Lunar Surface Navigation Using Gravity and Star Tracker Measurements

Thomas J. Taylor II

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Aerospace Engineering

Riley M. Fitzgerald, Chair
Mark L. Psiaki
Harry P. Artis

April 28, 2025
Blacksburg, Virginia

Keywords: Kalman filter, Quaternion attitude estimation, Lunar gravity, Celestial navigation, Gravity Bias

Lunar Surface Navigation Using Gravity and Star Tracker Measurements

Thomas J. Taylor II

(ABSTRACT)

This thesis develops and analyzes a method that uses measured gravity and starlight vectors to provide position and attitude estimates, given a predefined high-fidelity lunar gravity field. The gravity field uses provided values for the surface gravity magnitude (g_{loc}) and the East-West (EW) and North-South (NS) deflections of the gravity vector across the lunar surface. These gravity measurements are location-specific and are shown to be influenced by nearby craters and other topographic features. Gravity is more concentrated in these regions, causing the gravity vector to deflect in their direction. The gravity field used in this study (Lunar Gravity Model 2011 (LGM2011)) was developed specifically for surface use, resulting in more accurate gravity measurements compared to other models calculated using spherical harmonic expansions. Future lunar missions prompt the need for new surface navigation techniques. Current position and orientation (POSE) methods employ star trackers (ST) and accelerometers but do not incorporate the use of an external gravity field. The Multiplicative Extended Kalman Filter (MEKF) developed in this study uses these sensors and their data to define a state consisting of the attitude error vector, $\underline{\boldsymbol{a}}$, between the estimated quaternion—found using Shuster's Quaternion Estimator (QUEST) algorithm—and the reference quaternion, the longitude (λ) and latitude (ϕ) positions, and the dynamic accelerometer bias, $\underline{\boldsymbol{b}}_a$. The MEKF assumes constant state propagation between time steps and a covariance update influenced solely by the Moon's rotation. It also assumes that the

reference gravity field is perfectly accurate. However, this assumption introduces inaccuracies in the position estimates due to discrepancies between the true measured gravity and the gravity predicted by the model. These biased position estimates, along with a known accurate reference position, are then used to solve for the correlated gravity bias in the lunar gravity field at various waypoints during a surface mission. As a test case, the reference position is determined using two-way ranging measurements between the ground system and the Lunar Pathfinder satellite. These measurements are processed through a combined weighted batch-to-EKF filter to produce the reference position. A least-squares cost function is then formulated using the MEKF outputs and reference positions to estimate biases in the gravity field. The gravity bias algorithms presented are demonstrated to enable successful surface navigation for a roving mission on the lunar surface.

Lunar Surface Navigation Using Gravity and Star Tracker Measurements

Thomas J. Taylor II

(GENERAL AUDIENCE ABSTRACT)

This thesis develops a method to provide position and system orientation estimates using gravity measurements and star images, given a predefined model of the Moon's surface gravity. The star tracker (ST) identifies stars in captured pixel images based on their brightness and a known star catalog. Unlike the typical assumption that gravity vectors point directly toward the center of the Moon, the actual surface gravity vectors are deflected slightly by nearby topographic features, resulting in distinct measurements at different surface locations. As a result, the gravity vectors become location-specific and useful for identifying particular surface regions. Future lunar missions require reliable techniques to determine position on the surface. Although ST's and accelerometers are already employed in existing positioning algorithms, they do not incorporate an external gravity field. In this thesis, ST and accelerometer data are used to define a state vector consisting of the orientation error vector, the latitude and longitude positions, and the inherent accelerometer bias. While accelerometers are generally calibrated and have known levels of uncertainty, the unknown value of $\underline{\boldsymbol{b}}_a$ can significantly degrade gravity-based measurements if not properly estimated. This bias is estimated through a series of rotations, with measurements gathered over time to improve accuracy. The Multiplicative Extended Kalman Filter (MEKF) presented in this study uses gravity and starlight vectors to estimate final position, orientation, and accelerometer bias. Gravity vectors are filtered with respect to a known gravity field, as they are specific to locations on the lunar surface. Starlight vectors, measured by ST, are used to estimate the system's orientation by relating the Moon's inertial frame to the system's body frame. Accelerometer bias is estimated through a series of rotations that track how gravity measurements from the accelerometer vary with orientation. This thesis also introduces a method for identifying errors in the lunar gravity field by comparing MEKF-derived position estimates at various surface waypoints with a highly accurate reference position. This reference is computed using a lunar satellite, specifically the Lunar Pathfinder in this study. The MEKF and reference positions are then used to isolate position errors arising from inaccuracies in the gravity model. The correlated gravity bias method developed in this study demonstrates its effectiveness in producing accurate position estimates for lunar surface navigation.

Acknowledgments

This thesis would not have been possible without the invaluable guidance of my advisor, Riley Fitzgerald. Your mentorship, encouragement, and unwavering support have made a profound impact on both my academic and personal growth. Your passion for teaching and dedication to your students is truly inspiring, and I am incredibly grateful for the time and effort you invested in helping me succeed. I would also like to acknowledge the many teachers I've had over the years who went above and beyond to make learning engaging and meaningful. Your dedication to your students has not gone unnoticed. You play an essential role in shaping young minds and are important to so many. Lastly, I extend my deepest gratitude to my family and friends for their constant support. Through long nights and difficult times, you stood by me, offering encouragement and motivation when I needed it most. I wouldn't be who I am today without your belief in me and your encouragement to keep striving toward my goals.

Contents

3 Methodology 22

List of Figures

List of Tables

List of Abbreviations

Acronyms

APR	Apollo Peak Ring
BPSK	Binary Phase-Shift Keying
CelNav	Celestial Navigation
CM	Center of Mass
DE	Development Ephemeris
DLL	Delay Lack Loop
DSN	Deep Space Network
ECI	Earth-Centered Inertial
EGM2008	Earth Gravitational Model 2008
EIRP	Effective Isotropic Radiated Power
EKF	Extended Kalman Filter
ELFO	Elliptical Lunar Frozen Orbit
ENU	East-North-Up
EW	East-West
FOV	Field of View

GDOP	Geometric Dilution of Precision
GGOS	Global Geodetic Observing System
GNSS	Global Navigation Satellite System
GOCE	Gravity Field and Steady-State Ocean Circulation Explorer
GRACE	Gravity Recovery and Climate Experiment
GRAIL	Gravity Recovery and Interior Laboratory
ICRF	International Celestial Reference Frame
INS	Inertial Navigation System
JPL	Jet Propulsion Laboratory
LGM2011	Lunar Gravity Model 2011
LGRS	Lunar Gravity Ranging System
LOLA	Lunar Orbiter Laser Altimeter
LOS	Line of Sight
LP	Lunar Prospector
MCI	Moon-Centered Inertial
MCMF	Moon-Centered Moon-Fixed
ME	Mean-Earth
MEKF	Multiplicative Extended Kalman Filter
NASA	National Aeronautics and Space Administration

NS	North-South
ODP	Orbit Determination Program
PA	Principal Axis
PDF	Probability Density Function
PDOP	Precision Dilution of Precision
POSE	Position and Orientation
QUEST	Quaternion Estimator
RCN	Real-Time CelNav
SCPO	Simultaneous Positioning and Orientation
SELENE	Selenological and Engineering Explorer
SLS	Space Launch System
ST	Star Tracker
UKF	Unscented Kalman Filter

Notation

$\underline{\bar{\boldsymbol{x}}}$	Bar vector notation indicates a propagated value
$\bar{x}$	Non-bold bar vector denotes a quaternion
$\bar{x}^c$	Non-bold bar vector with superscript c denotes the conjugate a quaternion
$\underline{\boldsymbol{x}}$	Bold and underlined italic font denotes a vector
$\underline{\boldsymbol{x}}^T$	Transpose of a vector or matrix

$\dot{x}$	A dot denotes the derivative with respect to time
$\underline{\hat{\boldsymbol{x}}}$	Vector notation with a hat denotes an estimate or unit vector
$R_{\mathrm{X}}^{\mathrm{Y}}$	Capitalized R with subscript X and superscript Y denotes a rotation matrix from frame X to frame Y
x^{c}	Superscript c denotes the conjugate of a quaternion

Common Symbols

$\underline{\boldsymbol{a}}$	Attitude Error Vector
$\underline{\boldsymbol{b}}_a$	Accelerometer Bias
λ	Longitude
ϕ	Latitude
$c(d)$	Correlated Gravity Function

Chapter 1

Introduction

Society's interest in deep-space exploration continues to grow, with particular emphasis on returning to the Moon and establishing a sustained presence. Humans have not set foot on the lunar surface since the Apollo 17 mission in December 1972, but the motivation to return remains strong. Since the final Apollo missions, the National Aeronautics and Space Administration (NASA) has explored numerous architectural approaches for resuming crewed lunar missions, with the ultimate objective of reaching Mars [1]. These efforts led to the development of the Moon-to-Mars architecture, which outlines a comprehensive framework for scientific exploration, infrastructure development, and transportation systems that support long-term habitation. This thesis focuses on the first essential component of this architecture—returning to the Moon—and highlights the need for innovative surface navigation techniques independent of significant external infrastructure.

NASA's Artemis program seeks to collaborate with both commercial and international partners to establish a long-term presence on the Moon [2]. Lunar exploration offers not only significant scientific potential but also economic opportunities, prompting the development of a lunar economy that will require new industries and a skilled workforce. NASA's Space Launch System (SLS) provides the launch capability necessary to send crews to the Moon. The Orion spacecraft will transport astronauts to lunar orbit, where it will dock with the Lunar Gateway space station orbiting the Moon [3]. Commercial partners such as SpaceX and Blue Origin will provide the lunar landers that will carry astronauts from orbit to the surface. Once on the surface, astronauts and robotic rovers will require accurate navigation techniques to operate around the landing site.

In addition to government-led missions, there have been commercial lunar landings. Firefly Aerospace recently achieved a successful landing and completed 14 days of surface operations [4], marking the first fully successful commercial lunar mission. Other companies have attempted landings but encountered challenges such as lander instability or tipping at touchdown [5]. While commercial missions offer a lower-cost alternative, they also carry higher risks. Nevertheless, their growing involvement in lunar exploration highlights the increasing private sector commitment. Once commercial landers can consistently achieve successful landings and complete mission objectives, they stand to benefit significantly from the surface navigation techniques presented in this thesis. If the lander is equipped with an accelerometer and a ST to measure gravity and starlight vectors, these measurements, along with a known lunar gravity field model, can be used to estimate their position and orientation accurately using the MEKF. The only additional requirement is that the accelerometer must be capable of rotating to allow for the estimation of accelerometer bias.

Currently, the Moon lacks a Global Navigation Satellite System (GNSS) equivalent, making highly accurate localization methods on the lunar surface unavailable. However, two reliable data sources are always accessible to a lunar surface system: gravity and starlight vectors. These consistent measurements serve as the foundation for the algorithms developed in this study. While previous research has explored the use of these data for localization, this work introduces a novel component: the incorporation of known high-fidelity surface gravity vector models. By leveraging a predefined lunar gravity model, the proposed algorithm estimates the system's position, attitude, and accelerometer bias. This additional reference information enables improved estimation accuracy at each location.

This thesis presents two main algorithmic components: one that uses stationary measurements at a single location over time, and another that applies the stationary procedure at multiple waypoints during a traverse mission. The stationary algorithm (based on the MEKF) uses measured gravity and starlight vectors to determine position and attitude. Attitude is represented as a quaternion and computed using Shuster's QUEST algorithm [6], which calculates the optimal rotation from the Moon-Centered Inertial (MCI) frame to the local body frame. The position and accelerometer bias are then estimated using a MEKF, which incorporates these vectors and a series of rotations to refine the estimates.

Exploration beyond the landing site is essential for future lunar settlements. Surrounding terrain must be studied to assess its potential to support or hinder expansion. The traverse mission algorithm developed in this thesis estimates biases in the gravity model by combining multiple Celestial Navigation (CelNav) measurements obtained using the MEKF. These measurements are then compared to an accurate reference position. This reference is provided by the future Lunar Pathfinder satellite, which will enable two-way ranging with ground-based systems. The resulting data are processed through a combined weighted batch and Extended Kalman Filter (EKF) to determine the state position in the Moon's Principal Axis (PA) frame. Biases in the gravity field are then computed using a least-squares cost function derived using the MEKF and reference position estimates. These biases can be used to improve gravity-based position estimates. Furthermore, as astronauts or rovers traverse the surface, the algorithms can be used to tag samples and geographic features for future scientific analysis.

This thesis follows the structure shown below:

- Chapter 1 has shown how the increased interest in lunar exploration leads to the need for accurate lunar surface navigation techniques, and has described the MEKF and correlated gravity bias methods used to calculate the system's attitude and position.

- Chapter 2 will go over the background information on known lunar gravity field models and other attitude and positioning algorithms.
- Chapter 3 will provide a step by step outline of the the algorithms used to solve for the position, attitude, accelerometer bias, and correlated gravity bias estimates.
- Chapter 4 will go over the different test cases used to analyze the performance of the algorithms and how they vary based on certain conditions.
- Chapter 5 will provide a summary of the results, discuss the limiting factors of the algorithms, and suggest future work based on this study.

Chapter 2

Review of Literature

This chapter provides background information on the material presented in this study. Because the navigation method developed in this work is dependent on lunar surface gravity data, we begin by discussing available gravity models and comparing them to the selected LGM2011 model. Next, other lunar positioning alternatives are examined to identify their limitations and explain why the improved MEKF algorithm in Section 3.7 is necessary.

2.1 Lunar Gravity Models

Several different lunar gravity models have been developed over the years as interest in lunar exploration increased. Understanding their benefits and how they were derived is important in selecting which is most applicable to the mission at hand. Gravity fields have been derived using spherical harmonic formulations. A brief description of spherical harmonics is defined here with more detailed mathematical definitions shown in Appendix A. These gravity models include GLGM-2, LP100K, GRGM1200A, and the eventually selected LGM2011.

2.1.1 Spherical Harmonics

Spherical harmonics are commonly used to calculate gravity fields of celestial bodies, such as the Earth, the Moon, Mars, and various comets/asteroids. Spherical harmonics follow the assumption that every material point of the body is contained within a Brillouin sphere, centered at the body's center of mass [7]. A visualization of the Brillouin sphere is shown in Figure 2.1.

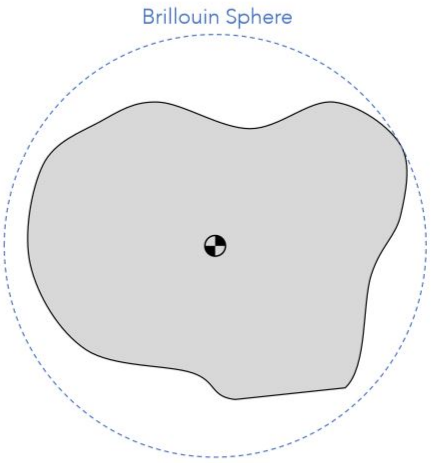


Figure 2.1: Brillouin sphere containing all mass components, centered at the body's center of mass. This region defines the boundary for where spherical harmonics are applicable.

Convergence is guaranteed outside of this sphere but not within it because the mathematical assumptions assume each spherical layer is empty. Therefore, it is only guaranteed for "shells" containing no matter. Since $\overline{C}_{n,m}$ and $\overline{S}_{n,m}$ are derived with this assumption, they lose guaranteed accuracy below this threshold. There are methods available for approximating the spherical harmonics within this sphere, one of which is the Runge-Krarup theorem described in [8]. Using the gravity implied by the lunar topography, the Runge-Krarup method aims to avoid the divergence within the Brillouin sphere within the 5 arc-min resolution defined in [8]. Calculating spherical harmonic gravity assumes that the $\overline{C}_{n,m}$ coefficients do not depend on the position of the evaluation point and can be interpreted as constants defining the gravitational field of the attracting body [9]. The $\overline{C}_{n,m}$ and $\overline{S}_{n,m}$ coefficients needed for spherical harmonic calculations are derived from data collected by various missions around the body of interest. A detailed breakdown of the spherical harmonic calculations and the coefficients used is provided in Appendix A. The lunar gravity models described in the following sections use spherical harmonics to calculate the gravity around the Moon. These models would be more suitable for a satellite mission outside of the Brillouin sphere, but since this study focuses on operations on the surface, the LGM2011 surface gravity approximations are more suitable due to the potential for spherical harmonic divergence at the surface.

2.1.2 GLGM-2 Gravity Model

The GLGM-2 is a spherical harmonic model of the lunar gravity field to degree and order 70 based on S-band Doppler tracking data from the Clementine mission, as well as historical tracking data from Lunar Orbiters 1–5 and the Apollo 15 and 16 subsatellites [10]. The data derived from these satellites had large gaps in coverage on the far side of the Moon, as will be seen with some of the other lunar gravity models. The size of the initial spherical harmonic solutions were limited based on the computers available during the Apollo program in the 1960s and 1970s. The Lunar Orbiters were tracked by the Deep Space Network (DSN) and had two mission phases: primary mission for photographic mapping and an extended mission with sparser tracking. The Apollo spacecraft subsatellites were designed to study the magnetic fields, plasmas, and energetic particles around the Moon. They carried S-band transponders to provide high-precision gravity mapping of the Moon and were spin-stabilized, eliminating the need for additional thrusting during the mission. This caused the data to be much smoother than the data from the Lunar Orbiters, which required multiple maneuvers per day. The Clementine satellite also carried an S-band transponder for gravity mapping and was tracked by the DSN.

The tracking data from the sources above were divided into independent data spans, called arcs, based on spacecraft orbit characteristics, frequency of maneuvers, and availability of tracking data. The GLGM-2 final result is based on 392 arcs, from 6 hours to 12 days in length, and included 708,854 observations, with the majority of these observations coming from the Clementine satellite. The force model included third-body perturbations of the Earth, the Sun, and all the planets, the solar radiation pressure, and the Earth-induced and solar-induced lunar tides. More on the derivation of the spherical harmonic solution can be found in [10]. The solution of the GLGM-2 gravity model has free-air gravity anomalies ranging for -294 to 358 mGals as shown in Figure 2.2, where 1 Gal=0.01 m/s^2. The GLGM-2 gravity model was not sufficient for this study due to its low-order spherical harmonic solution and lack of far side data.

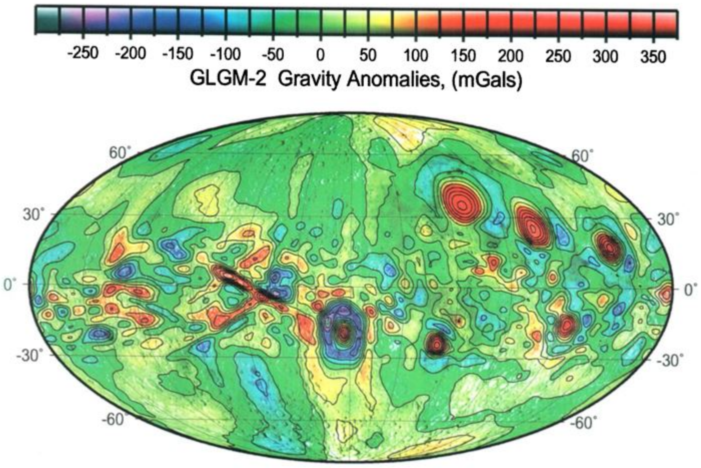


Figure 2.2: The GLGM-2 free-air gravity anomalies in mGals computed with respect to a reference radius of 1738 km. The anomalies show higher magnitudes in regions surround craters and other topographic features. (Figure reproduced from [10])

2.1.3 LP100K Gravity Model

The LP100K gravity model is based on previous lunar mission tracking data, along with the 1998–1999 Lunar Prospector (LP) mission [11]. The LP was NASA's third Discovery mission and was placed in a polar, near-circular orbit at an altitude of 100 km for its nominal mission, which lasted approximately one year. This provided global coverage for the gravity experiment, with the exception of when it passed behind the Moon. The LP then had an extended mission at a lowered altitude of 30 km, obtaining higher resolution gravity, spectrometer, and magnetic data. The LP far side gravity information comes from the long-term effect on the spacecraft's orbit, creating a gap in the data for about 33% of the surface. The LP100k gravity model is solved using spherical harmonics to the 100th degree. LP100k uses the Lunar Orbiters, Apollo spacecraft, and Clementine satellite defined above in Section 2.1.2, along with the LP to calculate the gravity field. The lunar mission observations from these sources were processed using JPL's Orbit Determination Program (ODP). The estimated parameters consist of arc-dependent variables that are determined separately for each data arc, such as spacecraft state, and global variables that are common to all data arcs, such as harmonic coefficients. More on the gravity modeling is described in [11]. The LP100K gravity model is not used in this study due to its lack of far side gravity data.

2.1.4 GRAIL Data and Resulting GRGM1200A Gravity Model

The next gravity model covered, GRGM1200A, is based on data obtained during the Gravity Recovery and Interior Laboratory (GRAIL) mission. Before going over the GRGM1200A model, it is important to understand the background of the GRAIL mission and how its data was collected. Hoffman provides an extensive GRAIL overview in [12], which is summarized here. The GRAIL mission was designed to provide a crust-to-core lunar gravity map using two tandem satellites. The GRAIL data provides better near side and much improved far side fidelity of the lunar gravity. The far side information for the models above has been limited due to them relying on the DSN, which fails to track a satellite once it passes behind the Moon. The GRAIL mission underwent an approximate 90 day science phase, where it completed three 27.3 day mapping cycles.

The Lunar Gravity Ranging System (LGRS) was the science instrument used on each GRAIL satellite. The LGRS used a dual-one-way ranging measurement to measure the relative motion between the two GRAIL satellites. Since the gravity of the Moon impacts each spacecraft's center of mass (CM), the change is reflected in the LGRS relative motion measurements. The Line of Sight (LOS) range-rate along with the DSN tracking data over the mapping cycles is then processed to develop the lunar gravity map. The resulting gravity field has a resolution of 30 km (equivalent to degree and order 180 in spherical harmonics [13]). The GRGM1200A utilizes this data to calculate a degree and order 1200 spherical harmonic gravity field model with a resolution of <5 km [13]. This is the most accurate gravity model discussed thus far but is not selected for use in this study due to the LGM2011 model having a higher resolution and being designed for surface use (inside the Brillouin sphere).

2.1.5 LGM2011 Gravity Model

The gravity field model selected for this study is the LGM2011 gravity field model outlined in [14]. The LGM2011 model provides gravity accelerations, free-air gravity anomalies, selenoid undulations, and vertical deflections of the gravity across the lunar surface with a resolution of 1.5 km. Its measurements are based on the Japanese Selenological and Engineering Explorer (SELENE) mission's SMG100i dataset, which provides important data on the composition of the Moon [15]. The gravity field signatures of medium and small-size impact craters are found using the high-frequency topographic information derived from the Lunar Orbiter Laser Altimeter (LOLA), which produces information on the Moon's surface brightness, surface slopes, surface roughness, and geodetic topography [16]. The LGM2011 model is a composite of three different input data sets that take into account the topography of the Moon:

1. SELENE data providing low and medium-frequency constituents of the gravity field,
2. Newtonian forward-modeling used to calculate a topography-implied gravity field,
3. Modeled normal gravity to account for the gravitational attraction of the Moon's total mass and decay of gravity with height.

The LGM2011 model is also based on three key assumptions:

1. Coherence of SELENE and LOLA data,
2. Uncompensated high-frequency topography, and
3. Constant mass density.

The 1.5 km scale of the LGM2011 model is assumed to be immune from biases and errors generated from laser altimetry data not completely matching the gravity data. The geometric flattening of the Moon is assumed to be zero since the Moon is more spherical than elliptical. The surface gravity accelerations, shown in Figure 2.3, are the sum of the SGM100i SELENE gravity, the Newtonian gravity, and the normal gravity shown in Figure 2.4.

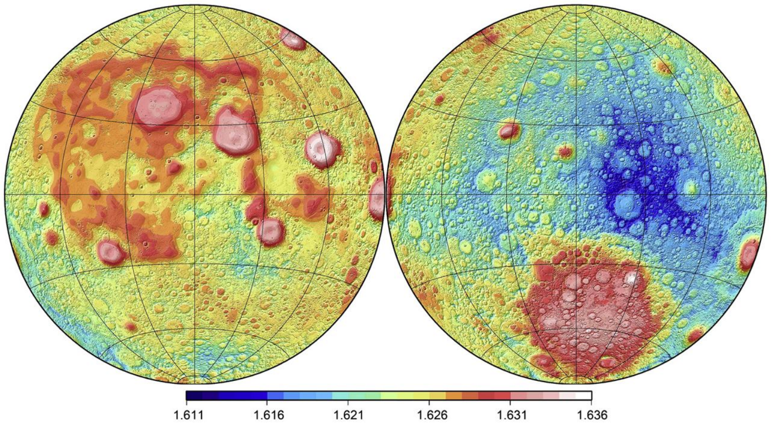
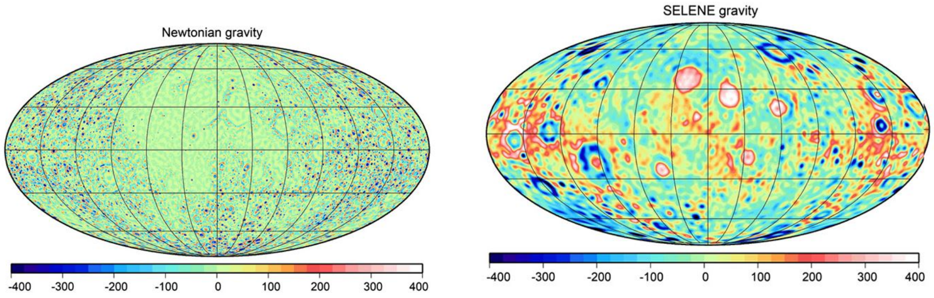
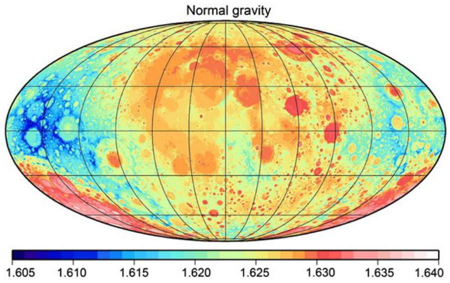


Figure 2.3: LGM2011 surface gravity accelerations in m/s^2, with the near side shown on the left and the far side on the right. The surface gravity values are higher in most craters and show specific regions where the magnitude is greater. (Figure reproduced from [14])



(a) Newtonian gravity (b) SELENE gravity anomalies



(c) Normal gravity in m/s^2

Figure 2.4: The three LGM2011 input datasets provide the values needed to calculate the deflection and surface gravity magnitudes across the lunar surface. The Newtonian forward-modeling is used to calculate the topographic-implied gravity field. The SELENE data provides the low and medium-frequency constituent components of the gravity field. Finally, the Normal gravity accounts for the Moon's total mass and gravity decay as altitude from the surface increases. (Figures reproduced from [14])

It is clear from Figure 2.3 how crater and other topographic features have significant impacts on surface gravity values, with highs generally occurring at the center of impact craters. The deflections of the gravity vector are calculated using the gradients of the selenoid in the north-south and east-west directions. These deflections are also highly affected by topographic features, as seen in Figure 2.5

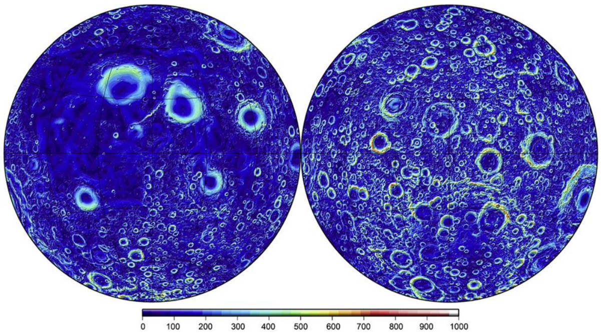


Figure 2.5: Near and far side LGM2011 magnitude of total vertical deflections at the lunar surface, provided in arcseconds ("), are shown to be heavily influenced by topographic features. (Figure reproduced from [14])

It is difficult to assess the accuracy of the LGM2011 data without multiple ground-truth measurements. Table 2.1 provides a comparison between measured gravity across four Apollo missions and LGM2011's predicted surface gravity.

Table 2.1: The comparison between measured Apollo surface gravity and LGM2011 predicted surface gravity allows Hirt and Featherstone to define the general accuracy of the LGM2011 model to be in 10s of mGal [14].

Mission	ϕ (°)	λ (°)	Measured Gravity (mGal)	LGM2011 (mGal)
Apollo 11	0.673	23.473	162,852	162,801
Apollo 12	-3.008	-23.425	162,674	162,700
Apollo 14	-3.644	-17.477	162,653	162,621
Apollo 17	20.192	30.765	162,695	162,685

Based on Table 2.1, it can be approximated that the LGM2011 gravity model provides surface gravity measurements accurate to 10s of mGal. The accuracy of the vertical deflections is assumed to be accurate to as few as 10 arcseconds, but like the surface gravity, it can't be proven without true ground measurements. The LGM2011 gravity model was selected over the ones discussed in Sections 2.1.2-2.1.4 because it provides accurate values for both the surface gravity magnitude and north-south and east-west deflection of the vertical across the entire lunar surface. The main purpose of this study focuses on the fact that gravity measured at any point on the Moon is not perfectly normal to the surface due to surrounding topographic features. Having a data set that covers the entire surface describing these behaviors is the backbone of the MEKF algorithm outlined in Section 3.7. All LGM2011 data can be downloaded at [17].

2.2 Other Lunar Positioning Methods

To understand the benefits of the proposed CelNav MEKF method, it is important to analyze already-available lunar positioning methods and algorithms. There are several different combinations of common navigation techniques, such as ordinary CelNav and the use of inertial navigation systems (INS). These methods and their performance are outlined in the following sections.

2.2.1 Other Celestial Navigation (CelNav)

Thein, Quinn, and Folta define a CelNav method on the lunar surface using an accelerometer and star tracker in [18], similar to the setup of this study. Their study focuses on navigation within craters on the lunar surface where limited sensor measurements are available. The goals of this CelNav method included:

1. Developing an algorithm to determine location based only on star tracker and ac-

celerometer measurements;

2. Determining navigational error based on ST, accelerometer, and alignment errors; and
3. Finding the error tolerances that ensure the position estimate is calculated with an accuracy of within 50 meters.

The ST and accelerometer outputs are assumed to be a quaternion and directional unit vector respectively. The errors in their study come from noise in the measurements taken and misalignment errors between the body frames of the individual components. They then define the accelerometer and ST measurement models as functions of body frame alignment with respect to the local East-North-Up (ENU) coordinate system, as well as errors in the devices' alignment and measurements. After defining the measurement models, they outline a process to calculate the local longitude and latitude (λ, ϕ) position. The simulated measurement noise for the ST is based on its accuracy within a certain boresight cone angle, defined as 0.001° in their study. The simulated accelerometer measurement noise is determined via a random number generator following a normal distribution with fixed magnitude based on the accuracy of the accelerometer. The navigational accuracy, including limited sensor noise, was found to be within the defined 50 m threshold. However, the accuracy significantly decreases as more sensor noise is introduced. This CelNav method cannot handle significant sensor noise and does not account for potential bias in the accelerometer. Additionally, it does not use an external gravity field reference to aid in position estimation, which limits the accuracy of the final position value. It also assumes a point-mass spherical gravity field in its accelerometer measurements instead of accounting for the mass contributions throughout the entire body.

2.2.2 INS/CelNav Combined Navigation Method Using EKF

Ning and Liu outline a two-mode INS/CelNav navigation method based on the kinematic state of a rover system, as described in [19]. Since this study is primarily focused on finding the CelNav solution in a stationary state, the tightly-coupled mode described by Ning and Liu is summarized here. In the tightly-coupled mode, the true velocity of the system is zero. Therefore, any measured INS velocity is defined as velocity error. Figure 2.6 shows the INS/CelNav test procedure used.

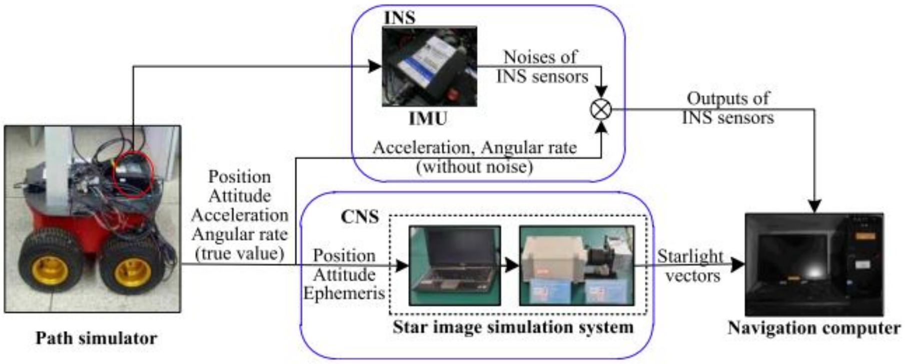


Figure 2.6: Ning and Liu's tightly-coupled INS/CelNav ground test system uses a path simulator, an IMU, and a star image simulation system to produce position outputs. (Figure reproduced from [19])

The state vector consists of the misalignment angle, velocity error (derived from stationary INS measurements), position error, gyroscope drift error, and the accelerometer bias vector. The measurement model consists of horizontal velocity errors, starlight vectors, and star altitude, which is a function of the rover's position and the star's geographical position. The process for determining the star altitude is discussed further in their study, completing the measurement model. The state definition and measurement model can then be used to define an EKF algorithm to solve for the state as velocity and ST measurements are received. An overview on EKFs as well as some of its variants is discussed in [20]. It is important to note that the accelerometer bias vector included in the state for this method is not the same as the accelerometer bias defined in Section 3.8. It refers to the known accelerometer bias that can be used to account for the true measured accelerometer values. This method does not utilize local gravity measurements or account for the dynamic bias that occurs while on the lunar surface, leading to less accurate position measurements.

2.2.3 INS/CelNav Combined Navigation Method Using UKF

An INS initialization method aided by celestial observations is described by Ning et al. in [21]. Initializing the INS and CelNav systems independently is challenging, but combining them simplifies the process. Traditional INS initialization methods are typically based on Kalman filters, with the INS error model as the state and the velocity error as the measurement model. These Kalman filters, like the one defined in Section 2.2.2, offer low observability. The method provided in [21] uses an Unscented Kalman Filter (UKF) to fuse together the information received from the ST and INS. More information on UKFs can be found in [22].

The UKF state model in [21] consists of the system's (λ, ϕ) position, Euler attitude angles, and gyroscope and accelerometer errors. The measurement model contains the star altitude, the starlight vector, the measured accelerometer gravity, and the gyroscopic outputs. Ning et al. defend their use of the UKF over a traditional EKF due to its lower estimation errors for nonlinear systems and not having to find the Jacobian matrices. They find an estimation error of the initial position within 300 m, and the estimation error of the initial attitude within 40". The estimation procedure described in this section provides a good method for the initialization of the system, but it is not accurate enough over long durations to provide sufficient position values.

2.2.4 Lunar Positioning Using Sun-Based CelNav

There are multiple sources of information that can be used for celestial positioning. Peng, Li, and Jilin described a positioning and orientation method for lunar rovers using an inclinometer's pitch and roll outputs and measured sun direction vector instead of the previously mentioned starlight vectors [23]. Instead of a typical ST, this method utilizes a sun sensor with three parts: the optical lens, photoelectric sensor, and the signal processor. The resulting sun direction vector is fed into the QUEST algorithm defined in Appendix A of [23] to determine the resulting attitude quaternion. Peng, Li, and Jilin use this information to define a simultaneous positioning and orientation (SCPO) method, which translates the nonlinear celestial positioning problem into a linear quaternion attitude determination problem. This means that the position and orientation are solved together instead of individually. The quaternion resulting from the measured sun direction vector and the inclinometer roll and pitch are then used to determine the final position and heading, as seen in Figure 2.7

The algorithm used in Figure 2.7 is influenced by two types of errors: system error and random error. System error arises from misalignment in the required rotation matrices, while random error results from zero-mean Gaussian white noise in the simulated measurements. They do not use an external gravity reference to aid in their calculations. The experimental and simulation conditions are discussed in detail in [23] and will not be covered here. The average navigation errors in heading and position are 0.1123° and 8.546 km, respectively. These errors are heavily dependent on noise in the measurements, as is typical for any measurement model.

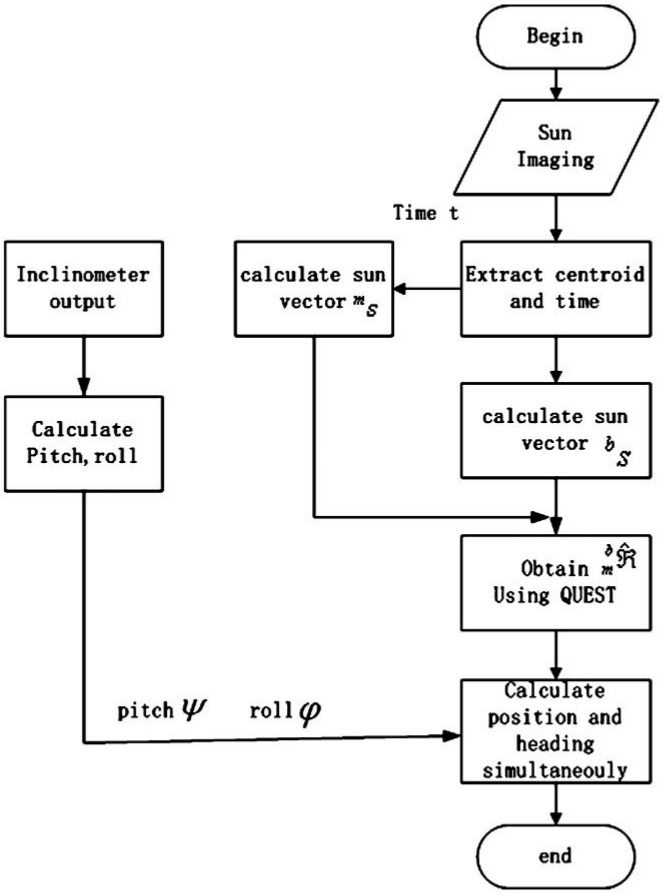

Begin
Sun Imaging
Time t
Inclinometer output
calculate sun vector m_S
Extract centroid and time
Calculate Pitch, roll
calculate sun vector b_S
Obtain
Using QUEST
pitch ψ
roll φ
Calculate position and heading simultaneouly
end

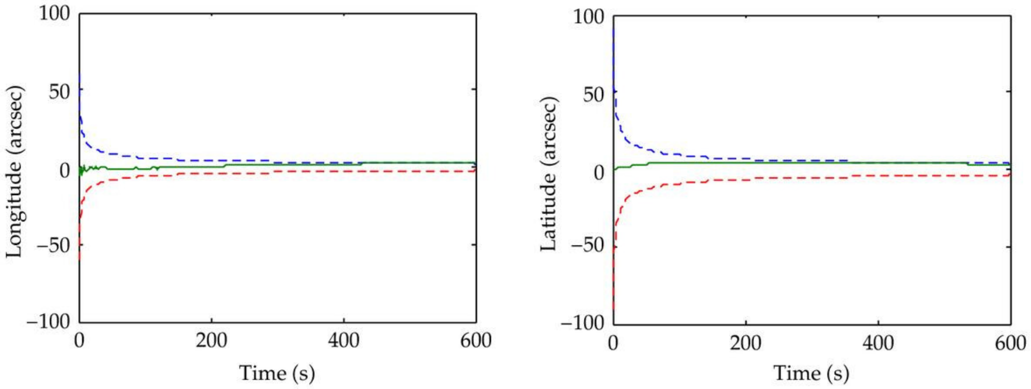
Figure 2.7: The ideal SCPO algorithm uses an inclinometer and quaternion output calculated using sun vectors instead of starlight vectors to estimate position and orientation. Sun vectors are an available source of information on the lunar surface, providing another attitude determination method. (Figure reproduced from [23])

2.2.5 Dual-EKF-Based CelNav

Xie, Yang, and Li provide a dual-EKF-based real-time CelNav (RCN) method in [24]. Generally, lunar navigation methods use either absolute or relative positioning. Absolute positioning methods result in position and heading errors that are bounded and do not accumulate over time. Relative positioning, such as the INS described in Sections 2.2.2–2.2.3, provides high accuracy over short periods but leads to larger errors that accumulate over time. Combining these absolute and relative positioning procedures utilizes the advantages in both. The algorithm used to calculate the satellite reference position in Section 3.9 similarly employs two methods to estimate the state, but it uses a batch-to-EKF filter instead of combining two EKFs.

The dual-EKF-based method derives a continuous-discrete nonlinear state space model using a state vector of the navigation system $\underline{\boldsymbol{x}}_s$, a system parameter vector $\underline{\boldsymbol{x}}_{\boldsymbol{p}}$, and observation vector $\underline{\tilde{\boldsymbol{y}}}_k$. The state vector contains the heading and attitude quaternion in the navigation frame, along with the constant gryoscope bias. The parameter vector contains the (λ, ϕ) position and the north and east speed on the lunar surface. The system then receives celestial and speed observations with included measurement noise.

The dual-EKF algorithm uses two EKFs working in parallel and a state estimator working between the system parameter time update and the measurement time update. The process equation about the system parameter vector is a linear equation, while the process equations for the state vector are nonlinear. Therefore, the Jacobian with respect to the state vector does not need to be found. Xie, Yang, and Li provide a thorough derivation on the linearization of the state process measurement equations. They also include a full algorithm that clearly defines the initialization, state measurement update, parameter measurement update, parameter time update, and state time update steps. A lunar rover is then simulated to undergo five different motions: static, rotation, uniform motion, rotation and uniform motion, and static again. Since the study of this thesis primarily deals with measuring the position in a static state, the first and fifth portions of the movement meet the conditions outlined for the proposed MEKF algorithm. The other segments of the lunar rover motion could be used as a basis for movement between positions as seen in Section 3.10, which could be analyzed more in future studies. The explicit error values for the rover motion are not clearly defined, but plots of the λ and ϕ position are shown in Figures 2.8a and 2.8b respectively.



(a) λ position error and 3σ boundary. (b) ϕ position error and 3σ boundary.

Figure 2.8: The Dual-EKF-based CelNav longitude position errors shown on the left demonstrate accurate results staying reasonably within its 3σ uncertainty boundary. The latitude position errors on the right exhibit similar behavior with a wider 3σ uncertainty boundary. (Figure reproduced from [24])

The dual-EKF-based CelNav method outlined in this section uses sun orientation and speed observations to estimate the system state and parameter vectors. This method differs from ours (outlined in Section 3.7) in that it does not use accelerometers in any way to aid in determining the lunar position. Even though the addition of accelerometers introduces more measurement noise and potential bias, this study outlines procedures to solve for the bias and provide highly accurate positioning with gravity measurements including Gaussian noise.

Chapter 3

Methodology

This chapter outlines the methods used to calculate lunar surface position and orientation, as well as dynamic accelerometer bias, using gravity and ST measurements. Additionally, the estimation of correlated gravity biases resulting from errors in the reference lunar gravity model is presented using MEKF CelNav and satellite-derived reference positions.

3.1 General Idea Behind The Study

Figure 3.1 illustrates the information used by the conventional CelNav algorithm to estimate position and attitude on the lunar surface. Given starlight and gravity vector measurements, the relative position on the surface can be determined. It is important to note from Figure 3.1 that the gravity vector assumed in typical CelNav applications follows the point mass gravity assumption, meaning the gravity vector is directed toward the center of the Moon with no deflections. The starlight vectors contribute to the algorithm by providing a coarse estimate of the surface location. The gravity vector, under the point mass assumption, is then used to compute the position vector to the surface location by negating its direction and assuming that the position vector lies along that path.

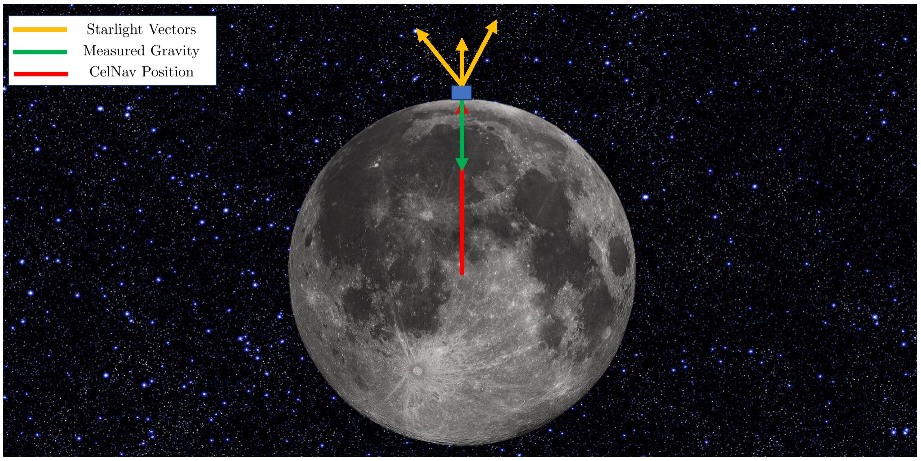


Figure 3.1: This diagram shows the CelNav concept, demonstrating that starlight and gravity vector measurements are able to provide a position and attitude estimation of where you are on the lunar surface.

However, as demonstrated by the LGM2011 gravity model, gravity vectors across the lunar surface exhibit deflections, which can lead to inaccuracies in position estimates when using pure CelNav techniques. This highlights the need for an algorithm capable of accounting for these deflections by referencing an external high-fidelity gravity field model to produce more accurate position and attitude estimates. The general process incorporating all available information is shown in Figure 3.2. This figure illustrates how the combination of starlight vectors, an external gravity field that includes deflections, and a measured local gravity vector can be used to calculate improved position estimates, in contrast to those obtained using the point mass gravity assumption from Figure 3.1. The errors resulting from neglecting deflections in the local gravity vectors are shown in Figure 3.3, which highlights significant errors when deflections are not considered. It is also noteworthy that these errors tend to be more concentrated in regions near topographic features such as craters—a trend that will appear throughout this study. These results validate the necessity of the methods proposed in this work, which incorporate gravity vector deflections through the use of an external gravity field reference.

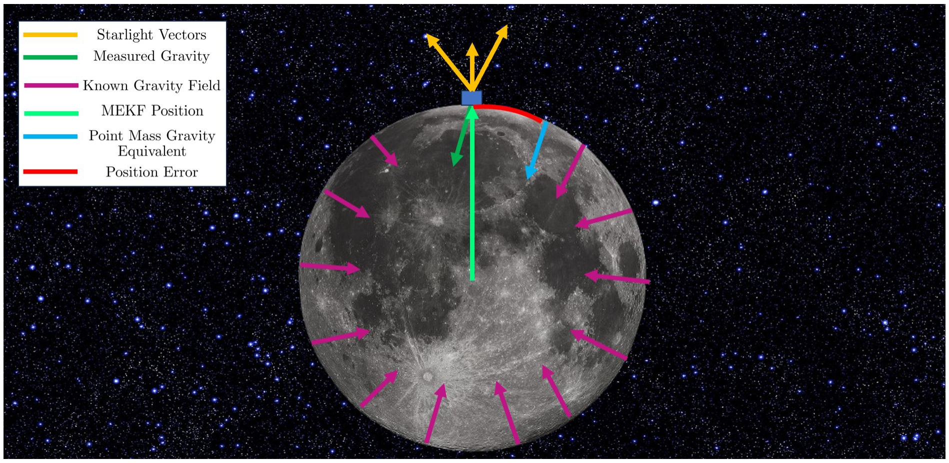


Figure 3.2: This diagram shows the MEKF CelNav concept used in this study. Unlike the typical CelNav convention, the MEKF CelNav algorithm accounts for potential gravity deflections and uses them in reference to an external gravity field model to provide more accurate position and attitude estimation. The error that would result by not accounting for the gravity deflections is also shown if only a point mass gravity assumption was used.

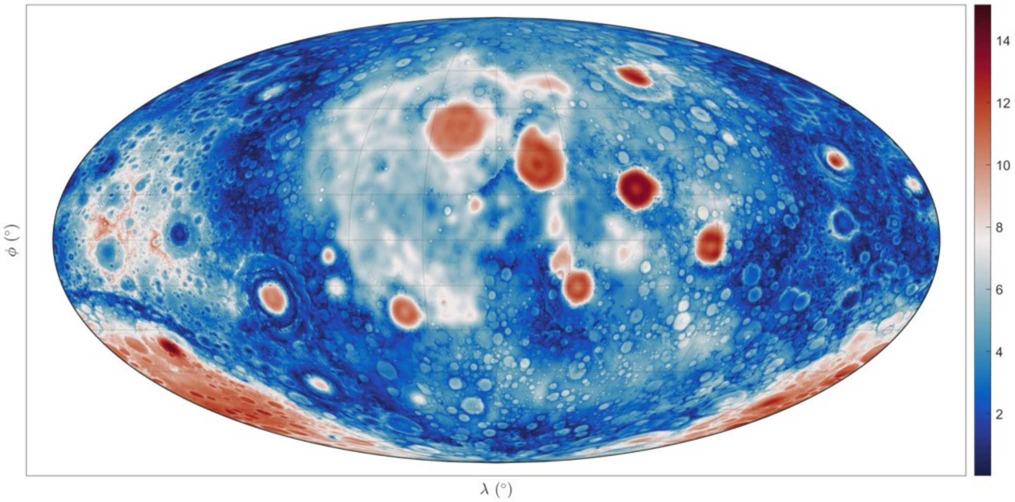


Figure 3.3: Position errors resulting from not accounting for local gravity deflections in km. It is clear that the errors are concentrated in regions near topographic features across the surface.

3.2 Workflow Overview

A diagram illustrating the flow of information and how it is passed between algorithms is shown in Figure 3.4. The individual components of this process will be detailed throughout this chapter, but the overview diagram is provided here to orient the reader. The dark blue parallelograms represent algorithm input values or data sources. For example, the starlight vectors serve as input for the QUEST algorithm. The red boxes represent the algorithms used to calculate the necessary or desired output values for this study. Finally, the orange ovals represent algorithm output values. This study has four primary algorithms:

1. QUEST
2. Satellite Reference Position
3. CelNav
4. Correlated Gravity

The QUEST algorithm processes star tracker images to provide the reference quaternion required for the CelNav algorithm, which produces a biased position estimate and a dynamic accelerometer bias. The satellite reference algorithm supplies a highly accurate initial position for use in the correlated gravity algorithm. Finally, the correlated gravity algorithm uses the calculated CelNav position as a measurement, along with the satellite reference position, to generate a position estimate and determine the bias in the LGM2011 gravity field for a predefined traversing mission. Figure 3.4 provides a clear overview of the information presented in this chapter and helps the reader follow the discussion in the subsequent sections.

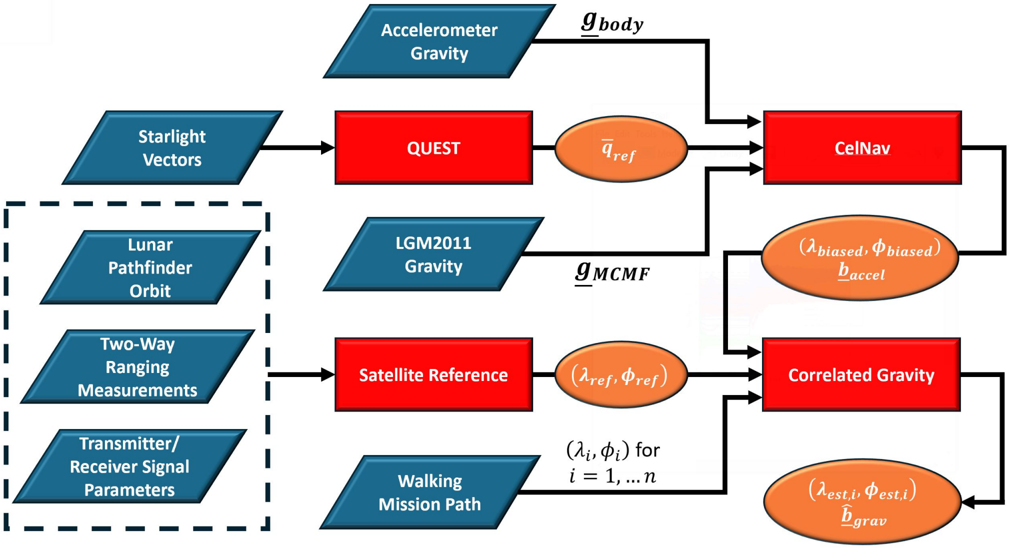

Accelerometer Gravity
$\underline{g}_{body}$
Starlight Vectors
QUEST
$\bar{q}_{ref}$
CelNav
Lunar Pathfinder Orbit
Two-Way Ranging Measurements
Transmitter/ Receiver Signal Parameters
LGM2011 Gravity
$\underline{g}_{MCMF}$
$(\lambda_{biased}, \phi_{biased})$
$\underline{b}_{accel}$
Satellite Reference
$(\lambda_{ref}, \phi_{ref})$
Correlated Gravity
(λ_i, ϕ_i) for $i = 1, \ldots n$
Walking Mission Path
$(\lambda_{est,i}, \phi_{est,i})$
$\hat{\underline{b}}_{grav}$

Figure 3.4: This diagram shows the framework of this thesis and how information is passed between different algorithms to calculate position and bias estimates.

3.3 Reading LGM2011 Gravity Data

The LGM2011 gravity model provides values for the magnitude of the surface gravity along with the north-south and east-west deflection of the vertical. The surface gravity magnitude and deflections are provided in mGal and arcseconds respectively, which can be converted to $\mathrm{m/s^2}$ and degrees using $1^\circ = 3600''$ and $1\,\mathrm{mGal} = 1 \times 10^{-5}\ \mathrm{m/s^2}$. After the values have been converted, the local gravity in the body frame at each (λ, ϕ) position can be calculated using

$$\underline{\boldsymbol{g}}_{\text{body}} = g_{\text{loc}} \begin{bmatrix} \sin(\text{EW}) \\ \sin(\text{NS}) \\ -\sqrt{1 - \sin^2(\text{EW}) - \sin^2(\text{NS})} \end{bmatrix} \tag{3.1}$$

where NS is the north-south deflection, EW is the east-west deflection, and g_{loc} is the magnitude of the local gravity. The NS deflection is measured positive in the north direction, and the EW deflection is measured positive in the east direction. For future filtering needs, the gravity vectors are rotated into the MCMF frame using

$$\underline{\boldsymbol{g}}_{\text{MCMF}} = R_{\text{ENU}}^{\text{MCMF}} \underline{\boldsymbol{g}}_{\text{body}} \tag{3.2}$$

where the ENU reference frame is assumed to be aligned with the body frame for all LGM2011 measurements. These MCMF gravity vectors for the LGM2011 gravity model are stored at each (λ, ϕ) position provided in $0.05°$ increments, matching the 1.5 km resolution described in [14].

3.4 Quaternion Convention

Quaternions provide an efficient method for analyzing the attitude of a spacecraft or surface device. The convention used throughout this study follows the Hamiltonian convention outlined in [25]. The quaternions used in this study result from the QUEST algorithm outlined in Section 3.6, representing the rotation from the MCI to ST frame (assumed to be aligned with overall system body frame), providing attitude information based on star vectors observed on the lunar surface.

The elements of a quaternion are derived from an Euler axis and angle following the theorem stating that a series of rotations of a rigid body about its center of mass is equivalent to a single rotation about a particular axis [26]. If the Euler axis is denoted $\hat{\underline{\boldsymbol{e}}} = [e_1 \ e_2 \ e_3]^T$ and the rotation angle is defined as θ, the associated quaternion is as follows

$$\bar{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} e_1 \sin \frac{\theta}{2} \\ e_2 \sin \frac{\theta}{2} \\ e_3 \sin \frac{\theta}{2} \\ \cos \frac{\theta}{2} \end{bmatrix} \tag{3.3}$$

where all 4×1 quaternions used in this study are denoted with an over bar. The standard orthonormal basis for $\mathbb{R}^3$ is provided using three unit vectors: $\underline{\hat{\boldsymbol{i}}} = [1\ 0\ 0]^T$, $\underline{\hat{\boldsymbol{j}}} = [0\ 1\ 0]^T$, $\underline{\hat{\boldsymbol{k}}} = [0\ 0\ 1]^T$. The quaternion itself can then be defined as a sum of a scalar value q_4 and vector $\underline{\boldsymbol{q}} = [q_1\ q_2\ q_3]^T$

$$\bar{q} = q_4 + \underline{\boldsymbol{q}} = q_4 + q_1\ \hat{\boldsymbol{i}} + q_2\ \hat{\boldsymbol{j}} + q_3\ \hat{\boldsymbol{k}} \tag{3.4}$$

defining the following 4×1 column vector

$$\bar{q} = \begin{bmatrix} \underline{\boldsymbol{q}} \\ q_4 \end{bmatrix} \tag{3.5}$$

The first three elements of $\bar{q}$ are considered to be the vector portion of the quaternion. The vector portion coincides with the imaginary components and a rotation in $\mathbb{R}^3$. The scalar portion is the real component and is a representation of the rotation angle as seen with Eq. 3.3. The definition of the quaternion shown in Eq. 3.5 is assumed to be of unit length, with the scalar component greater than or equal zero, matching typical conventions. Additionally, it is important to note that the quaternions $\bar{q}$ and $-\bar{q}$ represent the same rotation [27]. The addition of two quaternions is component-wise, behaving the same as any two $n \times 1$ vectors.

The sum of $\bar{q}$ with a second quaternion $\bar{p}$ is given by

$$\bar{p} + \bar{q} = (p_4 + q_4) + (p_1 + q_1)\ \hat{\boldsymbol{i}} + (p_2 + q_2)\ \hat{\boldsymbol{j}} + (p_3 + q_3)\ \hat{\boldsymbol{k}} = \begin{bmatrix} \underline{\boldsymbol{p}} + \underline{\boldsymbol{q}} \\ (p_4 + q_4) \end{bmatrix} \tag{3.6}$$

The product of two quaternions must satisfy Hamiltonian's rules shown below [28]

$$\boldsymbol{i} \otimes \boldsymbol{i} = \boldsymbol{j} \otimes \boldsymbol{j} = \boldsymbol{k} \otimes \boldsymbol{k} = \boldsymbol{i} \otimes \boldsymbol{j} \otimes \boldsymbol{k} = -1 \tag{3.7}$$

$$\boldsymbol{i} \otimes \boldsymbol{j} = \boldsymbol{k} = -\boldsymbol{j} \otimes \boldsymbol{i} \tag{3.8}$$

$$\boldsymbol{k} \otimes \boldsymbol{i} = \boldsymbol{j} = -\boldsymbol{i} \otimes \boldsymbol{k} \tag{3.9}$$

All quaternion multiplications in this study will be denoted using $\otimes$. Section 3.10 introduces the Kronecker product, which also uses $\otimes$, but the difference will be clearly noted. Therefore, the multiplication of two quaternions satisfying the Hamiltonian rules becomes

$$\begin{aligned}\bar{p} \otimes \bar{q} &= (p_4 + p_1 \hat{\boldsymbol{i}} + p_2 \hat{\boldsymbol{j}} + p_3 \hat{\boldsymbol{k}}) \otimes (q_4 + q_1 \hat{\boldsymbol{i}} + q_2 \hat{\boldsymbol{j}} + q_3 \hat{\boldsymbol{k}}) \\ &\equiv p_4 q_4 - (p_1 q_1 + p_2 q_2 + p_3 q_3) + p_4 (q_1 \hat{\boldsymbol{i}} + q_2 \hat{\boldsymbol{j}} + q_3 \hat{\boldsymbol{k}}) + q_4 (p_1 \hat{\boldsymbol{i}} + p_2 \hat{\boldsymbol{j}} + p_3 \hat{\boldsymbol{k}}) \\ &+ (p_2 q_3 - p_3 q_2) \hat{\boldsymbol{i}} + (p_3 q_1 - p_1 q_3) \hat{\boldsymbol{j}} + (p_1 q_2 - p_2 q_1) \hat{\boldsymbol{k}} \end{aligned} \tag{3.10}$$

Eq. 3.10 can be simplified using the definitions for the inner and cross product of two vectors in $\mathbb{R}^3$ space to

$$\bar{p} \otimes \bar{q} = p_4 q_4 - \underline{\boldsymbol{p}} \cdot \underline{\boldsymbol{q}} + p_4 \underline{\boldsymbol{q}} + q_4 \underline{\boldsymbol{p}} + \underline{\boldsymbol{p}} \times \underline{\boldsymbol{q}} = \begin{bmatrix} p_4 q_4 - \underline{\boldsymbol{p}} \cdot \underline{\boldsymbol{q}} \\ p_4 \underline{\boldsymbol{q}} + q_4 \underline{\boldsymbol{p}} + \underline{\boldsymbol{p}} \times \underline{\boldsymbol{q}} \end{bmatrix} \tag{3.11}$$

The conjugate and inverse of a quaternion using the same convention defined in Eq. 3.5 is

$$\bar{q}^c = \begin{bmatrix} -\underline{\boldsymbol{q}} \\ q_4 \end{bmatrix} \tag{3.12}$$

$$\bar{q}^{-1} = \frac{\bar{q}^c}{|\bar{q}|^2} \tag{3.13}$$

The inverse and conjugate are the same for unit quaternions. Each quaternion also has an associated rotation matrix describing the orientation from one reference frame to another. This rotation matrix is defined in Eq. 3.14:

$$A(\bar{q}) = (q_4^2 - \underline{\boldsymbol{q}} \cdot \underline{\boldsymbol{q}})\, I_{3\times3} + 2\, \underline{\boldsymbol{q}} \otimes \underline{\boldsymbol{q}}^T + 2q_4 \left[\left[\underline{\boldsymbol{q}}\right]\right] \tag{3.14}$$

$$\left[\left[\underline{\boldsymbol{q}}\right]\right] = \begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix} \tag{3.15}$$

where Eq. 3.15 defines the cross-product matrix of any arbitrary 3 × 1 vector.

It is also possible to take an arbitrary rotation matrix and find the quaternion representing the same rotation. Bar-Itzhack provides the following method to go from a rotation matrix to a quaternion representing the same attitude rotation [29]. Given an orthogonal 3 × 3 matrix, D, form the following K matrix using matrix D's components d_{ij}:

$$K = \frac{1}{2} \begin{bmatrix} d_{11} - d_{22} & d_{21} + d_{12} & d_{31} & -d_{32} \\ d_{21} + d_{12} & d_{22} - d_{11} & d_{32} & d_{31} \\ d_{31} & d_{32} & -d_{11} - d_{22} & d_{12} - d_{21} \\ -d_{32} & d_{31} & d_{12} - d_{21} & d_{11} + d_{22} \end{bmatrix} \tag{3.16}$$

Compute the eigenvalues and eigenvectors of K, and define the eigenvector corresponding to an eigenvalue of one to be the quaternion for the given rotation matrix. This quaternion represents the same properties as the original rotation matrix in vector form and can be used in attitude estimation algorithms.

3.5 Bright Star Catalog and Simulated Star Vectors

Simulated starlight vectors are needed as inputs for the QUEST algorithm outlined in Section 3.6. These will be the vectors that a ST measures whil on the lunar surface. The accuracy of these vectors is very important in ensuring the legitimacy of the quaternion attitude produced by the QUEST algorithm. Therefore, Kitt Peak National Observatory's Bright Star Catalog was used to provide information about 9110 star objects [30]. The positions of these objects were calculated from equinox J2000, epoch 2000.0. The stars identified in the catalog have information about equatorial positions for B1900.0 and J2000.0, galactic coordinates, proper motions (J2000.0), parallax, radial and rotational velocity, spectral types, and Right Ascension (α) and Declination (δ) in hours-minutes-seconds and degrees-minutes-seconds, respectively [31]. The only information needed from the catalog for this study is each star's α and δ with respect to the Earth-Centered Inertial (ECI) frame; proper motion and parallax are ignored. After converting the α and δ values into degrees (or radians), the unit vector to each star in the ECI frame is calculated using

$$\underline{\hat{\boldsymbol{s}}}_{\mathrm{ECI}} = \begin{bmatrix} \cos(\delta)\cos(\alpha) \\ \cos(\delta)\sin(\alpha) \\ \sin(\delta) \end{bmatrix} \tag{3.17}$$

The unit vector to each star can then be rotated to the MCI frame using Eq. 3.18, based on the rotation in Eq. 3.18 outlined by Ahrens [32]:

$$R_{\text{MCI}}^{\text{ECI}} = R_x(-0.3^{"})R_y(-78.56^{"})R_z(-67.92^{"}) \tag{3.18}$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{3.19}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{3.20}$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.21}$$

where the units used in Eq. 3.18 are in arcseconds. The unit vectors for each star in the MCI frame represent $\underline{\hat{\boldsymbol{V}}}_i$ in the QUEST algorithm (see Section 3.6). The stars observed by the ST are simulated in this study based on the dot product between the starlight vectors in the MCMF frame and the local normal unit vector. The MCI starlight vectors are rotated into the MCMF frame using

$$\underline{\hat{\boldsymbol{s}}}_{\text{MCMF}} = R_{\text{MCI}}^{\text{MCMF}} R_{\text{ECI}}^{\text{MCI}} \underline{\hat{\boldsymbol{s}}}_{\text{ECI}} \tag{3.22}$$

The local normal unit vector is defined using the local (λ, ϕ) position

$$\underline{\hat{\boldsymbol{u}}}_{\text{MCMF}} = \begin{bmatrix} \cos(\phi)\cos(\lambda) \\ \cos(\phi)\sin(\lambda) \\ \sin(\phi) \end{bmatrix} \tag{3.23}$$

The stars seen by the ST depend on the field of view (FOV) of the sensor. The viewable stars in the ST FOV are found by checking

$$\underline{\hat{\boldsymbol{s}}}_{\text{MCMF}} \cdot \underline{\hat{\boldsymbol{u}}}_{\text{MCMF}} \geq \cos(\text{FOV}) \tag{3.24}$$

Each star meeting the above criteria is "observed" by the ST and supplied to the QUEST algorithm to determine the reference quaternion.

3.6 QUEST Algorithm

The attitude determination method used in this study involves calculating an attitude quaternion representing rotation from the MCI to body frame using ST vector measurements. Shuster provides an algorithm for executing the QUEST algorithm based on Wahba's problem [6]. The QUEST algorithm is a least-square problem first proposed by Grace Wahba in 1965. The problem focuses on finding the attitude which minimizes the following cost function:

$$L(A) = \frac{1}{2}\sum_{i=1}^{n} a_i \|\underline{\hat{\boldsymbol{W}}}_i - A\,\underline{\hat{\boldsymbol{V}}}_i\|^2 \tag{3.25}$$

where $\underline{\hat{\boldsymbol{W}}}_i,\ i = 1, ..., n$, are a set of unit-vector observations (with error) in the spacecraft-fixed reference frame (the body frame in this study), and $\underline{\hat{\boldsymbol{V}}}_i,\ i = 1, ..., n$, are the same unit vectors represented in the primary reference frame (MCI in this study, see $\underline{\hat{\boldsymbol{s}}}_{\text{MCI}}$ from Section

3.5). The a_i values are a set of non-negative weights discussed further in this section. Based on the problem in Eq. 3.25, a unique minimizing attitude matrix, A, will always exist given that at least two of the vector observations are not parallel. The first step in solving for the optimal attitude is to rewrite Eq. 3.25 using the gain function $g(A)$

$$L(A) = \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} \left(a_i \, \underline{\hat{\boldsymbol{W}}}_i \cdot A \, \underline{\hat{\boldsymbol{V}}}_i \right) = \sum_{i=1}^{n} \left(a_i - g(A) \right) \tag{3.26}$$

$$g(A) = \mathrm{tr}(B^T A) \tag{3.27}$$

The B matrix in the gain function shown in Eq. 3.27 is the attitude profile matrix given by

$$B = \sum_{i=1}^{n} a_i \, \underline{\hat{\boldsymbol{W}}}_i \, \underline{\hat{\boldsymbol{V}}}_i^T \tag{3.28}$$

Even though $g(A)$ is linear, minimizing $g(A)$ is not so simple since the attitude matrix is subject to six nonlinear constraints. Therefore, the attitude matrix is rewritten as a quadratic function of the quaternion, matching the convention shown in Eq. 3.5. The attitude matrix is then defined using Eqs. 3.14 and 3.15 as $A(\bar{q})$. The QUEST algorithm also needs the following matrix and scalar values:

$$S = B + B^T \tag{3.29}$$

$$s = \mathrm{tr}(B) \tag{3.30}$$

$$[[\underline{\boldsymbol{Z}}]] = B - B^T \tag{3.31}$$

The gain function can be rewritten using the new quaternion definition and matrix K from Eq. 3.33 as

$$g(\bar{q}) \equiv g(A(\bar{q})) = \bar{q}^T K \bar{q} \tag{3.32}$$

$$K = \begin{bmatrix} S - sI_{3\times3} & \underline{\boldsymbol{Z}} \\ \underline{\boldsymbol{Z}}^T & s \end{bmatrix} \tag{3.33}$$

Maximizing $g(A)$ based on Eq. 3.32 leads to the following eigenvalue equation for the optimal quaternion, $\bar{q}^*$:

$$K\bar{q}^* = \lambda_{\max}\bar{q}^* \tag{3.34}$$

where $\lambda_{\max}$ is the largest eigenvalue of the K matrix in Eq. 3.33. Finding the eigenvector associated with the largest eigenvalue of the 4×4 eigenvalue problem in Eq. 3.34 is referred to as Davenport's q-method. QUEST provides a fast implementation of the q-method and is formulated in terms of the Gibbs vector $\underline{\boldsymbol{Y}}$, defined as

$$\underline{\boldsymbol{Y}} \equiv \frac{\boldsymbol{q}}{q_4} \tag{3.35}$$

The optimal attitude written in terms of the Gibbs vector is

$$\underline{\boldsymbol{Y}}^* = [(\lambda_{\max} + s)I_{3\times3} - S]^{-1}\underline{\boldsymbol{Z}} \tag{3.36}$$

The optimal quaternion can then be found based on Eq. 3.36 using

$$\bar{q}^* = \frac{1}{\sqrt{1 + \|\underline{\boldsymbol{Y}}^*\|^2}} \begin{bmatrix} \underline{\boldsymbol{Y}}^* \\ 1 \end{bmatrix} \tag{3.37}$$

The optimal quaternion solved for in Eq. 3.37 is only as accurate as the ST vector measurements. The star vectors are assumed to have Gaussian random errors of the form

$$\underline{\hat{\boldsymbol{W}}}_i = A\underline{\hat{\boldsymbol{V}}}_i + \Delta\underline{\hat{\boldsymbol{W}}}_i \tag{3.38}$$

where the Gaussian nature of the error in the measured $\underline{\hat{\boldsymbol{W}}}_i$ vectors satisfies the following statistical properties:

$$\mathbb{E}\left\{\Delta\underline{\hat{\boldsymbol{W}}}_i\right\} = 0 \tag{3.39}$$

$$\mathbb{E}\left\{\Delta\underline{\hat{\boldsymbol{W}}}_i\Delta\underline{\hat{\boldsymbol{W}}}_i^T\right\} = \sigma_i^2[I_{3\times3} - (A\underline{\hat{\boldsymbol{V}}}_i)(A\underline{\hat{\boldsymbol{V}}}_i)^T] \tag{3.40}$$

The weights $a_i,\ i = 1, ..., n$ seen in Eq. 3.25 are a function of the σ_i standard deviation of sensor error values used in Eq. 3.40.

$$a_i = \frac{c}{\sigma_i^2} \tag{3.41}$$

where c is some arbitrary constant value. The total attitude covariance matrix representing the accuracy in QUEST's output optimal quaternion $\bar{q}^*$ is defined as

$$P_{\theta\theta} = \left[\sum_{i=1}^{n}\frac{1}{\sigma_i^2}(I_{3\times3} - (\underline{\hat{\boldsymbol{W}}}_i)_{\text{true}}(\underline{\hat{\boldsymbol{W}}}_i)_{\text{true}}^T)\right]^{-1} \tag{3.42}$$

$$(\underline{\hat{\boldsymbol{W}}}_i)_{\text{true}} \equiv A_{\text{true}}\underline{\hat{\boldsymbol{V}}}_i \tag{3.43}$$

where $\underline{\hat{\boldsymbol{W}}}_i$ typically replaces $(\underline{\hat{\boldsymbol{W}}}_i)_{\text{true}}$ in Eq. 3.42 since the true values are not typically known. $P_{\theta\theta}$ and $\bar{q}^*$ are very important in this study, helping to define initial covariance guesses and the accelerometer bias in the MEKF algorithm outlined in Section 3.7.

3.7 CelNav MEKF

3.7.1 Defining the Attitude Error Vector

The Celestial Navigation (CelNav) portion of this study is performed using a Multiplicative Extended Kalman Filter (MEKF) outlined in [26]. The MEKF is used over the traditional EKF due to the difficulty the EKF experiences when trying to estimate the system's attitude quaternion elements. The attitude quaternion produced by a typical EKF will not be of unit norm, creating issues in the QUEST algorithm. The MEKF estimates the three elements of an attitude error vector, $\underline{\boldsymbol{a}}$, instead of all four components of the attitude quaternion. The system's attitude quaternion is represented as a product of the two quaternions:

$$\bar{q} = \delta\bar{q} \otimes \bar{q}_{\text{ref}} \tag{3.44}$$

The first quaternion, $\delta\bar{q}$, represents a small rotation from the reference frame to the current true attitude of the system. The second quaternion, $\bar{q}_{\text{ref}}$, represents the reference rotation from the MCI to the body frame. From the quaternion definition in Eq. 3.3, $\delta\bar{q}$ can be represented as a rotation around the Euler axis, $\underline{\hat{\boldsymbol{e}}} = [e_1\ e_2\ e_3]^T$, by some angle amount θ

$$\delta\bar{q} = \begin{bmatrix} e_1 \sin(\frac{\theta}{2}) \\ e_2 \sin(\frac{\theta}{2}) \\ e_3 \sin(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) \end{bmatrix} \tag{3.45}$$

Assuming θ is very small, the small-angle approximations $\sin\theta \approx \theta$ and $\cos\theta \approx 1$ can be used to rewrite Eq. 3.45 as

$$\delta\bar{q} = \begin{bmatrix} e_1\frac{\theta}{2} \\ e_2\frac{\theta}{2} \\ e_3\frac{\theta}{2} \\ 1 \end{bmatrix} \tag{3.46}$$

The attitude vector is then defined to have magnitude θ along the Euler axis $\underline{\hat{\boldsymbol{e}}}$

$$\underline{\boldsymbol{a}} := \theta\underline{\hat{\boldsymbol{e}}} \tag{3.47}$$

Finally, $\delta\bar{q}$ can be defined in terms of the attitude vector

$$\delta\bar{q}(\underline{\boldsymbol{a}}) \approx \begin{bmatrix} \frac{1}{2}\underline{\boldsymbol{a}} \\ 1 \end{bmatrix} \tag{3.48}$$

More on the derivation of the attitude error vector and its dynamics can be found in [33].

3.7.2 Filter State Vector and Measurement Model

The MEKF state is composed of the attitude error vector, longitude, latitude, and accelerometer biases:

$$\underline{\boldsymbol{x}} = \begin{bmatrix} \underline{\boldsymbol{a}} \\ \lambda \\ \phi \\ \underline{\boldsymbol{b}}_a \end{bmatrix} \tag{3.49}$$

The attitude error vector is included in the state to represent the attitude quaternion error between estimates. After each estimate, $\underline{\boldsymbol{a}}$ is reset to $\underline{\mathbf{0}}_{3\times1}$ and the reference quaternion is updated accordingly, consistent with the MEKF definition. The λ and ϕ values represent the position with respect to the MCMF coordinate frame. The LGM2011 gravity model is used to reference expected gravity measurements and how they vary with position across the lunar surface. The accelerometer bias, $\underline{\boldsymbol{b}}_a$, is included to improve the accuracy of the λ and ϕ estimates. Accelerometers inherently include biases that are dynamic over time. Known biases can be accounted for, but dynamic biases leading to inaccuracy in gravity measurements cannot be determined solely by stationary measurement. How this bias is calculated through different system rotations is discussed further in Section 3.8.

The MEKF measurement model can be defined using

$$\underline{\boldsymbol{h}}(\underline{\boldsymbol{x}}) = \begin{bmatrix} \underline{\boldsymbol{a}} \\ \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}}) \end{bmatrix} \tag{3.50}$$

where $\underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})$ is the gravity measurement model defined by

$$\underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}}) = R(\underline{\boldsymbol{a}}) R^{\text{Body}}_{\text{MCI}}(\bar{q}_{\text{ref}}) R^{\text{MCI}}_{\text{MCMF}}(t) \underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda, \phi) + \underline{\boldsymbol{b}}_a \tag{3.51}$$

The gravity measurement model is a function of the attitude error vector, the reference attitude quaternion, the time between the initial and current step, the expected LGM2011 gravity based on current λ and ϕ estimates, and the accelerometer bias. The rotation matrix associated with the attitude error vector, $R(\underline{\boldsymbol{a}})$, is defined as

$$R(\underline{\boldsymbol{a}}) = I_{3\times3} + [[\underline{\boldsymbol{a}}]] \tag{3.52}$$

where $[[\underline{\boldsymbol{a}}]]$ is the cross-product matrix of the attitude error vector found using Eq. 3.15. The matrix $R^{\text{Body}}_{\text{MCI}}(\bar{q}_{\text{ref}})$ is the rotation matrix associated with the reference quaternion found using Eq. 3.14. The $R^{\text{MCI}}_{\text{MCMF}}(t)$ matrix represents the rotation from the MCMF to MCI frame found by taking the transpose of

$$R_{\text{MCI}}^{\text{MCMF}}(t) = \begin{bmatrix} \cos(\alpha_g) & \sin(\alpha_g) & 0 \\ -\sin(\alpha_g) & \cos(\alpha_g) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.53}$$

$$\alpha_g = \alpha_{g0} + \omega_{\text{moon}}(t - t_0) \tag{3.54}$$

where α_{g0} is the hour angle at time t_0 and $\omega_{\text{moon}} = 0.266\,169\,95 \times 10^{-5}$ rad/s is the rotation rate of the Moon. $\underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda, \phi)$ represents the gravity vector based on the LGM2011 gravity model at (λ, ϕ). The Jacobian of the measurement model, $H(\underline{\boldsymbol{x}})$, is found by taking the derivative of $\underline{\boldsymbol{h}}(\underline{\boldsymbol{x}})$ with respect to each state element, as shown in Eq. 3.55:

$$\begin{aligned} H(\underline{\boldsymbol{x}}) &= \begin{bmatrix} \frac{\partial \underline{\boldsymbol{a}}}{\partial a_1} & \frac{\partial \underline{\boldsymbol{a}}}{\partial a_2} & \frac{\partial \underline{\boldsymbol{a}}}{\partial a_3} & \frac{\partial \underline{\boldsymbol{a}}}{\partial \lambda} & \frac{\partial \underline{\boldsymbol{a}}}{\partial \phi} & \frac{\partial \underline{\boldsymbol{a}}}{\partial b_1} & \frac{\partial \underline{\boldsymbol{a}}}{\partial b_2} & \frac{\partial \underline{\boldsymbol{a}}}{\partial b_3} \\ \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial a_1} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial a_2} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial a_3} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial \lambda} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial \phi} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial b_x} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial b_y} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial b_z} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial a_1} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial a_2} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial a_3} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial \lambda} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial \phi} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial b_x} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial b_y} & \frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial b_z} \end{bmatrix} \end{aligned} \tag{3.55}$$

where the individual derivatives are defined as

$$\frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial a_i} = [[\underline{\boldsymbol{e}}_i]] R_{\text{MCI}}^{\text{Body}}(\bar{q}_{\text{ref}}) R_{\text{MCMF}}^{\text{MCI}}(t) \underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda, \phi) \tag{3.56}$$

$$\frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial \lambda} = R(\underline{\boldsymbol{a}}) R_{\text{MCI}}^{\text{Body}}(\bar{q}_{\text{ref}}) R_{\text{MCMF}}^{\text{MCI}}(t) \frac{\partial \underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda, \phi)}{\partial \lambda} \tag{3.57}$$

$$\frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial \phi} = R(\underline{\boldsymbol{a}}) R_{\text{MCI}}^{\text{Body}}(\bar{q}_{\text{ref}}) R_{\text{MCMF}}^{\text{MCI}}(t) \frac{\partial \underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda, \phi)}{\partial \phi} \tag{3.58}$$

$$\frac{\partial \underline{\boldsymbol{h}}_g(\underline{\boldsymbol{x}})}{\partial b_i} = \underline{\boldsymbol{e}}_i \tag{3.59}$$

where $\underline{\boldsymbol{e}}_i$ is a vector of all zeros with a single one in position i. From Eqs. (3.56-3.59), see that values for $\underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda, \phi)$, $\frac{\partial \underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda,\phi)}{\partial \lambda}$, and $\frac{\partial \underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda,\phi)}{\partial \phi}$ are needed to complete the Jacobian matrix (and values between LGM2011 grid points will be needed in general). MATLAB's Cubic Spline with Additional Piecewise End Conditions (`csape`) function can be used to approximate function and derivative values at arbitrary (λ, ϕ) positions given the LGM2011 gravity values across the lunar surface. MATLAB's `csape` function returns the cubic spline interpolation of data supplied in `ppform`, a data structure used to represent splines or piecewise polynomial functions [34]. Lagrange end conditions are applied to each end of the data and matches spline endslopes to the slope of the cubic polynomial that fits the last four data points at each end. Example function calls are in Algorithm 1. where $\{\phi_{\text{range}}, \lambda_{\text{range}}\}$ are the latitude and longitude ranges at the region of interest on the lunar surface. It is important to note that the order of $\{\phi_{\text{range}}, \lambda_{\text{range}}\}$ in the csape input matches the LGM2011 data format. The LGM2011 model provides the $g_{x,\text{LGM}}$, $g_{y,\text{LGM}}$ and $g_{z,\text{LGM}}$ gravity values in 3600×7200 arrays, corresponding to the dimensions of the ϕ and λ vectors respectively.

Algorithm 1 Compute Gravity Vector Spline and Derivatives Using `csape`

```
Input: $\phi_{\text{range}}, \lambda_{\text{range}}, g_{x,\text{LGM}}, g_{y,\text{LGM}}, g_{z,\text{LGM}}$
Output: $\underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda, \phi), \frac{\partial \underline{\boldsymbol{g}}_{\text{MCMF}}}{\partial \phi}, \frac{\partial \underline{\boldsymbol{g}}_{\text{MCMF}}}{\partial \lambda}$
$\underline{\boldsymbol{g}}_{\text{LGM,Range}} \leftarrow [g_{x,\text{LGM}}(\phi_{\text{range}}, \lambda_{\text{range}}), g_{y,\text{LGM}}(\phi_{\text{range}}, \lambda_{\text{range}}), g_{z,\text{LGM}}(\phi_{\text{range}}, \lambda_{\text{range}})]^T$
$\underline{\boldsymbol{g}}_{\text{MCMF}}(\lambda, \phi) \leftarrow$ csape($\{\phi_{\text{range}}, \lambda_{\text{range}}\}, \underline{\boldsymbol{g}}_{\text{LGM,Range}}$)
$\frac{\partial \underline{\boldsymbol{g}}_{\text{MCMF}}}{\partial \phi} \leftarrow$ fnder($\underline{\boldsymbol{g}}_{\text{MCMF}}, [1, 0]$)
$\frac{\partial \underline{\boldsymbol{g}}_{\text{MCMF}}}{\partial \lambda} \leftarrow$ fnder($\underline{\boldsymbol{g}}_{\text{MCMF}}, [0, 1]$)
```

3.7.3 Initial State and Error Covariance Estimates

The MEKF requires an initial state and covariance estimate to begin the algorithm. The MEKF starts by running the QUEST algorithm to provide an initial reference attitude quaternion, $\bar{q}_{\text{ref}}$, and attitude covariance matrix, $P_{\theta\theta}$. Since no other references are available to compare with, the initial attitude error vector estimate is $\underline{\mathbf{0}}_{3\times 1}$. Future lunar missions have specific plans and landing sites for where they want to explore on the surface. Therefore, the initial (λ_0, ϕ_0) position is assumed to follow the normal distributions shown below:

$$\lambda_0 \backsim \mathcal{N}(\lambda_{\text{true}}, \sigma_\lambda^2) \tag{3.60}$$

$$\phi_0 \backsim \mathcal{N}(\phi_{\text{true}}, \sigma_\phi^2) \tag{3.61}$$

where σ_λ and σ_ϕ are the uncertainties in the initial λ_0 and ϕ_0 guesses. Lastly, the initial bias estimate, $\underline{\boldsymbol{b}}_{a,0}$, is defined to be $\underline{\mathbf{0}}_{3\times 1}$ since the bias is unknown. The initial state estimate components described above are combined to form

$$\underline{\boldsymbol{x}}_0 = \begin{bmatrix} \underline{\boldsymbol{a}}_0 \\ \lambda_0 \\ \phi_0 \\ \underline{\boldsymbol{b}}_{a,0} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{0}}_{3\times 1} \\ \lambda_0 \\ \phi_0 \\ \underline{\mathbf{0}}_{3\times 1} \end{bmatrix} \tag{3.62}$$

The initial covariance matrix depends on the uncertainty in the initial state estimates. Using Eq. 3.42, the QUEST algorithm provides a covariance matrix, $P_{\theta\theta}$, describing the accuracy of the output reference quaternion. The initial (λ_0, ϕ_0) covariance matrix is defined using the standard deviation values of the Gaussian distributions in Eqs. 3.60 and 3.61

$$P_{\lambda_0,\phi_0} = \begin{bmatrix} \sigma_\lambda^2 & 0 \\ 0 & \sigma_\phi^2 \end{bmatrix} \tag{3.63}$$

The initial covariance associated with the bias, $P_{\underline{\boldsymbol{b}}_{a,0}}$, is found by assuming a σ_b value. Since the initial bias behavior is unknown, σ_b is assumed to have a higher value in comparison to typical uncertainty values. The definition of $P_{\underline{\boldsymbol{b}}_{a,0}}$ is shown below:

$$P_{\underline{\boldsymbol{b}}_{a,0}} = \sigma_b^2 I_{3\times3} \tag{3.64}$$

The initial component covariance matrices are combined to form the overall P_0 matrix shown in Eq. 3.65, completing the initial state and covariance guesses needed for the MEKF.

$$P_0 = \begin{bmatrix} P_{\theta\theta} & 0_{3\times2} & 0_{3\times3} \\ 0_{2\times3} & P_{\lambda_0,\phi_0} & 0_{2\times3} \\ 0_{3\times3} & 0_{3\times2} & P_{\underline{\boldsymbol{b}}_{a,0}} \end{bmatrix} \tag{3.65}$$

3.7.4 MEKF Notation Convention

It is important to understand the conventions used in the MEKF algorithm and what data is used during each step, and there are many existing conventions for this notation. Table 3.1 shows the MEKF variable names used in this study, what they do, and what data it used to calculate that value.

Table 3.1: Kalman filter variable convention and step dependence.

Variable	Definition
$\hat{\underline{\boldsymbol{x}}}(k\|k)$	State estimate at step k based on step k information
$P(k\|k)$	Covariance matrix at step k based on step k information
$\overline{\underline{\boldsymbol{x}}}(k+1\|k)$	Dynamically propagated state based on step k information
$\overline{P}(k+1\|k)$	Dynamically propagated covariance based on step k information
$\hat{\underline{\boldsymbol{x}}}(k+1\|k+1)$	State estimate at step k+1 based on step k+1 information
$P(k+1\|k+1)$	Covariance matrix at step k+1 based on step k+1 information

3.7.5 State Dynamic Propagation

Dynamic propagation involves predicting the state and covariance at step $k+1$ using step k information. Since the system is stationary on the surface, the state vector ideally remains constant. Therefore, the state dynamic propagation is defined as

$$\overline{\underline{\boldsymbol{x}}}(k+1|k) = \hat{\underline{\boldsymbol{x}}}(k|k) \tag{3.66}$$

The attitude vector in the state remains unchanged; however, it is important to note that the reference quaternion is updated based on the Moon's rotation between time steps, requiring a readjustment of the uncertainties in the covariance matrix. The covariance elements corresponding to the (λ, ϕ) position and accelerometer bias are assumed unchanged between time steps, leaving the elements of $P_{\lambda,\phi}$ and $P_{\underline{\boldsymbol{b}}_a}$ unaffected. In real operation, the accelerometer bias may change and require process noise to inflate the corresponding terms in the covariance matrix. We're assuming that bias is constant over one estimation cycle. The quaternion between time steps, however, is changed due to the Moon's rotation, causing the attitude matrix relating the MCI frame to the body frame to vary. The new quaternion resulting from the rotation introduces more uncertainty that needs to be accounted for in the overall covariance matrix propagation. Two different methods can be used to account for the change in the quaternion due to rotation of the Moon. First, it can be assumed that the moon rotates purely about its z-axis. Second, the precession/nutation of the Moon over time can be used, causing the rotation to no longer be purely about the z-axis.

3.7.6 Covariance Dynamic Propagation for Pure Rotation

A method for accounting for the quaternion rotation is provided in [33]. The first step involves finding how much the moon has rotated since the last time step

$$\Delta\theta = \omega_{\text{moon}}\Delta t \tag{3.67}$$

$$\Delta t = t(k+1) - t(k) \tag{3.68}$$

Next, the rotation matrix corresponding to the rotation about the z axis is found using

$$R(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & \sin(\Delta\theta) & 0 \\ -\sin(\Delta\theta) & \cos(\Delta\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.69}$$

The quaternion associated with the $R(\Delta\theta)$ rotation matrix, $\bar{q}_{\text{rot}}$, is found using Bar Itzhack's method outlined in Eq. 3.16 with the components of $R(\Delta\theta)$. Then, a new reference quaternion using the quaternion product of the previous quaternion reference, $\bar{q}_{\text{ref,prev}}$, and the new quaternion due to the Moon's rotation, $\bar{q}_{\text{rot}}$, is defined using

$$\bar{q}_{\text{ref,new}} = \bar{q}_{\text{ref,prev}} \;\otimes\; \bar{q}_{\text{rot}} \tag{3.70}$$

The matrix representing the quaternion change due to the Moon's rotation, M_{rot}, is found using

$$M_{\text{rot}} = \{\bar{q}_{\text{ref,prev}} \;\otimes\; \bar{q}^{c}_{\text{ref,new}}\}_4 \; I_{3\times3} + [[\{\bar{q}_{\text{ref,prev}} \;\otimes\; \bar{q}^{c}_{\text{ref,new}}\}_{1:3}]] \tag{3.71}$$

Finally, the propagated covariance matrix, $\bar{P}(k+1|k)$, is defined using the M_{rot} matrix shown in 3.71 above:

$$\bar{P}(k+1|k) = \begin{bmatrix} M_{\text{rot}} & 0_{3\times5} \\ 0_{5\times3} & I_{5\times5} \end{bmatrix} P(k|k) \begin{bmatrix} M_{\text{rot}} & 0_{3\times5} \\ 0_{5\times3} & I_{5\times5} \end{bmatrix}^T + Q(k) \tag{3.72}$$

where $Q(k)$ represents the accuracy in knowing the Moon's rotation. $Q(k)$ is assumed to be constant between time steps for both pure rotation and precession cases. Eq. 3.72 validates the claim that the position and bias components of the covariance matrix remain unchanged, indicated by the 5 × 5 identity matrices.

3.7.7 Lunar Optical Librations

Before proceeding with the discussion, it is important to clarify that the libration effects described in this section are primarily due to our perspective of the Moon from Earth, rather than changes in its actual orientation in space. Eckhardt goes into detail about the theory of optical lunar librations and their root causes in [35]. Eckardt talks about Jean Dominique Cassini and his elegant description of the Moon's rotations, which govern its optical librations as the sum of two uniform motions. These motions are synchronized with both the period and precession of its orbit around the Earth. These descriptions are referred to as Cassini's rules for lunar rotation and are as follows:

1. The Moon rotates uniformly about its polar axis with a rotational period equal to the mean sidereal period of its orbit about the Earth.
2. The descending node of the lunar equator on the ecliptic precesses in coincidence with the ascending node of the lunar orbit on the ecliptic.

Cassini's rules lead to two primary optical librations: libration in longitude and libration in latitude. The libration in longitude is due to Cassini's first rule, which is caused by the Moon's rotation being relatively uniform even though the Moon's orbital motion is variable due to its eccentricity. This means that the orbital speed of the Moon around Earth is greater than its rotation rate near periapsis and slower than its rotation near apoapsis. The Moon's rotation rate is approximately one full rotation per revolution around the Earth. The difference in orbital speed and rotation rate leads to the Moon showing more of its eastern and western portions of its surface at a particular time. A visualization of this phenomenon is shown in Figure 3.5. An easier way to think of this is that the Moon is slowly shaking its head side to side when viewed from the Earth. Libration in latitude is caused by the Moon's equator having a slight inclination relative to the lunar orbit around the Earth, as seen in Figure 3.6. This leads to the Moon showing more of its northern and southern portions of its surface at a particular time, which can be described as the Moon slowly nodding up and down. Video demonstrations of these librations can be found at [36] and [37].

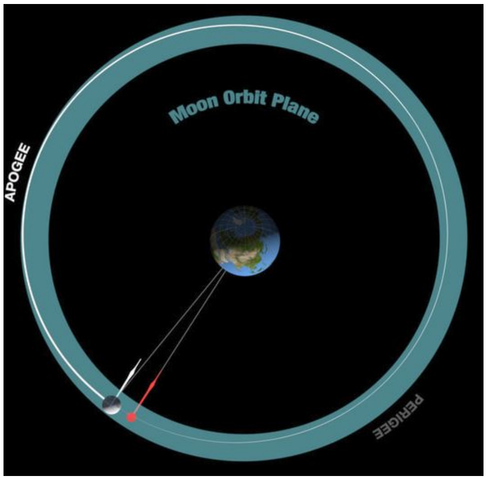


Figure 3.5: The eccentricity of the lunar orbit around Earth shown with respect to the rotation rate shown causes the optical libration in longitude observed the Earth's surface. (Figure reproduced from [37])

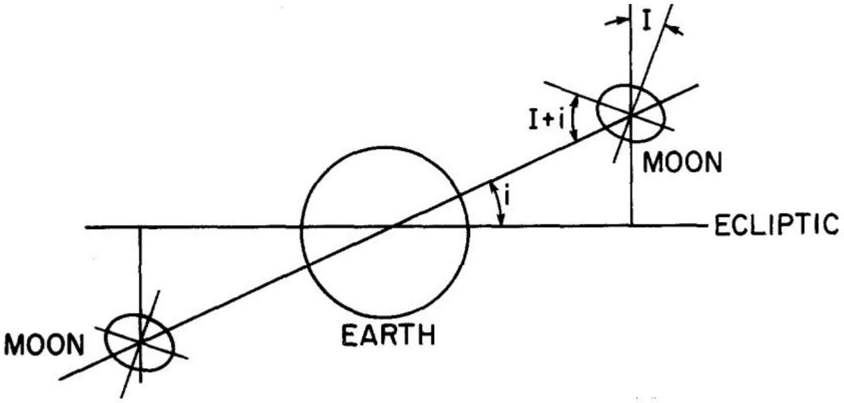


Figure 3.6: The inclination of lunar equator relative to its orbit around the Earth shown here causes the optical libration in latitude observed from the Earth's surface. (Figure reproduced from [35])

3.7.8 JPL DE405 Database

To incorporate precise lunar orientation data in the dynamic propagation of the attitude quaternion, a dataset containing ephemeris and other planetary values is needed. The Jet Propulsion Laboratory (JPL) has multiple Development Ephemeris (DE) models available for the public to use. JPL planetary ephemerides are saved as files of Chebyshev polynomials fit to the Cartesian positions and velocities of the planets, Sun, and the Moon, usually in 32-day intervals [38]. Most of the JPL planetary ephemeris data include Chebyshev polynomials fit to the lunar libration angles, which are integrated along with the planetary positions. This study uses the DE405 dataset due to its MATLAB compatibility and rotation matrix from Moon Principal Axis (PA) frame to Mean Earth (ME) frame shown in Eq. 3.73 [39] . The ME frame is assumed to be equal to the MCMF frame in this study.

$$R_{\mathrm{PA}}^{\mathrm{MCMF}} = R_x(-0.1462'')R_y(-79.0768'')R_z(-63.8986'') \tag{3.73}$$

The DE405 dataset replaced the DE403 model in 1998 and provides improvements in the orientation of the ephemerides on the J2000 International Celestial Reference Frame (ICRF)

[40]. The DE405 data stems from a least-squares adjustment of the previous DE403 ephemeris model based on improved observational data and measurements [41]. The dynamical equations of motion are then numerically integrated using the updated data to describe the gravitational physics of the solar system. The primary values needed from the DE405 database are the Euler angles of the Moon with respect to the ICRF reference frame.

3.7.9 Moon Euler Angles

Another thorough analysis of lunar libration with additional information about Euler angle calculation is found in [39]. The Euler angles, ϕ, θ, and ψ, describe the orientation of the Moon's PA frame with respect to the ECI frame. More on how they are calculated can be found in [42]. The angles are defined relative to the ICRS Earth equator and equinox and describe the orientation of the Moon principal axes of inertia as shown in Figure 3.7

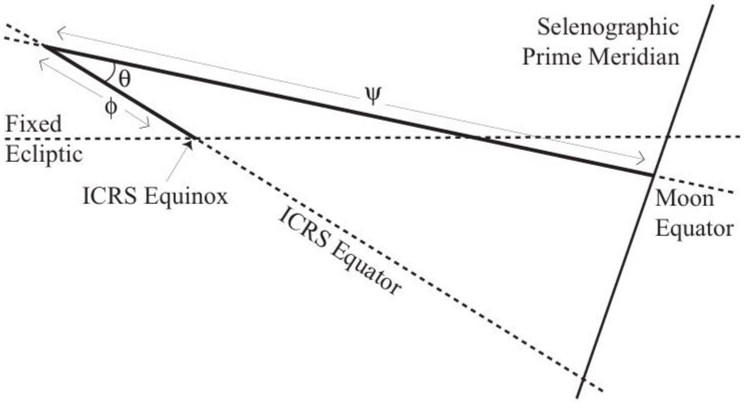


Figure 3.7: The Euler angles ϕ, θ, and ψ, are defined relative to the equatorial reference frame and used to describe the lunar PA system. (Figure reproduced from [39])

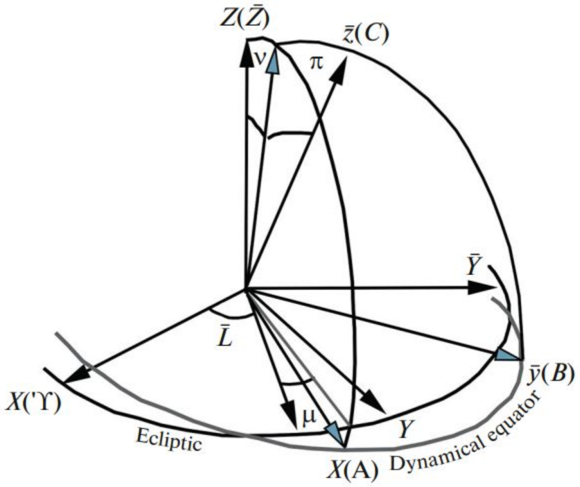


Figure 3.8: Libration angles μ, ν, and π with respect to the Selenocentric ecliptic coordinate system $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$, the uniformly rotating ecliptic system $(\overline{\boldsymbol{X}}, \overline{\boldsymbol{Y}}, \overline{\boldsymbol{Z}})$, and the dynamic coordinate system $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{z}})$. (Figure reproduced from [43])

The orientation of the moon and how it rotates is very important in accounting for the attitude quaternion between time steps. Table 3.2 contains the definitions of each of the Euler angles.

Table 3.2: Euler angle geometric definition [39].

Euler Angle	**Definition**
ϕ	The angle from the ICRS x-axis to the ascending node of the lunar equator
θ	The inclination of the lunar equator to the ICRS equator
ψ	The angle along the lunar equator from the node to the lunar prime meridian

Zagidullin et al. performed a study to improve the accuracy of the theory of the Moon's physical libration in comparison with the theory previously know for a solid Moon model [43]. Within this study, they provide equations that use the Moon's Euler angles to form the libration angles μ, ν, and π shown in Figure 3.8.

They go on to define the mathematically necessary transition between the ecliptic and dynamic coordinate systems and how it is carried out using rotation matrices. The libration angles can then be related through the product of the following rotation matrices

$$R_x(-\pi)R_y(\nu)R_z(M) = R_z(\psi)R_x(-\theta)R_z(\phi) \tag{3.74}$$

where $M = L + \mu$, with L defining the mean longitude of the Moon, measured from the axis pointed towards the vernal equinox. From the rotations in Eq. 3.74 above, the equations used to solve for the libration angles as a function of the Euler angles at a specific time can be defined.

$$\sin(\nu) = \sin(\psi)\sin(\theta) \tag{3.75}$$

$$\sin(\pi)\cos(\nu) = \cos(\psi)\sin(\theta) \tag{3.76}$$

$$\sin(M)\cos(\nu) = \cos(\psi)\sin(\phi) + \sin(\psi)\cos(\theta)\cos(\phi) \tag{3.77}$$

$$\cos(M)\cos(\nu) = \cos(\psi)\cos(\phi) - \sin(\psi)\cos(\theta)\sin(\phi) \tag{3.78}$$

Eqs. (3.75–3.78) allow the initial values of the libration angles to be calculated. More on how to find the derivatives of Eqs. (3.75–3.78) is also discussed in [43], but those values are not needed for this study. The Euler angles can be calculated directly from MATLAB using the `moonLibration` function. The function inputs include the Julian Date (either a single time or vector of times) and desired DE dataset. The outputs include the Euler angles and their respective rates. An example function call is shown below.

$$[\text{angles}, \text{rates}] = \texttt{moonLibration}(\text{jdate}, 405) \tag{3.79}$$

where DE405 is the dataset used in the example above. MATLAB documentation provides further information on what all datasets can be used. The "angles" output contains a $N \times 3$

array with the Euler angles in radians at all jdate times. It is important to ensure the correct order of the output angles, with ϕ, θ, and ψ being in the proper sequence.

3.7.10 Covariance Dynamic Propagation Including Librations and Precessions

The covariance dynamic propagation incorporating high-fidelity lunar orientation uses the Euler angles, (ϕ, θ, and ψ), and rotation matrices, $R_{\text{MCI}}^{\text{ECI}}$ and $R_{\text{PA}}^{\text{MCMF}}$, defined in the sections above to calculate $\bar{P}(k+1|k)$. The rotation matrix from the PA to ECI frame is found using the Euler angles at a given time step

$$R_{\text{PA}}^{\text{ECI}} = R_z(-\phi)R_x(-\theta)R_z(-\psi) \tag{3.80}$$

The quaternion produced by Shuster's QUEST algorithm defines the rotation from the MCI to body frame. Therefore, the rotations corresponding to quaternions at times $t_k = t(k)$ and $t_{k+1} = t(k+1)$ are

$$\bar{q}_1 = R_{\text{MCI}}^{\text{Body}}(t_k) = R_{\text{MCI}}^{\text{MCMF}}(t_k)R_{\text{MCMF}}^{\text{Body}} \tag{3.81}$$

$$\bar{q}_2 = R_{\text{MCI}}^{\text{Body}}(t_{k+1}) = R_{\text{MCI}}^{\text{MCMF}}(t_{k+1})R_{\text{MCMF}}^{\text{Body}} \tag{3.82}$$

where $R_{\text{MCMF}}^{\text{Body}}$ is a constant rotation matrix representing rotation from MCMF to local body frame. The rotation from the body frame at time t_k to time t_{k+1} can then be found using

$$\begin{aligned} R_{\text{Body},t_k}^{\text{Body},t_{k+1}} &= R_{\text{MCI}}^{\text{Body}}(t_{k+1}) \left[R_{\text{MCI}}^{\text{Body}}(t_k)\right]^{-1} \\ &= R_{\text{MCI}}^{\text{MCMF}}(t_{k+1})R_{\text{MCMF}}^{\text{Body}}R_{\text{Body}}^{\text{MCMF}}R_{\text{MCMF}}^{\text{MCI}}(t_k) \end{aligned} \tag{3.83}$$

Since $R_{\text{MCMF}}^{\text{Body}}$ is an orthogonal matrix, $\left[R_{\text{MCMF}}^{\text{Body}}\right]^T = \left[R_{\text{MCMF}}^{\text{Body}}\right]^{-1}$, causing the terms in the above equation to cancel, leaving

$$R_{\text{Body},t_k}^{\text{Body},t_{k+1}} = R_{\text{MCI}}^{\text{MCMF}}(t_{k+1}) R_{\text{MCMF}}^{\text{MCI}}(t_k) \tag{3.84}$$

Can define an explicit expression for the $R_{\text{MCI}}^{\text{MCMF}}(t)$ matrix needed to complete the $R_{\text{Body},t_k}^{\text{Body},t_{k+1}}$ quaternion rotation between time steps using

$$R_{\text{MCI}}^{\text{MCMF}}(t) = R_{\text{PA}}^{\text{MCMF}} \left[R_{\text{PA}}^{\text{ECI}}(\phi, \theta, \psi)\right]^T R_{\text{MCI}}^{\text{ECI}} \tag{3.85}$$

Substituting Eq. 3.85 into Eq. 3.84 gives the final rotation of the quaternion attitude between time steps given the Moon's Euler angles at each time step

$$R_{\text{Body},t_k}^{\text{Body},t_{k+1}} = R_{\text{MCI}}^{\text{MCMF}}(t_{k+1}) \left[R_{\text{MCI}}^{\text{MCMF}}(t_k)\right]^{-1} \tag{3.86}$$

The remaining process to calculate the covariance dynamic propagation follows the same procedure outlined in Section 3.7.6, with the only change being that the matrix corresponding to the quaternion rotation is found using Eq. 3.86 instead of Eq. 3.69. The K_2 matrix is formed using the elements of $R_{\text{Body},t_k}^{\text{Body},t_{k+1}}$, producing the needed $\bar{q}_{\text{rot}}$ quaternion. M_{rot} and $\bar{P}(k+1|k)$ are still calculated using Eqs. 3.71 and 3.72 defined in Section 3.7.6.

3.7.11 Visualization of Pure Rotation and Precession Difference

This section shows what happens when an example vector is rotated over time by both the $R(\Delta\theta)$ and $R_{\text{Body},t_k}^{\text{Body},t_{k+1}}$ rotation matrices. Four different cases are shown, with each vector corresponding to points on the lunar surface. The first test case uses a vector initially on the MCMF x axis, $\underline{\boldsymbol{v}}_{x \text{ axis}} = [r_{\text{moon}}\ 0\ 0]^T$. The time step between each rotation is $\Delta t = 86\,400\,\text{s}$

(one day), ensuring that a significant change in the vector can be seen. One year's worth of data is taken for both cases. The magnitude of the vector differences is also stored over time. Figure 3.9 shows the three-dimensional position of $\underline{\boldsymbol{v}}_{x \text{ axis}}$ over time for both pure and librating rotation. It is clear from Figure 3.9 that the primary change in position is in the z component of $\underline{\boldsymbol{v}}_{x \text{ axis}}$. Each component position over time is shown in Figure 3.10. Figure 3.10 shows that even though the x and y components of $\underline{\boldsymbol{v}}_{x \text{ axis}}$ have a slight variation between the two rotation cases over time, it is not on the scale of the z component shown in Figure 3.10c. This leads to the linearly increasing overall magnitude difference between the pure and librating cases, as shown in Figure 3.11.

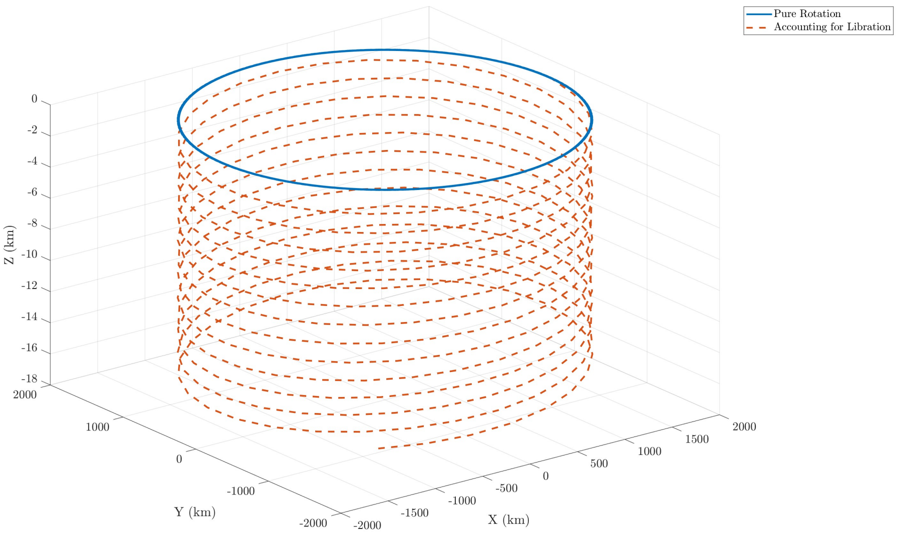
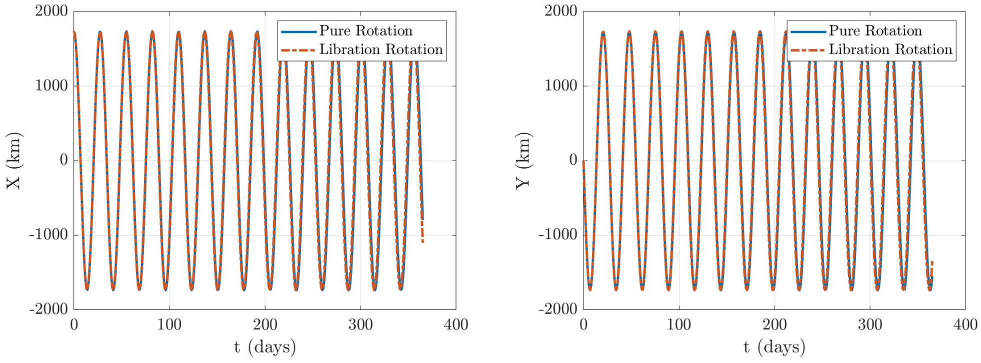
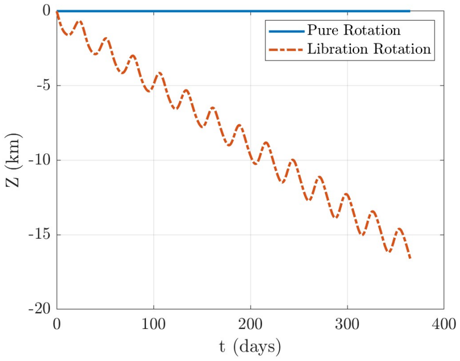


Figure 3.9: Pure and librating rotation of vector along MCMF x axis over one year. The pure rotation vector remains in its original plane, while the libration vector spirals over time primarily in the negative z direction. See that the scale of the z axis is much less than the x and y axes.



(a) x component (b) y component



(c) z component

Figure 3.10: Pure rotation and libration component position values of $\underline{\boldsymbol{v}}_{x \text{ axis}}$ over a year.

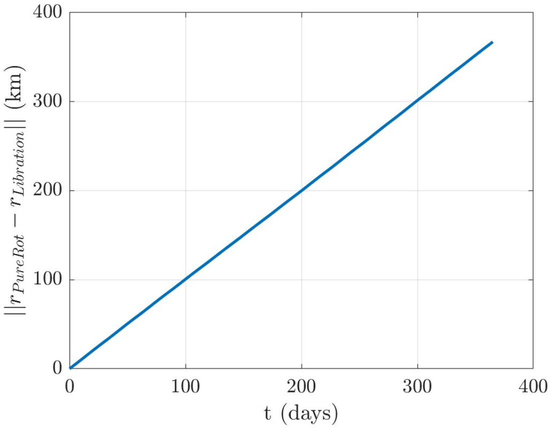


Figure 3.11: Magnitude of difference vector between pure rotation and libration over one year with initial vector aligning with the MCMF x axis.

The next test case shows the change in position over time based on a new initial vector, $\underline{\boldsymbol{v}}_{y \text{ axis}} = [0\ r_{\text{moon}}\ 0]^T$. The time step and overall data collection period remains the same as the x axis case, producing the results shown in Figures 3.12-3.14.

Figures 3.12–3.14 show a significant change in behavior compared to the initial x axis vector case, primarily in the z direction. The z component change in position is much less compared to the x axis initial vector case. This is clearly shown in Figure 3.12, where the vector rotation over time is no longer purely vertical like in Figure 3.9. The x and y positions along with the overall magnitude difference between the two rotation cases follow the same basic behavior seen with the x axis case.

The last test case shows the change in the position over time based on a new initial vector, $\underline{\boldsymbol{v}}_{z \text{ axis}} = [0\ 0\ r_{\text{moon}}]^T$. The time step and overall data collection period remain the same as in the previous two cases. The results are shown below in Figures 3.15-3.17.

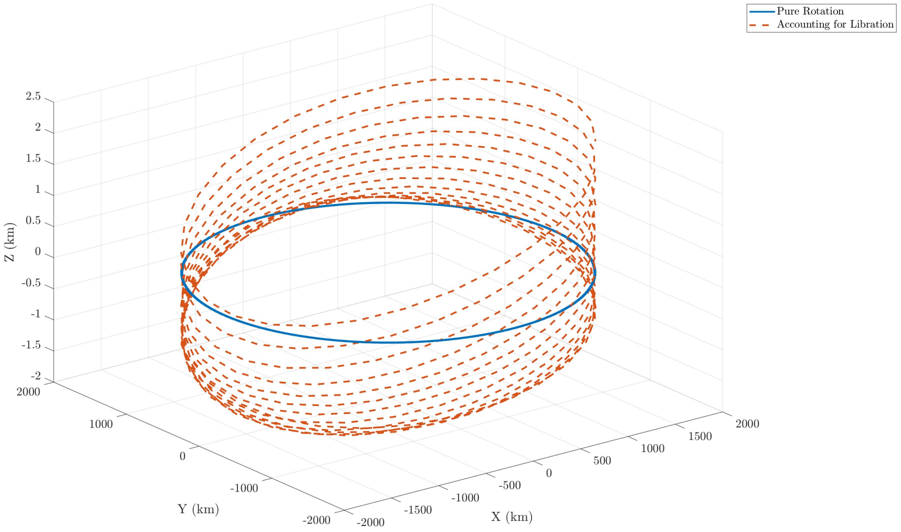
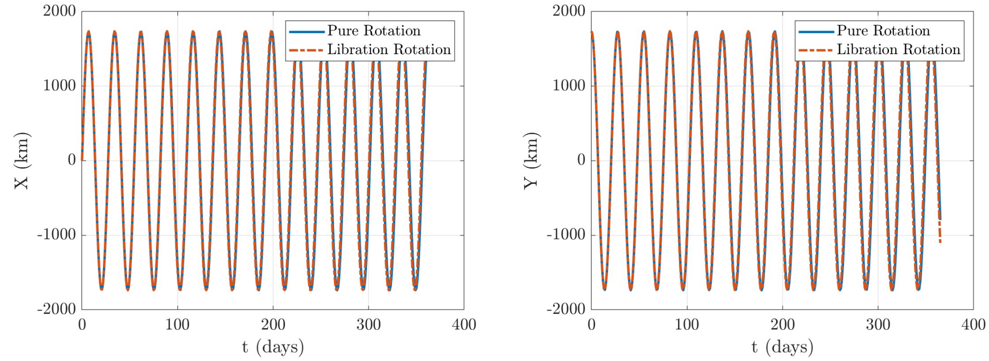
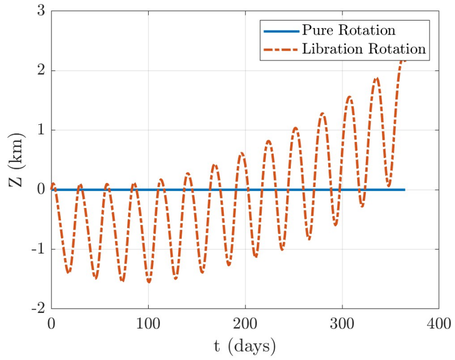


Figure 3.12: Pure and librating rotation of vector along MCMF y axis over one year. See a less drastic change in the z direction and more of a tilting motion of the libration vector instead of a spiral. Notice that the scale of the z axis is much less than the x and y axes.



(a) x component (b) y component



(c) z component

Figure 3.13: Component position values of $\underline{\boldsymbol{v}}_{y \text{ axis}}$ over a year.

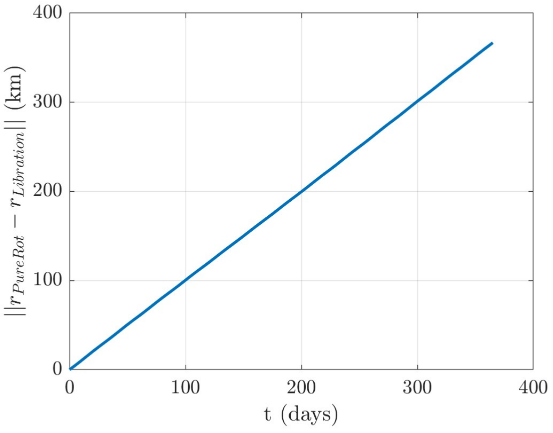


Figure 3.14: Magnitude of difference vector between pure rotation and libration over one year with initial vector aligning with the MCMF y axis.

Figures 3.15–3.17 show a drastic change in the behavior of the x and y position components over time. There is only a very slight change in the z position, shown by the order of the y axis in Figure 3.16c. Figures 3.16a and 3.16b show an increasing difference in the z and y positions between the pure rotation and librating cases. Figure 3.17 demonstrates that the new x and y component behavior causes the overall magnitude difference to no longer follow the straight line shown in Figures 3.11 and 3.14. The three test cases above provide insight into the behavior of the lunar librations and how they differ from pure rotation around the z axis. An initial vector not on one of the MCMF axes would have a combination of the behaviors seen above. This lunar libration analysis shows that not accounting for lunar libration would lead to inaccuracies in the rotated quaternion attitude needed for the covariance dynamic propagation shown in Section 3.7.10.

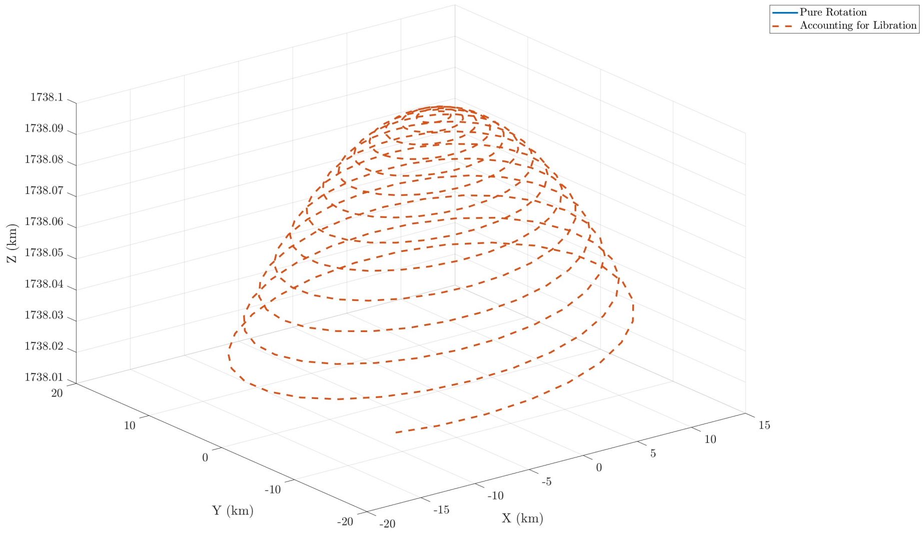
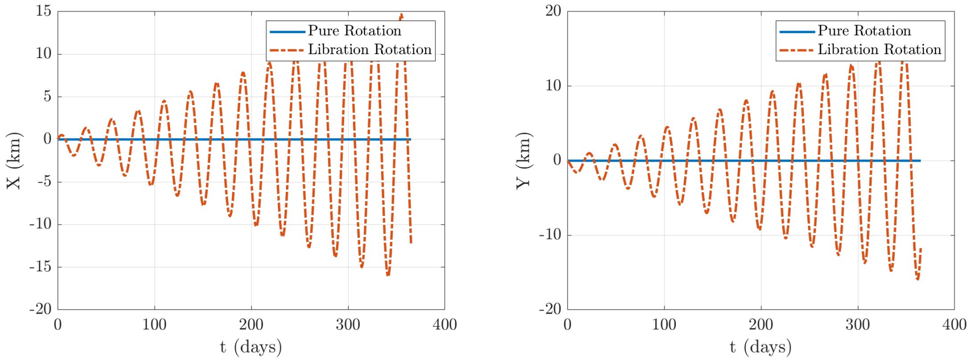
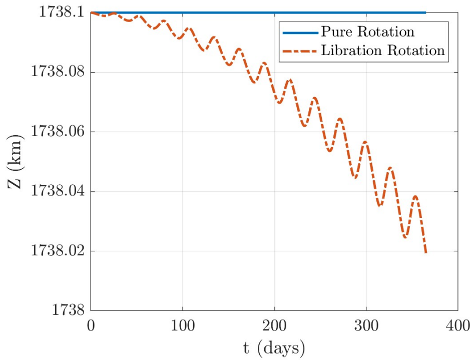


Figure 3.15: Pure and librating rotation of vector along MCMF z axis over one year. There is little variation in the z position based on the scaling of the z axis. See that the scale of the z axis is much greater than the x and y axes.



(a) x component (b) y component



(c) z component

Figure 3.16: Component position values of $\underline{\boldsymbol{v}}_{z \text{ axis}}$ over a year.

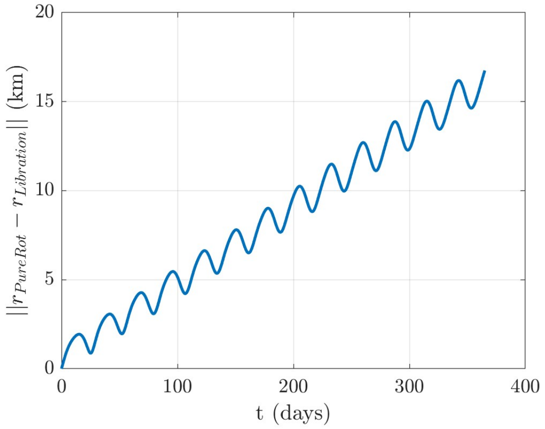


Figure 3.17: Magnitude of difference vector between pure rotation and libration over one year with initial vector aligning with the MCMF z axis.

3.7.12 Measurement and State Updates

Once dynamic propagation of the state and covariance have been performed, the next step in the MEKF is to update measurements using new data from the accelerometer and ST. The first step in this process involves calculating the innovation, $\underline{\boldsymbol{\nu}}(k+1)$, using the nonlinear measurement model defined in Eq. 3.50 and the measured attitude and gravity:

$$\underline{\boldsymbol{\nu}}(k+1) = \underline{\boldsymbol{z}}(k+1) - \underline{\boldsymbol{h}}(\underline{\bar{\boldsymbol{x}}}(k+1|k)) \tag{3.87}$$

The innovation covariance matrix, S, represents the difference in the actual measurement and the predicted measurement, which is calculated using

$$S(k+1) = H(\underline{\bar{\boldsymbol{x}}}(k+1|k))\bar{P}(k+1|k)H^T(\underline{\bar{\boldsymbol{x}}}(k+1|k)) + R(k+1) \tag{3.88}$$

where $R(k+1)$ is the measurement noise covariance matrix representing the uncertainty in the ST and accelerometer measurements. $R(k+1)$ is calculated using

$$R(k+1) = \begin{bmatrix} P_{\theta\theta} & 0_{3\times3} \\ 0_{3\times3} & \sigma_g^2 I_{3\times3} \end{bmatrix} \tag{3.89}$$

where $P_{\theta\theta}$ is the covariance matrix associated with the QUEST quaternion measurement and σ_g is the uncertainty in the accelerometer gravity measurement. The Kalman Gain, $W(k+1)$, determines how much the predicted state estimate should be adjusted based on new ST and accelerometer measurements. $W(k+1)$ is calculated using

$$W(k+1) = \bar{P}(k+1|k)H^T(\underline{\bar{\boldsymbol{x}}}(k+1|k))S^{-1}(k+1) \tag{3.90}$$

The new state estimate, $\underline{\hat{\boldsymbol{x}}}(k+1|k+1)$, is then calculated using the dynamically-propagated state, the Kalman Gain, and the innovation vector as follows:

$$\underline{\hat{\boldsymbol{x}}}(k+1|k+1) = \underline{\overline{\boldsymbol{x}}}(k+1|k) + W(k+1)\underline{\boldsymbol{\nu}}(k+1) \tag{3.91}$$

There are multiple equations that can be used to update the state covariance, but the Joseph Covariance Update is chosen for this study [44]. The Joseph formula provides a general covariance update valid not only for the Kalman gain, but for any linear unbiased estimator under standard Kalman filtering assumptions. The Joseph covariance update uses The Kalman Gain, the Jacobian matrix, the dynamically propagated covariance, and the measurement noise covariance matrix to provide the following update

$$\begin{aligned} P(k+1|k+1) = {} & [I_{n_x \times n_x} - W(k+1)H(\underline{\overline{\boldsymbol{x}}}(k+1|k))] \\ & \times [I_{n_x \times n_x} - W(k+1)H(\underline{\overline{\boldsymbol{x}}}(k+1|k))]^T + W(k+1)R(k+1)W(k+1)^T \end{aligned} \tag{3.92}$$

The attitude error vector estimate, $\underline{\hat{\boldsymbol{a}}}(k+1|k+1)$, is found by taking the first three elements of the $\underline{\hat{\boldsymbol{x}}}(k+1|k+1)$ state update

$$\underline{\hat{\boldsymbol{a}}}(k+1|k+1) = \{\underline{\hat{\boldsymbol{x}}}(k+1|k+1)\}_{1:3} \tag{3.93}$$

The quaternion estimate accounting for the estimated attitude error vector is calculated using the following quaternion product:

$$\hat{\bar{q}}(k+1|k+1) = \delta\bar{q}(\underline{\hat{\boldsymbol{a}}}(k+1|k+1)) \otimes \bar{q}_{\text{ref}} \tag{3.94}$$

where $\delta\bar{q}(\hat{\underline{\boldsymbol{a}}}(k+1|k+1))$ is found using Eq. 3.48, and $\bar{q}_{\text{ref}}$ is the reference quaternion after accounting for moon libration in Eq. 3.70. The updated $P(k+1|k+1)$ covariance matrix must also account for the attitude error vector estimate, similar to modifying the dynamic propagation covariance matrix to account for rotation of the Moon. The matrix corresponding the the change between $\bar{q}_{\text{ref}}$ and $\hat{\bar{q}}(k+1|k+1)$, M_{update}, is calculated using the same process outlined in Eqs. 3.71 and 3.72.

$$M_{\text{update}} = \{\bar{q}_{\text{new}} \otimes \hat{\bar{q}}(k+1|k+1)\}_4 I_{3\times3} + \left[\left[\{\bar{q}_{\text{new}} \otimes \hat{\bar{q}}(k+1|k+1)\}_{1:3}\right]\right] \tag{3.95}$$

$$P(k+1|k+1)_{\text{new}} = \begin{bmatrix} M_{\text{update}} & 0_{3\times5} \\ 0_{5\times3} & I_{5\times5} \end{bmatrix} P(k+1|k+1)_{\text{prev}} \begin{bmatrix} M_{\text{update}} & 0_{3\times5} \\ 0_{5\times3} & I_{5\times5} \end{bmatrix}^T \tag{3.96}$$

The final steps after updating the covariance matrix to account for the attitude error vector include setting the the attitude error estimate back to zero (required by MEKF definition), defining the reference quaternion to be equal to the estimate quaternion found using the attitude vector, and incrementing k by one. These steps are shown in the equations below.

$$\hat{\underline{\boldsymbol{a}}}(k+1|k+1) = \underline{\boldsymbol{0}}_{\boldsymbol{3\times1}} \tag{3.97}$$

$$\bar{q}_{ref} = \hat{\bar{q}}(k+1|k+1) \tag{3.98}$$

$$k = k+1 \tag{3.99}$$

3.8 Solving for Accelerometer Biases

The CelNav MEKF procedure above in Section 3.7 estimates the attitude error vector, the (λ, ϕ) position, and the accelerometer bias. The accelerometer bias produced by the MEKF algorithm is not accurately calculated using only one measurement orientation due to the lack of observability. Batista et al. describe a method for accelerometer calibration and dynamic bias and gravity estimation in [45]. They show that the accelerometer bias can be calculated by taking gravity measurements in multiple orientations. This study utilizes two rotations in order to solve for the bias. The procedure to do so is as follows:

1. Place system in an initial orientation and run MEKF algorithm shown in Section 3.7.
2. Rotate the accelerometer by a certain amount in an arbitrary direction (first rotation was chosen to be +90° around local body frame z axis in this study).
    - After rotating, perform another QUEST measurement in new orientation to get new $\bar{q}_{\text{ref}}$ and $P_{\theta\theta}$ estimates.
    - Keep output covariance matrix, P, from the last time step of the MEKF algorithm the same except for the elements corresponding to star tracker measurements.
    - Replace the first 3×3 block of the P matrix with the $P_{\theta\theta}$ covariance matrix taken from the most recent QUEST algorithm run:

$$P(1:3, 1:3) = P_{\theta\theta} \tag{3.100}$$

3. Maintain orientation resulting from first rotation and run the MEKF algorithm again.
4. From current orientation, rotate the accelerometer one more time in an arbitrary direction (second rotation was chosen to be +90° around local body frame x axis).
    - Perform another QUEST measurement and update most recent covariance with the new $P_{\theta\theta}$ matrix as seen in Eq. 3.100.
5. Run the MEKF algorithm one more time in the orientation resulting from the combination of the first and second rotation.

6. The final state vector, $\underline{\hat{\boldsymbol{x}}}(k+1|k+1)$ calculated after all rotations and MEKF runs contains the final (λ, ϕ) position and bias estimate, $\underline{\boldsymbol{b}}_a$.

The procedure shown above causes the estimated bias values to fully converge after three MEKF runs and two reorientations (though as will be shown one reorientation achieves most of the effect). Failing to solve for the accelerometer bias would produce inaccurate (λ, ϕ) position estimates; therefore, this is a crucial step in ensuring the accuracy of the final MEKF results. A visual representation of the rotation process is shown below in Figure 3.18.

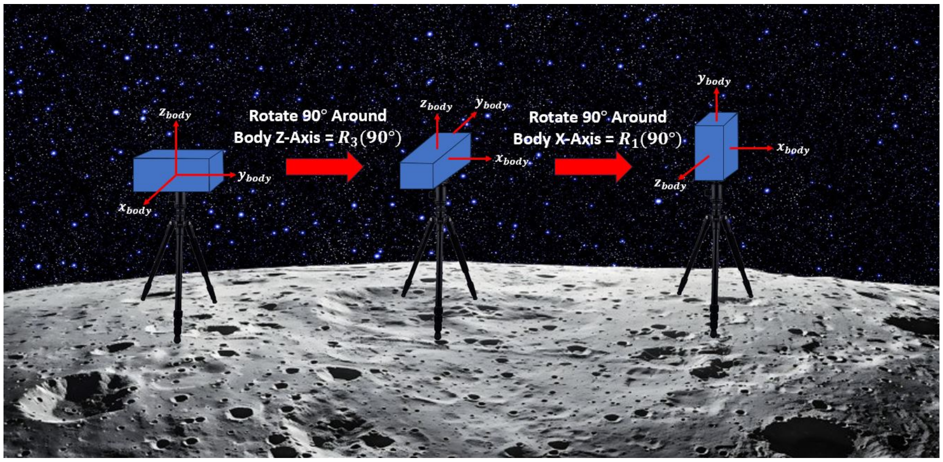


Figure 3.18: Accelerometer rotation procedure used to calculate unknown accelerometer bias. The first rotation takes place about the body-frame z axis followed by a second rotation about the body-frame's x axis.

3.9 Reference Position from Satellite Ranging

This section outlines the procedure used to simulate external reference measurements for our system.

3.9.1 Defining Localization Process and System State

The accurate reference position needed in Section 3.10 is provided using two-way ranging measurements with a lunar satellite. Cortinovis, Mina, and Gao outline a method for lunar positioning using a single satellite in [46]. Their analysis was created for use in the future Endurance mission, but their two-way ranging, single satellite based localization can also be applied to this study. Any external reference could be used for this purpose. This is just the example chosen for this study.

The key components to their model include a stationary rover and a satellite providing extended communication access. The state of the rover defines its position on the lunar surface in the Moon's PA frame:

$$\underline{\boldsymbol{x}}_{\text{rov,PA}} = \begin{bmatrix} x_{\text{PA}} \\ y_{\text{PA}} \\ z_{\text{PA}} \end{bmatrix} \tag{3.101}$$

Time synchronization between the rover and satellite is not needed assuming the system remains stationary between time steps and the satellite is able to perform two-way ranging.

3.9.2 Lunar Pathfinder Satellite

Future lunar missions have significant interest in exploring portions of the lunar south pole and the South Pole-Aitken Basin. The Apollo Peak Ring (APR) location was selected for this thesis due to its proximity to this region. The satellite communicating with the ground system needs to have long exposure to the APR. The Lunar Pathfinder satellite fulfills this requirement. The Lunar Pathfinder satellite follows an Elliptical Lunar Frozen Orbit (ELFO) around the moon, leading to wider coverage of the south pole region. Its orbital parameters are defined in Table 3.3. The resulting Pathfinder orbit in the Moon's PA frame, shown in Figure 3.19, corresponds two full orbital periods starting on October 1, 2030 at midnight. This orbit was propagated using MATLAB's `ode45` function.

Table 3.3: Lunar Pathfinder orbital parameters and orbital period [46].

Orbital Parameter	Value
Semi-major Axis (a)	5740 km
Eccentricity (e)	0.58
Inclination (i)	54.856°
RAAN (Ω)	0°
Argument of Periapsis (ω)	86.322°
Mean Anomaly (M)	180°
Orbital Period (T_0)	10.84 hrs

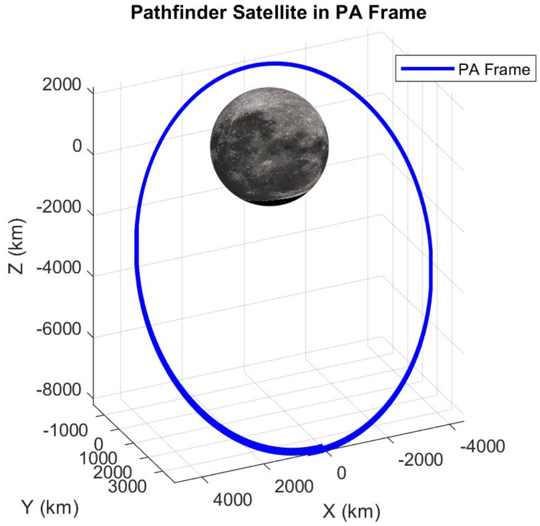
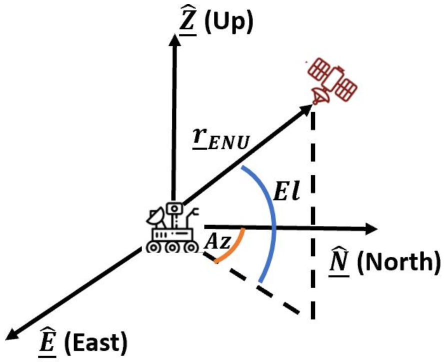


Figure 3.19: Lunar Pathfinder orbit in the Moon's PA frame. This orbit is calculated using MATLAB's `ode45` function using the provided orbital parameter for two full orbits [46].

3.9.3 Calculating Elevation Angle from Rover to Satellite

The satellite's ability to communicate with the rover on the surface is a function of the elevation angle, El, between the rover and the satellite. If the elevation is below a certain value, ranging measurements cannot be supplied to the rover. The elevation is defined as the angle above the horizon of the local ENU coordinate frame as seen in Figure 3.20.



$\hat{\underline{Z}}$ (Up)
$\underline{r}_{ENU}$
El
Az
$\hat{\underline{N}}$ (North)
$\hat{\underline{E}}$ (East)

Figure 3.20: Elevation angle definition based on local ENU system. Can see that the elevation corresponds to the angular value between the local horizon and the position vector of the satellite relative to the ground system.

The vector from the rover to the satellite in the local ENU frame, $\underline{\boldsymbol{r}}_{\text{ENU}} = [x_{\text{ENU}}\ y_{\text{ENU}}\ z_{\text{ENU}}]^T$, shown in Figure 3.20 is found using

$$\underline{\boldsymbol{r}}_{\text{ENU}} = R^{\text{ENU}}_{\text{MCMF}} R^{\text{MCMF}}_{\text{PA}} \underline{\boldsymbol{r}}_{\text{PA}} \tag{3.102}$$

$$R^{\text{ENU}}_{\text{MCMF}} = \begin{bmatrix} -\sin(\lambda) & \cos(\lambda) & 0 \\ -\sin(\phi)\cos(\lambda) & -\sin(\phi)\sin(\lambda) & \cos(\phi) \\ \cos(\phi)\cos(\lambda) & \cos(\phi)\sin(\lambda) & \sin(\phi) \end{bmatrix} \tag{3.103}$$

where $R^{\text{MCMF}}_{\text{PA}}$ is the matrix defined in Eq. 3.73 and $\underline{\boldsymbol{r}}_{\text{PA}}$ is the vector between the rover and satellite in the Moon's PA frame. Finally, the elevation can be calculated using

$$\text{El} = \sin^{-1}\left(\frac{z_{\text{ENU}}}{\|\underline{\boldsymbol{r}}_{\text{ENU}}\|}\right) \tag{3.104}$$

3.9.4 Navigation Signal Model

The Pathfinder satellite's ranging signal is modeled as a binary phase-shift keying (BPSK) signal, with carrier frequency f_S and spreading code frequency f_c shown in Table 3.4.

Table 3.4: Lunar Pathfinder transmitter signal parameters [46].

Signal Parameter	Value
f_S	2492.028 MHz
f_c	5.115 MHz

Table 3.5: Receiver signal parameters [46].

Receiver Parameter	Value
LNA Noise Figure ($\mathrm{NF}_{\mathrm{LNA}}$)	1 dB
System Noise Temperature (T_{sys})	113 K
DLL Bandwidth (B_{DLL})	0.5 Hz
Early-Late Spacing (d)	1 chip
Coherent Integration Time (T)	0.02 s
Front-End Bandwidth (B_{fe})	10.23 MHz

The rover's S-Band radio navigation receiver system and delay lock loop (DLL) parameters needed for this study are shown in Table 3.5. The quality of the received two-way ranging measurement, reflected by the measurement error covariance, depends on the received Carrier-to-Noise Density Ratio (C/N_0). The C/N_0 of the received measurement is defined using

$$C/N_0 = P_r + g/T - k_B \tag{3.105}$$

where P_r is the received isometric power in dBW, g/T is the receiver's gain-to-noise temperature ratio in dB/K, and k_B is the Boltzmann constant equal to -228.6 dBW/K/Hz. P_r

and g/T are found using

$$P_r = \mathrm{EIRP}(\beta) - 20\log_{10}\left(\frac{4\pi r f_S}{c}\right) \tag{3.106}$$

$$g/T = G_r(\theta) - T_{\mathrm{eq}} \tag{3.107}$$

$$T_{\mathrm{eq}} = 10\log_{10}\left(T_{\mathrm{sys}} + 290(10^{\mathrm{NF_{LNA}}/10} - 1)\right) \tag{3.108}$$

where $\mathrm{EIRP}(\beta)$ is the Elliptical Isotropic Radiated Power (antenna gain) of the Lunar Pathfinder at off-boresight angle β, c is the speed of light, $G_r(\theta)$ is the rover antenna gain at off-boresight angle θ, and T_{eq} is the equivalent noise temperature of the receiver. The antenna gains for the transmitter and receiver are defined in [47]. Figures 3.21 and 3.22 show the antenna gains versus off-boresight angle for the transmitter and receiver respectively.

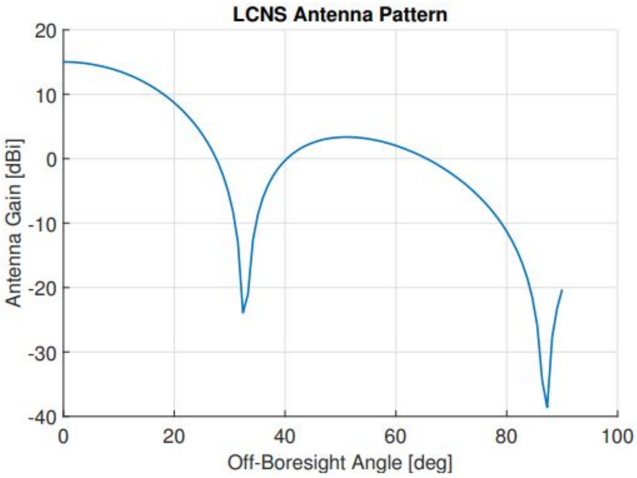


Figure 3.21: Lunar Pathfinder transmitter antenna gain $\mathrm{EIRP}(\beta)$ v. off-boresight angle β. (Figure reproduced from [47])

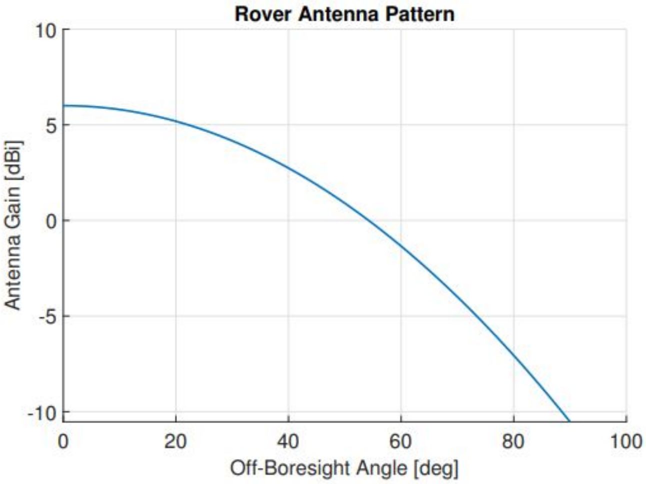


Figure 3.22: Receiver antenna gain $G_r(\theta)$ v. off-boresight angle θ. (Figure reproduced from [47])

The off-boresight angles for both the transmitter and receiver are needed to calculate the antenna gains and resulting C/N_0. The boresight of the transmitter is assumed to be aligned with the vertical of the antenna. The antenna is assumed to be aligned with the satellite's position vector in the PA frame. The boresight of the receiver is assumed to be aligned with the local nadir of the rover. Figure 3.23 shows these relationships.

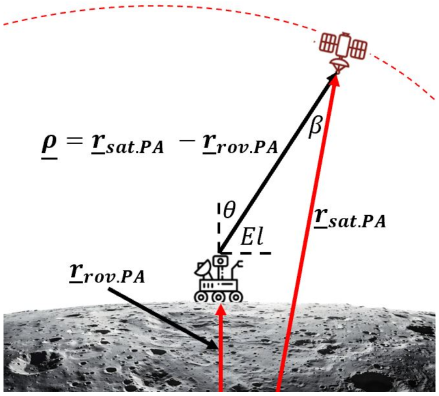

$\underline{\boldsymbol{\rho}} = \underline{\boldsymbol{r}}_{sat.PA} - \underline{\boldsymbol{r}}_{rov.PA}$
β
θ
El
$\underline{\boldsymbol{r}}_{sat.PA}$
$\underline{\boldsymbol{r}}_{rov.PA}$

Figure 3.23: Off-boresight angle geometry with respect to satellite position vector and local rover vertical. See that the satellite off-boresight angle is the angle between the position vector of the satellite with respect to the Moon's PA frame and the position vector of the satellite with respect to the ground system in the PA frame. The receiver off-boresight angle is simply 90° minus the elevation angle of the satellite.

The vector from the rover to the satellite in Figure 3.23 is defined as

$$\underline{\boldsymbol{\rho}} = \underline{\boldsymbol{r}}_{\text{sat,PA}} - \underline{\boldsymbol{r}}_{\text{rov,PA}} \tag{3.109}$$

The off-boresight angle for the receiver is simply a function of satellite elevation angle

$$\theta = 90^\circ - \text{El} \tag{3.110}$$

The off-boresight angle for the transmitter can be found using the dot product formula between $-\underline{\boldsymbol{r}}_{\text{sat,PA}}$ and $-\underline{\boldsymbol{\rho}}$ shown in Eq. 3.111:

$$\beta = \cos^{-1}\left(\frac{-\underline{\boldsymbol{r}}_{\text{sat,PA}} \cdot -\underline{\boldsymbol{\rho}}}{\| - \underline{\boldsymbol{r}}_{\text{sat,PA}}\| \; \| - \underline{\boldsymbol{\rho}}\|}\right) \tag{3.111}$$

The antenna gains can then be calculated using Figures 3.21 and 3.22 using the calculated off-boresight angles. After calculating the antenna gains, a final value for the C/N_0 can be found. The C/N_0 directly influences the uncertainty in the two-way ranging measurement, $\sigma_{\text{p,tot}}$, which is a combination of the errors resulting from ephemeris and non-ephemeris data.

$$\sigma^2_{p,\text{tot}} = 4\sigma^2_{p,\text{eph}} + \sigma^2_{p,\text{non}-\text{eph}} \tag{3.112}$$

The ephemeris error is multiplied by a factor of four since the error is reflected both to and from the satellite and gets squared in the equation. The non-ephemeris data is composed of error due to the receiver's delay lock loop, lunar multipath, and relative timing delay in signal transmission as shown below:

$$\sigma^2_{p,\text{non}-\text{eph}} = \sigma^2_{p,\text{DLL}} + \sigma^2_{p,\text{rel}} + \sigma^2_{p,\text{mul}} \tag{3.113}$$

The lunar multipath error, $\sigma_{p,\mathrm{mul}}$, and residual relay delay, $\sigma_{p,\mathrm{rel}}$, are defined as 0.2 m and 0.31 m, respectively in this study. The DLL error is a function of the C/N_0 values as shown in Eq. 3.114. Computing the ephemeris and non-ephemeris error provides the total uncertainty in the two-way ranging measurements, which influences the EKF's measurement model equation.

$$\sigma^2_{p,\mathrm{DLL}} = (cT_c)^2 \left(\frac{B_{\mathrm{DLL}}(1 - 0.5B_{\mathrm{DLL}}T)}{2C/N_0}\right)\left(\frac{1}{T_cB_{fe}} + \frac{T_cB_{fe}}{\pi - 1}\left(d - \frac{1}{T_cB_{fe}}\right)^2\right)\left(1 + \frac{2}{TC/N_0(2-d)}\right) \tag{3.114}$$

Once the values above are calculated, Cortinovis, Mina, and Gao solve for the local rover state using a combined weighted batch filter and EKF algorithm with two-way ranging measurements. Their batch-to-EKF algorithm is used in this study to provide the reference position needed in Section 3.10. The procedure for executing their filter is clearly described in [46].

3.10 Estimation with Correlated Gravity Biases

This section aims to use CelNav measurements obtained from the MEKF algorithm, along with the accurate reference position provided by the Lunar Pathfinder satellite, to determine position estimates and identify biases in the LGM2011 gravity model resulting from inaccuracies in gravity modeling.

3.10.1 Estimation of Positions

Using the methods described in Sections 3.7 and 3.9, biased position estimates obtained from CelNav and highly accurate reference positions can be used to perform a journey across the lunar surface, composed of n positions $\underline{\boldsymbol{r}}_1, \ldots, \underline{\boldsymbol{r}}_n$, where each $\boldsymbol{r}_i = [\lambda_i \ \phi_i]^T$. The

CelNav measurements assume that the LGM2011 gravity model perfectly represents the true gravity field, which is not the case in real-world scenarios. The discrepancy between the true gravity values and those predicted by LGM2011 introduces errors in navigation, resulting in biased CelNav measurements within the region where the model is applied. There are two measurement methods considered; a biased measurement (CelNav) and an unbiased reference measurement (e.g. from Lunar Pathfinder). The unbiased measurements are modeled by

$$\bar{\underline{\boldsymbol{z}}}_m = \underline{\boldsymbol{r}}_{i_m} + \bar{\underline{\boldsymbol{\nu}}}_m, \qquad \bar{\underline{\boldsymbol{\nu}}}_m \sim \mathcal{N}(\mathbf{0}, \bar{R}_m) \tag{3.115}$$

(note the overbars) which only occur at some steps i, and biased (CelNav) measurements are taken by the ST/accelerometer at each position i according to

$$\underline{\boldsymbol{z}}_i = \underline{\boldsymbol{r}}_i + \underline{\boldsymbol{b}}_i + \underline{\boldsymbol{\nu}}_i, \qquad \underline{\boldsymbol{\nu}}_i \sim \mathcal{N}(\underline{\mathbf{0}}, R_i) \tag{3.116}$$

where $\underline{\boldsymbol{b}}_{\boldsymbol{i}}$ is the unknown bias in the CelNav longitude/latitude estimates output due to unmodeled gravity errors from the gravity field at position $\underline{\boldsymbol{r}}_{\boldsymbol{i}}$ and not the accelerometer bias seen above. The biases, as a function of position, are modeled as a Gaussian random field with the following statistics

$$\mathbb{E}\{\underline{\boldsymbol{b}}_i\} = \underline{\mathbf{0}}, \qquad \mathbb{E}\{\underline{\boldsymbol{b}}_i \underline{\boldsymbol{b}}_j^T\} = I \cdot \sigma_b^2 \, c(d_{ij}) \tag{3.117}$$

where σ_b^2 is the total bias uncertainty, d_{ij} is the distance between positions i and j, and $c(d_{ij})$ is the gravity deflection correlation function with respect to the distance d_{ij}. The rover or astronaut is capable of moving a given distance $\Delta\underline{\boldsymbol{r}}$ with some limited accuracy, such that

$$\underline{\boldsymbol{r}}_i = \underline{\boldsymbol{r}}_{i-1} + \Delta\underline{\boldsymbol{r}}_i + \underline{\boldsymbol{\eta}}, \qquad \underline{\boldsymbol{\eta}} \sim \mathcal{N}(\underline{\mathbf{0}}, Q_i) \tag{3.118}$$

where it is possible that Q_i is a function of the distance d_{ij}.

Construct the following matrices (using $\otimes$ for the Kronecker product):

$$R = \begin{bmatrix} R_1 & & \\ & \ddots & \\ & & R_n \end{bmatrix} \tag{3.119}$$

$$B = \begin{bmatrix} c(d_{11}) & \cdots & c(d_{1n}) \\ \vdots & \ddots & \vdots \\ c(d_{n1}) & \cdots & c(d_{nn}) \end{bmatrix} \otimes \sigma_b^2 I_{2\times 2} \tag{3.120}$$

$$D = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \otimes I_{2\times 2} \tag{3.121}$$

$$Q = \begin{bmatrix} Q_2(d_{12}) & & \\ & \ddots & \\ & & Q_n(d_{n-1,n}) \end{bmatrix} \tag{3.122}$$

Assume the measurement vector, reference vector, and step vectors are

$$\underline{\boldsymbol{z}} = \begin{bmatrix} \underline{\boldsymbol{z}}_1 \\ \vdots \\ \underline{\boldsymbol{z}}_n \end{bmatrix} \tag{3.123}$$

$$\bar{\underline{\boldsymbol{z}}} = \sum_m \underline{\boldsymbol{e}}_{i_m} \otimes \bar{\underline{\boldsymbol{z}}}_m \tag{3.124}$$

$$\underline{\boldsymbol{d}} = \begin{bmatrix} \Delta\underline{\boldsymbol{r}}_2' \\ \vdots \\ \Delta\underline{\boldsymbol{r}}_n' \end{bmatrix} \tag{3.125}$$

where $\underline{\boldsymbol{e}}_i$ is the n-length vector of all zeros with a single one in position i and $\Delta\underline{\boldsymbol{r}}_i'$ represents the intended step taken to get to position $\underline{\boldsymbol{r}}_i$. Note that other initial position guesses or other position information (for example landmark sightings) can be accommodated as if they were reference measurements. The least squares cost function for simultaneous estimation of all route biases and positions is

$$J = \frac{1}{2}\Bigg[\underline{\boldsymbol{b}}^T B^{-1}\underline{\boldsymbol{b}} + (D\underline{\boldsymbol{r}} - \underline{\boldsymbol{d}})^T Q^{-1}(D\underline{\boldsymbol{r}} - \underline{\boldsymbol{d}}) + (\underline{\boldsymbol{r}} + \underline{\boldsymbol{b}} - \underline{\boldsymbol{z}})^T R^{-1}(\underline{\boldsymbol{r}} + \underline{\boldsymbol{b}} - \underline{\boldsymbol{z}}) \\ + \sum_m (\underline{\boldsymbol{r}}_{i_m} - \bar{\underline{\boldsymbol{z}}}_m)^T \bar{R}_m^{-1}(\underline{\boldsymbol{r}}_{i_m} - \bar{\underline{\boldsymbol{z}}}_m)\Bigg]$$

and the derivatives of this with respect to $\underline{\boldsymbol{r}}$ and $\underline{\boldsymbol{b}}$ are

$$\frac{\partial J}{\partial \underline{\boldsymbol{r}}} = D^T Q^{-1}(D\underline{\boldsymbol{r}} - \underline{\boldsymbol{d}}) + R^{-1}(\underline{\boldsymbol{r}} + \underline{\boldsymbol{b}} - \underline{\boldsymbol{z}}) + \sum_m \underline{\boldsymbol{e}}_{i_m} \otimes \bar{R}_m^{-1}(\underline{\boldsymbol{r}}_{i_m} - \bar{\underline{\boldsymbol{z}}}_m) \tag{3.126}$$

$$\frac{\partial J}{\partial \underline{\boldsymbol{b}}} = B^{-1}\underline{\boldsymbol{b}} + R^{-1}(\underline{\boldsymbol{r}} + \underline{\boldsymbol{b}} - \underline{\boldsymbol{z}}) \tag{3.127}$$

and setting each of these to zero at the least squares estimate $\hat{\underline{\boldsymbol{r}}}$, $\hat{\underline{\boldsymbol{b}}}$ yields the following equalities:

$$\left(D^{\mathsf{T}} Q^{-1} D + R^{-1} + \sum_m E_{i_m i_m} \otimes \bar{R}_m^{-1}\right)\hat{\underline{\boldsymbol{r}}} + R^{-1}\hat{\underline{\boldsymbol{b}}} \\ = R^{-1}\underline{\boldsymbol{z}} + D^T Q^{-1}\underline{\boldsymbol{d}} + \sum_m \underline{\boldsymbol{e}}_{i_m} \otimes \bar{R}_m^{-1}\bar{\underline{\boldsymbol{z}}}_m \tag{3.128}$$

$$R^{-1}\hat{\underline{\boldsymbol{r}}} + (B^{-1} + R^{-1})\hat{\underline{\boldsymbol{b}}} = R^{-1}\underline{\boldsymbol{z}} \tag{3.129}$$

where E_{ij} is a matrix of all zeros with a single one in position i, j. Subtracting the second from the first gives

$$\left(D^{\mathsf{T}}Q^{-1}D + \sum_{m} E_{i_m i_m} \otimes \bar{R}_m^{-1}\right)\underline{\hat{\boldsymbol{r}}} - B^{-1}\underline{\hat{\boldsymbol{b}}} = D^{\mathsf{T}}Q^{-1}\underline{\boldsymbol{d}} + \sum_{m} \underline{\boldsymbol{e}}_{i_m} \otimes \bar{R}_m^{-1}\,\underline{\bar{\boldsymbol{z}}}_m \tag{3.130}$$

which can be put entirely in terms of $\underline{\boldsymbol{r}}$ by first solving the second equation for $\underline{\boldsymbol{b}}$ to get

$$\underline{\hat{\boldsymbol{b}}} = B(R+B)^{-1}(\underline{\boldsymbol{z}} - \underline{\hat{\boldsymbol{r}}}) \tag{3.131}$$

and then substituting into the difference:

$$\begin{aligned}\left(D^{\mathsf{T}}Q^{-1}D + \sum_{m} E_{i_m i_m} \otimes \bar{R}_m^{-1} + (R+B)^{-1}\right)\underline{\hat{\boldsymbol{r}}}& \\ = (R+B)^{-1}\underline{\boldsymbol{z}} + D^{T}Q^{-1}\underline{\boldsymbol{d}} + \sum_{m} \underline{\boldsymbol{e}}_{i_m} \otimes \bar{R}_m^{-1}\underline{\bar{\boldsymbol{z}}}_m&\end{aligned} \tag{3.132}$$

This can be solved for $\underline{\hat{\boldsymbol{r}}}$ and simplified to give the final equation for the $\underline{\boldsymbol{r}}$ estimate (presented along with a simplified bias estimate equation):

$$\underline{\hat{\boldsymbol{r}}} = (I + YZ)^{-1}\left(\underline{\boldsymbol{z}} + Y\underline{\boldsymbol{\zeta}}\right) \tag{3.133}$$

$$\underline{\hat{\boldsymbol{b}}} = BY^{-1}\left(\underline{\boldsymbol{z}} - \underline{\hat{\boldsymbol{r}}}\right) \tag{3.134}$$

where

$$Y = R + B \tag{3.135}$$

$$Z = D^{T}Q^{-1}D + \sum_{m} E_{i_m i_m} \otimes \bar{R}_m^{-1} \tag{3.136}$$

$$\underline{\boldsymbol{\zeta}} = D^{T}Q^{-1}\underline{\boldsymbol{d}} + \sum_{m} \underline{\boldsymbol{e}}_{\boldsymbol{i_m}} \otimes \bar{R}_m^{-1}\underline{\bar{\boldsymbol{z}}}_{\boldsymbol{m}} \tag{3.137}$$

Note that to actually use this equation for estimation, the Q_i^{-1} and B matrices are needed, which both have components that vary with the distances d_{ij} between positions. To solve the equation the first time, we need to use some estimate of the distances between positions (e.g. using an initial guess at the positions). For improved accuracy, the matrices B and Q_i can be reevaluated using the newly-estimated positions, and the estimate can be iteratively recomputed. For full accuracy, this process should be repeated until convergence.

3.10.2 Covariances

Using this estimator, it is found that

$$\underline{\boldsymbol{\mu}}_{\hat{\boldsymbol{r}}} = \mathbb{E}\{\underline{\hat{\boldsymbol{r}}}\} = (I+YZ)^{-1}\left[\underline{\boldsymbol{r}} + Y\left(D^{-1}Q^{-1}\underline{\boldsymbol{d}} + \sum_m \underline{\boldsymbol{e}}_{\boldsymbol{i_m}} \otimes \bar{R}_m^{-1}\underline{\boldsymbol{r}}_{\boldsymbol{i_m}}\right)\right] \tag{3.138}$$

$$\underline{\hat{\boldsymbol{r}}} - \underline{\boldsymbol{\mu}}_{\hat{\boldsymbol{r}}} = (I+YZ)^{-1}\left[\underline{\boldsymbol{b}} + \underline{\boldsymbol{\nu}} - Y\left(\sum_m \underline{\boldsymbol{e}}_{\boldsymbol{i_m}} \otimes \bar{R}_m^{-1}\underline{\bar{\boldsymbol{\nu}}}_{\boldsymbol{m}}\right)\right] \tag{3.139}$$

which gives an expression for the covariance in the position estimates:

$$P_{rr} = (I+YZ)^{-1}\left[Y + Y\left(\sum_m E_{i_m i_m} \otimes \bar{R}_m^{-1}\right)Y\right](I+YZ)^{-\mathsf{T}} \tag{3.140}$$

Note that if only the covariance of the final position estimate is desired, taking the complete matrix inverses can be avoided in favor of using numerical linear equation solutions like

$$(\underline{\boldsymbol{e}}_{\boldsymbol{n}}^T \otimes I)/(I+YZ) \tag{3.141}$$

on each side of the matrix to extract the relevant block.

The bias estimate has expected values

$$\underline{\boldsymbol{\mu}}_{\hat{\boldsymbol{b}}} = \mathbb{E}\{\underline{\hat{\boldsymbol{b}}}\} = BY^{-1}\left(\underline{\boldsymbol{r}} - \underline{\boldsymbol{\mu}}_{\hat{\boldsymbol{r}}}\right) \tag{3.142}$$

$$\underline{\hat{\boldsymbol{b}}} - \underline{\boldsymbol{\mu}}_{\hat{\boldsymbol{b}}} = BY^{-1}\left[\underline{\boldsymbol{b}} + \underline{\boldsymbol{\nu}} - (\underline{\hat{\boldsymbol{r}}} - \underline{\boldsymbol{\mu}}_{\hat{\boldsymbol{r}}})\right] \tag{3.143}$$

which gives covariance

$$P_{bb} = BY^{-1}\left(Y + P_{rr}\right)Y^{-1}B \tag{3.144}$$

The covariance between bias and position can be found similarly, but is not needed for this work. A diagram showing a theoretical mission on the surface and how the above process would unfold is shown in Figure 3.24. The initial $(\lambda_{\text{ref}}, \phi_{\text{ref}})$ position comes from the external satellite reference, the $\underline{\boldsymbol{r}}_{\boldsymbol{i}}$ values are the biased CelNav position estimates at each waypoint, and the d_{ij} values are the distances between each waypoint.

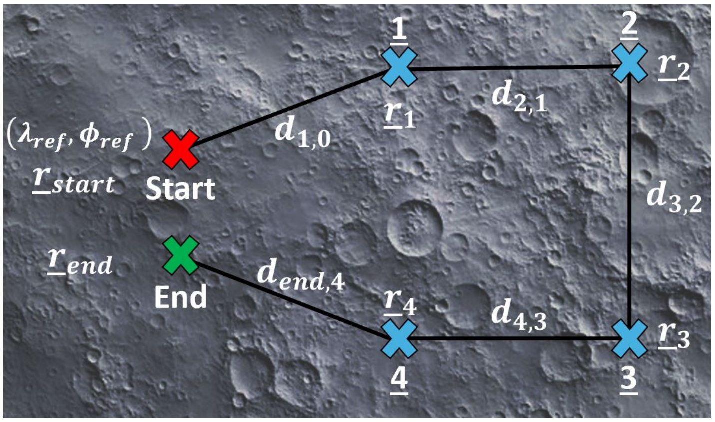


Figure 3.24: Diagram showing hypothetical lunar mission of an astronaut/rover starting from an initial position and moving to multiple waypoints, collecting CelNav position estimates at each location. An accurate satellite reference position is also shown to be calculated at the starting point in this case, but the reference could be provided at any point during the mission.

Chapter 4

Results

This chapter presents various Monte Carlo and test case simulations used to evaluate the accuracy and behavior of the algorithms introduced in Chapter 3. The analysis begins with an examination of the MEKF algorithm executed at multiple locations across the lunar surface, comparing results from pure rotation and libration cases. Monte Carlo simulations are then conducted for different rotation values used in the accelerometer bias estimation process described in Section 3.8 to identify which value leads to the most accurate final results. The impact of the MEKF's initial guess on position estimates is also examined to determine any potential divergence in the MEKF algorithm. A standard Monte Carlo simulation at a single location is included to demonstrate how measurement randomness affects the final position estimates. Another Monte Carlo study is conducted to analyze how accelerometer accuracy (amount of noise in the measurements) influences the MEKF's final position and bias estimates. Finally, example missions using both Earth and the Moon gravity are presented to validate the correlated gravity algorithm described in Section 3.10.

4.1 Simulating Gravity Measurements

The MEKF algorithm requires simulated gravity measurements. Like all sensors, the accelerometer's gravity measurements contain random noise, leading to uncertainty. This randomness depends on the type of accelerometer and its specified accuracy. In this study, the simulated gravity is calculated using the following equation

$$\underline{\boldsymbol{g}}_{\text{simulated}} = \underline{\boldsymbol{g}}_{\text{true}} + \underline{\boldsymbol{g}}_{\text{random}} + \underline{\boldsymbol{b}}_{a} \tag{4.1}$$

where $\underline{\boldsymbol{g}}_{\text{true}}$ is the true gravity, $\underline{\boldsymbol{g}}_{\text{random}}$ is the noise associated with the uncertainty in the accelerometer, and $\underline{\boldsymbol{b}}_{a}$ is the accelerometer bias discussed in Section 3.8. The true gravity is found using the LGM2011 model at the (λ, ϕ) measurement position. The noise vector is modeled using the following normal distributions

$$\underline{\boldsymbol{g}}_{\text{random}} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{accel}}^{2} I_{3\times 3}) \tag{4.2}$$

where $\sigma_{\text{accel}} = 1 \times 10^{-3}\ \text{m/s}^2$. This uncertainty is specific to the accelerometer and is typically specified by the device manufacturer. How this value affects the accuracy of the MEKF algorithm is discussed in Section 4.6. The true random accelerometer bias, $\underline{\boldsymbol{b}}_{a}$, is randomly generated using

$$\underline{\boldsymbol{b}}_{a} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{bias}}^{2} I_{3\times 3}) \tag{4.3}$$

where $\sigma_{\text{bias}} = 1 \times 10^{-3}\ \text{m/s}^2$ represents the general magnitude of the true accelerometer bias. The exact value of the bias is not important, but the magnitude should be small. The bias is regenerated for each run, but is assumed constant within a run. The simulated gravity definitions above use values from the known LGM2011 gravity field and common noise models to provide realistic gravity measurements the system would experience on the lunar surface.

4.2 Pure Rotation v. Precession Monte Carlo

This section discusses a Monte Carlo simulation that ran the MEKF algorithm at 5000 random positions across the lunar surface. Two full MEKF runs were conducted at each location: one using pure rotation propagation and the other using precession-based covariance propagation. The simulation conditions for both runs were kept identical, with the only difference being the covariance propagation method. This setup allowed the two methods to be compared based on the accuracy of their final estimates. The Monte Carlo process began by selecting a random true position $(\lambda_{\text{true}}, \phi_{\text{true}})$ from the LGM2011 position data. The MEKF algorithm then selected an initial guess, (λ_0, ϕ_0), following the normal distribution described in Eqs. 3.60 and 3.61, based on the true reference position. The initial guesses for attitude and accelerometer bias were set to zero, consistent with the MEKF algorithm definition. The initial covariance estimate for the attitude error was calculated using an initial run of the QUEST algorithm from the true reference position. The initial covariance estimate for (λ_0, ϕ_0) corresponds to the uncertainty of the normal distributions in Eqs. 3.60 and 3.61, as shown in the following equation.

$$P_{\lambda_0,\phi_0} = \begin{bmatrix} \sigma_\lambda^2 & 0 \\ 0 & \sigma_\phi \end{bmatrix} = \begin{bmatrix} 0.04^{\circ 2} & 0 \\ 0 & 0.04^{\circ 2} \end{bmatrix} \tag{4.4}$$

where $\sigma_\lambda = 0.2^\circ$ and $\sigma_\phi = 0.2^\circ$. Since the LGM2011 model states that a spacing of 0.05° in λ and ϕ corresponds to a resolution of 1.5 km, this standard deviation indicates that the initial guess is within an approximate 6 km radius of the true position, as shown in Figure 4.1.

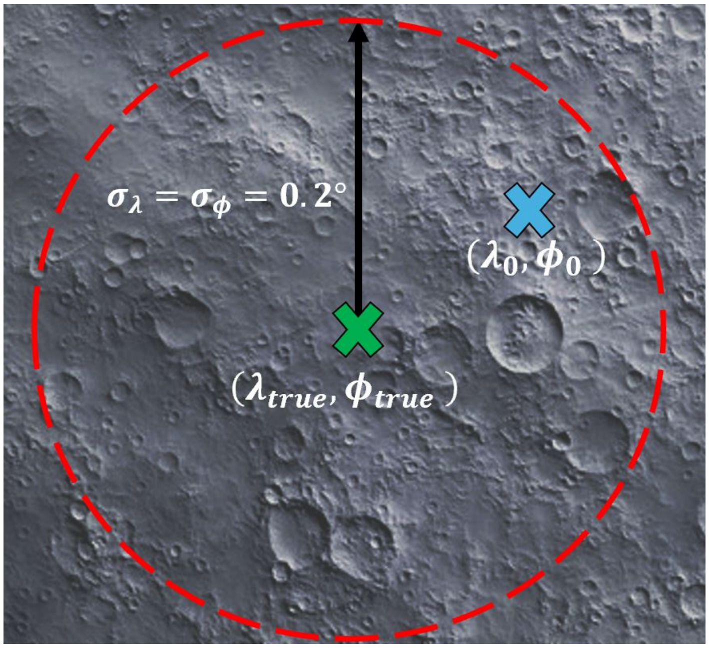

$\sigma_\lambda = \sigma_\phi = 0.2°$
(λ_0, ϕ_0)
$(\lambda_{true}, \phi_{true})$

Figure 4.1: Diagram showing initial (λ_0, ϕ_0) guess falling within circle surrounding true $(\lambda_{\text{true}}, \phi_{\text{true}})$ position. This circle is defined using the uncertainty in the initial guesses, σ_λ and σ_ϕ, equating to approximately 6 km.

Since the initial accelerometer bias guess is set to zero, the associated uncertainty is high. This Monte Carlo simulation, along with the others presented in this chapter, used an initial bias uncertainty of $\sigma_{\underline{\boldsymbol{b}}_a,0} = 1 \times 10^{-3}\,\text{m/s}^2$, leading to

$$P_{\underline{\boldsymbol{b}}_{\boldsymbol{a},\mathbf{0}}} = \sigma^2_{\underline{\boldsymbol{b}}_a,0} I_{3\times3} = \begin{bmatrix} 1\times10^{-6} & 0 & 0 \\ 0 & 1\times10^{-6} & 0 \\ 0 & 0 & 1\times10^{-6} \end{bmatrix} \text{m}^2/\text{s}^4 \tag{4.5}$$

The MEKF simulations were run at each position using the initial guess parameters defined above and summarized in Table 4.1.

Table 4.1: Initial guess parameters used for the pure rotation v. libration Monte Carlo at 5000 locations on the lunar surface. These parameters define the uncertainty in the initial position and accelerometer bias estimates.

Initial Guess Parameter	Value
λ_0	$\mathcal{N}(\lambda_{\text{true}}, \sigma_\lambda^2)$
ϕ_0	$\mathcal{N}(\phi_{\text{true}}, \sigma_\phi^2)$
σ_λ	0.2°
σ_ϕ	0.2°
$\sigma_{\underline{\boldsymbol{b}}_a,0}$	$1 \times 10^{-3}\,\text{m/s}^2$

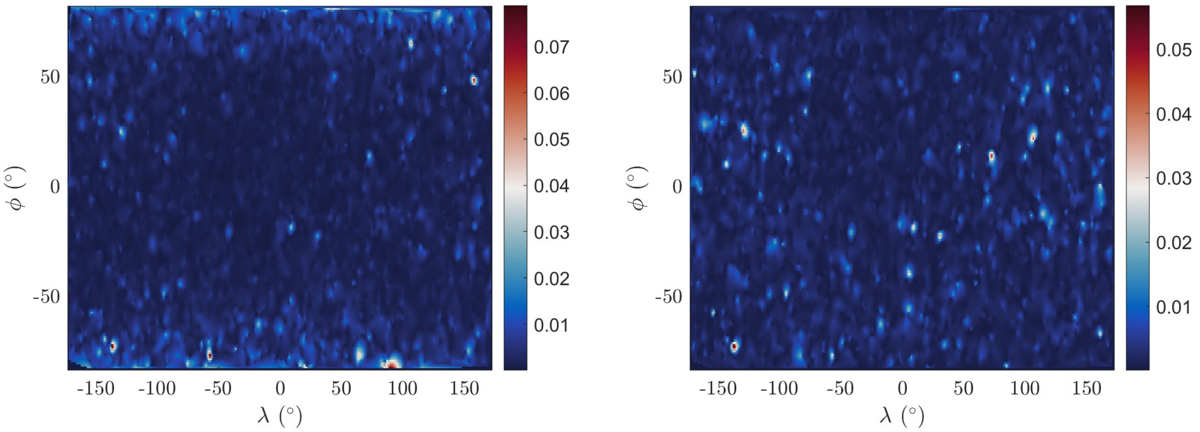
To ensure that randomness in the simulated gravity vectors did not lead to falsely identifying one method as better than the other, a predefined array containing simulated gravity at all time steps was created. This array was then used for both algorithms at all time steps, exposing both methods to the same random gravity. This eliminates potential discrepancies in the results due to one method experiencing worse gravity measurements, which would lead to more inaccurate position estimates. The complete process used to find the position estimates is summarized below:

1. Randomly select $(\lambda_{\text{true}}, \phi_{\text{true}})$ from LGM2011 position data.
2. Define (λ_0, ϕ_0) using normal distribution.
3. Set initial attitude error and accelerometer bias to be zero, completing the initial $\underline{\boldsymbol{x}}_0$ state vector.
4. Use uncertainty parameters in Table 4.1 to complete initial P_0 state covariance.
5. Generate random gravity measurements at all time steps to allow both methods to use the same values.
6. Run two separate MEKF algorithms at the true position to find final position and

accelerometer bias estimates based on covariance propagation method: one for pure rotation and the other for precession.

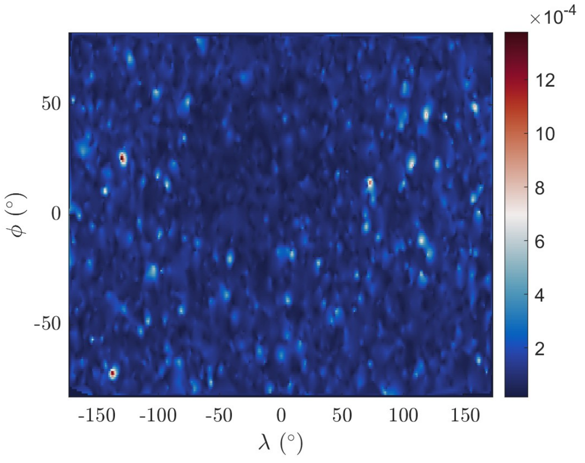
7. Repeat steps 1–6 for 5000 points across the surface.

The longitude, latitude, and accelerometer bias errors for pure rotation covariance propagation are shown in Figure 4.2. It is clear from Figure 4.2 that there are certain regions of the Moon where the position and bias error are larger. These regions primarily correspond to topographic features such as hills or craters. The LGM2011 gravity and deflection values are shown in Figure 4.3. Craters and other features are easily identifiable in these figures based on their higher surface gravity and deflection values. These locations can be directly traced to areas in Figures 4.2a–4.2c where the error is higher. These regions are where the gravity vector points in the same direction; therefore, shifts in position would not appreciably change the measurements/estimate, leading to a high Geometric Dilution of Precision (GDOP).



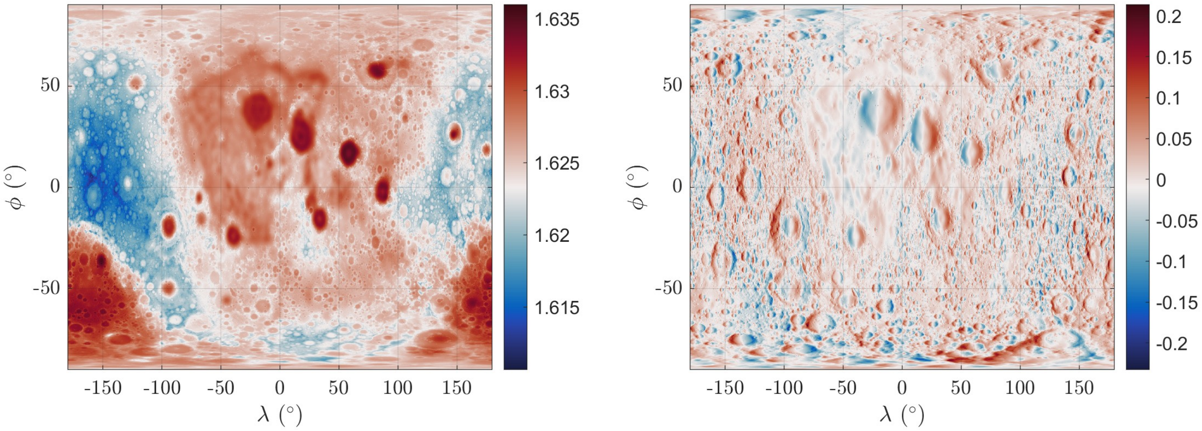
(a) $|\lambda_{\text{est}} - \lambda_{\text{true}}|$ estimate error (°)

(b) $|\phi_{\text{est}} - \phi_{\text{true}}|$ estimate error (°)

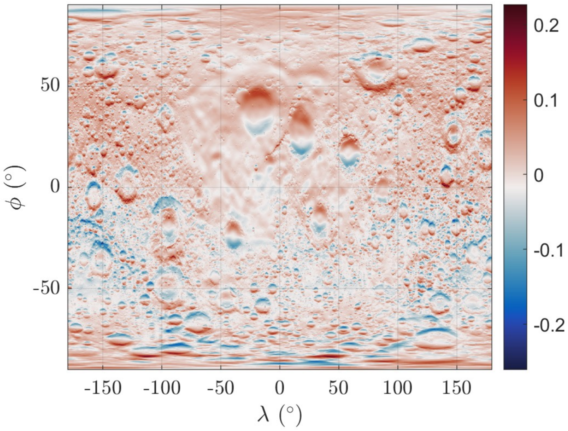


(c) $\|\underline{\boldsymbol{b}}_{a,\text{est}} - \underline{\boldsymbol{b}}_{a,\text{true}}\|$ bias error (m/s^2)

Figure 4.2: Heat map showing the $|\lambda_{\text{est}} - \lambda_{\text{true}}|$, $|\phi_{\text{est}} - \phi_{\text{true}}|$, and $\|\underline{\boldsymbol{b}}_{a,\text{est}} - \underline{\boldsymbol{b}}_{a,\text{true}}\|$ estimate error across the lunar surface using pure rotation covariance propagation



(a) Surface gravity across lunar surface (m/s^2) (b) EW gravity deflection across lunar surface °



(c) NS gravity deflection across lunar surface °

Figure 4.3: LGM2011 surface gravity magnitude, and EW/NS deflections across the lunar surface [14]. This new representation of the gravity is presented for easy comparison with the longitude, latitude, and accelerometer bias heatmaps containing the MEKF error across the lunar surface, making it easier to identify any trends between topographic features and increased error.

The GDOP at each location on the surface provides an idea of which regions will produce more accurate position results when including the LGM2011 gravity data. The GDOP is calculated using

$$G = \frac{1}{r_{\text{moon}}} \begin{bmatrix} \frac{\partial \boldsymbol{g}_{LGM}}{\partial \lambda} \left(\frac{1}{\phi}\right) & \frac{\partial \boldsymbol{g}_{LGM}}{\partial \phi} \left(\frac{1}{\lambda}\right) \end{bmatrix} \tag{4.6}$$

$$Q = \left(G^T G\right)^{-1} \tag{4.7}$$

$$\text{GDOP} = \sqrt{\text{tr}(Q)} \tag{4.8}$$

The GDOP at every (λ, ϕ) position provided by the LGM2011 gravity model is shown in Figure 4.4. The GDOP is shown to be higher near the poles compared to the Moon's equator. It is also clear that the regions directly next to crater have higher GDOP's, matching the errors seen so far in the MEKF position estimates.

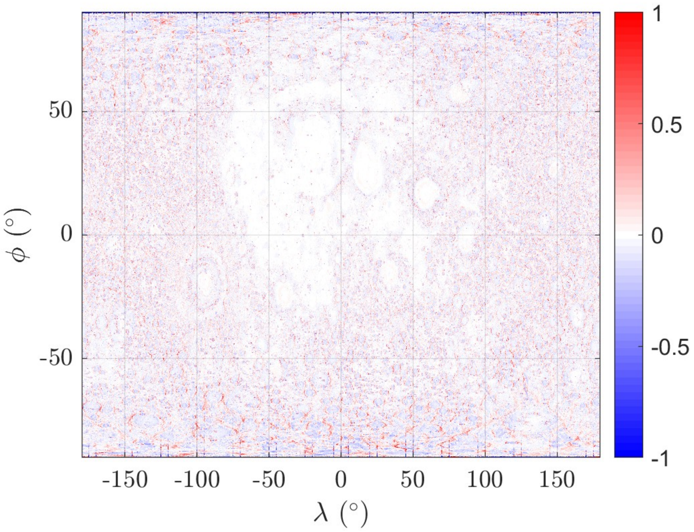
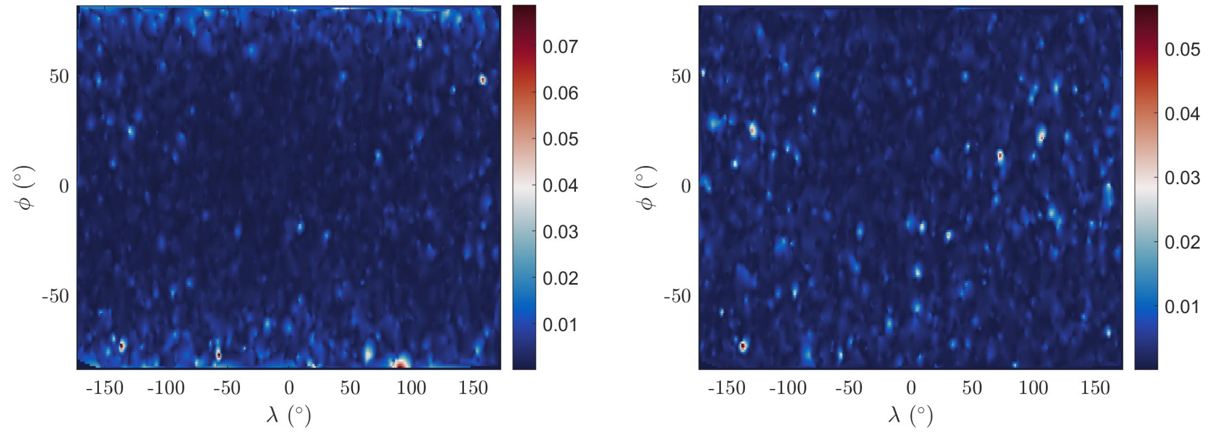
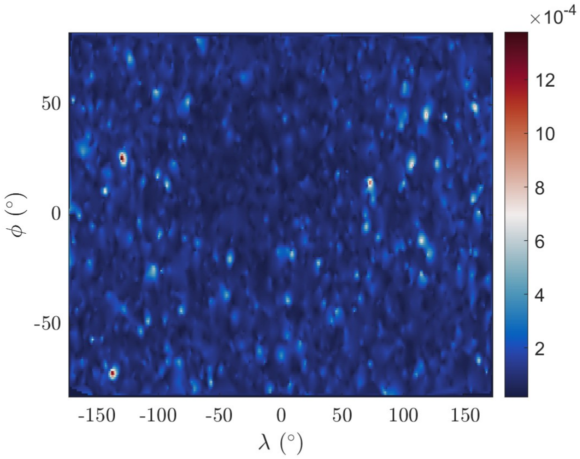


Figure 4.4: Heat map showing the GDOP across the lunar surface using the LGM2011 gravity model.

The longitude, latitude, and accelerometer bias errors incorporating the Moon's precession are shown in Figure 4.5.



(a) $|\lambda_{\text{est}} - \lambda_{\text{true}}|$ estimate error (°) (b) $|\phi_{\text{est}} - \phi_{\text{true}}|$ estimate error (°)



(c) $\|\underline{\boldsymbol{b}}_{a,\text{est}} - \underline{\boldsymbol{b}}_{a,\text{true}}\|$ bias error (m/s^2)

Figure 4.5: Heat map showing the $|\lambda_{\text{est}} - \lambda_{\text{true}}|$, $|\phi_{\text{est}} - \phi_{\text{true}}|$, and $\|\underline{\boldsymbol{b}}_{a,\text{est}} - \underline{\boldsymbol{b}}_{a,\text{true}}\|$ estimate error across the lunar surface using lunar precession covariance propagation

The results in Figure 4.5 show higher errors near craters and other topographic features, matching the behavior seen in the pure rotation case. Based on graphs shown for both cases, it is hard to see a distinct difference in performance. Therefore, the errors can be put into terms of meters using the haversine formula shown below

$$d(\lambda_1, \lambda_2, \phi_1, \phi_2) = 2r_{moon}\left(\sqrt{\frac{1 - \cos(\Delta\phi) + \cos(\phi_1)\cos(\phi_2)(1 - \cos(\Delta\lambda))}{2}}\right) \tag{4.9}$$

where $\Delta\phi = \phi_2 - \phi_1$ and $\Delta\lambda = \lambda_2 - \lambda_1$. The haversine formula provides the distance between two points specified by longitude and latitude on a sphere, unlike the normal straight-line distance calculation. It offers a more accurate representation of the distance by accounting for the fact that the Moon can be approximated as a sphere. Tables 4.2 and 4.3 show the error statistics across the surface and show that the accuracy is improved when accounting for the Moon's precession.

Table 4.2: Statistical properties of the MEKF haversine position error for both pure rotation and libration covariance propagation.

Case	Mean (m)	Minimum (m)	Maximum (m)
Pure Rotation	101.806	0.8315	2261.36
Precession	101.805	0.8305	2261.28

Table 4.3: Statistical properties of the MEKF $\|\underline{\boldsymbol{b}}_{a,\text{est}} - \underline{\boldsymbol{b}}_{a,\text{true}}\|$ values for pure rotation and libration.

Case	Mean (m/s^2)	Minimum (m/s^2)	Maximum (m/s^2)
Pure Rotation	1.08494×10^{-4}	7.33415×10^{-6}	0.00176122
Precession	1.08493×10^{-4}	7.33254×10^{-6}	0.00176119

The Monte Carlo analysis presented in this section illustrates the variation in MEKF accuracy across the lunar surface. Two MEKF runs were performed at each location—one using pure rotation and the other using lunar precession for covariance propagation. Tables 4.2–4.3 show that including the Moon's precession in the covariance propagation step results in improved estimate accuracy, consistent with the findings discussed in Section 3.7.10. Therefore, the precession covariance propagation method is used in the remaining Monte Carlo studies.

4.3 Rotation Sequence Influence on Accelerometer Bias Estimation Accuracy

The procedure used to observe the inherent accelerometer dynamic bias, described in Section 3.8, involves a combination of two rotations: the first around the z axis and the second around the x axis. The procedure states that the accelerometer can be rotated by any arbitrary amount in these directions and still yield a reasonably accurate bias estimate. This section analyzes different combinations of rotation amounts and how they influence the final estimated accelerometer bias. The various rotations for the z and x axes are shown below

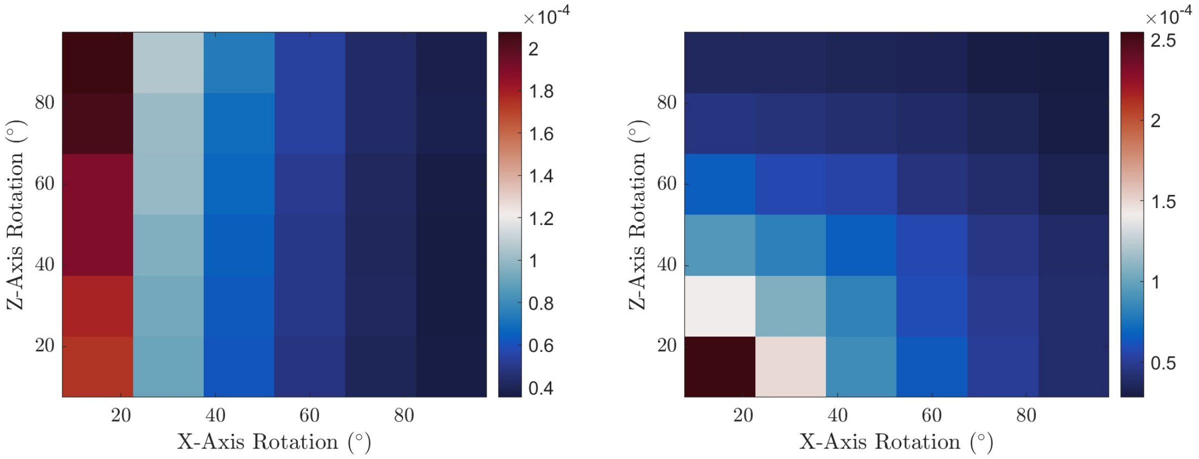
$$\theta_z \in \{15, 30, 45, 60, 75, 90\}^\circ \tag{4.10}$$

$$\theta_x \in \{15, 30, 45, 60, 75, 90\}^\circ \tag{4.11}$$

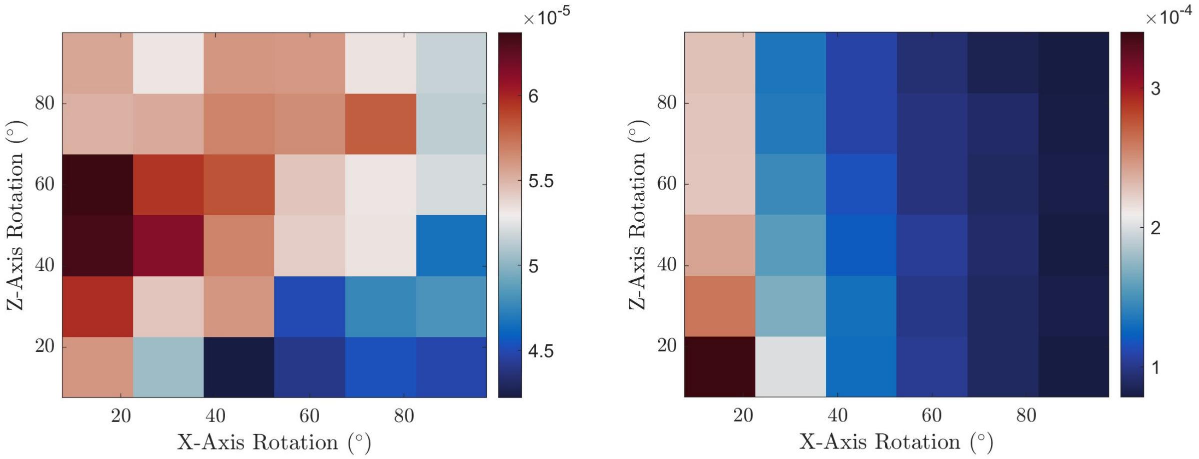
These angles provide a sufficient number of rotation combinations capable of revealing the behavior across different test cases. A 50-run Monte Carlo simulation was conducted to observe how the various rotation combinations influenced the accelerometer bias. Each Monte Carlo run began by selecting a z axis rotation angle, θ_z. Then, the simulation executed the MEKF algorithm with all θ_x angle combinations corresponding to the selected θ_z value. This process was repeated for each θ_z value until all z axis rotations had been tested. In total, 36 different rotation combinations were generated, fully representing the range of achievable system orientations.

Each run was performed at the Apollo Peak Ring (APR) position on the lunar surface, allowing the Monte Carlo simulation to include the randomness of the simulated gravity vectors at a fixed location. It is important to note that each rotation combination used the same simulated gravity sequence, ensuring that every test case was exposed to identical measurement conditions. The MEKF time step between each accelerometer and star tracker measurement was $\Delta t = 1\,\mathrm{s}$. Each orientation ran for 300 s, producing a total of 900 measurement updates during the entire MEKF (see Section 3.8 for a refresher on the rotation procedure and MEKF execution for each orientation).

The true bias for each run was generated using the distributions shown in Eq. 4.3. The component-wise and total bias errors for each combination were recorded and averaged over all Monte Carlo runs to evaluate the general performance of each configuration. The results are shown in Figure 4.6.



(a) $|b_{x,\text{est}} - b_{x,\text{true}}|$ x component bias error (m/s^2) (b) $|b_{y,\text{est}} - b_{y,\text{true}}|$ y component bias error (m/s^2)



(c) $|b_{z,\text{est}} - b_{z,\text{true}}|$ z component bias error (m/s^2) (d) $\|\underline{\mathbf{b}}_{\mathbf{a},\text{est}} - \underline{\mathbf{b}}_{\mathbf{a},\text{true}}\|$ total bias error in (m/s^2)

Figure 4.6: Heat map showing the MEKF's $|b_{x,\text{est}} - b_{x,\text{true}}|$, $|b_{y,\text{est}} - b_{y,\text{true}}|$, $|b_{z,\text{est}} - b_{z,\text{true}}|$, and $\|\underline{\mathbf{b}}_{\mathbf{a},\text{est}} - \underline{\mathbf{b}}_{\mathbf{a},\text{true}}\|$ accelerometer bias error in (m/s^2) at the APR for each rotation combination.

Figures 4.6a–4.6d show a distinct improvement in accuracy for the $(\theta_z, \theta_x) = (90°, 90°)$ rotation combination, validating the choice of this final orientation in the MEKF algorithm. The z component of the bias is also shown to be estimated more accurately than the x and y components. This is because the gravity measured by the accelerometer is primarily oriented along the z axis. As a result, greater variation is observed along this direction when the system is rotated about the body z and x axes. Figure 4.7 shows the accelerometer bias over time for each run of the MEKF algorithm, corresponding to 50 runs $\times$ 36 orientations $= 1800$ total MEKF runs.

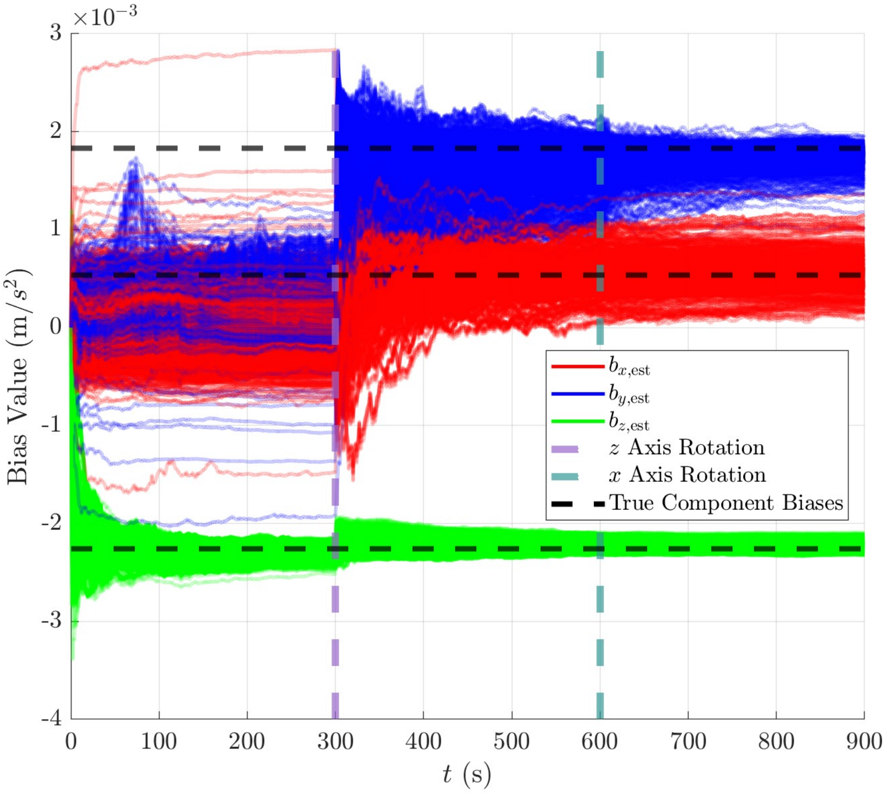
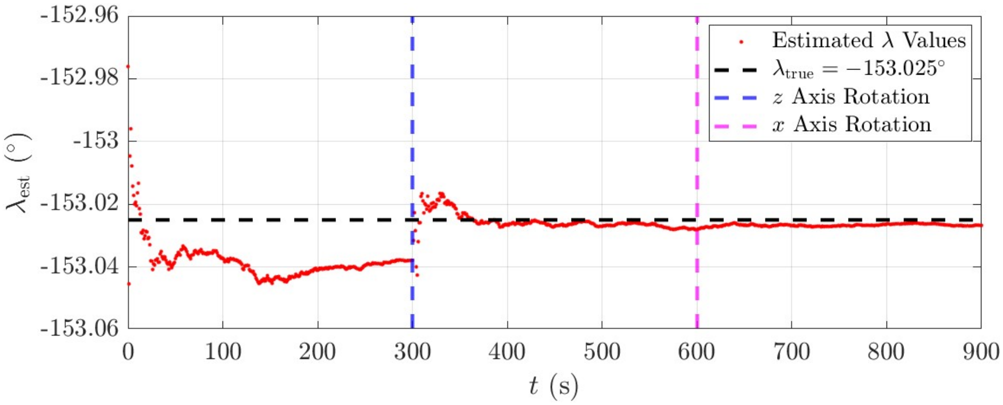


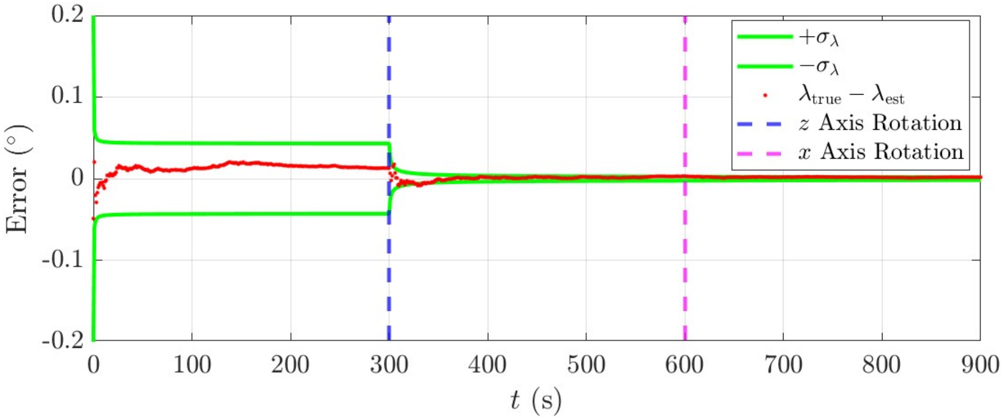
Figure 4.7: Accelerometer component bias estimates over time in (m/s^2) with respect to the true $\underline{\boldsymbol{b}}_a$ components for each rotation sequence combination. The order of the axis rotations is the z axis followed by the x axis.

Figure 4.7 shows the range of error values resulting from the different orientations experienced by the system during the Monte Carlo. The errors in the x and y components of the bias are clearly larger than the z component, as reflected in Figures 4.6a–4.6c. There is also a distinct change in the component estimates after the first z axis rotation at $t = 300\,\mathrm{s}$. The estimates before this time appear random with respect to the true value. The second rotation occurs at $t = 600\,\mathrm{s}$, showing a slight improvement. The majority of the increase in accuracy in the component estimates occurs after the first rotation. This is again due to the direction of the gravity vector being mostly along the body z axis. By rotating the accelerometer by a large amount around the z axis, the change in gravity measured along this axis becomes more pronounced, causing the estimate to quickly approach the true value.

The increase in accelerometer bias accuracy directly leads to improved accuracy in the position estimates. This follows logically, as knowing the accelerometer bias enhances the accuracy of the gravity measurement model in the MEKF algorithm. Figures 4.8 and 4.9 show the $(\lambda_{\text{est}}, \phi_{\text{est}})$ values and their uncertainties over time, using a $90°$ rotation about the z axis followed by a $90°$ rotation about the x axis.

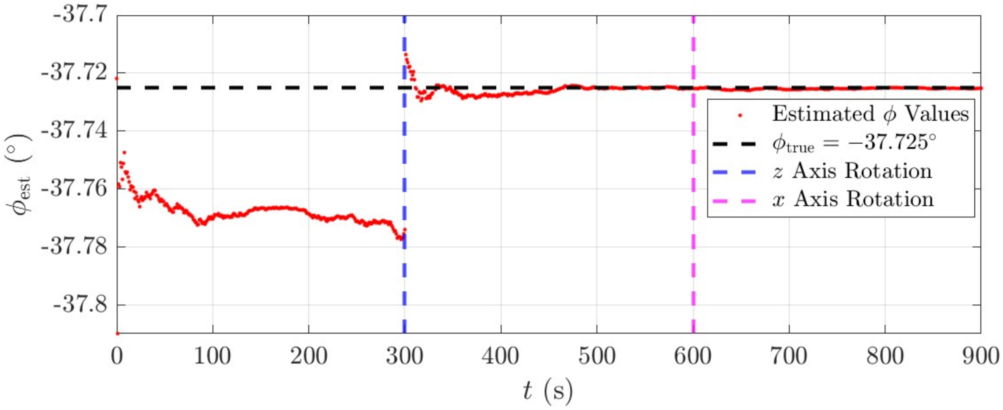


(a) λ_{est} time history for one full MEKF run

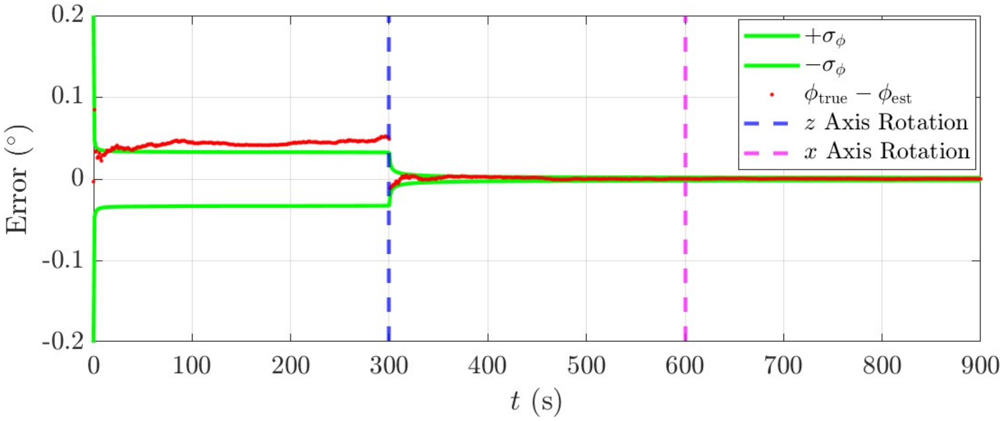


(b) λ_{est} error time history for one full MEKF run

Figure 4.8: MEKF λ_{est} estimate and error over time with respect to the true value, λ_{true}. These estimates were found using a 90° rotation about the z axis followed by a 90° rotation about the x axis.



(a) ϕ_{est} time history for one full MEKF run



(b) ϕ_{est} error time history for one full MEKF run

Figure 4.9: MEKF ϕ_{est} estimate and error over time with respect to the true value, ϕ_{true}. These estimates were found using a 90° rotation about the z axis followed by a 90° rotation about the x axis.

Figures 4.8 and 4.9 show a sudden increase in position estimate accuracy at $t = 300\,\mathrm{s}$, reflecting the improved accuracy in the accelerometer bias estimates from Figure 4.7. The initial state and covariance guesses follow the procedure outlined in Section 3.7. The σ_λ and σ_ϕ bounds in Figures 4.8b and 4.9b represent the MEKF uncertainty produced by the covariance matrix after each measurement update. For this example, the final estimates and their errors are shown in Table 4.4. This section examined different rotation combinations and how they impacted the final accelerometer bias estimate accuracy. The best rotation combination was found to be a 90° rotation about the z axis followed by a 90° rotation about the x axis, which is the process used in the MEKF for this study. The bias estimates showed a significant improvement after the initial rotation about the z axis, leading to the same increase in accuracy observed in the position estimates. By using this combination of rotations, the MEKF algorithm is able to produce an accurate final estimate for both the accelerometer bias and the lunar surface position. Although, if desired by the user, it can be inferred from the results presented in this section that one rotation will also lead to accurate accelerometer bias estimation.

Table 4.4: Final $(\lambda_{\text{est}}, \phi_{\text{est}})$ estimate and error values using a 90° rotation about the z axis followed by a 90° rotation about the x axis.

Variable	Value (°)
λ_{true}	-153.025
$\lambda_{\text{est},f}$	-153.023
$\lvert\lambda_{\text{true}} - \lambda_{\text{est},f}\rvert$	2.17833×10^{-3}
ϕ_{true}	-37.7250
$\phi_{\text{est},f}$	-37.7248
$\lvert\phi_{\text{true}} - \phi_{\text{est},f\rvert}$	2.20632×10^{-4}

4.4 Initial (λ_0, ϕ_0) Impact on MEKF Position Accuracy

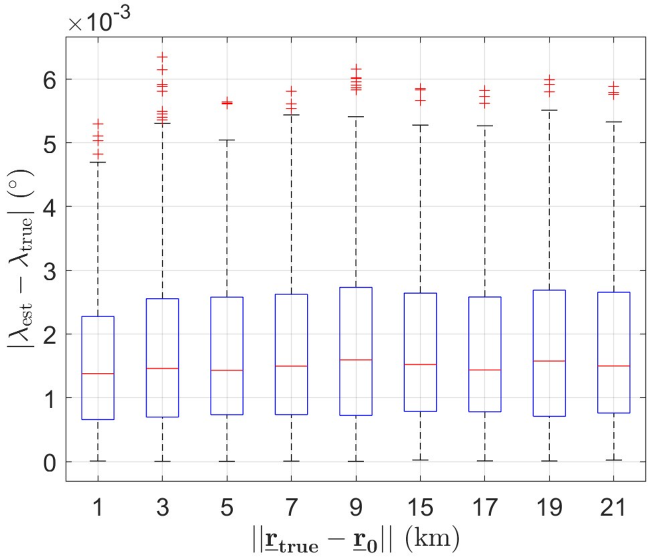
The MEKF requires an initial (λ_0, ϕ_0) position guess to begin the algorithm. This section analyzes how this guess—and its distance from the true position—influences the accuracy of the final position estimate. Several algorithms, such as the Newton-Raphson method, are highly dependent on the proximity of the initial guess to the true value. While the Newton-Raphson method is highly efficient when the initial guess is close to the true solution, it can diverge and lead to an inaccurate result if the guess is too far off. Therefore, this section evaluates the MEKF's sensitivity to the initial guess to determine whether divergence occurs when the guess is far from the true value.

A 150-run Monte Carlo simulation was performed to analyze this behavior, with the true reference position located at the APR. In each run, simulated gravity was generated using the procedure described in Section 4.1. The initial (λ_0, ϕ_0) guesses were sampled from normal distributions with standard deviations 0.025°[1 2 ...15] as shown below.

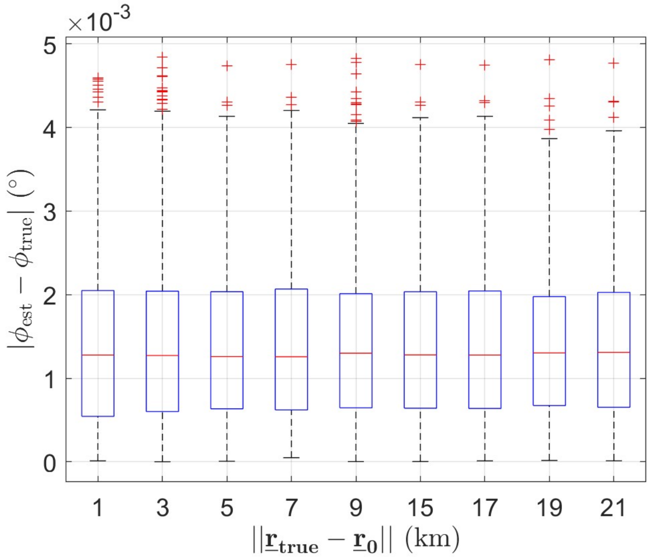
$$\lambda_0 = 0.025^\circ \times \begin{bmatrix} 1 & 2 & \dots & 15 \end{bmatrix} \times \mathcal{N}(0, 1) \tag{4.12}$$

$$\phi_0 = 0.025^\circ \times \begin{bmatrix} 1 & 2 & \dots & 15 \end{bmatrix} \times \mathcal{N}(0, 1) \tag{4.13}$$

Each run used the same simulated gravity measurements and iterated through each initial guess position. The simulated gravity was kept constant throughout each initial guess run to ensure consistency between test cases. The final position and accelerometer bias estimates for each initial position run were saved for post-processing. This process was repeated for each Monte Carlo run, with new random gravity measurements generated for each run to replicate measurement variability. The results from each run were then used to generate boxplots with sections ranging from the minimum to the maximum distance from the true position. Boxplots describing the position error as a function of initial guess error are shown in Figure 4.10.



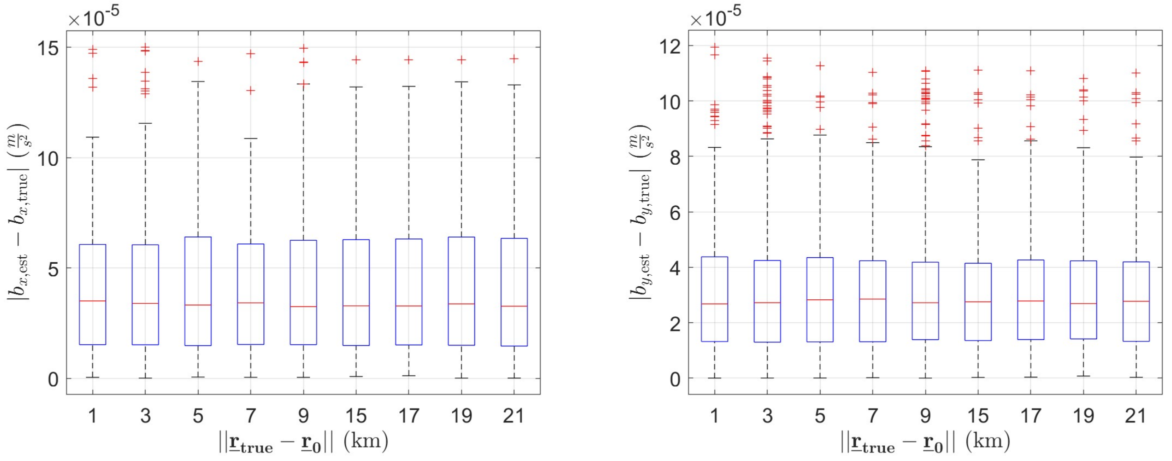
(a) $|\lambda_{\text{est}} - \lambda_{\text{true}}|$ as a function of $\|\underline{\boldsymbol{r}}_{\text{true}} - \underline{\boldsymbol{r}}_0\|$.



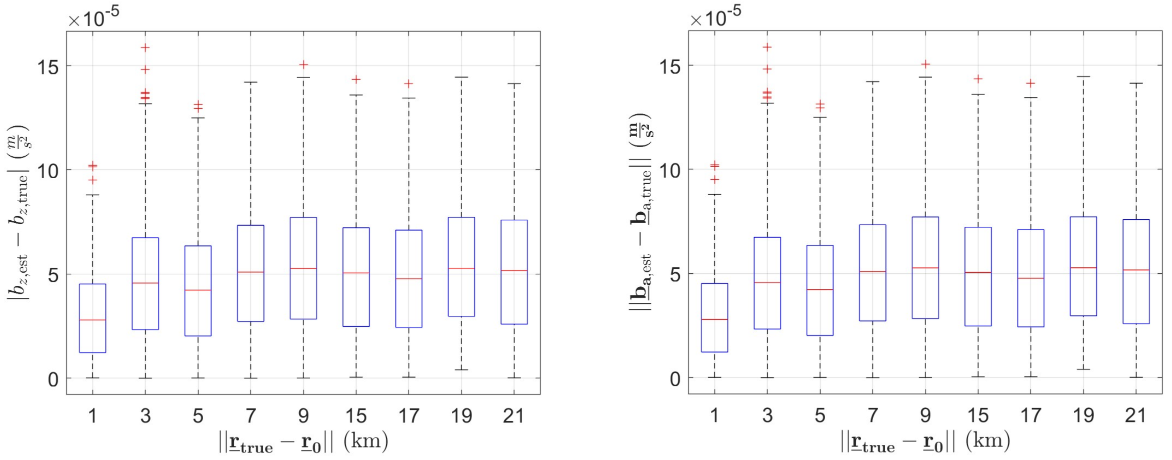
(b) $|\phi_{\text{est}} - \phi_{\text{true}}|$ as a function of $\|\underline{\boldsymbol{r}}_{\text{true}} - \underline{\boldsymbol{r}}_0\|$.

Figure 4.10: Boxplots showing $|\lambda_{\text{est}} - \lambda_{\text{true}}|$ and $|\phi_{\text{est}} - \phi_{\text{true}}|$ as a function of $\|\underline{\boldsymbol{r}}_{\text{true}} - \underline{\boldsymbol{r}}_0\|$ over 150 MEKF runs.

The boxplots contain information on the median, the lower and upper quartiles, outliers, and the minimum and maximum values not included with the outliers. Figures 4.10a and 4.10b show a steady median across all measurements with no noticeable change as the initial guess distance increases. One interesting behavior that is observed is that the maximum errors seem to grow as the distance from the true reference position increases, although the median does not change, as previously mentioned. Figure 4.11 show the boxplots for the component and total accelerometer bias error resulting from the Monte Carlo.



(a) $|b_x - b_{x,\text{true}}|$ as a function of $\|\underline{\boldsymbol{r}}_{\text{true}} - \underline{\boldsymbol{r}}_0\|$. (b) $|b_y - b_{y,\text{true}}|$ as a function of $\|\underline{\boldsymbol{r}}_{\text{true}} - \underline{\boldsymbol{r}}_0\|$.



(c) $|b_z - b_{z,\text{true}}|$ as a function of $\|\underline{\boldsymbol{r}}_{\text{true}} - \underline{\boldsymbol{r}}_0\|$. (d) $\|\underline{\boldsymbol{b}}_{a,\text{est}} - \underline{\boldsymbol{b}}_{a,\text{true}}\|$ as a function of $\|\underline{\boldsymbol{r}}_{\text{true}} - \underline{\boldsymbol{r}}_0\|$.

Figure 4.11: Boxplots showing $|b_x - b_{x,\text{true}}|$, $|b_y - b_{y,\text{true}}|$, $|b_z - b_{z,\text{true}}|$, and $\|\underline{\boldsymbol{b}}_{a,\text{est}} - \underline{\boldsymbol{b}}_{a,\text{true}}\|$ as a function of $\|\underline{\boldsymbol{r}}_{\text{true}} - \underline{\boldsymbol{r}}_0\|$ over 150 MEKF runs.

Figures 4.11a and 4.11b show that the x and y component accelerometer bias errors do not have much variation as the initial guess error increases. Figure 4.11c, however, does show an increase in the z component error.. The z component error gradually increases until an initial position error of approximately 7 km and stays relatively constant beyond that point. This leads to the same increasing behavior in the total error in Figure 4.11d. Even though there is increased error in the accelerometer bias as the distance grows, the difference before and after the approximate 7 km point is relatively small. Still, it is safe to define a smaller initial guess region that produces better MEKF final estimates. For this study, that threshold is set at approximately 6 km (0.2°).

Based on the boxplots above, it is clear that the initial guess distance from the true position does not severely inhibit the accuracy of the MEKF's final results. Future lunar missions are strategically planned and already have a good idea of where they are going to land or navigate on the lunar surface. Therefore, the initial guess position should typically fall within the approximate 6 km upper bound defined above, leading to convergence in the MEKF algorithm as demonstrated.

4.5 1000-Run Monte Carlo at APR

The simulated noisy gravity measurements and their impact on the MEKF's performance were analyzed using a 1000-run Monte Carlo simulation at the APR location. Based on the rotation combination and initial guess analyses shown in Sections 4.3 and 4.4, each run used a 90° rotation about the z axis followed by a 90° rotation about the x axis to calculate the accelerometer bias, and an initial guess within an approximate 6 km circular area of the true position as defined in Table 4.5.

Table 4.5: Initial guess parameters used for 1000-run Monte Carlo at the APR.

Parameter	Value
λ_{true}	$-153.025°$
ϕ_{true}	$-37.725°$
λ_0	$\mathcal{N}(\lambda_{\text{true}}, \sigma_\lambda^2)$
ϕ_0	$\mathcal{N}(\phi_{\text{true}}, \sigma_\phi^2)$
σ_λ	$0.2°$
σ_ϕ	$0.2°$
$\underline{\boldsymbol{g}}_{\text{LGM}}$	$\begin{bmatrix}1.15236 & 0.586934 & 0.998957\end{bmatrix}^T$ (m/s^2)

$\underline{\boldsymbol{g}}_{\text{LGM}}$ in Table 4.5 represents the gravity taken from the LGM2011 gravity model. Each run used a $\Delta t = 1\,\text{s}$ time step between each measurement update and collected 300 measurements per orientation, resulting in a total MEKF run time of 900 s. The gravity measurements followed the distributions shown in Section 4.1 and were predefined for each time step before being used in the MEKF. The gravity was regenerated after each run so that it differed between test cases. This process was repeated for each run to provide a large final estimate dataset. The procedure described above is summarized as follows:

1. Determine (λ_0, ϕ_0) within approximate 6 km region of APR location using the above normal distributions.

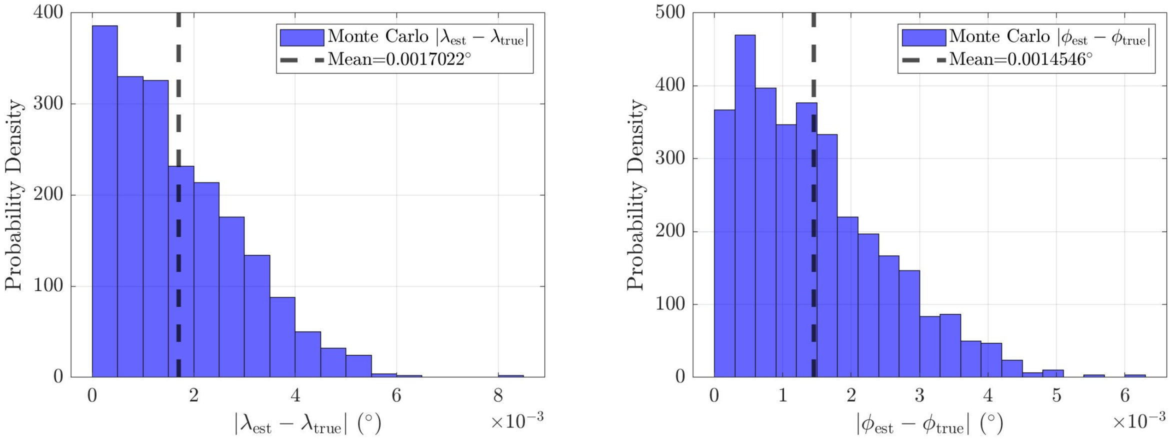
2. Generate simulated gravity measurements for each time step at the APR.

3. Run MEKF algorithm with 90° rotation about the z axis followed by a 90° rotation about the x axis to calculate final accelerometer bias and position estimates.

4. Keep final results for postprocessing and repeat Steps 1–3 for a total of 1000 Monte Carlo runs.

4.5.1 Position $(\lambda_{\text{est}}, \phi_{\text{est}})$ Results

The results for the absolute λ_{est} and ϕ_{est} position errors are plotted in the histograms shown in Figure 4.12, which shows how the noise in the simulated gravity measurements influences the final position accuracy. The mean error, standard deviation, minimum, and maximum for the $|\lambda_{\text{est}} - \lambda_{\text{true}}|$ and $|\phi_{\text{est}} - \phi_{\text{true}}|$ values presented in Figure 4.12 are summarized in Table 4.6.



(a) $|\lambda_{\text{est}} - \lambda_{\text{true}}|$ distribution

(b) $|\phi_{\text{est}} - \phi_{\text{true}}|$ distribution

Figure 4.12: Histograms showing the distributions of the $|\lambda_{\text{est}} - \lambda_{\text{true}}|$ and $|\phi_{\text{est}} - \phi_{\text{true}}|$ absolute errors resulting from 1000 MEKF runs at the APR.

Table 4.6: Error statistics for λ_{est} and ϕ_{est} estimates resulting from 1000 MEKF runs at the APR.

Variable	Mean (°)	Std. Dev. (°)	Minimum (°)	Maximum (°)
$\lvert\lambda_{\text{est}} - \lambda_{\text{true}}\rvert$	1.70218×10^{-3}	1.27047×10^{-3}	2.74502×10^{-6}	8.04099×10^{-3}
$\lvert\phi_{\text{est}} - \phi_{\text{true}}\rvert$	1.45462×10^{-3}	1.05667×10^{-3}	4.26746×10^{-6}	6.29580×10^{-3}

Table 4.6 shows that the average longitude and latitude error is on the order of $10^{-3\circ}$. The latitude is shown to be calculated more accurately than the longitude, matching the estimated position behavior shown in the previous sections. These errors can be put into terms of meters using the haversine formula shown in Eq. 4.9. Figure 4.13 shows the distribution of the haversine distance error between $(\lambda_{\text{est}}, \phi_{\text{est}})$ and $(\lambda_{\text{true}}, \phi_{\text{true}})$ for each Monte Carlo run, along with a fitted χ probability density function (PDF) with $k = 2$ degrees of freedom $(\lambda_{\text{err}}, \phi_{\text{err}})$. The haversine distance error statistics are summarized in Table 4.7.

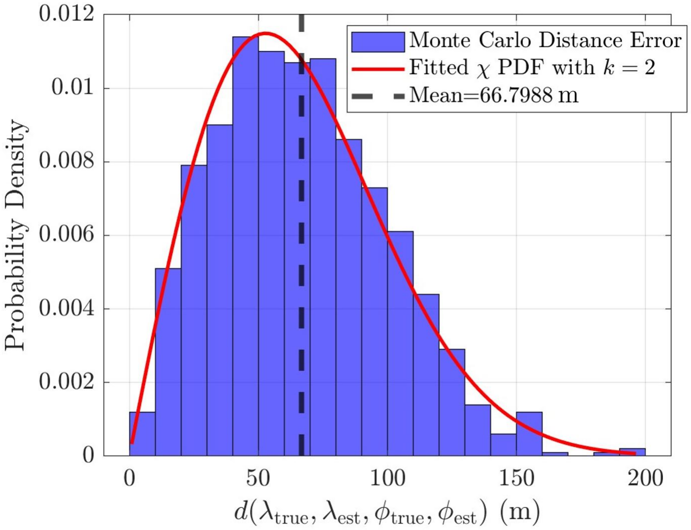
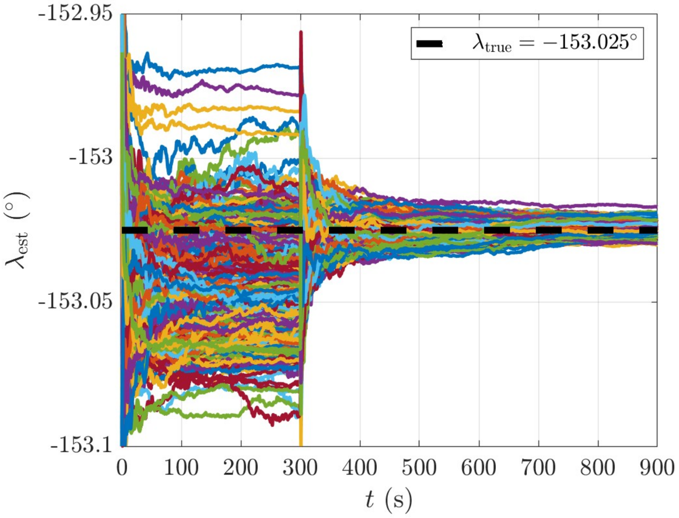


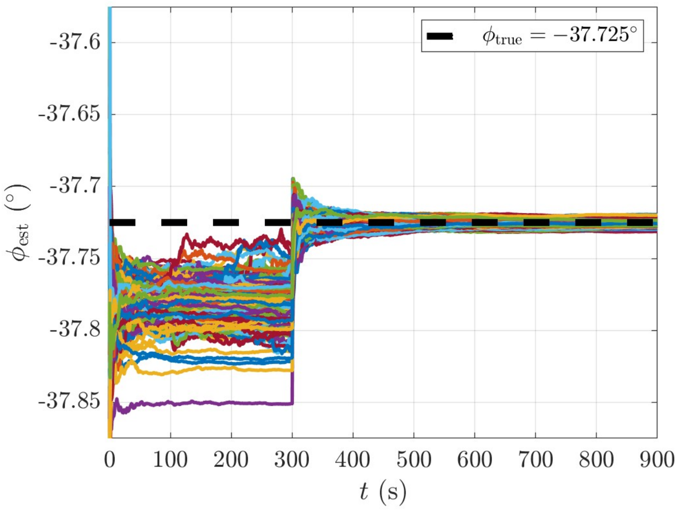
Figure 4.13: Histogram showing the haversine distance error distribution between $(\lambda_{\text{est}}, \phi_{\text{est}})$ and $(\lambda_{\text{true}}, \phi_{\text{true}})$ resulting from 1000 MEKF runs at the APR.

Table 4.7: Statistics for the haversine distance error between $(\lambda_{\text{est}}, \phi_{\text{est}})$ and $(\lambda_{\text{true}}, \phi_{\text{true}})$ resulting from the 1000-run Monte Carlo at the APR.

Mean (m)	Std. Dev (m)	Minimum (m)	Maximum (m)
66.7988	33.3038	0.920929	196.262



(a) λ_{est}



(b) ϕ_{est}

Figure 4.14: λ_{est} and ϕ_{est} values over time for all 1000 MEKF runs.

Figure 4.14 shows the λ_{est} and ϕ_{est} estimates over time for the 1000-run Monte Carlo. Figures 4.14a and 4.14b show the MEKF's ability to converge to a final solution after the system rotations are performed to estimate the accelerometer bias. The variation from the true $(\lambda_{\text{true}}, \phi_{\text{true}})$ values reflects the distributions shown above in Figures 4.12a and 4.12b. The average gravity noise vector magnitude over the MEKF simulation time for each iteration, shown in Figure 4.15, follows a normal distribution, as expected.

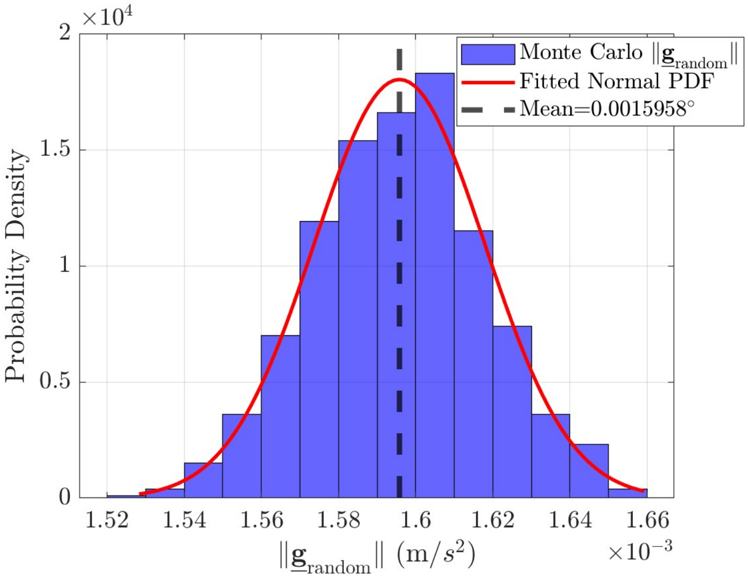


Figure 4.15: Histogram showing the distribution of the mean gravity noise magnitude, $\|\underline{\boldsymbol{g}}_{\text{random}}\|$, for each MEKF run at the APR.

The large range between the minimum and maximum haversine distance errors results from certain random samples being worse than others solely due to the randomness in the measurement noise, as shown in Figure 4.15. The error size can be verified if the MEKF output covariance matrix, $P_{\lambda,\phi}$, is approximated as diagonal, implying that the error is approximately circular.

$$\text{Error Check} = \begin{bmatrix} \lambda_{\text{err}} & \phi_{\text{err}} \end{bmatrix} P_{\lambda,\phi} \begin{bmatrix} \lambda_{\text{err}} \\ \phi_{\text{err}} \end{bmatrix} \tag{4.14}$$

The histogram of Eq. 4.14 should follow the probability density function of a χ^2 distribution with $k = 2$, since the error is two-dimensional, $(\lambda_{\text{err}}, \phi_{\text{err}})$, which is verified in Figure 4.16.

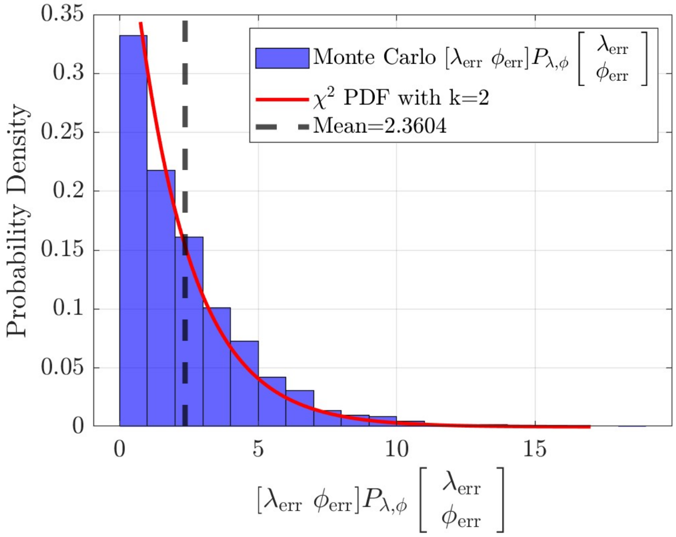


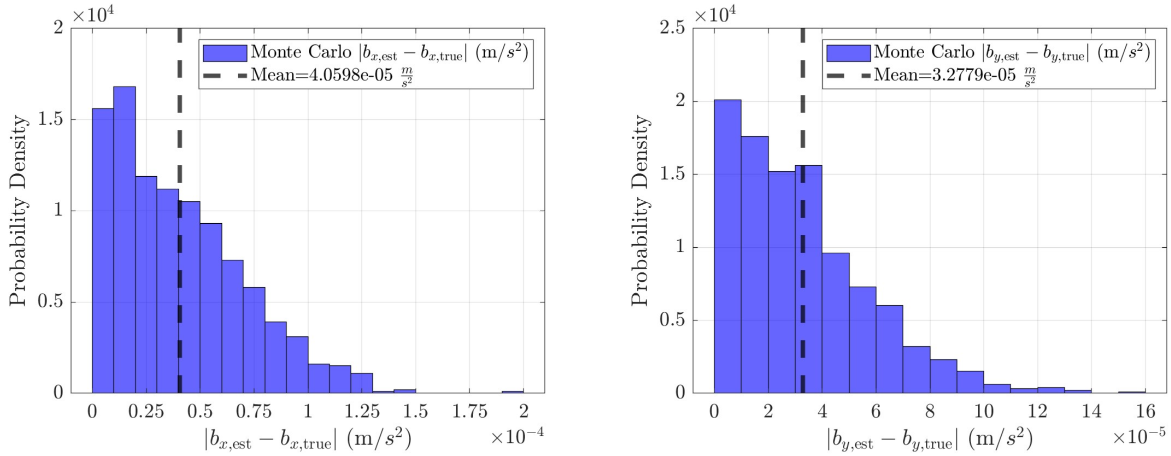
Figure 4.16: Histogram verifying that the error size distribution follows a χ^2 distribution with $k = 2$ degrees of freedom.

4.5.2 Accelerometer Bias Results

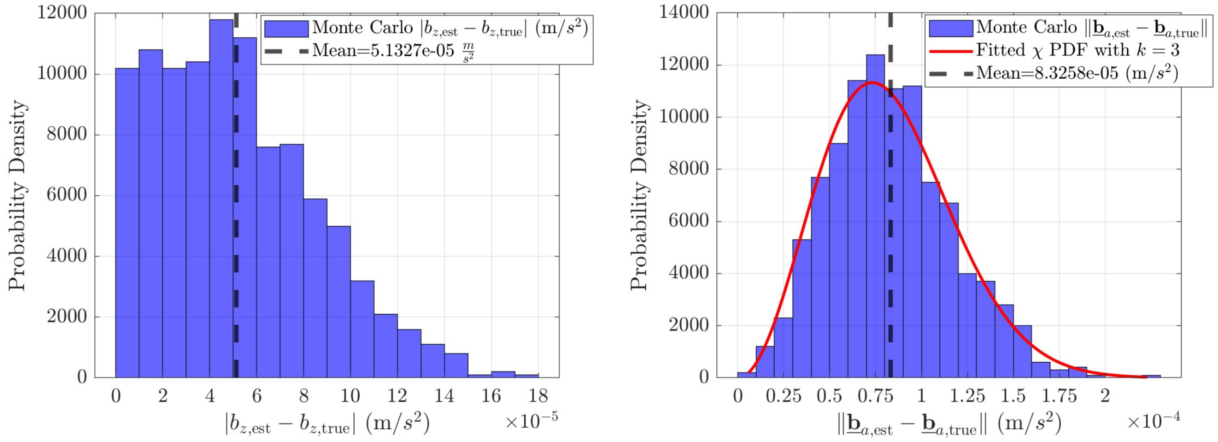
The same process was used to calculate the component and magnitude errors of the estimated accelerometer bias at the APR. The accelerometer bias error directly influences the final position estimates due to the gravity measurement model being a function of the bias estimate. Figure 4.17 shows the histograms describing the component and magnitude distributions of the absolute error relative to the true values. Each component bias error distribution follows a $k = 1$ degree of freedom χ distribution, with an average on the order of 10^{-5}m/s^2. The y component bias has the best overall average, with the z component error having the worst. The x component has the smallest minimum error, with the z component again being the worst among the three components. Lastly, the z component error has the highest maximum error, with the y component having the smallest. Based on this, the z component bias estimation—although it may have a smaller window of convergence, as seen in Figure 4.18—has the worst performance among the three bias error components. The x and y component estimates for each Monte Carlo iteration in Figure 4.18 show a region that is split by the true accelerometer values, producing an average closer to the correct value. The z component converges slightly above the true value, leading to its poorer performance. The y component bias performs the best among the three due to it having the best average error and lowest maximum error. The overall bias magnitude, shown in Figure 4.17d, follows a $k = 3$ degree of freedom χ distribution with its average on the order of $10^{-5}\mathrm{m/s^2}$. The three degrees of freedom come from the contribution of each component error. The error statistics described in this section and shown in the histograms in Figure 4.17 are summarized in Table 4.8.

Table 4.8: Statistics for $b_{x,\text{est}}$, $b_{y,\text{est}}$, $b_{z,\text{est}}$, and $\underline{\boldsymbol{b}}_{a,\text{est}}$ errors resulting from the 1000-run Monte Carlo at the APR.

Variable	Mean (m/s^2)	Std. Dev. (m/s^2)	Minimum (m/s^2)	Maximum (m/s^2)
$\lvert b_{x,\text{est}} - b_{x,\text{true}}\rvert$	4.05976×10^{-5}	3.04275×10^{-5}	6.82635×10^{-9}	1.90291×10^{-4}
$\lvert b_{y,\text{est}} - b_{y,\text{true}}\rvert$	3.27794×10^{-5}	2.50649×10^{-5}	3.46675×10^{-8}	1.51803×10^{-4}
$\lvert b_{z,\text{est}} - b_{z,\text{true}}\rvert$	5.13267×10^{-5}	3.39171×10^{-5}	1.22657×10^{-7}	1.75213×10^{-4}
$\lVert \underline{\boldsymbol{b}}_{a,\text{est}} - \underline{\boldsymbol{b}}_{a,\text{true}}\rVert$	8.32576×10^{-5}	3.35877×10^{-5}	5.93538×10^{-6}	2.22741×10^{-4}



(a) $|b_{x,\mathrm{est}} - b_{x,\mathrm{true}}|$ error distribution (b) $|b_{y,\mathrm{est}} - b_{y,\mathrm{true}}|$ error distribution



(c) $|b_{z,\mathrm{est}} - b_{z,\mathrm{true}}|$ error distribution (d) $\|\underline{\boldsymbol{b}}_{a,\mathrm{est}} - \underline{\boldsymbol{b}}_{a,\mathrm{true}}\|$ error distribution

Figure 4.17: Histograms showing the $|b_{x,\mathrm{est}} - b_{x,\mathrm{true}}|$, $|b_{y,\mathrm{est}} - b_{y,\mathrm{true}}|$, $|b_{z,\mathrm{est}} - b_{z,\mathrm{true}}|$, and $\|\underline{\boldsymbol{b}}_{a,\mathrm{est}} - \underline{\boldsymbol{b}}_{a,\mathrm{true}}\|$ error distributions resulting from the 1000-run Monte Carlo at the APR.

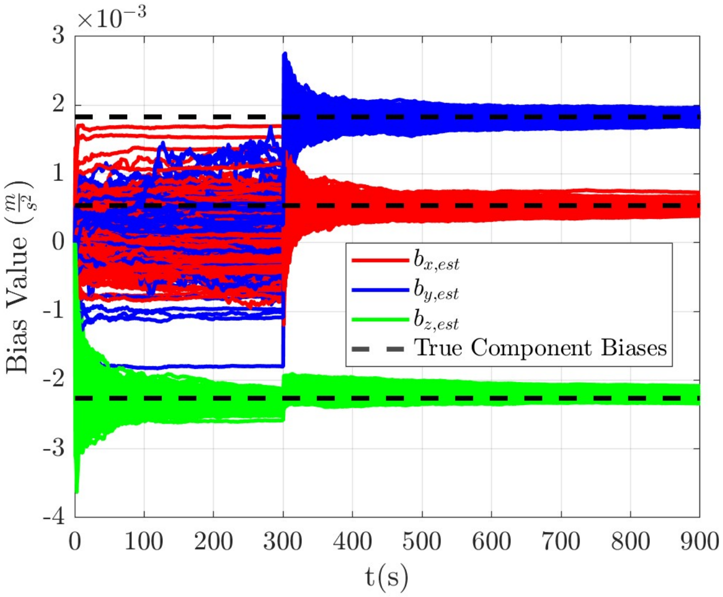


Figure 4.18: Accelerometer component bias over time for the 1000-run APR Monte Carlo.

4.6 Accelerometer Measurement Error Impact on MEKF Final Estimates

The MEKF final estimate accuracy is only as good as the accelerometer's ability to measure the local gravity field. As shown in Eq. 4.2, the gravity measurement noise follows a normal distribution in each component with variance σ^2_{accel}. This section evaluates the MEKF's performance for the four σ_{accel} values shown below

$$\underline{\boldsymbol{\sigma}}_{\text{accel,test}} \in \{1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}\} \text{ m/s}^2 \tag{4.15}$$

The second term in Eq. 4.15 represents the same value used in previous sections. The $\underline{\boldsymbol{\sigma}}_{\text{accel,test}} = 1 \times 10^{-2}$ m/s^2 value corresponds to the greatest noise, while $\underline{\boldsymbol{\sigma}}_{\text{accel,test}} = 1 \times 10^{-5}$ m/s^2 reflects the least. This section analyzes the MEKF's performance across this range of measurement accuracies. A Monte Carlo simulation was conducted for each test value, with

200 MEKF runs performed in each case. The final results from each run were retained for post-processing. All MEKF runs followed the same rotation sequence described earlier and were carried out at the APR. The bias estimates over time for each accuracy case is shown in Figures 4.19–4.22.

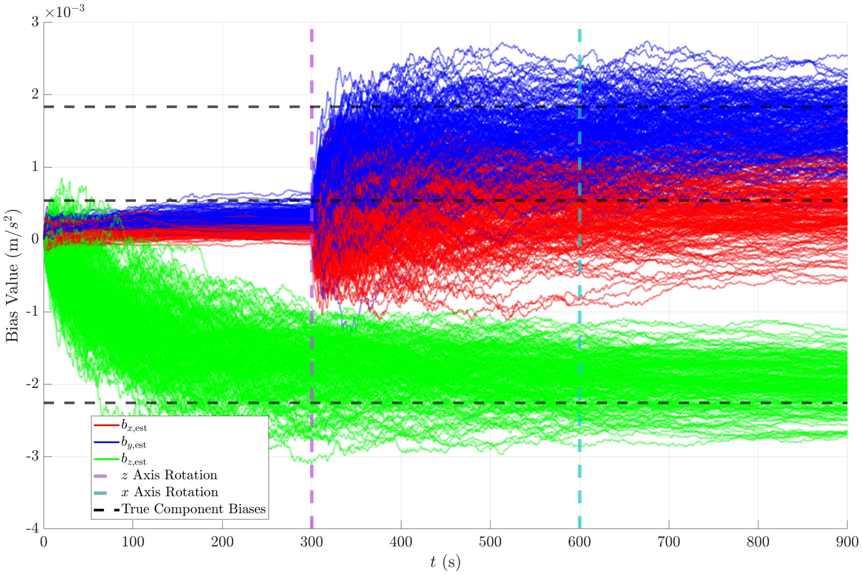


Figure 4.19: Component bias estimates for 200 MEKF simulations with $\sigma_{\text{accel}} = 1 \times 10^{-2}$ m/s^2.

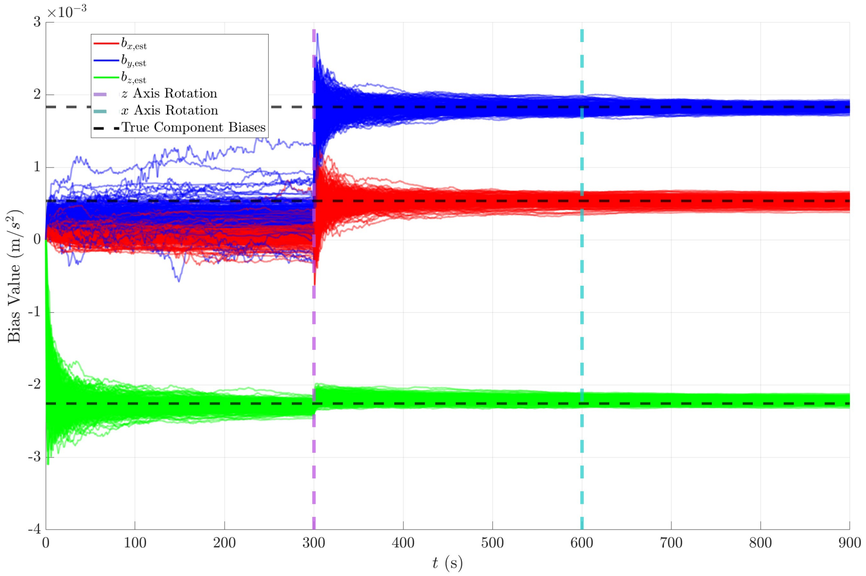


Figure 4.20: Component bias estimates for 200 MEKF simulations with $\sigma_{\text{accel}} = 1 \times 10^{-3}$ m/s^2.

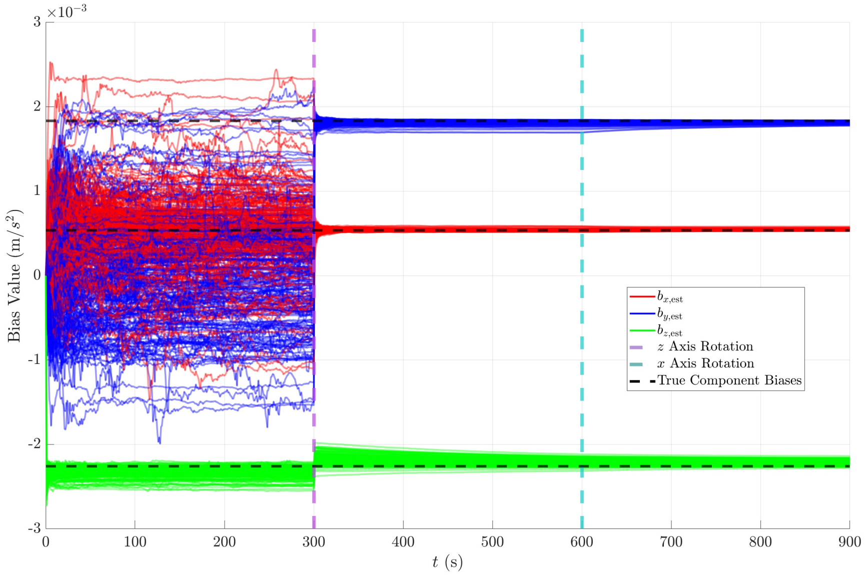


Figure 4.21: Component bias estimates for 200 MEKF simulations with $\sigma_{\text{accel}} = 1 \times 10^{-4}$ m/s^2.

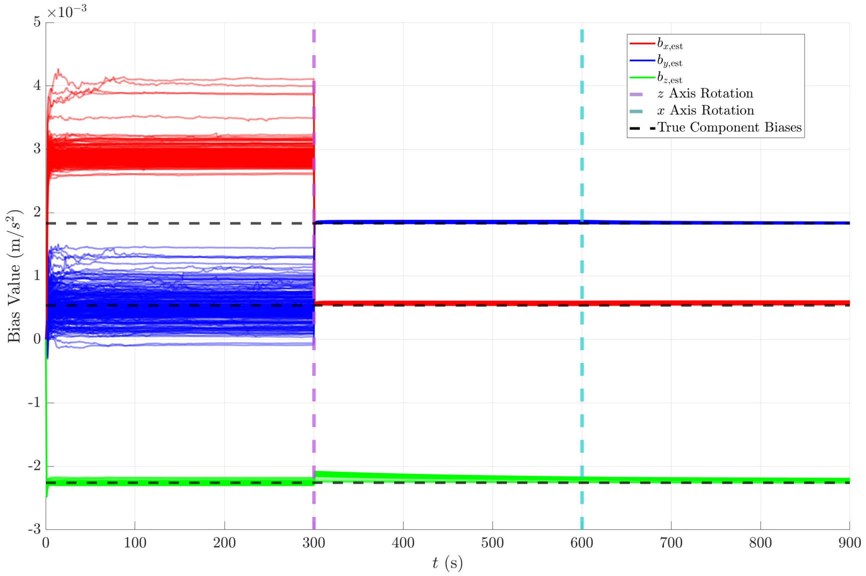
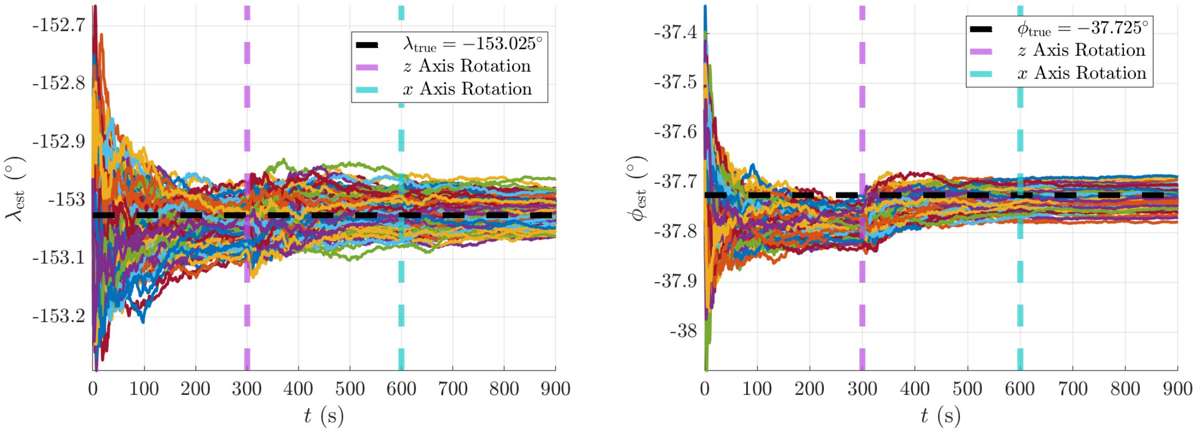


Figure 4.22: Component bias estimates for 200 MEKF simulations with $\sigma_{\text{accel}} = 1 \times 10^{-5}$ m/s^2.

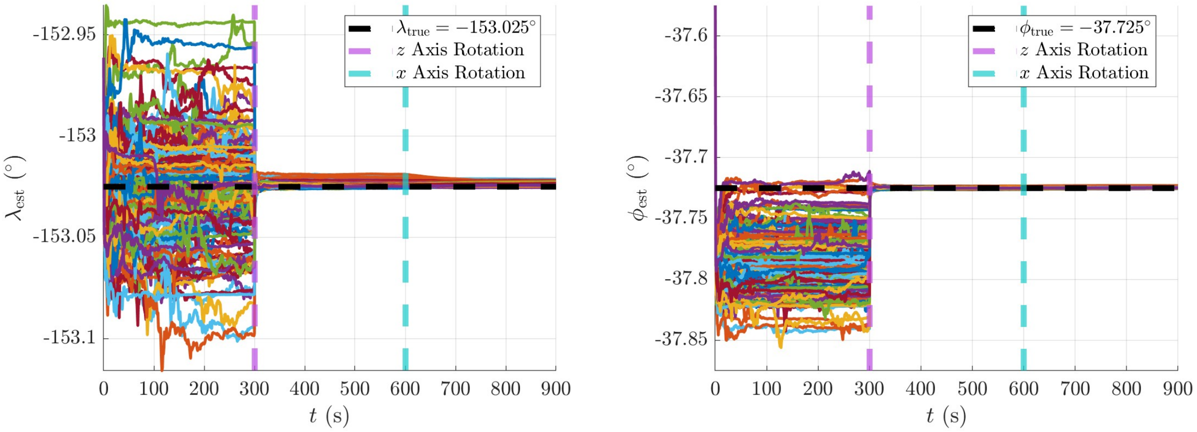
Figures 4.19–4.22 demonstrate a clear change in estimation accuracy across the tested cases. As expected, higher noise levels in the gravity measurements result in less accurate bias estimations. In the highest-noise scenario, the estimated values do not converge to a consistent solution, despite trending toward the correct values. Figure 4.20, previously discussed, is included again here for comparison. The two low-noise cases, shown in Figures 4.21 and 4.22, exhibit high-accuracy accelerometer bias estimates after the initial 90° rotation about the z axis. The spread around the true bias values is significantly narrower than in the high-noise cases.

The corresponding $(\lambda_{\text{est}}, \phi_{\text{est}})$ values over time for each test case are shown in Figures 4.23–4.25. The position estimate results for $\sigma_{\text{accel}} = 1 \times 10^{-3}$ m/s^2 can be referenced from Figure 4.14. Summary statistics for each test case are provided in Table 4.9.



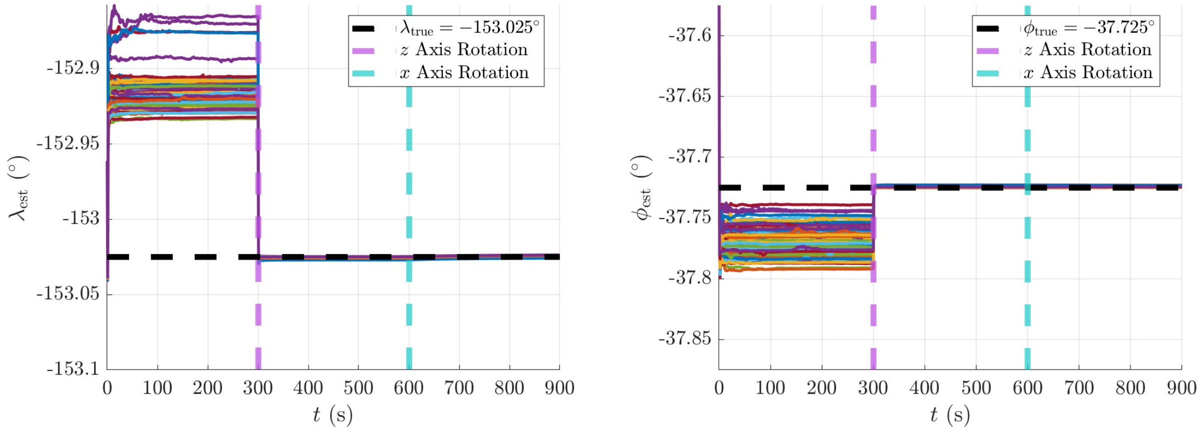
(a) λ_{est} (b) ϕ_{est}

Figure 4.23: λ_{est} and ϕ_{est} estimates for 200 MEKF runs with $\sigma_{\text{accel}} = 1 \times 10^{-2}$ m/s^2.



(a) λ_{est} (b) ϕ_{est}

Figure 4.24: λ_{est} and ϕ_{est} estimates for 200 MEKF runs with $\sigma_{\text{accel}} = 1 \times 10^{-4}$ m/s^2.



(a) λ_{est}

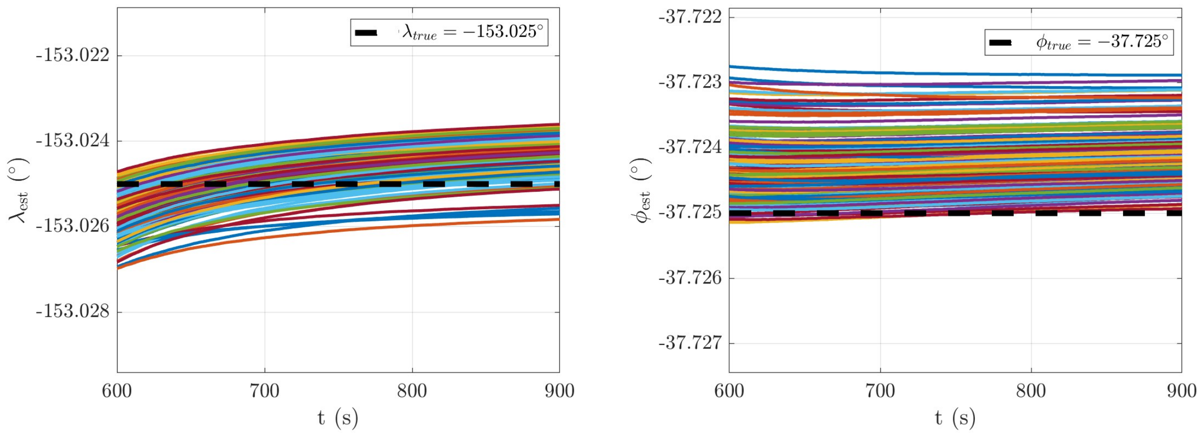
(b) ϕ_{est}

Figure 4.25: λ_{est} and ϕ_{est} estimates for 200 MEKF runs with $\sigma_{\text{accel}} = 1 \times 10^{-5}$ m/s^2.

Table 4.9: Statistics for the haversine distance error between $(\lambda_{\text{est}}, \phi_{\text{est}})$ and $(\lambda_{\text{true}}, \phi_{\text{true}})$ resulting from 200 MEKF runs at the APR using varying values of σ_{accel}.

σ_{accel} (m/s^2)	Average (m)	Std. Dev (m)	Minimum (m)	Maximum (m)
1×10^{-2}	572.501	320.585	32.4098	1652.12
1×10^{-3}	66.5596	33.5272	3.47882	186.799
1×10^{-4}	23.3957	17.8185	1.30790	90.2105
1×10^{-5}	32.0829	9.11244	11.7182	66.3709

As expected, estimation accuracy improves as the noise in the gravity measurements decreases. However, some notable behavior is observed in the $\sigma_{\text{accel}} = 1 \times 10^{-5}$ m/s^2 case. While improved accelerometer precision would typically be expected to yield lower average errors, the results in Table 4.9 show that the average error is slightly higher than in the $\sigma_{\text{accel}} = 1 \times 10^{-4}$ m/s^2 case. From the zoomed-in plots in Figure 4.26, it is evident that λ_{est} and ϕ_{est} converge to values slightly greater than λ_{true} and ϕ_{true}, respectively. This suggests the presence of a limiting factor in the achievable accuracy of position estimates based on the noise level in gravity measurements. The analysis presented in this section confirms that the accuracy of the MEKF algorithm is directly influenced by the quality of the accelerometer data. However, it also reveals a limit beyond which further improvements in accelerometer accuracy do not yield better position estimate accuracy.



(a) λ_{est} (b) ϕ_{est}

Figure 4.26: Zoomed in λ_{est} and ϕ_{est} estimates for 200 MEKF simulations with $\sigma_{\text{accel}} = 1 \times 10^{-5}$ m/s^2, showing the estimates converging to a value greater than λ_{true}.

4.7 Surface Navigation with Correlated Gravity Biases

4.7.1 Lunar Pathfinder Ranging

The correlated gravity procedure requires a highly accurate reference position at some point during the traversing mission. In this study, the reference position is calculated prior to the start of the traversing mission. Two-way ranging measurements with the Lunar Pathfinder satellite are used to establish this reference. The weighted batch-to-EKF filter outlined for the Endurance mission by Cortinovis, Mina, and Gao in [46] provides an effective method for using the ranging measurements to produce a final state estimate and covariance. The process begins by collecting a large batch of measurements over an extended period, assuming the satellite is observable and able to communicate with the ground system (as determined by El and C/N_0). These measurements are then processed by a weighted batch filter, which generates an estimate of the ground position and its associated covariance. This batch filter output serves as the initial guess for the EKF, which subsequently performs measurement updates as new ranging data become available. The true location of the ground system is at the APR, aligning with the previous analyses. The two-way ranging measurement model used in the EKF incorporates Gaussian white noise, v_t, as defined below

$$v_t \sim \mathcal{N}(0, R_t) = \mathcal{N}\left(0, \sigma^2_{p,\text{tot},k+1}\right) \tag{4.16}$$

where R_t is the measurement noise covariance, which is one-dimensional since the measurement model is based on the magnitude of the difference between the dynamically propagated state and the ranging measurement. The $(k+1)$ subscript simply denotes the $\sigma_{p,\text{tot}}$ uncertainty at time step $k+1$. Further details on the state prediction, measurement update, measurement models, and final state and covariance estimation can be found in [46]. As shown in Eq. 4.16, the noise in the measurement model is characterized by the $\sigma_{p,\text{tot}}$ uncertainty defined in Eq. 3.112. The $\sigma_{p,\text{tot}}$ value varies with each ranging measurement, depending on the C/N_0 value for that sample, which is itself influenced by the satellite's elevation angle, El. The elevation angle of the Lunar Pathfinder satellite over time at the APR is shown in Figure 4.27.

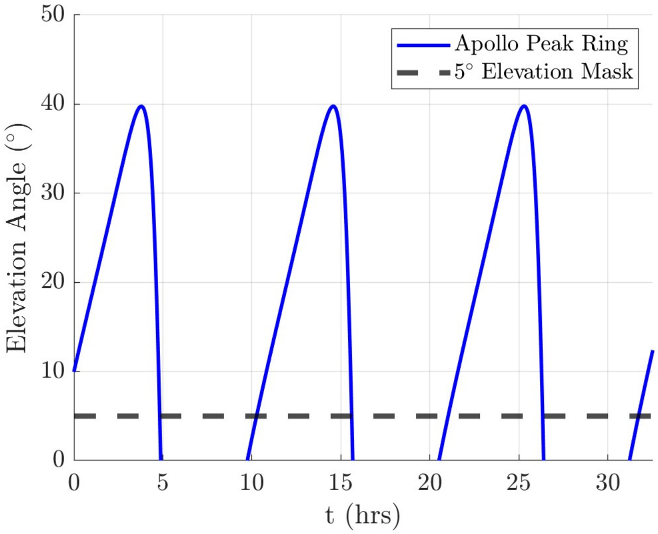


Figure 4.27: Lunar Pathfinder elevation angle over time from the APR. There are periods where the satellite is unable to communicate with the ground system, occurring when he elevation is below the 5° threshold.

Figure 4.27 shows that there are periods when the satellite is unavailable to the ground system, occurring when the elevation angle is below the 5° threshold. This shows that the satellite is typically available in periods of approximately 5 hrs. These elevation angles are used to calculate the receiver and transmitter off-boresight angles and the resulting antenna gains. Those gains are then used to calculate the C/N_0 values shown in Figure 4.28.

The C/N_0 values in Figure 4.28 indicate that the signal strength to the ground system exceeds the minimum threshold during approximately the same periods as the elevation angle, enabling reliable two-way ranging measurements at those times. Given the El and C/N_0 values in Figures 4.27 and 4.28, $\sigma_{p,\text{tot}}$ can be calculated for each two-way ranging measurement.

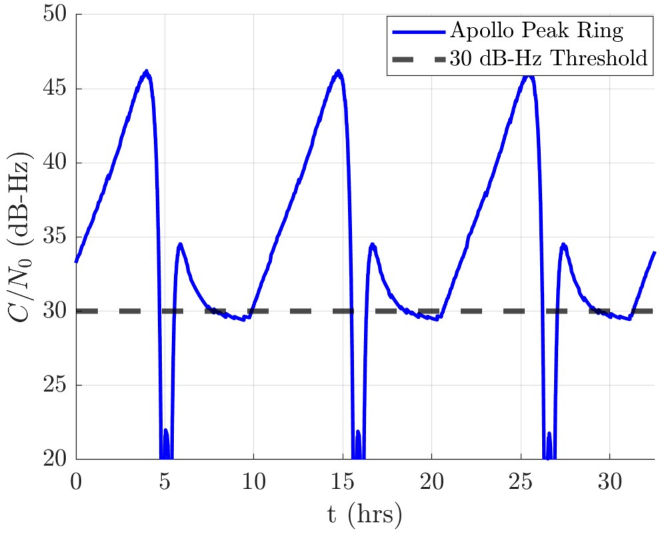


Figure 4.28: Lunar Pathfinder C/N_0 over time at the APR.

The batch filter in this study was simulated to collect ranging measurements over the course of one hour. It requires an initial position guess in the Moon PA frame to begin the algorithm. The initial guess, relative to the true APR location shown in Eq. 4.17, was set to the value in Eq. 4.18, which is approximately 135 m from the true location. This initial guess was generated by applying a random perturbation to each component of the true position.

$$\underline{\boldsymbol{r}}_{\text{APR,PA}} = \begin{bmatrix} -1.22539 \\ -0.622944 \\ -1.06364 \end{bmatrix} \text{km} \tag{4.17}$$

$$\underline{\boldsymbol{x}}_{0,\text{batch}} = \begin{bmatrix} -1.20571 \\ -0.743713 \\ -1.00704 \end{bmatrix} \text{km} \tag{4.18}$$

The batch filter then used the initial guess and ranging measurements to perform a nonlinear weighted least squares optimization, converging to the final state estimate of the ground system position in the PA frame. The resulting state estimate from the batch filter was then used as the initial guess for the EKF algorithm. The initial covariance estimate was determined based on the uncertainty in the final batch filter measurement and the Position Dilution of Precision (PDOP), defined using

$$\text{PDOP} = \text{tr}\left(\left(H_K^T H_k\right)^{-1}\right) \tag{4.19}$$

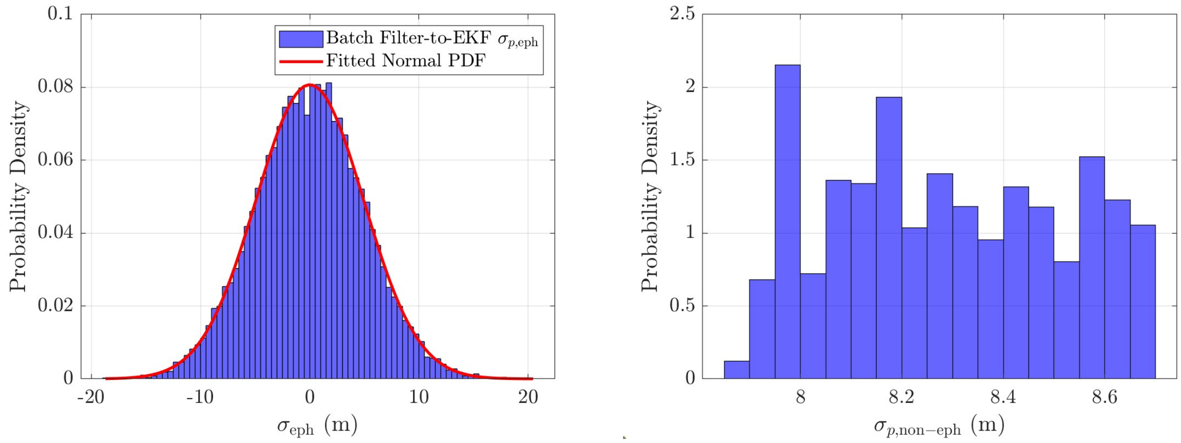
where H_k is the batch filter's Jacobian matrix at the final convergence step k. More on the batch filter Jacobian is defined in [46]. The EKF's initial covariance is then defined using

$$P_{0,\text{EKF}} = (\sigma_{p,\text{tot},f}\text{PDOP})^2 I_{3\times 3} \tag{4.20}$$

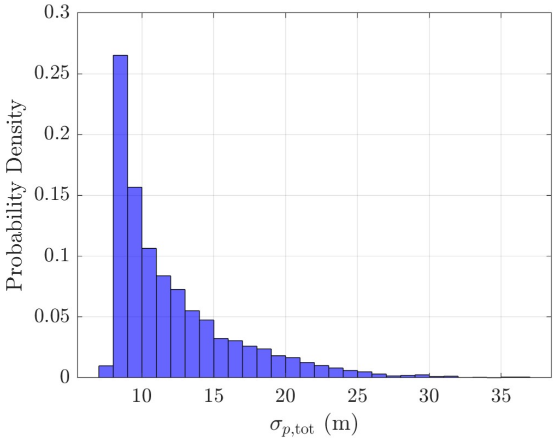
The EKF $\sigma_{p,\text{eph}}$ errors due to the uncertainty in the satellite ephemeris data is assumed to follow the normal distribution shown in Eq. 4.21, where the standard deviation of the error is shown to be 5 m.

$$\sigma_{p,\text{eph}} \sim \mathcal{N}(0, (5\,\text{m})^2) \tag{4.21}$$

The non-ephemeris error in the ranging measurements was calculated using the $\sigma_{p,\text{DLL}}$ errors (relies on the C/N_0 values above) and the constant $\sigma_{p,\text{mul}}$ and $\sigma_{p,\text{rel}}$ values. The $\sigma_{p,\text{eph}}$, $\sigma_{p,\text{non-eph}}$, and resulting $\sigma_{p,\text{tot}}$ uncertainty distributions for the batch filter-to EKF algorithm are shown in Figure 4.29.



(a) $\sigma_{p,\text{eph}}$ (b) $\sigma_{p,\text{non}-\text{eph}}$



(c) $\sigma_{p,\text{tot}}$

Figure 4.29: Histogram showing the satellite $\sigma_{p,\text{eph}}$, $\sigma_{p,\text{non}-\text{eph}}$, and $\sigma_{p,\text{tot}}$ error distributions during the batch-to-EKF algorithm. A normal distribution can be fitted to the $\sigma_{p,\text{eph}}$ data as shown.

The $\sigma_{p,\mathrm{tot}}$ errors in the ranging measurements directly influence the accuracy of the final position estimate. The ranging measurements from the ground system to the Lunar Pathfinder satellite used in this analysis are shown in Figure 4.30.

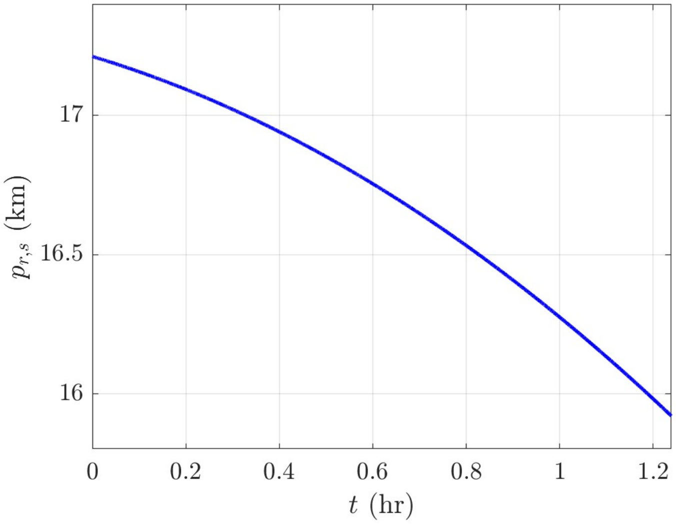
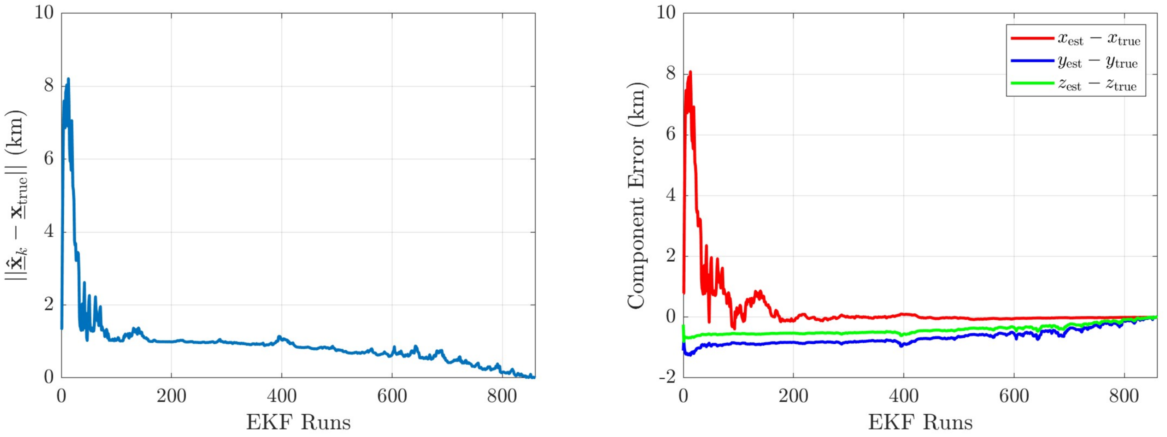


Figure 4.30: Two-way ranging measurements, $p_{r,s}$ during the batch-to-EKF simulation.

These measurements indicate a decreasing distance between the ground system and the satellite over the duration of the batch-to-EKF run. The magnitude of the position error and the individual component errors during the EKF phase of the algorithm are illustrated in Figure 4.31. Both figures show an initial spike in error at the start of the EKF, followed by a steady decrease as the filter converges toward an accurate final estimate. The convergence criterion for the final estimated position was defined as an error of less than 5 m between the estimated and true positions. While more rigorous methods exist for assessing convergence of the batch-to-EKF algorithm, this threshold was deemed sufficient for the purposes of this study, which requires only an accurate reference position.



(a) $\|\hat{\underline{\boldsymbol{x}}}_k - \underline{\boldsymbol{x}}_{\text{true}}\|$

(b) Component error

Figure 4.31: $\|\hat{\underline{\boldsymbol{x}}}_k - \underline{\boldsymbol{x}}_{\text{true}}\|$ and component position error during the EKF simulation.

The final estimated position was found in the Moon's PA frame. Since the correlated gravity algorithm uses (λ, ϕ) positions instead of cartesian vectors, the final state was rotated into the MCMF frame as shown in Eq. 4.22 where its (λ, ϕ) could then be found. This rotation was performed according to

$$\underline{\boldsymbol{x}}_{\text{ref,MCMF}} = R_{\text{PA}}^{\text{MCMF}} \underline{\boldsymbol{x}}_{\text{ref,PA}} \tag{4.22}$$

The process to convert from the cartesian position vector, $\underline{\boldsymbol{x}}_{\text{ref,MCMF}} = [x_{\text{ref}} \; y_{\text{ref}} \; z_{\text{ref}}]^{\text{T}}$, to the final $(\lambda_{\text{ref}}, \phi_{\text{ref}})$ coordinates starts by solving for λ_{ref}:

$$\lambda_{\text{ref}} = \text{atan2}(y_{\text{ref}}, x_{\text{ref}}) \tag{4.23}$$

The process to compute ϕ_{ref} is simplified in this study by assuming a spherical Moon. ϕ_{ref} is calculated using the radius of the Moon and the z component of the MCMF cartesian position vector.

$$\phi_{\text{ref}} = \sin^{-1}\left(\frac{z_{\text{ref}}}{r_{\text{moon}}}\right) \tag{4.24}$$

Using the equations for λ_{ref} and ϕ_{ref} above produced the satellite reference position in Table

4.10, which is used in the correlated gravity algorithm.

Table 4.10: Calculated satellite reference position using two-way ranging measurements and the combined weighted batch-to-EKF algorithm.

Variable	Value (°)
λ_{ref}	-153.042
ϕ_{ref}	-37.7113

4.7.2 Calculating Gravity Field Correlation Function

The correlated gravity procedure outlined in Section 3.10 produces position and bias estimates in the LGM2011 gravity field. The algorithm is analyzed using two test case conditions: a traversing mission on the lunar surface and a traversing mission in an Earth-like gravity field. The traversing mission on the Moon incorporates the LGM2011 gravity model and uses the MEKF algorithm to solve for the measurement position at each waypoint. Since the resolution of the LGM2011 gravity model is only 0.05°, there is uncertainty in the correlations between gravity vectors within the $0.05° \times 0.05°$ grids. The Earth-like gravity field was calculated below the resolution of the LGM2011 model to find the correlated gravity function $c(d)$ that is unknown within the resolution of the LGM2011 model.

Since the methods presented in Section 3.10 require a $c(d)$ function, we turn to another test case where the gravity is known very accurately at a fine scale, leading to the necessity of a traversing mission using Earth's gravity field. There are numerous highly-accurate gravity models of the Earth's surface obtained over the years through the use of satellite and other remote sensing data. The Earth gravity model used in this study is the SGG-UGM-2 gravity model developed by Liang et al. in [48]. This model uses satellite gravimetry, satellite altimetry, and the Earth Gravitational Model 2008 (EGM2008) to produce a high-resolution gravity field model. The model is evaluated up to degree 2190 and order 2159 by combining the observations of the Gravity Field and Steady-State Ocean Circulation Explorer (GOCE), data from the Gravity Recovery and Climate Experiment (GRACE), and marine gravity data from satellite altimetry.

The Global Geodetic Observing System (GGOS) provides a service that takes input (λ, ϕ) positions and uses a known gravity field to produce needed gravity data [49]. The user can input the gravity field parameters they desire at those points, some of which include surface gravity magnitude, EW/NS deflections from the vertical, geoid, normal gravity, and many more. The only parameters needed for this study are the surface gravity magnitude and EW/NS deflections from the vertical, matching the LGM2011 output data. The service can only evaluate up to 1000 (λ, ϕ) coordinates at a time. To try and match the terrain of the lunar surface as best as possible, the area around the Meteor Crater National Landmark in Arizona, shown in Figure 4.32, was used.

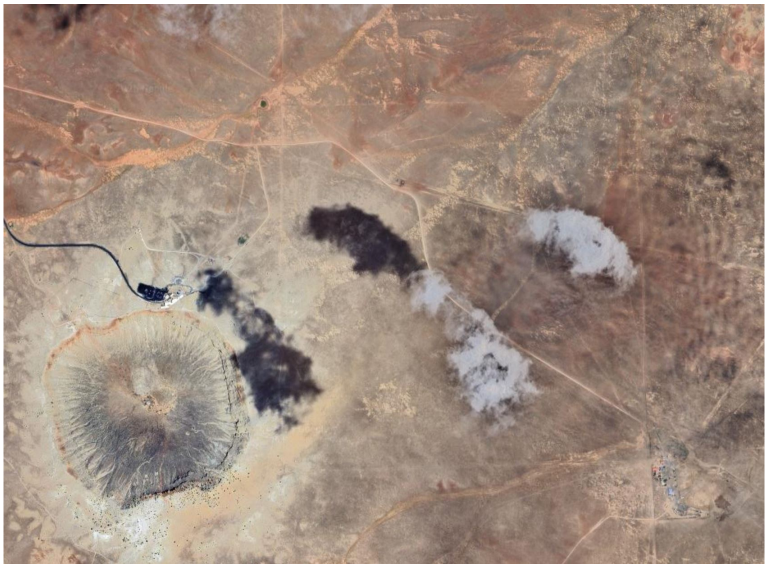
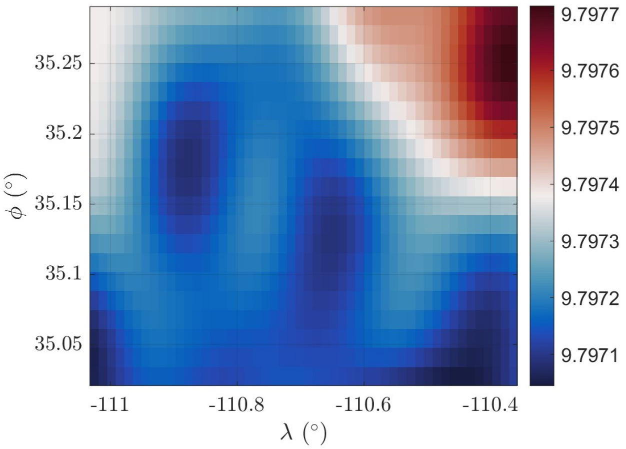
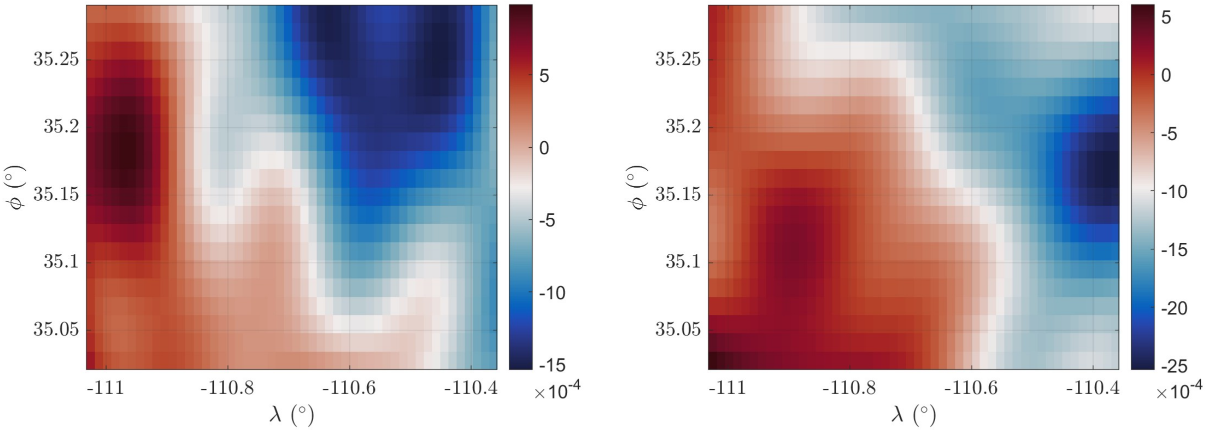


Figure 4.32: Region surrounding Meteor Crater National Landmark used in Earth traversing mission. This region was selected to try and resemble the lunar terrain (Image taken from Google Earth).

Two different resolution grids of this region were taken: one with an approximately 1.5 km resolution, matching that of the LGM2011 gravity model, and the other having a finer approximate 30 m resolution. The 1.5 km resolution field represents the LGM2011 model on the Earth, and the 30 m resolution data is used to show the errors resulting from not having finer spatial scale data. (This will provide the $c(d)$ correlation function.) The 1.5 km gravity data ranges from $\lambda_{\text{min}} = -111.025°$ to $\lambda_{\text{max}} = -110.3635°$ and $\phi_{\text{min}} = 35.0275°$ to $\phi_{\text{max}} = 35.2840°$ in 0.0135° increments, producing the 1.5 km resolution on the Earth. This is different from the 0.05° spacing in the LGM2011 gravity model due to the Earth's radius being larger than the Moon's. The 30 m resolution model was contained within one $0.0135° \times 0.0135°$ grid of the larger 1.5 km resolution field to show the changes within each grid size. The 30 m gravity data ranges from $\lambda_{\text{min}} = -111.025°$ to $\lambda_{\text{max}} = -111.0115°$ and $\phi_{\text{min}} = 35.0275°$ to $\phi_{\text{max}} = 35.0410°$ in approximately $2.7750 \times 10^{-5°}$ increments. The surface gravity magnitude and EW/NS deflections are shown below for both resolution grids in Figures 4.33 and 4.34.



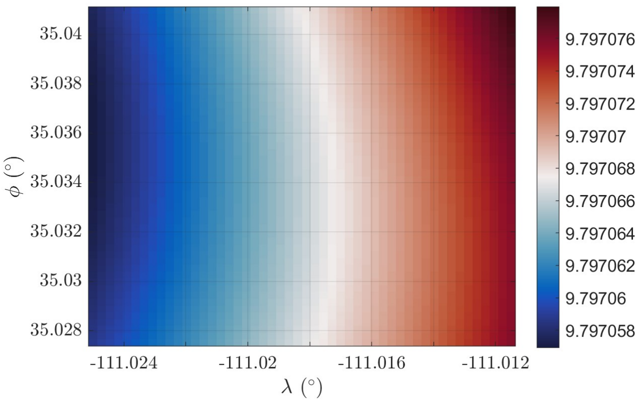
(a) Surface gravity magnitude (m/s^2)



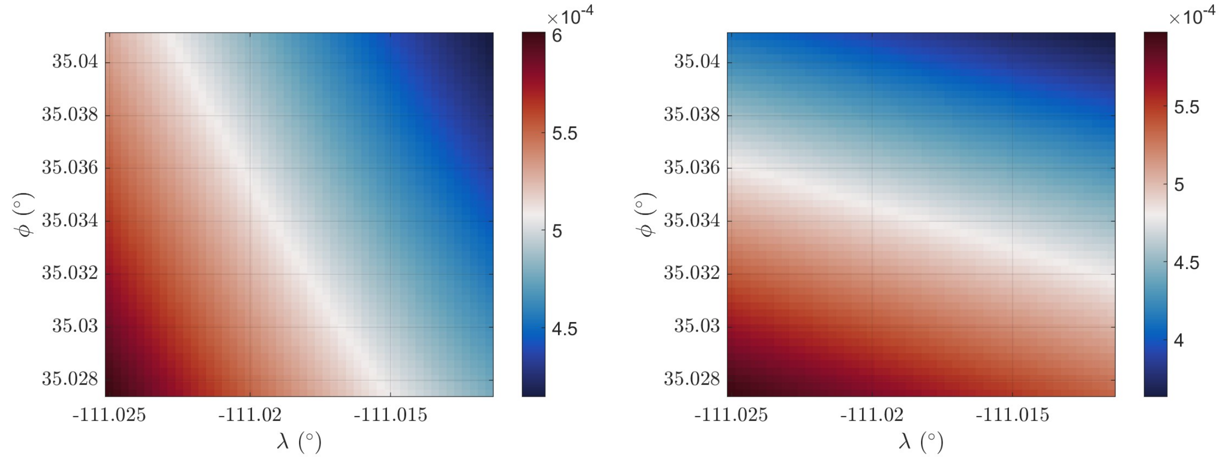
(b) EW gravity deflections (°)

(c) NS gravity deflections (°)

Figure 4.33: Surface gravity magnitude and EW/NS deflections of the 1.5 km resolution Earth gravity model.



(a) Surface gravity magnitude (m/s^2)



(b) EW gravity deflections (°)

(c) NS gravity deflections (°)

Figure 4.34: Surface gravity magnitude and EW/NS deflections of the 30 m resolution Earth gravity model.

To find the $c_{\text{earth}}(d)$ function relating the correlation between local gravity vectors, the first step was to take the difference between the high-resolution EW/NS deflections, $\delta_{\text{EW/NS,high-res}}$, and the low-resolution EW/NS deflections, $\delta_{\text{EW/NS,low-res}}$, at each (λ, ϕ) position. Since the (λ, ϕ) coordinates do not coincide between the two models, the low-resolution deflection data was interpolated at the high-resolution $(\lambda_{\text{high-res}}, \phi_{\text{high-res}})$ positions. This deflection difference is

$$\Delta\delta_{(\lambda_i,\phi_i)} = \delta_{\text{EW/NS,high-res}}(\lambda_i, \phi_i) - \delta_{\text{EW/NS,low-res}}(\lambda_i, \phi_i) \tag{4.25}$$

$\Delta\delta_{(\lambda_i,\phi_i)}$ was calculated at each $(\lambda_{\text{high-res}}, \phi_{\text{high-res}})$ position and used to calculate the overall bias uncertainty, σ_b, between measurements using

$$\mathbb{E}\{\Delta\delta_{(\lambda_i,\phi_i)}\Delta\delta_{(\lambda_i,\phi_i)}\} = \sigma_b^2 \tag{4.26}$$

where $\mathbb{E}\{\Delta\delta_{(\lambda_i,\phi_i)}\Delta\delta_{(\lambda_i,\phi_i)}\}$ represents the mean of the vector containing each $\Delta\delta_{(\lambda_i,\phi_i)}\Delta\delta_{(\lambda_i,\phi_i)}$ value. The σ_b uncertainty shown above corresponds to the same value in Eq. 3.117. Equation 4.26 leads to the following two uncertainty values for the EW and NS deflections: $\sigma_{b,\text{EW}}^2 = 1.5023 \times 10^{-13\circ}$ and $\sigma_{b,\text{NS}}^2 = 5.7625 \times 10^{-13\circ}$. After σ_b was calculated, the $\Delta\delta_{(\lambda_i,\phi_i)}$ deflection differences at each (λ_i, ϕ_i) position and its product with the deflection difference at every other (λ_j, ϕ_j) point in the high-resolution grid was found. The high-resolution gravity data is a 50×50 array due to the longitude and latitude ranges and the increments within them, leading to a total of $N = 2500$ coordinates. The total number of (λ_i, ϕ_i) and (λ_j, ϕ_j) pairs without repeating is found using

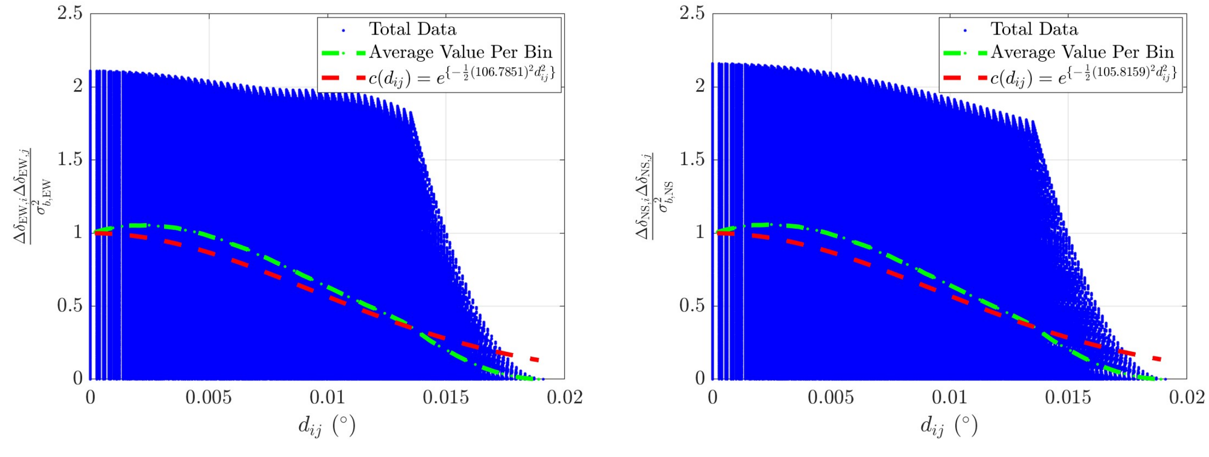
$$\text{Total Pairs} = \frac{N(N+1)}{2} \tag{4.27}$$

This leads to a vector containing 3 126 250 $\Delta\delta_{(\lambda_i,\phi_i)}\Delta\delta_{(\lambda_j,\phi_j)}$ values. The $c_{\text{earth}}(d)$ function is found by fitting the resulting $\Delta\delta_{(\lambda_i,\phi_i)}\Delta\delta_{(\lambda_j,\phi_j)}/\sigma_b^2$ values with respect to the distance,

d_{ij}, between each (λ_i, ϕ_i) and (λ_j, ϕ_j) pairing. This distance is calculated in degrees using $d_{ij} = \|\underline{\boldsymbol{r}}_j - \underline{\boldsymbol{r}}_i\|$, where $\underline{\boldsymbol{r}}_j = (\lambda_j, \phi_j)$ and $\underline{\boldsymbol{r}}_i = (\lambda_i, \phi_i)$. The data points calculated using this method produce a wide range of values. The $c(d)$ function is fit to match the mean of the values within bins corresponding to each distance value d_{ij}. The function is constrained such that it never exceeds one and satisfies the boundary conditions outlined in [50] (which may be derived from the requirement that a B matrix constructed using $c(d)$ must remain positive definite). Any function with a non-negative Hankel transform will meet these requirements. To fit the data, we select

$$c(d) = e^{-\frac{1}{2}a^2 d^2} \tag{4.28}$$

where a is a tunable scalable parameter. Matlab's `fittype` function was used to model the Hankel function in 4.28 and solve for a using the average deflection data. The resulting $c(d)$ equations for the EW and NS deflections are shown in Eqs. 4.29 and 4.30 and plotted with respect to the overall data and mean values in Figure 4.35.



(a) $c(d)_{\mathrm{EW}}$

(b) $c(d)_{\mathrm{NS}}$

Figure 4.35: Correlation functions $c(d)_{\mathrm{EW}}$ and $c(d)_{\mathrm{NS}}$ found using the Earth gravity model's EW and NS gravity deflections.

The equations use the tunable parameters $a_{\text{EW}} = 106.7851$ and $a_{\text{NS}} = 105.8159$ and closely approximate the mean values while adhering to the constraints for the $c(d)$ function.

$$c(d)_{\text{EW}} = e^{-\frac{1}{2}(106.7851)^2 d^2} \tag{4.29}$$

$$c(d)_{\text{NS}} = e^{-\frac{1}{2}(105.8159)^2 d^2} \tag{4.30}$$

4.7.3 Earth Gravity Surface Mission

The traversing missions on the Earth and Moon both used the EW and NS $c(d)$ equations to represent the correlation between the gravity vectors. The true initial position, $\underline{\boldsymbol{r}}_m$, for the Earth traversing mission began near the crater shown in Figure 4.32 at the following position:

$$\underline{\boldsymbol{r}}_m = \begin{bmatrix} -111.014^\circ \\ 35.0328^\circ \end{bmatrix} \tag{4.31}$$

The Earth satellite reference position, $\bar{\underline{\boldsymbol{z}}}_m$, was set to be within a specified distance of the defined position vector $\underline{\boldsymbol{r}}_m$, rather than being determined through satellite communication. This simplification was adopted to reduce complexity while still replicating the behavior of the satellite reference provided by the Lunar Pathfinder. The satellite reference was assumed to be accurate within 5 m of the true position, consistent with the reference accuracy at the APR described in Section 4.7.1. This uncertainty in the satellite reference position results in the covariance $\bar{R}_m$. Additionally, the uncertainty in the initial position was assumed to be 100 m. The associated satellite reference and initial position uncertainties are

$$\bar{R}_m = (4.49167 \times 10^{-5}\text{deg})^2 I_{2\times 2} \tag{4.32}$$

$$P_0 = (8.98334 \times 10^{-4}\text{deg})^2 I_{2\times 2} \tag{4.33}$$

A total of 15 waypoints were used throughout the mission. The spacing between each waypoint ranged from 0.00675° to 0.009546°, corresponding to distances of approximately 0.750 km and 1.0626 km, respectively. The time to traverse each segment was assumed to be 1000 s, resulting in average speeds between 0.7514 m/s and 1.063 m/s. Although these speeds are slower than the average human walking speed, they were selected to reflect the expected motion of a rover on the surface. The waypoints, along with their relationship to the initial and final positions, are illustrated in Figure 4.37.

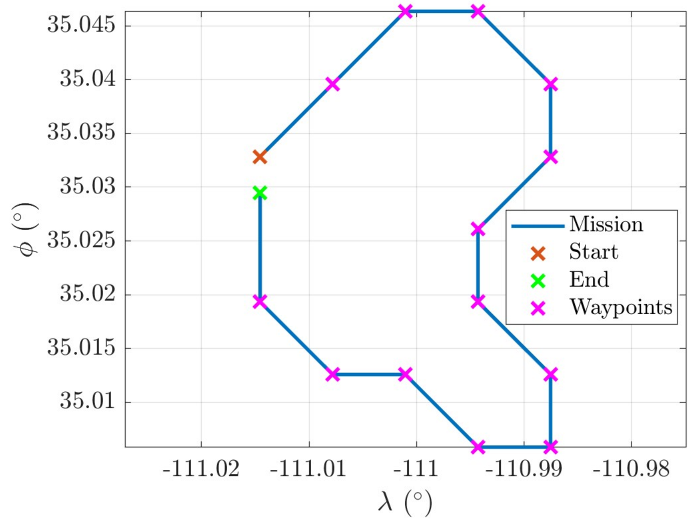


Figure 4.36: Earth gravity surface mission waypoints with respect to the start and end positions.

The waypoints shown in Figure 4.36 represent a diverse set of movements and emulate a typical mission trajectory by returning to the home base. The uncertainty in reaching the intended position after each segment is captured by Q_i, which denotes the process noise used in the bias/position estimation calculations. The value of Q_i between each step is modeled as a linear function of the distance traveled. In this study, it is assumed that for every 100 m traversed, the uncertainty region around the target location increases by approximately 5

m. The Q_i function used in this analysis is

$$Q_i(d_{ij}) = d(0.0025)I_{2\times 2} \tag{4.34}$$

where d_{ij} is the distance between the current and next estimated position. CelNav measurements were taken at each waypoint using the MEKF algorithm, with the rotation sequence defined previously and accelerometer noise set to $\sigma_{\text{accel}} = 1 \times 10^{-3}$ m/s^2. The MEKF used a time step of $\Delta t = 1$ s between each measurement update, collecting 900 samples over the course of its runtime. The MEKF's final $(\lambda_{\text{est}}, \phi_{\text{est}})$ estimates at each waypoint were used to define the $\underline{\boldsymbol{z}}$ vector in Eq.3.123, which contains the biased position measurements. The corresponding final output covariance from the MEKF served as the R_i measurement uncertainty matrix in Eq. 3.119. The parameter values used for the Earth traversing mission, as described above, are summarized in Table 4.11.

Table 4.11: Earth gravity surface mission variable definitions and uncertainty values used in correlated gravity calculation algorithm.

Variable	Value
$\sigma_{\text{b,EW}}$	$3.875\,96 \times 10^{-7\circ}$
$\sigma_{\text{b,NS}}$	$7.591\,13 \times 10^{-7\circ}$
$\underline{\boldsymbol{r}}_m$	$\begin{bmatrix} -111.014^\circ \\ 35.0328^\circ \end{bmatrix}$
$\bar{R}_m$	$(4.49167 \times 10^{-5}\text{deg})^2 I_{2\times 2}$
P_0	$(8.98334 \times 10^{-4}\text{deg})^2 I_{2\times 2}$
$Q_i(d_{ij})$	$d(0.0025)I_{2\times 2}$ deg
$\underline{\boldsymbol{z}}_i$	MEKF $(\lambda_{\text{est}}, \phi_{\text{est}})$
R_i	MEKF $P_{\lambda,\phi}$

Using these values, the correlated gravity method defined in Section 3.10 was performed to provide a state estimate, $\hat{\underline{\boldsymbol{r}}}$, and position bias, $\hat{\underline{\boldsymbol{b}}}$, due to errors in the LGM2011 gravity field. The resulting estimates over the entire mission are shown in Figure 4.37.

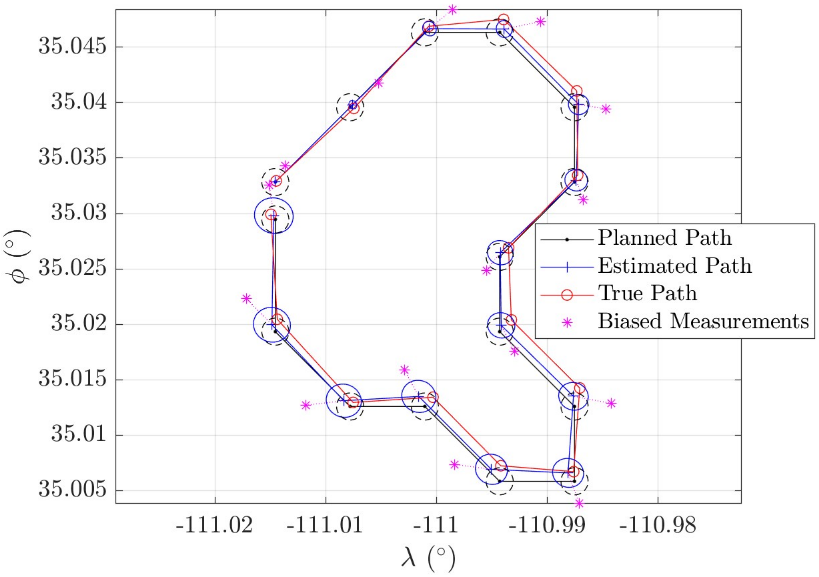


Figure 4.37: Position and bias results and their corresponding uncertainties for the Earth gravity surface mission.

Figure 4.37 shows that the estimated mission trajectory closely follows the true position values. The planned path, shown in black, represents the intended trajectory prior to the mission. The true path, shown in red, deviates slightly from the desired trajectory due to accumulated error from traversing over large distances. The process noise, Q_i, associated with traversing uncertainty, is illustrated as dashed black circles around each desired waypoint. These envelopes represent the uncertainty region through which the user moves from one location to the next. The estimated path, shown in blue, closely tracks the true path, with deviations resulting from biased measurements calculated using the MEKF. The uncertainty in these estimates is represented by a dashed blue circle. Can see that the error grows as more measurements are taken over time. The positional bias is shown as a dotted magenta line, representing the relative error introduced by inaccuracies in the LGM2011 gravity field. Knowledge of these biases is valuable for future navigation, as they can be used to correct the estimated position and yield more accurate localization.

This Earth gravity surface mission validates the navigation procedure with correlated bias correction described in Section 3.10, demonstrating that the method is effective when a highly accurate surface gravity model is available. While the LGM2011 model provides accuracy at a 1.5 km scale, the results suggest that as lunar gravity models improve in resolution and accuracy, the correlated gravity bias method will become increasingly effective for providing accurate position estimates and bias corrections.

4.7.4 Moon Gravity Surface Mission

The same general processes used for the Earth traversing mission were used on the Moon. The primary changes between the two are the location and gravity models used. The waypoints used starting from the APR are shown in Figure 4.38.

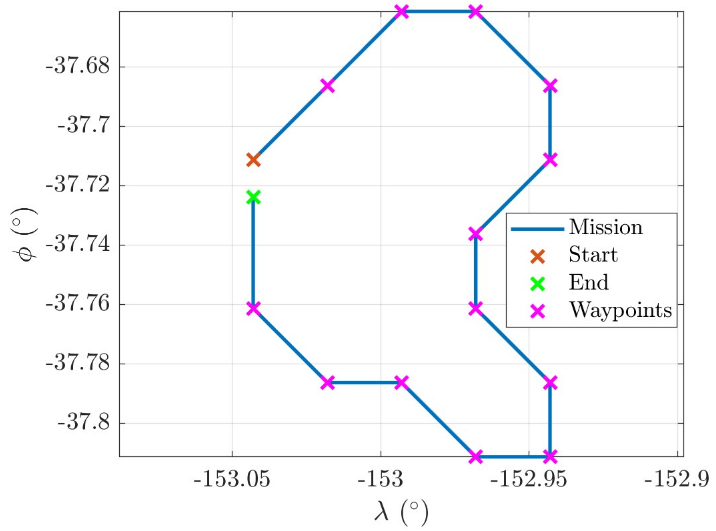


Figure 4.38: Moon gravity surface mission waypoints starting from the APR and ending near the start location.

The mission follows a path with the same shape as the Earth mission and maintains the same distance and speed ranges. The procedure for traversing between waypoints and collecting biased CelNav measurements remains unchanged. However, the parameter uncertainties differ for the Moon-based traversing mission due to the change in planetary radius and the use of an actual calculated reference position. The updated uncertainty and parameter values, along with the resulting position and bias estimates, are presented in Table 4.12.

Table 4.12: Moon gravity surface mission variable definitions and uncertainty values used in correlated gravity calculation algorithm.

Variable	Value
$\sigma_{\text{b,LGM}}$	$8.333\,33 \times 10^{-3\circ}$
$\underline{\boldsymbol{r}}_m$	$\begin{bmatrix} -153.043^\circ \\ -37.7115^\circ \end{bmatrix}$
$\bar{R}_m$	$(1.64823 \times 10^{-4}\text{deg})^2 I_{2\times 2}$
P_0	$(3.29646 \times 10^{-3}\text{deg})^2 I_{2\times 2}$
$Q_i(d_{ij})$	$d(0.0025) I_{2\times 2}$ deg
$\underline{\boldsymbol{z}}_i$	MEKF $(\lambda_{\text{est}}, \phi_{\text{est}})$
R_i	MEKF $P_{\lambda,\phi}$

The value of $\sigma_{\text{b,LGM}}$ in Table 4.12 is calculated using the LGM2011's specification that its deflections are generally accurate to approximately 30”. The position results for the surface mission using the LGM2011 lunar gravity are shown in Figure 4.39.

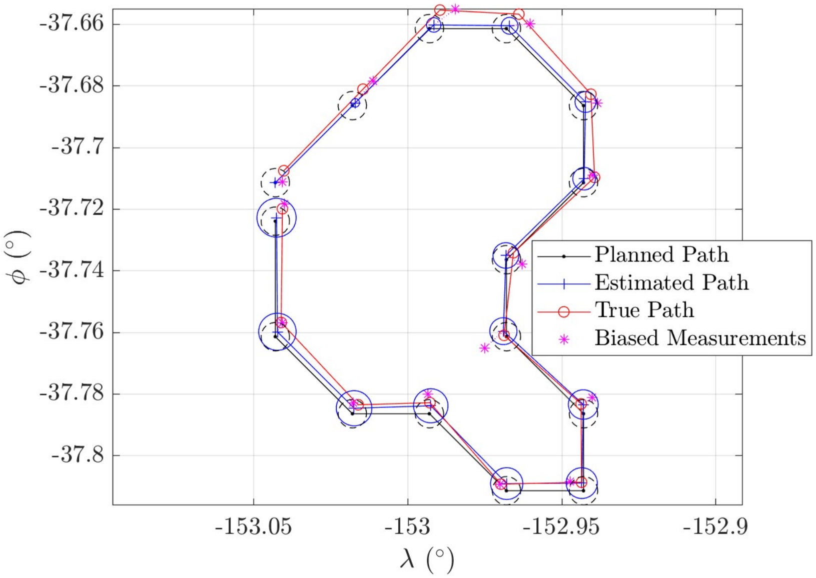


Figure 4.39: Position and bias results and their corresponding uncertainties for the Moon gravity surface mission.

As with the Earth traversing mission, the true position values on the Moon deviate from the desired path due to process noise associated with navigating between waypoints. The estimated positions shown in Figure 4.39 closely match the true values, validating the accuracy of the correlated gravity algorithm.An interesting observation, distinct from the Earth mission, is that the position errors shown in Figure 4.39 are smaller than those in Figure 4.37. This indicates that the position estimates on the Moon are comparatively more accurate than those on Earth. This is an encouraging result, given that the methodology in this study is intended for lunar application. Both the Earth and Moon gravity surface missions demonstrate the capability of the correlated gravity method to estimate position and bias given a known gravity field. The calculated biases may be applied in future navigation missions to refine location estimates based on the user's relative position on the lunar surface. This chapter presented various Monte Carlo and test case simulations to evaluate the accuracy and behavior of the algorithms introduced in Chapter 3. Several locations across the lunar surface were used to compare the performance between pure rotation and libration covariance propagation methods. Monte Carlo simulations were also conducted for different rotation angle combinations used in the accelerometer bias estimation process were tested to determine which configuration yielded the most accurate accelerometer bias estimates. The sensitivity of the MEKF to different initial state values was examined to identify any divergent behavior in the state estimation. A standard Monte Carlo simulation at a single location was also performed to analyze how measurement randomness influenced the final position estimates. Another Monte Carlo study was conducted to assess how varying accelerometer noise levels affected the MEKF's final position and bias estimates. Finally, missions involving traversing on the surface using both Earth and lunar gravity data were presented to validate the correlated gravity bias estimation algorithm described in Section 3.10.

Chapter 5

Conclusions

This chapter discusses the primary results and findings of this thesis. It also addresses the limitations inherent in both the methodology and their effects on the final results. Lastly, suggestions for future work to expand upon the research presented in this thesis are provided.

5.1 Motivation

As humanity strives to return to the Moon, new navigation techniques are needed to support a permanent presence on the lunar surface. This study references a known gravity field model to compare with locally measured gravity values in order to estimate an accurate longitude and latitude position (λ, ϕ). A star tracker (ST) is also used to determine the system's attitude via the QUEST algorithm. Various lunar gravity models were analyzed to identify which one is best suited for compatibility with gravity vector filtering using a multiplicative extended Kalman filter (MEKF). The models evaluated include GLGM-2, LP100K, GRGM1200A, and LGM2011. The first three models rely on spherical harmonics of varying degree and order. Both the GLGM-2 and LP100K models suffer from inaccuracies due to their low-degree, low-order spherical harmonic calculations and the lack of farside lunar gravity data. The GRGM1200A model is derived from coefficients obtained during the GRAIL mission. While this model is relatively accurate, it is only valid outside the Brillouin sphere surrounding the Moon. The only feasible gravity model that includes gravity deflection across the lunar surface and is specifically designed for surface use is the LGM2011 model. This model provides data on the EW and NS deflections of the gravity vector, along with the associated surface gravity magnitude. These data were used to calculate local gravity vectors across the Moon's surface in the MCMF frame, providing the needed gravity field reference.

Other lunar surface navigation techniques were also examined to identify potential improvements to existing methods. The first method studied was the alternative CelNav technique developed by Thein, Quinn, and Folta in [18]. This method was primarily designed for navigation inside lunar craters and does not use any external reference gravity fields. It also failed to account for dynamic bias in accelerometer-based gravity measurements. Another approach combined INS data with CelNav measurements in an EKF to estimate the state of the ground system [19]. However, this method experienced drift over time due to the accumulation of INS errors and was primarily effective for system initialization before dynamic rover missions. An Unscented Kalman Filter (UKF) approach combining INS and CelNav data was also explored [21]. This method aimed to reduce nonlinear estimation errors and avoid the need to compute Jacobian matrices, as required in EKFs. Like the previous method, it is generally useful only for initialization purposes. Additional attitude estimation techniques using sun vectors in place of star trackers were also investigated [23]. Based on the techniques reviewed, it is evident that no current algorithms that combine a known gravity field with ST measurements to provide a complete attitude and position solution that account for biases and gravity field complexity.

5.2 Methods Used

Chapter 3 described the algorithms used to calculate the system state position and accelerometer bias values, as well as the method to identify position biases resulting from potential gravity errors in the LGM2011 model. The system attitude was determined using Shuster's QUEST algorithm, which minimizes the attitude matrix, $A(\bar{q})$, using measured starlight vectors. The output is the optimal quaternion representing the system's attitude.

The MEKF was used to estimate the state composed of the attitude error vector, $\underline{\boldsymbol{a}}$, between the estimated and reference quaternions, the (λ, ϕ) position, and the dynamic accelerometer bias, $\underline{\boldsymbol{b}}_a$. The MEKF is preferred over the traditional EKF because the EKF cannot adequately handle the four-component quaternion error in state estimation. Instead, $\underline{\boldsymbol{a}}$ is used to represent the quaternion error in three components. The nonlinear measurement model was defined using both the attitude error vector and a gravity measurement model. The gravity model incorporates random noise from the accelerometer as well as the dynamic bias present in the accelerometer measurements. The relation between the measured and true gravity vectors depends on the MEKF's ability to estimate the accelerometer bias. State propagation was defined as static, given the system is stationary on the lunar surface. This definition was able to be made since the reference quaternion $\bar{q}_{\text{ref}}$ was updated nonlinearly.

Covariance propagation was analyzed under two scenarios: pure rotation about the Moon's z axis and lunar libration over time. Only the attitude error vector components of the covariance matrix were affected between time steps, as the only change in the system was quaternion attitude rotation. The high-fidelity lunar orientation method utilized the Moon's Euler angles from the Jet Propulsion Laboratory's (JPL) DE405 ephemeris database to model its rotation, resulting in more accurate state estimates.

The dynamic accelerometer bias was estimated using a procedure involving two system rotations. The process began with the MEKF running in the initial orientation. Then, the system was rotated about the z axis for a second MEKF run, followed by a rotation about the x axis for a final run. This sequence of rotations enabled effective convergence of the accelerometer bias estimates to their true values.

Estimation with correlated gravity biases was also developed. This required a highly-accurate reference position, which was determined using the Lunar Pathfinder satellite's two-way ranging measurements with the ground system. This approach was chosen for its applicability in limited satellite exposure scenarios. The reference was calculated at the Apollo Peak Ring location. The strength and reliability of the ranging measurements were characterized by the total uncertainty, $\sigma_{\mathrm{p,tot}}$, which includes both satellite ephemeris errors, $\sigma_{\mathrm{p,eph}}$, and non-ephemeris signal transmission errors, $\sigma_{\mathrm{non-eph}}$.

These ranging measurements were used to calculate the final state position through a combined weighted batch filter–EKF algorithm. The batch filter collected one hour of ranging data and applied nonlinear weighted least squares to solve for the state estimate. This output served as the EKF's initial state estimate. The initial covariance was computed using the final ranging measurement's $\sigma_{\mathrm{p,tot}}$ and PDOP. With these initial conditions, the EKF used sequential ranging data to estimate the final state within 5 m of the true Apollo Peak Ring position. This reference and its uncertainty define the correlated gravity bias parameters, $\underline{\boldsymbol{r}}_m$ and $\bar{R}_m$.

Estimation with correlated gravity biases then used biased position estimates, $\underline{\boldsymbol{z}}_i$, from the MEKF and the accurate reference position described above to solve for the estimated state and local gravity field bias during a walking mission on the lunar surface. The biases in the gravity field were assumed to be a function of the average bias uncertainty, σ_b, between gravity vectors in the local navigation region. The expectation of the gravity field bias was defined to be zero, with the covariance modeled as a function of σ_b and a correlation function,

$c(d)$, of the distance between positions. This function, $c(d)$, was determined using the EW and NS gravity deflections.

Since the LGM2011 model only includes data down to 1.5 km resolution, gravity models with 1.5 km and 30 m resolutions were generated for Earth using the SGG-UGM-2 model. The 30 m resolution data were nested within one grid cell of the 1.5 km model to analyze correlation error. Differences in the EW and NS deflections between high- and low-resolution models were computed at each high-resolution grid point and defined as $\Delta\delta_{(\lambda_i,\phi_i)}$. The average bias σ_b was defined as $\mathbb{E}\{\Delta\delta_{(\lambda_i,\phi_i)}\Delta\delta_{(\lambda_i,\phi_i)}\}$ for $i = 1, \ldots, N$, where N is the number of coordinate points. Products of deflection differences at all position pairs, $\Delta\delta_{(\lambda_i,\phi_i)}\Delta\delta_{(\lambda_j,\phi_j)}$, were calculated, normalized by σ_b^2, and plotted against the distances d_{ij} between coordinate pairs. The mean values within each distance bin were computed and plotted at bin centers. These data were fit using the Hankel transform function, $e^{-\frac{1}{2}a^2 d_{ij}^2}$, to determine the tunable parameter a. This $c(d)$ function was calculated for both EW and NS deflections.

Next, the process noise Q_i was defined to reflect how accurately the astronaut or rover could reach each new waypoint in the walking mission. The uncertainty was modeled as a linear function of distance traveled, assuming an error of 5 m for every 100 m. A least-squares cost function was then formulated using the biased measurements, correlated gravity biases, and the reference position. Derivatives of the cost function were taken with respect to both position and gravity bias, yielding equations for the final estimates and their covariances.

5.3 Key Results

The methods summarized above provide a comprehensive framework for navigation on the lunar surface. This section presents key results from test cases used to evaluate the performance of the proposed techniques.

The first test case involved randomly selecting 5000 points across the lunar surface and running the MEKF algorithm using both pure rotation and lunar libration for covariance propagation. The final position and accelerometer bias estimates at each location were compared and averaged between the two methods, with the lunar libration propagation producing better results. This outcome is expected, as the Moon does not rotate purely on its axis. Failing to account for the “wobble” leads to increased error in attitude propagation between time steps. This case also showed that the state estimate’s errors increased with proximity to craters and other topographic features.

The next test case examined which rotation angle values for the z axis and x axis used in the accelerometer bias estimation produced the most accurate results. Six different angle values for each axis (36 rotation combinations) were evaluated using a 50-run Monte Carlo simulation per combination. The results were averaged for each case, revealing that the 90°-90° z axis and x axis rotation sequence yielded the most accurate accelerometer bias estimates. Most of the correction occurred after the z axis rotation, with slight improvements contributed by the x axis rotation.

The influence of the MEKF’s initial state estimate on final results was also analyzed to assess any potential divergence when the initial guess is far from the true position. Initial guess positions, $\underline{\boldsymbol{x}}_0$, ranging from approximately 0 km to 22 km from the true position were tested. For each initial guess, a 150-run Monte Carlo simulation was executed using the MEKF with the 90°-90° rotation sequence. The results showed that the final position estimates were not significantly affected over the initial guess range. A slight increase in accelerometer bias estimates was observed at an initial guess approximately 7 km from the true position, but the change was negligibly small. Based on lunar mission planning assumptions, it is expected that the astronaut or rover will know their location within 6 km of the desired position, defining the maximum $\underline{\boldsymbol{x}}_0$ error used in all MEKF runs.

A 1000 −run Monte Carlo simulation was then performed at the Apollo Peak Ring to assess the impact of random gravity noise on final position estimates. The haversine distances between the true and estimated positions were computed, revealing that the distance error followed a χ distribution with $k = 2$ degrees of freedom, consistent with the presence of two sources of error: λ_{err} and ϕ_{err}. The average distance error was 66.7988 m, with minimum and maximum values of 0.920 929 m and 196.262 m, respectively. The wide range of errors reflects the variation in quality of the random gravity noise. Accelerometer bias estimates showed a similar trend, which is expected since the bias affects the gravity measurements, and therefore the position estimates.

The correlated gravity field bias correction algorithm was then evaluated using test cases with both Earth and Moon gravity fields, employing the same $c(d)$ correlation functions in both cases (which was inferred from the more-precise Earth surface data). The lunar mission began at the Apollo Peak Ring, and the Earth mission started near the Meteor Crater National Landmark, chosen for its similarity to lunar terrain. Both missions followed similar paths, departing from its initial position and later returning close to the initial start location. The state estimates in both cases were comparably accurate, though a noticeable difference in gravity field bias was observed. The Moon case exhibited significantly lower gravity field bias than the Earth case, indicating more accurate position estimates. Overall, the tests confirmed that the correlated gravity bias algorithm can produce accurate results when provided with $c(d)$, a known gravity field, a highly accurate reference position, and biased CelNav measurements from the MEKF.

Lastly, four different distributions of noise in measured gravity values were tested to evaluate their influence on the final MEKF state estimates. These distributions ranged from high to very low noise levels. As expected, large noise in gravity measurements produced poor final position estimates, averaging around 572 m. The two lower noise levels yielded significantly improved position accuracy, with average errors of approximately 23 m and 32 m. Interestingly, the smallest noise level resulted in slightly worse performance, suggesting a limit to the accuracy gains achievable through improving accelerometer accuracy. These findings demonstrate that MEKF accuracy is ultimately constrained by the quality of the accelerometer's gravity measurements.

5.4 Limitations

There are limitations to the models and algorithms presented in this thesis. To begin, the LGM2011 model is assumed to be sufficiently accurate for use in the MEKF algorithm. It incorporates data from SELENE, LOLA, and Newtonian gravity to estimate surface gravity magnitudes and deflections of the gravity vector. Although this model is assumed to be accurate down to the spatial scales provided, Table 2.1 shows Apollo measurements that contradict the values estimated by LGM2011. If the astronaut or rover reaches the lunar surface and the true gravity at their current location deviates significantly from the LGM2011 model, it can have a substantial impact on the MEKF's accuracy. The correlated gravity bias method aims to mitigate this issue, but some inaccuracies will persist due to deviations from the LGM2011 model; in addition, the statistical model used for the correlated bias correction may not be correct or properly tuned to capture the true errors. .

Second, time is not explicitly considered in either the MEKF or satellite reference position algorithms. Since the ground system is stationary, it is assumed that the receiver can perform two-way ranging with the satellite, eliminating the need for clock synchronization between the satellite and ground system. Although this assumption is also made in [46], including clock synchronization could potentially lead to slightly more accurate position estimates. In terms of the MEKF, time is not considered because the only time-dependent factor is the

Moon's rotation rate, which is assumed to be known with high accuracy. In addition, the Moon's rotation rate is very small as compared to the timescales of feasible clock accuracy. Therefore, any timing error results in negligibly small errors in the final position estimates.

The true accelerometer bias is assumed to remain constant throughout the duration of a single CelNav cycle (15 minutes). However, this assumption does not fully capture the behavior observed in real-world scenarios. In practice, the accelerometer bias drifts randomly over time relative to its initial value. This could be addressed in future work by modeling the true dynamic bias as a function that continuously introduces random perturbations to the initial bias value.

There are also gravitational assumptions that are made that could influence gravity measured by the accelerometer. This study does not include tidal or other accelerations in the measured gravity. The gravity is assumed to solely measure the pure lunar surface value. Another assumption made is that the measurements take place at $h = 0$ m altitude on the surface. There could be instances in actual use where the user is on a hill or other feature above the surface assumption that could lead to deviations in the measured gravity in comparison to the LGM2011 gravity model.

Lastly, it is assumed that both the ST and accelerometer are aligned with the system's body frame, with no alignment errors. Neglecting such alignment errors could introduce inaccuracies in the attitude quaternion produced by the QUEST algorithm and in the gravity vector measured by the accelerometer. As demonstrated earlier, inaccurate accelerometer measurements significantly impact MEKF accuracy. These alignment errors can be accounted for by incorporating rotation matrices that define the transformations from the body frame to the true ST and accelerometer frames. Errors in these matrices could also be accounted for with appropriate additions to the correlated bias correction methodology.

5.5 Potential Future Work

The first potential direction for future work relates to possible errors in the LGM2011 model after measuring true gravity values on the lunar surface. Given the correlation function, $c(d)$, calculated using Earth gravity data, assumptions can be made about the relationships between neighboring gravity vectors. Once a true gravity value is measured on the surface, the LGM2011 gravity model can be adjusted to reflect these new measurements. In other words, the model can be refined as additional true gravity data becomes available. Defining a method to smooth this relationship and optimize the updating procedure for the gravity model could be a promising topic of future research. This approach could also lead to new techniques for generating lunar gravity models by identifying how discrepancies between estimated and true values arise within the current estimation schemes. Multiple lunar missions could contribute to a highly accurate lunar gravity model through the combination of measured values and their integration with the existing model.

Another avenue for future work involves incorporating a dynamic component into the model, enabling measurements while the system is in motion. This would require new assumptions and equations to account for the state and covariance propagation between estimates. Such a dynamic model could improve the accuracy of the correlated gravity bias calculations by providing a better understanding of the final position error upon reaching a waypoint, rather than assuming a linear trend based solely on distance traveled.

Current methods already employ visual-based navigation systems (VNS) to assist with attitude and position estimation on the lunar surface. However, there is currently no approach that integrates a ST, accelerometer, known gravity field, and VNS to compute final position and attitude estimates. Investigating whether the addition of a VNS component could improve estimation accuracy would be of interest. Naturally, the effectiveness of the VNS system depends on lighting conditions, which are highly variable on the lunar surface.

The methods presented in this study are susceptible to singularities at the poles, leading to an increase in position error if solely calculating the λ and ϕ in the state. A source of future work could be to convert the methods shown into a pure position vector representation so that it can be used more accurately at polar locations. This could be really important since there is the desire to land on the lunar South Pole, so having an alternative that avoids computation issues and singularities could enhance the feasibility of the MEKF algorithm.

Lastly, future research could explore the use of multiple systems implementing the MEKF method outlined in this thesis to define a triangulated region for improved surface navigation accuracy. These systems could serve as beacons for ascent navigation, real-time kinematic (RTK) position correction, and other related applications. It would also be interesting to investigate how the accuracy of the correlated gravity bias estimates changes as the number of lunar satellites capable of ranging to the system increases.

5.6 Conclusion

The work presented in this thesis has introduced algorithms capable of estimating a system's position, attitude, accelerometer bias, and a highly accurate reference position on the lunar surface using a ST, accelerometer, and a predefined gravity field. The accuracy of these algorithms, as well as their response to various test conditions, was evaluated and used to assess the performance of the overall navigational framework provided. Position and accelerometer bias estimates were computed across the lunar surface, offering insight into how well the methods perform near topographic features such as craters. It was shown that noise in the gravity measurements significantly affects the state estimates, highlighting the need for a highly accurate accelerometer. The correlated gravity bias algorithm was demonstrated to effectively account for errors in the LGM2011 model, potentially improving future navigation in regions where these biases are present.

The algorithms presented in this thesis could address a wide range of needs in future lunar surface navigation, particularly as humanity aims to return to the Moon for sustained missions. This work can also serve as a foundation for future research on advanced lunar navigation techniques.

Bibliography

[1] NASA. Moon to mars architecture - strategy and objectives. https://www.nasa.gov/moontomarsarchitecture-strategyandobjectives/, 2024.

[2] NASA. Artemis. https://www.nasa.gov/feature/artemis/, 2024.

[3] NASA. Gateway. https://www.nasa.gov/mission/gateway/, 2024.

[4] Firefly Aerospace. Firefly aerospace successfully completes 14 days of surface operations on the moon. https://fireflyspace.com/news/firefly-aerospace-successfully-completes-14-days-of-surface-operations-on-the-moon/ 2024.

[5] Space Insider. Intuitive machines lunar lander tips over but collects data for nasa before power loss. https://spaceinsider.tech/2025/03/08/intuitive-machines-lunar-lander-tips-over-but-collects-data-for-nasa-before-power-l 2025.

[6] Malcolm D. Shuster. New quests for better attitudes. In *Proceedings of the Flight Mechanics/Estimation Theory Symposium*, pages 125–137, Greenbelt, Maryland, May 1991. NASA Goddard Space Flight Center.

[7] Carlos M. Roithmayr. Contributions of spherical harmonics to gravitational moment. *AIAA Journal*, 57(10):4291–4299, 2019.

[8] Blažej Bucha, Christian Hirt, and Michael Kuhn. Divergence-free spherical harmonic gravity field modelling based on the runge–krarup theorem: a case study for the moon. *Journal of Geodesy*, 93(4):489–513, 2019.

[9] Michal Šprlák and Shin-Chan Han. On the use of spherical harmonic series inside the minimum brillouin sphere: Theoretical review and evaluation by grail and lola satellite data. *Earth-Science Reviews*, 222:103739, 2021.

[10] Frank G. R. Lemoine, David E. Smith, Maria T. Zuber, Gregory A. Neumann, and David D. Rowlands. A 70th degree lunar gravity model (glgm-2) from clementine and other tracking data. *Journal of Geophysical Research: Planets*, 102(E7):16339–16359, 1997.

[11] A.S. Konopliv, S.W. Asmar, E. Carranza, W.L. Sjogren, and D.N. Yuan. Recent gravity models as a result of the lunar prospector mission. *Icarus*, 150(1):1–18, 2001.

[12] Tom L. Hoffman. Grail: Gravity mapping the moon. In *2009 IEEE Aerospace conference*, pages 1–8, 2009.

[13] Erwan Mazarico. Lunar gravity field: Grgm1200a. `https://pgda.gsfc.nasa.gov/products/50`. Accessed: 2025-03-15.

[14] C. Hirt and W.E. Featherstone. A 1.5km-resolution gravity field model of the moon. *Earth and Planetary Science Letters*, 329-330:22–30, 2012.

[15] NASA National Space Science Data Center (NSSDC). Kaguya. `https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2007-039A`, 2025.

[16] NASA Science. Lunar orbiter laser altimeter (lola). `https://science.nasa.gov/mission/lro/lola/`, 2025. Accessed: March 15, 2025.

[17] Curtin University. Lgm2011 data. `https://ddfe.curtin.edu.au/models/LGM2011/data/`, 2015.

[18] May-Win Thein, David Quinn, and David Folta. *Celestial Navigation (CelNav): Lunar Surface Navigation*.

[19] Xiaolin Ning and Lingling Liu. A two-mode ins/cns navigation method for lunar rovers. *IEEE Transactions on Instrumentation and Measurement*, 63(9):2170–2179, 2014.

[20] Claudio Urrea and Rayko Agramonte. Kalman filter: Historical overview and review of its use in robotics 60 years after its creation. *Journal of Sensors*, 2021(1):9674015, 2021.

[21] Xiaolin Ning, Longhua Wang, Weiren Wu, and Jiancheng Fang. Autonomous initialization of lunar rover navigation using celestial and inertial information. *Sensors (Basel, Switzerland)*, 11(7):6991–7003, July 2011.

[22] Eric A. Wan and Rudolph van der Merwe. *The Unscented Kalman Filter*, chapter 7, pages 221–280. John Wiley Sons, Ltd, 2001.

[23] Peng Yang, Li Xie, and Jilin Liu. Simultaneous celestial positioning and orientation for the lunar rover. *Aerospace Science and Technology*, 34:45–54, 2014.

[24] Li Xie, Peng Yang, Thomas Yang, and Ming Li. Dual-ekf-based real-time celestial navigation for lunar rover. *Mathematical Problems in Engineering*, 2012(1):578719, 2012.

[25] Yan-Bin Jia. Quaternions*. Unpublished lecture notes, September 2024. Lecture Notes, COMS 4770/5770, Iowa State University.

[26] Christopher J. Aceto. Modeling and analysis of a thermospheric density measurement system based on torque estimation. Master's thesis, Virginia Polytechnic and State University, Blacksburg, United States, 2023.

[27] Riley M. Fitzgerald. Quaternion kalman filtering and initialization. Unpublished notes, 2020, 2020.

[28] William Rowan Hamilton. On quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(163):10–13, 1844.

[29] Itzhack Bar-Itzhack. New method for extracting the quaternion from a rotation matrix. *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM*, 23, 11 2000.

[30] Kitt Peak National Observatory. Bright star catalog-5th edition, 2015.

[31] Goddard Space Flight Center. Bsc5p - bright star catalog, 2024.

[32] Caitlin Ahrens. *Inertia, Inertial Frame*, pages 329–331. Springer International Publishing, Cham, 2023.

[33] F. Landis Markley. Multiplicative versus additive filtering for spacecraft attitude determination. In *International Conference on Control of Systems and Structures in Space*, December 2003.

[34] MathWorks. Cubic Spline Interpolation with End Conditions - MATLAB csape, 2024. Accessed: Mar. 5, 2025.

[35] Donald H. Eckhardt. Theory of the libration of the moon. *The Moon and the Planets*, 25(1):3–49, August 1981.

[36] E. Wright. Moon essentials: Libration in latitude. NASA Scientific Visualization Studio, June 24 2024.

[37] E. Wright. Moon essentials: Libration in longitude. NASA Scientific Visualization Studio, June 24 2024.

[38] Jpl planetary and lunar ephemerides. ssd.jpl.nasa.gov.

[39] D. B. Taylor, S. A. Bell, J. L. Hilton, and A. T. Sinclair. Computation of the quantities describing the lunar librations in the astronomical almanac. Nao technical report, H.M. Nautical Almanac Office, 2010.

[40] E. Standish. Jpl planetary and lunar ephemerides, de405/le405. Memorandum, Jet Propulsion Laboratory (JPL), August 26 1998.

[41] E. Standish and J. Williams. *CHAPTER 8: Orbital Ephemerides of the Sun, Moon, and Planets*. University Science Books, 2006.

[42] X. X. Newhall and J. G. Williams. Cel. mech. and dyn. ast. *Celestial Mechanics and Dynamical Astronomy*, 66:21, 1996.

[43] A. A. Zagidullin, V. S. Usanin, N. K. Petrova, Yu. A. Nefedyev, A. O. Andreev, and T. V. Gudkova. Physical libration of the moon: An extended problem. *Astronomy Reports*, 64(12):1093–1106, 2020.

[44] Renato Zanetti and Kyle J. DeMars. Joseph formulation of unscented and quadrature filters with application to consider states. *Journal of Guidance, Control, and Dynamics*, 36(6):1860–1864, 2013.

[45] Pedro Batista, Carlos Silvestre, Paulo Oliveira, and Bruno Cardeira. Accelerometer calibration and dynamic bias and gravity estimation: Analysis, design, and experimental evaluation. *IEEE Transactions on Control Systems Technology*, 19(5):1128–1137, 2011.

[46] Marta Cortinovis, Tara Mina, and Grace Gao. Assessment of single satellite-based lunar positioning for the nasa endurance mission. In *2024 IEEE Aerospace Conference*, pages 1–11, 2024.

[47] Floor Melman, Paolo Zoccarato, Csilla Orgel, Richard Swinden, Pietro Giordano, and

Javier Ventura-Traveset. Lcns positioning of a lunar surface rover using a dem-based altitude constraint. *Remote Sensing*, 14, August 2022.

[48] Wei Liang, Jiancheng Li, Xinyu Xu, Shengjun Zhang, and Yongqi Zhao. A high-resolution earth's gravity field model sgg-ugm-2 from goce, grace, satellite altimetry, and egm2008. *Engineering*, 6(8):860–878, 2020.

[49] International Association of Geodesy. Global gravity field models, 2024.

[50] P Schneider and J Hartlap. Constrained correlation functions. *Astron. Astrophys.*, 504(3):705–717, September 2009.

[51] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits: Models, Methods, and Applications*. 2000.

[52] Leland E. Cunningham. On the computation of the spherical harmonic terms needed during the numerical integration of the orbital motion of an artificial satellite. *Celestial Mechanics*, 2(2):207–216, 1970.

Appendices

Appendix A

Spherical Harmonics Gravity Expansion

Section 2.1.1 provides a brief description of spherical harmonics and how they are used to develop common gravity models. Montenbruck and Gill provide a recursive method to calculate the spherical harmonic gravity field based on given normalized gravitational coefficients $\overline{C}_{n,m}$ and $\overline{S}_{n,m}$ [51]. Their analysis of the spherical harmonic gravity field is based on the derivation provided by Cunningham in [52], which provides similar formulas in a complex form with a different normalization. The final form of the spherical harmonic potential expansion is defined to be

$$V = -\frac{\mu}{r}\left\{1 + \sum_{n=2}^{\infty}\left(\frac{R}{r}\right)^{n}\left[\bar{J}_n P_{n,0}(\sin\phi) + \sum_{m=0}^{n}\left(\bar{C}_{n,m}\cos(m\lambda) + \bar{S}_{n,m}\sin(m\lambda)\right)P_{n,m}\sin(\phi)\right]\right\} \tag{A.1}$$

where μ is the gravitational constant, R is the radius of the planetary body (the Moon in this study), r, ϕ, and λ are the distance, latitude, and longitude respectively in the body fixed frame, and $\bar{J}_n$, $\bar{C}_{n,m}$, $\bar{S}_{n,m}$, and $P_{n,m}(\zeta)$ are defined below with the $P_{n,m}(\zeta)$ being the associated Legendre function of the n, m, and ζ inputs:

$$\bar{J}_n = -\frac{J_n}{\mu R^n} \tag{A.2}$$

$$\bar{C}_{n,m} = -\frac{C_{n,m}}{\mu R^n} \tag{A.3}$$

$$\bar{S}_{n,m} = -\frac{S_{n,m}}{\mu R^n} \tag{A.4}$$

$$P_{n,m}(\zeta) = \frac{(1-\zeta^2)^{m/2}}{2^n n!} \frac{d^{n+m}}{d\zeta^{n+m}}(\zeta^2 - 1)^n = (-1)^m P_n^m(\zeta) \tag{A.5}$$

The n and m values above define the degree and order of the spherical harmonic term respectively. Based on the definition of the gravitational potential, Montenbruck and Gill define recurrence relations to find matrices $V_{n,m}$ and $W_{n,m}$ based on Cunningham's definition. The first recurrence relation for $n = m$ is defined as

$$V_{n,n} = (2n-1)\left\{\frac{xR}{r^2}V_{n-1,n-1} - \frac{yR}{r^2}W_{n-1,n-1}\right\} \tag{A.6}$$

$$W_{n,n} = (2n-1)\left\{\frac{xR}{r^2}W_{n-1,n-1} + \frac{yR}{r^2}V_{n-1,n-1}\right\} \tag{A.7}$$

The (x, y, z) values shown in the above and following harmonic equations are the desired position in the MCMF frame. The initial conditions of these matrices are $V_{0,0} = \frac{R_{moon}}{r}$ and $W_{0,0} = 0$. The second recurrence relations for n not equal to m are

$$V_{n,m} = \frac{(2n-1)}{(n-m)}\frac{zR_{\text{moon}}}{r^2}V_{n-1,m} - \frac{(n+m-1)}{(n-m)}\frac{R_{\text{moon}}^2}{r^2}V_{n-2,m} \tag{A.8}$$

$$W_{n,m} = \frac{(2n-1)}{(n-m)}\frac{zR_{\text{moon}}}{r^2}W_{n-1,m} - \frac{(n+m-1)}{(n-m)}\frac{R_{\text{moon}}^2}{r^2}W_{n-2,m} \tag{A.9}$$

where a special case for $n = m + 1$ must be defined as

$$V_{n,m} = \frac{(2n-1)}{(n-m)} \frac{zR_{\text{moon}}}{r^2} V_{n-1,m} \tag{A.10}$$

$$W_{n,m} = \frac{(2n-1)}{(n-m)} \frac{zR_{\text{moon}}}{r^2} W_{n-1,m} \tag{A.11}$$

Unlike the Cunningham derivation, the Montenbruck and Gill recurrence relations require normalization coefficients to properly compute the gravity using GRAIL $C_{n,m}$ and $S_{n,m}$ coefficients (note lack of overbars). There are many different spherical harmonic normalizations, but the normalization factor shown in Eq. A.12 is commonly used in Geodesy disciples and takes the degree and order as inputs.

$$f_{n,m} = \sqrt{\frac{(2n+1)(n-m)!}{(n+m)!}} \tag{A.12}$$

From the recurrence relations in Eqs. (A.6-A.11), the following factors are needed:

$$\bar{f}_1 = \frac{f_{n,n}}{f_{n-1,n-1}}, \qquad \bar{f}_2 = \frac{f_{n,m}}{f_{n-1,m}}, \qquad \bar{f}_3 = \frac{f_{n,m}}{f_{n-2,m}}, \tag{A.13}$$

Substituting back into the first recurrence relations gives

$$V_{n,n} = (2n-1) \left\{ \bar{f}_1 \frac{xR_{\text{moon}}}{r^2} V_{n-1,n-1} - \bar{f}_1 \frac{yR_{\text{moon}}}{r^2} W_{n-1,n-1} \right\} \tag{A.14}$$

$$W_{n,n} = (2n-1) \left\{ \bar{f}_1 \frac{xR_{\text{moon}}}{r^2} W_{n-1,n-1} + \bar{f}_1 \frac{yR_{\text{moon}}}{r^2} V_{n-1,n-1} \right\} \tag{A.15}$$

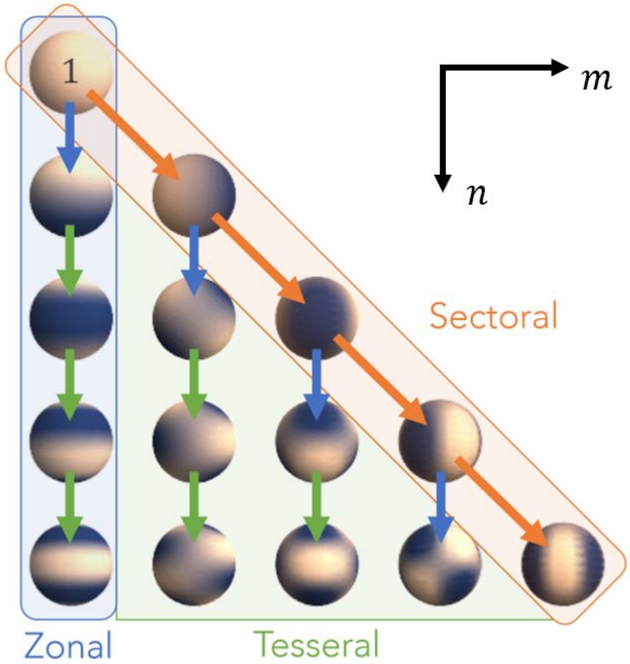


Figure A.1: Visualization of the process used to calculate zonal, tesseral, and sectoral terms of the $V_{n,m}$ and $W_{n,m}$ matrices using Montenbruck and Gill's recurrence relations [51].

The second recurrence relation becomes

$$V_{n,m} = \bar{f}_2 \left(\frac{2n-1}{n-m}\right) \frac{zR_{\text{moon}}}{r^2} V_{n-1,m} - \bar{f}_3 \left(\frac{n+m-1}{n-m}\right) \frac{R_{\text{moon}}^2}{r^2} V_{n-2,m} \tag{A.16}$$

$$W_{n,m} = \bar{f}_2 \left(\frac{2n-1}{n-m}\right) \frac{zR_{\text{moon}}}{r^2} W_{n-1,m} - \bar{f}_3 \left(\frac{n+m-1}{n-m}\right) \frac{R_{\text{moon}}^2}{r^2} W_{n-2,m} \tag{A.17}$$

The full $V_{n,m}$ and $W_{n,m}$ matrices can then be defined using the process shown in Figure A.1.

The procedure to calculate the $V_{n,m}$ and $W_{n,m}$ matrices shown in Figure A.1 is as follows:

1. Define initial $V_{0,0} = \frac{R_{\text{moon}}}{r}$ and $W_{0,0} = 0$.
2. Calculate the sectoral terms using the first recurrence relation where $n = m$.
3. Calculate the zonal terms using the second recurrence relation, accounting for the $n = m + 1$ case.
4. Calculate the tesseral terms using the second recurrence relation, accounting for the $n = m + 1$ case.

The component gravity accelerations can be calculated once the $V_{n,m}$ and $W_{n,m}$ matrices are found. The gravitational acceleration is evaluated using

$$\ddot{x} = \sum_{n,m} \ddot{x}_{n,m} \qquad \ddot{y} = \sum_{n,m} \ddot{y}_{n,m} \qquad \ddot{z} = \sum_{n,m} \ddot{z}_{n,m} \tag{A.18}$$

where each component acceleration is found using

$$\ddot{x}_{n,m} = \begin{cases} \dfrac{GM_{\text{moon}}}{R_{\text{moon}}^2} \left(-C_{n,0} V_{n+1,1}\right), & \text{if } m = 0 \\ \dfrac{GM_{\text{moon}}}{2R_{\text{moon}}^2} \Big[\left(-C_{n,m} V_{n+1,m+1} - S_{n,m} W_{n+1,m+1}\right) & \\ \quad + \frac{(n-m+2)!}{(n-m)!} \left(C_{n,m} V_{n+1,m-1} + S_{n,m} W_{n+1,m-1}\right)\Big], & \text{if } m > 0 \end{cases} \tag{A.19}$$

$$\ddot{y}_{n,m} = \begin{cases} \dfrac{GM_{\text{moon}}}{R_{\text{moon}}^2} \left(-C_{n,0} W_{n+1,1}\right), & \text{if } m = 0 \\ \dfrac{GM_{\text{moon}}}{2R_{\text{moon}}^2} \Big[\left(-C_{n,m} W_{n+1,m+1} + S_{n,m} V_{n+1,m+1}\right) & \\ \quad + \frac{(n-m+2)!}{(n-m)!} \left(-C_{n,m} W_{n+1,m-1} + S_{n,m} V_{n+1,m-1}\right)\Big], & \text{if } m > 0 \end{cases} \tag{A.20}$$

$$\ddot{z}_{n,m} = \frac{GM_{\text{moon}}}{R_{\text{moon}}^2} \left[(n - m + 1)\left(-C_{n,m} V_{n+1,m} - S_{n,m} W_{n+1,m}\right)\right] \tag{A.21}$$

Just like for the $V_{n,m}$ and $W_{n,m}$ matrices, normalization factors are needed to compute the component gravity accelerations. Those factors are shown below

$$\bar{f}_4 = \frac{f_{n,0}}{f_{n+1,1}} \quad \bar{f}_5 = \frac{f_{n,m}}{f_{n+1,m+1}} \quad \bar{f}_6 = \frac{f_{n,m}}{f_{n+1,m-1}} \quad \bar{f}_7 = \frac{f_{n,m}}{f_{n+1,m}} \tag{A.22}$$

These are substituted back into the above component equations to find

$$\ddot{x}_{n,m} = \begin{cases} \frac{GM_{\text{moon}}}{R^2_{\text{moon}}} \left(-\bar{f}_4 C_{n,0} V_{n+1,1}\right), & \text{if } m = 0 \\ \frac{GM_{\text{moon}}}{2R^2_{\text{moon}}} \left[\left(-\bar{f}_5 C_{n,m} V_{n+1,m+1} - \bar{f}_5 S_{n,m} W_{n+1,m+1}\right)\right. & \\ \quad \left. + \frac{(n-m+2)!}{(n-m)!} \left(\bar{f}_6 C_{n,m} V_{n+1,m-1} + \bar{f}_6 S_{n,m} W_{n+1,m-1}\right)\right], & \text{if } m > 0 \end{cases} \tag{A.23}$$

$$\ddot{y}_{n,m} = \begin{cases} \frac{GM_{\text{moon}}}{R^2_{\text{moon}}} \left(-C_{n,0} W_{n+1,1}\right), & \text{if } m = 0 \\ \frac{GM_{\text{moon}}}{2R^2_{\text{moon}}} \left[\left(-\bar{f}_5 C_{n,m} W_{n+1,m+1} + \bar{f}_5 S_{n,m} V_{n+1,m+1}\right)\right. & \\ \quad \left. + \frac{(n-m+2)!}{(n-m)!} \left(-\bar{f}_6 C_{n,m} W_{n+1,m-1} + \bar{f}_6 S_{n,m} V_{n+1,m-1}\right)\right], & \text{if } m > 0 \end{cases} \tag{A.24}$$

$$\ddot{z}_{n,m} = \frac{GM_{\text{moon}}}{R^2_{\text{moon}}} \left[(n-m+1)\left(-\bar{f}_7 C_{n,m} V_{n+1,m} - \bar{f}_7 S_{n,m} W_{n+1,m}\right)\right] \tag{A.25}$$

The above equations define the spherical harmonic calculation of the gravity field given $C_{n,m}$ and $S_{n,m}$ coefficients (with normalizations compatible with GRAIL data). The alternative gravity fields described in Section 2.1 can be calculated using these methods. The LGM2011 gravity model was chosen over models found using spherical harmonics due to their divergence at the celestial body's surface.