})^2)</td>
<td>((5 \times 10^{-6})^2)</td>
</tr>
</tbody>
</table>

For a complete Monte Carlo simulation, 300 sample runs are performed. This number was selected in order to get enough data that the average trajectory generated by the simulation would be close to the nominal, error free, trajectory.

The simulation is run with a time step of 0.1 seconds. The guidance, navigation, and attitude control algorithms are run once every time step during a given data run, and the truth state, used to generate simulated measurements, is also updated once every time
step. Sensor measurements are generated every 1 second, except for measurements from the gyroscopes, accelerometers, and star camera, which are generated once every time step.

The specifications for the thrusters and the reaction control system are given in Table 4.7.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>1-sigma error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
</tr>
<tr>
<td>Thrusters</td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>0.0033 mm/s²</td>
</tr>
<tr>
<td>Bias</td>
<td>0.0033 mm/s²</td>
</tr>
<tr>
<td>Scale Factor</td>
<td>3.33 ppm</td>
</tr>
<tr>
<td>Misalignment</td>
<td>0.68 arcsec</td>
</tr>
</tbody>
</table>

Maximum Thrust: 18627 lbf
Minimum Thrust: 1862.7 lbf
Thruster Specific Impulse: 446.85 s

<table>
<thead>
<tr>
<th>Reaction Control System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise</td>
</tr>
<tr>
<td>Bias</td>
</tr>
<tr>
<td>Misalignment</td>
</tr>
</tbody>
</table>

Maximum Thrust: 100 lbf

The guidance targets used in the simulation are given in section 2.3.5. The lunar gravity model selected is a point mass model with a J2 perturbation, and is described in further detail in section 4.3. Lunar constants are given in Table 4.8.
Table 4.8: Lunar constants.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gravitational Parameter ($\mu$)</td>
<td>$4902.8 \times 10^9 m^3/s^2$</td>
</tr>
<tr>
<td>J2 Gravitational Coefficient ($J_2$)</td>
<td>$2.033542 \times 10^{-4}$</td>
</tr>
<tr>
<td>Radius ($R_{moon}$)</td>
<td>$1737.4 \times 10^3 m$</td>
</tr>
<tr>
<td>Angular Velocity Vector ($\vec{\omega}_{moon}$)</td>
<td>$\begin{bmatrix} -9.432 \times 10^{-11} \ -1.0597 \times 10^{-6} \ 2.4417 \times 10^{-6} \end{bmatrix} rad/s$</td>
</tr>
</tbody>
</table>

The specifications for each of the sensors used in the simulation, including standard deviations for noise, bias, misalignment, and scale factor errors, are given in Table 4.9.

4.2 Simulation Outputs and Metrics

The simulation outputs and metrics can be seen in Table 4.10. In order to investigate the thesis of this research, it was necessary to perform Monte Carlo runs for a number of different scenarios. The metrics used to compare the performance of different scenarios were the average fuel used, the fuel use standard deviation, and the 1-sigma errors on the position and velocity.

Table 4.10: Simulation outputs and metrics.

<table>
<thead>
<tr>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time histories of average position, velocity, and acceleration (in LVLH and inertial)</td>
</tr>
<tr>
<td>1-sigma error bounds on position and velocity (in LVLH and inertial)*</td>
</tr>
<tr>
<td>Fuel used (average and standard deviation)*</td>
</tr>
<tr>
<td>1-sigma error bounds on all other navigation states</td>
</tr>
<tr>
<td>Total descent time (average and standard deviation)</td>
</tr>
</tbody>
</table>

*Indicates a metric

At each time step in the simulation, the current values of the true position, velocity,
Table 4.9: Sensor specifications.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Best 1-σ</th>
<th>Typical 1-σ</th>
<th>Worst 1-σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>Noise: $1.7 \times 10^{-6}$ m/s/$\sqrt{s}$</td>
<td>$1.7 \times 10^{-5}$ m/s/$\sqrt{s}$</td>
<td>$8.5 \times 10^{-5}$ m/s/$\sqrt{s}$</td>
</tr>
<tr>
<td></td>
<td>Bias: $0.3 \mu g$</td>
<td>$3 \mu g$</td>
<td>$15 \mu g$</td>
</tr>
<tr>
<td></td>
<td>Scale Factor: 0.66 ppm</td>
<td>6.6 ppm</td>
<td>33 ppm</td>
</tr>
<tr>
<td></td>
<td>Misalignment: 0.2 arcsec</td>
<td>2 arcsec</td>
<td>10 arcsec</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>Noise: $5 \times 10^{-7}$ deg/$\sqrt{s}$</td>
<td>$5 \times 10^{-6}$ deg/$\sqrt{s}$</td>
<td>$2.5 \times 10^{-5}$ deg/$\sqrt{s}$</td>
</tr>
<tr>
<td></td>
<td>Bias: 0.0002 deg/hr</td>
<td>0.002 deg/hr</td>
<td>0.01 deg/hr</td>
</tr>
<tr>
<td></td>
<td>Scale Factor: 0.016 ppm</td>
<td>0.16 ppm</td>
<td>0.8 ppm</td>
</tr>
<tr>
<td></td>
<td>Misalignment: 0.2 arcsec</td>
<td>2 arcsec</td>
<td>10 arcsec</td>
</tr>
<tr>
<td>Star Tracker</td>
<td>Noise: 0.5 arcsec</td>
<td>5 arcsec</td>
<td>25 arcsec</td>
</tr>
<tr>
<td>Altimeter</td>
<td>Noise (&gt; 2 km): 0.05 m</td>
<td>0.5 m</td>
<td>2.5 m</td>
</tr>
<tr>
<td></td>
<td>Noise (&lt; 2 km): 0.01 m</td>
<td>0.1 m</td>
<td>0.5 m</td>
</tr>
<tr>
<td></td>
<td>Bias: 0.005 m</td>
<td>0.05 m</td>
<td>0.25 m</td>
</tr>
<tr>
<td></td>
<td>Scale Factor: 0.001%</td>
<td>0.01%</td>
<td>0.05%</td>
</tr>
<tr>
<td></td>
<td>Terrain height ($h_{terrain}$) noise: 33 m, Correlation Distance ($d_{corr}$): 10 m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Velocimeter</td>
<td>Noise: 0.005 m/s</td>
<td>0.05 m/s</td>
<td>0.25 m/s</td>
</tr>
<tr>
<td></td>
<td>Bias: 0.001 m/s</td>
<td>0.01 m/s</td>
<td>0.05 m/s</td>
</tr>
<tr>
<td></td>
<td>Scale Factor: 0.001%</td>
<td>0.01%</td>
<td>0.05%</td>
</tr>
<tr>
<td></td>
<td>Misalignment: 0.5 arcsec</td>
<td>5 arcsec</td>
<td>25 arcsec</td>
</tr>
<tr>
<td>Range</td>
<td>Noise: 1.05 m</td>
<td>1.05 m</td>
<td>5.25 m</td>
</tr>
<tr>
<td></td>
<td>Bias: 1.05 m</td>
<td>1.05 m</td>
<td>5.25 m</td>
</tr>
<tr>
<td></td>
<td>Scale Factor: $1 \times 10^{-5}$ %</td>
<td>$1 \times 10^{-4}$ %</td>
<td>$5 \times 10^{-4}$ %</td>
</tr>
<tr>
<td>Range Rate</td>
<td>Noise: 0.005 mm/s</td>
<td>0.05 mm/s</td>
<td>0.25 mm/s</td>
</tr>
<tr>
<td></td>
<td>Bias: 0.005 mm/s</td>
<td>0.05 mm/s</td>
<td>0.25 mm/s</td>
</tr>
<tr>
<td></td>
<td>Scale Factor: $1 \times 10^{-7}$ %</td>
<td>$1 \times 10^{-6}$ %</td>
<td>$5 \times 10^{-6}$ %</td>
</tr>
</tbody>
</table>
and acceleration, navigation estimate of position, velocity, and landing site position, and the covariance matrix for the navigation states are saved. In addition, when the full navigation state is used, the estimates of all of the additional navigation states are saved at each time step. At the end of each sample run, the true landing site position, true and estimated descent time, true and estimated delta-v, and the true values of all sensor errors (if the full navigation state is used) are saved.

In order to calculate the metrics used to compare different cases, it is necessary to calculate the mean and standard deviation over all runs for several quantities of interest. The equation for calculating the sample mean [19], $x_{avg}$, of a set of $n$ scalar samples of $x$ is Eq. 4.1. The equation for calculating the sample standard deviation [19], $s_x$, of a set of $n$ samples of $x$ is Eq. 4.2.

$$x_{mean} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - x_{mean})^2}$$

The average value of a quantity of interest is calculated by taking the average of the true value of that quantity over all sample runs, as shown in Eq. 4.3. The nominal value of a quantity, $\bar{x}_{nom}$, is defined as the value of that quantity from a sample run with no errors.

$$x_{avg} = \frac{1}{n} \sum_{i=1}^{n} x_{true,i}$$

For a vector quantity, Eq. 4.3 becomes Eq. 4.4. The covariance of a vector is given in Eq. 4.5.

$$\bar{x}_{mean} = \frac{1}{n} \sum_{i=1}^{n} \bar{x}_{true,i}$$

$$S_x = \frac{1}{n-1} \sum_{i=1}^{n} [\bar{x}_i - \bar{x}_{mean}] (\bar{x}_i - \bar{x}_{mean})^T$$
There are three different types of errors used as metrics that must be calculated in the simulation. The first is the true trajectory dispersion, \( E_{x,\text{true}} \), which is the covariance over all sample runs between the true value of a variable, \( \bar{x}_{\text{true}} \), and the average value of that variable, \( \bar{x}_{\text{avg}} \), as shown in Eq. 4.6.

\[
E_{x,\text{true}} = \frac{1}{n-1} \sum_{i=1}^{n} [(\bar{x}_{\text{true},i} - \bar{x}_{\text{avg},i})(\bar{x}_{\text{true},i} - \bar{x}_{\text{avg},i})^T] \tag{4.6}
\]

The second type of error calculated in the simulation is the true navigation error, which is the standard deviation of the difference between the navigation estimate, \( x_{\text{nav}} \), of the variable of interest and the true value of that variable, \( x_{\text{true}} \), over all sample runs. The equation for calculating the true navigation error is Eq. 4.7.

\[
E_{x,\text{nav}} = \frac{1}{n-1} \sum_{i=1}^{n} [(\bar{x}_{\text{nav},i} - \bar{x}_{\text{true},i})(\bar{x}_{\text{nav},i} - \bar{x}_{\text{true},i})^T] \tag{4.7}
\]

The final type of error is the filter estimate of the navigation error, which is the square root of the variance, \( \sigma_{x,\text{nav}}^2 \), of the variable of interest, \( x_{\text{nav}} \), as calculated in the navigation filter. The variance of the variable of interest is simply the diagonal term which corresponds to the variable of interest in the navigation state covariance matrix. The navigation filter error is shown in Eq. 4.8.

\[
E_{x,\text{filter}} = \sqrt{P_{xx,\text{nav}}} = \sigma_{x,\text{nav}} \tag{4.8}
\]

In order to calculate the errors in the LVLH frame, the following procedures were used. For each time step, the average of the position and velocity over all data runs was calculated. This position and velocity was then used to create a transformation matrix between the inertial and LVLH frames. Then, this transformation matrix, \( T_{I\rightarrow LVLH} \), was used to calculate the metrics of interest, as shown in Eq. 4.9 through Eq. 4.12.

\[
\bar{x}_{\text{avg},LVLH} = T_{I\rightarrow LVLH} \bar{x}_{\text{avg}} \tag{4.9}
\]
\[ E_{x,\text{true},LV LH} = T_{I \rightarrow LV LH} E_{x,\text{true}} T_{I \rightarrow LV LH}^{T} \]  \hspace{1cm} (4.10)

\[ E_{x,\text{nav},LV LH} = T_{I \rightarrow LV LH} E_{x,\text{nav}} T_{I \rightarrow LV LH}^{T} \]  \hspace{1cm} (4.11)

\[ E_{x,\text{filter},LV LH} = T_{I \rightarrow LV LH} E_{x,\text{filter}} T_{I \rightarrow LV LH}^{T} \]  \hspace{1cm} (4.12)

In order to calculate the average and the standard deviation of the delta-v used by the vehicle during landing, the following equations are used.

\[ \Delta v_{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} \Delta v_{i,\text{true}} \]  \hspace{1cm} (4.13)

\[ e \Delta v_{\text{true}} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (\Delta v_{\text{true},i} - \Delta v_{\text{avg},i})^2} \]  \hspace{1cm} (4.14)

\[ e \Delta v_{\text{nav}} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (\Delta v_{\text{nav},i} - \Delta v_{\text{true},i})^2} \]  \hspace{1cm} (4.15)
4.3 Dynamics

In order to describe the translational motion of the lunar lander, it will be necessary to use Newton’s second law, which states that the summation of all the forces acting on a body is equal to the change of momentum of the body in an inertial frame. This is Eq. 4.16. For cases where the mass of the body is not changing, Eq. 4.16 can be written as Eq. 4.17.

\[ \Sigma \vec{F}^I = \frac{d}{dt}(m\vec{v}) \]  

(4.16)

\[ \Sigma \vec{F}^I = \vec{m\ddot{a}}^I \]  

(4.17)

For powered lunar descent, an appropriate model for the forces acting on the vehicle is shown in Eq. 4.25. The terms \( \vec{F}_{grav} \) and \( \vec{F}_{thr} \) are the force of gravity and the force generated by the lander’s thrusters, respectively. Any forces not included in the models for \( \vec{F}_{grav} \) and \( \vec{F}_{thr} \) are combined into a single term, \( \vec{F}_{misc} \). This leads to the model for the vehicle dynamics shown in Eq. 4.19 and Eq. 4.20. The term \( \vec{w}_a \) is white process noise which is used to account for any variations in the acceleration of the vehicle which are not included in the gravitational model or the model of the thrusters. In the simulation, \( \vec{w}_a \) has strength \( Q_a = (1 \times 10^{-8}) \times I_{3x3} \).

\[ \Sigma \vec{F}^I = \vec{F}_{grav} + \vec{F}_{thr} + \vec{F}_{misc} \]  

(4.18)

\[ \dot{\vec{r}} = \vec{v} \]  

(4.19)

\[ \dot{\vec{v}} = \vec{a}_{grav} + \vec{a}_{thr} + \vec{w}_a \]  

(4.20)

A simple model for the Moon’s gravitational acceleration was selected for this simulation. It is simply a point mass gravity model with a J2 perturbation. This is shown in Eq. 4.21. The gravitational parameter, \( \mu \), for the moon is 4902.801 km\(^3\)/s\(^2\) and the J2 coefficient
is $2.033542 \times 10^{-4}$ [20].

$$\bar{a}_{\text{grav}} = \frac{\mu}{|\vec{r}|^3} \vec{r} + \bar{a}_{J_2}$$

(4.21)

The acceleration due to the oblateness of the moon, $\bar{a}_{J_2}$, is calculated using Eq. 4.22. This equation is dependent on the equatorial radius of the moon and also includes a vector, $\bar{n}$, which is simply equal to $[0 \ 0 \ 1]^T$.

$$\bar{a}_{J_2} = -\frac{\mu J_2 R_{\text{eq}}}{2 |\vec{r}|^5} \left\{ 6 \left( \vec{r} \cdot \bar{n} \right) \bar{n} + 3 \vec{r} - 15 \left( \vec{r} \cdot \bar{n} \right)^2 \vec{r} \right\}$$

(4.22)

To describe the rotational motion of the lander, Euler’s equation will be used. Euler’s equation is found by taking the derivative of the angular momentum equation [7], and is given in Eq. 4.23. Euler’s equation can be rearranged to solve for the angular accelerations of the vehicle, as shown in Eq. 4.24. The inertia tensor of the vehicle is $I^B$, the vehicle’s angular velocity is $\dot{\omega}^B$, and $M^B$ are external torques on the vehicle.

$$\Sigma \bar{M}^B = \frac{d}{dt} \left( I^B \dot{\omega}^B \right) = I^B \ddot{\omega}^B + I^B \dot{\omega}^B \times I^B \dot{\omega}^B$$

(4.23)

$$\dot{\omega}^B = \left( I^B \right)^{-1} \left( \Sigma \bar{M}^B - I^B \dot{\omega}^B - \omega^B \times I^B \dot{\omega}^B \right)$$

(4.24)

The moments acting on the lander can be described using Eq. 4.25. The only major moments acting on the vehicle in the simulation are those which come from the reaction control system, $\bar{M}_{\text{RCS}}$, but small moments not included in the model of the reaction control system may exist, and they are modeled as white noise, $\bar{w}_M$.

$$\Sigma \bar{M}^B = \bar{M}_{\text{RCS}} + \bar{w}_M$$

(4.25)

At the beginning of each time step during a sample run, the vehicle attitude and angular velocity is updated using the torque, $\bar{M}_{\text{RCS}}$, generated by the RCS in the previous step, the previous vehicle attitude, $\bar{q}$, and the previous angular velocity, $\bar{\omega}$. The vehicle attitude is
updated using Eq. 4.26 through 4.30.

\[
d\theta = \dot{\theta} \cdot dt_{\text{sim}} \tag{4.26}
\]

\[
d\theta = |\dot{\theta}| \tag{4.27}
\]

\[
\dot{u} = \frac{\ddot{d}\theta}{d\theta} \tag{4.28}
\]

\[
dq = [-\dot{u}^T \cdot \sin(d\theta/2) \cos(d\theta/2)]^T \tag{4.29}
\]

\[
\bar{q}_1 = dq \otimes \bar{q}_0 \tag{4.30}
\]

The vehicle angular velocity is then updated using Euler’s equation, as shown in Eq. 4.31 and Eq. 4.32.

\[
\dot{\omega} = I^{-1} \left( \ddot{T} - \bar{\omega}_0 \times I \cdot \ddot{\omega}_0 \right) \tag{4.31}
\]

\[
\bar{\omega}_1 = \omega_0 + \dot{\omega} \cdot dt_{\text{sim}} \tag{4.32}
\]

4.4 Actuators

It would be impossible to generate thrust or torque at exactly the levels commanded by the guidance and attitude control algorithms in a real world application. Therefore, it was necessary to include appropriate error models for the actuators used in the Monte Carlo simulation.

Thrusters are used to generate the commanded accelerations for translational motion. The error model for the thrusters includes a random noise term, thruster bias, misalignment errors, and scale factor errors. This model can be seen in Eq. 4.33. In the following models,
$[\vec{\varepsilon} \times]$ is the cross product matrix formed from a vector, $\vec{\varepsilon}$, of small misalignment errors, $\vec{s}$ is a scale factor, $\vec{b}$ is a bias, and $\vec{n}$ is random Gaussian noise.

$$\vec{a}_{thr} = (I_{3x3} + \text{diag}(\vec{s}_{thrust})) \cdot (I_{3x3} - [\vec{\varepsilon}_{thrust} \times]) \cdot \vec{a}_{com} + \vec{b}_{thrust} + \vec{n}_{thrust} \tag{4.33}$$

The actuators used in the attitude control system are assumed to be thrusters in a reaction control system (RCS). The RCS generates a torque based on the commanded torque. The error model for the RCS considers random noise, bias, misalignment error, and scale factor error. This model can be seen in Eq. 4.34.

$$\vec{a}_{thr} = (I_{3x3} + \text{diag}(\vec{s}_{torque})) \cdot (I_{3x3} - [\vec{\varepsilon}_{torque} \times]) \cdot \vec{a}_{com} + \vec{b}_{torque} + \vec{n}_{torque} \tag{4.34}$$

4.5 Sensors

The sensors used in the sensor suite for this Monte Carlo simulation are an inertial measurement unit (IMU) that includes accelerometers and gyroscopes, a star camera, an altimeter, a velocimeter, and a radio navigation system. The radio navigation system is used to obtain range and range-rate measurements from a beacon at the desired landing site. A summary of the sensors used in the simulation can be found in Table 4.11, and the errors incorporated into their models can be found in Table 4.12.

In order to create simulated measurements, models for each measurement were developed. The models use the truth state in order to generate the appropriate type of measurement for each sensor. In each model, $\vec{b}$ is a bias (or $\tilde{\vec{b}}$ for vector bias), and $\eta$ is white noise (or $\tilde{\eta}$ for vector noise). Other errors accounted for in these sensor models are scale factor errors ($s$ or $\vec{s}$) and misalignment errors ($\vec{\varepsilon}$).

The accelerometer measures the body-fixed accelerations of the vehicle. The primary accelerations experienced by the vehicle are thrust and gravitational forces. The model for the accelerometer measurement does not include gravitational accelerations because ac-
Table 4.11: Sensors included in simulation.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Time Step</th>
<th>Condition for Use</th>
<th>Measurement</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometers</td>
<td>$\Delta t_{\text{sim}} = 0.1 , s$</td>
<td>Used at all times</td>
<td>$\tilde{\mathbf{a}}_v^B$</td>
<td>$3 \times 1$</td>
</tr>
<tr>
<td>Gyroscopes</td>
<td>$\Delta t_{\text{sim}} = 0.1 , s$</td>
<td>Used at all times</td>
<td>$\tilde{\mathbf{\omega}}_v^B$</td>
<td>$3 \times 1$</td>
</tr>
<tr>
<td>Star Camera</td>
<td>$\Delta t_{\text{sim}} = 0.1 , s$</td>
<td>Used at all times</td>
<td>$\tilde{\mathbf{q}}_{\text{ST}}^B$</td>
<td>$4 \times 1$</td>
</tr>
<tr>
<td>Altimeter</td>
<td>$\Delta t_{\text{sensors}} = 1 , s$</td>
<td>$h &lt; 20 , \text{km}$</td>
<td>$h$</td>
<td>Scalar</td>
</tr>
<tr>
<td>Velocimeter</td>
<td>$\Delta t_{\text{sensors}} = 1 , s$</td>
<td>$h &lt; 2 , \text{km}$</td>
<td>$\bar{v}_{\text{surface}}$</td>
<td>$3 \times 1$</td>
</tr>
<tr>
<td>Radio</td>
<td>$\Delta t_{\text{sensors}} = 1 , s$</td>
<td>$</td>
<td>\bar{r}_v^I - \bar{r}_v^L</td>
<td>&lt; 100 , \text{km}$</td>
</tr>
<tr>
<td>Navigation</td>
<td>$\Delta t_{\text{sensors}} = 1 , s$</td>
<td>$</td>
<td>\bar{r}_v^I - \bar{r}_v^L</td>
<td>&lt; 100 , \text{km}$</td>
</tr>
</tbody>
</table>

Accelerometers cannot detect accelerations due to gravity. The model for the accelerometer is given in Eq. 4.35. The transformation matrix between the inertial frame and the body-fixed frame is necessary because accelerometer measurements are taken in the body frame, and is represented by $T^B_I$. The acceleration produced by the thrusters is $a^I_{\text{thrust}}$.

$$
\mathbf{\tilde{a}}^B = (I_{3 \times 3} + \text{diag}(\mathbf{s}_{\text{accel}}))(I_{3 \times 3} - \mathbf{\varepsilon}_{\text{accel}} \times)T^B_I(a^I_{\text{thrust}}) + \mathbf{\bar{b}}_{\text{accel}} + \mathbf{\eta}_{\text{accel}} \tag{4.35}
$$

The gyroscope is used to determine the body-fixed angular velocity of the lander, and its model is similar to the model for the accelerometer. The model for the gyroscope measurement is Eq. 4.36. The angular velocity generated by the attitude control system is $\mathbf{\omega}^I_{\text{RCS}}$.

$$
\mathbf{\tilde{\omega}}^B = (I_{3 \times 3} + \text{diag}(\mathbf{s}_{\text{gyro}}))(I_{3 \times 3} - \mathbf{\varepsilon}_{\text{gyro}} \times)(\mathbf{\omega}^B_{\text{RCS}}) + \mathbf{\bar{b}}_{\text{gyro}} + \mathbf{\eta}_{\text{gyro}} \tag{4.36}
$$

The star camera is also used to determine the attitude of the vehicle. It gives a measurement in the form of a quaternion, so the misalignment error and noise must be incorporated as small rotations, represented as $\delta \mathbf{q}$. The model for the star camera measurement is shown in Eq. 4.37. The terms $\mathbf{q}^B_{\text{ST}}$ and $\mathbf{q}^B_I$ are quaternions representing the transformation between the body-fixed frame and the star camera frame and the transformation between the
Table 4.12: Typical errors included in sensor models (1-sigma values).

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Noise</th>
<th>Bias</th>
<th>Scale Factor</th>
<th>Misalignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometers</td>
<td>$1.7 \times 10^{-5}$ m/s/(\sqrt{s}/\text{axis})</td>
<td>$3 \mu g/\text{axis}$</td>
<td>6.6 ppm/\text{axis}</td>
<td>2 arcsec/\text{axis}</td>
</tr>
<tr>
<td>Gyroscopes</td>
<td>$5 \times 10^{-6}$ deg/(\sqrt{s}/\text{axis})</td>
<td>0.002 deg/hr/\text{axis}</td>
<td>0.16 ppm/\text{axis}</td>
<td>2 arcsec/\text{axis}</td>
</tr>
<tr>
<td>Star Camera</td>
<td>5 arcsec/\text{axis}</td>
<td>None</td>
<td>None</td>
<td>2 arcsec/\text{axis}</td>
</tr>
<tr>
<td>Altimeter</td>
<td>0.5 m (altitude &gt; 2 km), 0.1 m (altitude &lt; 2 km)</td>
<td>0.05 m</td>
<td>0.01 %</td>
<td>None</td>
</tr>
<tr>
<td>Velocimeter</td>
<td>0.05 m/s/\text{axis}</td>
<td>0.01 m/s/\text{axis}</td>
<td>0.01 %/\text{axis}</td>
<td>5 arcsec</td>
</tr>
<tr>
<td>Radio Navigation - Range</td>
<td>1.05 m</td>
<td>1.05 m</td>
<td>$1 \times 10^{-4}$%</td>
<td>None</td>
</tr>
<tr>
<td>Radio Navigation - Range Rate</td>
<td>0.05 mm/s</td>
<td>0.05 mm/s</td>
<td>$1 \times 10^{-6}$%</td>
<td>None</td>
</tr>
</tbody>
</table>

inertial frame and the body-fixed frame, respectively.

$$\tilde{\mathbf{q}}^S_T = \delta \tilde{\mathbf{q}}(\tilde{\eta}^S_T) \otimes \delta \tilde{\mathbf{q}}(\tilde{\varepsilon}^S_T) \otimes \tilde{\mathbf{q}}^B_T \otimes \tilde{\mathbf{q}}^B$$ \hspace{1cm} (4.37)

The altimeter outputs the altitude of the vehicle. The model for the altimeter is Eq. 4.38.

$$\tilde{h} = (1 + s_{alt})(|\tilde{\mathbf{r}}^I| - h_{\text{terrain}} - R_{\text{moon}}) + b_{alt}$$ \hspace{1cm} (4.38)

Note that $R_{\text{moon}}$ is the lunar radius. Also, $h_{\text{terrain}}$ is a time-varying quantity used to model the altitude variations on the lunar surface. This quantity is modeled as a first order Markov process, as shown in Eq. 4.39-4.40. The terrain time constant, $\tau_{\text{terrain}}$, is calculated using the horizontal velocity of the lander, $v_{\text{horiz}}$, and a correlation distance, $d_{\text{corr}}$, which describes the variability of the terrain. The term $w_{\text{terrain}}$ represents white noise with variance $\frac{2\sigma_{\text{terrain}}^2}{\tau_{\text{terrain}}}$. 
\[
\dot{h}_{\text{terrain}} = \frac{h_{\text{terrain}}}{\tau_{\text{terrain}}} + w_{\text{terrain}} \tag{4.39}
\]

\[
\tau_{\text{terrain}} = \frac{d_{\text{corr}}}{v_{\text{horiz}}} \tag{4.40}
\]

When using a small simulation time step, the terrain altitude can be approximated using Eq. 4.41.

\[
h_{\text{terrain},k+1} = h_{\text{terrain},k} \cdot \exp \left( -\frac{\Delta t_{\text{sensors}}}{\tau_{\text{terrain},k}} \right) + w_{\text{terrain},k} \cdot \sigma_{\text{terrain}}^2 \cdot \left[ 1 - e^{\left( -\frac{2\Delta t_{\text{sensors}}}{\tau_{\text{terrain},k}} \right)} \right] \tag{4.41}
\]

The velocimeter outputs the velocity of the lander with respect to the lunar surface. The model for the velocimeter is Eq. 4.42.

\[
\tilde{v}_{v/surface} = (I_{3 \times 3} + \text{diag}(\tilde{s}_{\text{vel}}))(I_{3 \times 3} - \tilde{v}_{\text{vel}} \times) (\tilde{v}^I - \tilde{\omega}_{\text{moon}} \times \tilde{r}^I) + \tilde{b}_{\text{vel}} + \tilde{\eta}_{\text{vel}} \tag{4.42}
\]

The radio navigation system employs a beacon located at the landing site in order to give the range and range rate to the landing site. The radio navigation system requires a model for both the range and the range-rate, and these models are given in Eq. 4.43 and Eq. 4.44.

\[
\tilde{r}_{v/LS} = (1 + s_{\text{range}})(|\tilde{r}^I - \tilde{r}_{LS}^I|) + b_{\text{range}} + \eta_{\text{range}} \tag{4.43}
\]

\[
\dot{\tilde{r}}_{v/LS} = (1 + s_{\text{rrate}}) \left( (\tilde{v}^I - \tilde{v}_{LS}^I) \cdot \left( \frac{\tilde{r}^I - \tilde{r}_{LS}^I}{|\tilde{r}^I - \tilde{r}_{LS}^I|} \right) \right) + b_{\text{rrate}} + \eta_{\text{rrate}} \tag{4.44}
\]

### 4.6 Attitude Control

In order to create a simulation that fully incorporates 6-degrees-of-freedom dynamics, the simulation needed to be able to estimate and control the attitude of the lunar lander.
The lander attitude is stored in quaternion form and updated using measurements from the star tracker and the gyros. Both the gyroscopes and the star tracker provide a measurement at every time step and are assumed to be sufficiently accurate, so the measurements are not processed through a Kalman filter like most of the measurements in the simulation.

Attitude control is provided by a proportional-plus-derivative (PD) controller. A PD controller with a zero at $z_c$ has the transfer function shown in Eq. 4.45. A controller of this type speeds up the response of a system, although it is still vulnerable to steady-state errors [21].

$$G_c(s) = s + z_c \tag{4.45}$$

A PD controller would probably not be the type chosen for an actual mission, as a more sophisticated type of attitude controller could probably yield better results. However, the goals of this simulation were focused on the navigation and guidance algorithms, so a relatively simple controller was selected so as not to add unnecessary complexity. A more sophisticated control algorithm can easily be incorporated later, if desired.

A convenient form of the PD controller for implementation in the simulation is shown in Eq. 4.46. This equation gives a torque command based on the errors from the desired angular position and velocity.

$$\vec{T}_{commanded} = K_\theta(\Delta \theta) + K_\omega(\vec{\omega}_{gyro} - \vec{\omega}_{desired}) \tag{4.46}$$

The small angle deviation between the desired vehicle attitude and the current vehicle attitude is calculated from the current inertial-to-body quaternion and the desired inertial-to-body quaternion, as shown in Eq. 4.47 and 4.48.

$$\bar{q}_{vehicle\rightarrow des} = \begin{bmatrix} \bar{q}_{vehicle\rightarrow des} \\ q_0 \end{bmatrix} = \bar{q}_{vehicle}^{-1} \otimes \bar{q}_{des} \tag{4.47}$$

$$\Delta \tilde{\theta} = 2 \cdot \bar{q}_{vehicle\rightarrow des} \tag{4.48}$$
The current attitude, $\bar{q}_{\text{vehicle}}$, is simply the output of the star tracker multiplied by a transformation from the body-fixed frame to the star tracker frame, as shown in Eq. 4.49. The desired attitude, $\bar{q}_{\text{des}}$, is calculated as shown in Eq. 4.50, using the commanded acceleration from the guidance algorithm. $T_{des}$ is the transformation matrix from the inertial frame to the desired orientation. It is calculated as shown in Eq. 4.51 through Eq. 4.55.

\begin{equation}
\bar{q}_{\text{vehicle}} = \bar{q}^{ST\rightarrow B} \otimes \bar{q}_{ST} \tag{4.49}
\end{equation}

\begin{equation}
\bar{q}_{\text{des}} = \bar{q}(T_{des}) \tag{4.50}
\end{equation}

\begin{equation}
T_{des} = \begin{bmatrix}
-\hat{\tau}_T^- \\
-\hat{\tau}_y^- \\
-\hat{\tau}_z^-
\end{bmatrix} \tag{4.51}
\end{equation}

\begin{equation}
\hat{i}_h = \left( \frac{\hat{r}^I \times \hat{v}^I}{|\hat{r}^I \times \hat{v}^I|} \right) \tag{4.52}
\end{equation}

\begin{equation}
\hat{i}_x = \frac{\hat{a}_{cmd}}{|\hat{a}_{cmd}|} \tag{4.53}
\end{equation}

\begin{equation}
\hat{i}_y = \left( \frac{\hat{x}_x \times \hat{i}_h}{|\hat{x}_x \times \hat{i}_h|} \right) \tag{4.54}
\end{equation}

\begin{equation}
\hat{i}_z = \left( \frac{\hat{x}_y \times \hat{i}_y}{|\hat{x}_y \times \hat{i}_y|} \right) \tag{4.55}
\end{equation}

The current angular velocity, $\bar{\omega}_{gyro}$, is the output of the gyroscope. The desired angular velocity, $\bar{\omega}_{\text{des}}$, is either assumed to be a 3 x 1 vector of zeroes, or calculated from the previous and current commanded acceleration using Eq. 4.56. The angle between the previous and current commanded acceleration is $\Delta \theta$.

\begin{equation}
\bar{\omega}_{\text{des}} = \left( \frac{\Delta \theta}{dt_{sim}} \right) \left( \frac{\hat{a}_{cmd,t-1} \times \hat{a}_{cmd,t}}{|\hat{a}_{cmd,t-1} \times \hat{a}_{cmd,t}|} \right) \tag{4.56}
\end{equation}
The controller gains, $K_\theta$ and $K_\omega$, are calculated using Eq. 4.57 and Eq. 4.58.

\[
K_\theta = \omega_n^2 I \quad \text{(4.57)}
\]
\[
K_\omega = 2\xi \omega_n I \quad \text{(4.58)}
\]

The vehicle inertia tensor is $I$. The terms $\omega_n$ and $\xi$ are the controller natural frequency and the controller damping ratio, respectively. For the purposes of this simulation, $\omega_n$ was selected so that the natural period of the controller was 5 times larger than the simulation time step. The controller damping ratio was assumed to be 0.707.
Chapter 5

Results

Four different cases were studied. These cases differed based on their use of a full or reduced navigation state and the sensor suite employed. The inertial only cases used accelerometers, gyroscopes, a star tracker, an altimeter, and a velocimeter. The inertial plus radio navigation cases used all of the sensors from the inertial only cases, in addition to a radio navigation system which gives a range and range rate to a beacon at the landing site. For each case, there were also three different sets of inputs (including initial conditions, sensor specifications, and actuator specifications), categorized as Typical, Best, and Worst inputs. The cases that were studied are summarized in Table 5.1, and the inputs for the Typical, Best, and Worst cases are given in Tables 4.5, 4.6, 4.9, and 4.7.

For each case the objective was to guide the lander to a position 100 m above the desired landing site with zero horizontal velocity and 8 m/s vertical velocity. In all cases the length of the fine count was 2 seconds in the braking phase and 4 seconds in the approach phase. When applicable, the acquisition of the radio navigation measurements occurred at a range of 100 km.

<table>
<thead>
<tr>
<th>Table 5.1: Summary of cases to be studied.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Full Navigation State</strong></td>
</tr>
<tr>
<td>Case 1  Inertial only</td>
</tr>
<tr>
<td>Case 2  Inertial plus radio navigation</td>
</tr>
<tr>
<td><strong>Reduced Navigation State</strong></td>
</tr>
<tr>
<td>Case 3  Inertial only</td>
</tr>
<tr>
<td>Case 4  Inertial plus radio navigation</td>
</tr>
</tbody>
</table>
5.1 Full Navigation State Data

The results for Case 1 and Case 2 can be found in Sections 5.1.1 and 5.1.2, respectively. These are the cases where the full navigation state was used. Sensor errors were estimated as part of the expanded navigation state, then those estimates were used to compensate for some of the sensor measurement errors.

5.1.1 Case 1 - Inertial Only

Case 1 uses the full navigation state and the inertial-only sensor suite. Figures 5.1, 5.2, and 5.3 show the position and velocity errors in LVLH for the best, typical, and worst scenarios, respectively. The final position and velocity for each of the different error levels are given in Fig. 5.4 through Fig. 5.6. The final errors for each of the three error levels are given in Fig. 5.7 through Fig. 5.9.

As expected, the final dispersions for position and velocity and the final errors are larger for the Typical and Worst scenarios than for the Best scenario. The final errors and distributions for the Best scenario are also more Gaussian than for the Typical and Worst scenarios, likely due to nonlinearities that become more pronounced at the higher error levels.

For all three error levels, there are peaks in the LVLH position and velocity true dispersions just after 3200 seconds, when the braking phase of descent begins. There are also visible peaks in the altitude position true dispersion and the velocity true dispersions at the beginning of the approach phase. Until braking begins, the true dispersions agree very closely with the filter standard deviation and the navigation errors.

5.1.2 Case 2 - Inertial and Radio Navigation

Case 2 uses the full navigation state and the inertial-plus-radio-navigation sensor suite. Figures 5.10, 5.11, and 5.12 show the position and velocity errors in LVLH for the best, typical, and worst scenarios, respectively. The final position and velocity for each of the different error levels are given in Fig. 5.13 through Fig. 5.15. The final errors for each of the three error levels are given in Fig. 5.16 through Fig. 5.18.
Fig. 5.1: LVLH errors, full navigation state, inertial measurements only, Best scenario.
Fig. 5.2: LVLH errors, full navigation state, inertial measurements only, Typical scenario.
Fig. 5.3: LVLH errors, full navigation state, inertial measurements only, Worst scenario.
Fig. 5.4: Final position and velocity, full navigation state, inertial measurements only, Best scenario.
Fig. 5.5: Final position and velocity, full navigation state, inertial measurements only, Typical scenario.
Fig. 5.6: Final position and velocity, full navigation state, inertial measurements only, Worst scenario.
Fig. 5.7: Crosstrack versus downrange errors, full navigation state, inertial measurements only, Best scenario.

Fig. 5.8: Crosstrack versus downrange errors, full navigation state, inertial measurements only, Typical scenario.
Some of the same behavior that was observed in Case 1 can also be seen in Case 2, including peaks in the position and velocity at the beginning of the braking and approach guidance phases, higher dispersions on cases with higher error levels, and more nonlinear distributions on the Best scenario than the Typical or Worst scenario.

Any differences in the results for Case 1 and Case 2 should be primarily due to the inclusion of range and range rate measurements in Case 2. The position and velocity true dispersions on Case 2 tend to be slightly higher than the dispersions from Case 1 for the Best and Typical scenarios, although the dispersions for the Worst scenario are higher for Case 2. The final position downrange errors seem to be less strongly correlated with the final position crosstrack errors in Case 2 than in Case 1.
Fig. 5.10: LVLH errors, full navigation state, inertial + radio navigation measurements, Best scenario.
Fig. 5.11: LVLH errors, full navigation state, inertial + radio navigation measurements, typical scenario.
Fig. 5.12: LVLH errors, full navigation state, inertial + radio navigation measurements, Worst scenario.
Fig. 5.13: Final position and velocity, full navigation state, inertial + radio navigation measurements, Best scenario.
Fig. 5.14: Final position and velocity, full navigation state, inertial + radio navigation measurements, Typical scenario.
Fig. 5.15: Final position and velocity, full navigation state, inertial + radio navigation measurements, Worst scenario.
Fig. 5.16: Crosstrack versus downrange errors, full navigation state, inertial + radio navigation measurements, Best scenario.

Fig. 5.17: Crosstrack versus downrange errors, full navigation state, inertial + radio navigation measurements, Typical scenario.
5.2 Reduced Navigation State Data

The results for Case 3 and Case 4 can be found in Sections 5.2.1 and 5.2.2. These are cases where a reduced navigation state (only position, velocity, and landing site position) was used.

5.2.1 Case 3 - Inertial Only

Case 3 uses the reduced navigation state and the inertial-only sensor suite. Figures 5.19, 5.20, and 5.21 show the position and velocity errors in LVLH for the best, nominal, and worst scenarios, respectively. The final position and velocity for each of the different error levels are given in Fig. 5.22 through Fig. 5.24. The final errors for each of the three error levels are given in Fig. 5.25 through Fig. 5.27.

There is virtually no difference in the average final differences from the position and velocity targets for Cases 1, 2, and 3. For all three error levels, the position true dispersions
for Case 3 tend to be higher than the true dispersions for Case 1, which used the full navigation state rather than the reduced navigation state used in Case 3. However, for the Best and Worst scenarios, the true dispersions in velocity are lower than the dispersions for Case 1.

5.2.2 Case 4 - Inertial and Radio Navigation

Case 4 uses the reduced navigation state and the inertial-plus-radio-navigation sensor suite. Figures 5.28, 5.29, and 5.30 show the position and velocity errors in LVLH for the best, typical, and worst scenarios, respectively. The final position and velocity for each of the different error levels are given in Fig. 5.31 through Fig. 5.33. The final errors for each of the three error levels are given in Fig. 5.34 through Fig. 5.36.

The average final differences from the position and velocity targets for Case 4 agree very closely with the values from Cases 1, 2 and 3. The position true dispersions for Case 4 are higher for both the Typical and Worst scenarios as compared to Case 2, which uses the full navigation state rather than the reduced navigation state. The velocity true dispersions for the Typical case are also higher for Case 4 than Case 2. Like in Case 2, which also used radio navigation measurements, there is not the strong correlation between the final downrange and crosstrack errors that was observed in Case 1 or Case 3.
Fig. 5.19: LVLH errors, reduced navigation state, inertial measurements only, Best Scenario.
Fig. 5.20: LVLH errors, reduced navigation state, inertial measurements only, Typical scenario.
Fig. 5.21: LVLH errors, reduced navigation state, inertial measurements only, Worst scenario.
Fig. 5.22: Final position and velocity, reduced navigation state, inertial measurements only, Best scenario.
Fig. 5.23: Final position and velocity, reduced navigation state, inertial measurements only, Typical scenario.
Fig. 5.24: Final position and velocity, reduced navigation state, inertial measurements only, Worst scenario.
Fig. 5.25: Crosstrack versus downrange errors, reduced navigation state, inertial measurements only, Best scenario.

Fig. 5.26: Crosstrack versus downrange errors, reduced navigation state, inertial measurements only, Typical scenario.
Fig. 5.27: Crosstrack versus downrange errors, reduced navigation state, inertial measurements only, Worst scenario.
Fig. 5.28: LVLH errors, reduced navigation state, inertial + radio navigation measurements, Best scenario.
Fig. 5.29: LVLH errors, reduced navigation state, inertial + radio navigation measurements, Typical scenario.
Fig. 5.30: LVLH errors, reduced navigation state, inertial + radio navigation measurements, Worst scenario.
Fig. 5.31: Final position and velocity, reduced navigation state, inertial + radio navigation measurements, Best scenario.
Fig. 5.32: Final position and velocity, reduced navigation state, inertial + radio navigation measurements, Typical scenario.
Fig. 5.33: Final position and velocity, reduced navigation state, inertial + radio navigation measurements, Worst scenario.
Fig. 5.34: Crosstrack versus downrange errors, reduced navigation state, inertial + radio navigation measurements, Best scenario.

Fig. 5.35: Crosstrack versus downrange errors, reduced navigation state, inertial + radio navigation measurements, Typical scenario.
Fig. 5.36: Crosstrack versus downrange errors, reduced navigation state, inertial + radio navigation measurements, Worst scenario.
5.3 Fuel Use

Scatter plots of the delta-v used for all 300 runs for each case and error level are given in Fig. 5.37 through Fig. 5.39.

When the error levels are increased, both the average total $\Delta V$ and the standard deviation of the $\Delta V$ used increase. In the majority of runs, the total $\Delta V$ used is less than the nominal $\Delta V$. This is likely because the nominal guidance trajectory is meant to approximate an optimal trajectory, so it should use close to the minimum possible $\Delta V$ for the trajectory.

The cases that use inertial measurements only are more likely to fall below the nominal trajectory than the cases that also utilize range and range rate measurements. This could be because the cases with radio navigation measurements use slightly more fuel correcting the trajectory after improvements in the landing site position estimate due to the additional measurements.
Fig. 5.37: $\Delta V$ used, all cases, Best errors; Clockwise from upper left: full navigation state with inertial measurements only, full navigation state with inertial and radio navigation measurements, reduced navigation state with inertial and radio navigation measurements, and reduced navigation state with inertial measurements only.
Fig. 5.38: $\Delta V$ used, all cases, Typical errors; Clockwise from upper left: full navigation state with inertial measurements only, full navigation state with inertial and radio navigation measurements, reduced navigation state with inertial and radio navigation measurements, and reduced navigation state with inertial measurements only.
Fig. 5.39: $\Delta V$ used, all cases, Worst errors; Clockwise from upper left: full navigation state with inertial measurements only, full navigation state with inertial and radio navigation measurements, reduced navigation state with inertial and radio navigation measurements, and reduced navigation state with inertial measurements only.

5.4 Results Summary Tables

The results from each of the cases studied for Best, Typical, and Worst case errors are summarized in the following tables. The total $\Delta V$ used as well as the final $\Delta V$ 1-$\sigma$ true dispersions are compared in Table 5.2. The final position 1-$\sigma$ true dispersions, as well as the nominal and average distance from the position guidance target, are compared in Table 5.3. Finally, the final velocity 1-$\sigma$ true dispersions and the nominal and average difference from
the velocity guidance target are in Table 5.4. It is noted that these results are calculated at the beginning of terminal descent so the nominal and average differences from the guidance targets are not expected to be zero.

Table 5.2: Total $\Delta V$ used, by case and error level.

<table>
<thead>
<tr>
<th>Case</th>
<th>Nominal $\Delta V$ (m/s)</th>
<th>Errors</th>
<th>Average $\Delta V$ for 300 runs (m/s)</th>
<th>$\Delta V$ Used 1-$\sigma$ (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inertial Only, Full State</td>
<td>1825.87</td>
<td>Best</td>
<td>1825.90</td>
<td>0.062566</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typical</td>
<td>1830.74</td>
<td>17.3070</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Worst</td>
<td>1959.74</td>
<td>263.483</td>
</tr>
<tr>
<td>Inertial + Radio Navigation, Full State</td>
<td>1825.87</td>
<td>Best</td>
<td>1825.94</td>
<td>0.076899</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typical</td>
<td>1830.49</td>
<td>43.9205</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Worst</td>
<td>2092.20</td>
<td>474.347</td>
</tr>
<tr>
<td>Inertial Only, Reduced State</td>
<td>1825.87</td>
<td>Best</td>
<td>1825.89</td>
<td>0.080940</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typical</td>
<td>1830.61</td>
<td>18.7763</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Worst</td>
<td>1943.57</td>
<td>243.030</td>
</tr>
<tr>
<td>Inertial + Radio Navigation, Reduced State</td>
<td>1825.87</td>
<td>Best</td>
<td>1825.93</td>
<td>0.055243</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typical</td>
<td>1829.21</td>
<td>32.7229</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Worst</td>
<td>2209.05</td>
<td>1759.65</td>
</tr>
</tbody>
</table>
Table 5.3: Final position dispersions, by case and error level.

<table>
<thead>
<tr>
<th></th>
<th>Nominal Distance from Target (m)</th>
<th>Errors</th>
<th>Average Distance from Target (m)</th>
<th>Position 1-σ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best 43.9963</td>
<td>[0.015568, 0.077655, 0.024526]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Typical 44.4000</td>
<td>[1.56868, 0.377548, 0.980345]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Worst 58.5238</td>
<td>[22.7136, 23.5279, 15.0253]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inertial Only, Full State</td>
<td>43.9900</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inertial + Radio Navigation, Full State</td>
<td>43.9900</td>
<td>Best 43.9950</td>
<td>[0.031541, 0.006625, 0.008360]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Typical 43.6227</td>
<td>[2.81257, 1.09026, 0.786026]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Worst 45.0915</td>
<td>[11.0279, 1.78699, 1.61403]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inertial Only, Reduced State</td>
<td>43.9900</td>
<td>Best 43.9900</td>
<td>[0.020520, 0.081536, 0.025288]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Typical 44.5567</td>
<td>[4.42360, 1.63983, 0.548847]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Worst 50.7718</td>
<td>[32.9216, 56.4854, 35.5162]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inertial + Radio Navigation, Reduced State</td>
<td>43.9900</td>
<td>Best 43.9923</td>
<td>[0.023015, 0.006643, 0.008629]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Typical 43.2241</td>
<td>[5.31469, 0.878223, 0.472677]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Worst 43.7132</td>
<td>[10.9420, 12.1093, 11.3980]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inertial Only, Full State</td>
<td>Nominal Difference from Target (m/s)</td>
<td>Errors</td>
<td>Average Difference from Target (m/s)</td>
<td>Velocity 1-σ (m/s) [RAD, CT, DR]</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------------</td>
<td>--------</td>
<td>-------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td></td>
<td>1.35183</td>
<td>Best</td>
<td>1.35191</td>
<td>[0.004610, 0.000348, 0.000096]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typical</td>
<td>1.32166</td>
<td>[1.17504, 0.025825, 0.028411]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Worst</td>
<td>1.18208</td>
<td>[23.7264, 3.29415, 5.07140]</td>
</tr>
<tr>
<td>Inertial + Radio Navigation, Full State</td>
<td>1.35183</td>
<td>Best</td>
<td>1.34557</td>
<td>[0.027405, 0.001111, 0.001885]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typical</td>
<td>1.65719</td>
<td>[3.96571, 0.504945, 0.104197]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Worst</td>
<td>2.61327</td>
<td>[10.3917, 0.231638, 0.372892]</td>
</tr>
<tr>
<td>Inertial Only, Reduced State</td>
<td>1.35183</td>
<td>Best</td>
<td>1.35234</td>
<td>[0.004469, 0.000386, 0.000106]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typical</td>
<td>1.07341</td>
<td>[2.94771, 0.082658, 0.021405]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Worst</td>
<td>3.30353</td>
<td>[8.28668, 9.41011, 2.51452]</td>
</tr>
<tr>
<td>Inertial + Radio Navigation, Reduced State</td>
<td>1.35183</td>
<td>Best</td>
<td>1.34764</td>
<td>[0.020355, 0.000837, 0.001646]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typical</td>
<td>1.45124</td>
<td>[1.23330, 0.329230, 0.049597]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Worst</td>
<td>2.31544</td>
<td>[9.92295, 7.81398, 1.21823]</td>
</tr>
</tbody>
</table>
Chapter 6

Conclusions

In general, cases using radio navigation measurements had higher fuel use dispersions (as measured by the $\Delta V$ standard deviation) than cases using only inertial measurements. This is likely due to the fact that range and range rate measurements are received fairly late in the trajectory, and extra $\Delta V$ is needed to correct the vehicle trajectory once the landing site position estimate is updated using these measurements. This is especially true when random errors in the range and range rate measurements or the landing site are particularly high. The cases including radio navigation measurements also had larger position dispersions (as measured by the final position standard deviation) for the same reasons. The final velocity dispersions for these cases tend to be slightly larger, as well, although this is not true for all cases.

Using the full navigation state rather than the reduced navigation state does not have a clear effect on the final dispersions in $\Delta V$, position, and velocity. On cases where inertial only measurements are used, the final dispersions for cases with a full navigation state tend to be lower than those with a reduced navigation state. Conversely, for cases including radio navigation measurements, the final dispersions tend to be higher for cases with the full navigation state rather than a reduced navigation state. For these cases, it is possible that nonlinearities caused by large sensor errors obscured some of the effect of navigation state selection, especially because it was noticed during testing that the radio navigation measurements tend to be very sensitive to high error values. Further testing may be able to better capture the trends caused by changing the size of the navigation state.

When compared on the final dispersions alone, the full state navigation has advantage. However, the additional code (and programming complexity) necessary to propagate and update 33 navigation states rather than 9 navigation states is significant. In the simulation...
developed for this research, the equations of motion used to propagate the navigation state required 44% more lines of code and the navigation subroutine required 23% more lines of code for the full state navigation cases. In addition, using a reduced navigation state results in a 10-20% savings on computational time for a simulation with 300 Monte Carlo sample runs. As long as slightly larger dispersions were acceptable, a reduced state navigation filter would be a better choice.

The thesis of this research, that a feedback control and guidance system with a reduced navigation state would perform nearly as well as a full-state feedback system, was supported. Using the reduced navigation state usually increased the total fuel use dispersions, position dispersions, and velocity dispersions. However, in most cases, a slight loss of accuracy would be acceptable in exchange for the significant savings in computational time and code complexity offered by the reduced navigation state system.
References


[10] Sostaric, R. R., “Powered Descent Trajectory Guidance and Some Considerations for
Human Lunar Landing,” *30th Annual AAS Guidance and Control Conference*, No. 07-
051, American Astronautical Society Rocky Mountain Section, 2007.

Space Vehicle,” *Supplement to IEEE Transactions on Aerospace and Electronic Sys-


Moscow, Russia, 2004.


[16] Bishop, R. H., Dubois-Matra, O., and Ely, T., “Robust Entry Navigation Using Hier-
archal Filter Architectures Regulated with Gating Networks,” *16th International Sym-

AAS 009-057, February 2009.

Descent and Landing,” *Journal of Spacecraft and Rockets*, Vol. 46, No. 6, November-
