5-2010

Guidance and Navigation Linear Covariance Analysis for Lunar Powered Descent

Travis J. Moesser
Utah State University

Recommended Citation

Moesser, Travis J., "Guidance and Navigation Linear Covariance Analysis for Lunar Powered Descent" (2010). *All Graduate Theses and Dissertations*. Paper 654.

GUIDANCE AND NAVIGATION LINEAR COVARIANCE ANALYSIS FOR LUNAR POWERED DESCENT

by

Travis J. Moesser

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

of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

Dr. David K. Geller
Major Professor

Dr. R. Rees Fullmer
Committee Member

Dr. Barton L. Smith
Committee Member

Dr. Byron R. Burnham
Dean of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2010

Abstract

Guidance and Navigation Linear Covariance Analysis for Lunar Powered Descent

by

Travis J. Moesser, Master of Science

Utah State University, 2010

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

A linear covariance analysis is conducted to assess closed-loop guidance, navigation, and control system (GN&C) performance of the Altair vehicle during lunar powered descent. Guidance algorithms designed for lunar landing are presented and incorporated into the closed-loop covariance equations. Navigation-based event triggering is also included in the covariance formulation to trigger maneuvers and control dispersions. Several navigation and guidance trade studies are presented demonstrating the influence of triggering and guidance and study parameters on the vehicle GN&C performance.

(123 pages)

To my wife and my children. I love you.

Acknowledgments

I would like to acknowledge the support and thank several people who helped make this thesis possible. The ALHAT teams provided key support for this research, specifically Stephen Paschall and Tom Fill at the Draper Laboratory and Ron Sostaric at the Johnson Space Center. Several current and former members of the Guidance, Navigation, and Control graduate research group in the Department of Mechanical Engineering at Utah State University provided very helpful support and feedback, too many to name. Thank you also to my supervisors, Tom Higgs and Brian Allen, and my coworkers in the Integrated Performance / Advanced Engineering group at ATK Aerospace Systems at Promontory, Utah, for their encouragement and helpfulness, technical expertise and professionalism, and willingness to be flexible with me as I finished my academic career.

Special thanks and acknowledgment belong to my major professor, Dr. David K. Geller, whose work is the basis of my thesis, and without whom I could have never finished (nor started) this research. I am especially grateful for his unflagging enthusiasm and earnest attention to me and my work. Thanks are also due to Bonnie Ogden, my departmental advisor, who never gave up on me and made sure that administrivia never impeded my thesis completion.

The most sincere thanks I can give are to my wife, Misty. Her unconditional patience with me, sacrificing uncounted nights and weekends at home alone, then raising our children, first Jack and then Charlie, while I struggled to finish this research, were of paramount importance. She has pushed aside her goals and put her life on hold while I arduously completed this one. We finally finished, Honey.

Travis J. Moesser

Contents

List of Tables

List of Figures

Figure Page

Chapter 1

Introduction

In January 2004, President George W. Bush announced the "Vision for Space Exploration," calling for NASA, along with completing the International Space Station and retiring the Space Shuttle, to develop technologies to explore the Moon within the next two decades, leading eventually to the exporation of Mars. NASA has begun to develop some of the vehicles and technologies necessary to return to the Moon, collectively called Project Constellation since August 2006.

One vehicle, the Lunar Surface Access Module (LSAM), named *Altair*, is designed to carry astronauts to the lunar surface and return them to the Orion (also known as the Crew Exploration Vehicle or CEV) capsule in low lunar orbit. NASA is developing a technology for Altiar known as Autonomous Landing and Hazard Avoidance Technology (ALHAT) that will aid astronauts make safe landings on the lunar surface. ALHAT will also be incorporated into unmanned lunar landers delivering scientific payloads or cargo to the Moon.

The LSAM guidance and navigation software operates in conjunction with the vehicle avionics and human operators, providing guidance commands to the vehicle based on the its navigation state and the guidance target. The vehicle's avionics system maintains direct control of the vehicle, translating guidance acceleration commands into vehicle attitude and thrust commands. Hazard avoidance systems will have the ability to detect surface hazards (boulders, craters, human installations, etc.) and autonomously redesignate guidance targets for landing. ALHAT can operate independently or with human-in-the-loop feedback. Collectively, the LSAM avionics, inertial sensors, and flight software are referred to here as the Altair Guidance, Navigation, and Control system (GN&C or GNC). The Altiar GNC system is the focus of the present research.

In the development of any control system, it is necessary to analyze its performance before building hardware. In the context of autonomous lunar landing guidance, “performance” can be measured primarily by the difference between the *target* final position and velocity to the *actual* final position and velocity, and secondarily by the amount of propellant consumed in getting there. The first measure of performance may be thought of as the statistical likelihood of the final position lying within a certain radius (in a two-dimensional sense) from the target. Expressing performance as a statistical uncertainty (or alternately as a confidence level) is a common means of quantitatively evaluating a system’s performance. In engineering, such data for nontrivial systems may be obtained in one of two ways—by experimentation or by computer modeling and analysis. For obvious reasons, computer analysis must be used for statistically analyzing the performance of the powered descent guidance and navigation system on the Lunar Surface Access Module.

Typically in computer modeling, obtaining statistics of a dynamic system requires simulating the entire system, with its inherent random effects also modeled, hundreds or thousands of times. The pertinent data from the simulations are statistically analyzed afterward using ensemble averaging to determine the time-evolution of the statistical data of interest. This sort of analysis is often called Monte Carlo analysis, and can be costly and time-consuming [1].

An alternative analysis technique is called Linear Covariance analysis (LinCov), sometimes called mean analysis or covariance analysis [1]. LinCov analysis is able to generate the necessary statistical information for performance analysis in a single run, if the dynamics, measurements, and controls are linear. Alternatively, the models may be *linearized* about a nominal reference trajectory [2]. If the linearized models are valid, the statistical results obtained may be sufficiently adequate for most engineering design applications. Large dispersions from the reference may indicate that the linear model is not valid for those regions.

Rather than simulating the actual states of a dynamic system, a LinCov tool only propagates the covariance of the state. The covariance equations are formulated much like an extended Kalman filter. Also, true- and navigation-state dispersions about a nominal trajectory are propagated using linearized equations of motion. Like an extended Kalman filter, the covariance of the state is propagated using linearized dynamics and updated using measurement partial derivatives [1,2].

LinCov analysis has several advantages when compared to Monte Carlo analysis. One use of LinCov is to isolate one source of error in the scenario to see its individual contribution to the total error, in what is called "sensitivity analysis." In this way, it is relatively simple to see the major and minor sources of uncertainty and how they change during the simulation. Another advantage is the capability to get results after only one run. This feature makes it easy to investigate different scenarios and to conduct trade studies, by running many scenarios in a short period of time. It is also possible to let one variable run the gambit of possibilities, making it possible to find an optimum if a performance or cost function is defined. LinCov lends itself to these sorts of parameter studies, and with few changes it would be possible to adapt LinCov to work with a parameter optimizer.

The linearization process necessary to develop a LinCov tool adds time and work necessary to derive the formulas, not so with Monte Carlo analysis. However, the advantages listed above show that the extra work necessary in derivation may be offset by the added capabilities and quicker speed of execution.

The goal of the present research is to use LinCov analysis to validate the capability of the Altair GN&C system to ensure safe and accurate landing on the lunar surface that meets desired landing requirements. The results of these studies are of interest to engineers at the NASA Johnson Space Center and at facilities designing ALHAT components such as the Charles Stark Draper Laboratory. This research demonstrates the flexibility and power of linear covariance as an engineering design tool.

Chapter 2

Literature Survey

The scope of this research work covers several areas of the broad topic of Guidance, Navigation, and Control. Areas surveyed for the purposes of this research include planetary landing guidance, planetary landing navigation, and linear covariance analysis.

2.1 Planetary Landing Guidance

Landing safely on another world is not an easy task. Engineers have employed a number of techniques to deliver scientific payloads to the surfaces of extraterrestrial bodies. In approaching this topic, it is important to note that the techniques used by one mission to land on a planet, very often, cannot be used by another mission to another planet. (In this context, the word "planet" is used to denote any orbiting body with a solid surface.) A few distinctions are made here in categorizing a landing mission.

A *soft landing* differs from an impact or hard landing in that the spacecraft reaches the surface of the planet with small velocities relative to the surface. Loosely defined, a "soft landing" is one in which a human would survive impact, while a "hard landing" is one in which only robotic probe appropriately designed would survive impact at moderately high velocities, and with an "impact" the spacecraft is not expected to survive. The difference is analogous to that of rendezvous and intercept in guidance problems, in that the final (relative) velocity is constrained to zero for rendezvous and soft landing [3]. A *powered* landing uses thrusters to null the vehicle's velocity up to immediately before and sometimes including touchdown. Unpowered landing typically incorporates passive means to reduce velocity, typically parachutes or airbags. Atmospheric landings typically require more design work than do landings in an *atmosphere-free* environment, due to the additional aerodynamic drag and lift forces, and entry heating. Naturally,

Table 2.1: Time Table of *Successful* Soft Landings on the Lunar Surface

Year	Mission	Nation	Notes
1966	Surveyor 1	United States	First soft landing[a]
1967	Surveyor 3, 5, 6	United States	
1968	Surveyor 7	United States	
1969	Apollo 11	United States	First manned landing
1969	Apollo 12	United States	Manned
1970	Luna 16	Soviet Union	Soil return
1970	Luna 17/Lunokhod 1	Soviet Union	Rover
1971	Apollo 14, 15	United States	Manned
1972	Luna 20	Soviet Union	Soil return
1972	Apollo 16, 17	United States	Manned
1973	Luna 21/Lunokhod 2	Soviet Union	Rover
1974	Luna 23	Soviet Union	Soil return (failed)
1976	Luna 24	Soviet Union	Soil return

[a] Soviet Luna 9 and 13 missions in 1966 were successful hard landings.

different dynamic limits must be used when landing *manned* spacecraft than unmanned, robotic spacecraft. The present research is applicable only to soft, powered landings in an atmosphere-free environment with either manned or unmanned vehicles. Particular emphasis is placed on the manned Lunar Surface Access Module or LSAM, named *Altair*.

The Apollo missions of the 1960s and 1970s are the prime examples of soft, powered landing on an atmosphere-free world. The six landings of the Lunar Modules are, to date, the only manned landings on an extraterrestrial body. Smaller unmanned spacecraft have also landed on the moon, Mars, Venus, the saturnian moon Titan (*Huygens*), and asteroids Eros (*NEAR Shoemaker* successfully landed though never designed for that purpose) and Itakowa (*Haybusa* landed although with several problems). A list of successful soft lunar landings is shown in Table 2.1. There are several future soft-landing missions to the surface of our moon and to other bodies in our solar system planned or in the works by many nations in various phases of development and planning. This research focuses on past Apollo missions and future Constellation/Altair missions.

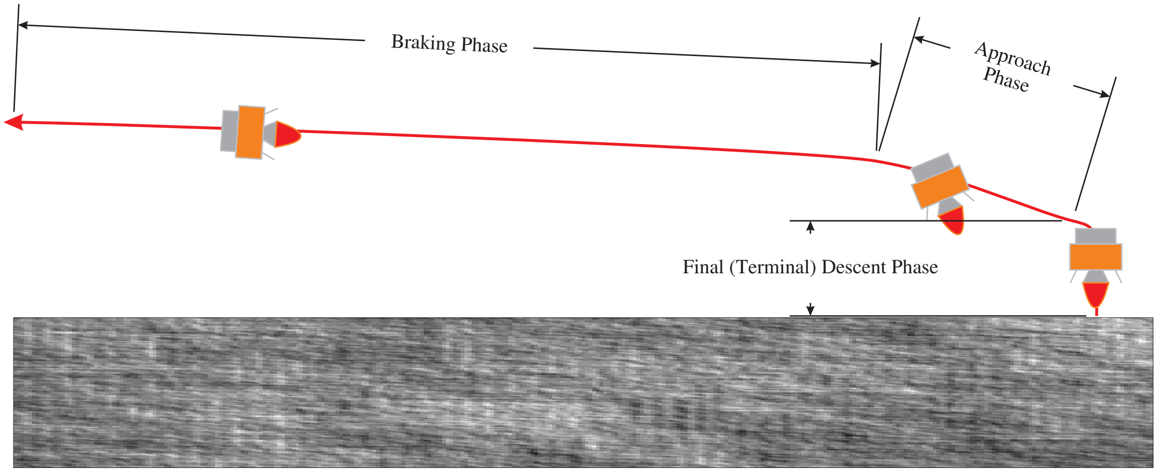

Fig. 2.1: Guidance phases for lunar powered descent, not to scale, adapted from [4].

2.1.1 Apollo/LM Guidance

Lunar powered descent was divided into distinct phases for Apollo, which have carried over into Project Constellation. The three phases, schematically shown in Fig. 2.1, are the *Braking Phase*, during which the majority of the vehicle's kinetic energy is removed by thrusting in the anti-velocity direction; the *Approach Phase*, which pitches the vehicle into an attitude to allow for surface observation along a nearly constant flight path angle or "glide-slope"; and the *Final Descent Phase*, also called the Terminal or Vertical Descent Phase, which slowly guides the vehicle along a nominally vertical path to the surface. Different guidance targets and typically different guidance laws are used for each phase, some of which are discussed in the following sections.

While unmanned vehicles had varying schemes to accomplish the soft landing, all of the Apollo missions used a set of guidance and control computer programs designed to allow crew-interactive targeting and landing, with the on-board GN&C computer executing the maneuvers required for a safe landing. Two variations on the guidance problem, termed "implicit" and "explicit," were proposed, the former being developed too late to be incorporated into the LM Guidance Computer (LGC). The explicit guidance equation was developed by Cherry [5] and simplified and generalized by Klumpp [6–8] for use in the LGC. The explicit guidance equation is a special case of the implicit guidance equation which Klumpp derives [7,8] from a reference trajectory evolving backwards in time from the target end state. Portions of his derivation follow.

Assume a reference trajectory for which five degrees of freedom exist, specifically the quartic polynomial

$$\mathbf{r}_{\text{ref}} = \mathbf{r}_t + \mathbf{v}_t T + \frac{1}{2}\mathbf{a}_t T^2 + \frac{1}{6}\mathbf{j}_t T^3 + \frac{1}{24}\mathbf{s}_t T^4 \tag{2.1}$$

where $\mathbf{r}_{\text{ref}}$ is the position vector on the reference trajectory, and $\mathbf{r}_t$, $\mathbf{v}_t$, $\mathbf{a}_t$, $\mathbf{j}_t$, and $\mathbf{s}_t$ are the target position, velocity, acceleration, jerk, and snap, respectively, where all the vectors are in a consistent guidance frame. The equation is thought as evolving backwards in time from the landing site, where T is the guidance time with $T = 0$ occurring at the target state, so the maneuver is actually flown during negative times T. Another common clock used is the time-to-go or t_{go} which is the time remaining until the guidance target is met, and is usually positive, such that $t_{\text{go}} = -T$.

The implicit guidance equation is derived from differentiation of Eq. (2.1), and applying feedback control to eliminate position and velocity deviations from the reference acceleration. In terms of the current state and guidance targets, the guidance equation is

$$\begin{aligned}\mathbf{a}_c &= \mathbf{a}_{\text{ref}} - \frac{K_V}{T}\left(\mathbf{v} - \mathbf{v}_{\text{ref}}\right) - \frac{K_R}{T^2}\left(\mathbf{r} - \mathbf{r}_{\text{ref}}\right) \\ &= -\mathbf{r}\frac{K_R}{T^2} - \mathbf{v}\frac{K_V}{T} + \mathbf{r}_t\frac{K_R}{T^2} + \mathbf{v}_t\frac{K_V + K_R}{T} + \mathbf{a}_t\left(1 + K_V + \frac{K_R}{2}\right) + \ldots \\ &\quad \mathbf{j}_t\left(1 + \frac{K_V}{2} + \frac{K_R}{6}\right)T + \mathbf{s}_t\left(\frac{1}{2} + \frac{K_V}{6} + \frac{K_R}{24}\right)T^2\end{aligned} \tag{2.2}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the lander's current estimated position and velocity, and K_R and K_V are nondimensional feedback gains. Eq. (2.2) may also be arranged to resemble a forced, linear second-order differential equation with $\omega_n^2 = K_R/T^2$ and $-2\zeta\omega_n = K_V/T$.

As a special case of Eq. (2.2), gains K_V and K_R may be *explicitly* selected so that the jerk and snap terms are eliminated. Those gains, $K_V = -6$ and $K_R = 12$ simplify the implicit guidance equation to what is called the "explicit guidance equation" used on Apollo [5,8]

$$\mathbf{a}_c = \frac{12}{T^2}\left(\mathbf{r}_t - \mathbf{r}\right) + \frac{6}{T}\left(\mathbf{v}_t + \mathbf{v}\right) + \mathbf{a}_t. \tag{2.3}$$

Note that we no longer require targets for the vehicle jerk and snap, since they have been eliminated by the choice of gains. It may be important, however, to know what those acceleration derivatives are, and may be useful in selecting the remaining degree-of-freedom, T.

In the guidance frame, denoted with a pre-superscript g, Eq. (2.1) and its derivative may be written in vector-matrix form as

$$\begin{bmatrix} {}^g\mathbf{r}(T) \\ {}^g\mathbf{v}(T) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & T\mathbf{I} & \frac{1}{2}T^2\mathbf{I} & \frac{1}{6}T^3\mathbf{I} & \frac{1}{24}T^4\mathbf{I} \\ \mathbf{0} & \mathbf{I} & T\mathbf{I} & \frac{1}{2}T^2\mathbf{I} & \frac{1}{6}T^3\mathbf{I} \end{bmatrix} \begin{bmatrix} {}^g\mathbf{r}_t \\ {}^g\mathbf{v}_t \\ {}^g\mathbf{a}_t \\ {}^g\mathbf{j}_{t,ach} \\ {}^g\mathbf{s}_{t,ach} \end{bmatrix}, \tag{2.4}$$

where $\mathbf{I}$ is the 3×3 identity matrix, $\mathbf{0}$ is the 3×3 zero matrix. The subscript *ach* (for "achieved") on the jerk and snap terms indicates that these terms are independent variables (i.e. unknowns) selected to satisfy the system of two equations. By arranging and solving Eq. (2.4), we find [7,8]

$$\begin{bmatrix} {}^g\mathbf{j}_{t,\text{ach}} \\ {}^g\mathbf{s}_{t,\text{ach}} \end{bmatrix} = \begin{bmatrix} -\frac{24}{T^3}\mathbf{I} & -\frac{18}{T^2}\mathbf{I} & -\frac{6}{T}\mathbf{I} & \frac{24}{T^3}\mathbf{I} & -\frac{6}{T^2}\mathbf{I} \\ \frac{72}{T^4}\mathbf{I} & \frac{48}{T^3}\mathbf{I} & \frac{12}{T^2}\mathbf{I} & -\frac{72}{T^4}\mathbf{I} & \frac{24}{T^3}\mathbf{I} \end{bmatrix} \begin{bmatrix} {}^g\mathbf{r}_t \\ {}^g\mathbf{v}_t \\ {}^g\mathbf{a}_t \\ {}^g\mathbf{r}(T) \\ {}^g\mathbf{v}(T) \end{bmatrix}. \tag{2.5}$$

Because T is arbitrary, an additional constraint may be imposed to calculate T, so target redesignations do not produce unnecessarily severe guidance commands. This additional constraint is on the downrange component (z in the Apollo guidance frame) of the jerk that would be achieved at the target. The scalar cubic polynomial that may be extracted from Eq. (2.5), setting a target on the jerk z-component, is

$$\begin{aligned}
{}^{g}j_{t,z} &= -\frac{24}{T^3}{}^{g}r_{t,z} - \frac{18}{T^2}{}^{g}v_{t,z} - \frac{6}{T}{}^{g}a_{t,z} + \frac{24}{T^3}{}^{g}r_z - \frac{6}{T^2}{}^{g}v_z \\
{}^{g}j_{t,z}T^3 &= -24{}^{g}r_{t,z} - 18{}^{g}v_{t,z}T - 6{}^{g}a_{t,z}T^2 + 24{}^{g}r_z - 6{}^{g}v_zT \\
0 &= {}^{g}j_{t,z}T^3 + 6{}^{g}a_{t,z}T^2 + \left(18{}^{g}v_{t,z} + 6{}^{g}v_z\right)T + 24\left({}^{g}r_{t,z} - {}^{g}r_z\right). \qquad (2.6)
\end{aligned}$$

One of the roots of the cubic Eq. (2.6) is the required time T to satisfy all the constraints [7,8].

Equation (2.3) is modified to allow for a lead-time or transport delay due to computation and command execution. By defining a predicted time for command execution, $T_P = T + T_{\text{lead}}$, the Apollo lunar-descent guidance equation becomes

$$\begin{aligned}
\mathbf{a}_c &= \left[3\left(\frac{T_P}{T}\right)^2 - 2\left(\frac{T_P}{T}\right)\right]\frac{12}{T^2}(\mathbf{r}_t - \mathbf{r}) + \left[4\left(\frac{T_P}{T}\right)^2 - 3\left(\frac{T_P}{T}\right)\right]\frac{6}{T}\mathbf{v}_t \\
&\quad + \left[2\left(\frac{T_P}{T}\right)^2 - \left(\frac{T_P}{T}\right)\right]\frac{6}{T}\mathbf{v} + \left[6\left(\frac{T_P}{T}\right)^2 - 6\left(\frac{T_P}{T}\right) + 1\right]\mathbf{a}_t. \qquad (2.7)
\end{aligned}$$

Also, to avoid potentially infinite gains as $T \to 0$, the computer target is chosen beyond the actually desired target, so the lander never achieves the programmed target. Instead, nominally 10 s before the target is achieved, the current guidance program terminates, and another is initiated [8].

The guidance equations for the braking and approach phase, developed by Klumpp [6–8] as outlined above, were coded into guidance programs *P63* and *P64* in the LGC, and was used during all six Apollo lunar landing missions.

2.1.2 Constellation LSAM/Altair Guidance

Engineers at the NASA Johnson Space Center (JSC) have already proposed guidance schemes for future lunar landings using the Altair vehicle (previously the Lunar Surface Access Module or LSAM). The approach taken by Sostaric [4,9] to define a guidance trajectory is to specify the form of the acceleration profile, then setting boundary conditions from its integrals. Two different acceleration profiles have been proposed for three distinct guidance phases. Originally, the braking phase was designed to use a linear acceleration profile in the form

$$\mathbf{a} = \mathbf{c}_0 + \mathbf{c}_1 \tau \tag{2.8}$$

while the approach and terminal descent phases use a quadratic acceleration profile

$$\mathbf{a} = \mathbf{c}_0 + \mathbf{c}_1 \tau + \mathbf{c}_2 \tau^2, \tag{2.9}$$

where $\tau = t - t_{\text{guid}}$ is the positive time elapsed from some guidance computation cycle, and $\mathbf{c}_i$ are 3×1 vectors of coefficients calculated at that cycle [4]. A modified version of the Shuttle Ascent Powered Explicit Guidance (PEG) developed at the Charles Stark Draper Laboratory (CSDL) has since been proposed for use with ALHAT [9]. PEG is described in detail in Section 2.1.3. Also proposed for the terminal descent phase is a simple proportional error feedback law [9].

By integrating the acceleration profiles from a zero initial time to a fixed final time t_{go}, the coefficients may be found analytically by solving the boundary-value problem suggested by the acceleration profiles. With the linear acceleration profile,

$$\begin{aligned} \int_0^{t_{\text{go}}} \mathbf{a}\, d\tau = \mathbf{v}_t = &\quad \mathbf{c}_0 t_{\text{go}} + \frac{1}{2}\mathbf{c}_1 t_{\text{go}}^2 + \mathbf{v}_0 \\ \int_0^{t_{\text{go}}} \int_0^{\tau} \mathbf{a}\, d\tau\, d\tau = \mathbf{r}_t = &\ \frac{1}{2}\mathbf{c}_0 t_{\text{go}}^2 + \frac{1}{6}\mathbf{c}_1 t_{\text{go}}^3 + \mathbf{v}_0 t_{\text{go}} + \mathbf{r}_0 \end{aligned}$$

Sostaric and Rea [4] solve for the unknown coefficients in these equations from the known initial conditions $\mathbf{r}_0$ and $\mathbf{v}_0$, the target conditions $\mathbf{r}_t$ and $\mathbf{v}_t$, and the time-to-go at the last guidance cycle $t_{\text{go}} = t_f - t_{\text{guid}}$, assumed to be known. Here it is noted that in order for these integrals to be valid, all vectors must be expressed in a non-rotating inertial frame.

The solution [4] to this linear system is

$$\begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix} = \begin{bmatrix} -\frac{2}{t_{\text{go}}}\mathbf{I}_3 & \frac{6}{t_{\text{go}}^2}\mathbf{I}_3 \\ \frac{6}{t_{\text{go}}^2}\mathbf{I}_3 & -\frac{12}{t_{\text{go}}^3}\mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{v}_t - \mathbf{v}_0 \\ \mathbf{r}_t - \mathbf{r}_0 - \mathbf{v}_0 t_{\text{go}} \end{bmatrix} \tag{2.10}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix. This formulation allows for coefficient updates to occur at lower frequency than thrust commands, perhaps with updates coinciding with navigation updates. As with the Apollo guidance, coefficients tend to infinity as $t_{\text{go}} \to 0$. This may be avoided by not recalculating coefficients during the final seconds, called the *fine count* [4,9].

The coefficients for the quadratic acceleration profile, Eq. (2.9) may be solved for in the same manner. Since there are 3 unknowns (in each direction), an additional boundary condition must be imposed, specifically the target acceleration vector $\mathbf{a}_t$. The solution [4] is

$$\begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & -\frac{6}{t_{\text{go}}}\mathbf{I}_3 & \frac{12}{t_{\text{go}}^2}\mathbf{I}_3 \\ -\frac{6}{t_{\text{go}}}\mathbf{I}_3 & \frac{30}{t_{\text{go}}^2}\mathbf{I}_3 & -\frac{48}{t_{\text{go}}^3}\mathbf{I}_3 \\ \frac{6}{t_{\text{go}}^2}\mathbf{I}_3 & -\frac{24}{t_{\text{go}}^3}\mathbf{I}_3 & \frac{36}{t_{\text{go}}^4}\mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{v}_t - \mathbf{v}_0 \\ \mathbf{r}_t - \mathbf{r}_0 - \mathbf{v}_0 t_{\text{go}} \end{bmatrix}. \tag{2.11}$$

As the time between coefficient updates limits to zero ($t \to 0$), both guidance Eqs. (2.8) and (2.9) simplify to $\mathbf{a} = \mathbf{c}_0$. For the linear acceleration profile, the acceleration command reduces to

$$\begin{aligned} \mathbf{a} &= -\frac{2}{t_{\text{go}}}(\mathbf{v}_t - \mathbf{v}_0) + \frac{6}{t_{\text{go}}^2}\left(\mathbf{r}_t - \mathbf{r}_0 - \mathbf{v}_0 t_{\text{go}}\right) \\ &= \frac{6}{t_{\text{go}}^2}(\mathbf{r}_t - \mathbf{r}_0) - \frac{2}{t_{\text{go}}}(\mathbf{v}_t + 2\mathbf{v}_0) \end{aligned} \tag{2.12}$$

For the quadratic acceleration profile, the acceleration becomes

$$\begin{aligned} \mathbf{a} &= \mathbf{a}_t - \frac{6}{t_{\text{go}}}(\mathbf{v}_t - \mathbf{v}_0) + \frac{12}{t_{\text{go}}^2}\left(\mathbf{r}_t - \mathbf{r}_0 - \mathbf{v}_0 t_{\text{go}}\right) \\ &= \frac{12}{t_{\text{go}}^2}(\mathbf{r}_t - \mathbf{r}_0) - \frac{6}{t_{\text{go}}}(\mathbf{v}_t + \mathbf{v}_0) + \mathbf{a}_t \end{aligned} \tag{2.13}$$

which we see is identical to Klumpp's explicit guidance, Eq. (2.3), when we note that $t_{\text{go}} = -T$. As would be expected, this quadratic acceleration profile from Sostaric produces the same quartic trajectory from which Klumpp derives his equations.

During the final (or terminal) descent phase, NASA engineers have proposed that the vehicle descends vertically at a small, constant velocity. As such a guidance law is not necessarily required, since a proportional feedback law will perform well. Where x and y are horizontal components and z is the vertical component, the commanded accelerations are

$$\begin{aligned} a_z &= a_{t,z} - K_{vz}\left(v_z - v_{t,z}\right) \\ a_x &= -K_{vx}v_x - K_{rx}\left(r_x - r_{t,x}\right) \\ a_y &= -K_{vy}v_y - K_{ry}\left(r_y - r_{t,y}\right) \end{aligned} \tag{2.14}$$

$$\text{where} \begin{cases} v_z,\ v_x,\ v_y,\ r_x,\ r_y & \text{are the vehicle states} \\ a_{t,z},\ v_{t,z},\ r_{t,x},\ r_{t,y} & \text{are the targets} \\ K_{vz},\ K_{vx},\ K_{vy},\ K_{rx},\ K_{ry} & \text{are control gains.} \end{cases}$$

When measured within a target-fixed coordinate system, $r_{t,x}$ and $r_{t,y}$ are zero (unless a redesignation occurs). The horizontal velocity targets, which are always zero, have been excluded.

2.1.3 Optimal Guidance

An optimal guidance system is one that performs the "best" for a given measure of performance. In the context of rocket-propelled vehicles, the optimal system might be one that minimizes the flight time and consequently the fuel required, or minimizes the dispersions from a desired trajectory, or maximizes the altitude obtained, and so forth.

The planetary landing guidance problem may be conceptually reduced to a 2-D rendezvous problem for a continuously thrusting vehicle. The equations of motion for this system are

$$\dot{x} = u \qquad \dot{y} = v \qquad \dot{u} = a\cos\beta \qquad \dot{v} = a\sin\beta - g$$

where $a = a(t)$ is a known thrust-acceleration profile, $\beta = \beta(t)$ is the thrust-direction angle, and g is the acceleration of an assumed constant gravity field. It can be shown [3] that the time-minimizing guidance law for these equations of motion is the "bilinear tangent law"

$$\tan\beta = \frac{-c_2 t + c_4}{-c_1 t + c_3} \tag{2.15}$$

where c_i are constants dependent upon the initial and final boundary conditions of the problem and the acceleration profile.

A variant on Eq. (2.15) which allows for some unspecified final states is the "linear tangent law" [3, 10]

$$\tan\beta = \tan\beta_0 - ct. \tag{2.16}$$

This guidance law has been used successfully during certain ascent phases of the Saturn V rocket [11] and the Space Shuttle [12].

Of particular interest is the Space Shuttle's Powered Explicit Guidance (PEG) algorithm [12] which is used in all phases of exoatmospheric powered flight. The direction of the thrust vector is given in a vector form of the linear tangent law:

$$\boldsymbol{\lambda}_F = \boldsymbol{\lambda}_v + \dot{\boldsymbol{\lambda}}\,(t - t_\lambda) \tag{2.17}$$

where $\boldsymbol{\lambda}_F$ is a vector defining the commanded thrust direction, as a function of time; $\boldsymbol{\lambda}_v$ is a unit vector in the direction of the velocity-to-be-gained, $\dot{\boldsymbol{\lambda}}$ is a vector normal to $\boldsymbol{\lambda}_v$ representing the rate of change of $\boldsymbol{\lambda}_F$, and t_λ is a time chosen such that the total velocity change due to thrust is along $\boldsymbol{\lambda}_v$. Using a truncated series expansion of the unit vector $\mathbf{u}_F = \boldsymbol{\lambda}_F / \|\boldsymbol{\lambda}_F\|$, the equation of motion for the vehicle is given by

$$\ddot{\mathbf{r}} = \frac{F}{m}\left\{\boldsymbol{\lambda}_v\left[1 - \frac{1}{2}\dot{\lambda}^2(t - t_\lambda)^2\right] + \dot{\boldsymbol{\lambda}}\,(t - t_\lambda)\right\} + \mathbf{g}. \tag{2.18}$$

A complete derivation of the equations used for the algorithm is not given here, but is summarized. The PEG algorithm iteratively solves for the steering direction using a predictor-corrector method using the velocity-to-go or velocity-to-be-gained, $\mathbf{v}_{\mathrm{go}}$, as the independent variable. There are three basic steps in the program [12], outlined here.

1. Calculate the steering parameters and burn time based upon $\mathbf{v}_{\mathrm{go}}$.

   The current velocity-to-be-gained is estimated from previous values (or an initial nominal value) and reduced by integrating the acceleration sensed by the inertial instruments. The time-to-go, t_{go}, is calculated by numerically integrating the thrust magnitude time history, which is known *a priori*, until the magnitude of velocity-to-be-gained is met. Once t_{go} is known, other integrals of the thrust are numerically calculated. From these integrals, two of the steering parameters are calculated: $\boldsymbol{\lambda}_v$ and t_λ. The position change remaining, $\mathbf{r}_{\mathrm{go}}$ is calculated from the integrals of thrust and known steering parameters, and the known and desired states. The final steering parameter, $\dot{\boldsymbol{\lambda}}$, may then be calculated. Steering commands are sent to the control software.

2. Predict the terminal (cutoff) state.

   With all the steering parameters, the net change in velocity and position due to thrusting are calculated, $\mathbf{v}_{\mathrm{thrust}}$ and $\mathbf{r}_{\mathrm{thrust}}$. The predicted cutoff velocity and position are then calculated, $\mathbf{v}_p$ and $\mathbf{r}_p$, from the present state and the calculated changes due to thrusting and due to gravity.

3. Correct $\mathbf{v}_{\mathrm{go}}$ to null terminal state errors.

   The predicted cutoff state is compared to the desired cutoff state, based on the desired orbital plane normal, desired velocity magnitude, flight path angle, and altitude. The difference of the predicted velocity and the desired velocity is calculated, and subtracted from $\mathbf{v}_{\mathrm{go}}$ using a relaxation factor if desired. The process then repeats.

PEG has proven successful on dozens of Shuttle flights. A variant of PEG is currently under development for use on the Altair during the braking phase [9].

Other optimal planetary landing guidance schemes exists, including one developed by D'Souza [13]. The guidance law is designed to minimize a cost function that is both a function of the time required (free end time) and the magnitude of the acceleration,

$$J = \Gamma t_f + \frac{1}{2}\int_{t_0}^{t_f} \left(a_x^2 + a_y^2 + a_z^2\right) d\tau \tag{2.19}$$

where Γ is a weighting parameter on the final time. Constraining the terminal state to zero and using variational methods found in Bryson & Ho [3], the optimal acceleration is found to be

$$\mathbf{a} = -\frac{4\mathbf{v}}{t_{\text{go}}} - \frac{6\mathbf{r}}{t_{\text{go}}^2} - \mathbf{g} \tag{2.20}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the relative position and velocity vectors, respectively, and $\mathbf{g}$ is the constant gravitational acceleration vector. The time of flight t_{go} is a root of the polynomial

$$\left(\Gamma + \frac{g^2}{2}\right) t_{\text{go}}^4 - 2\mathbf{v}\cdot\mathbf{v}t_{\text{go}}^2 - 12\mathbf{v}\cdot\mathbf{r}t_{\text{go}} - 18\mathbf{r}\cdot\mathbf{r} = 0. \tag{2.21}$$

Note the similarity of Eq. (2.20) to the linear acceleration profile in Eq. (2.12). D'Souza accounts for gravity in the dynamics (constant "flat-earth" model) while Sostaric considers it in the total acceleration of the vehicle. When the target state is zero, the two guidance laws are actually identical. Without assuming a specific trajectory shape, D'Souza proves that the linear acceleration profile is an optimal guidance law. He also provides a way to calculate t_{go} when weighing the relative importance of fuel consumed (total acceleration) and time to land.

Application of guidance laws to this research is described in detail in Chapter 4.

2.2 Planetary Landing Navigation

Space navigation is the term uses for estimating a vehicle's dynamic state, that is position, velocity, attitude, and angular velocity, using sensors. Advanced vehicle navigation

Table 2.2: Continuous-Discrete Extended Kalman Filter Equations [16]

Dynamic Model	$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t) + G(t)\mathbf{w}(t);\ \mathbf{w}(t) \sim N(\mathbf{0},Q(t))$	(2.22)
Sensor Model	$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k;\ \mathbf{v}_k \sim N(\mathbf{0},R_k)$	(2.23)
Initialize State	$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$	(2.24)
Initialize Covariance	$P_0 = \mathrm{E}\left\{\tilde{\mathbf{x}}(t_0)\tilde{\mathbf{x}}^{\mathsf{T}}(t_0)\right\}$	(2.25)
Kalman Gain	$K_k = P_k^- H_k^{\mathsf{T}}(\hat{\mathbf{x}}_k^-)[H_k(\hat{\mathbf{x}}_k^-)P_k^- H_k^{\mathsf{T}}(\hat{\mathbf{x}}_k^-) + R_k]^{-1}$	(2.26)
Measurement Partial	$H_k(\hat{\mathbf{x}}_k^-) \equiv \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right\|_{\hat{\mathbf{x}}_k^-}$	(2.27)
State Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k[\tilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)]$	(2.28)
Covariance Update	$P_k^+ = [I - K_k H_k(\hat{\mathbf{x}}_k^-)]P_k^-$	(2.29)
State Propagation	$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t),\mathbf{u}(t),t)$	(2.30)
Covariance Propagation	$\dot{P}(t) = F(\hat{\mathbf{x}}(t),t)P(t) + P(t)F^{\mathsf{T}}(\hat{\mathbf{x}}(t),t) + G(t)Q(t)G^{\mathsf{T}}(t)$	(2.31)
Dynamic Partial	$F(\hat{\mathbf{x}}(t),t) \equiv \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right\|_{\hat{\mathbf{x}}(t)}$	(2.32)

systems may estimate additional states to provide higher accuracy. Optimal methods exist that can process measurements to estimate the state, and also quantify how well a state is estimated. Kalman filters [1,14], one type of optimal measurement filter, are frequently used in spacecraft for navigation. Space navigation has a noble history involving manned lunar missions, when Apollo astronauts manually took angle measurements to celestial bodies, and, with the on-board navigation computer running a Kalman filter, processed them (along with automatic radio measurements) to get a highly accurate estimate of the vehicle's true state [15].

A Kalman filter processes measurements sequentially in an optimal, least-squares fashion, to produce a state estimate $\hat{\mathbf{x}}$ and its covariance P. The derivation of various Kalman filters, for different applications, is given by Crassidis and Junkins [16]. The Continuous-Discrete Extended Kalman Filter is applicable for the dynamics and sensors used in planetary landing, and a summary of equations required for its use is given in

Table 2.2.

A Kalman filter requires equations modeling the dynamics of the system Eq. (2.22) as well as the sensor measurements Eq. (2.23). The filter also requires reasonable estimates of the uncertainty of the dynamic model (process noise strength, Q, and distribution, G) and sensors (instrument noise strength, R). The operation of a Kalman filter can be summarized in just a few steps.

1. Initialize state Eq. (2.24) and covariance Eq. (2.25), based on *a priori* stored information and/or initial measurements.
2. If a measurement is available,
   (a) Calculate the measurement partial derivative Eq. (2.27) and the Kalman gain Eq. (2.26).
   (b) Update the filter state (2.28) and the covariance (2.29). Note that another form of (2.29), called the Joseph formulation [14], exists which reduces the likelihood of P becoming asymmetric.
3. Propagate to the next time step.
   (a) Use the nonlinear dynamics to propagate the state (2.30), often with a Runge-Kutta ODE integrating method.
   (b) Calculate the values of the dynamics partial derivative Eq. (2.32), and optionally the state time-transition matrix $\Phi = e^{F\Delta t} \approx I + F\Delta t + \frac{1}{2!}FF\Delta t^2 + \frac{1}{3!}FFF\Delta t^3 + \ldots$ and discrete noise strength $Q_d = \int_{t_{k-1}}^{t_k} \Phi G Q G^{\mathsf{T}} \Phi^{\mathsf{T}}\, d\tau \approx G Q G^{\mathsf{T}} \Delta t$.
   (c) Propagate the covariance using the linearized dynamics and the Ricatti equation in Eq. (2.31), or its discrete form $P(t_k) = \Phi(t_{k-1}, t_k) P(t_{k-1}) \Phi^{\mathsf{T}}(t_{k-1}, t_k) + Q_d(t_{k-1}, t_k)$.
4. Return to step 2.

Measurements are an essential component for a Kalman filter, and for the filter to function correctly, accurate equations must be derived that model the actual operation of the sensor. Equations for common celestial navigation measurements have already been derived [2, 15, 17].

2.3 Linear Covariance Analysis

Linear Covariance Analysis, sometimes simply called *covariance analysis*, is a class of statistical analysis techniques separately formulated by Maybeck [1] and Battin [15]. When used for specific purposes, linear covariance is also known by other names. *Performance analysis* [1] or *statistical error analysis* [15] is using linear covariance techniques to produce "an accurate statistical portrayal of estimation errors committed by [a Kalman] filter in the 'real world' environment," that is to characterize the filter error process statistically. *Consider covariance analysis* [1, 18] is "a technique to assess the impact of neglecting to estimate [certain unknown or poorly known model parameters] on the accuracy of the state estimate." This type of analysis may be included in the larger design task called *filter tuning* [1], which may also include optimizing measurement selection and scheduling [15]. *Sensitivity analysis* [1, 15] is the production of an "error budget" that "consists of repeated covariance analyses in which the error sources (or small groups of sources) in the truth model are 'turned on' individually to determine the separate effects of these sources." In general, covariance analysis provides an effective means of conducting a *tradeoff analysis* among various proposed designs [1].

An extensive LinCov tool that includes both trajectory control errors and navigation errors has been previously developed for the application of orbital rendezvous error and dispersion analysis by Geller [2,17]. This tool, written for MATLAB, has also been adapted for the present research on the Altair GN&C, with which inertial navigation sensitivity analyses and a terrain relative navigation (TRN) tradeoff analysis have been performed [19]. Geller's theory, adapted from Battin and Maybeck, provides the background for present research in which his tool is augmented as described in Chapters 2 and 4.

In GN&C analysis, it is common to compare the time-history of the simulated "truth" state, $\mathbf{x}$, to that of a nominal or reference trajectory, $\bar{\mathbf{x}}$. The difference between the two is a common performance metric called the *true dispersion*, $\delta\mathbf{x} \equiv \mathbf{x} - \bar{\mathbf{x}}$. Another important measure is the *navigation error*, how well the navigation filter estimates the true state, $\delta\mathbf{e} \equiv \hat{\mathbf{x}} - \mathbf{x}$. Often a less import performance measure, the *navigation dispersion*, $\delta\hat{\mathbf{x}} \equiv \hat{\mathbf{x}} - \bar{\mathbf{x}}$, is also used. It is easily shown that the navigation error may be expressed in terms of the two dispersions, $\delta\mathbf{e} = \delta\hat{\mathbf{x}} - \delta\mathbf{x}$. Often, the navigation filter does not estimate all of the "truth" states, so a selection coefficient may be introduced, $\delta\mathbf{e} = \delta\hat{\mathbf{x}} - M_x\delta\mathbf{x}$ [1,2].

A single run of a Monte Carlo analysis will produce a single time history of the dispersions and errors, so the results of many runs are needed to produce a statistical measure of GN&C system's performance. The covariance, or statistical spread, of the dispersions and navigation error, as calculated by a Monte Carlo program using the results of of N simulations, may be found though [2]

$$D_{\text{true}} \approx \frac{1}{N-1}\sum_{i=1}^{N}\delta\mathbf{x}\delta\mathbf{x}^{\mathsf{T}}, \quad D_{\text{nav}} \approx \frac{1}{N-1}\sum_{i=1}^{N}\delta\hat{\mathbf{x}}\delta\hat{\mathbf{x}}^{\mathsf{T}}, \quad P_{\text{true}} \approx \frac{1}{N-1}\sum_{i=1}^{N}\delta\mathbf{e}\delta\mathbf{e}^{\mathsf{T}}. \tag{2.33}$$

Often specific components of these matrices are of primary interest, such as trajectory dispersions, navigation errors, or fuel usage.

Linear covariance analysis generates the same covariances in Eq. (2.33) in a single simulation by directly propagating, updating, and correcting an augmented state covariance matrix that encompasses both the true and navigation covariances [2]. Geller presents a thorough derivation of the linear covariance equations, a portion of which follows.

2.3.1 Nonlinear Equations

As with a Kalman filter, dynamic and instrument models must be formulated before coding a LinCov tool. Also separate truth and filter models must be considered, although these may be the same initially. The formulation by Geller allow for several types of sensors and actuators, including continuous inertial measurements, $\tilde{\mathbf{y}}$, discrete inertial measurements, $\Delta\tilde{\mathbf{y}}_j$, discrete non-inertial measurements, $\tilde{\mathbf{z}}_k$, continuous actuator commands, $\hat{\mathbf{u}}$, and discrete (instantaneous/impulsive) actuator commands, $\Delta\hat{\mathbf{u}}_j$. The nonlinear truth equations take the generic form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\, \hat{\mathbf{u}},\, t) + \mathbf{w} \qquad \mathrm{E}[\mathbf{w}(t)\mathbf{w}^{\mathsf{T}}(t')] = S_w(t)\delta(t-t') \tag{2.34}$$

$$\tilde{\mathbf{y}} = \mathbf{c}(\mathbf{x},\, t) + \boldsymbol{\eta} \qquad \mathrm{E}[\boldsymbol{\eta}(t)\boldsymbol{\eta}^{\mathsf{T}}(t')] = S_\eta(t)\delta(t-t') \tag{2.35}$$

$$\Delta\tilde{\mathbf{y}}_j = \Delta\mathbf{c}(\mathbf{x}_j,\, t_j) + \Delta\boldsymbol{\eta}_j \qquad \mathrm{E}[\Delta\boldsymbol{\eta}_j\Delta\boldsymbol{\eta}_{j'}^{\mathsf{T}}] = S_{\Delta\eta}(t_j)\delta_{jj'} \tag{2.36}$$

$$\tilde{\mathbf{z}}_k = \mathbf{h}(\mathbf{x}_k,\, t_k) + \boldsymbol{\nu}_k \qquad \mathrm{E}[\boldsymbol{\nu}_k\boldsymbol{\nu}_{k'}^{\mathsf{T}}] = R_\nu(t_k)\delta_{kk'} \tag{2.37}$$

where $\mathbf{w}$, $\boldsymbol{\eta}$, $\Delta\boldsymbol{\eta}_j$, $\boldsymbol{\nu}_k$ are white noise errors with covariances as shown. An instantaneous correction to the state vector may be applied at time t_j with

$$\mathbf{x}_j^{+c} = \mathbf{x}_j^{-c} + \mathbf{d}(\mathbf{x}_j^{-c},\, \Delta\hat{\mathbf{u}}_j,\, t_j) + \Delta\mathbf{w}_j \qquad \mathrm{E}[\Delta\mathbf{w}_j\Delta\mathbf{w}_{j'}^{\mathsf{T}}] = S_{\Delta w}(t_j)\delta_{jj'}. \tag{2.38}$$

The nonlinear filter equations are

$$\dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}(\hat{\mathbf{x}},\, \hat{\mathbf{u}},\, \tilde{\mathbf{y}},\, t) \tag{2.39}$$

$$\dot{\hat{P}} = \left[\hat{F}_{\hat{x}} + \hat{F}_{\tilde{y}}\hat{C}_{\hat{x}}\right]\hat{P} + \hat{P}\left[\hat{F}_{\hat{x}} + \hat{F}_{\tilde{y}}\hat{C}_{\hat{x}}\right]^{\mathsf{T}} + \hat{F}_{\tilde{y}}\hat{S}_\eta\hat{F}_{\tilde{y}}^{\mathsf{T}} + \hat{S}_w \tag{2.40}$$

$$\hat{\mathbf{u}} = \hat{\mathbf{g}}(\hat{\mathbf{x}},\, t) \tag{2.41}$$

$$\Delta\hat{\mathbf{u}} = \Delta\hat{\mathbf{g}}(\hat{\mathbf{x}}_j^{-c},\, t_j) \tag{2.42}$$

where uppercase characters denote partial derivatives taken with respect to the variable indicated by the subscript evaluated at the nominal condition, i.e. $\hat{F}_{\hat{x}} \equiv \partial\hat{\mathbf{f}}/\partial\hat{\mathbf{x}}\Big|_{\bar{\mathbf{x}}}$. $\hat{P}$ represents the flight computer's estimate of the navigation state error covariance matrix, which is propagated using the Ricatti equation [16]. Note that while the filter state is propagated using the true inertial measurement model ($\tilde{\mathbf{y}} = \mathbf{c}(\mathbf{x}, t) + \boldsymbol{\eta}$), its covariance must be propagated with the filter's model ($\hat{\mathbf{y}} = \hat{\mathbf{c}}(\hat{\mathbf{x}}, t) + \hat{\boldsymbol{\eta}}$). Naturally, the filter does not actually add noise ($\hat{\boldsymbol{\eta}}$) to the measurements, but this term is included in the model as reminder to include the filter noise strength estimate ($\hat{S}_\eta$) to the covariance propagation equation.

A standard or extended Kalman filter implementation may be used to *update* the state and covariance matrix, with

$$
\begin{aligned}
\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \hat{K}(t_k)[\hat{\mathbf{z}}_k - \hat{\mathbf{h}}(\hat{\mathbf{x}}_k,\, t_k)] && (2.43)\\
\hat{P}^+(t_k) &= [\mathbf{I} - \hat{K}(t_k)\hat{H}_{\hat{x}}(t_k)]\hat{P}^-(t_k)[\mathbf{I} - \hat{K}(t_k)\hat{H}_{\hat{x}}(t_k)]^\mathsf{T} + \hat{K}(t_k)\hat{R}_v(t_k)\hat{K}^\mathsf{T}(t_k) && (2.44)
\end{aligned}
$$

using the Joseph formulation [1] for the covariance update, and the Kalman gain is calculated by

$$
\hat{K}(t_k) = \hat{P}(t_k)\hat{H}_{\hat{x}}^\mathsf{T}(t_k)\left[\hat{H}_{\hat{x}}(t_k)\hat{P}(t_k)\hat{H}_{\hat{x}}^\mathsf{T} + \hat{R}_\nu(t_k)\right]^{-1} \tag{2.45}
$$

where $\hat{H}_{\hat{x}} \equiv \partial\hat{\mathbf{h}}/\partial\hat{\mathbf{x}}|_{\bar{\mathbf{x}}_k^-}$ is the measurement sensitivity matrix and $\hat{R}_\nu$ is the measurement noise covariance. Note that set of update equations are a restatement of Eqs. (2.26)-(2.32) for the Continuous-Discrete Extended Kalman Filter, in Table 2.2 on page 16.

In addition to incorporating continuous and discrete measurements with the filter, additional equations are also given for discrete corrections to the filter state and covariance. This filter correction technique should be used whenever an instantaneous actuator command occurs, but could be used in other ways. The correction algorithm is

$$
\begin{aligned}
\hat{\mathbf{x}}_j^{+c} &= \hat{\mathbf{x}}_j^{-c} + \hat{\mathbf{d}}(\hat{\mathbf{x}}^{-c},\, \Delta\hat{\mathbf{u}}_j,\, \Delta\tilde{\mathbf{y}}_j,\, t_j) && (2.46)\\
\hat{P}^{+c}(t_j) &= \left[\mathbf{I} + \hat{D}_{\hat{x}}(t_j) + \hat{D}_{\Delta\tilde{y}}(t_j)\Delta\hat{C}_{\hat{x}}(t_j)\right]\hat{P}^{-c}(t_j)\left[\mathbf{I} + \hat{D}_{\hat{x}}(t_j) + \hat{D}_{\Delta\tilde{y}}(t_j)\Delta\hat{C}_{\hat{x}}(t_j)\right]^\mathsf{T}\ldots\\
&\quad\cdots + \hat{D}_{\Delta\tilde{y}}(t_j)\hat{S}_{\Delta\eta}\hat{D}_{\Delta\tilde{y}}(t_j)^\mathsf{T} + \hat{S}_{\Delta w}(t_j) && (2.47)
\end{aligned}
$$

where $\hat{S}_{\Delta\eta}$ is the inertial measurement noise variance, $\hat{S}_{\Delta w}$ is the correction application noise variance, and other matrices are partial derivatives following the pattern above.

2.3.2 Linearized Equations

The nonlinear equations for the truth and filter are linearized about a reference trajectory denoted $\bar{\mathbf{x}}(t)$ using a truncated Taylor's series expansion, such that

$$\begin{aligned}\mathbf{f}(\mathbf{x},\, \hat{\mathbf{u}},\, t) &\approx \bar{\mathbf{f}}(\bar{\mathbf{x}},\, \hat{\mathbf{u}},\, t) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\bar{\mathbf{x}}}(\mathbf{x} - \bar{\mathbf{x}}) + \frac{\partial \mathbf{f}}{\partial \hat{\mathbf{u}}}|_{\bar{\mathbf{x}}}(\hat{\mathbf{u}} - \bar{\mathbf{u}}) \\ \delta \mathbf{f} &\approx F_x \delta \mathbf{x} + F_{\hat{u}} \delta \mathbf{u}.\end{aligned}$$

Thus, from Eqs. (2.34) and (2.39), the linearized truth and filter dynamic models become

$$\begin{aligned}\delta \dot{\mathbf{x}} &= F_x \delta \mathbf{x} + F_{\hat{u}} \delta \hat{\mathbf{u}} + \mathbf{w} \\ &= F_x \delta \mathbf{x} + F_{\hat{u}} \hat{G}_{\hat{x}} \delta \hat{\mathbf{x}} + \mathbf{w} \qquad (2.48) \\ \delta \dot{\hat{\mathbf{x}}} &= \hat{F}_{\hat{x}} \delta \hat{\mathbf{x}} + \hat{F}_{\hat{u}} \delta \hat{\mathbf{u}} + \hat{F}_{\tilde{y}} \delta \tilde{\mathbf{y}} \\ &= (\hat{F}_{\hat{x}} + \hat{F}_{\hat{u}} \hat{G}_{\hat{x}}) \delta \hat{\mathbf{x}} + \hat{F}_{\tilde{y}} C_x \delta \mathbf{x} + \hat{F}_{\tilde{y}} \boldsymbol{\eta}. \qquad (2.49)\end{aligned}$$

with updates, from Eq. (2.28), taking the form

$$\delta \mathbf{x}_k^+ = \delta \mathbf{x}_k^- \qquad (2.50)$$

$$\delta \hat{\mathbf{x}}_k^+ = \hat{K}(t_k) H_x(t_k) \delta \mathbf{x}_k^- + [I - \hat{K}(t_k) \hat{H}_{\hat{x}}(t_k)] \delta \hat{\mathbf{x}}_k^- + \hat{K}(t_k) \boldsymbol{\nu}_k. \qquad (2.51)$$

Note that the true dispersions $\delta \mathbf{x}$ are unaffected by measurements—actuators are required to correct dispersions. Geller also linearizes Eq. (2.46) for linearized discrete truth and filter state corrections, which are given by

$$\begin{aligned}\delta \mathbf{x}_j^{+c} &= [\mathbf{I} + D_x(t_j)] \delta \mathbf{x}_j^{-c} + D_{\Delta \hat{u}}(t_j) \Delta \hat{G}_x(t_j) \delta \hat{\mathbf{x}}_j^{-c} + \Delta \mathbf{w}_j \qquad (2.52) \\ \delta \hat{\mathbf{x}}_j^{+c} &= [\mathbf{I} + \hat{D}_{\hat{x}}(t_j) + \hat{D}_{\Delta \hat{u}}(t_j) \Delta \hat{G}_x(t_j)] \delta \hat{\mathbf{x}}_j^{-c} + \hat{D}_{\Delta \tilde{y}}(t_j) \Delta C_x(t_j) \delta \mathbf{x}_j^{-c} \dots \\ &\quad \dots + \hat{D}_{\Delta \tilde{y}}(t_j) \Delta \boldsymbol{\eta}_j. \qquad (2.53)\end{aligned}$$

2.3.3 Augmented State and Covariance Equations

Since the true and navigation dispersions are strongly coupled in the linearized equations of motion and update equations, there is a strong correlation (covariance) between these states. It is then not unreasonable to join the true and filter states into an augmented state

$$\mathbf{X} = \begin{bmatrix} \delta\mathbf{x} \\ \delta\hat{\mathbf{x}} \end{bmatrix} \tag{2.54}$$

with an augmented covariance matrix $\mathcal{C}_A(t) = E[\mathbf{X}(t)\mathbf{X}^{\mathsf{T}}(t)]$. In terms of the augmented state, Eqs. (2.48)-(2.53) may be rewritten

$$\dot{\mathbf{X}} = \mathcal{F}\mathbf{X} + \mathcal{G}\boldsymbol{\eta} + \mathcal{W}\mathbf{w} \tag{2.55}$$

$$\mathbf{X}_k^+ = \mathcal{A}_k\mathbf{X}_k^- + \mathcal{B}_k\boldsymbol{\nu}_k \tag{2.56}$$

$$\mathbf{X}_j^{+c} = \mathcal{D}_j\mathbf{X}_j^{-c} + \mathcal{M}_j\Delta\boldsymbol{\eta}_j + \mathcal{N}_j\Delta\mathbf{w}_j, \qquad \text{where} \tag{2.57}$$

$$\mathcal{F} = \begin{bmatrix} F_x & F_{\hat{u}}\hat{G}_{\hat{x}} \\ \hat{F}_{\tilde{y}}C_x & \hat{F}_{\hat{x}} + \hat{F}_{\hat{u}}\hat{G}_{\hat{x}} \end{bmatrix}, \qquad \mathcal{G} = \begin{bmatrix} \mathbf{0}_{n\times n_y} \\ \hat{F}_{\tilde{y}} \end{bmatrix}, \qquad \mathcal{W} = \begin{bmatrix} \mathbf{I}_{n\times n} \\ \mathbf{0}_{\hat{n}\times n} \end{bmatrix},$$

$$\mathcal{A}_k = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times\hat{n}} \\ \hat{K}(t_k)H_x(t_k) & \mathbf{I}_{\hat{n}\times\hat{n}} - \hat{K}(t_k)\hat{H}_{\hat{x}} \end{bmatrix}, \qquad \mathcal{B}_k = \begin{bmatrix} \mathbf{0}_{n\times n_z} \\ \hat{K}(t_k) \end{bmatrix},$$

$$\mathcal{D}_j = \begin{bmatrix} \mathbf{I}_{n\times n} + D_x(t_j) & D_{\Delta\hat{u}}(t_j)\Delta\hat{G}_{\hat{x}}(t_j) \\ \hat{D}_{\Delta\tilde{y}}(t_j)\Delta C_x(t_j) & \mathbf{I}_{\hat{n}\times\hat{n}} + \hat{D}_{\hat{x}}(t_j) + \hat{D}_{\Delta\hat{u}}(t_j)\Delta\hat{G}_{\hat{x}}(t_j) \end{bmatrix},$$

$$\mathcal{M}_j = \begin{bmatrix} \mathbf{0}_{n\times n_{\Delta y}} \\ \hat{D}_{\Delta\tilde{y}}(t_j) \end{bmatrix}, \qquad \mathcal{N}_j = \begin{bmatrix} \mathbf{I}_{n\times n} \\ \mathbf{0}_{\hat{n}\times n} \end{bmatrix}.$$

From the linear system above, it has been shown [1,2,15,16] the dynamic equation, update equation, and correction equation of the augmented covariance matrix $\mathcal{C}_A$ are

$$\begin{aligned}
\dot{\mathcal{C}}_A &= \mathcal{F}\mathcal{C}_A + \mathcal{C}_A\mathcal{F}^\mathsf{T} + \mathcal{G}S_\eta\mathcal{G}^\mathsf{T} + \mathcal{W}S_w\mathcal{W}^\mathsf{T} && (2.58)\\
\mathcal{C}_A(t_k^+) &= \mathcal{A}_k\mathcal{C}_A(t_k^-)\mathcal{A}_k^\mathsf{T} + \mathcal{B}_kR_v(t_k)\mathcal{B}_k^\mathsf{T}. && (2.59)\\
\mathcal{C}_A(t_j^{+c}) &= \mathcal{D}_j\mathcal{C}_A(t_j^{-c})\mathcal{D}_j^\mathsf{T} + \mathcal{M}_jS_{\Delta\eta}(t_j)\mathcal{M}_j^\mathsf{T} + \mathcal{N}_jS_{\Delta w}(t_j)\mathcal{N}_j^\mathsf{T} && (2.60)
\end{aligned}$$

When the flight computer covariance equations, Eqs. (2.40), (2.26), and (2.29), are appended to Eqs. (2.58), (2.59), and (2.60), a complete set of linear covariance analysis equations is formed [2].

2.3.4 Performance Evaluation

Geller presents a way to evaluate the performance of the GN&C system by using the augmented covariance matrix and separate filter covariance matrix. In standard Monte Carlo analysis, the covariance of the dispersions is found through Eq. (2.33). To find them, we must extract the true and navigation dispersion covariance matrices from $\mathcal{C}_A$. This may be done simply with

$$D_\text{true}(t) = \mathrm{E}\left[\delta\mathbf{x}(t)\delta\mathbf{x}^\mathsf{T}(t)\right] = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times\hat{n}} \end{bmatrix} \mathcal{C}_A \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times\hat{n}} \end{bmatrix}^\mathsf{T} \quad (2.61)$$

$$D_\text{nav}(t) = \mathrm{E}\left[\delta\hat{\mathbf{x}}(t)\delta\hat{\mathbf{x}}^\mathsf{T}(t)\right] = \begin{bmatrix} \mathbf{0}_{\hat{n}\times n} & \mathbf{I}_{\hat{n}\times\hat{n}} \end{bmatrix} \mathcal{C}_A \begin{bmatrix} \mathbf{0}_{\hat{n}\times n} & \mathbf{I}_{\hat{n}\times\hat{n}} \end{bmatrix}^\mathsf{T}, \quad (2.62)$$

and the covariance and the true navigation state *errors* is found by

$$\begin{aligned}
P_\text{true}(t) &= \mathrm{E}\left[\{\delta\hat{\mathbf{x}}(t) - M_x\delta\mathbf{x}(t)\}\{\delta\hat{\mathbf{x}}(t) - M_x\delta\mathbf{x}(t)\}^\mathsf{T}\right]\\
&= \begin{bmatrix} -M_x & \mathbf{I}_{\hat{n}\times\hat{n}} \end{bmatrix} \mathcal{C}_A \begin{bmatrix} -M_x & \mathbf{I}_{\hat{n}\times\hat{n}} \end{bmatrix}^\mathsf{T}. \quad (2.63)
\end{aligned}$$

The true navigation error covariance P_true can then be compared to the filter navigation error covariance $\hat{P}$ to evaluate the performance of the onboard navigation system.

The closed-loop guidance and control system performance may be evaluated by examining D_{true}.

Geller's original formulation, as applied for orbital rendezvous, has been modified slightly for previous research [19], and further modified for the current research as described in Chapter 4.

2.4 Maneuver Trigger Equations

Because linear covariance analysis requires a reference trajectory, implementing what would be, in actual flight or a Monte Carlo simulation, a time-varying maneuver is not apparent at first. For instance, during lunar landing, it may be (and is) desirable to begin powered descent, not at a specific clock time, but at a certain range to the landing site. The flight computer would calculate the great circle distance from the lander's subsatellite point to the landing site based on its state estimate, $\hat{\mathbf{x}}$. If that distance is found to be less than a predetermined value, the engines are commanded to ignite and PDI begins. In a Monte Carlo analysis, a few lines of code can perform that task, and even though each run may have the lander arriving at the designated range at a different time (and different altitude), they all perform the maneuver based on that range trigger.

In linear covariance analysis, a reference trajectory is used which has only one time of maneuver execution. Some additional thought is required to implement the flight computer logic for LinCov analysis. We may expect that the true trajectory dispersions will be influenced by the navigation errors at the trigger condition, and this is the case as we shall see in Chapter 6.

Gossner [20] has developed linear covariance techniques that solve a very similar problem, specifically varying the time of flight of a rendezvous maneuver where the elevation angle from the chaser vehicle's horizontal to the target vehicle is used to trigger a maneuver burn. He defines an "arbitrary scalar maneuver condition," ξ, which is a function of the terminal state, although by extension ξ may be a function of solely the filter state at any time,

$$\xi = \xi(\hat{\mathbf{x}}(t_{\xi})). \tag{2.64}$$

Gossner also allows for a “time slip” or time dispersion to be added to the state, δt_s. His primary analytic objective is the calculation of a condition transition matrix Φ_C that propagates the covariance matrix, E in his notation, to an arbitrary condition at which the maneuver occurs

$$E_C = \Phi_C E_i \Phi_C^{\mathsf{T}}. \tag{2.65}$$

The maneuver condition ξ is used to constrain the transition matrix by requiring the condition’s total derivative to be zero, that is

$$\delta\xi = \frac{\partial\xi}{\partial x_f}\delta x_f = \mathbf{k}^{\mathsf{T}}\delta x_f = 0. \tag{2.66}$$

The final dispersion is expressed in terms of the initial dispersion and the time slip

$$\delta x_f = \Phi_T \delta x_i + \dot{x}_f \delta t_s, \tag{2.67}$$

where Φ_T is the time transition matrix. Solving Eqs. (2.66) and (2.67), the time slip can be expressed

$$\delta t_s = \frac{-\mathbf{k}^{\mathsf{T}}\Phi_T\delta x_i}{\mathbf{k}^{\mathsf{T}}\dot{x}_f} \tag{2.68}$$

and the condition state

$$\begin{aligned}
\delta x_C &= \Phi_T\delta x_i - \frac{\dot{x}_f\mathbf{k}^{\mathsf{T}}}{\dot{x}_f^{\mathsf{T}}\mathbf{k}}\Phi_T\delta x_i && (2.69)\\
&= \left[\mathbf{I} - \frac{\dot{x}_f\mathbf{k}^{\mathsf{T}}}{\dot{x}_f^{\mathsf{T}}\mathbf{k}}\right]\Phi_T\delta x_i && (2.70)\\
&= \Phi_I\Phi_T\delta x_i, && (2.71)
\end{aligned}$$

where the desired condition transition matrix is $\Phi_C = \Phi_I \Phi_T$. Note that Φ_I is an idempotent "shaping" matrix that has the property

$$\begin{aligned}
\Phi_I^2 = \Phi_I \Phi_I = \left[\mathbf{I} - \frac{\dot{x}_f \mathbf{k}^\mathsf{T}}{\dot{x}_f^\mathsf{T} \mathbf{k}}\right] \left[\mathbf{I} - \frac{\dot{x}_f \mathbf{k}^\mathsf{T}}{\dot{x}_f^\mathsf{T} \mathbf{k}}\right] &= \mathbf{I} - 2\frac{\dot{x}_f \mathbf{k}^\mathsf{T}}{\dot{x}_f^\mathsf{T} \mathbf{k}} + \frac{\dot{x}_f \mathbf{k}^\mathsf{T} \dot{x}_f \mathbf{k}^\mathsf{T}}{\left(\dot{x}_f^\mathsf{T} \mathbf{k}\right)^2} \\
&= \mathbf{I} - 2\frac{\dot{x}_f \mathbf{k}^\mathsf{T}}{\dot{x}_f^\mathsf{T} \mathbf{k}} + \frac{\dot{x}_f \left(\dot{x}_f^\mathsf{T} \mathbf{k}\right) \mathbf{k}^\mathsf{T}}{\left(\dot{x}_f^\mathsf{T} \mathbf{k}\right)^2} = \mathbf{I} - 2\frac{\dot{x}_f \mathbf{k}^\mathsf{T}}{\dot{x}_f^\mathsf{T} \mathbf{k}} + \frac{\dot{x}_f \mathbf{k}^\mathsf{T}}{\dot{x}_f^\mathsf{T} \mathbf{k}} \\
\Phi_I^2 = \mathbf{I} - \frac{\dot{x}_f \mathbf{k}^\mathsf{T}}{\dot{x}_f^\mathsf{T} \mathbf{k}} &= \Phi_I. \qquad (2.72)
\end{aligned}$$

Gossner also notes that the idempotent matrix can be used to shape a covariance matrix that has already been propagated to the nominal final time. From Eq. (2.65),

$$\begin{aligned}
E_C &= \Phi_C E_i \Phi_C^\mathsf{T} = \Phi_I \Phi_T E_i \Phi_T^\mathsf{T} \Phi_I^\mathsf{T} \\
E_C &= \Phi_I E_f \Phi_I^\mathsf{T}. \qquad (2.73)
\end{aligned}$$

Gossner's specific application requires a constraint on the final state, although there is no reason why such a shaping matrix could not be applied at any point on the trajectory where a "time slip" in an event may be required. The topic of scalar maneuver triggers is explored further and applied to the current research in Chapter 6.

Chapter 3

Dynamic Formulation

The LinCov tool developed by Geller was originally designed for earth-orbit rendezvous of a “chaser” with a “object” vehicle [2,17,21]. Since then, its capability has been expanded for lunar landing navigation [19]. Consequent of these enhancements, some of the states required for rendezvous are no longer needed, and additional states have been added. A list of the LinCov states is given in Table 3.1.

The LinCov tool was designed, among other things, to explore the sensitivity of the GN&C’s performance to instrument errors. In addition to white noise, the instruments were modeled with bias, scale factor, and misalignment errors all of which could be time-varying. The same error sources were also applied to actuators. To allow for all these time-varying error sources, states were designated for them with their own dynamics. Because all of these error sources are designed to be included in the navigation filter, the vehicle GN&C can use its instrument measurements to estimate and cancel out these error sources.

In order to properly develop a linear covariance tool to analyze a dynamic problem, a complete set of equations modeling the dynamic problem must be formulated. Linear covariance required *linearized* equations of motion, obtained through first-order Taylor’s series expansions of the nonlinear equations of motion. This chapter details the nonlinear dynamic problem, such as would be used for Monte Carlo analysis, as well as the linearized equations. The equations presented here comprise specific elements of the general state vectors described in Sections 2.3.1 and 2.3.2.

3.1 Truth State Dynamics

This section details dynamic formulation used for the true landing site and vehicle

Table 3.1: List of Nonlinear, Nominal, and Linearized States

Nonlinear†	Nom.	Linear†	State Description	Count
Landing site states, $\mathbf{x}_{\text{site}}$ $(\delta\mathbf{x}_{\text{site}})$				*12*
${}^{i}\mathbf{r}_{\text{site}}$	${}^{i}\bar{\mathbf{r}}_{\text{site}}$	${}^{i}\delta\mathbf{r}_{\text{site}}$	Site inertial position	3
$\mathbf{e}_{\text{map}}$	$\mathbf{0}_{3\times1}$	$\mathbf{e}_{\text{map}}$	Map error	3
(unused)			Site attitude	3
(unused)			Site angular velocity	3
Vehicle states, $\mathbf{x}_{\text{veh}}$ $(\delta\mathbf{x}_{\text{veh}})$				*12*
${}^{i}\mathbf{r}$	${}^{i}\bar{\mathbf{r}}$	${}^{i}\delta\mathbf{r}$	Vehicle inertial position	3
${}^{i}\mathbf{v}$	${}^{i}\bar{\mathbf{v}}$	${}^{i}\delta\mathbf{v}$	Vehicle inertial velocity	3
${}^{b}_{i}\mathbf{q}$	${}^{b}_{i}\bar{\mathbf{q}}$	${}^{b}\delta\boldsymbol{\theta}$	Vehicle attitude	3
${}^{b}\boldsymbol{\omega}$	${}^{b}\bar{\boldsymbol{\omega}}$	${}^{b}\delta\boldsymbol{\omega}$	Vehicle angular velocity	3
Parameter states, $\mathbf{p}$				*75*
$\xi_{\text{gyro}} = \begin{bmatrix} \mathbf{s}_{\text{gyro}} \\ \epsilon_{\text{gyro}} \\ \mathbf{b}_{\text{gyro}} \end{bmatrix}$	$\mathbf{0}_{9\times1}$	ξ_{gyro}	Gyro errors (scale, alignment, bias)	9
$\epsilon_{\text{starcam}}$	$\mathbf{0}_{3\times1}$	$\epsilon_{\text{starcam}}$	Star camera errors (alignment)	3
(unused)			Optical feature tracker (alignment)	3
$\xi_{\text{torque}} = \begin{bmatrix} \mathbf{s}_{\text{torque}} \\ \epsilon_{\text{torque}} \\ \mathbf{b}_{\text{torque}} \end{bmatrix}$	$\mathbf{0}_{9\times1}$	ξ_{torque}	Torquer errors (scale, alignment, bias)	9
(unused)			Impulsive ΔV errors	9
$\xi_{\text{thr}} = \begin{bmatrix} \mathbf{s}_{\text{thr}} \\ \epsilon_{\text{thr}} \\ \mathbf{b}_{\text{thr}} \end{bmatrix}$	$\mathbf{0}_{9\times1}$	ξ_{thr}	Continuous thrust errors (scale, alignment, bias)	9
$\xi_{\text{velmtr}} = \begin{bmatrix} \mathbf{s}_{\text{velmtr}} \\ \epsilon_{\text{velmtr}} \\ \mathbf{b}_{\text{velmtr}} \end{bmatrix}$	$\mathbf{0}_{9\times1}$	ξ_{velmtr}	Velocimeter errors (scale, alignment, bias)	9
$\xi_{\text{TRN}} = \begin{bmatrix} \mathbf{s}_{\text{TRN}} \\ \epsilon_{\text{TRN}} \\ \mathbf{b}_{\text{TRN}} \end{bmatrix}$	$\mathbf{0}_{9\times1}$	ξ_{TRN}	Lidar / TRN errors (scale, alignment, bias)	9
$\xi_{\text{accmtr}} = \begin{bmatrix} \mathbf{s}_{\text{accmtr}} \\ \epsilon_{\text{accmtr}} \\ \mathbf{b}_{\text{accmtr}} \end{bmatrix}$	$\mathbf{0}_{9\times1}$	ξ_{accmtr}	Accelerometer errors (scale, alignment, bias)	9
$\xi_{\text{alt}} = \begin{bmatrix} s_{\text{alt}} \\ b_{\text{alt}} \\ b_{\text{terr}} \end{bmatrix}$	$\mathbf{0}_{3\times1}$	ξ_{alt}	Altimeter errors (scale, bias, terrain bias)	3
g_d	$\mathbf{0}_{3\times1}$	g_d	Gravitational model errors	3
Miscellaneous states				*1*
ΔV	$\Delta\bar{V}$	$\delta\Delta V$	Main engine delta-velocity spent	1
Total Number of States Allotted				*100*

† Symbols shown are for truth states. Filter state symbols are decorated with a *hat* $(\hat{\ })$.

inertial states, and parameter states. We shall describe the ΔV states separately in Chapter 5. The equations in this section are a part of Eqs. (2.34) on page 20 and (2.48) on page 22.

3.1.1 True Landing Site Dynamics

The nominal landing site is assumed to be at a known latitude and longitude fixed to the lunar surface. The initial position vector of the landing site in the inertial frame may be described notationally as [22]

$$ {}^{i}\mathbf{r}_{\text{site}}(0) = R_{\text{moon}} \begin{bmatrix} \cos L_{\text{site}} \cos\theta \\ \cos L_{\text{site}} \sin\theta \\ \sin L_{\text{site}} \end{bmatrix} \tag{3.1} $$

where we assume zero altitude on a spherical surface of $R_{\text{moon}} = 1\,737.4\,\text{km}$ [23], selenocentric latitude L_{site} and inertial longitude $\theta = \theta_0 + \lambda_{\text{site}}$ measured from the inertial vernal equinox vector where λ_{site} is the surface-fixed longitude and θ_0 is the inertial longitude of the lunar meridian (analogous to the Greenwich hour angle on earth) at our initial time. In the inertial reference frame, the inertial site position vector changes according to

$$ {}^{i}\dot{\mathbf{r}}_{\text{site}} = {}^{i}\boldsymbol{\omega}_{\text{moon}} \times {}^{i}\mathbf{r}_{\text{site}}, \tag{3.2} $$

where ${}^{i}\boldsymbol{\omega}_{\text{moon}}^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 2.68063496 \times 10^{-6} \end{bmatrix}$ rad/s is the rotation rate of the moon about the inertial z axis, corresponding to a sidereal period of 27 d 07 h 43 m 12 s [23].

Equation (3.2) defines the velocity vector of the site and may be used in place of ${}^{i}\mathbf{v}_{\text{site}}$ if it is required in formulation. If ${}^{i}\mathbf{a}_{\text{site}}$ is required, this expression may be substituted:

$$ {}^{i}\mathbf{a}_{\text{site}} = {}^{i}\dot{\mathbf{v}}_{\text{site}} = {}^{i}\ddot{\mathbf{r}}_{\text{site}} = {}^{i}\boldsymbol{\omega}_{\text{moon}} \times \left({}^{i}\boldsymbol{\omega}_{\text{moon}} \times {}^{i}\mathbf{r}_{\text{site}} \right). \tag{3.3} $$

In lieu of the site velocity state, a “map error” state ($\boldsymbol{e}_{\text{map}}$) is used in its place. This state is discussed in detail in the parameter dynamics section below. States for landing site attitude and rotation rate vector are reserved in the state vector, but not used. These states were allocated for the original LinCov orbital rendezvous problem for a “resident space object” (i.e. target) vehicle, but are not applicable to the current research since the surface inertial orientation and rotation is defined sufficiently with its location.

The landing site position dynamics are easily linearized by substituting $\mathbf{r}_{\text{site}} = \bar{\mathbf{r}}_{\text{site}} + \delta\mathbf{r}_{\text{site}}$ into Eq. (3.2). Assuming the moon rotation rate is inertially fixed and constant,

$$\begin{aligned} {}^{i}\dot{\bar{\mathbf{r}}}_{\text{site}} + {}^{i}\delta\dot{\mathbf{r}}_{\text{site}} &= {}^{i}\boldsymbol{\omega}_{\text{moon}} \times \left({}^{i}\bar{\mathbf{r}}_{\text{site}} + {}^{i}\delta\mathbf{r}_{\text{site}}\right) \\ {}^{i}\delta\dot{\mathbf{r}}_{\text{site}} &= {}^{i}\boldsymbol{\omega}_{\text{moon}} \times {}^{i}\delta\mathbf{r}_{\text{site}} \end{aligned} \tag{3.4}$$

It may be desired to have an expression for the site velocity dispersion dynamics (that is the site acceleration dispersion), which is

$${}^{i}\delta\ddot{\mathbf{r}}_{\text{site}} = {}^{i}\delta\dot{\mathbf{v}}_{\text{site}} = {}^{i}\boldsymbol{\omega}_{\text{moon}} \times \left({}^{i}\boldsymbol{\omega}_{\text{moon}} \times {}^{i}\delta\mathbf{r}_{\text{site}}\right). \tag{3.5}$$

3.1.2 True Six Degrees-of-Freedom Vehicle Dynamics

The equations of motion for three- and six-degrees-of-freedom motion are well known from classical dynamics. The translational motion, that is the *position* and *velocity* of the center-of-mass of an idealized rigid body (or theoretical point mass) is postulated by Newton’s second law of motion, which may be written in vector notion as

$$\sum {}^{i}\mathbf{F} = \frac{\mathrm{d}}{\mathrm{d}t}\left(m\,{}^{i}\mathbf{v}\right) \tag{3.6}$$

where the pre-superscript i indicates that the force and velocity vectors are constructed in an inertial frame of reference. Evaluating the time derivative yields $m\dot{\mathbf{v}} + \dot{m}\mathbf{v}$, although traditionally in propulsion problems, the $\dot{m}\mathbf{v}$ term is written as the thrust force created by expelling propellant. In the vicinity of the moon, the only external force of consequence is the moon’s gravitational force. Other unmodeled accelerations, and uncertainty in the gravitational force, may be modeled as random accelerations. So, the acceleration of the spacecraft may be written as

$$^{i}\dot{\mathbf{v}}(t) = \frac{1}{m(t)} \left(^{i}\mathbf{F}_{\text{grav}}(t) + {}^{i}\mathbf{F}_{\text{thr}}(t) + {}^{i}\mathbf{F}_{\text{other}}(t) \right). \tag{3.7}$$

For this analysis, vehicle mass is assumed to be known for all times deterministically, so we may consider for simplicity only the resulting accelerations, $^{i}\mathbf{a}_k(t) = {}^{i}\mathbf{F}_k(t)/m(t)$.

Gravity is assumed to be a combination of simple two-body point-mass (or Newtonian) gravitation and a stochastic disturbance component. This stochastic component may be considered to be due to uneven mass distribution (i.e. mass concentrations or mass-cons), while third bodies are lumped into the unmodeled acceleration noise, $\mathbf{w}_{\text{vel}}$ in Eq. (3.16) on page 34. The equation modeling gravity is

$$^{i}\mathbf{a}_{\text{grav}}(t) = \frac{^{i}\mathbf{F}_{\text{grav}}(t)}{m(t)} = -\frac{\mu}{r^3}{}^{i}\mathbf{r} + {}^{i}\boldsymbol{g}_d, \tag{3.8}$$

where $\mu = \mu_{\text{moon}} = GM_{\text{moon}}$ is the gravitational parameter of the moon ($4\,902.8\,\text{km}^3\text{s}^{-2}$ [23]), $r = \left\| {}^{i}\mathbf{r} \right\| = \sqrt{^{i}\mathbf{r}^{\mathsf{T}}\,{}^{i}\mathbf{r}}$ is the magnitude of the inertial position vector state, that is the distance from the center-of-mass of the moon to the vehicle, and $^{i}\boldsymbol{g}_d$ is a vector of gravity disturbance acceleration. This stochastic disturbance is colored noise term discussed in detail in 3.1.3. In future references, $\mathbf{a}_{\text{grav}}$ will refer only to the lunar two-body gravitational acceleration, and other gravitational accelerations included in surface-correlated gravitational disturbance, $\boldsymbol{g}_d$, and random acceleration noise, $\mathbf{w}_v$.

The other major acceleration in lunar descent is thrust. We may allow that the true thrust acceleration imparted to the vehicle differs from the acceleration that is commanded to the engine by introducing errors. The errors modeled here are *thrust acceleration bias*, *scale factor*, and *misalignment*. Each of these errors will also be included in the truth state vector with their own dynamics. The errors are introduced to the commanded thrust which, for now, is written as a general vector function

$$^{i}\mathbf{a}_{\text{thr}} = {}^{i}_{b}\mathcal{T} \left[\mathbf{e}\left(^{b}\hat{\mathbf{a}}_{\text{cmd}}, \boldsymbol{\xi}_{\text{thr}} \right) + \mathbf{w}_{\text{thr}} \right] \tag{3.9}$$

where the vector $\boldsymbol{\xi}_{\text{thr}}$ is a vector containing all the error parameters, $\mathbf{w}_{\text{thr}}$ is a vector of random noise associated with the engine thrust, and where we assume the commanded thrust acceleration vector and errors are constructed in the vehicle's body frame (notated with a pre-superscript b), and transformed to the inertial frame by premultiplication with the body-to-inertial transformation matrix ${}^{i}_{b}\mathcal{T}$. We will discuss this transformation in more detail when we introduce the attitude states. We will also use $\mathcal{T}$ to denote a function to construct a direction-cosine matrix from a quaternion or rotation vector argument.

In general, the function to introduce error to the generic three-element vector $\mathbf{z}$ may be written as

$$\mathbf{z}' = \mathbf{e}(\mathbf{z}, \boldsymbol{\xi}) = \mathbf{e}(\mathbf{z}, \mathbf{b}, \mathbf{s}, \boldsymbol{\epsilon}) = \mathcal{T}\left({}^{\text{misalign}}_{\text{nominal}}\boldsymbol{\epsilon}\right)\left[\mathcal{S}\mathbf{z} + \mathbf{b}\right] \tag{3.10}$$

where $\mathbf{b}$ is the bias, and the scale factor $\mathbf{s}$ is applied with

$$\mathcal{S} = \begin{bmatrix} 1+s_x & 0 & 0 \\ 0 & 1+s_y & 0 \\ 0 & 0 & 1+s_z \end{bmatrix} = \mathbf{I}_{3\times3} + \text{diag}\left(\begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix}\right) = \mathbf{I}_{3\times3} + \text{diag}\left(\mathbf{s}\right). \tag{3.11}$$

The three-element Euler rotation vector $\boldsymbol{\epsilon}$, whose direction is the axis of rotation and magnitude ϵ is the angle of rotation, is the misalignment error, such that [24, 25]

$$\mathcal{T}(\boldsymbol{\epsilon}) = \cos(\epsilon)\mathbf{I}_{3\times3} - \frac{\sin\epsilon}{\epsilon}\left[\boldsymbol{\epsilon}^{\times}\right] + \frac{1-\cos\epsilon}{\epsilon^2}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathsf{T}}. \tag{3.12}$$

In the limit as $\epsilon \rightarrow 0$, and dropping the second order term, the rotation matrix may be expressed as

$$\mathcal{T}(\boldsymbol{\epsilon}) \approx \delta\mathcal{T}(\boldsymbol{\epsilon}) = \mathbf{I}_{3\times3} - \left[\boldsymbol{\epsilon}^{\times}\right] = \begin{bmatrix} 1 & \epsilon_z & -\epsilon_y \\ -\epsilon_z & 1 & \epsilon_x \\ \epsilon_y & -\epsilon_x & 1 \end{bmatrix}, \tag{3.13}$$

which is clearly not an orthogonal transformation matrix since $\delta\mathcal{T}(\boldsymbol{\epsilon})\delta\mathcal{T}(\boldsymbol{\epsilon})^{\mathsf{T}} = \mathbf{I}_{3\times3} - \left[\boldsymbol{\epsilon}^{\times}\right]\left[\boldsymbol{\epsilon}^{\times}\right] \neq \mathbf{I}_{3\times3}$, but is acceptable to first order, which is all that is required for the linearization necessary for linear covariance analysis and developing an extended Kalman filter. With the linearized transformation matrix, the error introduction function may be written

$$\mathbf{z}' = \mathbf{e}(\mathbf{z}, \boldsymbol{\xi}) = \delta\mathcal{T}(\boldsymbol{\epsilon})\left(\mathcal{S}\mathbf{z} + \mathbf{b}\right) = \left(\mathbf{I}_{3\times 3} - \left[\boldsymbol{\epsilon}^{\times}\right]\right)\left(\left[\mathbf{I}_{3\times 3} + \mathrm{diag}\left(\mathbf{s}\right)\right]\mathbf{z} + \mathbf{b}\right). \tag{3.14}$$

The translational equations of motion of the vehicle, then, are

$$\begin{aligned} {}^{i}\dot{\mathbf{r}} &= {}^{i}\mathbf{v} && (3.15) \\ {}^{i}\dot{\mathbf{v}} &= -\frac{\mu}{r^3}{}^{i}\mathbf{r} + {}^{i}g_d + {}^{i}_{b}\mathcal{T}\mathbf{e}({}^{b}\hat{\mathbf{a}}_{\mathrm{cmd}}, \boldsymbol{\xi}_{\mathrm{thr}}) + \mathbf{w}_v, && (3.16) \end{aligned}$$

where $\mathbf{w}_v$ is a collection of independent random white noise disturbance terms, including $\mathbf{w}_{\mathrm{thr}}$ and $\mathbf{w}_{\mathrm{grav}}$ and $\mathbf{w}_{v,\mathrm{actuator}}$, which are described generically in Section 2.3.1 with their respective noise strengths. Note that the thrust noise in Eq. (3.9) has moved from the body frame to the inertial frame. This is permitted because all white noise processes are independent of coordinate system and of each element in the noise vector.

Rotational dynamics are given by the well-known Euler's equation, found by differentiating the angular momentum equation in the body frame

$$\begin{aligned} \sum {}^{b}\mathbf{M} &= \frac{\mathrm{d}}{\mathrm{d}t}\left({}^{b}I^{b}\boldsymbol{\omega}\right) = {}^{b}I^{b}\dot{\boldsymbol{\omega}} + {}^{b}\dot{I}^{b}\boldsymbol{\omega} + {}^{b}\boldsymbol{\omega} \times {}^{b}I^{b}\boldsymbol{\omega} \\ {}^{b}\dot{\boldsymbol{\omega}} &= {}^{b}I^{-1}\left(\sum {}^{b}\mathbf{M} - {}^{b}\dot{I}^{b}\boldsymbol{\omega} - {}^{b}\boldsymbol{\omega} \times {}^{b}I^{b}\boldsymbol{\omega}\right), && (3.17) \end{aligned}$$

where ${}^{b}\boldsymbol{\omega}$ is the angular rotation rate, ${}^{b}I$ is the spacecraft inertia tensor, and ${}^{b}\mathbf{M}$ are moments or torques applied to the vehicle, all in the body frame.

For the problem considered, the only torque acting on the vehicle is the reaction control system (RCS) torques created by firing small thrusters designed and placed for attitude control. It is assumed that the Descent Module Main Propulsion (DMMP) system's thrust acts through the vehicle center-of-gravity (cg) at all times. The control torque, like the control thrust, is made imperfect by the introduction of *torquer bias*, *scale factor*, and *misalignment*, so we have

$$ {}^{b}\dot{\boldsymbol{\omega}} = {}^{b}I^{-1}\left[\mathbf{e}\left({}^{b}\hat{\mathbf{M}}_{\text{cmd}}, \boldsymbol{\xi}_{\text{trq}}\right) - {}^{b}\boldsymbol{\omega} \times {}^{b}I^{b}\boldsymbol{\omega}\right] + \mathbf{w}_{\omega}, \tag{3.18} $$

where the $\dot{I}\boldsymbol{\omega}$ term is considered a less significant term to be included in the unmodeled angular acceleration term, $\mathbf{w}_{\omega}$, which also includes attitude-control system torquer disturbances ($\mathbf{w}_{\text{ACS}}$) and miscellaneous actuator disturbances ($\mathbf{w}_{\omega,\text{actuator}}$).

Rotational kinematics, that is how angular rotation rate translates into rotation, in three dimensions is a non-trivial problem about which an expansive amount of material has been written [15, 16, 24–27, to name just a few]. A full treatment into the different parametrizations of attitude, their singularities, advantages and disadvantages, applicable equations, and so forth, is not given here, but can be found among these references. For this research, a hybrid approach is used—for the nominal, nonlinear "Monte Carlo" formulation, a quaternion is used to represent the attitude, but for the linearized analysis, the delta-attitude state is represented by a small-angle rotation vector.

Euler's theorem states that any solid body rotation (or coordinate system transformation) may be represented by a rotation about a single axis over a finite angle. A unit quaternion may be used to represent these quantities as a four-dimensional vector

$$ \mathbf{q} = \begin{pmatrix} x \\ y \\ z \\ s \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ s \end{pmatrix} = \begin{pmatrix} \mathbf{u}\sin(\varphi/2) \\ \cos(\varphi/2) \end{pmatrix} \tag{3.19} $$

where $\mathbf{u}$ is a unit vector of the axis of rotation, and φ is the angle of rotation. The fourth element, s, is the scalar component, while the first three elements, $\mathbf{r}$, are known as the vector component. The direction cosine matrix (DCM, or transformation matrix) represented by this quaternion may be written as [24, 25]

$$ \mathcal{T}(\mathbf{q}) = (\cos\varphi)\mathbf{I} - (\sin\varphi)\left[\mathbf{u}^{\times}\right] + (1-\cos\varphi)\mathbf{u}\mathbf{u}^{\mathsf{T}}, \tag{3.20} $$

although there are more computationally efficient ways of transforming a vector coordinatized in one frame into another than to extract the DCM from the quaternion [15,25]. For applying multiple transformations, we will use the notation [24,25]

$$\mathbf{q}_3 = \mathbf{q}_1 \otimes \mathbf{q}_2 = \left[\mathbf{q}_1^{\otimes}\right] \mathbf{q}_2 \tag{3.21}$$

where

$$\left[\mathbf{q}^{\otimes}\right] = \begin{bmatrix} s\mathbf{I}_{3\times3} - [\mathbf{r}^{\times}] & \mathbf{r} \\ -\mathbf{r}^{\mathsf{T}} & s \end{bmatrix} = \begin{bmatrix} s & z & -y & x \\ -z & s & x & y \\ y & -x & s & z \\ -x & -y & -z & s \end{bmatrix} \tag{3.22}$$

such that $\mathcal{T}(\mathbf{q}_3) = \mathcal{T}(\mathbf{q}_1)\mathcal{T}(\mathbf{q}_2)$.

A quaternion dynamically evolves by the relation [2,25]

$${}^{b}_{i}\dot{\mathbf{q}} = \frac{1}{2}{}^{b}\boldsymbol{\omega}' \otimes {}^{b}_{i}\mathbf{q} \tag{3.23}$$

where ${}^{b}\boldsymbol{\omega}' = \binom{{}^{b}\boldsymbol{\omega}}{0}$ is a form of the rotational velocity vector for quaternion multiplication.

To linearize the attitude and rotational states, first we define the dispersion states $\delta\boldsymbol{\theta}$ and $\delta\boldsymbol{\omega}$ in terms of the nonlinear and nominal states. Substituting ${}^{b}\boldsymbol{\omega} = {}^{b}\bar{\boldsymbol{\omega}} + {}^{b}\delta\boldsymbol{\omega}$ into the rotational dynamics Eq. (3.18) and linearizing using a Taylor's Series expansion, we write

$$\begin{aligned} {}^{b}\delta\dot{\boldsymbol{\omega}} &= \left[\frac{\partial^{b}\dot{\boldsymbol{\omega}}}{\partial\mathbf{x}}\right]_{\bar{\mathbf{x}}} \delta\mathbf{x} + \left[\frac{\partial^{b}\dot{\boldsymbol{\omega}}}{\partial\hat{\mathbf{u}}}\right]_{\bar{\mathbf{x}}} \delta\hat{\mathbf{u}} + \mathbf{w}_{\omega} \\ &= {}^{b}I^{-1}\left[\left.\frac{\partial\mathbf{e}}{\partial^{b}\hat{\mathbf{M}}_{\text{cmd}}}\right|_{\bar{\mathbf{x}}} {}^{b}\delta\hat{\mathbf{M}}_{\text{cmd}} + \left.\frac{\partial\mathbf{e}}{\partial\boldsymbol{\xi}_{\text{trq}}}\right|_{\bar{\mathbf{x}}} \boldsymbol{\xi}_{\text{trq}} - \left.\frac{\partial\left({}^{b}\boldsymbol{\omega} \times {}^{b}I^{b}\boldsymbol{\omega}\right)}{\partial^{b}\boldsymbol{\omega}}\right|_{\bar{\mathbf{x}}} {}^{b}\delta\boldsymbol{\omega}\right] + \mathbf{w}_{\omega} \end{aligned}$$

Here it is worth evaluating the partial derivatives of the error introduction function, because it will be used is several places. For generic arguments $\mathbf{z}$ and $\boldsymbol{\xi}_z$, we have from

Eq. (3.14):

$$\begin{aligned} E_z &= \left.\frac{\partial \mathbf{e}}{\partial \mathbf{z}}\right|_{\bar{\mathbf{x}}} = \delta\mathcal{T}(\boldsymbol{\epsilon})\mathcal{S}|_{\bar{\mathbf{x}}} = \mathbf{I}_{3\times 3}, \qquad \text{and} & (3.24) \\ E_{\xi}(\mathbf{z}) &= \left[\begin{array}{ccc} E_s & E_\epsilon & E_b \end{array}\right] = \left[\begin{array}{ccc} \frac{\partial \mathbf{e}}{\partial \mathbf{s}} & \frac{\partial \mathbf{e}}{\partial \boldsymbol{\epsilon}} & \frac{\partial \mathbf{e}}{\partial \mathbf{b}} \end{array}\right]_{\bar{\mathbf{x}}} \\ &= \left[\begin{array}{ccc} \delta\mathcal{T}(\boldsymbol{\epsilon})\mathrm{diag}(\mathbf{z}) & [\mathcal{S}\mathbf{z}+\mathbf{b}]^{\times} & \delta\mathcal{T}(\boldsymbol{\epsilon}) \end{array}\right]_{\bar{\mathbf{x}}} \\ &= \left[\begin{array}{ccc} \mathrm{diag}(\bar{\mathbf{z}}) & [\bar{\mathbf{z}}^{\times}] & \mathbf{I}_{3\times 3} \end{array}\right]. & (3.25) \end{aligned}$$

Therefore, in general we may linearize $\mathbf{z}' = \mathbf{e}(\mathbf{z}, \boldsymbol{\xi}_z)$ as

$$\delta\mathbf{z}' = \delta\mathbf{z} + E_{\xi}(\bar{\mathbf{z}})\xi_z. \tag{3.26}$$

Substituting Eq. (3.26) into the linearized rotational dynamics and simplifying, we have

$${}^{b}\delta\dot{\boldsymbol{\omega}} = {}^{b}I^{-1}\left[{}^{b}\delta\hat{\mathbf{M}}_{\text{cmd}} + E_{\xi}\left({}^{b}\bar{\mathbf{M}}_{\text{cmd}}\right)\boldsymbol{\xi}_{\text{trq}} - \left(\left[{}^{b}\bar{\boldsymbol{\omega}}^{\times}\right]{}^{b}I - \left[{}^{b}I\,{}^{b}\bar{\boldsymbol{\omega}}\right]^{\times}\right){}^{b}\delta\boldsymbol{\omega}\right] + \mathbf{w}_{\omega}. \tag{3.27}$$

The attitude dispersion, ${}^{b}\delta\boldsymbol{\theta}$, needs to be handled carefully. We define the attitude dispersion such that

$${}^{b}_{i}\mathcal{T} = \mathcal{T}({}^{b}_{i}\mathbf{q}) = \delta\mathcal{T}({}^{b}\delta\boldsymbol{\theta})\bar{\mathcal{T}}({}^{b}_{i}\bar{\mathbf{q}}). \tag{3.28}$$

As dispersions are expected to be small, we use the small angle rotation approximation of the direction cosine matrix for the 3-element small rotation vector $\delta\boldsymbol{\theta}$,

$$\delta\mathcal{T}({}^{b}\delta\boldsymbol{\theta}) = \mathbf{I}_{3\times 3} - \left[{}^{b}\delta\boldsymbol{\theta}^{\times}\right]. \tag{3.29}$$

We note also that since ${}^{i}_{b}\mathcal{T} = \mathcal{T}({}^{b}_{i}\mathbf{q})^{\mathsf{T}} = \mathcal{T}({}^{b}_{i}\bar{\mathbf{q}})^{\mathsf{T}}\delta\mathcal{T}({}^{b}\delta\boldsymbol{\theta})^{\mathsf{T}}$,

$${}^{i}_{b}\mathcal{T} = {}^{i}_{b}\bar{\mathcal{T}}\delta\mathcal{T}\left(-{}^{b}\delta\boldsymbol{\theta}\right) = {}^{i}_{b}\bar{\mathcal{T}}\left\{\mathbf{I}_{3\times 3} + \left[{}^{b}\delta\boldsymbol{\theta}^{\times}\right]\right\} \tag{3.30}$$

are all expressions of the body-to-inertial transformation in terms of the small angle rotation.

We can use estimate the rotation dispersion as a small quaternion $\delta\mathbf{q} = \binom{\delta\boldsymbol{\theta}/2}{1}$, $\delta\dot{\mathbf{q}} = \binom{\delta\dot{\boldsymbol{\theta}}/2}{0}$, and use quaternion dynamics Eq. (3.23) to determine the attitude dispersion dynamic equation, recognizing that the time derivative of the *nominal* quaternion is ${}^b_i\dot{\bar{\mathbf{q}}} = \frac{1}{2}{}^b\bar{\boldsymbol{\omega}}' \otimes {}^b_i\bar{\mathbf{q}}$.

$$\begin{aligned}
{}^b_i\dot{\mathbf{q}} = \frac{\mathrm{d}}{\mathrm{d}t}\left[{}^b\delta\mathbf{q} \otimes {}^b_i\bar{\mathbf{q}}\right] &= \frac{1}{2}{}^b\boldsymbol{\omega}' \otimes {}^b_i\mathbf{q} \\
\left[{}^b\delta\dot{\mathbf{q}}^{\otimes}\right]{}^b_i\bar{\mathbf{q}} + \left[{}^b\delta\mathbf{q}^{\otimes}\right]{}^b_i\dot{\bar{\mathbf{q}}} &= \frac{1}{2}{}^b\boldsymbol{\omega}' \otimes \left[{}^b\delta\mathbf{q} \otimes {}^b_i\bar{\mathbf{q}}\right] \\
\left[{}^b\delta\dot{\mathbf{q}}^{\otimes}\right]{}^b_i\bar{\mathbf{q}} + \left[{}^b\delta\mathbf{q}^{\otimes}\right]\frac{1}{2}\left[{}^b\bar{\boldsymbol{\omega}}'^{\otimes}\right]{}^b_i\bar{\mathbf{q}} &= \frac{1}{2}\left[{}^b\boldsymbol{\omega}'^{\otimes}\right]\left[{}^b\delta\mathbf{q}^{\otimes}\right]{}^b_i\bar{\mathbf{q}}
\end{aligned}$$

The nominal quaternion, ${}^b_i\bar{\mathbf{q}}$, may be eliminated from both sides of the equation without concern since we have defined it to be a unit (i.e. nonzero) quaternion, therefore the coefficient matrices must be equal. Using the definitions of ${}^b\delta\mathbf{q}$ and of the quaternion composition rule Eq. (3.21), we continue:

$$\begin{bmatrix} -\frac{1}{2}\left[{}^b\delta\dot{\boldsymbol{\theta}}^{\times}\right] & \frac{1}{2}{}^b\delta\dot{\boldsymbol{\theta}} \\ -\frac{1}{2}{}^b\delta\dot{\boldsymbol{\theta}}^{\mathsf{T}} & 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \mathbf{I}_{3\times 3} - \frac{1}{2}\left[{}^b\delta\boldsymbol{\theta}^{\times}\right] & \frac{1}{2}{}^b\delta\boldsymbol{\theta} \\ -\frac{1}{2}{}^b\delta\boldsymbol{\theta}^{\mathsf{T}} & 1 \end{bmatrix}\begin{bmatrix} -\left[{}^b\bar{\boldsymbol{\omega}}^{\times}\right] & {}^b\bar{\boldsymbol{\omega}} \\ -{}^b\bar{\boldsymbol{\omega}}^{\mathsf{T}} & 0 \end{bmatrix} =$$

$$\frac{1}{2}\begin{bmatrix} -\left[{}^b\boldsymbol{\omega}^{\times}\right] & {}^b\boldsymbol{\omega} \\ -{}^b\boldsymbol{\omega}^{\mathsf{T}} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{I}_{3\times 3} - \frac{1}{2}\left[{}^b\delta\boldsymbol{\theta}^{\times}\right] & \frac{1}{2}{}^b\delta\boldsymbol{\theta} \\ -\frac{1}{2}{}^b\delta\boldsymbol{\theta}^{\mathsf{T}} & 1 \end{bmatrix}$$

It is not necessary to fully evaluate the matrix multiplication in order to find an equation for ${}^b\delta\dot{\boldsymbol{\theta}}$—we may wisely choose only to look at the 3×1 upper-right section of the resultant partitioned matrix after multiplication:

$$\begin{aligned}
{}^b\delta\dot{\boldsymbol{\theta}} + {}^b\bar{\boldsymbol{\omega}} - \frac{1}{2}\left[{}^b\delta\boldsymbol{\theta}^{\times}\right]{}^b\bar{\boldsymbol{\omega}} &= -\left[{}^b\boldsymbol{\omega}^{\times}\right]\frac{1}{2}{}^b\delta\boldsymbol{\theta} + {}^b\boldsymbol{\omega} \\
{}^b\delta\dot{\boldsymbol{\theta}} &= -\frac{1}{2}\left({}^b\bar{\boldsymbol{\omega}} + {}^b\delta\boldsymbol{\omega}\right) \times {}^b\delta\boldsymbol{\theta} + {}^b\boldsymbol{\omega} - {}^b\bar{\boldsymbol{\omega}} - \frac{1}{2}{}^b\bar{\boldsymbol{\omega}} \times {}^b\delta\boldsymbol{\theta} \\
{}^b\delta\dot{\boldsymbol{\theta}} &= {}^b\delta\boldsymbol{\omega} - {}^b\bar{\boldsymbol{\omega}} \times {}^b\delta\boldsymbol{\theta} - \frac{1}{2}{}^b\delta\boldsymbol{\omega} \times {}^b\delta\boldsymbol{\theta} \qquad (3.31)
\end{aligned}$$

Equation (3.31) is a form of the Bortz equation [24, 26] which is used to propagate small angle rotation vectors. We do not necessarily require the final term because it is a second-order term which will drop out when evaluating partial derivatives, because $\delta\boldsymbol{\omega} = \delta\boldsymbol{\theta} = \mathbf{0}_{3\times1}$ along the nominal.

Finally, we consider the true translational dynamics. We substitute $\mathbf{r} = \bar{\mathbf{r}} + \delta\mathbf{r}$ and $\mathbf{v} = \bar{\mathbf{v}} + \delta\mathbf{v}$ into Eq. (3.15) to quickly obtain

$$ {}^{i}\delta\dot{\mathbf{r}} = {}^{i}\delta\mathbf{v}. \tag{3.32}$$

To linearize the velocity derivative Eq. (3.16), we will need to use Taylor's Series expansion and substitute Eq. (3.66), and find

$$\begin{aligned} {}^{i}\dot{\mathbf{v}} &= -\frac{\mu}{r^3}{}^{i}\mathbf{r} + {}^{i}\mathbf{g}_d + {}^{i}_{b}\mathcal{T}\,\mathbf{e}({}^{b}\hat{\mathbf{a}}_{\text{cmd}}, \boldsymbol{\xi}_{\text{thr}}) + \mathbf{w}_v \\ {}^{i}\delta\dot{\mathbf{v}} &= \mathbb{G}_r{}^{i}\delta\mathbf{r} + {}^{i}\mathbf{g}_d + {}^{i}_{b}\bar{\mathcal{T}}\left[{}^{b}\delta\hat{\mathbf{a}}_{\text{cmd}} + E_{\xi}({}^{b}\bar{\mathbf{a}}_{\text{cmd}})\boldsymbol{\xi}_{\text{thr}}\right] + {}^{i}_{b}\bar{\mathcal{T}}\left[-{}^{b}\bar{\mathbf{a}}^{\times}_{\text{cmd}}\right]{}^{b}\delta\boldsymbol{\theta} + \mathbf{w}_v \end{aligned} \tag{3.33}$$

where the partial of the two-body gravitational acceleration is known as the *gravity gradient* matrix,

$$\mathbb{G}_r = \frac{\partial}{\partial{}^{i}\mathbf{r}}\left[-\frac{\mu}{r^3}{}^{i}\mathbf{r}\right]_{\bar{\mathbf{x}}} = -\frac{\mu}{\bar{r}^3}\left[\mathbf{I} - 3{}^{i}\bar{\mathbf{i}}_r{}^{i}\bar{\mathbf{i}}_r^{\mathsf{T}}\right]. \tag{3.34}$$

Note also that we have used ${}^{i}\mathbf{g}_d$ and $\boldsymbol{\xi}_{\text{thr}}$ instead of perturbation states with the usual notation ${}^{i}\delta\mathbf{g}_d$ and $\delta\boldsymbol{\xi}_{\text{thr}}$. That is because they are identical, since ${}^{i}\bar{\mathbf{g}}_d = \mathbf{0}$ and $\bar{\boldsymbol{\xi}}_{\text{thr}} = \mathbf{0}$, or more generically $\bar{p}_i = 0$, as described in the next section.

3.1.3 True Parameter Dynamics

Reference has been made in previous sections to error parameters that perturb a nominal condition. For this research, every instrument and actuator has, in addition to white noise, associated alignment, bias, and/or scale factor terms, as appropriate for the system. Each of these parameters (except for white noise) is modeled as a first-order Markov process [1, 15, 16] with its own associated noise strength and time constant. The dynamics of a first-order Markov process (or colored noise process) may be written

$$\dot{p}_i = -\frac{1}{\tau_i}p_i + w_{p_i}, \quad i = 1, 2, \ldots n_p \tag{3.35}$$

where p_i is the scalar parameter, τ_i is its time constant, and w_{p_i} is its associated zero-mean white noise with strength

$$\mathrm{E}\left[w_{p_i}(t)w_{p_i}(t')\right] = \sigma_{p_i}^2\delta(t-t'), \tag{3.36}$$

where δ here is the Dirac delta function.

A error parameter vector is formulated so that its components are independent from each other. That is,

$$\dot{\mathbf{p}} = -\mathrm{diag}[\boldsymbol{\tau}]^{-1}\mathbf{p} + \mathbf{w}_p \qquad \mathrm{E}\left[\mathbf{w}_p\mathbf{w}_p^{\mathsf{T}}\right] = S_p, \tag{3.37}$$

where S_p is a diagonal matrix of mutually independent noise strength terms.

Most of the parameters are time-correlated, and τ has units of time. We allow also for parameters to be correlated with distance traveled over the lunar surface. This is useful for uncertainties related to location above the surface or the features on the terrain below. Three parameter states use correlation distance rather than time constant, which are altimeter terrain bias, b_{terr}, accounting for topography changes; map error, $\boldsymbol{e}_{\mathrm{map}}$, degrading the quality of terrain-relative navigation; and gravitational disturbances, $\boldsymbol{g}_d$, due primarily to lunar mass concentrations. To make distance correlation possible, we define the parameter dynamics as Markov processes with respect to surface location, x_{surf}, with a distance correlation constant, d_{corr}, and perform a change of variable:

$$p_j' = \frac{\mathrm{d}p_j}{\mathrm{d}x_{\mathrm{surf}}} = -\frac{1}{d_{\mathrm{corr}}}p_j + w_{p_j} \tag{3.38}$$

$$\dot{p}_j = \frac{\mathrm{d}p_j}{\mathrm{d}t} = \frac{\mathrm{d}p_j}{\mathrm{d}x_{\mathrm{surf}}}\frac{\mathrm{d}x_{\mathrm{surf}}}{\mathrm{d}t} = -\frac{v_{\mathrm{surf}}}{d_{\mathrm{corr}}}p_j + v_{\mathrm{surf}}w_{p_j} \tag{3.39}$$

The noise term could be written $v_{\mathrm{surf}}w_{p_j}$ as shown, or simply as w_{p_j} depending on the desired noise formulation. The surface velocity, or the velocity component perpendicular to the position vector, is defined

$$v_{\text{surf}} = v_{\perp} = \left\| \mathbf{v} - (\mathbf{i}_r \cdot \mathbf{v})\mathbf{i}_r - \boldsymbol{\omega}_{\text{moon}} \times \mathbf{r} \right\| . \tag{3.40}$$

Normally, we define a perturbation state to replace the nonlinear state, but this is not required with the error parameters because we define

$$\bar{p}_i = 0, \qquad i = 1, 2, 3, \ldots n_p. \tag{3.41}$$

Therefore, the true nonlinear state written in terms of the nominal and true linearized variable is $p_i = \bar{p}_i + \delta p_i = \delta p_i$. For simplicity, we will not use δ when identifying linearized parameter states.

By inspection of Eq. (3.35), we can simply state that the linearized dynamics for time-correlated parameter states is identical to the nonlinear dynamics, since Eq. (3.35) is linear with respect to the truth states (τ_i are not part of the state vector). Equation (3.39) for distance-correlated parameters is a bit more complicated. We may linearize this equation by a Taylor's Series expansion using the chain rule:

$$\dot{p}_j = \left[-\frac{v_{\perp}}{d_{\text{corr}}} \right]_{\bar{\mathbf{x}}} p_j + \left[-\frac{p_j}{d_{\text{corr}}} \right]_{\bar{\mathbf{x}}} \delta v_{\perp} + w_{p_j}$$

Before evaluating the $\delta v_{\perp}$, note that its coefficient is $\bar{p}_j / d_{\text{corr}}$ and we have all ready defined $\bar{p}_i = 0$ for all parameters, including distance-correlated ones. Therefore, the above equation reduces nicely to

$$\dot{p}_j = -\frac{\bar{v}_{\perp}}{d_{\text{corr}}} p_j + w_{p_j}. \tag{3.42}$$

For all parameters then, and in vector form, we can write

$$\dot{\mathbf{p}} = \text{diag}(\boldsymbol{\tau})^{-1} \mathbf{p} + \mathbf{w}_p, \tag{3.43}$$

where τ_i are required time constants and $\tau_j = d_{\text{corr}}/\bar{v}_\perp$ are the equivalent time constants for distance-correlated variables.

3.2 Continuous Inertial Instrument Models

Before describing the equations modeling the filter state dynamics, we first evaluate the models of the continuous inertial instruments since these instruments are used by the filter to propagate its state estimate.

The presence of a set of three orthogonal, body-fixed accelerometers (or collectively, "the accelerometer") on a spacecraft greatly simplifies the problem of estimating the vehicle's inertial position and velocity. Accelerometers are capable of sensing accelerations caused by non-gravitational forces. For this research, only lunar gravitational attraction and vehicle thrust are considered, so the only sensed acceleration is the thrust acceleration. We, therefore, model the accelerometer-sensed acceleration, with

$$ {}^{b}\tilde{\mathbf{a}}_{\text{accmtr}} = \mathbf{e}({}^{b}\mathbf{a}_{\text{thr}}, \boldsymbol{\xi}_{\text{accmtr}}) + \boldsymbol{\eta}_{\text{accmtr}}, \tag{3.44} $$

where $\boldsymbol{\xi}_{\text{accmtr}}$ is the vector of true accelerometer error parameters and $\boldsymbol{\eta}_{\text{accmtr}}$ is its noise. The true thrust acceleration vector, $\mathbf{a}_{\text{thr}}$, is not a state but an intermediate calculation based on true thruster errors and control commands. This vector is discussed in detail in Section 4.1, but we have already partially defined it in Eq. (3.9). Substituting, $\mathbf{a}_{\text{thr}}$ in the body frame, we obtain

$$ {}^{b}\tilde{\mathbf{a}}_{\text{accmtr}} = \mathbf{e}(\mathbf{e}({}^{b}\hat{\mathbf{a}}_{\text{cmd}}, \boldsymbol{\xi}_{\text{thr}}) + \mathbf{w}_{\text{thr}}, \boldsymbol{\xi}_{\text{accmtr}}) + \boldsymbol{\eta}_{\text{accmtr}}. \tag{3.45} $$

The vehicle rotation is sensed by a set of body-fixed or inertially mounted gyroscopes, or collectively just "the gyro." As with the accelerometer, we model the gyro using true error parameters as

$$ {}^{b}\tilde{\boldsymbol{\omega}}_{\text{gyro}} = \mathbf{e}({}^{b}\omega, \boldsymbol{\xi}_{\text{gyro}}) + \boldsymbol{\eta}_{\text{gyro}}. \tag{3.46} $$

This time, the true rotation rate vector ${}^{b}\boldsymbol{\omega}$ is part of the truth state, as are the error parameters $\boldsymbol{\xi}_{\text{gyro}}$.

As with the state dynamics equations, we must linearize these inertial instrument models for use in Section 3.3. By inspection and following the pattern developed in Section 3.1, we write

$$
\begin{aligned}
{}^{b}\delta\tilde{\mathbf{a}}_{\text{accmtr}} &= {}^{b}\delta\hat{\mathbf{a}}_{\text{cmd}} + E_{\xi}({}^{b}\bar{\mathbf{a}}_{\text{thr}})\left[\boldsymbol{\xi}_{\text{thr}} + \boldsymbol{\xi}_{\text{accmtr}}\right] + \mathbf{w}_{\text{thr}} + \boldsymbol{\eta}_{\text{accmtr}} && (3.47)\\
{}^{b}\delta\tilde{\boldsymbol{\omega}}_{\text{gyro}} &= {}^{b}\delta\boldsymbol{\omega} + E_{\xi}({}^{b}\bar{\boldsymbol{\omega}})\boldsymbol{\xi}_{\text{gyro}} + \boldsymbol{\eta}_{\text{gyro}}. && (3.48)
\end{aligned}
$$

Equation (3.46) fits easily into the generic continuous measurement model $\tilde{\mathbf{y}}$ in Eq. (2.35) on page 20, and Eq. (3.48) into the linearized form, $\delta\tilde{\mathbf{y}} = C_x\delta\mathbf{x} + \boldsymbol{\eta}$. Note, however, that Eqs. (3.45) and (3.47) contain $\hat{\mathbf{a}}_{\text{cmd}}$ and $\delta\hat{\mathbf{a}}_{\text{cmd}}$, respectively, which are members of the command vector $\hat{\mathbf{u}}$ and dispersion $\delta\hat{\mathbf{u}}$, not the true state $\mathbf{x}$ and dispersion $\delta\mathbf{x}$. These terms will be thoroughly defined in Chapter 4, and we will discuss in Chapter 5 how these terms are to be handled in the context of Eq. (2.35).

3.3 Navigation Filter State Dynamics

An important element of a sophisticated spacecraft is its ability to autonomously navigate, that is to be able to track its position and velocity (and other important values) within its flight computer based on its sensor data and programmed mathematical model. This section describes the state dynamic equations propagated by the navigation computer. These equations are a part of Eq. (2.49) on page 22.

3.3.1 Filter Landing Site Dynamics

As with the true landing site state, the filter estimate of the landing site location evolves by simply rotating its inertial position about the lunar axis,

$$
{}^{i}\dot{\hat{\mathbf{r}}}_{\text{site}} = {}^{i}\hat{\mathbf{v}}_{\text{site}} = {}^{i}\boldsymbol{\omega}_{\text{moon}} \times {}^{i}\hat{\mathbf{r}}_{\text{site}}. \qquad (3.49)
$$

We could use a filter estimate of the moon rotation ${}^i\hat{\boldsymbol\omega}_{\text{moon}}$, but we will consider the lunar rotation vector to be deterministic, that is a constant input within the flight computer. We can also define the site acceleration estimate, if it is necessary, as

$$ {}^i\dot{\hat{\mathbf{v}}}_{\text{site}} = {}^i\hat{\mathbf{a}}_{\text{site}} = {}^i\boldsymbol{\omega}_{\text{moon}} \times \left({}^i\boldsymbol{\omega}_{\text{moon}} \times {}^i\hat{\mathbf{r}}_{\text{site}} \right). \tag{3.50} $$

In the filter state as with the truth state, the site velocity is not an estimated state and there are unused states for landing site attitude and rotation rate vector reserved.

Linearization of these equations is straight forward, as it is with the truth state dynamics. Following the pattern of equations in Section 3.1.1, we write

$$ \begin{aligned} {}^i\delta\dot{\hat{\mathbf{r}}}_{\text{site}} &= {}^i\boldsymbol{\omega}_{\text{moon}} \times {}^i\delta\hat{\mathbf{r}}_{\text{site}} \\ {}^i\delta\dot{\hat{\mathbf{v}}}_{\text{site}} &= {}^i\boldsymbol{\omega}_{\text{moon}} \times \left({}^i\boldsymbol{\omega}_{\text{moon}} \times {}^i\delta\hat{\mathbf{r}}_{\text{site}} \right). \end{aligned} $$

3.3.2 Filter Vehicle Translational Dynamics

The vehicle Kalman filter attempts to model the vehicle true dynamics by integrating corrected accelerometer data.

$$ \begin{aligned} {}^i\dot{\hat{\mathbf{r}}} &= {}^i\hat{\mathbf{v}} && (3.51) \\ {}^i\dot{\hat{\mathbf{v}}} &= {}^i_b\hat{\mathcal{T}}\,\hat{\mathbf{e}}^{-1}({}^b\tilde{\mathbf{a}}_{\text{accmtr}}, \hat{\boldsymbol{\zeta}}_{\text{accmtr}}) + {}^i\hat{\mathbf{a}}_{\text{grav}}({}^i\hat{\mathbf{r}}) + {}^i\hat{\mathbf{g}}_d && (3.52) \end{aligned} $$

Since accelerometers do not detect gravitational forces, the guidance routine calculates the gravitational acceleration vector using simple two-body gravitation,

$$ {}^i\hat{\mathbf{a}}_{\text{grav}} = -\frac{\mu}{\|\hat{\mathbf{r}}\|^3}\,{}^i\hat{\mathbf{r}}. \tag{3.53} $$

The error correction done by the flight computer (notated as the functional inverse of Eq. (3.10) on page 33) is given by

$$\hat{\mathbf{e}}^{-1}(\tilde{\mathbf{y}}, \hat{\boldsymbol{\xi}}) = \hat{\mathbf{e}}^{-1}(\tilde{\mathbf{y}}, \hat{\mathbf{b}}, \hat{\mathbf{s}}, \hat{\epsilon}) = \hat{\mathcal{S}}^{-1}\left(\left[\mathcal{T}\left({}^{\text{misalign}}_{\text{nominal}}\hat{\epsilon}\right)\right]^{-1}\tilde{\mathbf{y}} - \hat{\mathbf{b}}\right), \tag{3.54}$$

where the error parameters are the filter analogues to the truth state parameters in Eq. (3.10) on page 33. The scale factor and direction cosine matrix are linearized as

$$\begin{aligned} \hat{\mathcal{S}}^{-1} &\approx \begin{bmatrix} 1-\hat{s}_x & 0 & 0 \\ 0 & 1-\hat{s}_y & 0 \\ 0 & 0 & 1-\hat{s}_z \end{bmatrix} = \mathbf{I}_{3\times3} - \operatorname{diag}(\hat{\mathbf{s}}), & (3.55) \\ \left[\mathcal{T}\left({}^{\text{misalign}}_{\text{nominal}}\hat{\epsilon}\right)\right]^{-1} &\approx \begin{bmatrix} 1 & -\hat{\epsilon}_z & \hat{\epsilon}_y \\ \hat{\epsilon}_z & 1 & -\hat{\epsilon}_x \\ -\hat{\epsilon}_y & \hat{\epsilon}_x & 1 \end{bmatrix} = \mathbf{I}_{3\times3} + \left[\hat{\epsilon}^{\times}\right]. & (3.56) \end{aligned}$$

As with the rotation matrix, the scale factor matrix is canceled by its approximate inverse to first-order, that is $\hat{\mathcal{S}}\hat{\mathcal{S}}^{-1} = \mathbf{I}_{3\times3} - \operatorname{diag}(\hat{\mathbf{s}})\operatorname{diag}(\hat{\mathbf{s}}) \approx \mathbf{I}_{3\times3}$. Therefore, the linearized error correction equation is written

$$\hat{\mathbf{e}}^{-1}(\tilde{\mathbf{y}}, \hat{\boldsymbol{\xi}}) = \hat{\mathcal{S}}^{-1}\left[\delta\mathcal{T}\left(\hat{\epsilon}\right)^{\mathsf{T}}\tilde{\mathbf{y}} - \hat{\mathbf{b}}\right] = \left(\mathbf{I}_{3\times3} - \operatorname{diag}(\hat{\mathbf{s}})\right)\left\{\left(\mathbf{I}_{3\times3} + \left[\hat{\epsilon}^{\times}\right]\right)\tilde{\mathbf{y}} - \hat{\mathbf{b}}\right\}. \tag{3.57}$$

The equations of motion for linearized filter dynamics are found in the same way, which we will do without repeating the derivations in Section 3.1.2. To reduce the complexity of these equations, we shall first define the partial derivatives of the generic error correction formula, Eq. (3.57). As with Eq. (3.24), we have

$$\begin{aligned} \hat{E}'_{\tilde{y}} &= \left.\frac{\partial\hat{\mathbf{e}}^{-1}}{\partial\tilde{\mathbf{y}}}\right|_{\bar{\mathbf{x}}} = \hat{\mathcal{S}}^{-1}\delta\hat{\mathcal{T}}(\hat{\epsilon})^{\mathsf{T}}\Big|_{\bar{\mathbf{x}}} = \mathbf{I}_{3\times3}, \qquad \text{and} & (3.58) \\ \hat{E}'_{\hat{\xi}}(\tilde{\mathbf{y}}) &= \begin{bmatrix} \hat{E}'_{\hat{s}} & \hat{E}'_{\epsilon} & \hat{E}'_{b} \end{bmatrix} = \begin{bmatrix} \frac{\partial\hat{\mathbf{e}}^{-1}}{\partial\hat{\mathbf{s}}} & \frac{\partial\hat{\mathbf{e}}^{-1}}{\partial\hat{\epsilon}} & \frac{\partial\hat{\mathbf{e}}^{-1}}{\partial\hat{\mathbf{b}}} \end{bmatrix}_{\bar{\mathbf{x}}} \end{aligned}$$

Since $\hat{\mathcal{S}}^{-1} = \mathbf{I}_{3\times3} - \text{diag}(\hat{\mathbf{s}})$ and $\delta\hat{\mathcal{T}}\,(\hat{\boldsymbol{\epsilon}})^{\mathsf{T}} = \mathbf{I}_{3\times3} + \left[\hat{\boldsymbol{\epsilon}}^{\times}\right]$ when linearized, we find that

$$\begin{aligned} \hat{E}'_{\hat{\xi}}(\tilde{\mathbf{y}}) &= \begin{bmatrix} -\text{diag}(\delta\hat{\mathcal{T}}(\hat{\boldsymbol{\epsilon}})^{\mathsf{T}}\tilde{\mathbf{y}}) & \hat{\mathcal{S}}^{-1}\left[-\tilde{\mathbf{y}}^{\times}\right] & -\hat{\mathcal{S}}^{-1} \end{bmatrix}_{\bar{\mathbf{x}}} \\ &= \begin{bmatrix} -\text{diag}(\bar{\mathbf{y}}) & -\left[\bar{\mathbf{y}}^{\times}\right] & -\mathbf{I}_{3\times3} \end{bmatrix}. \end{aligned} \tag{3.59}$$

Comparing Eq. (3.59) with Eq. (3.25) on page 37, we see that for a given vector,

$$\hat{E}'_{\hat{\xi}}(\mathbf{z}) = -E_{\xi}(\mathbf{z}). \tag{3.60}$$

That these matrices are additively inverse is both desirable and expected from the formulation of the nonlinear error perturbation and correction equations. The navigation filter is designed to negate the effects of errors that are being estimated, so for perfect navigation, we expect $E_{\xi}(\mathbf{z})\boldsymbol{\xi} + \hat{E}'_{\hat{\xi}}\hat{\boldsymbol{\xi}} = \mathbf{0}$. For simplicity we will write $-E_{\xi}$ for $\hat{E}'_{\hat{\xi}}$ unless the latter form is desired for unambiguity. Therefore, the generic linearization of the error-correction function $\hat{\mathbf{z}} = \hat{\mathbf{e}}^{-1}(\mathbf{z}', \hat{\boldsymbol{\xi}}_z)$ is

$$\delta\hat{\mathbf{z}} = \delta\mathbf{z}' + \hat{E}'_{\hat{\xi}}(\bar{\mathbf{z}})\hat{\boldsymbol{\xi}}_z = \delta\mathbf{z}' - E_{\xi}(\bar{\mathbf{z}})\hat{\boldsymbol{\xi}}_z \tag{3.61}$$

Linearizing Eqs. (3.51) and (3.52) using the same pattern as for the true vehicle dynamics, we have

$${}^{i}\delta\dot{\hat{\mathbf{r}}} = {}^{i}\delta\hat{\mathbf{v}} \tag{3.62}$$

$${}^{i}\delta\dot{\hat{\mathbf{v}}} = {}^{i}_{b}\bar{\mathcal{T}}\left\{{}^{b}\delta\tilde{\mathbf{a}}_{\text{accmtr}} - E_{\xi}({}^{b}\bar{\mathbf{a}}_{\text{accmtr}})\hat{\boldsymbol{\xi}}_{\text{accmtr}} - \left[{}^{b}\bar{\mathbf{a}}_{\text{thr}}^{\times}\right]{}^{b}\delta\hat{\boldsymbol{\theta}}\right\} + \mathbb{G}_{r}{}^{i}\delta\hat{\mathbf{r}} + {}^{i}\hat{\mathbf{g}}_{d}. \tag{3.63}$$

For the model of the onboard filter, we treat accelerometer data as deterministic, and define its perturbation as ${}^{b}\delta\tilde{\mathbf{a}}_{\text{accmtr}} = \mathbf{0}$. Otherwise, for the augmented state dynamics, we use the definition of ${}^{b}\delta\tilde{\mathbf{a}}_{\text{accmtr}}$ from Eq. (3.47).

3.3.3 Filter Vehicle Rotational Dynamics

The quaternion dynamics are the same as with the truth state,

$$ {}^i_b\dot{\hat{\mathbf{q}}} = \frac{1}{2}{}^b\hat{\omega}' \otimes {}^i_b\hat{\mathbf{q}}, \tag{3.64} $$

where ${}^b\hat{\omega}'$ is the estimated rotation rate vector, augmented with a zero fourth element to create a non-unit quaternion. The $\otimes$ (O-times) operator is defined in Eqs. (3.21) and (3.22).

As with the truth state, we define the filter attitude dispersion as the small-angle rotation vector ${}^b\delta\hat{\boldsymbol{\theta}}$ such that $\mathcal{T}({}^b_i\hat{\mathbf{q}}) = \delta\mathcal{T}({}^b\delta\hat{\boldsymbol{\theta}})\bar{\mathcal{T}}({}^b_i\bar{\mathbf{q}})$. The direction cosine matrix for corresponding to this transformation is approximated

$$ \delta\mathcal{T}({}^b\delta\hat{\boldsymbol{\theta}}) = \mathbf{I}_{3\times3} - \left[{}^b\delta\hat{\boldsymbol{\theta}}^{\times}\right]. \tag{3.65} $$

As ${}^b\delta\hat{\boldsymbol{\theta}}$ defines the nominal-body-to-perturbed-body estimated rotation vector, the expression for the estimated perturbed-body-to-inertial transformation is ${}^i_b\hat{\mathcal{T}} = \mathcal{T}({}^b_i\hat{\mathbf{q}})^{\mathsf{T}} = \mathcal{T}({}^b_i\bar{\mathbf{q}})^{\mathsf{T}}\delta\mathcal{T}({}^b\delta\hat{\boldsymbol{\theta}})^{\mathsf{T}}$, and

$$ {}^i_b\hat{\mathcal{T}} = {}^i_b\bar{\mathcal{T}}\delta\mathcal{T}\left(-{}^b\delta\hat{\boldsymbol{\theta}}\right) = {}^i_b\bar{\mathcal{T}}\left\{\mathbf{I}_{3\times3} + \left[{}^b\delta\hat{\boldsymbol{\theta}}^{\times}\right]\right\}. \tag{3.66} $$

The dynamic equation for the filter attitude dispersion vector is similar to that of the true vector, Eq. (3.31) on page 38. Without deriving the formula again, we state

$$ {}^b\delta\dot{\hat{\boldsymbol{\theta}}} = {}^b\delta\hat{\omega} - {}^b\bar{\omega} \times {}^b\delta\hat{\boldsymbol{\theta}} \tag{3.67} $$

where the second-order term has been dropped since it is not required. The angular rate dispersion, ${}^b\delta\hat{\omega}$, is defined below.

The estimated body rotation rate, ${}^b\hat{\omega}$, must be handled carefully. Because the gyro, as modeled, outputs a rotation rate vector, ${}^b\tilde{\omega}$, there are at least three possible ways to formulate the filter.

1. Do not include a rotation rate vector as part of the filter state by substituting $\mathbf{e}^{-1}({}^b\tilde{\boldsymbol{\omega}}, \hat{\boldsymbol{\xi}}_{\text{gyro}})$ wherever ${}^b\hat{\boldsymbol{\omega}}$ is required. The disadvantage of this method is that it results in more complicated partial derivatives. Also, the rotation rate uncertainty (that is, its covariance) may not be available since the errors are integrated directly into the quaternion (or rotation vector, later defined).

2. Define ${}^b\hat{\boldsymbol{\omega}}$ as a traditional state with dynamics defined by Euler's equation and filter updates provided by gyro measurements. The problem with this approach is, because the gyro data is assumed to be continuously available, a relatively large computational expense required in propagating then updating the state and covariance every time step. Also, uncertainties in the vehicle inertia make this method less accurate.

3. Define ${}^b\hat{\boldsymbol{\omega}}$ as a state without dynamics, but *corrected* at every time step with gyro data, as in Eq. (2.53). This is the approached used for this research because it offers that advantages of having state covariance information immediate available, while not requiring integration of Euler's equation within the filter. Corrected gyro measurements are used directly as the filter state value, and the covariance is updated to reflect uncertainty in the gyro. The disadvantage of this method is that the filter does not have an opportunity to reduce gyro noise.

As stated, it is desired that

$$ {}^b\hat{\boldsymbol{\omega}} = \mathbf{e}^{-1}({}^b\tilde{\boldsymbol{\omega}}, \hat{\boldsymbol{\xi}}_{\text{gyro}}) \tag{3.68} $$

at all navigation computer update cycles. To accomplish this, we use the state correction formulation, $\hat{\mathbf{x}}_j^{+c} = \hat{\mathbf{x}}_j^{+c} + \hat{\mathbf{d}}(t_j)$, which requires an increment change,

$$ \begin{aligned} {}^b\hat{\boldsymbol{\omega}}^{+c} &= {}^b\hat{\boldsymbol{\omega}}^{-c} + \hat{\mathbf{d}}_\omega \\ {}^b\hat{\boldsymbol{\omega}}^{+c} &= {}^b\hat{\boldsymbol{\omega}}^{-c} + \left[\mathbf{e}^{-1}({}^b\tilde{\boldsymbol{\omega}}, \hat{\boldsymbol{\xi}}_{\text{gyro}}) - {}^b\hat{\boldsymbol{\omega}}^{-c}\right] \\ \hat{\mathbf{d}}_\omega &= \mathbf{e}^{-1}({}^b\tilde{\boldsymbol{\omega}}, \hat{\boldsymbol{\xi}}_{\text{gyro}}) - {}^b\hat{\boldsymbol{\omega}}^{-c}. \end{aligned} \tag{3.69} $$

So long as the rotation rate vector is corrected at every navigation cycle, we may get away with not propagating a rotation rate vector in the filter, so we allow

$$^{b}\dot{\hat{\boldsymbol{\omega}}} = \mathbf{0}_{3\times 1}. \tag{3.70}$$

Linearizing this state's equations obviously is different from those previously done in this chapter, but not difficult. The desired angular velocity dispersion is

$$^{b}\delta\hat{\boldsymbol{\omega}} = {}^{b}\delta\tilde{\boldsymbol{\omega}} - E_{\xi}({}^{b}\bar{\boldsymbol{\omega}})\hat{\boldsymbol{\xi}}_{\text{gyro}} \tag{3.71}$$

where we substitute $^{b}\tilde{\boldsymbol{\omega}}\big|_{\bar{\mathbf{x}}} = {}^{b}\bar{\boldsymbol{\omega}}$ from evaluating Eq. (3.46) on page 42at nominal conditions. The state correction version of this equation is

$$^{b}\delta\hat{\boldsymbol{\omega}}^{+c} = {}^{b}\delta\hat{\boldsymbol{\omega}}^{-c} + \left[{}^{b}\delta\tilde{\boldsymbol{\omega}} - E_{\xi}({}^{b}\bar{\boldsymbol{\omega}})\hat{\boldsymbol{\xi}}_{\text{gyro}} - {}^{b}\delta\hat{\boldsymbol{\omega}}^{-c}\right]. \tag{3.72}$$

The dispersion dynamics are simply

$$^{b}\delta\dot{\hat{\boldsymbol{\omega}}} = \mathbf{0}_{3\times 1}. \tag{3.73}$$

3.3.4 Filter Parameter Dynamics

Although not strictly necessary, we allow the vehicle navigation filter to estimate the same parameter states as are in the truth model. Therefore, the same equations shown in Section 3.1.3 may be used, using filter (hat) states instead of truth states. Noise, however and for obvious reasons, is not intentionally added to the filter dynamics. In general, the filter parameter dynamics are

$$\dot{\hat{p}}_i = -\frac{1}{\hat{\tau}_i}\hat{p}_i, \tag{3.74}$$

where we may allow a different time constant for the truth and filter parameter states.

Again, these time constants are not part of the state vector. For distance correlated parameters, we model

$$\begin{aligned} \hat{p}_j' &= -\frac{1}{\hat{d}_{\text{corr}}}\hat{p}_j \\ \dot{\hat{p}}_j &= -\frac{\hat{v}_\perp}{\hat{d}_{\text{corr}}}\hat{p}_j, \qquad \text{where} && (3.75) \\ \hat{v}_\perp &= \left\| \hat{\mathbf{v}} - (\hat{\mathbf{i}}_r \cdot \hat{\mathbf{v}})\hat{\mathbf{i}}_r - \boldsymbol{\omega}_{\text{moon}} \times \hat{\mathbf{r}} \right\|. && (3.76) \end{aligned}$$

These filter parameter dynamics are easily linearized as

$$\begin{aligned} \dot{\hat{p}}_i &= -\frac{1}{\hat{\tau}_i}\hat{p}_i, \qquad \text{and} && (3.77) \\ \dot{\hat{p}}_j &= -\frac{\bar{v}_\perp}{\hat{d}_{\text{corr}}}\hat{p}_j, \qquad \text{where} && (3.78) \\ \bar{v}_\perp &= \left\| \bar{\mathbf{v}} - (\bar{\mathbf{i}}_r \cdot \bar{\mathbf{v}})\bar{\mathbf{i}}_r - \boldsymbol{\omega}_{\text{moon}} \times \bar{\mathbf{r}} \right\| && (3.79) \end{aligned}$$

where $\hat{p}_i$ are time-correlated parameters, and $\hat{p}_j$ are distance-correlated parameters.

This chapter has presented the formulation of the dynamic equations of all truth and filter states, with the exception of the miscellaneous state ΔV, its perturbation $\delta\Delta V$, and the filter versions of those states. The state is best approached after we have detailed the guidance and control algorithms, particularly the vector $\mathbf{a}_{\text{thr}}$ and its variants, in the next chapter.

Chapter 4

Linearized Guidance Algorithms

In Section 2.1.2, guidance algorithms that are proposed for Altair lunar landing were presented with their formulations. In this chapter, some of these algorithms will be linearized for use in the Linear Covariance tool.

4.1 General Formulation

The guidance algorithms implemented in the Altair flight computer may be considered to output continuous actuator commands. In a Monte Carlo sense, this output may be considered a member of the vector $\hat{\mathbf{u}}$ in Eq. (2.41) and the algorithms part of the function $\hat{\mathbf{g}}$. As is expected, the commands issued by the flight computer are dependent exclusively upon the flight computer's estimate of the vehicle's state (and clock if not a part of the state).

The continuous commands are then acted upon by the appropriate vehicle actuators, in this case the throttle control for the Altair Descent Module Main Propulsion (DMMP) system, controlling the magnitude of the main engine thrust, and the Descent Module Attitude Control System (DMACS), controlling the thrust direction. The response of the vehicle is modeled by Eq. (2.34), which may include error sources that perturb the vehicle's motion relative to the desired motion calculated by the flight computer. Specifically, Eqs. (3.16) and (3.18) model the translational and rotational responses to ${}^{b}\hat{\mathbf{a}}_{\text{cmd}}$ and ${}^{b}\hat{\mathbf{M}}_{\text{cmd}}$, respectively.

The flight computer anticipates and estimates the true error sources $\boldsymbol{\xi}$ and accounts for them before issuing a command. The computer also corrects for other forces acting on the vehicle, such as gravity. These equations are also considered part of the function $\hat{\mathbf{g}}$.

As already described in Section 3.1.2, the true thrust-acceleration vector, $\mathbf{a}_{\text{thr}} \equiv \mathbf{F}_{\text{thr}}/m(t)$, is also called the true *specific thrust*. Since mass is assumed to be known deterministically and is not a state variable, thrust may be expressed as either the actual force or as the resultant acceleration. In this section, references to *thrust* actually refer to the *thrust-acceleration.*

The thrust-acceleration vector is nominally the commanded acceleration vector from the guidance routine, $\hat{\mathbf{a}}_{\text{cmd}}$. As suggested by Eq. (3.9), accounting for misalignment, scale factors, and bias, however, the true thrust-acceleration vector may be written to first-order accuracy in the body frame as

$$ {}^{b}\mathbf{a}_{\text{thr}} = \mathbf{e}({}^{b}\hat{\mathbf{a}}_{\text{cmd}}, \boldsymbol{\xi}_{\text{thr}}) + \mathbf{w}_{\text{thr}}, \tag{4.1}$$

where the error function $\mathbf{e}$ is described in Eqs. (3.10)-(3.14). The final term, $\mathbf{w}_{\text{thr}}$, is a 3-dimensional vector of zero-mean white-noise with covariance

$$ \mathrm{E}\left[\mathbf{w}_{\text{thr}}(t)\mathbf{w}_{\text{thr}}^{\mathsf{T}}(t')\right] = S_{w_{\text{thr}}}(t)\,\delta(t-t'), \tag{4.2}$$

where δ is the Dirac delta function.

A well designed GN&C system will attempt to estimate these error sources and account for them when commanding a vehicle orientation and thrust magnitude. We will describe more thoroughly the formulation of the guidance laws used for the present research later in this chapter, but for now let us generically call the total acceleration vector required by the guidance law to achieve the desired trajectory ${}^{i}\hat{\mathbf{a}}_{\text{guid}}$, which is generally a function of the vehicle state for closed-loop guidance schemes. This acceleration is what we desire ${}^{i}\dot{\mathbf{v}}$ to be, so solving Eq. (3.16) using our navigation filter knowledge, we desire

$$ {}^{b}\hat{\mathbf{a}}_{\text{ideal}} = \mathbf{e}^{-1}\left({}^{b}_{i}\hat{\mathcal{T}}\left[{}^{i}\hat{\mathbf{a}}_{\text{guid}} - \left(-\frac{\mu}{\hat{r}^{3}}{}^{i}\hat{\mathbf{r}}\right) - {}^{i}\hat{\mathbf{g}}_{d}\right], \hat{\boldsymbol{\xi}}_{\text{thr}}\right). \tag{4.3}$$

where the symbols with hats represent the navigation filter estimates of the corresponding truth states, $\mathbf{e}^{-1}$ is described by Eqs. (3.54)-(3.57), and ${}^{b}_{i}\hat{\mathcal{T}} = \delta\mathcal{T}({}^{b}\delta\hat{\boldsymbol{\theta}})\mathcal{T}({}^{b}_{i}\bar{\mathbf{q}})$ is the 3 ×

3 inertial-to-body transformation matrix corresponding to the inertial-to-body *estimated* attitude. We leave the negative on the gravitational acceleration term because it will help clarify why the sign of the gravity gradient matrix is negative below.

Unfortunately, the GN&C cannot command that thrust vector because we assume that the DMMP engine is fixed with respect to the lander's body frame and the engine is not gimballed (i.e. there is no thrust vectoring). So, the portion of the thrust command in the direction of the engine is

$$\hat{a}_{\text{cmd}} = {}^b\mathbf{i}_{\text{eng}} \cdot {}^b\hat{\mathbf{a}}_{\text{ideal}}, \qquad \text{or} \tag{4.4}$$

$${}^b\hat{\mathbf{a}}_{\text{cmd}} = {}^b\mathbf{i}_{\text{eng}} \left[{}^b\mathbf{i}_{\text{eng}} \cdot {}^b\hat{\mathbf{a}}_{\text{ideal}}\right] \tag{4.5}$$

where ${}^b\mathbf{i}_{\text{eng}}$ is the unit vector (coordinatized and fixed in the body frame) in the direction of the engine thrust, that is, anti-parallel to the direction of the plume. At this time we do not consider the possibility of a negative thrust command since we expect dispersions off the nominal attitude to be small, and we must assume a completely (non-quantized) throttleable engine because of our linearity restrictions.

Before linearizing Eq. (4.5), it is worthwhile to note that the nominal ideal acceleration vector, found by evaluating Eq. (4.3) on the nominal, is

$${}^b\bar{\mathbf{a}}_{\text{ideal}} = {}^b_i\bar{\mathcal{T}} \left[{}^i\bar{\mathbf{a}}_{\text{guid}} - \left(-\frac{\mu}{\bar{r}^3}{}^i\bar{\mathbf{r}}\right)\right], \tag{4.6}$$

and when the ideal acceleration vector is aligned with the engine axis, as is the case for the unperturbed trajectory, we know

$${}^b\bar{\mathbf{a}}_{\text{ideal}} = {}^b\bar{\mathbf{a}}_{\text{cmd}} = {}^b\bar{\mathbf{a}}_{\text{thr}}. \tag{4.7}$$

Then, Eq. (4.3) may be linearized as

$${}^b\delta\hat{\mathbf{a}}_{\text{ideal}} = \left[{}^b\bar{\mathbf{a}}_{\text{thr}}^{\times}\right] {}^b\delta\hat{\boldsymbol{\theta}} + {}^b_i\bar{\mathcal{T}} \left[{}^i\delta\hat{\mathbf{a}}_{\text{guid}} - \mathbb{G}_r\,{}^i\delta\hat{\mathbf{r}} - {}^i\hat{\mathbf{g}}_d\right] - E_{\zeta}({}^b\bar{\mathbf{a}}_{\text{thr}})\hat{\boldsymbol{\zeta}}_{\text{thr}}. \tag{4.8}$$

Then, the thrust vector command Eq. (4.5) is easily linearized:

$$\begin{aligned} {}^{b}\delta\hat{\mathbf{a}}_{\text{cmd}} &= {}^{b}\mathbf{i}_{\text{eng}}{}^{b}\mathbf{i}_{\text{eng}}^{\mathsf{T}}\,{}^{b}\delta\hat{\mathbf{a}}_{\text{ideal}} \\ &= {}^{b}\mathbf{i}_{\text{eng}}{}^{b}\mathbf{i}_{\text{eng}}^{\mathsf{T}}\left\{\left[{}^{b}\bar{\mathbf{a}}_{\text{thr}}^{\times}\right]{}^{b}\delta\hat{\boldsymbol{\theta}} + {}^{b}_{i}\bar{\mathcal{T}}\left[{}^{i}\delta\hat{\mathbf{a}}_{\text{guid}} - \mathbb{G}_{r}{}^{i}\delta\hat{\mathbf{r}} - {}^{i}\hat{\mathbf{g}}_{d}\right] - E_{\xi}({}^{b}\bar{\mathbf{a}}_{\text{thr}})\hat{\boldsymbol{\xi}}_{\text{thr}}\right\} \end{aligned} \quad (4.9)$$

where $\mathbb{G}_r$ is defined in Eq. (3.34) on page 39 and E_ξ in Eq. (3.25) on page 37and also Eq. (3.59) on page 46. The guidance dispersion, ${}^{i}\delta\hat{\mathbf{a}}_{\text{guid}}$, will naturally depend on the specific guidance law being used. The linearization of the guidance laws will be covered later in the chapter.

As suggested above, we may not always have the ideal thrust vector aligned with the engine axis, and the commanded vector will change with time. The attitude control system (ACS) is required to perform thrust vector control, so we will create a steering law based on this requirement. This law will leave a single degree of freedom uncontrolled (i.e. roll about the engine axis), but this does not cause a problem in Linear Covariance Analysis.

We may compute a torque command using proportional-derivative error control [2], such that

$${}^{b}\hat{\mathbf{M}}_{\text{ACS}} = {}^{b}I^{-1}\left\{K_{\theta}{}^{b}\Delta\hat{\boldsymbol{\theta}} + K_{\omega}({}^{b}\hat{\boldsymbol{\omega}}_{\text{cmd}} - {}^{b}\hat{\boldsymbol{\omega}})\right\}, \quad (4.10)$$

where ${}^{b}I$ is the vehicle moment and product of inertia tensor in the body frame, assumed to be deterministic for this analysis, and K_θ and K_ω are the proportional and derivative gains, respectively, and may be either scalars or matrices.

The angular error in the thrust vector (that is, the rotation vector required to align the engine axis with the ideal thrust vector) can be found by

$${}^{b}\Delta\hat{\boldsymbol{\theta}} = {}^{b}\mathbf{i}_{\text{eng}} \times {}^{b}\hat{\mathbf{a}}_{\text{ideal}} / \left\|\hat{\mathbf{a}}_{\text{ideal}}\right\| = {}^{b}\mathbf{i}_{\text{eng}} \times {}^{b}\hat{\mathbf{i}}_{a_{\text{ideal}}}. \quad (4.11)$$

The angular rate command, ${}^{b}\hat{\boldsymbol{\omega}}_{\text{cmd}}$, can be thought as the necessary rate of change of the required thrust unit vector, ${}^{b}\dot{\hat{\mathbf{i}}}_{a_{\text{ideal}}}$, in radians per second. This command vector could be either treated as the nominal angular rate or tied to the guidance law—effectively open-loop or closed-loop steering, respectively. In the former case, the derivative term reduces to $K_\omega({}^b\bar{\boldsymbol{\omega}} - {}^b\hat{\boldsymbol{\omega}}) = -K_\omega {}^b\delta\hat{\boldsymbol{\omega}}$. In the latter case, we need to find the time derivative of the unit vector of Eq. (4.3), and then linearize it. While it is not impossible to perform this calculation analytically, particularly if the error parameter terms (${}^i\hat{\mathbf{g}}_d$ and $\hat{\boldsymbol{\xi}}_{\text{thr}}$) are neglected, it is not necessary to do so since the first method is adequate.

An improvement on ${}^b\hat{\boldsymbol{\omega}}_{\text{cmd}} = {}^b\bar{\boldsymbol{\omega}}$ is to use the nominal rotation-rate vector, rotated from the nominal to perturbed body frame. This proves to perform well for the present research, and does not unnecessarily muddle the $\hat{G}_{\hat{x}}$ matrix with low-order terms, as the second method would. So we allow

$$ {}^b\hat{\boldsymbol{\omega}}_{\text{cmd}} = \delta\hat{\mathcal{T}}({}^b\delta\hat{\boldsymbol{\theta}})^b\bar{\boldsymbol{\omega}} = \left\{ \mathbf{I}_{3\times3} - \left[{}^b\delta\hat{\boldsymbol{\theta}}^\times \right] \right\} {}^b\bar{\boldsymbol{\omega}}, \tag{4.12} $$

which causes the derivative term of the steering law to simplify to

$$ K_\omega({}^b\bar{\boldsymbol{\omega}} - {}^b\delta\hat{\boldsymbol{\theta}} \times {}^b\bar{\boldsymbol{\omega}} - {}^b\hat{\boldsymbol{\omega}}) = -K_\omega({}^b\delta\hat{\boldsymbol{\omega}} - {}^b\bar{\boldsymbol{\omega}} \times {}^b\delta\hat{\boldsymbol{\theta}}). $$

Note that the expression inside the parentheses is the first-order form of the Bortz equation [24,26] derived in Section 3.1.2 and is identical to ${}^b\delta\dot{\hat{\boldsymbol{\theta}}}$ from Eq. (3.67). This formulation does not turn Eq. (4.10) into an open-loop law since Eq. (4.11) incorporates the closed-loop guidance law by way of $\hat{\mathbf{a}}_{\text{ideal}}$.

We linearize Eq. (4.10) in steps by applying the chain rule. First, we obtain

$$ {}^b\delta\hat{\mathbf{M}}_{\text{ACS}} = {}^bI^{-1} \left\{ K_\theta \left[{}^b\mathbf{i}_{\text{eng}}^\times \right] {}^b\delta\hat{\mathbf{i}}_{a_{\text{ideal}}} - K_\omega \left({}^b\delta\hat{\boldsymbol{\omega}} - \left[{}^b\bar{\boldsymbol{\omega}}^\times \right] {}^b\delta\hat{\boldsymbol{\theta}} \right) \right\}. $$

A unit vector $\mathbf{i}_x$ is linearized to $\delta\mathbf{i}_x = \frac{1}{\bar{x}} \left[\mathbf{I} - \bar{\mathbf{i}}_x \bar{\mathbf{i}}_x^\mathsf{T} \right] \delta\mathbf{x}$, and $\bar{\mathbf{i}}_{a_{\text{ideal}}} = \mathbf{i}_{\text{eng}}$ so the equation above becomes

$$ {}^b\delta\hat{\mathbf{M}}_{\text{ACS}} = {}^bI^{-1} \left\{ \frac{K_\theta}{\bar{a}_{\text{thr}}} \left[{}^b\mathbf{i}_{\text{eng}}^\times \right] \left[\mathbf{I}_{3\times3} - {}^b\mathbf{i}_{\text{eng}} {}^b\mathbf{i}_{\text{eng}}^\mathsf{T} \right] {}^b\delta\hat{\mathbf{a}}_{\text{ideal}} - K_\omega \left({}^b\delta\hat{\boldsymbol{\omega}} - \left[{}^b\bar{\boldsymbol{\omega}}^\times \right] {}^b\delta\hat{\boldsymbol{\theta}} \right) \right\}, \tag{4.13} $$

where ${}^b\delta\hat{\mathbf{a}}_{\text{ideal}}$ is given in Eq. (4.8).

4.2 Targets and t_{go}

In the previous section, we referred to the guidance system-calculated vector ${}^{i}\hat{\mathbf{a}}_{\text{guid}}$ to determine the thrust throttle and attitude control system commands. In general we can describe this vector for a closed-loop guidance system as

$$ {}^{i}\hat{\mathbf{a}}_{\text{guid}} = \hat{\mathbf{a}}(\hat{\mathbf{x}}, \mathbf{x}_t, t_{\text{go}}), \tag{4.14} $$

where $\hat{\mathbf{x}}$ is the current state, $\mathbf{x}_t$ is the target state, and t_{go} is the time-to-go or the time remaining to achieve the target state.

The most critical parameter is the time-to-go t_{go}, and in order to prevent instability just prior to phase changes, the fine-count method must be used, that is t_{go} is not allowed to drop below a certain value. Ironically, the actual guidance target does not affect the Linear Covariance analysis since the target is not included in the state. The exception to this is if, for example during the terminal descent phase, the target is the landing site, with its position coordinates part of the truth and filter state which would be estimated by the navigation filter.

It is reasonable to consider that the target state of a particular guidance phase, $\mathbf{x}_t$, could fall into one of three categories:

1. $\mathbf{x}_t$ is a static inertial position and velocity target which is to be achieved at some fixed time t_t, with the time-to-go $t_{\text{go}} = t_t - t$; or

2. $\mathbf{x}_t$ is a relative state (e.g. position and velocity) described as an offset (or delta-position and delta-velocity) from the current estimate of the landing site, that is $\mathbf{x}_t = \hat{\mathbf{x}}_{\text{site}} + \Delta\mathbf{x}_{\text{rel}}$, with the target time t_t either fixed or changed based on the updates to $\hat{\mathbf{x}}_{\text{site}}$ and $\mathbf{x}_t$; or

3. $\mathbf{x}_t$ is a relative state which is determined based on the current estimated vehicle and landing site states, $\mathbf{x}_t = \mathbf{f}(\hat{\mathbf{x}}_{\text{veh}}, \hat{\mathbf{x}}_{\text{site}}, t)$, and again t_t either fixed or updated with $\mathbf{x}_t$.

Without any explicit information about the targets or target times, and only a nominal trajectory to work from, assumptions must be made in order to perform any analysis.

Let us consider each possibility individually. Option 1 is the simplest for linear covariance analysis because information about the target states is not required—only the target time. This option seems unlikely, however, because of the array of instruments aboard Altair designed to estimate the landing site position relative to the vehicle. Option 1 is akin to flying blindly to a predesignated set of inertial coordinates without regard to any uncertainty in that position or dispersion in time causing that location on the surface to move.

Option 2 is a more reasonable alternative which uses the current navigation data to correct the target state if needed. The consequence of using this method of describing $\mathbf{x}_t$ is that equations describing the relationship between the guidance law and $\mathbf{x}_{\text{site}}$ must be included in the linear covariance matrices, but this is a relatively simple matter with the guidance laws used for this research.

Option 3 regards the target state as an function of estimated landing site and vehicle states and of time. As the estimated landing site location and vehicle states change, the target state, and possibly the target time to achieve it, must also change according to the function $\mathbf{f}$. Obviously $\mathbf{f}$ must be known in order determine $\mathbf{x}_t$. Option 2 is a special case of Option 3, where the function is known and linear.

Using Option 2 definition of $\mathbf{x}_t$ above, we can define the target state as

$$\begin{bmatrix} {}^{i}\mathbf{r}_t \\ {}^{i}\mathbf{v}_t \\ {}^{i}\mathbf{a}_t \end{bmatrix} = \begin{bmatrix} {}^{i}\hat{\mathbf{r}}_{\text{site}} \\ {}^{i}\hat{\mathbf{v}}_{\text{site}} \\ {}^{i}\hat{\mathbf{a}}_{\text{site}} \end{bmatrix} + \begin{bmatrix} {}^{i}\Delta\mathbf{r}_{\text{rel}} \\ {}^{i}\Delta\mathbf{v}_{\text{rel}} \\ {}^{i}\Delta\mathbf{a}_{\text{rel}} \end{bmatrix}, \tag{4.15}$$

which may readily be substituted into the generic Eq. (4.14). As seen in Table 3.1 on page 29, however, the only landing site state in the LinCov navigation state is ${}^{i}\hat{\mathbf{r}}_{\text{site}}$. This is not a problem because the site's velocity and acceleration may be expressed in terms of its position, as shown in Eqs. (3.2) and (3.3) on page 30.

We may now rewrite Eq. (4.15) as

$$\begin{bmatrix} {}^i\mathbf{r}_t \\ {}^i\mathbf{v}_t \\ {}^i\mathbf{a}_t \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3\times3} \\ \left[{}^i\boldsymbol{\omega}^\times_{\text{moon}}\right] \\ \left[{}^i\boldsymbol{\omega}^\times_{\text{moon}}\right]\left[{}^i\boldsymbol{\omega}^\times_{\text{moon}}\right] \end{bmatrix} {}^i\hat{\mathbf{r}}_{\text{site}} + {}^i\Delta\mathbf{x}_{\text{rel}}, \tag{4.16}$$

where ${}^i\boldsymbol{\omega}^\times_{\text{moon}} = \left[{}^i\boldsymbol{\omega}_{\text{moon}}\right]^\times$ is the cross-product matrix of $\boldsymbol{\omega}_{\text{moon}}$. For simplicity, we will assume that ${}^i\Delta\mathbf{x}_{\text{rel}}$ is a fixed vector programmed *a priori* into the GN&C. The use of this formulation changes the analysis from using an inertial guidance system to a relative guidance system.

4.3 Proportional-Derivative (PD) Guidance

Perhaps the simplest guidance system is one that seeks to eliminate all dispersions as quickly as possible, usually designed in manner that trades propellant usage and vehicle response with time and known capability limits. Strictly speaking, this method of control is not closed-loop guidance and is more accurately described as *error correction.* As we are trying to control position and velocity simultaneously, we have what is traditionally known as a PD control law (for Proportional-Derivative).

A generic PD control law for translational motion control is simply to modify the nominal pre-programmed acceleration command to drive dispersions to zero and restore the state to the nominal, by

$$\begin{aligned} \underset{\text{PD Law}}{{}^i\delta\hat{\mathbf{a}}_{\text{guid}}} &= K_r\left({}^i\bar{\mathbf{r}} - {}^i\hat{\mathbf{r}}\right) + K_v\left({}^i\bar{\mathbf{v}} - {}^i\hat{\mathbf{v}}\right) \\ \underset{\text{PD Law}}{{}^i\delta\hat{\mathbf{a}}_{\text{guid}}} &= -K_r{}^i\delta\hat{\mathbf{r}} - K_v{}^i\delta\hat{\mathbf{v}}. \end{aligned} \tag{4.17}$$

K_r and K_v are gains (either scalar or matrix) designed to produce the desired error correction without violating limits or restrictions on vehicle dynamics.

It should be clear that a law such as Eq. (4.17) will restore the vehicle the *inertial* reference trajectory, provided the gains selected produce a stable response. Specific emphasis with Altair and ALHAT is on *relative* navigation and guidance. If the Terrain Relative Navigation instruments determine that the desired landing site is actually offset so-many meters in that direction, the vehicle needs to respond by steering into a path so-many meters in that direction. We can think of landing site position dispersions as deltas to offset the reference trajectory, such that the new reference trajectory resembles ${}^{i}\bar{\mathbf{r}}' = {}^{i}\bar{\mathbf{r}} + {}^{i}\delta\hat{\mathbf{r}}_{\text{site}}$. Then our PD law becomes

$$\begin{aligned}
\underset{\text{Rel PD Law}}{{}^{i}\delta\hat{\mathbf{a}}_{\text{guid}}} &= K_r\left({}^{i}\bar{\mathbf{r}} + {}^{i}\delta\hat{\mathbf{r}}_{\text{site}} - {}^{i}\hat{\mathbf{r}}\right) + K_v\left({}^{i}\bar{\mathbf{v}} + {}^{i}\delta\hat{\mathbf{v}}_{\text{site}} - {}^{i}\hat{\mathbf{v}}\right) \\
&= K_r\left({}^{i}\delta\hat{\mathbf{r}}_{\text{site}} - {}^{i}\delta\hat{\mathbf{r}}\right) + K_v\left({}^{i}\delta\hat{\mathbf{v}}_{\text{site}} - {}^{i}\delta\hat{\mathbf{v}}\right).
\end{aligned}$$

Remembering that we do not actually have a ${}^{i}\delta\hat{\mathbf{v}}_{\text{site}}$ state, we refer to Eq. (3.4), and rewrite the law as

$$\underset{\text{Rel PD Law}}{{}^{i}\delta\hat{\mathbf{a}}_{\text{guid}}} = -K_r{}^{i}\delta\hat{\mathbf{r}} - K_v{}^{i}\delta\hat{\mathbf{v}} + \left(K_r + K_v\left[{}^{i}\boldsymbol{\omega}^{\times}_{\text{moon}}\right]\right){}^{i}\delta\hat{\mathbf{r}}_{\text{site}}. \tag{4.18}$$

4.4 Linear and Quadratic Acceleration Profiles

Let us consider the continuous linear acceleration profile given by Sostaric in Eq. (2.12) on page 11 and D'Souza in Eq. (2.20) on page 15. Given in matrix-vector form, this law may be expressed as

$$\underset{\text{Linear Accel}}{{}^{i}\hat{\mathbf{a}}_{\text{guid}}} = \begin{bmatrix} -\frac{6}{t_{\text{go}}^2}\mathbf{I} & -\frac{4}{t_{\text{go}}}\mathbf{I} \end{bmatrix}\begin{bmatrix} {}^{i}\hat{\mathbf{r}} \\ {}^{i}\hat{\mathbf{v}} \end{bmatrix} + \begin{bmatrix} \frac{6}{t_{\text{go}}^2}\mathbf{I} & -\frac{2}{t_{\text{go}}}\mathbf{I} \end{bmatrix}\begin{bmatrix} {}^{i}\mathbf{r}_t \\ {}^{i}\mathbf{v}_t \end{bmatrix}, \tag{4.19}$$

where $\mathbf{I} = \mathbf{I}_{3\times 3}$. We may substitute a reduced form of Eq. (4.15) on page 57 and linearize to obtain

$$\underset{\text{Linear Accel}}{{}^{i}\delta\hat{\mathbf{a}}_{\text{guid}}} = \begin{bmatrix} -\frac{6}{t_{\text{go}}^2}\mathbf{I} & -\frac{4}{t_{\text{go}}}\mathbf{I} \end{bmatrix}\begin{bmatrix} {}^{i}\delta\hat{\mathbf{r}} \\ {}^{i}\delta\hat{\mathbf{v}} \end{bmatrix} + \left\{\frac{6}{t_{\text{go}}^2}\mathbf{I} - \frac{2}{t_{\text{go}}}\left[{}^{i}\boldsymbol{\omega}^{\times}_{\text{moon}}\right]\right\}{}^{i}\delta\hat{\mathbf{r}}_{\text{site}}. \tag{4.20}$$

The relative offset vector ${}^{i}\Delta\mathbf{x}_{\text{rel}}$ is eliminated during linearization since it is unchanged from a nominal trajectory to a perturbed trajectory. Since Eq. (4.19) is linear, we see essentially no change in the form of the equation after linearization. We leave Eq. (4.20) in its matrix multiplication form to ease in extracting partial derivatives for elements of the matrix $\hat{G}_{\hat{x}}$.

Likewise, the continuous quadratic acceleration profile of Klummp in Eq. (2.3) and Sostaric in Eq. (2.13) may be written

$$\underset{\text{Quadratic Accel}}{{}^{i}\hat{\mathbf{a}}_{\text{guid}}} = \begin{bmatrix} -\frac{12}{t_{\text{go}}^{2}}\mathbf{I} & -\frac{6}{t_{\text{go}}}\mathbf{I} \end{bmatrix} \begin{bmatrix} {}^{i}\hat{\mathbf{r}} \\ {}^{i}\hat{\mathbf{v}} \end{bmatrix} + \begin{bmatrix} \frac{12}{t_{\text{go}}^{2}}\mathbf{I} & -\frac{6}{t_{\text{go}}}\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} {}^{i}\mathbf{r}_{t} \\ {}^{i}\mathbf{v}_{t} \\ {}^{i}\mathbf{a}_{t} \end{bmatrix}. \tag{4.21}$$

The linearized equation, in terms of the landing site, is quickly found to be

$$\underset{\text{Quadratic Accel}}{{}^{i}\delta\hat{\mathbf{a}}_{\text{guid}}} = \begin{bmatrix} -\frac{12}{t_{\text{go}}^{2}}\mathbf{I} & -\frac{6}{t_{\text{go}}}\mathbf{I} \end{bmatrix} \begin{bmatrix} {}^{i}\delta\hat{\mathbf{r}} \\ {}^{i}\delta\hat{\mathbf{v}} \end{bmatrix} + \left\{ \frac{12}{t_{\text{go}}^{2}}\mathbf{I} - \left(\frac{6}{t_{\text{go}}}\mathbf{I} + \left[{}^{i}\boldsymbol{\omega}_{\text{moon}}^{\times}\right] \right) \left[{}^{i}\boldsymbol{\omega}_{\text{moon}}^{\times}\right] \right\} {}^{i}\delta\hat{\mathbf{r}}_{\text{site}}. \tag{4.22}$$

4.5 Other Guidance Laws

The methods demonstrated in this chapter can also be applied to other guidance or steering laws, modified as required for the law. In general, what is required is an equation for the control (or command) variable or vector in terms of the initial (or current) state and the desired state, or in terms of variables that are functions of those states. Some laws are given as a time history from the initial state to the final state with fixed coefficients; it may be beneficial to re-derive or take the limit as $t \to 0$ of these laws, so they are not functions of time but rather time-varying states. This was done for the linear and quadratic acceleration laws in Eq. (2.12) and Eq. (2.13) when both laws were reduced to $\mathbf{a} = \mathbf{c}_0$, where $\mathbf{c}_0$ is expressed in terms of the current and target states. Recursive or iterative algorithms may be approached by linearizing the equations, and solving the implicit equations.

For instance, using the linear tangent law in Eq. (2.16), the thrust-acceleration magnitude $a(t)$ is known, the control variable is β, the pitch angle, and the coefficients c and

β_0 are determined by the boundary conditions of the problem (i.e. the initial current estimated state and the target estimated state). Begin by evaluating the partial derivative of the law with respect to the navigation state.

$$\begin{aligned} \frac{\partial \beta}{\partial \hat{\mathbf{x}}} \sec^2 \beta &= \frac{\partial \beta_0}{\partial \hat{\mathbf{x}}} \sec^2 \beta_0 - \frac{\partial c}{\partial \hat{\mathbf{x}}} t \\ \frac{\partial \beta}{\partial \hat{\mathbf{x}}} &= \frac{\partial \beta_0}{\partial \hat{\mathbf{x}}} \left(\frac{\cos \beta}{\cos \beta_0} \right)^2 - \frac{\partial c}{\partial \hat{\mathbf{x}}} t \cos^2 \beta \end{aligned} \tag{4.23}$$

As $t \to 0$, $\beta \to \beta_0$ so we have, using the coefficients in the bilinear tangent law, Eq. (2.15),

$$\begin{aligned} \frac{\partial \beta}{\partial \hat{\mathbf{x}}} = \frac{\partial \beta_0}{\partial \hat{\mathbf{x}}} &= \frac{\partial}{\partial \hat{\mathbf{x}}} \left(\arctan \frac{c_4}{c_3} \right) \\ &= \frac{1}{\left(\frac{c_4}{c_3}\right)^2 + 1} \left[\frac{c_3 \frac{\partial c_4}{\partial \hat{\mathbf{x}}} - c_4 \frac{\partial c_3}{\partial \hat{\mathbf{x}}}}{c_3^2} \right] \\ &= \frac{c_3^2}{c_3^2 + c_4^2} \left[\frac{c_3 \frac{\partial c_4}{\partial \hat{\mathbf{x}}} - c_4 \frac{\partial c_3}{\partial \hat{\mathbf{x}}}}{c_3^2} \right] \\ &= \frac{c_3 \frac{\partial c_4}{\partial \hat{\mathbf{x}}} - c_4 \frac{\partial c_3}{\partial \hat{\mathbf{x}}}}{c_3^2 + c_4^2} \\ \frac{\partial \beta}{\partial \hat{\mathbf{x}}} &= \left(\frac{-c_4}{c_3^2 + c_4^2} \right) \frac{\partial c_3}{\partial \hat{\mathbf{x}}} + \left(\frac{c_3}{c_3^2 + c_4^2} \right) \frac{\partial c_4}{\partial \hat{\mathbf{x}}} \end{aligned} \tag{4.24}$$

where the coefficients c_3 and c_4 are calculated from the vehicle current state and known acceleration profile.

Other guidance laws, such as Powered Explicit Guidance [12] (PEG) and cross-product steering [15], are reliant on the velocity-to-be-gained vector, $\mathbf{v}_{\text{go}}$. Simply stated, the velocity-to-be-gained vector may be expressed

$$\mathbf{v}_{\text{go}} = \mathbf{v}_t - \hat{\mathbf{v}}. \tag{4.25}$$

And the cross-product steering law may be expressed

$$\frac{d\mathbf{v}_{\text{go}}}{dt} \times \mathbf{v}_{\text{go}} = \mathbf{0}. \tag{4.26}$$

Obviously, more equations are required to form a working guidance law, but if the law can be expressed in terms of the velocity-to-be-gained (or its unit vector), the final derivative in the chain rule expansion is simply $\partial \mathbf{v}_{\text{go}}/\partial \hat{\mathbf{v}} = -\mathbf{I}_{3\times 3}$. Other vehicle states will likely also play a part.

None of these guidance laws shown in this section are a part of the current research, although they provide a starting point for future research.

Chapter 5

Linear Covariance Analysis

Linear Covariance (or LinCov) Analysis is at the center of this research. Much of the theory that is required to understand how LinCov works has been described in Section 2.3, and may be found elsewhere [1,2,15,18,21]. The foundation of this work is that of Geller, with some modifications.

5.1 Modified LinCov Formulation

The nonlinear equations describing the dynamic system are given in Eqs. (2.34)-(2.42) on page 20. Let us modify the equation describing the continuous inertial measurements (2.35) so we may anticipate the measurement based on our current filter state knowledge. The appropriate nonlinear and linear forms of the equation become

$$\tilde{\mathbf{y}} = \mathbf{c}(\mathbf{x},\, \hat{\mathbf{x}},\, t) + \boldsymbol{\eta} \tag{5.1}$$

$$\delta\tilde{\mathbf{y}} = C_x \delta\mathbf{x} + C_{\hat{x}} \delta\hat{\mathbf{x}} + \boldsymbol{\eta}. \tag{5.2}$$

The dispersed filter state dynamics in Eq. (2.49) are rewritten

$$\begin{aligned} \delta\dot{\hat{\mathbf{x}}} &= \hat{F}_{\hat{x}} \delta\hat{\mathbf{x}} + \hat{F}_{\hat{u}} \delta\hat{\mathbf{u}} + \hat{F}_{\tilde{y}} \delta\tilde{\mathbf{y}} \\ &= (\hat{F}_{\hat{x}} + \hat{F}_{\hat{u}} \hat{G}_{\hat{x}} + \hat{F}_{\tilde{y}} C_{\hat{x}}) \delta\hat{\mathbf{x}} + \hat{F}_{\tilde{y}} C_x \delta\mathbf{x} + \hat{F}_{\tilde{y}} \boldsymbol{\eta}, \end{aligned} \tag{5.3}$$

where the new term $C_{\hat{x}}$ is defined $C_{\hat{x}}(t) = \partial\mathbf{c}/\partial\hat{\mathbf{x}}|_{\bar{\mathbf{x}}}$.

The coefficient $\mathcal{F}$ of the augmented state dynamics in Eq. (2.55) is also changed to include the new term,

$$\mathcal{F} = \begin{bmatrix} F_x & F_{\hat{u}} \hat{G}_{\hat{x}} \\ \hat{F}_{\tilde{y}} C_x & \hat{F}_{\hat{x}} + \hat{F}_{\hat{u}} \hat{G}_{\hat{x}} + \hat{F}_{\tilde{y}} C_{\hat{x}} \end{bmatrix}. \tag{5.4}$$

Rather than using Eq. (2.58) on page 24 to propagate the augmented covariance $\mathcal{C}_A$, a discrete form of the Ricatti equation is used

$$\mathcal{C}_{A,i+1} = \Phi(t, t+\Delta t)\mathcal{C}_{A,i}\Phi^{\mathsf{T}}(t, t+\Delta t) + \mathcal{S}\Delta t \tag{5.5}$$

with state transition matrix Φ defined as

$$\Phi(t, t+\Delta t) = e^{\mathcal{F}(t)\Delta t} = \mathbf{I} + \mathcal{F}(t)\Delta t + \frac{1}{2}\mathcal{F}(t)\mathcal{F}(t)\Delta t^2 + \cdots + \frac{1}{n!}\left[\mathcal{F}(t)\right]^n \Delta t^n \tag{5.6}$$

where n is the order of expansion desired, and augmented discrete process noise, as shown in Eqs. (2.55) and (2.58) on page 24, is

$$\mathcal{S} = \mathcal{G}S_\eta\mathcal{G}^{\mathsf{T}} + \mathcal{W}S_w\mathcal{W}^{\mathsf{T}} = \begin{bmatrix} S_w & \mathbf{0}_{n\times\hat{n}} \\ \mathbf{0}_{\hat{n}\times n} & \hat{F}_{\tilde{y}}S_\eta\hat{F}_{\tilde{y}} \end{bmatrix}. \tag{5.7}$$

5.2 Miscellaneous LinCov States

The LinCov tool used by Geller for lunar landing navigation analysis had 99 states each for the truth states and for the navigation states. For the present research, it is desired to know the effect of error states on the thrust ΔV dispersion. "Delta-V" or ΔV is a measure of the magnitude of velocity change required to perform a maneuver or trajectory change, and is a commonly cited performance value when sizing a propulsion system. The ΔV dispersion cannot be back-calculated from the dispersions of the 99 states, and so we require another state be defined (each of the truth and navigation states) assigned to ΔV. The formulations of the truth and filter ΔV states follow in next subsections.

5.2.1 True Delta-V State ΔV

Let us define the state ΔV, a scalar, which is the magnitude of the thrust-acceleration integrated with respect to time, that is

$$\Delta\mathrm{V}(t) = \int_0^t \left\|\mathbf{a}_{\mathrm{thr}}(\tau)\right\| \,\mathrm{d}\tau, \tag{5.8}$$

which may be rewritten as the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\Delta \mathrm{V}\right) = \Delta\dot{\mathrm{V}} = a_{\text{thr}} = \left(\mathbf{a}_{\text{thr}}^{\mathsf{T}}\mathbf{a}_{\text{thr}}\right)^{1/2}, \ \Delta \mathrm{V}(t_0) = 0. \tag{5.9}$$

The true thrust-acceleration vector, $\mathbf{a}_{\text{thr}} \equiv \mathbf{F}_{\text{thr}}/m(t)$, is the true *specific thrust* vector discussed in Section 4.1 on page 51. The equations required to calculate $\Delta\mathrm{V}$ may thus be expected to be related to the commanded thrust acceleration ${}^{b}\hat{\mathbf{a}}_{\text{cmd}}$ and the error sources related to the thrust actuator.

Linearization of Eq. (5.9) is straight-forward since we have done most of the work in the previous two chapters. The dynamic equation of the $\Delta\mathrm{V}$ dispersion (expressed by the awkward symbol $\delta\Delta\dot{\mathrm{V}}$, such that $\Delta\mathrm{V} = \Delta\bar{\mathrm{V}} + \delta\Delta\mathrm{V}$) is

$$\delta\Delta\dot{\mathrm{V}} = \left[\frac{{}^{b}\mathbf{a}_{\text{thr}}^{\mathsf{T}}}{a_{\text{thr}}}\right]_{\bar{\mathbf{x}}} {}^{b}\delta\mathbf{a}_{\text{thr}} = {}^{b}\mathbf{i}_{\text{eng}}^{\mathsf{T}}\,{}^{b}\delta\mathbf{a}_{\text{thr}}. \tag{5.10}$$

We have chosen to expand $\Delta\dot{\mathrm{V}}$ using vectors coordinatized in the vehicle body-fixed frame for convenience. The inertial thrust-acceleration vector, ${}^{i}\mathbf{a}_{\text{thr}}$, is defined in Eq. (3.9) on page 32, and its linearized form is within Eq. (3.33) on page 39, which we include here for clarity:

$${}^{b}\delta\mathbf{a}_{\text{thr}} = {}^{b}\delta\hat{\mathbf{a}}_{\text{cmd}} + E_{\xi}({}^{b}\bar{\mathbf{a}}_{\text{cmd}})\boldsymbol{\xi}_{\text{thr}} + \mathbf{w}_{\text{thr}} \tag{5.11}$$

The combination of Eqs. (5.10) and (5.11) gives us an equation in the form required by Eq. (2.48) for the calculation of F_x and $F_{\hat{u}}$:

$$\delta\Delta\dot{\mathrm{V}} = {}^{b}\mathbf{i}_{\text{eng}}^{\mathsf{T}}\left[{}^{b}\delta\hat{\mathbf{a}}_{\text{cmd}} + E_{\xi}({}^{b}\bar{\mathbf{a}}_{\text{cmd}})\boldsymbol{\xi}_{\text{thr}} + \mathbf{w}_{\text{thr}}\right] \tag{5.12}$$

The elements of $\hat{G}_{\hat{x}}$ required for LinCov for this state may be extracted from the linearized equation modeling ${}^{b}\delta\hat{\mathbf{a}}_{\text{cmd}}$, Eq. (4.9) on page 54, and from equations modeling ${}^{i}\delta\hat{\mathbf{a}}_{\text{guid}}$ in Chapter 4.

It is important to note that Eq. (5.12) includes thrust-acceleration noise ${}^{b}\mathbf{w}_{\text{thr}}$, so we need to be sure to map its noise strength to the $\Delta\mathrm{V}$ state. It can be quickly shown that the thrust-acceleration noise, defined in Eq. (4.2), is mapped to the ΔV state by

$$S_{w_{\Delta V}} = {}^b\mathbf{i}_{\text{eng}}^{\mathsf{T}}\, {}^bS_{w_{\text{thr}}}\, {}^b\mathbf{i}_{\text{eng}}. \tag{5.13}$$

5.2.2 Estimated Delta-V State $\Delta\hat{V}$

Since the thrust-acceleration is modeled as the only non-gravitational disturbance forces present in the lunar descent problem, the vehicle's accelerometer may be used to estimate the thrust ΔV. As we do in Eq. (3.52) to propagate the vehicle's estimated translational state, we correct the continuous accelerometer data using

$${}^b\hat{\mathbf{a}}_{\text{corrected}} = \hat{\mathbf{e}}^{-1}({}^b\tilde{\mathbf{a}}_{\text{accmtr}}, \hat{\boldsymbol{\xi}}_{\text{accmtr}}) \tag{5.14}$$

where ${}^b\tilde{\mathbf{a}} = {}^b\tilde{\mathbf{a}}_{\text{accmtr}}$ is the acceleration vector sensed by the body-fixed accelerometers, and defined by Eqs. (3.44) and (3.45) on page 42. The corrected acceleration magnitude is then propagated by

$$\Delta\dot{\hat{V}} = \left\| {}^b\hat{\mathbf{a}}_{\text{corrected}} \right\|. \tag{5.15}$$

Linearized and combined with Eq. (5.14), Eq. (5.15) becomes

$$\delta\Delta\dot{\hat{V}} = {}^b\bar{\mathbf{i}}_{\text{eng}}^{\mathsf{T}} \left[{}^b\delta\tilde{\mathbf{a}}_{\text{accmtr}} - E_{\xi}({}^b\bar{\mathbf{a}}_{\text{thr}})\boldsymbol{\xi}_{\text{accmtr}} \right]. \tag{5.16}$$

The expression for ${}^b\delta\tilde{\mathbf{a}}_{\text{accmtr}}$ is found in Eq. (3.47) on page 43.

$$\delta\Delta\dot{\hat{V}} = {}^b\bar{\mathbf{i}}_{\text{eng}}^{\mathsf{T}} \left[{}^b\delta\hat{\mathbf{a}}_{\text{cmd}} + E_{\xi}({}^b\bar{\mathbf{a}}_{\text{thr}})\boldsymbol{\xi}_{\text{thr}} + \mathbf{w}_{\text{thr}} + \boldsymbol{\eta}_{\text{accmtr}} \right]$$

5.2.3 Non-monotonicity of ΔV State Variance

As ΔV is a measure of energy and propellant expended, and that it is impossible to reclaim energy or propellant, it is reasonable to conclude that the ΔV states defined above are monotonically increasing, which they are. It may not be expected, however, that the variance of the ΔV states, are *not* monotonic. In fact the dispersion of ΔV can decrease

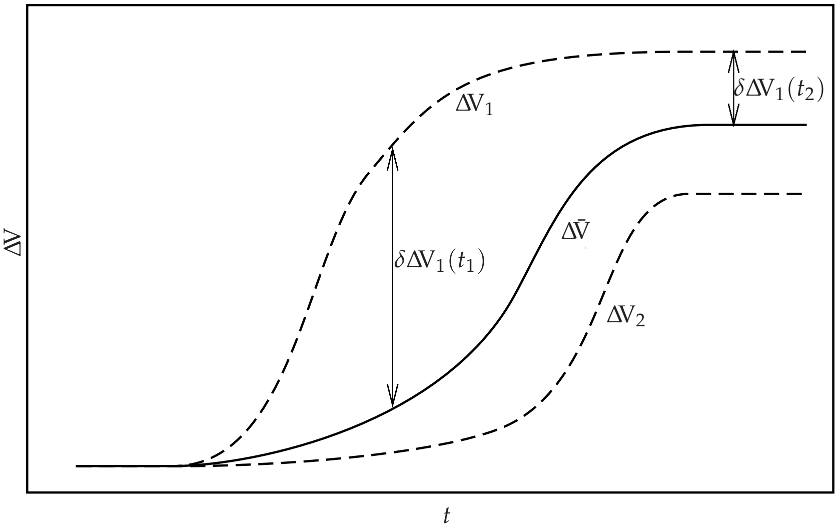

Fig. 5.1: Monotocity of ΔV with the non-monotocity of the ΔV dispersion.

over time, as we shall see visually and mathematically.

On a throttleable propulsion system, such as on the Altair, this decrease in dispersion may be thought of as an increased need for ΔV early in a dispersed maneuver followed later by a decreased need for ΔV. This concept is shown qualitatively in Fig. 5.1. While both ΔV curves are monotonically increasing, the ΔV dispersion, $\delta\Delta V(t) = \Delta V(t) - \Delta\bar{V}(t)$, can both increase and decrease over time. Specifically in the figure, $\delta\Delta V(t_2) < \delta\Delta V(t_1)$ even though $t_2 > t_1$. By imagining many more dashed dispersed curves overlaid on the plot, it could be reasoned that it is also possible that $\sigma_{\Delta V}(t_2) < \sigma_{\Delta V}(t_1)$.

We may use the expected value operator to demonstrate that the variance of ΔV dispersion is not monotonic. From Eq. (5.8), we know that

$$\Delta V(t) = \int_0^t a_{\text{thr}}(\tau)\,\mathrm{d}\tau$$

and that since $t \geq 0$ and $a_{\text{thr}}(t) \geq 0$, $\Delta V(t) \geq 0$. The variance of a scalar variable may be written $\sigma^2 = \mathrm{E}[(x-\bar{x})^2]$, where $\bar{x} = \mathrm{E}[x]$, or equivalently, $\sigma^2 = \mathrm{E}[\delta x^2]$, so

$$\sigma^2_{\Delta V}(t) = \mathrm{E}\left\{\left(\Delta V(t) - \Delta\bar{V}(t)\right)^2\right\} = \mathrm{E}[\delta\Delta V^2]. \tag{5.17}$$

The question is whether the variance (or the standard deviation σ) is monotonic over time, that is whether $\frac{\mathrm{d}}{\mathrm{d}t}\sigma^2_{\Delta V}(t) \geq 0$. Since the expected value operator is linear, it is correct to say that $\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}[X] = \mathrm{E}[\frac{\mathrm{d}}{\mathrm{d}t}X]$, differentiating Eq. (5.17) yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\sigma^2_{\Delta V}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\left\{\left(\Delta V(t) - \Delta\bar{V}(t)\right)^2\right\} = \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\left[\delta \Delta V^2\right]$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\sigma^2_{\Delta V}(t) = 2\left(\mathrm{E}\left[\Delta V\, a_{\mathrm{thr}}\right] - \Delta\bar{V}\, \bar{a}_{\mathrm{thr}}\right) \tag{5.18}$$

$$= 2\mathrm{E}\left[\delta\Delta V\, \delta a_{\mathrm{thr}}\right]. \tag{5.19}$$

which is twice the covariance of ΔV and a_{thr}, or $\frac{\mathrm{d}}{\mathrm{d}t}\sigma^2_{\Delta V}(t) = 2\mathrm{Cov}(\Delta V(t), a_{\mathrm{thr}}(t))$.

Whether Eq. (5.18) or Eq. (5.19) is considered, we may conclude that the ΔV dispersion is not necessarily monotonic. From Eq. (5.18), all four variables on the right-hand side are non-negative for all times by definition, therefore both terms are non-negative and since we cannot know the difference in magnitude between them, $\frac{\mathrm{d}}{\mathrm{d}t}\sigma^2_{\Delta V}$ may be either positive, zero, or negative. From Eq. (5.19), we have two dispersions multiplied, and dispersions may be zero, positive, or negative. Therefore, the product of these dispersions, the expected value, and variance rate of change may be zero, positive, or negative. Both approaches support the statement that the variance and standard deviation of the ΔV state is not necessarily monotonic, even though the state itself is.

5.3 LinCov Program

The Linear Covariance Analysis program used for the present research is coded in MATLAB, originally by Geller [2,17,19,21]. This section describes the flow of the program operations and how the formulas presented in the previous chapters and section fit into the program.

The basic operation of the LinCov analysis is as follows:

1. Load idealized reference trajectory $\bar{\mathbf{x}}(t)$
2. Define variable names for matrix indexing and define simulation parameters. Simulation parameters include initial conditions, instrument operation regimes, noise

strengths and error time constants for instruments and actuators, and so forth.

3. Initialize augmented covariance matrix $\mathcal{C}_A$ and navigation filter covariance matrix $\hat{P}_f$ based on initial uncertainties, and allocate empty history matrices. The $(n+\hat{n}) \times (n+\hat{n})$ augmented covariance matrix $\mathcal{C}_A$ is kept separate from the $\hat{n} \times \hat{n}$ navigation filter covariance matrix $\hat{P}_f$ so that the performance of the filter may be compared with the coupled true/estimated covariance.

4. Loop on each time step of reference trajectory.

   (a) Propagate covariance matrices to the next time step.

   (b) Update covariance matrices using discrete measurements available.

   (c) Correct covariance matrices, specifically for gyro measurements but also allowed for instantaneous maneuvers.

   (d) Save selected state covariances in history matrices.

   (e) Shape covariance matrices based on scalar maneuver triggers, when appropriate.

5. Post-process history matrices to compute data of interest, e.g. relative navigation errors and dispersions from inertial state dispersions.

6. Plot and save data.

Several of these steps are detailed in the following subsections. Ideally, the reference trajectory should be generated using the same equations as those that are linearized for LinCov, where all noise and disturbance terms are zero, but that may not always be possible.

5.3.1 Covariance Propagation

The covariance matrices are propagated, that is advanced from the previous time step to the next time step, using the following procedure.

First, matrices C_x $(n_{\tilde{y}} \times n)$, $C_{\hat{x}}$ $(n_{\tilde{y}} \times \hat{n})$, and S_η $(n_{\tilde{y}} \times n_{\tilde{y}})$ are computed from the linearized true inertial instrument models. We have two triaxial inertial instruments (the gyro set and the accelerometer set) so $n_{\tilde{y}} = 6$. The elements of C_x and $C_{\hat{x}}$ are found by extracting the coefficients of each linearized state from Eqs. (3.47) and (3.48) in Section 3.2. The instrument noise strength matrix S_η is populated with the gyro, accelerometer, and thrust noise strengths. The accelerometer elements are only populated if the main descent engine is active at the current time step in the nominal trajectory. We may write

$$C_x = \begin{bmatrix} \dfrac{\partial^b \delta \tilde{\boldsymbol{\omega}}_{\text{gyro}}}{\partial \delta \mathbf{x}} \\ \dfrac{\partial^b \delta \tilde{\mathbf{a}}_{\text{accmtr}}}{\partial \delta \mathbf{x}} \end{bmatrix} \qquad C_{\hat{x}} = \begin{bmatrix} \dfrac{\partial^b \delta \tilde{\boldsymbol{\omega}}_{\text{gyro}}}{\partial \delta \hat{\mathbf{x}}} \\ \dfrac{\partial^b \delta \tilde{\mathbf{a}}_{\text{accmtr}}}{\partial \delta \hat{\mathbf{x}}} \end{bmatrix} \qquad S_\eta = \begin{bmatrix} S_{\eta_{\text{gyro}}} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & S_{\eta_{\text{accmtr}}} + S_{w_{\text{thr}}} \end{bmatrix}. \tag{5.20}$$

Second, the dynamic partial derivative matrices F_x $(n \times n)$, $F_{\hat{u}}$ $(n \times n_{\hat{u}})$ and process noise strength matrix S_w $(n \times n)$ are generated. These are found from the linearized true state dynamic equations found in Section 3.1. These matrices are generically written

$$F_x = \begin{bmatrix} \dfrac{\partial \delta \dot{\mathbf{x}}}{\partial \delta \mathbf{x}} \end{bmatrix} \qquad F_{\hat{u}} = \begin{bmatrix} \dfrac{\partial \delta \dot{\mathbf{x}}}{\partial \delta \hat{\mathbf{a}}_{\text{cmd}}} & \dfrac{\partial \delta \dot{\mathbf{x}}}{\partial \delta \hat{\mathbf{M}}_{\text{cmd}}} \end{bmatrix} \qquad S_w = \mathrm{E}\left[\mathbf{w}(t)\mathbf{w}^{\mathsf{T}}(t)\right] \tag{5.21}$$

with the process noise strength matrix S_w a diagonal matrix of each state's mutually independent noise strength.

Next, the continuous actuator command partial derivative matrix $\hat{G}_{\hat{x}}$ $(n_{\hat{u}} \times \hat{n})$ is calculated using Eqs. (4.9) and (4.13), and other related equations in Chapter 4. The general formula is expressed

$$\hat{G}_{\hat{x}} = \begin{bmatrix} \dfrac{\partial \delta \hat{\mathbf{a}}_{\text{cmd}}}{\partial \delta \hat{\mathbf{x}}} \\ \dfrac{\partial \delta \hat{\mathbf{M}}_{\text{cmd}}}{\partial \delta \hat{\mathbf{x}}} \end{bmatrix}. \tag{5.22}$$

With the filter version of the inertial instrument partial derivative matrix used for propagating the onboard covariance with the Kalman filter, we correct for estimated errors in the gyro and accelerometer data, but treat those measurements as deterministic. Therefore, we do not require $\hat{C}_{\hat{x}}$ in Eq. (2.40), the filter covariance propogation equation, or Ricatti equation. We still do require the $\hat{n} \times n_{\tilde{y}}$ matrix $\hat{F}_{\tilde{y}}$ and the $n_{\tilde{y}} \times n_{\tilde{y}}$ inertial insrument noise strength matrix $\hat{S}_\eta$:

$$\hat{F}_{\tilde{y}} = \left[\frac{\partial \delta \dot{\hat{\mathbf{x}}}}{\partial \delta \tilde{\mathbf{y}}} \right] \qquad \hat{S}_\eta = \left[\begin{array}{cc} \hat{S}_{\eta_{\text{gyro}}} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \hat{S}_{\eta_{\text{accmtr}}} + \hat{S}_{w_{\text{thr}}} \end{array} \right]. \tag{5.23}$$

And we need the filter model of the state dynamics expressed in the equations $\hat{F}_{\hat{x}}$ ($\hat{n} \times \hat{n}$), $\hat{F}_{\hat{u}}$ ($\hat{n} \times n_{\hat{u}}$) and process noise strength matrix $\hat{S}_w$ ($\hat{n} \times \hat{n}$)

$$\hat{F}_{\hat{x}} = \left[\frac{\partial \delta \dot{\hat{\mathbf{x}}}}{\partial \delta \hat{\mathbf{x}}} \right] \qquad \hat{F}_{\hat{u}} = \left[\begin{array}{cc} \frac{\partial \delta \dot{\hat{\mathbf{x}}}}{\partial \delta \hat{\mathbf{a}}_{\text{cmd}}} & \frac{\partial \delta \dot{\hat{\mathbf{x}}}}{\partial \delta \hat{\mathbf{M}}_{\text{cmd}}} \end{array} \right] \qquad \hat{S}_w = \mathrm{E}\left[\hat{\mathbf{w}}(t) \hat{\mathbf{w}}^{\mathsf{T}}(t) \right]. \tag{5.24}$$

Note that $\hat{F}_{\hat{u}}$ is a $\hat{n} \times n_u$ zero matrix, since according to our formulation in Section 3.3, we are not integrating actuator commands into the filter state, but rather the result of those commands sensed by the accelerometer and gyro. Also note that the gyro is treated specially since we are using the replacement model as described in Section 3.3 on page 43.

The filter covariance is propagated first by calculating $\hat{F} = \hat{F}_{\hat{x}} + \hat{F}_{\tilde{y}} \hat{C}_{\hat{x}} = \hat{F}_{\hat{x}}$, then calculating the state transition matrix

$$\hat{\Phi} = e^{\hat{F}\Delta t} = \mathbf{I}_{\hat{n}\times\hat{n}} + \hat{F}\Delta t + \frac{1}{2}\hat{F}\hat{F}\Delta t^2 + \ldots \frac{1}{N!}\hat{F}^N \Delta t^N \tag{5.25}$$

where N is the order of the Taylor's series expansion of the matrix exponential function. Using a state transition matrix is numerically simpler than integrating the Ricatti equation, as in Eq. (2.31). The exact form of Φ for the first-order Markov processes is

$$\Phi_i = e^{-\Delta t/\tau_i} \tag{5.26}$$

which is used in place of the appropriate elements in $\hat{\Phi}$ where, of course, the filter time constants $\hat{\tau}_i$ are used instead.

Then we may write the expression for the propagated filter covariance matrix as

$$\hat{P}_{i+1} = \hat{\Phi}\hat{P}\hat{\Phi}^{\mathsf{T}} + \hat{F}_{\tilde{y}} \hat{S}_\eta \hat{F}_{\tilde{y}}^{\mathsf{T}} + \hat{S}_w.$$

By a similar process, using $\mathcal{F}$ from Eq. (5.4), we propagate $\mathcal{C}_A$ with Eq. (5.5).

5.3.2 Covariance Update

The covariance update routine incorporates discrete instrument measurements at certain intervals, with the effect of reducing the overall uncertainty of state estimate. This function is a large part of the greater "navigation" aspect of GN&C.

While instrument models are a part of this research, and navigation performance is studied in Chapter 7, these models and equations are not the primary focus of this research and so will not be thoroughly detailed here. Much of the appropriate documentation can be found in Geller's previous work [19, 21], and in general literature for the broad topic of Kalman filtering with Battin [15] and Crassidis and Junkins [16] being the most applicable.

The linear covariance update equation is found in Eq. (2.59) and the smaller filter covariance update is in Eq. (2.44).

5.3.3 Covariance Correction

The final major operation in manipulating the covariance matrix is correction, or the direct substitution of instrument measurements (i.e. the gyro as we have discussed before) into the filter and augmented states. The corresponding covariances must also be corrected to reflect that substitution.

The equations required to correct the covariance have already been formulated. These are found in Eq. (2.47) for the filter covariance and in Eq. (2.57) for the augmented covariance. The function required for linearization to populate the matrix $\hat{D}_{\hat{x}}$ is found in Eq. (3.69). Linearizing it gives

$$\delta\hat{\mathbf{d}}_{\omega} = -E_{\xi}({}^{b}\bar{\boldsymbol{\omega}})\hat{\boldsymbol{\xi}}_{\text{gyro}} - {}^{b}\delta\hat{\boldsymbol{\omega}}^{-c} \tag{5.27}$$

from which the elements of the $\hat{n} \times \hat{n}$ matrix $\hat{D}_{\hat{x}}$ are easily extracted.

Since we do not have any discrete inertial measurements $\Delta\mathbf{c}$, and there are not instantaneous maneuvers performed, $\hat{D}_{\Delta\tilde{y}}$ and $\hat{D}_{\Delta\hat{u}}$ are null. The gyro also does not affect the truth state (only the filter state), so D_x and $D_{\Delta\hat{u}}$ are also empty.

5.3.4 Covariance Shaping

Covariance shaping occurs when, from a Monte Carlo standpoint, a scalar maneuver trigger calculated from the navigation state estimate is used to change the timing of certain events. In LinCov, this is simulated by reshaping the covariance matrix based on the effect of this trigger at the nominal event time. This is a major component of the present research and discussed in detail in the next chapter.

Chapter 6

Maneuver Triggers

As mentioned in Section 2.4, the vehicle's downrange distance from the landing site will be used as a trigger for Powered Descent Initiation (PDI). This research includes an analysis evaluating the effects of including this maneuver trigger in the lunar descent linear covariance analysis.

6.1 Trigger Shaping Matrix

This section will outline the calculations require to create Gossner's shaping matrix for Linear Covariance analysis which will be used at the trigger event to simulate the effects of a time-variable maneuver trigger. While we will only consider a single range trigger for PDI for this research, this method may be generalized and applied multiple times with different trigger definitions on a single analysis case. The mathematical definition of *range* will be elaborated later in Section 6.2.

6.1.1 Derivation

Let the true *range* between the true landing site and vehicle positions be denoted by $\rho = \rho(\mathbf{x})$. The trajectory is designed such that PDI occurs when $\rho = \rho^{\star}$, a predetermined range value. Since the onboard GN&C must ignite the engine, it relies on its filter calculation of $\hat{\rho} = \hat{\rho}(\hat{\mathbf{x}})$. On the reference trajectory, $\bar{\mathbf{x}}(t)$, the maneuver event (PDI) occurs at time t_E when $\bar{\rho} = \rho^{\star}$. On a dispersed trajectory, the true state, before and after the event occurs, may be expressed in terms of the reference state plus perturbations

$$\begin{aligned}
\mathbf{x}^{-E} &= \bar{\mathbf{x}} + \delta\mathbf{x}^{-E} \\
\mathbf{x}^{+E} &= \bar{\mathbf{x}} + \delta\mathbf{x}^{+E} = \bar{\mathbf{x}} + \delta\mathbf{x}^{-E} + \left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\bar{\mathbf{x}}} \delta t.
\end{aligned} \tag{6.1}$$

The true and filter dispersions after the event may be written in terms of those before the event by subtracting the reference trajectory,

$$\begin{aligned} \delta\mathbf{x}^{+E} &= \delta\mathbf{x}^{-E} + \dot{\mathbf{x}}_{\text{true}}\delta t, && (6.2) \\ \delta\hat{\mathbf{x}}^{+E} &= \delta\hat{\mathbf{x}}^{-E} + \dot{\hat{\mathbf{x}}}_{\text{filter}}\delta t. && (6.3) \end{aligned}$$

The flight computer schedules the event to occur such that $\hat{\rho}(\hat{\mathbf{x}}^{+E}(t)) = \rho^{\star}$. Expanding and using a truncated Taylor series, and solving for the time dispersion,

$$\begin{aligned} \hat{\rho}\left(\bar{\mathbf{x}}(t) + \delta\hat{\mathbf{x}}^{-E} + \dot{\hat{\mathbf{x}}}_{\text{filter}}\delta t\right) &= \rho^{\star} \\ \rho^{\star} + \left.\frac{\partial\hat{\rho}}{\partial\hat{\mathbf{x}}}\right|_{\bar{\mathbf{x}}} \left[\delta\hat{\mathbf{x}}^{-E} + \dot{\hat{\mathbf{x}}}_{\text{filter}}\delta t\right] &= \rho^{\star} \\ \mathbf{k}^{\mathsf{T}}\dot{\hat{\mathbf{x}}}_{\text{filter}}\delta t &= -\mathbf{k}^{\mathsf{T}}\delta\hat{\mathbf{x}}^{-E} \\ \delta t &= -\frac{\mathbf{k}^{\mathsf{T}}\delta\hat{\mathbf{x}}^{-E}}{\mathbf{k}^{\mathsf{T}}\dot{\hat{\mathbf{x}}}_{\text{filter}}} && (6.4) \end{aligned}$$

where, using Gossner's notation in Eq. (2.66), $\mathbf{k}^{\mathsf{T}} \equiv \left.\frac{\partial\hat{\rho}}{\partial\hat{\mathbf{x}}}\right|_{\bar{\mathbf{x}}}$ is the vector derivative of the estimated range with respect to the estimated state, and called the sensitivity or constraint vector [20]. Note that the division as shown is possible since this is a scalar equation and the $1 \times \hat{n}$ matrix $\mathbf{k}^{\mathsf{T}}$ is being multiplied by $\hat{n} \times 1$ vectors to produce a scalar dot product. Also note that the magnitude of $\mathbf{k}^{\mathsf{T}}$ is not important, only its "direction" in the state-space. In Linear Covariance analysis, we do not work with dispersions directly, but with covariances. It is useful to determine the variance of the time slip, which we may obtain from the augmented covariance matrix $\mathcal{C}_{\mathcal{A}} = \mathrm{E}\left[\mathbf{X}^{-E}\mathbf{X}^{-E,\mathsf{T}}\right]$.

$$\begin{aligned} \mathrm{E}\left[\delta t\delta t^{\mathsf{T}}\right] &= \mathrm{E}\left[\tfrac{1}{\left(\mathbf{k}^{\mathsf{T}}\dot{\hat{\mathbf{x}}}_{\text{filter}}\right)^2}\mathbf{k}^{\mathsf{T}}\delta\hat{\mathbf{x}}^{-E}\delta\hat{\mathbf{x}}^{-E,\mathsf{T}}\mathbf{k}\right] \\ &= \tfrac{1}{\left(\mathbf{k}^{\mathsf{T}}\dot{\hat{\mathbf{x}}}_{\text{filter}}\right)^2}\mathrm{E}\left[\mathbf{k}^{\mathsf{T}}\begin{pmatrix} \mathbf{0}_{\hat{n}\times n} & \mathbf{I}_{\hat{n}\times\hat{n}} \end{pmatrix}\mathbf{X}^{-E}\mathbf{X}^{-E,\mathsf{T}}\begin{pmatrix} \mathbf{0}_{\hat{n}\times n} & \mathbf{I}_{\hat{n}\times\hat{n}} \end{pmatrix}^{\mathsf{T}}\mathbf{k}\right] \\ \sigma_{\delta t}^2 &= \tfrac{1}{\left(\mathbf{k}^{\mathsf{T}}\dot{\hat{\mathbf{x}}}_{\text{filter}}\right)^2}\begin{bmatrix} \mathbf{0}_{1\times n} & \mathbf{k}^{\mathsf{T}} \end{bmatrix}\mathcal{C}_A\begin{bmatrix} \mathbf{0}_{1\times n} & \mathbf{k}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. && (6.5) \end{aligned}$$

Substituting Eq. (6.4) into Eq. (6.3), we find an equation similar to Eq. (2.70).

$$\delta\hat{\mathbf{x}}^{+E} = \delta\hat{\mathbf{x}}^{-E} - \dot{\hat{\mathbf{x}}}_{\text{filter}} \frac{\mathbf{k}^{\mathsf{T}}\delta\hat{\mathbf{x}}^{-E}}{\mathbf{k}^{\mathsf{T}}\dot{\hat{\mathbf{x}}}_{\text{filter}}}$$

$$\delta\hat{\mathbf{x}}^{+E} = \left[\mathbf{I}_{\hat{N}\times\hat{N}} - \frac{\dot{\hat{\mathbf{x}}}_{\text{filter}}\mathbf{k}^{\mathsf{T}}}{\mathbf{k}^{\mathsf{T}}\dot{\hat{\mathbf{x}}}_{\text{filter}}}\right]\delta\hat{\mathbf{x}}^{-E} \tag{6.6}$$

$$\delta\hat{\mathbf{x}}^{+E} = \left[\mathbf{I}_{\hat{N}\times\hat{N}} - S\right]\delta\hat{\mathbf{x}}^{-E} = \Phi_I\delta\hat{\mathbf{x}}^{-E}, \tag{6.7}$$

where Φ_I is Gossner's idempotent shaping matrix from Eq. (2.70). However, when Eq. (6.4) is substituted into Eq. (6.2), a slightly different result is obtained.

$$\delta\mathbf{x}^{+E} = \delta\mathbf{x}^{-E} - \dot{\hat{\mathbf{x}}}_{\text{true}} \frac{\mathbf{k}^{\mathsf{T}}\delta\hat{\mathbf{x}}^{-E}}{\mathbf{k}^{\mathsf{T}}\dot{\hat{\mathbf{x}}}_{\text{filter}}}$$

$$\delta\mathbf{x}^{+E} = \delta\mathbf{x}^{-E} - S_{\text{true}}\delta\hat{\mathbf{x}}^{-E}, \tag{6.8}$$

where S_{true} is the $n \times \hat{n}$ matrix $\dot{\hat{\mathbf{x}}}_{\text{true}}\mathbf{k}^{\mathsf{T}}/(\mathbf{k}^{\mathsf{T}}\dot{\hat{\mathbf{x}}}_{\text{filter}})$. Combining Eqs. (6.6) and (6.8), we obtain

$$\mathbf{X}^{+E} = \begin{bmatrix} \delta\mathbf{x}^{+E} \\ \delta\hat{\mathbf{x}}^{+E} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n\times n} & -S_{\text{true}} \\ \mathbf{0}_{\hat{n}\times n} & \mathbf{I}_{\hat{n}\times\hat{n}} - S \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}^{-E} \\ \delta\hat{\mathbf{x}}^{-E} \end{bmatrix} = \mathcal{M}\mathbf{X}^{-E}, \tag{6.9}$$

where $\mathcal{M}$ is the shaping matrix for the augmented state. If the true and filter states and their dynamics are identical, it is easily seen that $S_{\text{true}} = S$. The augmented covariance matrix after the event is found using the expected value operator,

$$\mathrm{E}\left[\mathbf{X}^{+E}\mathbf{X}^{+E,\mathsf{T}}\right] = E\left[\mathcal{M}\mathbf{X}^{-E}\mathbf{X}^{-E,\mathsf{T}}\mathcal{M}^{\mathsf{T}}\right] \tag{6.10}$$

$$\mathcal{C}_A^{+E} = \mathcal{M}\mathcal{C}_A^{-E}\mathcal{M}^{\mathsf{T}}. \tag{6.11}$$

With this derivation and known state dynamics, all that is required to calculate $\mathcal{M}$ is the vector $\hat{\mathbf{k}}^{\mathsf{T}} \equiv \partial\hat{\rho}/\partial\hat{\mathbf{x}}$. Fortunately, all three definitions of range discussed in the following section only involve two states, $\hat{\mathbf{r}}$ and $\hat{\mathbf{r}}_{\text{site}}$. The derivations of the partial derivatives of each definition of $\hat{\rho}$ is done in Section 6.2, and is summarized in Table 6.1 on page 82.

6.1.2 Nominal State Dynamics

The matrices S and S_{true} require the calculation of $\dot{\mathbf{x}}_{\text{filter}}$ and $\dot{\mathbf{x}}_{\text{true}}$. This appears to be straightforward since the nonlinear equations of motion for the entire state were presented in Chapter 3, and we are using the same state variables and models for both the truth and filter states, so we expect $\dot{\mathbf{x}} = \dot{\mathbf{x}}_{\text{filter}} = \dot{\mathbf{x}}_{\text{true}}$. What is not clear from the derivation above is whether the values required are the pre- or post-event dynamics, that is $\dot{\mathbf{x}}^{-E}$ or $\dot{\mathbf{x}}^{+E}$. This section will address this problem, and define the $\dot{\mathbf{x}}$ used for the present research.

For our research, $\dot{\mathbf{x}}^{-E}$ and $\dot{\mathbf{x}}^{+E}$ differ because the event in question is Powered Descent Initiation, when the Descent Module Main Propulsion engine is ignited, therefore the nominal thrust-accleration $\bar{\mathbf{a}}_{\text{thr}}$ is zero on the $-E$ side of the event but significantly nonzero on the $+E$ side. By reviewing the state dynamics in Chapter 3 and applying nominal conditions, we have (from the equations listed):

$$\text{3.2, 3.49} \qquad {}^i\dot{\bar{\mathbf{r}}}_{\text{site}} = {}^i\boldsymbol{\omega}_{\text{moon}} \times {}^i\bar{\mathbf{r}}_{\text{site}} \tag{6.12}$$

$$\text{3.15, 3.51} \qquad {}^i\dot{\bar{\mathbf{r}}} = {}^i\bar{\mathbf{v}} \tag{6.13}$$

$$\text{3.16, 3.52, 3.53, 3.44, 4.7} \qquad {}^i\dot{\bar{\mathbf{v}}} = \begin{cases} -\dfrac{\mu}{\bar{r}^3}\,{}^i\bar{\mathbf{r}} & (-E) \\[2ex] -\dfrac{\mu}{\bar{r}^3}\,{}^i\bar{\mathbf{r}} + {}^i_b\bar{\mathcal{T}}\,{}^b\bar{\mathbf{a}}_{\text{thr}} & (+E) \end{cases} \tag{6.14}$$

$$\text{3.23, 3.22} \qquad {}^b\dot{\bar{\boldsymbol{\theta}}} = {}^b\bar{\boldsymbol{\omega}} \tag{6.15}$$

$$\text{3.18, 3.68, 3.46} \qquad {}^b\dot{\bar{\boldsymbol{\omega}}} = {}^bI^{-1}\left[{}^b\bar{\mathbf{M}}_{\text{cmd}} - {}^b\bar{\boldsymbol{\omega}} \times {}^bI\,{}^b\bar{\boldsymbol{\omega}}\right] \tag{6.16}$$

$$\text{3.35, 3.37, 3.74} \qquad \dot{\bar{\mathbf{p}}} = \mathbf{0}_{n_p \times 1} \tag{6.17}$$

$$\text{5.9, 5.15, 5.14, 4.7} \qquad \Delta\dot{\bar{V}} = \begin{cases} 0 & (-E) \\[2ex] \bar{a}_{\text{thr}} = \|\bar{\mathbf{a}}_{\text{thr}}\| & (+E) \end{cases} \tag{6.18}$$

In general, we may desire to write the nominal nonlinear state dynamics used for the shaping matrix as

$$\dot{\mathbf{x}} = (1-\beta)\dot{\mathbf{x}}^{-E} + \beta\dot{\mathbf{x}}^{+E}, \qquad 0 \le \beta \le 1 \tag{6.19}$$

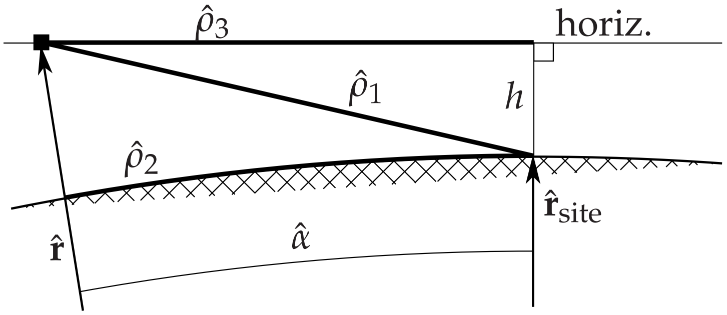

Fig. 6.1: Alternative definitions for range: slant range $\hat{\rho}_1$, great circle distance $\hat{\rho}_2$, and horizontal range $\hat{\rho}_3$.

where β is a weighting factor or mixing ratio between the dynamics on the two sides of the event. Obvious values for β worth consideration are 0, 1, and 1/2.

We may use a thought exercise to determine whether $\dot{\mathbf{x}}^{-E}$ or $\dot{\mathbf{x}}^{+E}$ or a combination of the two is most appropriate. Since we know that $\hat{\rho}$ is a function only of ${}^i\hat{\mathbf{r}}$ and ${}^i\hat{\mathbf{r}}_{\text{site}}$, the values of δt and its covariance are not dependent on whether we include $\bar{\mathbf{a}}_{\text{thr}}$ in the nominal dynamics. Consider a dispersed trajectory where the engine is ignited $\sigma_{\delta t}$ seconds *before* the nominal PDI event time t_E. During the ensuing guidance phase, and even all the way to the surface, it would be reasonable to assume that each of the guidance target times, $t_{t,i}$, where $t_{\text{go}} = t_t - t$, would need to also move forward by $\sigma_{\delta t}$ seconds. Otherwise, the vehicle would burn longer than necessarily required to achieve the target, resulting in large ΔV dispersions. We may conclude, therefore, that it is not necessary to include the $\bar{\mathbf{a}}_{\text{thr}}$ term in ${}^i\dot{\hat{\mathbf{v}}}$ or $\bar{a}_{\text{thr}}$ in $\Delta\dot{\hat{V}}$, because we recognize the time slip dispersion δt exists from PDI to touchdown. Then, we may choose the true dynamics used for trigger covariance shaping to be defined on the pre-event side of the event time. This condition is achieved when $\beta = 0$ in Eq. (6.19), such that,

$$\dot{\mathbf{x}}_{\text{true}}(t_E) = \dot{\mathbf{x}}^{-E}_{\text{true}}(t_E), \qquad \dot{\mathbf{x}}_{\text{filter}}(t_E) = \dot{\mathbf{x}}^{-E}_{\text{filter}}(t_E). \tag{6.20}$$

6.2 Range Definitions and Derivatives

At least three possibilities "ranges" exist that may be used for this value, which are shown in Fig. 6.1 and described mathematically below. While three possibilities are presented, namely *slant range*, *great circle range*, and *horizontal range*, the present research only uses great circle range as a maneuver trigger.

6.2.1 Slant Range

The first definition considered is the estimated slant range, that is the estimated distance between the vehicle and the landing site as measured by a straight line connecting the two. Mathematically, this may be written

$$\hat{\rho}_1 = \left\| {}^i\Delta\hat{\mathbf{r}} \right\| = \left\| {}^i\hat{\mathbf{r}}_{\text{site}} - {}^i\hat{\mathbf{r}} \right\| = \sqrt{\hat{r}_{\text{site}}^2 + \hat{r}^2 - 2\hat{\mathbf{r}}_{\text{site}} \cdot \hat{\mathbf{r}}} \tag{6.21}$$

where ${}^i\Delta\hat{\mathbf{r}}$ is the estimated relative position vector between ${}^i\hat{\mathbf{r}}_{\text{site}}$, the estimated landing site position (magnitude $\hat{r}_{\text{site}}$), and ${}^i\hat{\mathbf{r}}$, the estimated vehicle position (magnitude $\hat{r}$), all expressed in the inertial frame for convenience.

Recalling that the partial derivative of the magnitude of vector with respect to that vector is its unit vector, that is

$$\frac{\partial z}{\partial \mathbf{z}} = \frac{\partial}{\partial \mathbf{z}} \left(\mathbf{z}^{\mathsf{T}}\mathbf{z}\right)^{1/2} = \frac{1}{2}\left(\mathbf{z}^{\mathsf{T}}\mathbf{z}\right)^{-1/2}\left(2\mathbf{z}^{\mathsf{T}}\right) = \frac{\mathbf{z}^{\mathsf{T}}}{z} = \mathbf{i}_z^{\mathsf{T}}.$$

So, we may concisely express the partial derivatives of slant range by writing its dispersions,

$$\delta\hat{\rho}_1 = {}^i\bar{\mathbf{i}}_{\Delta r}^{\mathsf{T}} \left[{}^i\delta\hat{\mathbf{r}}_{\text{site}} - {}^i\delta\hat{\mathbf{r}} \right], \tag{6.22}$$

where ${}^i\bar{\mathbf{i}}_{\Delta r}^{\mathsf{T}} = {}^i\Delta\bar{\mathbf{r}}/\bar{\rho}_1$ is the unit vector in the direction of the spacecraft from the landing site.

6.2.2 Great Circle Distance

Great circle distance is the length of the of the arc drawn on the surface connecting the sub-satellite point to the landing site, or in other words, the length of the ground track yet to be accomplished if no cross-track motion is considered.

$$\hat{\rho}_2 = R_{\text{moon}} \hat{\alpha} \tag{6.23}$$

where R_{moon} is the mean lunar radius, and $\hat{\alpha}$ is the central angle in radians between the estimated vehicle and landing site position vectors, whose unit vectors are $\mathbf{i}_{\hat{r}}$ and $\mathbf{i}_{\hat{r}_{\text{site}}}$, respectively, such that

$$\cos \hat{\alpha} = \mathbf{i}_{\hat{r}} \cdot \mathbf{i}_{\hat{r}_{\text{site}}}. \tag{6.24}$$

We may write the partial derivatives of the great circle distance by linearizing these equations, so we have $\delta \hat{\rho}_2 = R_{\text{moon}} \delta \hat{\alpha}$, and recalling that $\delta \mathbf{i}_z = \frac{1}{z} \left[\mathbf{I} - \bar{\mathbf{i}}_z \bar{\mathbf{i}}_z^{\mathsf{T}} \right] \delta \mathbf{z}$, where Eq. (6.24) is linearized

$$(-\sin \hat{\alpha})_{\bar{\mathbf{x}}}\, \delta \hat{\alpha} = \left({}^{i}\mathbf{i}_{\hat{r}}^{\mathsf{T}} \frac{1}{\hat{r}_{\text{site}}} \left[\mathbf{I}_{3\times 3} - {}^{i}\mathbf{i}_{\hat{r}_{\text{site}}} {}^{i}\mathbf{i}_{\hat{r}_{\text{site}}}^{\mathsf{T}} \right] \right)_{\bar{\mathbf{x}}} {}^{i}\delta \hat{\mathbf{r}}_{\text{site}} + \left({}^{i}\mathbf{i}_{\hat{r}_{\text{site}}}^{\mathsf{T}} \frac{1}{\hat{r}} \left[\mathbf{I}_{3\times 3} - {}^{i}\mathbf{i}_{\hat{r}} {}^{i}\mathbf{i}_{\hat{r}}^{\mathsf{T}} \right] \right)_{\bar{\mathbf{x}}} {}^{i}\delta \hat{\mathbf{r}}. \tag{6.25}$$

Combining these equations, we may write

$$\begin{aligned}
\delta \hat{\rho}_2 &= -\frac{R_{\text{moon}}}{\sin \hat{\alpha}} \bigg|_{\bar{\mathbf{x}}} \left\{ \frac{1}{\hat{r}_{\text{site}}} \bigg|_{\bar{\mathbf{x}}} \left[{}^{i}\mathbf{i}_{\hat{r}}^{\mathsf{T}} - {}^{i}\mathbf{i}_{\hat{r}}^{\mathsf{T}} {}^{i}\mathbf{i}_{\hat{r}_{\text{site}}} {}^{i}\mathbf{i}_{\hat{r}_{\text{site}}}^{\mathsf{T}} \right]_{\bar{\mathbf{x}}} {}^{i}\delta \hat{\mathbf{r}}_{\text{site}} + \frac{1}{\hat{r}} \bigg|_{\bar{\mathbf{x}}} \left[{}^{i}\mathbf{i}_{\hat{r}_{\text{site}}}^{\mathsf{T}} - {}^{i}\mathbf{i}_{\hat{r}_{\text{site}}}^{\mathsf{T}} {}^{i}\mathbf{i}_{\hat{r}} {}^{i}\mathbf{i}_{\hat{r}}^{\mathsf{T}} \right]_{\bar{\mathbf{x}}} {}^{i}\delta \hat{\mathbf{r}} \right\} \\
\delta \hat{\rho}_2 &= \frac{R_{\text{moon}}}{\sin \bar{\alpha}} \left\{ \frac{1}{\bar{r}_{\text{site}}} \left[{}^{i}\bar{\mathbf{i}}_{r_{\text{site}}} \cos \bar{\alpha} - {}^{i}\bar{\mathbf{i}}_{r} \right]^{\mathsf{T}} {}^{i}\delta \hat{\mathbf{r}}_{\text{site}} + \frac{1}{\bar{r}} \left[{}^{i}\bar{\mathbf{i}}_{r} \cos \bar{\alpha} - {}^{i}\bar{\mathbf{i}}_{r_{\text{site}}} \right]^{\mathsf{T}} {}^{i}\delta \hat{\mathbf{r}} \right\}.
\end{aligned} \tag{6.26}$$

6.2.3 Landing Site Horizontal Range

The landing site local horizontal range is the distance measured in the plane tangent to the surface at the landing site, between the landing site and the spacecraft's location projected perpendicularly into that plane, as shown in Fig. 6.1 in two dimensions. By drawing another plane parallel to the tangent plane and passing through the vehicle, we can see a right triangle with hypotenuse $\hat{r}$, long leg $\hat{r}_{\text{site}} + h$, and short side $\hat{\rho}_3$. Using the definition of the central angle $\hat{\alpha}$ in the previous section, we may write

$$\hat{\rho}_3 \quad = \quad \hat{r} \sin \hat{\alpha}. \tag{6.27}$$

Using the product rule, the derivative of Eq. (6.27) is $\delta\hat{\rho}_3 = \delta\hat{r}\left(\sin\hat{\alpha}\right)_{\bar{\mathbf{x}}} + \left(\hat{r}\cos\hat{\alpha}\right)_{\bar{\mathbf{x}}}\delta\hat{\alpha}$. Substituting the derivative of $\hat{\alpha}$ taken from Eq. (6.26), the partial derivatives of horizontal range can be found from its linearization:

$$\delta\hat{\rho}_3 = \left(\sin\bar{\alpha}\right){}^i\bar{\mathbf{i}}_r^{\mathsf{T}}\,{}^i\delta\hat{\mathbf{r}} + \frac{\bar{r}\cos\bar{\alpha}}{\sin\bar{\alpha}}\left\{\frac{1}{\bar{r}_{\text{site}}}\left[{}^i\bar{\mathbf{i}}_{r_{\text{site}}}\cos\bar{\alpha} - {}^i\bar{\mathbf{i}}_r\right]^{\mathsf{T}}{}^i\delta\hat{\mathbf{r}}_{\text{site}} + \frac{1}{\bar{r}}\left[{}^i\bar{\mathbf{i}}_r\cos\bar{\alpha} - {}^i\bar{\mathbf{i}}_{r_{\text{site}}}\right]^{\mathsf{T}}{}^i\delta\hat{\mathbf{r}}\right\} \tag{6.28}$$

Equation (6.31) can be used to directly extract the partial derivatives, but some rearranging can be done to simplify this expression. To do so, we will take the terms separately and eliminate the trigonometric terms by multiplying Eq. (6.27) and substituting in Eq. (6.24). First, for the vehicle inertial position state,

$$\begin{aligned}
\frac{\partial\delta\hat{\rho}_3}{\partial^i\delta\hat{\mathbf{r}}} &= \left(\sin\bar{\alpha}\right){}^i\bar{\mathbf{i}}_r^{\mathsf{T}} + \frac{\bar{r}\cos\bar{\alpha}}{\bar{r}\sin\bar{\alpha}}\left[{}^i\bar{\mathbf{i}}_r\cos\bar{\alpha} - {}^i\bar{\mathbf{i}}_{r_{\text{site}}}\right]^{\mathsf{T}} \\
\bar{\rho}_3\frac{\partial\delta\hat{\rho}_3}{\partial^i\delta\hat{\mathbf{r}}} &= \bar{r}\sin^2\bar{\alpha}\,{}^i\bar{\mathbf{i}}_r^{\mathsf{T}} + \bar{r}\cos^2\bar{\alpha}\,{}^i\bar{\mathbf{i}}_r^{\mathsf{T}} - \bar{r}\cos\bar{\alpha}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\mathsf{T}} \\
\bar{\rho}_3\frac{\partial\delta\hat{\rho}_3}{\partial^i\delta\hat{\mathbf{r}}} &= {}^i\bar{\mathbf{r}}^{\mathsf{T}} - \bar{r}\left({}^i\bar{\mathbf{i}}_r^{\mathsf{T}}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}\right){}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\mathsf{T}} \\
\frac{\partial\delta\hat{\rho}_3}{\partial^i\delta\hat{\mathbf{r}}} &= \frac{1}{\bar{\rho}_3}{}^i\bar{\mathbf{r}}^{\mathsf{T}}\left[\mathbf{I}_{3\times3} - {}^i\bar{\mathbf{i}}_{r_{\text{site}}}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\mathsf{T}}\right].
\end{aligned} \tag{6.29}$$

With respect to the inertial landing site state, we have

$$\begin{aligned}
\frac{\partial\delta\hat{\rho}_3}{\partial^i\delta\hat{\mathbf{r}}_{\text{site}}} &= \frac{\bar{r}\cos\bar{\alpha}}{\sin\bar{\alpha}}\frac{1}{\bar{r}_{\text{site}}}\left[{}^i\bar{\mathbf{i}}_{r_{\text{site}}}\cos\bar{\alpha} - {}^i\bar{\mathbf{i}}_r\right]^{\mathsf{T}} \\
\bar{\rho}_3\frac{\partial\delta\hat{\rho}_3}{\partial^i\delta\hat{\mathbf{r}}_{\text{site}}} &= \frac{\bar{r}^2}{\bar{r}_{\text{site}}}\left({}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\mathsf{T}}\,{}^i\bar{\mathbf{i}}_r\right)\left[\left({}^i\bar{\mathbf{i}}_r^{\mathsf{T}}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}\right){}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\mathsf{T}} - {}^i\bar{\mathbf{i}}_r^{\mathsf{T}}\right] \\
\frac{\partial\delta\hat{\rho}_3}{\partial^i\delta\hat{\mathbf{r}}_{\text{site}}} &= \frac{1}{\bar{\rho}_3\bar{r}_{\text{site}}}{}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\mathsf{T}}\,{}^i\bar{\mathbf{r}}\,{}^i\bar{\mathbf{r}}^{\mathsf{T}}\left[{}^i\bar{\mathbf{i}}_{r_{\text{site}}}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\mathsf{T}} - \mathbf{I}_{3\times3}\right].
\end{aligned} \tag{6.30}$$

If we wish to combine Eqs. (6.29) and (6.30) concisely, we may write

$$\delta\hat{\rho}_3 = \frac{1}{\bar{\rho}_3}{}^i\bar{\mathbf{r}}^{\mathsf{T}}\left[\mathbf{I}_{3\times3} - {}^i\bar{\mathbf{i}}_{r_{\text{site}}}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\mathsf{T}}\right]\left\{{}^i\delta\hat{\mathbf{r}} - \frac{\left({}^i\bar{\mathbf{i}}_{r_{\text{site}}}\cdot{}^i\bar{\mathbf{r}}\right)}{\bar{r}_{\text{site}}}{}^i\delta\hat{\mathbf{r}}_{\text{site}}\right\}. \tag{6.31}$$

All the partial derivatives of range are summarized in Table 6.1 on page 82. These partial derivatives are the required components of the row vector $\mathbf{k}^{\mathsf{T}}$, which is used to

Table 6.1: Partial Derivatives of Three Definitions of Range w.r.t. Filter States.

	$\partial\delta\hat{\rho}/\partial\delta\hat{\mathbf{r}}\vert_{\bar{\mathbf{x}}}$	$\partial\delta\hat{\rho}/\partial\delta\hat{\mathbf{r}}_{\text{site}}\vert_{\bar{\mathbf{x}}}$
Slant Range, $\hat{\rho}_1$	$-{}^i\bar{\mathbf{i}}_{\Delta r}^{\top} = \frac{\left({}^i\bar{\mathbf{r}} - {}^i\bar{\mathbf{r}}_{\text{site}}\right)^{\top}}{\bar{\rho}_1}$	${}^i\bar{\mathbf{i}}_{\Delta r}^{\top} = \frac{\left({}^i\bar{\mathbf{r}}_{\text{site}} - {}^i\bar{\mathbf{r}}\right)^{\top}}{\bar{\rho}_1}$
Great Circle, $\hat{\rho}_2$	$\frac{R_{\text{moon}}}{\bar{r}\sin\bar{\alpha}}\,{}^i\bar{\mathbf{i}}_r^{\top}\left[\mathbf{I}_{3\times3} - {}^i\bar{\mathbf{i}}_r\,{}^i\bar{\mathbf{i}}_r^{\top}\right]$	$\frac{R_{\text{moon}}}{\bar{r}_{\text{site}}\sin\bar{\alpha}}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\top}\left[\mathbf{I}_{3\times3} - {}^i\bar{\mathbf{i}}_{r_{\text{site}}}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\top}\right]$
Horizontal Range, $\hat{\rho}_3$	$\frac{1}{\bar{\rho}_3}\,{}^i\bar{\mathbf{r}}^{\top}\left[\mathbf{I}_{3\times3} - {}^i\bar{\mathbf{i}}_{r_{\text{site}}}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\top}\right]$	$\frac{{}^i\bar{\mathbf{r}}^{\top}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}}{\bar{\rho}_3\bar{r}_{\text{site}}}\,{}^i\bar{\mathbf{r}}^{\top}\left[{}^i\bar{\mathbf{i}}_{r_{\text{site}}}\,{}^i\bar{\mathbf{i}}_{r_{\text{site}}}^{\top} - \mathbf{I}_{3\times3}\right]$

calculate the shaping matrix Φ_I. For this research only the second definition of range, the great circle distance $\hat{\rho}_2$, is used, although all three are present in the LinCov program.

Chapter 7
Navigation Studies

7.1 Inertial Navigation Performance Study

Two important features of ALHAT technology are Terrain Relative Navigation and Hazard Detection and Avoidance [28]. Both to justify their inclusion in the LSAM and to assess the potential vehicle performance in their absence and failure, an inertial-only navigation study is performed where these two features are removed, that is, turned off. In this study, the LSAM navigation covariance matrix is propagated with updates only from inertial sensors. This allows parameter sweeps to gauge the relative significance of Deep Space Network (DSN) update quality, gravity model errors, altimeter operation altitude, terrain uncertainty, and altimeter quality on the performance of the navigation system.

A summary of assumptions used for this analysis is given in Tables 7.1 and 7.2. Each of the five parameters in Table 7.2 are varied to study the sensitivity of the *relative* navigation error between the lander and landing site at three critical locations on the trajectory—immediately prior to PDI, at the beginning of the final descent phase, and at touchdown.

Results from the inertial-only study indicate that high-quality DSN updates and improved lunar gravity modeling are greatly beneficial to the navigation knowledge at these key locations. Table 7.3 compares the results of the best performing variations of each of the five parameters in Table 7.2. The DSN improvement and gravity model improvement produce nearly identical reductions in position uncertainty magnitude, but the improvements are complimentary. That is, the better gravity model sharply reduces the cross-track position uncertainty, while the better DSN update improves the position knowledge in the downrange direction. From a trade standpoint, if a choice must be made between

Table 7.1: Fixed Simulation Parameters for Inertial Navigation Study

Error source		1-σ uncertainty
Random accelerations		0.4 mm/s/$\sqrt{\text{s}}$
Vehicle ICs	Attitude	50 arcsec
	Angular Velocity	5 arcsec/s
Landing site inertial position IC		33.3 m north/south, east/west; 3.33 m up/down
Star tracker	Noise, each axis	50 arcsec
	Alignment, each axis	20 arcsec
Gyroscope	Scale factor	1.6 PPM
	Alignment, each axis	20 arcsec
	Bias	0.02 arcsec/s
	Angular random walk	0.000 05 deg/$\sqrt{\text{s}}$
Accelerometer	Scale factor	66 PPM, τ=10 000 s
	Alignment	20 arcsec, τ=10 000 s
	Bias	30 μg, τ=10 000 s
	Velocity random walk	0.000 05 m/s/$\sqrt{\text{s}}$
Velocimeter	Scale factor	0.1 %, τ=100 s
	Alignment	50 arcsec
	Bias	0.1 m/s, τ=100 s
	Noise	0.5 m/s

Table 7.2: Variable Simulation Parameters for Inertial Navigation Study

#	Error source (1-σ) or instrument parameter	Nominal	Varied
1	Gravity model error	20 mGal (400 km correlation distance)	2 mGal (400 km correlation distance)
2	Initial lander position error	1500 m down-range, 200 m cross-track, 50 m altitude	100 m down-range, 100 m cross-track, 10 m altitude
	Initial lander velocity error	0.04 m/s down-range, 0.2 m/s cross-track, 1.5 m/s altitude	0.01 m/s down-range, 0.1 m/s cross-track, 0.08 m/s altitude
3	Altimeter operation altitude	20 km, down to 10 m	30, 25, 15, 20 km, down to 10 m
4	Terrain uncertainty	33.3 m (10 km correlation distance)	6.67 m, 66.7 m (10 km correlation distance)
5	Altimeter scale factor	0.1 %, τ=100 s	0 % (perfect)
	Altimeter bias	0.5 m, τ=100 s	0 m (perfect)
	Altimeter noise, by altitude h	$\begin{cases} 1\,\text{m}, & 10\,\text{m} \leq h \leq 2\,\text{km} \\ 5\,\text{m}, & 2\,\text{km} \leq h \leq 20\,\text{km} \\ 10\,\text{m}, & h > 20\,\text{km} \end{cases}$	0 m (perfect)

Table 7.3: Inertial Navigation Study Best-Case Scenario Comparison

3-σ Uncertainty			Nominal	**#1**		**#2**		**#3 (50 km)**		**#4 (6.67 m)**		**#5**	
			3-σ error	3-σ error	%ch	3-σ error	%ch	3-σ error	%ch	3-σ error	%ch	3-σ error	%ch
Pre-PDI (17.5 km)	Position (m)	DR	4880	4690	-4%	1030	**-79%**	4500	*-8%*	4740	-3%	4740	-3%
		CT	894	686	**-23%**	671	**-25%**	894	0%	894	0%	894	0%
		ALT	71.5	68.2	-5%	67.5	*-6%*	60.1	**-16%**	53	**-26%**	47.1	**-34%**
		Mag	4960	4740	-5%	1230	**-75%**	4590	*-8%*	4820	-3%	4820	-3%
	Velocity (m/s)	DR	0.412	0.078	**-81%**	0.362	**-12%**	0.388	*-6%*	0.411	0%	0.411	0%
		CT	0.72	0.543	**-25%**	0.549	**-24%**	0.72	0%	0.72	0%	0.72	0%
		ALT	4.57	4.36	-5%	0.731	**-84%**	4.35	-5%	4.51	-1%	4.51	-1%
		Mag	4.65	4.39	*-6%*	0.983	**-79%**	4.43	-5%	4.59	-1%	4.59	-1%
Begin Final Descent (3 km alt.)	Position (m)	DR	1000	877	**-12%**	636	**-36%**	992	-1%	934	*-7%*	798	**-20%**
		CT	661	352	**-47%**	598	*-9%*	661	0%	661	0%	661	0%
		ALT	16.4	16.4	0%	16.3	-1%	16.4	0%	15.2	*-8%*	10.8	**-34%**
		Mag	1200	945	**-21%**	873	**-27%**	1190	0%	1140	-5%	1040	**-14%**
	Velocity (m/s)	DR	0.6	0.461	**-23%**	0.581	-3%	0.569	*-5%*	0.591	-1%	0.559	*-7%*
		CT	1.2	1.05	**-13%**	0.989	**-18%**	1.2	0%	1.2	0%	1.2	0%
		ALT	0.37	0.336	*-9%*	0.341	*-8%*	0.368	-1%	0.296	**-20%**	0.064	**-83%**
		Mag	1.39	1.19	**-14%**	1.2	**-14%**	1.38	-1%	1.37	-2%	1.33	-5%
Touch Down	Position (m)	DR	839	763	*-9%*	566	**-33%**	834	-1%	835	-1%	781	*-7%*
		CT	650	332	**-49%**	600	*-8%*	650	0%	650	0%	650	0%
		ALT	11.6	11.6	0%	11.6	0%	11.6	0%	9.69	**-17%**	10.7	*-8%*
		Mag	1060	832	**-22%**	824	**-22%**	1060	0%	1060	0%	1020	-4%
	Velocity (m/s)	DR	0.316	0.286	*-10%*	0.315	0%	0.313	-1%	0.314	-1%	0.29	*-8%*
		CT	0.364	0.359	-1%	0.36	-1%	0.364	0%	0.364	0%	0.364	0%
		ALT	0.0846	0.0799	*-5%*	0.0843	0%	0.0842	0%	0.0843	0%	0.00569	**-93%**
		Mag	0.489	0.466	-5%	0.486	-1%	0.487	0%	0.488	0%	0.465	-5%

improving the initial state update and the quality of the gravity model, neither is better considering just the touchdown error, but the combination of the two would yield improvements greater than each separately.

Also, it is seen, as shown in Fig. 7.1, that high-altitude measurements from the altimeter are not required, so long as the altimeter is operated prior to PDI. Varying terrain uncertainty and perfect altimeter quality are shown to produce relatively insignificant changes to the navigation errors, when compared to gravity model compensation and DSN initial state error.

7.2 Final Approach Navigation Study

The final approach navigation study evaluates the hazard-relative horizontal navigation error at touchdown as a function of five trajectory parameters—flight path angle (or glide slope), slant range from final Hazard Detection and Avoidance (HDA) scan to target, acceleration of the vehicle, and initial position and velocity navigation uncertainty.

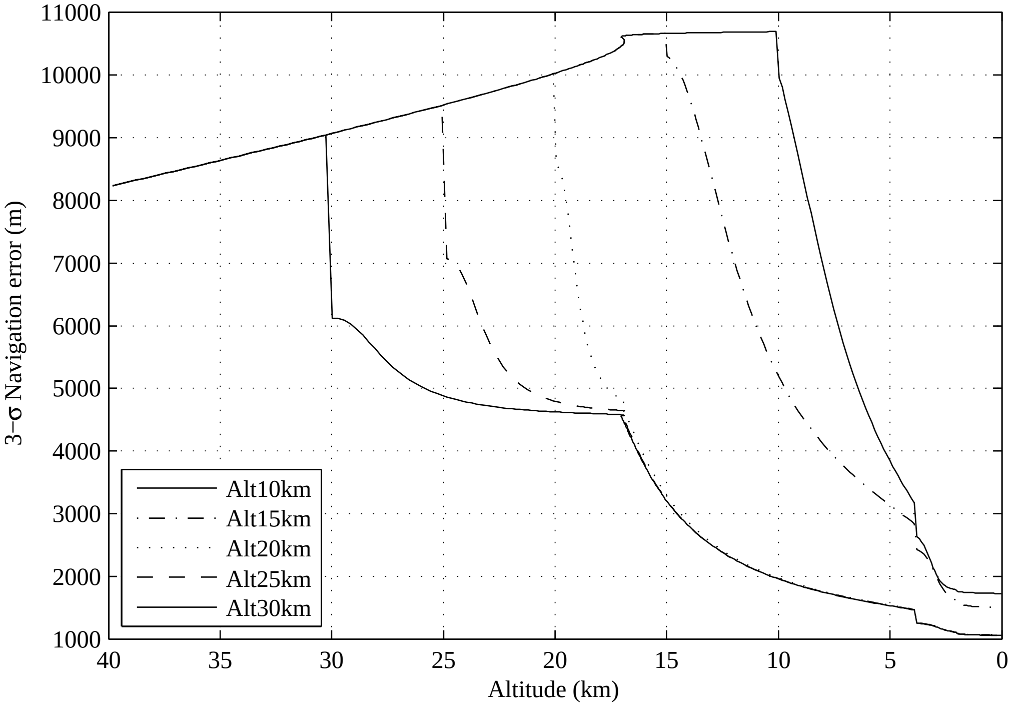

Fig. 7.1: Inertial-only navigation position errors during powered lunar descent for varying altimeter operation altitudes.

That is, for this study, HDA measurements are taken during a defined window, and after the final HDA imaging, the LSAM flies on inertial sensors only, and it is desired to know the navigation uncertainty at touchdown.

Several different trajectories are flown with varying flight-path angles (glide-slopes), thrust-acceleration levels, and using different velocimeters and HDA operation regimes. The analysis parameters are shown in Tables 7.4 and 7.5.

From the final approach study, the horizontal navigation error at touchdown is nearly linearly proportional to the time remaining at the final HDA scan, as shown in Fig. 7.2. As multiple HDA measurements are permitted, the dependency of the final error on the trajectory approach and thrust level is reduced. The capabilities of velocimeter and the HDA system become the driving parameters for the footprint size at touchdown.

Generally, the JPL TDS radar velocimeter system outperformed the LaRC Doppler Lidar velocimeter, especially when few HDA measurements were available. The primary reason for this was the specifications for the radar system were for relatively small bias and large noise, while the Lidar system had medium bias and medium noise. The data show that over time the navigation filter was able to estimate out the noise, while the biases could only be estimated so well since they are permitted to change with a time constant of 100 s.

Using a very conservative value for the size of the touchdown error ellipse, the analysis shows that the LSAM is capable of navigating to within a 1.1-meter radius. The parameter combination giving this value is a 60° approach at 1.05 lunar g's, using the JPL TDS radar system operating to 5 meters altitude, and running HDA scans between 2000 and 30 m slant range from the landing target.

Table 7.4: Final Approach Navigation Analysis Task Comparison

	Case 1	Case 2
HDA sampling	Once	Multiple, $\Delta t = 10\,\text{s}$
HDA operation window (by slant range)	Only during first time step	One of (4): $\begin{cases} {[30, 500]} \\ {[30, 2000]} \\ {[250, 500]} \\ {[250, 2000]} \end{cases}$ m
HDA Specifications (3-σ)	All: $\begin{cases} \text{Scale factor} & 300\,\text{PPM} \\ \text{Alignment} & 0.5/\sqrt{3}\,\text{mrad}/\text{axis} \\ \text{Bias} & 1/\sqrt{6}\,\text{m}/\text{axis} \\ \text{Noise} & 1/\sqrt{6}\,\text{m}/\text{axis} \end{cases}$	
Flight path angles	One of (6): 15°, 30°, 45°, 60°, 75°, 90°	
Thrust acceleration	One of (7): 1.05, 1.1, 1.2, 1.3, 1.4, 1.5, 2.0 lunar-g's	
Initial slant range	One of (10): 2, 1.5, and 1 km; 800, 667, 500, 400, 300, 200, and 100 m	2 km
Initial lander position error (3-σ)	All: 90 m downrange and crosstrack, 30 m altitude	
Initial lander velocity error (3-σ)	One of (2): $\begin{cases} 0.3 \\ 0.6 \end{cases}$ m/s/axis	0.1 m/s/axis
Landing site map-tie error (3-σ)	All: 100 m north/south and east/west, 10 m down	
Velocimeter Type	One of (2): $\begin{cases} \text{JPL TDS Doppler Radar} \\ \text{LaRC Doppler Lidar} \end{cases}$ (see Table 7.5)	
Velocimeter Cutoff Altitude	One of (2): $\begin{cases} \text{Specification (Table 7.5)} \\ 30\,\text{m} \end{cases}$	
Altimeter specs	All: See Table 7.2, Nominal	
Other Environment and Sensor Models	All: See Tables 7.1 and 7.2, Nominal	
Total simulations	3360	672

Table 7.5: Velocimeter Comparison for Final Approach Navigation Analysis

1-σ Specifications	JPL TDS Doppler Radar	LaRC Doppler Lidar
Bias, $\tau = 100\,\text{s}$	0.001 m/s	0.017 m/s
Noise	0.06 m/s	0.017 m/s
Misalignment	50 arcsec	50 arcsec
Min. Operation Altitude	5 m	2 m

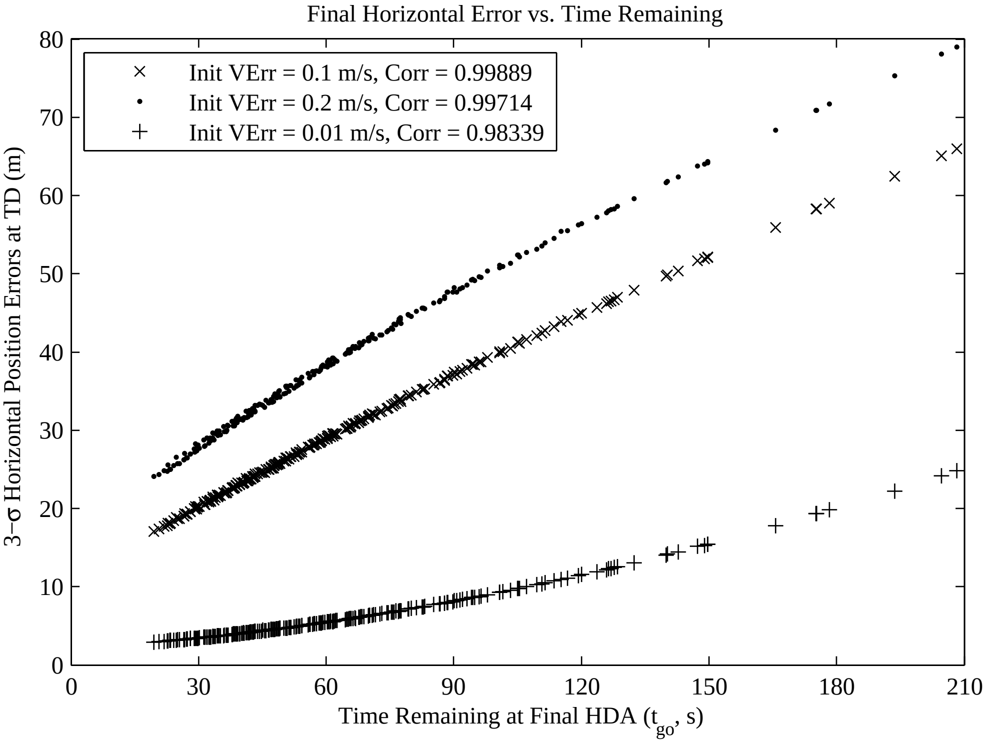

Fig. 7.2: Final approach study indicating high correlation between final navigation position error and the time remaining after the last HDA scan.

Chapter 8

Guidance Studies

As shown in the previous chapter, a powerful application of the LinCov tool is that it lends itself to performing trade studies. In this chapter, we will demonstrate the capability of LinCov to produce dispersion variances for a closed-loop GN&C system and perform trade studies to assess the performance of the system for different GN&C operation options and environment uncertainties.

8.1 Reference Trajectory

The navigation studies from Chapter 7 all utilized nominal reference trajectories developed by NASA/JSC or CSDL. During the progression of this research, the guidance laws used for the Braking Phase of powered descent progressed from the original Apollo-like Linear Acceleration Profile to a more complex law based on Shuttle Powered Explicit Guidance (compare Sostaric [4] and [9]).

This change in guidance laws did not affect the navigation studies because the instrument measurements (and therefore the navigation errors) are not dependent on the guidance law. When proportional-derivative error-correction guidance is applied with appropriate gains, the dispersions behave as would be expected for a PD-control response with characteristic oscillation and settling, no matter the guidance law used to generate the reference trajectory. It was found during the development of the guidance-enhanced LinCov tool, however, that when applying closed-loop laws within LinCov, the trajectory state dispersions become unstable when the applied laws do not match those that were used to generate the reference trajectory, $\bar{\mathbf{x}}$.

Consequently, it was necessary to generate a reference trajectory independent of NASA's baseline in order to complete this research. A new lunar descent trajectory was developed in NASA's three-degree-of-freedom Program to Optimize Simulated Trajectories (POST) using the built-in Generalized Acceleration Steering guidance mode. The Linear Acceleration and Quadratic Acceleration guidance laws described in Section 2.1.2 were implemented through generalized dependent variable (GENV) tables, constructed to produce the inertial guidance acceleration command ${}^{i}\mathbf{a}_{\text{guid}}$. The guidance targets and time to achieve them were treated as independent variables, and trajectory was optimized to produce a reasonable descent trajectory with near minimum ΔV expended.

The reference lunar descent profile is shown in Fig. 8.1 on the following page. The trajectory begins at apolune of a 100×22.5 km descent orbit. At a down-range distance of 500 km (and a near-perilune altitude of 22.65 km), Powered Descent Initiation (PDI) occurs and the Braking Phase begins. After a nine minute Braking Phase, the 20-second Powered Pitch-Up and Throttle-Down (PPU) maneuver begins, during which the vehicle covers 979 m of the remaining 1 368 m range to the landing site and drops from 412 m to 136 m in altitude. The Approach Phase lasts 50 seconds, follows a 15° glideslope, and brings the vehicle to roughly 30 m above the landing site with approximately zero horizontal velocity and a vertical velocity of just under 1 m/s, relative to the surface. The Terminal Descent phase lasts 30 s with the engine throttle keeping the descent rate at a fixed 1 m/s. The thrust acceleration profile and vehicle pitch angle required to achieve this trajectory are shown in Fig. 8.2 on page 96. The nominal ΔV required for this landing trajectory, not including the deorbit burn, is 1 934.4 m/s.

8.2 Baseline GN&C Performance

As with the Navigation Studies in Chapter 7, it will be useful to establish a performance baseline and discuss some of the features of the results. This section will define the baseline perturbation inputs and discuss the results, specifically as it relates to the vehicle state dispersion, ΔV dispersions, and the effect of the PDI trigger.

Tables 8.1 and 8.2 list the nominal perturbation uncertainties of all the input parameters required for our LinCov formulation, in the form of standard deviations, noise

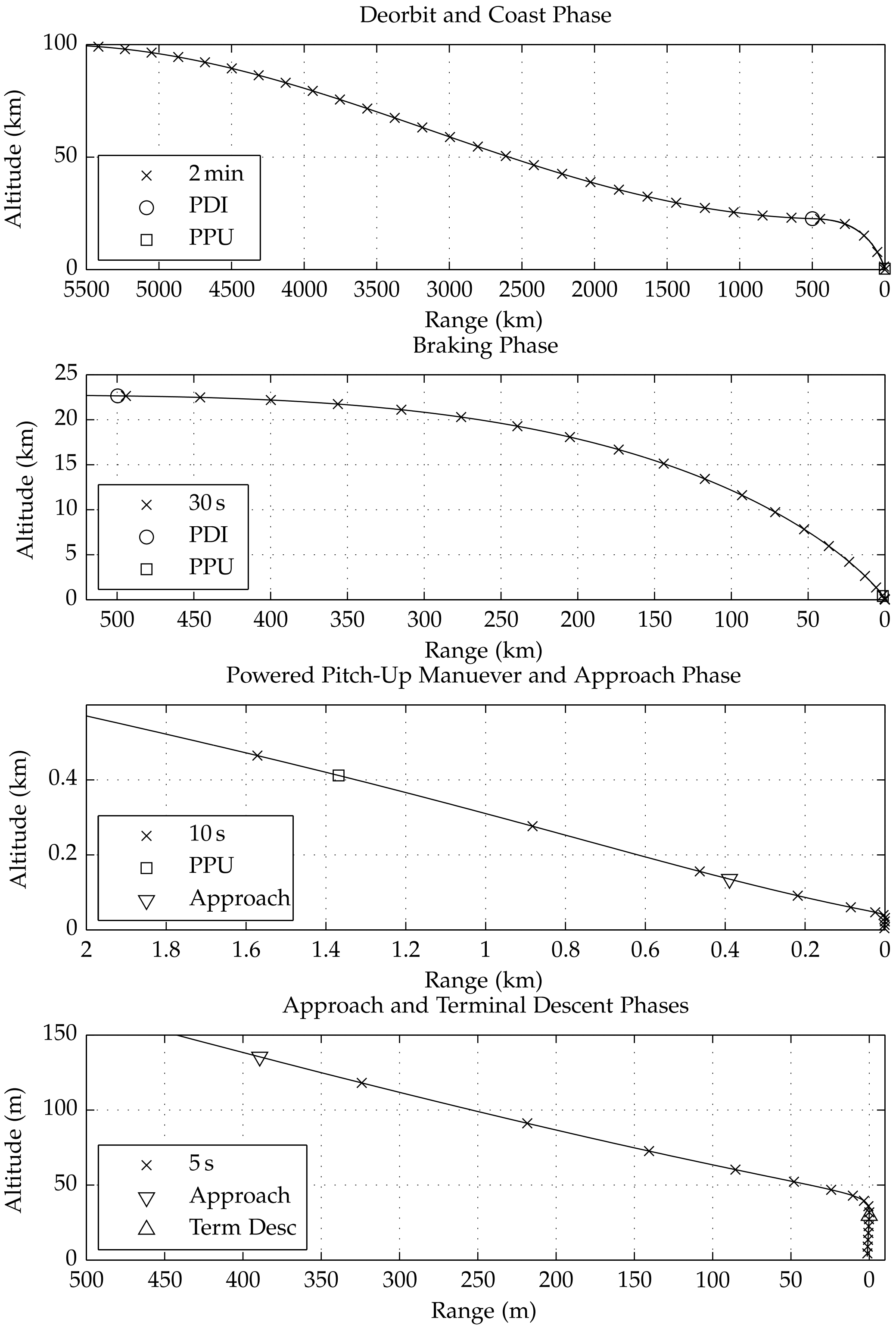

Fig. 8.1: Guidance studies nominal lunar descent trajectory, altitude and range.

Table 8.1: Baseline Environment and Initial Condition Parameter Uncertainties

<table>
<tr><th>Error source</th><th>1σ Uncertainty and Time Constant or Noise Strength</th></tr>
<tr><td>Gravity Model (Masscons), $\boldsymbol{g}_d$</td><td>2 mGal, $d_{\text{corr}} = 400$ km
("Gravity Compensation On" or "Improved")</td></tr>
<tr><td>Map error, $\mathbf{e}_{\text{map}}$</td><td><table>
<tr><th>Range to Landing Site</th><th>> 5 km</th><th>< 5 km</th></tr>
<tr><td>Horizontal Error</td><td>11.67 m</td><td>0.67 m</td></tr>
<tr><td>Vertical Error</td><td>1.167 m</td><td>0.067 m</td></tr>
</table>$d_{\text{corr}} = 4$ km</td></tr>
<tr><td>Random accelerations, $\mathbf{w}_v$, $\mathbf{w}_\omega$</td><td><table>
<tr><td>Gravitational $\mathbf{w}_{\text{grav}}$</td><td>$0.4\,\text{mm/s}/\sqrt{\text{s}}$</td></tr>
<tr><td>Actuator, $\mathbf{w}_{v,\text{actuator}}$</td><td>$0.01\,\text{mm/s}/\sqrt{\text{s}}$</td></tr>
<tr><td>Actuator, $\mathbf{w}_{\omega,\text{actuator}}$</td><td>$0.5\,\mu\text{rad/s}/\sqrt{\text{s}}$</td></tr>
</table></td></tr>
<tr><td>Vehicle Rotational Initial Conditions, $^b\delta\boldsymbol{\theta}$, $^b\delta\boldsymbol{\omega}$</td><td><table>
<tr><td>Attitude</td><td>50 arcsec</td></tr>
<tr><td>Angular Velocity</td><td>5 arcsec/s</td></tr>
</table></td></tr>
<tr><td>Vehicle Translational Initial Conditions (DSN Update), Local Vertical Local Horizontal (LVLH) Frame, $^{\text{LVLH}}\begin{bmatrix} S_r & S_{rv} \\ S_{rv}^{\mathsf{T}} & S_v \end{bmatrix}$ $^{\text{LVLH}}\delta\mathbf{r}$, $^{\text{LVLH}}\delta\mathbf{v}$</td><td>"Intermediate" or "Medium Quality"<table>
<tr><td>$S_r = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_c^2 & 0 \\ 0 & 0 & \sigma_a^2 \end{bmatrix}$</td><td>Downrange $\sigma_r = 1500$ m
Crossrange $\sigma_c = 50$ m
Altitude $\sigma_a = 200$ m</td></tr>
<tr><td>$S_v = \begin{bmatrix} \sigma_{rr}^2 & 0 & 0 \\ 0 & \sigma_{cr}^2 & 0 \\ 0 & 0 & \sigma_{ar}^2 \end{bmatrix}$</td><td>Downrange Rate $\sigma_{rr} = 0.047$ m/s
Crossrange Rate $\sigma_{cr} = 1.5$ m/s
Altitude Rate $\sigma_{ar} = 0.2$ m/s</td></tr>
<tr><td>$S_{rv} = \rho \begin{bmatrix} 0 & 0 & \sigma_r\sigma_{ar} \\ 0 & \sigma_c\sigma_{cr} & 0 \\ \sigma_r\sigma_{ar} & 0 & 0 \end{bmatrix}$</td><td>Correlation Factor $\rho = -0.9$</td></tr>
</table></td></tr>
<tr><td>Landing Site Inertial Position IC, $^b\delta\mathbf{r}_{\text{site}}$</td><td><table>
<tr><td>Horizontal</td><td>33.3 m</td></tr>
<tr><td>Vertical</td><td>3.33 m</td></tr>
</table>Horizontal uncertainty corresponds to a 4 arcsec (1σ) map-tie error for the landing site.</td></tr>
</table>

Table 8.2: Baseline GN&C System Parameter Uncertainties

Error source	1σ Uncertainty and Time Constant or Noise Strength	
Star tracker, ξ_{starcam} ($\Delta t = 2\,\text{s}$)	Alignment, each axis	20 arcsec, $\tau = 10^6$ s
	Noise, each axis	50 arcsec
Gyroscope, ξ_{gyro}	Scale factor	1.6 PPM, $\tau = 10^6$ s
	Alignment	20 arcsec, $\tau = 10^6$ s
	Bias	0.02 arcsec/s, $\tau = 10^6$ s
	Angular random walk	0.000 05 deg/$\sqrt{\text{s}}$
Accelerometer, ξ_{accmtr}	Scale factor	66 PPM, $\tau = 10^6$ s
	Alignment	20 arcsec, $\tau = 10^6$ s
	Bias	30 μg, $\tau = 10^6$ s
	Velocity random walk	0.05 mm/s/$\sqrt{\text{s}}$
Velocimeter, ξ_{velmtr} (Operation altitude: 10 m–2000 m , $\Delta t = 2\,\text{s}$)	Scale factor	0 %, $\tau = 100$ s
	Alignment	1/3 μrad,$\tau = 10^6$ s
	Bias	1 mm/s, $\tau = 100$ s
	Noise	60 mm/s
Altimeter, ξ_{alt} (Operation altitude: 10 m–30 km , $\Delta t = 2\,\text{s}$)	Scale factor	0.1 %, $\tau = 100$ s
	Bias	0.1 m/s, $\tau = 100$ s
	Terrain Bias	33 m, $d_{\text{corr}} = 10$ km
	Vehicle Altitude	Noise
	Above 20 km	10 m
	2-20 km	5 m
	10 m-2000 m	2 m
	Within final landing area (range < 0.5 km), terrain bias uncertainty is reduced to 10 % of original value and measurements become relative to landing site.	
Terrain Relative Navigation (TRN) Lidar, ξ_{TRN} (Operation altitude: 4–6 km , $\Delta t = 10\,\text{s}$)	Scale factor	0.1 %, $\tau = 100$ s
	Alignment	1/3 μrad, $\tau = 10^6$ s
	Bias	1 m, $\tau = 100$ s
	Noise	1 m
Attitude Control System (ACS or Torquer), ξ_{ACS}	Scale factor	33.3 PPM, $\tau = 10^4$ s
	Alignment	1/3 mrad,$\tau = 10^4$ s
	Bias	0.033 μN-m, $\tau = 10^4$ s
	Noise	1.414 μN-m
Main Engine Thrust Acceleration, ξ_{thr}	Scale factor	333 PPM, $\tau = 10^4$ s
	Alignment	1/3 mrad,$\tau = 10^4$ s
	Bias	1/3 mm/s^2, $\tau = 10^4$ s
	Noise	1/3 mm/s^2

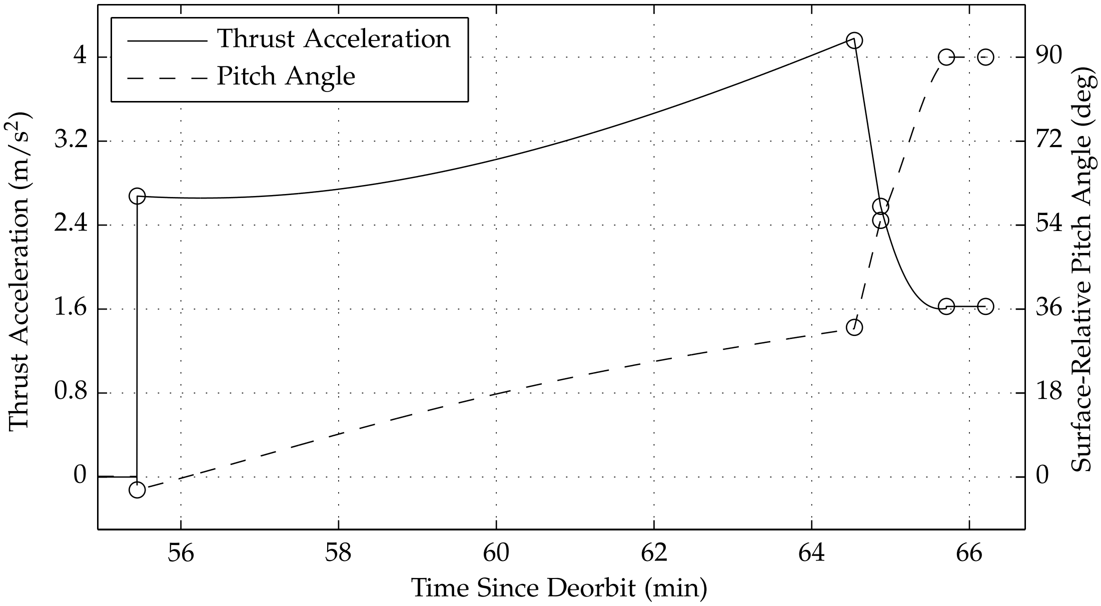

Fig. 8.2: Guidance studies nominal lunar descent trajectory, thrust and pitch.

strengths, time constants or correlation distances for the 1st order Markov processes (also known as ECRVs—Exponentially Correlated Random Variables), and ranges of applicability or operation. As was done with the Navigation Studies, the vehicle inertial translational state initial condition is coordinatized in the Local-Vertical-Local-Horizontal (LVLH) system and then transformed into the inertial frame.

Not included in either table are the gains K_r and K_v from Eqs. (4.17) and (4.18) or K_θ and K_ω from Eq. (4.13). In order to use the proportional-derivative (PD) control laws formulated there, these gains are required. A basic exercise in elementary control theory is the design of gains for a proportional-derivative controller for a linear time-invariant, single-input-single-output (LTI SISO) system. Although the controller model equations which require these gain terms have been linearized, as have the plant dynamics, the closed-loop system is a time-variant, coupled, multiple-input-multiple-output (MIMO) system, and analytical controller design for such a system is beyond the scope of this work. Therefore, we will fall back on elementary LTI SISO methods for PD controller design.

Table 8.3: Baseline Control Gain Characteristics

	Rotational	Translational
Natural Period	$P = 10\,\mathrm{s}$	$P = 30\,\mathrm{s}$
Damping Ratio	$\zeta = 1$	$\zeta = 1$
Natural Frequency	$\omega_n = \frac{2\pi}{10} = 0.628\,\mathrm{rad/s}$	$\omega_n = \frac{2\pi}{30} = 0.209\,\mathrm{rad/s}$
Overshoot	0%	0%
Settling Time	6.37 s	19.1 s

Let us assume (albeit naïvely) that the closed-loop system dynamic system may be modelled by a single-loop second-order system represented, in the Laplace domain, by the transfer function

$$T(s) = \frac{K_P}{s^2 + K_D s + K_p}. \tag{8.1}$$

In terms of response characteristics to a step input, we know from elementary control theory that $K_P = \omega_n^2$ and $K_D = 2\zeta\omega_n$ where ω_n is the natural frequency of the system, and ζ is the damping ratio. The natural period, settling time, and percent overshoot (for underdamped systems) of such a system is given by

$$P = \frac{2\pi}{\omega_n} \qquad T_s = \frac{4}{\zeta\omega_n} \qquad P.O. = (100\%)e^{-\zeta\pi/\sqrt{1-\zeta^2}},\ \zeta < 1.$$

We may then choose K_P and K_D based on the desired response. For the guidance performance study baseline, we choose

$$K_r = 0.3948\,\mathrm{s}^{-2}\mathbf{I}_{3\times3} \;\; K_v = 1.2566\,\mathrm{s}^{-1}\mathbf{I}_{3\times3} \qquad K_\theta = 0.04386\,\mathrm{s}^{-2}\mathbf{I}_{3\times3} \;\; K_\omega = 0.4189\,\mathrm{s}^{-1}\mathbf{I}_{3\times3},$$

where the second-order response characteristics are given in Table 8.3.

Using the baseline specifications described above, the results of LinCov analysis are illustrated in Figures 8.2, 8.4, and 8.5. Figure 8.2 on page 99 shows the magnitude of the position, velocity, and attitude 3σ dispersions for when the PDI range trigger is not used (solid) and when it is used (dashed). The plots on the left side encompass the entire trajectory, including the coast phase, while the right side plots show the details near the end of the trajectory. Vertical dotted lines indicate guidance mode changes, from Coasting Phase to Braking Phase to Powered Pitch-Up to Approach Phase to Terminal Descent. The true and navigation filter errors, which are coincident by design, are the dotted and dash-dot lines of data below the dispersions. Figure 8.4 on page 100 shows the position and velocity LVLH components from PDI to touchdown, using the same legend as Fig. 8.2.

The drop in navigation error at approximately 45 min is the result of the altimeter providing data beginning at an altitude of 30 km. At PDI, the trigger and non-trigger dispersion lines diverge begin to diverge. Note the downrange position dispersion decreases due to the trigger such that the dispersion takes the same value as the downrange position navigation error. This is by design, and indicates that the formulation for the trigger shaping matrix is correct. The drop in the range position dispersion is accompanied by a decrease in the altitude rate error, and otherwise the LVLH dispersions are nearly unchanged.

8.3 TRN Operation Altitude Study

As detailed previously described, the Terrain Relative Navigation (TRN) component of the ALHAT suite of GN&C instrumentation is designed to provide high quality position and velocity data with respect to mapped surface features to the flight computer. As evidenced in the navigation error plots in the previous section, TRN eliminates a large majority of the relative navigation error, in both position and velocity in all three directions, when first activated and regularly sampled. The study described in this section evaluates the sensitivity of the ΔV dispersion to the altitudes at which TRN measurements are available.

Nominally, we defined TRN measurements to be available when the vehicle is between 4000 and 6000 m altitude, with a cycle of 2 s. We will break this study into two parts. In the first we allow the TRN to be operated at higher altitudes keeping 4000 m as the “off” altitude, and in the second we keep the “on” altitude at 6000 m but allow the TRN to operate at lower altitudes. We evaluate these cases both with and without the

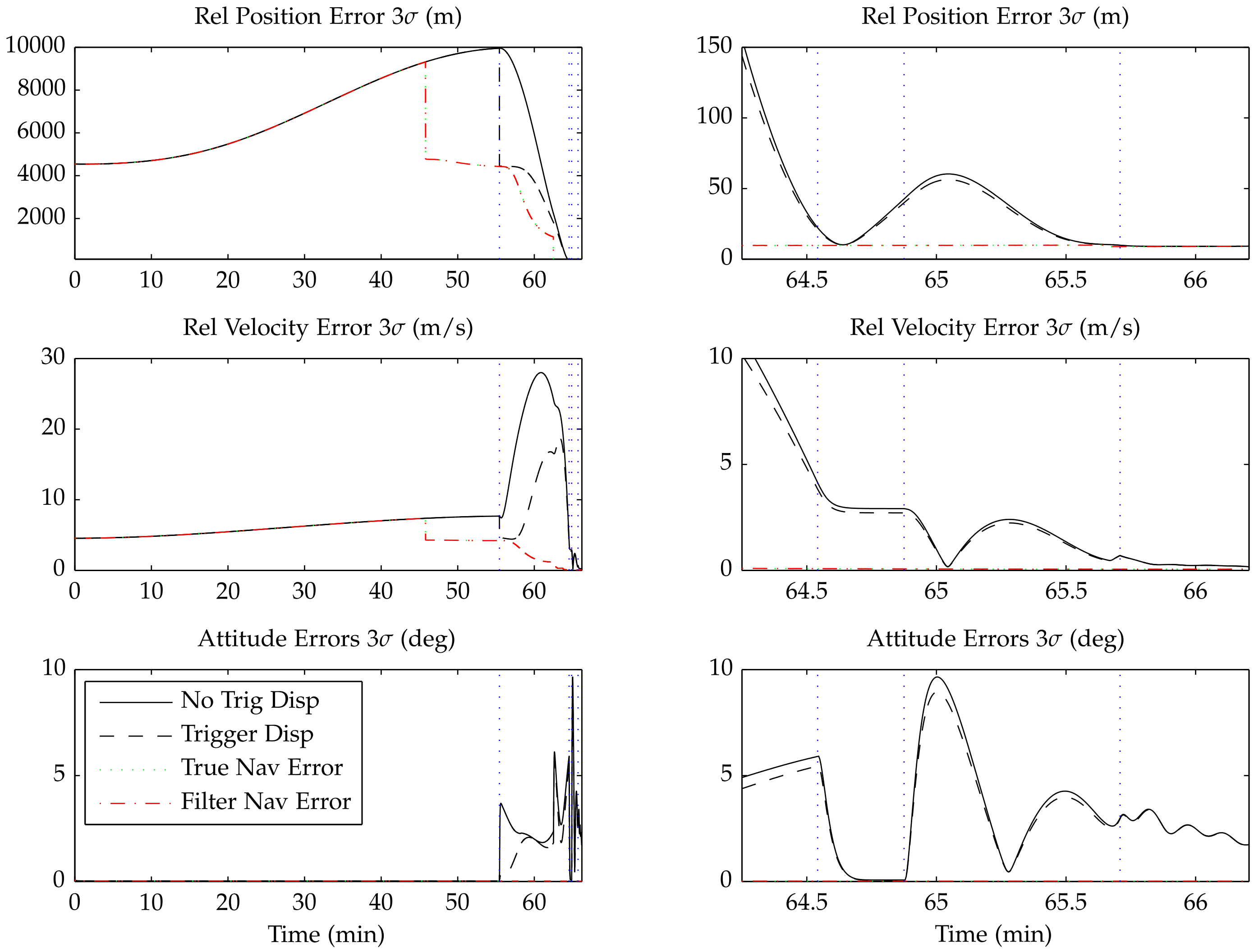

Fig. 8.3: Baseline position, velocity, and attitude dispersion and navigation error magnitudes.

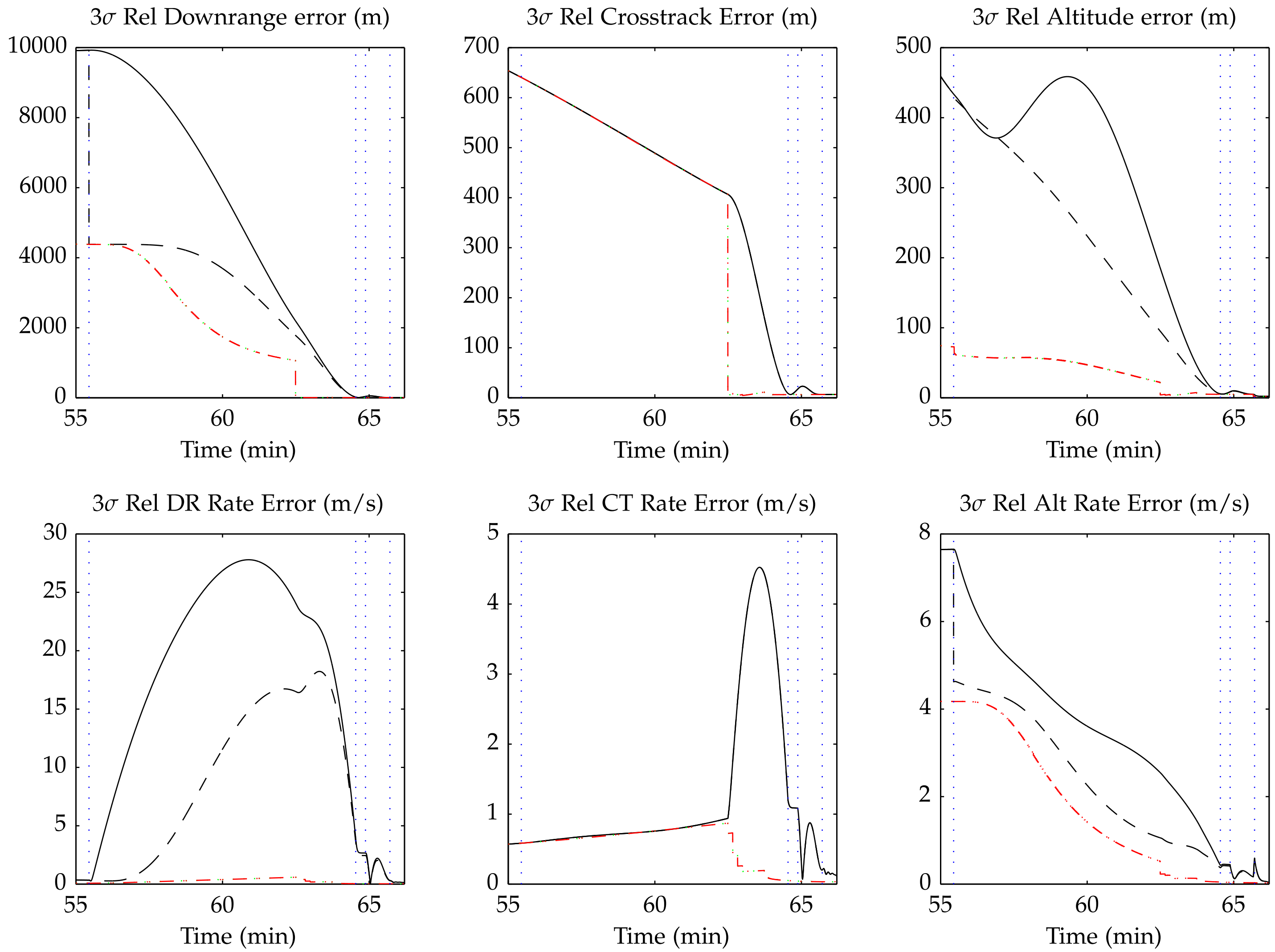

Fig. 8.4: Baseline LVLH dispersions and navigation errors.

PDI range trigger. The results from the two parts of the study are shown respectively in Figures 8.6 and 8.7 on page 108.

In both Figures 8.6 and 8.7, we see that there is essentially no difference between the touchdown position dispersion whether using the PDI trigger or not. As may be expected intuitively, widening the range of TRN operation (by increasing the upper limit in Fig. 8.6, and by decreasing the lower limit in Fig. 8.7) decreases the touchdown dispersion. The zig-zag pattern visible in the position dispersion lines in Fig. 8.6 is due to the discrete number of measurements available increasing between each trial point. The jump in position dispersion off of the left side of Fig. 8.6 and the right side of Fig. 8.7 is due to the TRN operation window closing to nothing, resulting in no TRN measurements being processed—the position dispersion for this case is over 480 m (3σ).

Also seen in Figures 8.6 and 8.7 is the weak sensitivity of the ΔV dispersion to the TRN operation regime. The only interesting feature of these data is the decrease in ΔV dispersion when the TRN is operated above 22.5 km altitude for the PDI trigger case. The savings in propellant is small, as is the decrease if any in touchdown dispersion. Otherwise, ΔV is largely unaffected by the TRN operation region.

Perhaps the most important result from this study is the highly linear nature of the touchdown position dispersion with respect to the lower TRN operation limit. In concurrence with the Final Approach Study in Section 7.2, we find that the touchdown dispersion is largely a function of how close to the surface the TRN (or HDA) instruments can operate, or roughly equivalently, how short a time period remains until touchdown after the final measurement is made. We also see the limited returns in this approach as TRN is permitted through final descent.

8.4 Thrust-Acceleration Uncertainty Study

The nominal uncertainties associated with the Descent Module Main Propulsion system, at the bottom of Table 8.2 on page 95, have remained constant during the previous studies, and their influence on the total ΔV dispersion is yet unknown. This section will

Table 8.4: Thrust-Acceleration Uncertainty Study Parameters, 3σ

Parameter $\boldsymbol{\xi}_{\text{thr}}$	Case 1 (Perfect) $3\sigma_{\text{thr}} = 0$	Case 2 (Nominal) $3\sigma_{\text{thr}} = 0.001$	Case 3 (Large) $3\sigma_{\text{thr}} = 0.01$	Case 4 (Largest) $3\sigma_{\text{thr}} = 0.02$
Scale Factor	0 %	0.1 %	1 %	2 %
Alignment	0 mrad	1 mrad	10 mrad	0.2 rad
Bias	$0\,\text{mm/s}^2$	$1\,\text{mm/s}^2$	$10\,\text{mm/s}^2$	$0.2\,\text{m/s}^2$
Noise	$0\,\text{mm/s}^2$	$1\,\text{mm/s}^2$	$10\,\text{mm/s}^2$	$0.2\,\text{m/s}^2$

demonstrate that the thrust-acceleration uncertainties have a fairly small influence on the overall performance of the GN&C system, as modeled.

For this study, we use all the baseline uncertainty parameters from Section 8.2, except for the four cases in Table 8.4. As we have done previously, we will evaluate the cases with and without the PDI range trigger.

The study results are illustrated in Fig. 8.8 on page 109. The Nominal Case (0.001) is nearly indistinguishable from the Perfect Case as their lines are almost coincident for both with and without the PDI trigger. We see that the Large and Largest Cases converge cleanly to parallel the Perfect Case. For the non-trigger case, the ΔV dispersion for the Largest Case increases by 178 mm/s (7.3%) over the Perfect Case, while for the non-trigger case, the dispersion is increased by 297 mm/s (21.9%).

Touchdown position dispersions (not shown) are very good as well. The Perfect Case brings that dispersion down to a 9.2-m (3σ) radius footprint, while the Largest Case which manages to bring the lander to within a 12.5-m radius footprint.

Each of these observations support the conclusion that thrust-acceleration uncertainties are a small or insignificant contribution to overall GN&C performance, at least when modeled, estimated, and accounted for by the flight software as we have done with this study.

8.5 Multi-Parameter Performance Trade Study

In Section 7.1, we identified two major drivers of inertial navigation error—the initial

Table 8.5: Multi-Parameter Study Trade Space

<table>
<tr><th>Parameter</th><th>Values</th></tr>
<tr><td>Use of PDI Maneuver Trigger</td><td>No Trigger / Great Circle Range Trigger</td></tr>
<tr><td>TRN Scan Cycle Time (Δt)</td><td>10 s / 30 s / 60 s</td></tr>
<tr><td>Gravity Disturbance (masscon) Model Quality</td><td>Unimproved (20 mGal 1σ) / Improved (2 mGal 1σ)</td></tr>
<tr><td>Truth and Filter Vehicle Inertial State Initial Condition (DSN) Covariance Quality, LVLH Axes (1σ, m and m/s)</td><td>
<table>
<tr><th>#</th><th>DR/DRR</th><th>CR/CRR</th><th>ALT/ALTR</th></tr>
<tr><td>1. High Qual</td><td>86/0.0095</td><td>70/0.07</td><td>10/0.08</td></tr>
<tr><td>2. Med Qual</td><td>1 500/0.047</td><td>200/0.2</td><td>50/1.5</td></tr>
<tr><td>3. Low Qual</td><td>10 000/0.28</td><td>1 000/1</td><td>300/9.5</td></tr>
</table>
DR = Down Range Location, DRR = Down Range Rate, CR = Cross Range Location, CRR = Cross Range Rate, ALT = Altitude Location, ALTR = Altitude Rate. Correlation Factor $\rho = -0.9$ between DR and ALTR, CR and CRR, ALT and DRR.
</td></tr>
</table>

navigation filter covariance matrix quality (i.e. the quality of the position and velocity navigation update from the earth-based Deep Space Network or DSN), and the surface-relative gravity model uncertainty due to, e.g., mass concentrations or "masscons." We also have an interest in GN&C design parameters, including how frequently the TRN must provide updates and the PDI trigger. In this section, we look at the effect that these four parameters (DSN quality, gravity model uncertainty, PDI trigger, and TRN operation cycle) have on the performance of the guidance system, in terms of dispersions at touchdown and ΔV expended.

The assumptions of this study are the same as those defined in Section 8.2 and Tables 8.1 and 8.2, except for with the four parameters to be traded. The scope of this study is shown in Table 8.5, where each of the 36 possible permutations is a single LinCov run. The initial condition parameter is traded over three sets of initial vehicle state covariance matrices defined in the LVLH frame. In order of increasing standard deviations, these sets are labeled High Quality, Medium Quality, and Low Quality.

The study took 43 minutes to run with MATLAB on a 64-bit Windows workstation

Table 8.6: Touchdown Position Dispersion (3σ) as a Function of Several Parameters

Position Dispersion Mag. at Touchdown (m, 3σ)		PDI Trigger Type and TRN Cycle Δt					
		No Trigger			Great Circle Range		
Gravity Model	DSN (I.C.)	10 s	30 s	60 s	10 s	30 s	60 s
Unimproved	High Qual	9.65	9.94	12.09	9.65	9.94	12.09
	Med Qual	9.72	10.01	12.25	9.72	10.01	12.25
	Low Qual	9.82	10.11	12.45	9.82	10.11	12.45
Improved	High Qual	8.93	9.22	10.72	8.93	9.22	10.72
	Med Qual	9.20	9.49	11.22	9.20	9.49	11.22
	Low Qual	9.54	9.84	11.90	9.54	9.83	11.90

Table 8.7: Trajectory ΔV Dispersion (3σ) as a Function of Several Parameters

ΔV Dispersion (m/s, 3σ)		PDI Trigger Type and TRN Cycle Δt					
		No Trigger			Great Circle Range		
Gravity Model	DSN (I.C.)	10 s	30 s	60 s	10 s	30 s	60 s
Unimproved	High Qual	2.71	2.71	2.71	3.19	3.19	3.19
	Med Qual	3.61	3.61	3.62	3.43	3.43	3.44
	Low Qual	14.59	14.59	14.60	8.01	8.01	8.04
Improved	High Qual	0.45	0.45	0.46	0.52	0.52	0.52
	Med Qual	2.43	2.43	2.43	1.36	1.36	1.37
	Low Qual	14.34	14.34	14.35	7.36	7.36	7.38

on a single processor of a 2.27 GHz quad-core CPU machine with 12 GB RAM. In terms of traditional GN&C trajectory simulation, this study would require 36 separate Monte Carlo studies, each requiring several hundred individual trajectory simulations, plus additional post-processing runs to compute the variances of the variables of interest.

The summary of results is shown Tables 8.6 and 8.7 on this page. From Table 8.6, we can see that the touchdown position dispersion magnitude (or landing footprint radius) is weakly correlated with the trigger mode and DSN quality, and moderately a function of the TRN operation frequency and gravity model. The small differences in dispersion magnitude between all these cases indicate that the guidance system, as implemented, does a good job controlling dispersions in spite of the various uncertainties studied.

From Table 8.7, we can see the effects that the four varied parameters have on the ΔV dispersion, or the amount of descent stage propellant required for landing. Comparing columns of data, the TRN cycle time apparently plays almost no role in the ΔV consumed. As should be expected, improving the quality of the DSN update provides a sizable improvement in ΔV control, and does improving the gravity model, as long as the gravity model improvement is not masked by a sizable initial uncertainty in the dispersions.

Comparing the left half of the table with the right, we see that for high quality initial conditions, using the range trigger caused a slight increase in ΔV expenditure, but for intermediate and low quality initial conditions, the range trigger use caused an improvement of increasing magnitude as the initial dispersions grew. This interesting result indicates that use of a trigger becomes detrimental if the deorbit dispersions and corresponding navigation update are extremely high quality.

The relative influence of each of the parameters is complex and depends on the specific cases compared, but these data show that low quality (high uncertainty) initial condition cases should be avoided to reduce the uncertainty in ΔV and propellant loading. Since mass to the surface is an important performance metric for a lunar lander like Altair, we may safely conclude that investments in improvements to the Earth-based Deep Space Network (or analogous tracking stations providing state data to the lander) can result in sizable savings in the propellant margin required for landing dispersion control on the lander. With these improvements in place, improved lunar gravity field modeling will provide additional improvements. These results are consistent with the navigation study results in Section 7.1.

In this chapter, we have demonstrated the power and capability of the LinCov tool to perform architecture-level trade studies, appropriate for preliminary GN&C design using closed-loop guidance laws, that would typically require dozens of Monte Carlo analyses to be performed, requiring tens or hundreds of thousands of individual trajectories to be simulated and post-processed. We have also demonstrated how an event trigger affects the trajectory dispersions by shaping the covariance matrix. For most of the studies presented here, using a range-dependent PDI trigger offers an advantage over time-based ignition as it decreases the ΔV dispersions, when compared to the analogous case without the trigger. We have demonstrated in this chapter a consistent convergence of position dispersions at touchdown on the order of 10 meters in most cases. The sensitivity of the

ΔV dispersion and touchdown position dispersion to problem uncertainties may be used by decision makers to define ALHAT or Altair GN&C design requirements, focus efforts more critical areas, or direct future research, in as much as this study is still applicable to NASA's lander and descent trajectory.

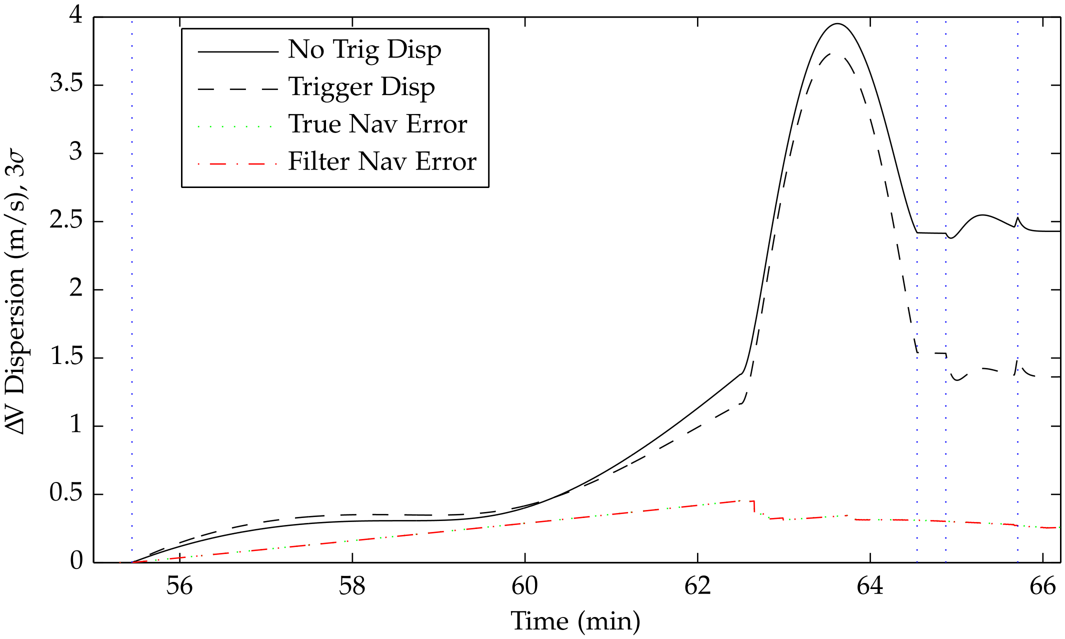

Fig. 8.5: Baseline ΔV dispersion and navigation error.

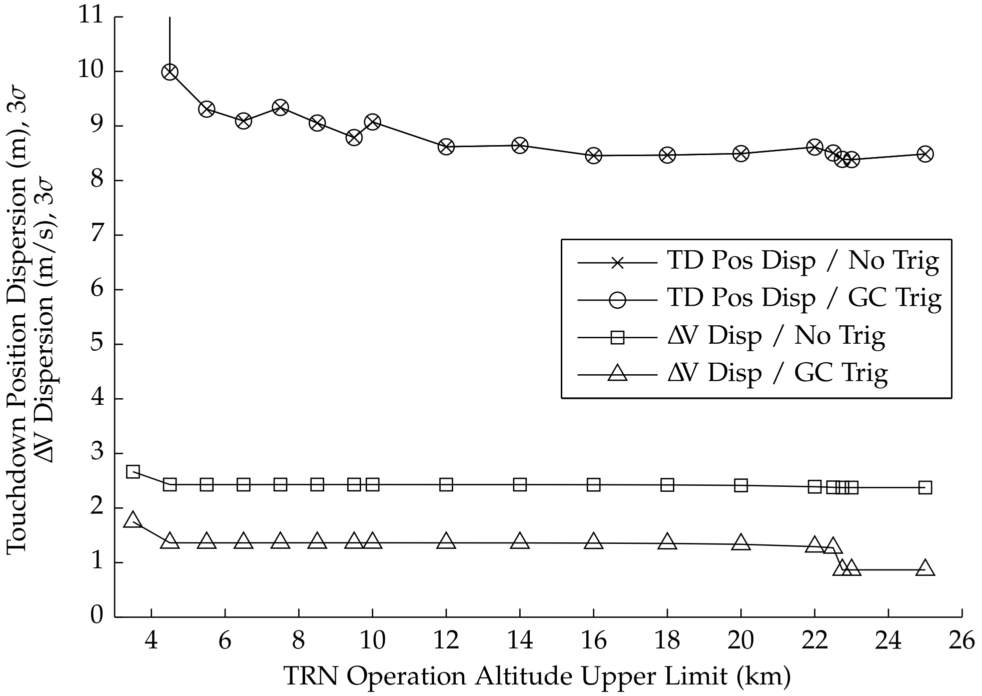

Fig. 8.6: Effect of TRN operation upper limit on ΔV and touchdown dispersions.

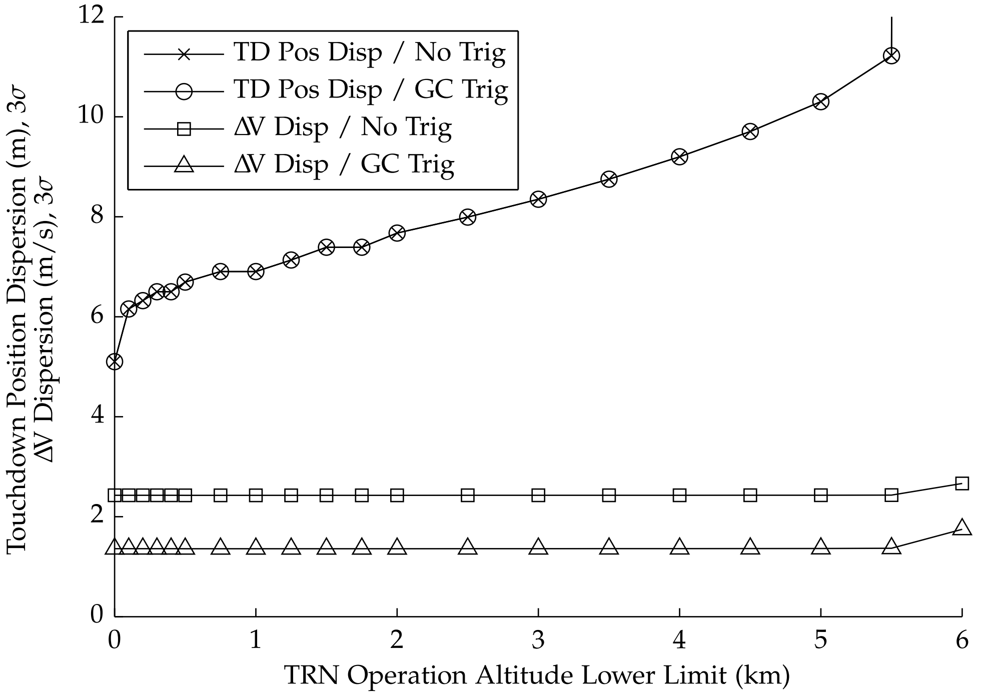

Fig. 8.7: Effect of TRN operation lower limit on ΔV and touchdown dispersions.

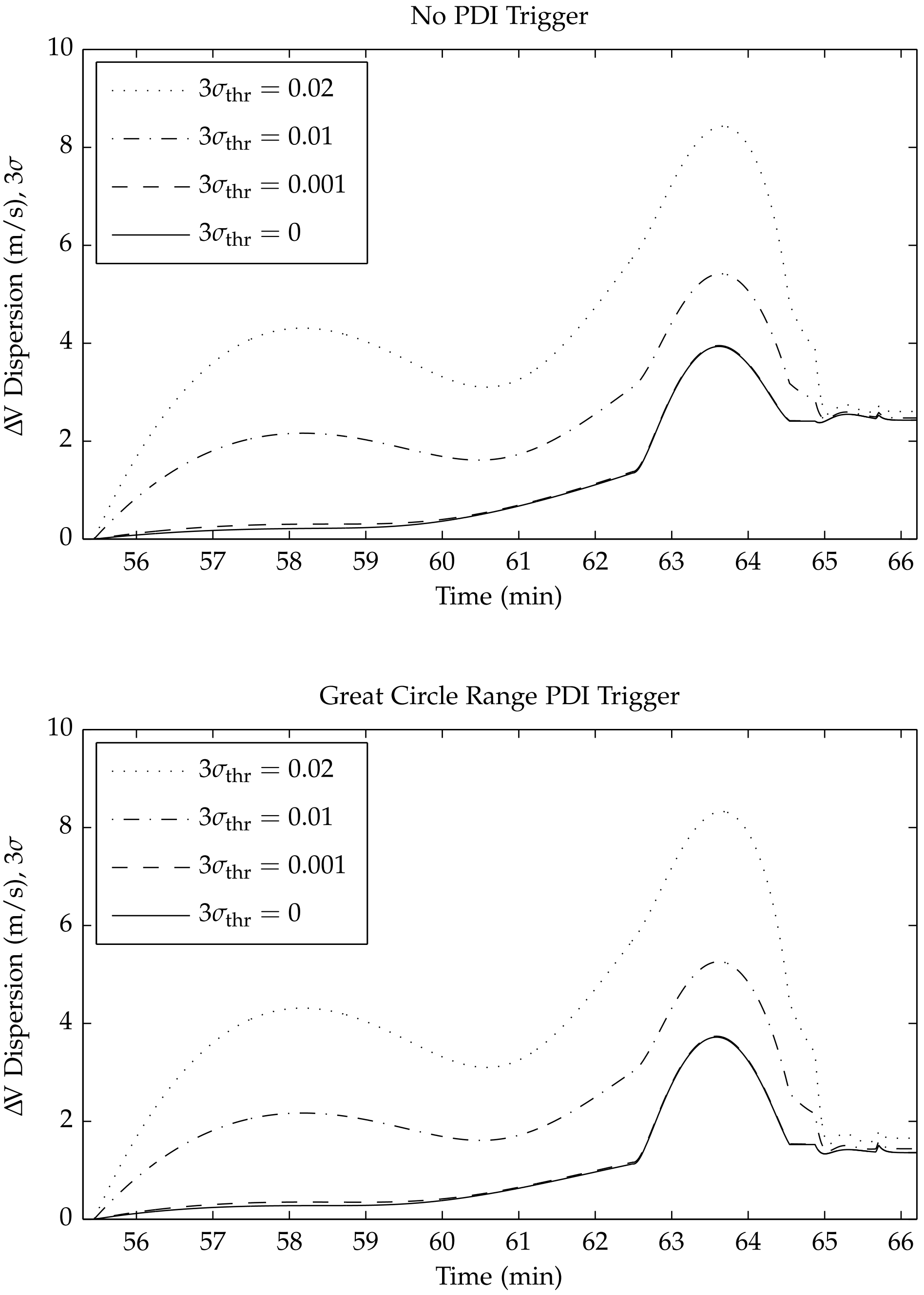

Fig. 8.8: Effect of thrust-acceleration uncertainties on ΔV dispersion.

Chapter 9

Conclusion

The primary goal of the research presented here is to validate the capabilities of the proposed Altair GN&C system using the novel Linear Covariance analysis technique. In as much as the assumptions presented with each study are valid or at least reasonably close to reality, this research has demonstrated that Altair and its GN&C are capable of guiding the lander as accurately as it can estimate its state, or navigate, relative to the landing site, and landing footprints of less than 100 meters, the NASA goal, are well within the capability of the GN&C system using Terrain Relative Navigation (TRN).

The ALHAT suite of instruments, it has been shown, can navigate to touchdown to within 10 meters (3σ relative error) on typical assumptions, and 3- and 1-meter accuracy is possible under generous assumptions and slow, steep trajectories. These accuracies, using the TRN system, are orders of magnitude better than the 1-km at touchdown baseline case for inertial-only navigation.

Additionally, we have shown that this level of accuracy in dispersion control is attainable with very small ΔV margin requirements. The typical ΔV dispersion computed for descent was approximately 2.5 m/s and 13 m/s (for non-triggered and triggered cases, respectively), which is 0.13 % and 0.67 % of the nominal ΔV requirement for the descent trajectory studied. These margins are small compared to typical space propulsion systems, a testament to the robustness and optimality of the guidance laws used, but also a red flag for engineers who rightly err on the side of caution and conservatism when designing such systems. Independent validation of these results, or at least a reexamination of the assumptions and methods, is not unwarranted.

One major advantage of LinCov analysis over Monte Carlo analysis is that it lends itself to parameter trade studies in a comparatively short amount of time. This capability was used to show the relative impact on GN&C performance caused by different instrument models or errors or operation modes, environment uncertainties, variations in approach trajectory, thruster uncertainties, and so forth. We have seen that the initial dispersions of the vehicle in orbit and gravity model of the moon have a much larger influence on the navigation errors and trajectory dispersions than do individual components of the lander.

The benefits of LinCov analysis do not come without a price, however, as Chapters 2-6 of this work evidence. A significant amount of formulation and development is required before such a tool may be used. Often, validated nonlinear simulation tools already exists for a problem of interest (and in fact, if they did not exist they must be developed anyway since LinCov requires the use of a nominal, unperturbed reference state or trajectory to operate), and the addition of a Monte Carlo driver in the form of a wrapper or loop may be simpler and less time consuming than developing an appropriate LinCov tool. While we have presented analytical linearization of the equations, it is conceivable (and demonstrated outside of this research) that numerical derivatives may be used in place of the partial derivatives in the Taylor's series expansions. It is ultimately in the hands of those tasked to perform analysis to decide whether LinCov analysis makes sense for their specific problem.

During the course of this research, the Constellation Project continued development of its vehicles and systems towards NASA's defined goal of returning Americans to the moon. Years of engineering effort have led to the development of mature designs of the Crew Exploration Vehicle *Orion* and the Crew Launch Vehicle *Ares I*, as well as refined concepts of the Cargo Launch Vehicle *Ares V* and the Lunar Surface Exploration Module *Altair*. In support of these vehicles, numerous systems and subsystems have been designed and built, new facilities erected and existing ones refurbished, and sub- and full-scale tests have been conducted, including the launch of the Ares I-X prototype test vehicle on 28 October 2009.

In May 2009, the White House Office of Science and Technology Policy ordered the formation of the Review of United States Human Space Flight Plans Committee (the so-called "Augustine Commission" after its chairman former Lockheed-Martin CEO Norm Augustine) to review the Constellation Project and evaluate the cost and schedule of alternative directions for U.S. manned space flight to and beyond low earth orbit after the retirement of the Space Shuttle in 2010. The Committee concluded in its September 2009 report that the "program of record," Constellation, was underfunded, and proposed three alternative exploration options. On 01 February 2010, President Barack Obama announced the cancellation of the Constellation Project in his budget proposal to Congress for Government Fiscal Year 2011. Congressional support for Constellation is bipartisan, and opposition to the proposed cancellation has been voiced by members of Congress from across the nation and political spectrum. It is uncertain at the time of writing what the ultimate fate of Altair and its technologies, or any of the other Constellation vehicles, shall be.

It is the hope of the author that Constellation continues forward to the moon or that, whatever the path the U.S. pursues for human spaceflight, Constellation-derived vehicles and technologies be a major part of that path. Whether or not the U.S. returns to the moon, creative and innovative methods for designing and analyzing engineered systems, such as Linear Covariance Analysis specifically for Guidance Navigation and Control systems, will necessarily play a vital role in any future spaceflight endeavor.

References

[1] Maybeck, P. S., *Stochastic Models, Estimation, and Control*, Vol. 141 of *Mathematics in Science and Engineering*, Academic Press, New York, 1979.

[2] Geller, D. K., "Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1404–1414.

[3] Bryson, A. E. and Ho, Y.-C., *Applied Optimal Control*, Hemisphere Publishing Corporation, 1975.

[4] Sostaric, R. R. and Rea, J. R., "Powered Descent Guidance Methods for the Moon and Mars," *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, Vol. 6, 2005, pp. 4495–4514.

[5] Cherry, G. W., "E Guidance — a General Explicit, Optimizing Guidance Law for Rocket-Propelled Spacecraft," Tech. Rep. R-539, MIT Instrumentation Laboratory, March 1966.

[6] Klumpp, A. R., "A Manually Retargeted Automatic Landing System for the LM," Tech. Rep. R-539 Rev. 1, MIT Instrumentation Laboratory, August 1967.

[7] Klumpp, A. R., "Apollo Lunar-Descent Guidance," Tech. Rep. R-695, MIT Charles Stark Draper Laboratory, June 1971.

[8] Klumpp, A. R., "Apollo Lunar Descent Guidance," *Automatica*, Vol. 10, No. 2, 1974, pp. 133–146.

[9] Sostaric, R. R., "Powered Descent Trajectory Guidance and Some Considerations for Human Lunar Landing," *30th Annual AAS Guidance and Control Conference*, 2007.

[10] Lawden, D. F., *Optimal Trajectories for Space Navigation*, Butterworths, London, 1963.

[11] Martin, D. T., Sievers, R. F., O'Brien, R. M., and Rice, A. F., "Saturn V Guidance, Navigation, and Targeting," *Journal of Spacecraft and Rockets*, Vol. 4, No. 7, July 1967, pp. 891–898.

[12] McHenry, R. L., Brand, T. J., Long, A. D., Cockrell, B. F., and Thibodeau III, J. R., "Space Shuttle Ascent Guidance, Navigation, and Control," *Journal of Astronautical Sciences*, Vol. 28, No. 1, January-March 1979, pp. 1–38.

[13] D'Souza, C. N., "An Optimal Guidance Law for Planetary Landing," *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, Vol. 3, 1997, pp. 1377–1382.

[14] Gelb, A., editor, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

[15] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, American Institute of Aeronautics and Astronautics, Reston, VA, 1987.

[16] Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, Chapman & Hall/CRC Applied Mathematics and Nonlinear Science Series, Boca Raton, 2004.

[17] Geller, D., "Analysis of the Relative Attitude Estimation and Control Problem for Satellite Inspection and Orbital Rendezvous," *29th Annual AAS Guidance and Control Conference*, 2006.

[18] Tapley, B. D., Shutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Elsevier Academic Press, Burlington, MA, 2004.

[19] Geller, D., "Linear Covariance Analysis for Lunar Powered Descent and Landing Navigation," *30th Annual AAS Guidance and Control Conference*, Breckenridge, CO, United States, 2007.

[20] Gossner, J. R., *Analytic Method of Propagating a Covariance Matrix to a Maneuver Condition for Linear Covariance Analysis During Rendezvous*, Master's thesis, Massachusetts Institute of Technology, 1991.

[21] Geller, D. K., "Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning," *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, Vol. 1, 2005, pp. 424–444.

[22] Bate, R. R., Mueller, D. D., and White, J. E., *Fundamentals of Astrodynamics*, Dover Publications, Inc., Mineola, NY, 1971.

[23] Roncoli, R. B., "Lunar Constants and Models Document," Tech. Rep. JPL D-32296, Jet Propulsion Laboratory, 2005.

[24] Pittelkau, M. E., "Rotation Vector in Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, November-December 2003 2003, pp. 855–860.

[25] Sidi, M. J., *Spacecraft Dynamics & Controls*, Cambridge University Press, Cambridge, England, 1997.

[26] Bortz, J. E., "A New Mathematical Formulation for Strapdown Inertial Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-7, No. 1, 1971, pp. 61–66.

[27] Fortescue, P., Stark, J., and Swinerd, G., editors, *Spacecraft Systems Engineering*, John Wiley & Sons Ltd., West Sussex, 2003.

[28] Brady, T., Schwartz, J., and Tillier, C., "System Architechure and Operaterational Concept for an Autonomous Precision Lunar Landing System," *30th Annual AAS Guidance and Control Conference*, No. AAS 07-053, 2007.