Supporting Autonomous Navigation with Flash Lidar Images in Proximity to Small Celestial Bodies

by

Ann B. Dietrich

B.S., Aerospace Engineering, University of Florida, 2012

M.S., Aerospace Engineering, University of Colorado at Boulder, 2014

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Aerospace Engineering Sciences
2017

This thesis entitled:
Supporting Autonomous Navigation with Flash Lidar Images in Proximity to Small Celestial Bodies
written by Ann B. Dietrich
has been approved for the Department of Aerospace Engineering Sciences

Prof. Jay McMahon

Prof. Hanspeter Schaub

Prof. Eric Frew

Date ______________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Dietrich, Ann B. (Ph.D., Aerospace Engineering)

Supporting Autonomous Navigation with Flash Lidar Images in Proximity to Small Celestial Bodies

Thesis directed by Prof. Jay McMahon

Spacecraft navigation in proximity to small celestial bodies, such as asteroids and comets, is challenging due to their complex dynamical environment and the lag in communications from Earth to the spacecraft. Increasing autonomous spacecraft navigation reduces the burden of ground-based planning and modeling, and enables insightful mission profiles. The state-of-the-art of relative spacecraft navigation uses optical images, requires an iterative procedure, and currently must be performed on the ground. A flash lidar instrument instantaneously returns a 3D elevation map of its target and shows promise for advancing autonomous spacecraft navigation. Using this instrument as a relative measurement source for navigation performed similarly or better than landmark-based navigation from optical images. The model-based approach used to compute the onboard flash lidar images eliminated the correlation procedures required of landmark-based approaches, and reduced the computational load. An extended Kalman filter (EKF), an unscented Kalman filter (UKF), and an iterative least-squares (LS) filter were investigated in this analysis. The iterative LS filter iterated the estimation state at each observation time, produced smaller errors than the EKF and UKF, and did not encounter filter saturation. The image properties of the flash lidar measurements allowed for pointing to be estimated. The UKF and LS filters were robust to initial position errors as long as an overlap occurred between the observed and computed flash lidar images. When introducing shape modeling errors, the filters did not diverge, and the majority of the state errors were captured with a sequential consider covariance analysis. Using the image properties of the flash lidar images, and assuming inertial spacecraft pointing knowledge, the filter was initialized through pre-processing algorithms and the iterative LS algorithm. Optimally reducing the number of altimetry measurements processed by maximizing their information contribution increased the computational efficiency and combated filter saturation without sacrificing accuracy.

Dedication

I would like to dedicate this to my parents. To my mom for being an example for me to always work hard. To my dad for instilling in me a sense of adventure and teaching me how to navigate in it. And to both for setting up this life for me that has given me so many opportunities and for supporting me in every one I wanted to take.

Acknowledgements

I would first like to acknowledge my advisor, Jay McMahon for his guidance in this research topic, his patience with me when things broke or when work got slow, and his support of me to persevere. I appreciate his willingness to always meet if needed, and to help tackle the math when it just wasn't working correctly. I want to especially recognize his grounded approach to his advising, as I never felt that I was given unreasonable expectations, but was always pushed to my potential.

I would like to thank my committee members for their time to read this dissertation, to attend my presentations, and for the conversations that have helped direct this research. I would especially like to thank George Born for hiring me five years ago, providing me with the Smead fellowship, and giving me this opportunity to attend the best space program in the country.

The Smead program has been instrumental in my achievements and support at CU, and I would like to thank Ann and Michael from the bottom of my heart for all that they have provided me with. I have also been supported by the National Science Foundation Graduate Research Fellowship, the Zonta Amelia Earhart Fellowship, the ARCS Scholar fellowship, and the Society of Women Engineers. Their financial support greatly decreased the burden of completing this dissertation.

Most importantly, I would like to thank the friends I have made in Colorado and our adventures that have kept me sane and motivated. I want to thank the grad students that were here when I arrived at CU for accepting me into this community, and for teaching me just as much research knowledge or more than I learned in my classes. I want to thank my fellow grad students for the help in classes and in research, the late nights, and the knowledge that we weren't going through this alone. Particularly, I would like thank the girlfriends I have made in this program for our shared understanding and our friendships. A specific acknowledgement is reserved for Samantha Rieger, who has shared with me the exact same experiences at almost the same times, and lent her support whenever I needed it.

Again, I would like to thank my parents for all of their support of me. From the after-school clubs to dance performances to science camps to helping me move in college, and for their continued support while I've been 1600 miles away, thank you.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Motivation

Small celestial bodies, such as asteroids and comets, are abundant in our Solar System and provide great scientific insight into the origins of our Solar System. These small bodies may be mined for resources, such as water or metals, and if one was inbound toward Earth, humans would need knowledge on how to deflect it for planetary protection. Furthermore, the size and composition of small celestial bodies varies, and spacecraft missions to multiple small bodies are required for adequate characterization.

Spacecraft navigation around these small bodies is difficult due to the complex dynamical environments that surround them. Most small bodies have irregular shapes that entail an asymmetric gravity field, and their small mass leads to other dynamical sources, such as solar radiation pressure and other planets, imparting a larger effect on the motion of the spacecraft. The rotation period of a small body may be on the order of hours or days, and its axis of rotation may be a nonprincipal axis.[96] Limited ground-based information is available due to their small size, and a lag in communications occurs between Earth and the spacecraft. Missions have experienced a round trip communications lag from 15 to 30 minutes.[43, 39]

Developing autonomous navigation systems for small body spacecraft missions can increase the return on scientific knowledge, and the spacecraft may perform more interesting mission profiles. Due to the complex dynamical environment, and the limited *a priori* knowledge of a small body, precise navigation of the orbiting spacecraft is prudent to estimate the position and velocity of the spacecraft, as well as uncertain dynamical and physical parameters of the small body. Measurements relative to the target body notably reduce the uncertainty in the estimation state by providing relative navigation information.[11] While missions have been successfully executed,[77, 72, 11] much of the planning and operations are currently performed on the ground, and further characterization of the small body is required once arriving.[43, 69]

Currently, relative navigation is performed with optical images and optical navigation (OpNav), and is the state-of-the-art for small body missions.[73, 18, 55] This process involves correlating the two-dimensional (2D) optical images with a three-dimensional (3D) shape model, and iterating between an OpNav team and an orbit determination (OD) team. The images must be directed to the sunlit side of the small body, and a 2D scale factor must be resolved within the estimation state or with an altitude measurement. This current architecture may take hours or days and also must be performed on the ground, hindering the real-time navigation capabilities of the spacecraft in these complex environments.[72]

This dissertation evaluates using a flash lidar instrument as a relative measurement source for OD in proximity to a small celestial body as a means to increase navigation autonomy. A flash lidar measurement instantaneously returns an array of altimetry measurements, where each pixel provides a range measurement. This measurement provides multiple altimetry measurements to the surface, contains image properties, and the altitude is inherently included. Additionally, this is used in any lighting conditions, and optical image processing procedures may be sidestepped.

Using a flash lidar instrument for relative navigation simplifies the navigation process, and allows the algorithms to be placed onboard the spacecraft. The onboard computer does not need to perform image correlations or render features (as with OpNav), and simply needs a shape model and relative orientation of the target body. Multiple lidar measurements at slightly different angles are captured at once, and information is gathered on the spacecraft relative position as well as pointing orientation. These characteristics add up to promising applications of flash lidar in onboard navigation, and thus enhancing spacecraft autonomy.

Altimetry measurements, such as those from a lidar instrument, have been studied for terrain relative navigation, planetary landings, and autonomous rendezvous and docking [5, 88, 49]. The dynamics in these studies are confined to terrestrial applications, and/or evolve over a short time period, such as a planetary landing. The application of a using a 3D lidar sensor, such as a flash lidar or scanning lidar, for relative measurements around a small body for OD and for long durations is still an open topic of research. Using a flash lidar sensor is advantageous over using a scanning lidar as it does not involve moving parts and may be treated as a time-of-flight camera. [29]

Challenges arise in processing flash lidar images with linearized estimation techniques. The dynamics around a small body are complex and nonlinear, and the relationship between an altimetry measurement and the spacecraft state is also nonlinear. Processing a considerable number of measurements at one observation time (as is the case with a flash lidar image) may lead to underestimation of the uncertainty of the state errors. In addition, comparing the observed and computed 3D images involves matching corresponding points and these processes may exclude information.

This dissertation focuses on relative navigation in the proximity operations phase of a small body mission. This phase would occur after the characterization phase is completed, and the spacecraft is performing low altitude orbits. Therefore, it is assumed a shape model has been defined, and the rotation axis and spin rate of the small body have been estimated in greater detail beyond that from solely ground observations. To simplify the problem, only flash lidar measurements for relative navigation were studied in order to analyze their capabilities. It was assumed that other measurement types, such as range and Doppler measurements from Earth, would also be used to estimate the spacecraft state in the inertial frame. For the algorithm development, simple and fast algorithms were given more merit over complex algorithms. This was chosen to advance the study of using flash lidar measurements for autonomous navigation.

> **Thesis Statement**
> A flash lidar instrument is evaluated as a means to increase spacecraft navigation autonomy in proximity to small bodies. The performance of flash lidar images is compared to the state-of-the-art of using optical images. Employing a model-based approach for computation of the flash lidar images is evaluated for increased computational efficiency, as well as the robustness of the flash lidar images to initial state and measurement model errors. Filter initialization algorithms are derived from the image properties of a flash lidar image, and fur-

ther computational efficiency is examined by reducing the number of lidar beams processed.

1.2 Background

Small celestial bodies tend to have highly irregular shapes and weak gravity fields, and comets may have drag and outgassing accelerations that are difficult to characterize. While asteroids and comets are our closest neighbors in the Solar System, their small size makes them difficult to observe. Observing techniques have advanced over the past decade and therefore so has our catalog of near earth objects (NEOs) [52]. However, our knowledge of their ephemeris and shape remains limited. Even on the most recent Rosetta mission, the knowledge of the comet's ephemeris before approach involved errors too large to plan and execute the orbit insertion burn [11].

NEOs overall are categorized as asteroids and comets in orbits with a semi-major axis less than 1.3 AU, and range from less than 30 m in diameter to over 1 km in diameter. The majority of NEOs are near Earth asteroids (NEAs) and these are further classified into subgroups based on their orbits. The number of cataloged NEAs has grown exponentially since 2000. Figure 1.1 shows the cumulative number of NEAs discovered by year, and categorized by their size. It is estimated that there are 990 ± 20 NEAs that are larger than 1 km in diameter and 90% of them have been discovered since August 2014. [52]

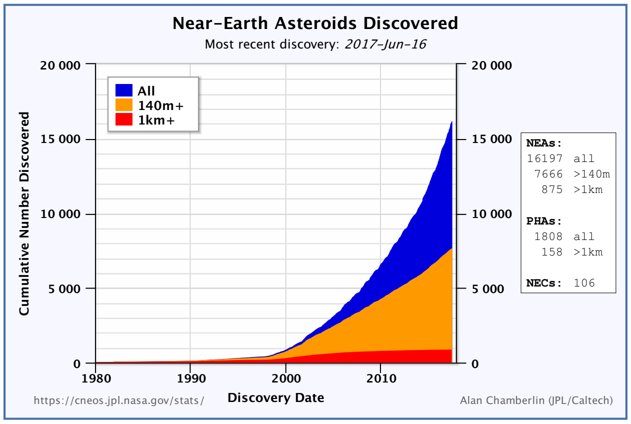


Figure 1.1: Cumulative number of discovered near Earth asteroids over time (Courtesy of NASA/JPL-Caltech).[2]

Spacecraft missions have flown to small celestial bodies, such as NEAR to the asteroid Eros in 2000, and are increasing in interest. The OSIRIS-REx spacecraft that recently launched in September 2016 will travel to the asteroid Bennu, and two asteroid missions were selected in the 2016 NASA Discovery Program: Lucy that will travel to the Trojan asteroids by Jupiter and Psyche that will travel to a metal asteroid in the main asteroid belt.

The following sections highlight some of the previous small body missions and their mission profiles, followed by the relative navigation techniques used in these missions once arriving at the small body. The next section overviews the flash lidar instrument and research involving its application to relative navigation in space. Spacecraft systems that have implemented autonomous navigation or autonomous operations is presented next. Relative navigation in space applications such as terrain relative navigation, planetary landings, and autonomous rendezvous and docking has acquired multiple publications, and a table is presented mapping the sensors investigated in this field. Finally, relevant algorithms for relative navigation is overviewed.

1.2.1 Small Body Mission Profiles

Small body missions have performed asteroid fly-bys, inserted into orbit around their small body target, deployed landers, and attempted sample returns. The US National Aeronautics and Space Administration (NASA), the European Space Agency (ESA), and the Japan Aerospace Exploration Agency (JAXA) have all completed missions to small bodies. As limited information on the target small body is available before arriving there, the mission design usually involves mapping and reconnaissance orbits at a safe distance from the small body. A spacecraft will typically insert into these orbits when first arriving at the small body to characterize it, and then perform lower altitude flybys or landings. Small body missions have successfully been executed, yet these mission still encounter difficulties and require unique engineering solutions. This section highlights some of these missions for context on small body mission profiles.

---

[2] Center for Near Earth Object Studies, Discovery Statistics $https://cneos.jpl.nasa.gov/stats/totals.html$

NEAR The NEAR spacecraft inserted into a 320 × 360 km orbit about the asteroid Eros in February 2000 and became the first spacecraft to orbit a small body. NEAR completed 17 orbit phases between February 2000 and October 2000. These orbits ranged from 360 km, down to 35 km, with a 5 km flyby of the surface in October of 2000. [77]

In February 2001, NEAR landed on Eros as its end-of-mission profile. NEAR was placed in a 36 km orbit in January 2001; a deorbit maneuver from this orbit and four breaking maneuvers beginning at 5 km of altitude were planned for NEAR's descent. At this time, Eros was 2.11 AU from Earth, imposing a 17.5 minute delay in communications from NEAR to Earth. In February 2001, NEAR touched down on Eros with an estimated vertical velocity of 1.5-1.8 m/s, achieving a soft landing. The fourth breaking maneuver was terminated early due to earlier-than-expected contact with the surface, and NEAR was pushed slightly into the surface. The last image taken by NEAR occurred at an altitude of 129 m with a resolution of 1 cm. [39]

Hayabusa After its launch in May 2003, and an Earth flyby in May 2004, the JAXA Hayabusa spacecraft arrived at the asteroid (25143) Itokawa in September 2005. For the initial arrival at Itokawa, the spacecraft was at an altitude of 20 km, and after reconnaissance operations, transferred to an altitude of 7 km above the surface at the end of September 2005. Throughout October 2005, the Hayabusa spacecraft did not insert into a specific orbit, but transferred to eight different hovering positions at different sun phase angles and altitudes. The lowest altitude the spacecraft reached was about 3 km from the surface. [43]

On November 19th and 25th, 2005, two touchdown attempts were made to the surface, known as touch-and-go (TAG) manuevers. A Tracking Marker (TM) was released to the surface, and this was tracked for relative position control during the descent. Following anomalies in both TAG attempts, including a crash landing, Hayabusa was placed on a return trajectory to Earth, arriving at Earth in June 2010. When Hayabusa returned to Earth, less than a milligram of a particles had been returned, yet this was the first direct sample of an asteroid to be returned to Earth. [67, 114]

Rosetta The Rosetta spacecraft and the Philae lander were launched in March 2004 by ESA toward the comet 67P/Churyumov-Gerasimenko. After three Earth flybys, a Mars flyby, and two asteroid flybys, the spacecraft was taken out of hibernation in January 2014. In August 2014, the Initial Characterization phase began as Rosetta inserted into a pyramid orbit. The subsequent orbit phases included a 30 km circular orbit, three plane changes to change the sun-relative angles, a close observation orbit of 10 km, and the Philae delivery orbit to minimize the separation ΔV of Philae from Rosetta. [11]

In November 2014, the Philae lander was released to land on the surface of 67P. However, the cold gas thruster and the harpoon subsystems meant to attach Phalie to the surface failed, and Philae bounced several times on the surface. It finally landed in an unknown location, but most likely in shade, and hindered its original scientific objectives.[14]

In the following two years, the Rosetta spacecraft characterized 67P, until its end of mission, landing on 67P in September 2016. In the months leading up to its landing, Rosetta was placed in a 3-day eccentric orbit with a decreasing periapsis down to approximately 1.5 km in altitude. Rosetta performed a maneuver to transfer itself to an altitude of 20 km, before descending and touching down on September 30th, 2016. [18]

OSIRIS-REx The OSIRIS-REx spacecraft launched in September 2016 for rendezvous with the near-Earth asteroid Bennu (101955). Once arriving in late 2018, the spacecraft will perform a series of flybys at a distance of 7 km and then insert into a 1.5 km circular terminator orbit. This first orbit allows for refinement of the characteristics of Bennu, such as the shape model and landmark identification. Next, a series of hyperbolic flybys allows for a detailed survey of Bennu at various sun phase angles. The next phase of the mission transfers the spacecraft to a 1 km circular terminator orbit. This will serve as the base orbit from which the touch-and-go (TAG) sequences wil initialize from toward the surface of Bennu. After the Reconnaissance phase to survey candidate sample sites, and two planned rehearsals of the TAG sequence, OSIRIS-REx will touchdown and collect samples from Bennu. In March 2021, OSRIS-REx is planned to depart Bennu and return to Earth in September 2023. [55]

1.2.2 Small Body Relative Navigation

The overall study of spacecraft navigation is determining the orbit of the spacecraft; this is known as orbit determination (OD). At a minimum, this involves estimating the position and velocity of the spacecraft either inertially or with respect to its orbiting body, in order to determine its orbit. Many parameters that describe the dynamical environment of the spacecraft also are estimated, since limited information is available on this environment. These additional parameters may include the mass of the small body, gravity parameters, the ephemeris of the small body, solar radiation pressure from the sun, or drag coefficients, to name a few. Measurement parameters may also be included in the estimation state, such as measurement biases or landmark locations for relative navigation.

Measurements are taken of the spacecraft to estimate its position, velocity, and the parameters of the estimation state. For small body missions, these measurements typically include range and range-rate (Doppler) measurements from Earth to the spacecraft. The range is measured by multiplying the time it takes for communications from Earth to arrive at the spacecraft (or vice versa) by the speed of light, and range-rate is the change in this range. For current small body missions, these measurements are provided by the Deep Space Network (DSN), operated at the NASA/Caltech Jet Propulsion Laboratory (NASA/JPL).

Once arriving at a small body, relative navigation is performed by acquiring measurements with respect to the body. These measurements are critical to decrease the uncertainty in the estimation state, and at times effect mission-critical events [11]. The current state-of-the-art for relative navigation is the use of optical images, known as OpNav. This section describes the evolution of optical navigation from crater landmarks to stereophotoclinometry, and their uses on previous small body missions.

1.2.2.1 Optical Navigation with Crater Landmarks

Asteroid proximity navigation concepts began around 1990 with interest in the Comet Rendezvous/Asteroid Flyby (CRAF) NASA mission planned for the Mariner Mark II spacecraft. While this mission was canceled, studies were performed to tackle the challenges with navigating a spacecraft in proximity to an asteroid. This included addressing the irregular shape, gravitational variations, and accelerations due to outgassing.[76] Simulated navigation measurements included range, Doppler, very long baseline interferometry (ΔVLBI), and optical imaging of surface landmarks. Estimation parameters included the asteroid attitude, landmark locations, spacecraft position and velocity, and gravitational and dynamical parameters. The shape model was depicted as an ellipsoid with different sized masses, or mascons, spread around the surface, and the gravity model included Legendre polynomials with a stochastic error model for varying non-gravitational forces. The landmarks were assumed to be features on the surface such as craters or valleys.[76]

During the NEAR mission, landmarks were used for relative navigation, usually craters on the surface, and a catalog of landmarks was kept for landmark tracking. By identifying the landmarks in subsequent images, this enabled OD down to a few meters, or the resolution of the camera. This accuracy was much greater than using radiometric data alone, or by fitting the limb of the asteroid to a reference model. The optical imaging allowed the OD schemes to be less dependent on referencing a precise shape model. [77] The inclusion of landmark tracking with Doppler data provided increased orbital position accuracy to the 10 to 20 m range, faster convergence after maneuvers, and quicker determination of physical properties, such as the spin axis and pole of Eros [23]. However, the crater detection and identification in subsequent images for NEAR was performed manually. Methods to automatically identify craters were theorized after the mission by edge detection and ellipse-fitting of the crater [23].

1.2.2.2 Optical Navigation and Stereophotoclinometry

The “landmarks” have since evolved into computed images of the surface based on the sun-phase angle and the reflectance and albedo of the surface. The current standard for optical navigation (OpNav) is the use of stereophotoclinometry (SPC). This technique creates “landmark maps” or L-maps to determine the surface topography and spacecraft orbit. These are created from a reference shape model of the small body, the angle of the sun, the surface illumination, and the albedo of the surface. These maps do not use actual landmarks, such as craters, and can be computed analytically. Therefore, they can be reconstructed in many different lighting conditions and cover diverse topography that may not contain craters or landmarks. Gaskell (2008) [45] outlined this technique, and it was used in the Hayabusa mission to Itokawa that did not have craters for feature identification.

The L-map is described as a vector from the center of the body to the local surface coordinate system. The relative heights and albedos throughout that coordinate system are calculated based on the surface albedo and reflectance functions. These L-maps have the ability to span over multiple images to be cross-correlated and matched. By knowing the vectors that describe the L-maps in the body-fixed frame of the small body, as well as the position and orientation of the camera’s image plane on the spacecraft, the spacecraft location and pointing is able to be estimated. However, the spacecraft position and landmark positions contain a scale bias that must be resolved by external data such as lidar or Doppler measurements. Through the L-maps, their position on the surface, and the spacecraft position and pointing, a shape model of the small body is constructed. A survey phase was envisioned to use this technique, from which the spacecraft would identify thousands of landmarks from different lightning conditions to build these maps. Once the maps are constructed, it is simple for the spacecraft to navigate and compute residuals by using the L-maps and the body-fixed vectors describing them. [45]

Geometric techniques are used to estimate the spacecraft state from known landmarks by computing the residuals of the landmark locations in the image frame. Estimating the sample/line location of a point in an image may involve many parameters such as the relative direction the photons are emitting from, distortion within the optical system, and pixel noise [84]. A simplified version of an OpNav formulation was used in this dissertation for comparison with the lidar methods, and more details on its mathematical formulation are found in Chapter 2.

1.2.2.3 Mission Navigation Profiles

Optical navigation and SPC are the current state-of-the-art for relative navigation about a small body. Optical navigation was used on the NEAR mission[77], and OpNav and SPC were the main source of relative navigation with Hayabusa[45], DAWN [72], Rosetta[11], and will be on OSIRIS-REx[55].

On approach to the small body, optical navigation is often used to locate the target small body as a light point source in the field of view of the camera. Once the spacecraft is close enough to the small body, and features are able to be resolved in the optical images, landmark tracking and the OpNav procedures explained in the previous section are used (DAWN[72], Rosetta[73], OSIRIS-REx[55]). As referenced in Section 1.2.1, most small body mission trajectory designs involve an initial survey phase. This is used to characterize the small body by capturing optical images and navigation observations, and construct a shape model (Hayabusa[45], DAWN [72], Rosetta [11], OSIRIS-REx[55]).

Once the spacecraft has inserted into orbit and/or the survey phase is completed, optical images for navigation may be taken every few minutes, as was done for the DAWN mission at Vesta[64], or every few hours as will be done for OSIRIS-REx[55].

Many parameters are estimated in the OD filters alongside the spacecraft's position and velocity that describe the dynamical environment and measurement parameters. Parameters in these missions have included the asteroid attitude, the asteroid ephemeris, the mass of the target body, the gravity field parameters of the target body, the spin pole and rate of the target body, solar radiation pressure on the spacecraft, the spacecraft maneuvers, the landmark locations, and measurement biases.

DAWN During the DAWN mission at the small body Vesta, many images with different resolutions, and incidence and emission angles were taken to create the shape model and the L-maps for SPC. Many of the processes and subprocesses to create these L-maps involved multiple iterations to converge on a solution. This included the model image brightness and albedo, the heights within the L-map, as well as solving for the landmarks, spacecraft position, position of Vesta, and the camera pointing. In operations, iterations occurred within each data set for aligning new images, adding new images to the L-maps, constructing new landmarks, and filing the surface with landmarks at the needed resolution. [72] Solving for the landmark locations, Vesta's ephemeris was reduced in uncertainty by two orders of magnitude in some orbital elements, and the new orbit had a difference of 2-sigmas from the previous solved-for orbit [64].

Rosetta In the Rosetta mission, NASA/JPL performed an independent shadow navigation effort to support the ESA mission. Optical images were received by JPL, and independent OpNav and SPC methods were employed to support this effort. During the initial characterization phase, the landmark maps required about 4-5 work days to create with the low-resolution images, and the landmark locations were known to within 3.1 m ($1-\sigma$). An iteration process occurred with shifting individual images to correlate the illumination of their geometry, and a global map was constructed by overlapping these images. The landmark vectors, camera pointing, and spacecraft state were estimated again. Once all the visible area had been covered, the OD process was carried out with these landmark maps, and recreating the landmark maps and brightness model was repeated until the post-fit residuals reached a steady state.

During the close observation phase, parts of the comet surface were in shadow, yet these images had to be used in order to extract all of the available information. This led to misidentification of landmarks that was resolved by manual processing. At the lowest altitudes, the landmark location uncertainty could grow from 10 m to 120 m due to the outgassing dynamics, sampling a small part of the surface, and the high phase angle. This was resolved through additional iterations of the same image set. [73]

1.2.2.4 Small Body Relative Navigation with Lidar

Lidar (light detection and ranging) for navigation has been explored as relative measurements to a celestial body or spacecraft. [16, 1, 75] A lidar sensor actively projects a light source (such as a laser) toward a target, measures the time traveled for the light source to return to the sensor, and calculates a range measurement. An advantage of using lidar is that it may be used in any lighting conditions, as oppose to optical imaging. It is noted though that when using lidar relative to a celestial body, the altitude returned is relative to the surface of the target body that may be at an unknown radius from the center of mass. This places reliance on the reference shape model.

While NEAR was in orbit around Eros, a lidar onboard the spacecraft was used primarily for shape model determination, and optical images were the primary source for OD during operations. One study [16] investigated the effectiveness of using lidar data in addition to optical data and radiometric data for OD. The reference orbits used were the operational NEAR orbits that were estimated with Doppler and optical imaging. The errors in the orbit with respect to the operational orbit were approximately 40 m when using lidar and radiometric data to estimate the orbit. When using only the lidar data, the orbit errors increased to the 100 to 150 m range. [16]

The use of multiple beam lidar measurements has been theorized for relative navigation around asteroids. Navigation using three lidar beams and comparing an ellipsoidal and faceted shape model within the navigation filter was studied, and it was found to produce similar results to OpNav [1]. Using one lidar beam at two discrete times was investigated, and it was found that pointing this beam towards the limb of the asteroid produced the most information [75].

1.2.3 Flash Lidar for Relative Navigation

This dissertation focuses on using a flash lidar instrument to obtain a three-dimensional (3D) image of its field of view (FOV). A flash lidar measurement operates similar to a conventional camera in that the measurement contains an array of pixels, and each pixel represents a range measurement. A single beam is spread out across the illuminated array, and the entire scene is illuminated at once to produce the 3D point cloud. Each pixel in the array can be treated as a single lidar beam directed at a slightly different angle. [91] The algorithms to calibrate a flash lidar can be complex, but these instruments are suitable for both short and long range targets, up to a few kilometers away. [29]

1.2.3.1 Flash Lidar Instrumentation

Flash lidar has been developed by aerospace ventures such as Advanced Scientific Concepts (ASC) and Ball Aerospace. The instrument at Ball Aerospace produces a 256 × 256 resolution array with rates up to 30 Hz [30].

ASC's current products include two lightweight (< 3 lbs), small, flash lidar cameras, and one space-rated flash lidar camera. The GoldenEye Space Camera has a full array of 128 × 128 pixels, with observation frame frequency up to 10 Hz. It is currently flying on the OSIRIS-REx spacecraft, and weighs 6.5 kg. It provides range up to 3 km, with range biases less than ±10 cm and range noise less than ±15 cm (3-σ). [3]

The measurements returned from a flash lidar instrument may include light intensity reflected back from the target object and/or range measurements to the objects in its FOV. An electronically steerable flash lidar has the ability to easily illuminate different patterns of pixels within the image and adjust the laser power based on the target's range and reflectivity. [89] For example, when every pixel in the image is illuminated, the maximum observable range may be up to 1 km, while if a smaller pattern of pixels within the image is illuminated, the maximum observable range increases. The errors in the measurements depend on the travel time of the lidar light, the distance of the pixel from the center of the image, and the intensity of the light returned. [89]

1.2.3.2 Flash Lidar Space Applications

Flash lidar has been investigated for terrain relative navigation and hazard avoidance for planetary landings. It has also been tested on the Space Shuttle and International Space Station

---

[3] http://www.advancedscientificconcepts.com/products/portable.html

(ISS) for spacecraft rendezvous and docking.

Using flash lidar measurements, crater identification algorithms such as template matching, feature identification, and finding local minima and maxima may be used for terrain relative navigation. These formulations require a database of craters on the surface, as well as crater features on the surface, and are similar to optical crater identification methods. [90]

The Autonomous Landing and Hazard Avoidance (ALHAT) mission, run by NASA Johnson Spaceflight Center (JSC) and NASA Langley Research Center (LaRC) explored the use of flash lidar for planetary landings through experimental testing and algorithm development. The ALHAT program aimed to achieve specific objectives for planetary landing that included altimetry, velocimetry, terrain relative navigation, hazard detection and avoidance, and hazard relative navigation. Flash lidar was proposed to perform all of these tasks, with the exception of the velocimetry, in which a Doppler lidar was proposed to fulfill this task. [4] [3] A sensor test bed for characterizing flash lidar and its uses for hazard avoidance was developed at NASA LaRC [86], and flight tests in airplanes and helicopters have been performed for the flash lidar system. [21] The ALHAT project has be upgraded to TRL 6 by performing a closed-loop flight test with the flash lidar system attached to the Morpheous rocket-propelled lander at NASA Kennedy Space Center. The system scanned a lunar-like surface, built an elevation map, identified a safe landing site, and guided the lander to this site safely. [88]

A flash lidar system also was tested on the ISS for rendezvous and docking of the space shuttle on STS-134 in the Sensor Test for Orion RelNav Risk Mitigation (STORRM) mission. The ISS was considered the “target” spacecraft and had reflectors on it in known locations for the flash lidar; the STS-134 was the “chaser” spacecraft with the flash lidar instrument. An end-to-end assessment of performing relative navigation with flash lidar to support autonomous rendezvous and docking algorithms was developed from this data. The “target” spacecraft was assumed to have reflectors on it in known positions, so the “chaser” spacecraft only had to identify the features and reduce their residuals to determine its relative position and attitude. [26]

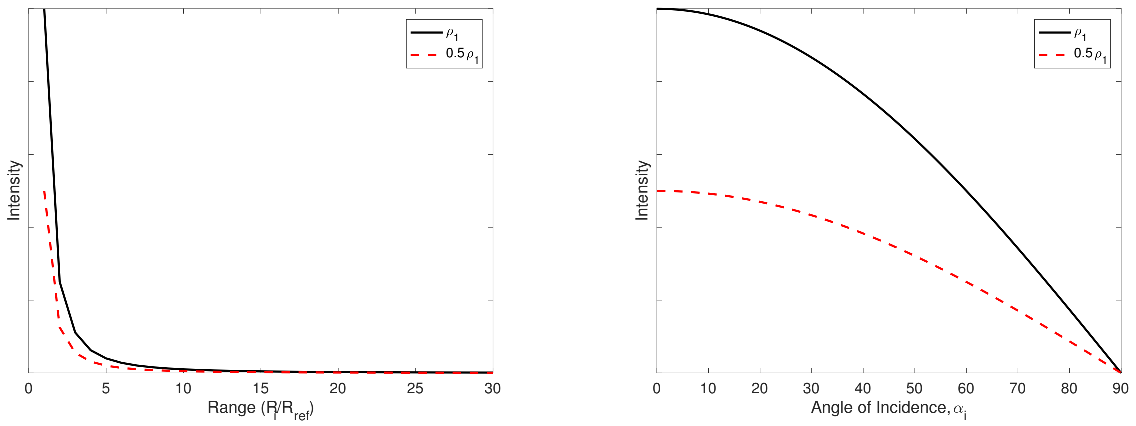
Relative navigation with the STORRM data also was investigated without the use of reflectors. The iterative closest point (ICP) method was explored to align successive range images, and to align range images to a defined model of the ISS. It studied using known features on the ISS for navigation. However, since the flash lidar instrument was calibrated to the reflectors on the ISS, the light intensity of specific features greatly varied between successive images. This resulted in sub-par navigation estimation because the algorithms were not able to align the images with each other or with a model. [91]

The ASC DragonEye camera was tested on Space Shuttle mission STS-127 to assess its approach capabilities for the SpaceX Dragon vehicle. At the time, it was slated as the primary sensor for relative navigation between the Dragon vehicle and the ISS. The DragonEye sensor used the Trajectory Control Sensor system with retroreflectors on the ISS, and this data was used to evaluate its performance. [80]

1.2.4 Lidar Capabilities

The intensity of a returned lidar beam is related to the instrument's power, range, the acquisition geometry, and the target surface's reflectance and roughness. [61] These contributing factors may be used for target/material classification and/or calibrated to provide a useful output for the end-user. Lidar intensity has seen terrestrial remote-sensing applications such as land classification (trees and glaciers), structural damage assessment, and transportation assessment. Surfaces with known reflectance properties may be used to calibrate the returned intensity. The range and incidence angle (the angle between the surface normal and the incident light) are two contributing factors that are accounted for in the majority of intensity calibration methods. [61]

The lidar range equation relates the received optical power to the transmitted power and other instrument and environmental properties. By assuming the target intercepts the entire lidar beam, and Lambertian reflectance, the received optical power is directly related to the transmitter power, receiver aperture diameter, atmospheric transmission factor, system transmission factor, target reflectance, and cosine of the incidence angle, and is inversely related to the square of the range. Figure 1.2 presents the theoretical relationship between the power returned and the inverse square



(a) Range

(b) Angle of incidence

Figure 1.2: Relationship of intensity to (a) range and (b) angle of incidence for two materials with different reflectance (ρ). Reproduced from Ref. [61].

of the range, R_i (Figure 1.2(a)), and the power returned and the cosine of the angle of incidence, α_i (Figure 1.2(b)) for two materials with different reflectance, ρ. This figure is reproduced from Ref. [61].

A variety of intensity correction methods have been derived from the range equation, with the most common based on the range and angle of incidence. Equation 1.1 displays the corrected intensity value, I_c such that intensity of all points would be equivalent if measured at the reference range, R_{ref} and an angle of incidence, α_i of zero. [61]

$$I_c = I \cdot \frac{R_i^2}{R_{ref}^2} \cdot \frac{1}{\cos \alpha_i} \tag{1.1}$$

The OSIRIS-REx Laser Altimeter (OLA) is a scanning lidar onboard OSIRIS-REx, and has been calibrated for the surface of Bennu. The low albedo of Bennu creates challenges for remote sensing [34], and was calibrated with a 3% Lambertian surface [31]. At a nominal power of 59 W, the high-energy laser has a maximum operational range of 9 km, with a range accuracy of < 0.31 m [31]. For the ASC GoldenEye flash lidar onboard OSIRIS-REx, the maximum range capability is 3 km with < 50 W of power and a 128 × 128 array. [4]

In this dissertation, an upper limit of 1 km is placed on the returned altimetry measurements

---

[4] http://www.advancedscientificconcepts.com/products/portable.html

in the flash lidar image, and a 64 × 64 pixel array is used. This is considered the maximum operational range, and assumes calibration of the target surface as well as the angle of incidence. In regards to the angle of incidence, this can be considered conservative, as a return was not recorded for a range-return larger than 1 km even if the incidence angle was 0°. In addition, if the surface is rough and the angle of incidence is high, a range still may be returned from boulders or rocks on the surface.

1.2.5 Autonomous Spacecraft Navigation

The dynamics around a small body are complex because of the typically irregular shapes and small masses that result in larger perturbations from other dynamical sources. With limited information known before visiting a small body, adequate characterization of the small body's gravity field, rotation axis, and overall dynamical environment is required before executing mission profiles such as landings or close-proximity operations.

Ground-based navigation operations are usually faster and more accurate during this characterization phase, and provide a more accurate model of the physical properties and dynamics surrounding the small body. These ground-based operations are limited by the roundtrip light-time delay of communications from the Earth to the spacecraft, and the time and processing power required to adequately characterize the small body.

Once the characterization phase is complete, the spacecraft may engage in trajectories that travel closer to the body such as low-altitude flying or landings for sample returns. These profiles require some degree of autonomy on the spacecraft, as the communications delay from Earth is usually too large to have ground-based operations in the loop.

Optical images, altimetry measurements, and the tracking of beacons on the surface (with range or Doppler measurements) are all feasible to be processed onboard the spacecraft. These relative measurements are usually coupled with onboard spacecraft measurements, such as attitude and acceleration estimates. Each of the relative measurements, however, requires the use of an onboard map fixed to the body in order to calculate residuals. This map may be of landmarks for optical images, surface topography for altimetry measurements, or the locations of beacons on the surface. The maps may be constructed on the ground and uploaded to the spacecraft to reduce the processing power required onboard.

The degree of spacecraft autonomy may be tailored for the specific mission. For example, a ballistic impactor may only need to execute one to three maneuvers, and have an active sensor for ensuring impact. A controlled descent requires a higher degree of autonomy, such as active OD and trajectory correction, altimetry measurements to control its rate of descent, and an onboard map to determine the corrections needed. A landing may be achieved if the characteristics of the small body are known precisely enough that maneuvers are able to be commanded from the ground. This was performed at the end of the NEAR mission, after extensive characterization of Eros. [96]

The onboard dynamics in a system and its representation of the "outside world" plays a role in its autonomy. A high fidelity model of the "outside world" requires more processing time, and may be dangerous if the real dynamics of the system are changing quickly. A low fidelity representation of the "outside world" results in quicker response times, but requires a higher reliability on its sensors.

Autonomy has been flown on spacecraft, including the Deep Impact mission, the MER Opportunity rover[40], and EO-1[25]. These autonomous systems incorporated autonomous navigation and autonomous observation planning. The following sections outline the autonomous navigation system AutoNav that was flown on Deep Impact, applications of AutoNav, and the proposed autonomous navigation system for the OSIRIS-REx TAG sequence.

1.2.5.1 Deep Impact Mission with AutoNav

The algorithm, AutoNav was used on the Deep Space 1 mission to the comet Borrelly, the Stardust mission to the asteroid Anne Frank and the comet Wild-2, and the Deep Impact mission to the comet Tempel 1.

The Deep Impact mission involved a Flyby spacecraft and an Impactor spacecraft, and on July 4, 2005, the Impactor spacecraft impacted Tempel 1 to create a crater and outburst that the

Flyby spacecraft imaged. Both the Flyby and Impactor spacecraft employed the AutoNav algorithm to autonomously detect an illuminated impact location and track this location for imaging. [65]

The Flyby and Impactor spacecraft were separated 24 hours before impact and ran identical AutoNav algorithms for the autonomous portion of the mission that spanned the two hours before impact. Both spacecraft autonomously computed the center of brightness of the images taken, and provided OD updates every minute. The Flyby spacecraft autonomously computed and updated the time of impact and time of final imaging. The Impactor spacecraft autonomously planned and executed three targeting maneuvers and aligned the camera with the computed velocity vector for high resolution images.

The residuals for the OD were determined from the observed and computed center of brightness in the image. This was computed with the Blobber algorithm that scanned the entire image for lit pixels, and the Centroid Box algorithm that computed first moment of the brightest pixels within a 400×400 centroid box. [65]

AutoNav was interfaced with the ground to receive the most up-to-date trajectory to initialize AutoNav, the ADCS, and the image sequencing. The image sequencing was controlled by AutoNav in the form of computing the time of impact and time of final imaging, and provided updates of these to the Flyby and Impactor spacecraft. [74]

The Impactor spacecraft took high-resolution images of the cometary surface seconds before impact, while the Flyby spacecraft tracked the predicted impact location for 800 seconds to image the plume. Exact imaging of the impact crater was not possible, however, because the brightness from the ejected plume lingered such that AutoNav directed the camera to point at the plume during the flyby. [65]

1.2.5.2 Planetary Landings with AutoNav

AutoNav for planetary landings has been proposed by incorporating it with the stereophotoclinometry (SPC) relative navigation methods. AutoNav was used for acquiring observations, performing orbit determination, and computing correction maneuvers based on this knowledge to follow a trajectory designed on the ground. AutoNav only used optical data in its navigation, which for proximity navigation was the landmarks provided from SPC. [13]

The objectives of the maneuvers were a three-dimensional location in the body-fixed frame and a target relative velocity. Only one of these three-dimensional states could be controlled. In the simulations, the position was controlled, and the final velocity was left as a free parameter. For both the asteroid landing and comet landing simulations, the position dispersion was within a five meter radius of the target landing site and landing velocities were within 2 cm/s of the target relative velocity. [13]

1.2.5.3 OSIRIS-REx Touch-and-Go Sequence

The OSIRIS-REx mission launched in September 2016, and to date, is on approach to the asteroid Bennu. Once arriving, the spacecraft will insert into orbit and perform a sample return to Earth. A touch-and-go (TAG) procedure at Bennu is planned to execute this sample return.

The nominal TAG sequence begins at a circular sun terminator orbit at 1 km. A deorbit maneuver is performed to arrive at the "Checkpoint" altitude of 125 m. A Checkpoint maneuver is performed here to cancel the lateral velocity and transfer 10 minutes later to the "Matchpoint" altitude of 45 m. A Matchpoint maneuver is performed next to achieve a targeted descent velocity of 10 cm/s for the TAG operations that are targeted for 8 minutes after this maneuver. The requirements for the TAG sequence are to land within a 25 m radius around the targeted TAG site. Before executing the full TAG sequence, two rehearsals are planned, one to the Checkpoint altitude, and one to the Checkpoint then Matchpoint altitude to ensure mission success. [9]

The estimated light-delay for communications is expected to be around 30 minutes, requiring autonomous operations of the spacecraft from the deorbit maneuver through the TAG sequence. Nominally, an lidar-based position estimation scheme was first proposed to reduce the uncertainty in the spacecraft state during descent. [9] At the critical design review of the mission, a backup autonomous navigation system was introduced, using optical images and landmark tracking, known as Natural Feature Tracking (NFT).[69] Both schemes are overviewed below.

For the nominal TAG sequence, two lidar beam measurements during descent are proposed to reduce the uncertainty of the spacecraft state. One lidar measurement will be taken once the limb of the asteroid is in view, and the second will be taken two minutes before the Checkpoint maneuver. The differences between the expected and observed values of the two lidar measurements are related to each state component by a two-dimensional second order polynomial. The position and velocity state will be solved for with six independent polynomials and 54 coefficients. These coefficients were solved for on the ground with a least-squares fit from Monte Carlo analysis. The computed state, based on these polynomials, will provide an update to the velocity and timing of the Checkpoint and Matchpoint maneuvers. Using this control scheme, the dispersion errors reduced to 12 m and 17 m in the nominal and worst-case scenarios, respectively, in Monte Carlo simulations. [9]

The NFT algorithm estimates the spacecraft state by comparing “landmark features” from the observed optical images with the predicted optical images. The onboard algorithm predicts which features it will see, renders those features, and then compares those features with the observed optical images. The features are rendered from a shape model that would be created from data previously acquired in the mission, albedo information of the surface, the predicted sun angle, and camera orientation. Image cross-correlation is used to match the predicted and observed features. The location of the correlation peak in the camera focal plane, and the correlation score determines the quality of the features. Various studies investigated creating the shape model with the onboard lidar altimeter and with SPC, as well as feature selection that is robust to model creation and correlation. [69]

1.2.6 Space Applications of Relative Navigation

Relevant work for relative navigation in space is categorized into the following: small body proximity operations; landings and terrain relative navigation; and spacecraft rendezvous and docking (R&D). Table 1.1 organizes relevant work in these space applications by relative navigation sensors. The published research is further organized by whether the work has been theorized (normal text), used in mission operations (**bold**), tested in space (underlined), or has autonomous capabilities (*italicized*). Publications that have originated from this dissertation are highlighted in red. Follow-on work that has cited publications from this thesis work are highlighted in gray.

Table 1.1: Relative navigation research in space applications, organized by sensor type and space application. Key: Theorized/Tested in Lab, **Used in mission operations**, Tested in space, *Autonomous capabilities*, Follow-on work, Thesis work

		Sensor					
		Optical Images	3D Vision Sensors	Optical + Lidar	Single/Multi-Beam Lidar	Scanning Lidar	Flash Lidar
Application	Small Body Proximity Operations	[76] **[77]** [23] **[45]** ***[74, 65]*** **[72]** **[73]** **[18]** [55]			[16] [75] [1]		[35, 36, 37, 38] [48]
	Landings and Terrain Relative Navigation	*[13]* *[69]* [58]		**[67]** **[39]** [5]	*[9]* [15]	*[32]*	[4, 3, 88] [86] [21] [90]
	Spacecraft Rendezvous and Docking	[41, 105] [41, 85]	[41, 63] [107] [41, 53]	***[87]***	[2, 41]	**[49, 17]** *[92]* **[29]**	[26] [91] [80]

Section 1.2.2 overviewed the research topics related to small body navigation from the "Small Body Proximity Operations" row of Table 1.1, and Section 1.2.3 highlighted the work done with relative navigation using a flash lidar sensor from the "Flash Lidar" column of Table 1.1. Section 1.2.5 described autonomous navigation systems as related to small body operations and landings.

The following sections highlight the research published in the remaining sections of Table 1.1. This includes an overview of other 3D lidar sensors, and their applications to planetary landings and hazard avoidance, and spacecraft R&D.

1.2.6.1 Three-Dimensional Lidar Sensors

A scanning lidar provides a 3D point cloud of its target. It uses one light source, one detector, and a series of mirrors or deflectors to change the direction of the lidar beam. It is relatively easy to calibrate and can direct its beam very precisely. Since a scanning lidar contains moving parts, potential for failure can occur. Additionally, scanning the scene with one lidar beam requires a finite amount of time; the target scene must not move faster than the time required to scan the scene, or the measurement will blur. [29]

A Continuous Wave (CW) lidar system is seen in robotics, computer vision, and home entertainment applications. It modulates the intensity of the projected laser and samples the return signal at least four times as fast as the modulated signal. This allows the time of flight of the laser to be easily obtained, yet must be used at close ranges, such as a few meters, as to avoid phase ambiguity. [29]

A flash lidar instantaneously captures a scene, while a scanning lidar needs a finite amount of time to scan and capture a scene. With the a CW lidar, the algorithms to calibrate a CW lidar are less complex than those to calibrate a flash lidar sensor, yet these lidars may only be used up to a few meters away, while a flash lidar may be used up to a few kilometers. The pixels in a flash lidar image may be treated as separate lidar beams with known orientation; therefore, one may know the instantaneous pointing directions of multiple lidar beams of a 3D point cloud.

1.2.6.2 Planetary Landings and Hazard Avoidance

Planetary landings are closely coupled to small body relative navigation in that both require terrain-relative navigation. With a landing, landing site selection and hazard avoidance are additional processes that are included in the landing algorithms. Due to the short time frame associated with landings, autonomous landings and hazard avoidance are usually investigated alongside the navigation. This was presented previously with the ALHAT mission [3], and with the OSIRIS-REx TAG sequence [9, 69].

Additional planetary landings investigations have studied an optical and lidar-aided simultaneous localization and mapping (SLAM) algorithm [5], and optical image matching and correlation with Fast Fourier Transforms (FFTs) [58]. With lidar sensors, a three-beam lidar and doppler system has been investigated [15], as well as a scanning lidar for autonomous hazard avoidance and landing, such as on Mars [32].

1.2.6.3 Spacecraft Rendezvous and Docking

Extensive work has been done in regards to spacecraft rendezvous and docking (R&D) and expanding its autonomous capabilities. Many of the same navigation sensors and algorithms investigated in the this field can be applied to relative navigation around small bodies. Most notable are the studies and applications of docking with the ISS.

The Trajectory Control Sensor (TCS) is a scanning lidar on the Space Shuttle that has been used for relative navigation for rendezvous with Mir, Hubble, and the ISS. The sensor switches from a pulse laser diode at long ranges greater than 1000 ft to a CW mode for ranges less than 1000 feet. The rendezvous and proximity operations program (RPOP) processed the relative position, velocity, and angle data on a separate laptop onboard the Space Shuttle and provided relative motion displays during proximity operations. [49, 17]

The ESA Automated Transfer Vehicle (ATV) operates a unique videometer that was specifically designed for docking with the ISS. A laser illuminates retro-reflectors installed on the service module, and the illumination is captured by a CCD sensor to provide range, line-of-sight angles and relative attitude within 30 m of rendezvous. With this sensor, the target is autonomously acquired and tracked during docking.[87]

The Neptec TriDar system was tested on STS-128 and STS-131 to the ISS to develop proximity operations and autonomous rendezvous and docking capabilities with noncooperative objects. This was done through using a 3D CAD model of the target instead of retro-reflectors or any features on the target. The TriDAR had onboard software to autonomously track a target object and estimate its relative orientation and position. Sparse data patterns were used to lock-on and track the target, and to initialize the ICP algorithm used for fine alignment. [92]

The "3D Vision Sensors" column in Table 1.1 refers to work done with spacecraft R&D that was not tied to a specific sensor. It was assumed that a 3D projection of the target scene could be acquired, and the referenced research focused on the algorithms used for navigation with that 3D image. These "3D Vision Sensors" included stereo cameras or lidar systems such as a scanning or flash lidar. The navigation methods investigated include ICP algorithms [92], motion prediction algorithms [53], featured-based algorithms [107], and model-based algorithms [85] [63]. An overview of these algorithms is presented below.

1.2.6.4 Terrestrial Relative Navigation

The use of lidar-based navigation systems are seen in terrestrial robotic applications and in small areal vehicles, such as quadrotors [106, 6, 51]. Onboard lidar sensors and inertial measurement units (IMUs) may determine the position of the vehicle, a map of its surroundings, or both, and this process is known as simultaneous localization and mapping (SLAM). Ground-based robots typically operate in a two-dimensional plane and are inherently stable [51], while airborne platforms face navigational challenges that are similar to those faced by an orbiting spacecraft such as fast dynamics, limited payload, and constant motion [6]. GPS-enabled air vehicles can easily determine their position and this can form a strong basis for autonomous navigation systems [106]. For GPS-denied areal vehicles, such as those operating inside, SLAM techniques include estimating the position and velocity of the vehicle with an onboard lidar scanner and probabilistic scan matching between 3D point clouds [83, 6], representing a 3D map with 2D representations [51], and using an iterative closest point (ICP) algorithm for surface matching [106]. These techniques are applicable to a spacecraft orbiting outside of the GPS constellation, such as at a small body, as it too would be denied GPS positioning capabilities.

1.2.7 Relative Navigation Algorithms

Algorithms to extract the relative position and pose information of a target object include the iterative closest point (ICP) algorithm, feature tracking algorithms, and model based algorithms. ICP algorithms are found primarily with 3D point clouds, while feature tracking and model-based algorithms are found with optical images and lidar images. Edge detection algorithms may be applied when a spacecraft is on approach to a celestial body, used in optical images for navigation, or used for correlating landmarks in optical images.

The ICP algorithm determines the relative translation and rotation between two 3D point clouds. Different variations of this algorithm have been studied, with one of the first instances found in Besl and McKay (1992) [10]. The basis of the ICP algorithm matches pairs of points within each point cloud by different methods, such as determining the closest point, normal shooting, and projection. [93] The sum of the square of the distance between each pair is minimized by applying and solving for a translation vector and rotation matrix. The translation vector and rotation matrix is solved by a quaternion-based method or a single value decomposition method [10]. Another ICP method often used minimizes the sum of the squared distances from the point to a plane, and may be solved by linearizing the problem.[93, 22] Applications of the ICP algorithm for 3D point clouds include the STORRM mission with a flash lidar [91], and autonomous rendezvous and docking with the ISS with a scanning lidar [92].

One drawback from using lidar measurements for navigation is their precision. The measurement function is usually nonlinear, yet a linear state update is applied when using conventional Kalman filtering techniques. This leads to errors in the state estimate and the state covariance does not accurately capture the state errors. The problem intensifies as the measurement becomes more accurate.[33] Both of these conditions apply to using lidar measurements for state estimation, and was seen when processing lidar measurements for relative navigation on the space shuttle. [116] Remedies to this problem have included iterating the state update with each observation, including the second order effects of the measurement function, and artificially inflating the measurement uncertainty.[116, 33] Investigations have found that an iterative Kalman filter is equivalent to the Gauss-Newton method that may be used to solve a linear least-squares problem.[8] Applying an iterative Kalman filter is essentially sequentially applying a linear least-squares fit to a nonlinear function.

A consider covariance is useful when it is known that errors exist in the measurement or dynamical models, yet these errors may be difficult or computationally intensive to estimate. A consider covariance systematically inflates the covariance to account for the effect the unmodeled errors have on the state estimate. [103] This approach is typically seen to account for the uncertainty in measurement biases such as seen in sensors [54], and may be expanded to include uncertainty in the dynamics as well [115].

Feature tracking algorithms determine certain features in a 2D or 3D image, and align these features between images. This is similar to the OpNav landmark tracking currently used in small body missions. With feature tracking algorithms, one must first build a database of the features to be tracked, and accurately match those between images. With planetary landings or terrain relative navigation, feature tracking algorithms include landmark tracking and crater identification and tracking, and can be performed with 3D lidar systems [90] or optical systems [5].

Feature tracking algorithms have been used and investigated in spacecraft R&D applications, including use as a contingency plan for docking with the Hubble Space Telescope [105], 3D feature mapping with a PMD (photonic mixer device) camera [107], and using retro-reflectors on the ISS for the ESA Automated Transfer Vehicle (ATV) [87] and JAXA H-II Transfer Vehicle (HTV)[29]. Feature tracking algorithms have been investigated with flash lidar for docking with the ISS in the STORRM mission[26, 91] with the ASC DragonEye [29].

Model-based algorithms typically use a 3D model of the target, and use this model to match the observed measurements to the computed measurements. These typically use the full 3D model within the algorithm, and no specific features are identified. This can be an advantage if the shape of the target is not known beforehand, and a feature database does not need to be constructed. Model-based algorithms have been investigated with spacecraft R&D by projecting a 3D model to a 2D surface [85], or extracting edges from camera images and minimizing their distances in a 2D plane[63]. A 3D model of the target has also been studied to produce 3D point data to be compared to a scanning lidar measurement through ICP [92].

Edge detection algorithms are a way to extract features from a scene to use in navigation. Relative spacecraft navigation with crater identification is an example of this application. In optical images, the edges of a crater may be extracted and fit to an ellipse.[23] A Fast Fourier Transform (FFT) for template matching between maps has been investigated for planetary landings with optical images.[58] With simultaneous localization and mapping (SLAM) algorithms, often used for relative autonomous navigation in robotics, feature detection algorithms aim to match the environment with corners, edges, trees (for outside applications), or curvature.[68] With optical navigation, limb scans may be used on approach to a small body. [72] Investigations have been underway to improve the detection of the limb of a moon or planet [28], fit the limb to an ellipse, and estimate the centroid of the moon or planet.[27]

1.3 Dissertation Overview

A flash lidar instrument is investigated for its performance as the primary data source for relative navigation around a small body when using Kalman filtering techniques. Flash lidar measurements may be used in any lighting conditions, include the spacecraft altitude, and bypass image processing and iteration procedures, all of which are advantages over OpNav. This instrument allows for simplification of the navigation algorithms and shows promise for advancing spacecraft navigation autonomy. The flash lidar images are processed with Kalman filters, and an OpNav OD simulation is used as a comparison to assess the OD performance of the flash lidar images.

Application of a model-based approach for the computation of the flash lidar images further reduces the computational load. This does not require the creation of a feature database, and avoids correlating the features within the images, as opposed to OpNav. In feature-tracking approaches, an onboard 3D model is required regardless to build the feature database and render onboard computed features. [69] Employing a model-based approach bypasses these additional steps. Model-based measurement algorithms are seen in spacecraft R&D applications, and in noncooperative debris-removal applications [92, 85, 63], but have yet to be employed in small body spacecraft navigation.

A flash lidar image may contain thousands of altimetry measurements in a single image. While each image contains significant information, processing a considerable number of measurements at one observation time can lead to underestimation of the uncertainty of the state errors. This is known as filter saturation, and has been seen when processing multiple precise lidar measurements for relative navigation [116]. Methods to mitigate filter saturation include iterating the state, using an underweighting approach that artificially inflates the measurement uncertainty matrix [116], and including second order effects of the measurement equation [33, 116]. Iteration of the relative state at one time step has been investigated with a nonlinear measurement function applied to space trajectories [33], and is seen in OpNav [72], spacecraft R&D research[63], and ICP algorithms [10].

The performance and limits of three sequential Kalman filters are evaluated with flash lidar images: an extended Kalman filter (EKF), an unscented Kalman filter (UKF), and an iterated least-squares (LS) filter. The iterative LS filter applies multiple linear state updates at one observation time before propagating the state and processing the next observation [8]. State iteration procedures are performed in OpNav as well within the image processing and correlation procedures, yet are separte from the OD process. The iterative LS filter combines the state iterations within the OD process so that observations may be processed sequentially.

Comparing two 3D point clouds is typically performed with an ICP algorithm that involves multiple iterations and a pair matching algorithm [10, 93, 92]. A flash lidar instrument may take advantage of the fact that the 3D point cloud is instantaneously captured, and one may directly match the returned altimetry measurements of the corresponding pixels in multiple image arrays. Overlapping two images omits a matching algorithm, and this is similar to how landmarks are compared in the image plane of an optical image. By omitting a matching algorithm, this further reduces computational time, yet requires the error between the truth and estimated state to be small enough such that an overlap occurs.

In addition to estimating the position and velocity, the ability to resolve a spacecraft pointing bias and random pointing error is assessed with the iterative LS filter. While OpNav and feature tracking approaches also are able to resolve the spacecraft pointing [12], it is more difficult to do so with a single beam [96] or scanning lidar system.

To support autonomous navigation, the relative simplicity and speed of the algorithms investigated is evaluated. A procedure is granted greater merit if it produces state errors consistent with the state uncertainty and is quicker than the other procedures investigated. An EKF is the simplest estimation filter for onboard estimation, however it may not always produce a consistent state estimate due to filter saturation. Implementing state iterations with the iterative LS filter may combat filter saturation at a lower computational cost than using a UKF. Implementing a low fidelity onboard shape model and optimally down-selecting the number of altimetry measurements to process both aim to further reduce the computation cost of measurement processing.

The image properties of the flash lidar measurements are examined for autonomous filter initialization. This may be useful if the spacecraft has exited safe mode and does not have knowledge of its relative state. With simple pre-processing algorithms, it may autonomously gain knowledge of its *a priori* position and pointing and initialize a sequential estimation filter. In addition, these algorithms address the challenge of ensuring an overlap occurs to compare observed and computed images. Similar procedures are seen in SLAM applications [83, 6], but have yet to be investigated for small body missions.

Optimally selecting a subset of pixels within the flash lidar image to process may further reduce the computational load and combat filter saturation. Reduction of measurements from a 3D point cloud is seen in ICP algorithms [93] and in spacecraft R&D applications [92].

1.3.1 Contributions

(1) Verified the performance of flash lidar measurements for orbit determination while in proximity to a small body is comparable to or better than the state-of-the-art of using optical images.

(2) Implemented a model-based approach for predicting the flash lidar measurements that simplified the measurement computation process when compared to a feature-based approach with optical images.

(3) Designed an iterative filter that handles multiple (lidar) measurements at one time within a sequential filter.

(4) Developed estimation framework to resolve a spacecraft pointing bias and pointing jitter with an iterative least-squares filter.

(5) Analyzed the robustness of the flash lidar measurements and estimation filters to shape modeling errors.

(6) Designed pre-processing algorithms to determine an *a priori* position, velocity, and pointing state to initialize the estimation filter using the flash lidar images and knowledge of the inertial spacecraft pointing and small body shape.

(7) Developed maximum-information pixel patterns within the flash lidar image to reduce the number of lidar measurements processed in the filter while maintaining the same level of accuracy.

1.3.2 Dissertation Outline

Chapter 2 details the methods used in this work and defines the estimation framework. This includes the estimation state definition, the state dynamics used in the simulations, and the analytical formulation of the flash lidar measurements. Optical navigation landmarks were used in a comparison simulation, and their analytical formulation is presented as well. An overview of the different estimation filters, such as the EKF, UKF, and iterative LS filter is provided, as well as a discussion of the consider covariance analysis performed and the definition of the Fisher Information Matrix. The realization of the flash lidar measurements in the estimation filters that includes their measurement partial derivatives with respect to the estimation state and their information content is specified.

Chapter 3 defines the small body orbits around Itokawa and Bennu that were investigated, as well as the definition of the flash lidar sensor model and measurement uncertainty. The orbits included stable circular terminator orbits around Itokawa and Bennu, a descent orbit around Itokawa, and an unstable eccentric orbit around Itokawa. OD simulations are presented in these various small body orbits, and the three estimation filters, the EKF, UKF, and LS filter, are compared. An OpNav simulation is provided for comparison with the flash lidar measurement simulations. The effect of including the spacecraft pointing in the estimation state is investigated, as well as the effect of adding process noise.

Chapter 4 introduces robustness studies of initial state errors, shape modeling errors, and estimating an off-nominal pointing correction. A Monte Carlo simulation while varying the initial state errors in each of the three estimation filters is studied to compare the filter state errors with the filter covariance bounds. Monte Carlo simulations are provided with the UKF and LS filter individually to characterize the limitations of these particular filters. Shape modeling errors were introduced by employing a low fidelity shape model in the filter and using a high fidelity shape model for the truth observations. A consider covariance analysis and consider covariance filter was implemented in this case to capture the shape modeling errors. The LS filter was used to solve a simulation with pointing jitter, in which a random error was applied to the spacecraft pointing at each time step.

Chapter 5 develops a pre-processing algorithm for filter initialization. With knowledge of the spacecraft's inertial pointing direction, the asteroid attitude, and an observation with the asteroid in view, this algorithm determines a relative position and pointing of the spacecraft. A framework is provided to initialize the filter with two successive observations and applications of this algorithm, and the accumulation of high-frequency observations. A Monte Carlo simulation with single observations is provided to characterize the limitations of this algorithm.

Chapter 6 determines the optimal pixel placement within a single flash lidar image in order to maximize the information returned on the state. From this analytical and numerical analysis, six pixel patterns are developed and compared for processing time and information content. Four of these pre-determined pixel patterns are tested in each of the three filters, and compared to an OD simulation with the full flash lidar image.

1.3.3 Publications

For reference, the journal publications that support this work are presented in reverse chronologically order. (* *Indicates first author*)

Journal Papers

- * Dietrich, A. and McMahon, J.W., "Filter Initialization and Optimization with Flash Lidar in Proximity to Small Bodies" In preparation for submission.

- * Dietrich, A. and McMahon, J.W., "Improved Robustness of Orbit Determination with Flash Lidar around Small Bodies," Journal of Guidance, Control, and Dynamics (in revision)

- * Dietrich, A. and McMahon, J.W., "Orbit Determination using Flash Lidar around Small Bodies," Journal of Guidance, Control, and Dynamics, Vol. 40, No. 3 (2017), pp. 650-665. doi: http://dx.doi.org/10.2514/1.G000615

Conferences

- * Dietrich, A. and McMahon, J. W., "Filter Robustness of Flash Lidar Based Navigation around Small Bodies." To be presented at AAS/AIAA Astrodynamics Specialist Conference. Stevenson, WA. August 2017

- Bercovici, B., Dietrich, A., and McMahon, J. W., "Autonomous Shape Estimation and Navigation using Lidar Observations." To be presented at AAS/AIAA Astrodynamics Specialist Conference. Stevenson, WA. August 2017

- Broschart, S., Bhaskaran, S., Bellerose, J., Dietrich, A., Han, D., Haw, R., Mastrodemos, N., Owen, W. M., Rush, B., and Surovik, D. "Shadow Navigation Support at JPL for the Rosetta Landing on Comet 67p/Churyumov-Gerasimenko." 26th International Symposium on Space Flight Dynamics ISSFD, Matsuyama, Japan. June 2017.

- * Dietrich, A. and McMahon, J. W., "Orbit Determination with Least-Squares and Flash Lidar Measurements in Proximity to Small Bodies." AIAA/AAS Astrodynamics Specialist Conference. San Antonio, TX. February 2017.

- * Dietrich, A. and McMahon, J. W. "Error Sensitivities for Flash LIDAR Based Relative Navigation around Small Bodies." AAS Guidance Navigation and Control Conference. Breckenridge, CO, February 2016.

- * Dietrich, A. and McMahon, J. W. “Autonomous Observation Planning with Flash LIDAR around Small Bodies.” AIAA/AAS Astrodynamics Specialist Conference. Vail, CO. August 2015.
- * Dietrich, A. and McMahon, J. W. “Asteroid Proximity Navigation using Flash LIDAR.” AIAA/AAS Astrodynamics Specialist Conference. San Diego, CA. August 2014.

Chapter 2

Orbit Determination Methods

Satellite navigation is the process of orbit determination (OD). This involves creating a model of the spacecraft's dynamical environment, and predicting the spacecraft's motion and expected measurements in this environment. The predicted measurements are compared to the truth observations, and the difference between the two provides an update to the current state of the spacecraft and how it will behave as it is propagated through time.

Typical measurements of a spacecraft may include range, range-rate, or relative angles from Earth to the spacecraft. Measurements relative to the spacecraft's central body are typical with small body missions, and may include optical images or altimetry measurements.

The estimation state at a minimum includes the spacecraft's position and velocity, and may include other parameters that help define the dynamical environment of the spacecraft and the expected measurements. Additional dynamical properties may be related to the gravity of the central body, gravitational forces separate from the central body, or atmospheric effects, to name a few. Measurement parameters may be related to measurement biases, the location of ground stations, or sensor pointing.

The OD measurement model will produce a predicted (computed) measurement, and compare it to the truth (observed) measurement. The measurements taken of the spacecraft state usually will not directly observe the spacecraft state, but exist in a nonlinear combination of the state. Through statistical models, linearizing the system with partial derivatives, and the difference between the observed and computed measurements, an estimate of the spacecraft state is produced. This state is then propagated through the dynamical model to predict its motion and the subsequent measurements.

Errors are likely to occur in the prediction of the spacecraft's motion and its predicted measurements. These can be due to errors such as unmodeled dynamics, linear approximations in the OD mathematical model, or errors in the observations. Therefore, to maintain an accurate estimation of the spacecraft state, the estimation procedure must be repeated as the spacecraft evolves in its dynamical environment.[103]

2.1 Estimation Framework

The estimation state is composed of the position, $\mathbf{r}$ and velocity, $\mathbf{v}$ of the spacecraft, or the position, velocity, and pointing, $\underline{\theta}$ of the spacecraft.

$$X = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \underline{\theta} \end{bmatrix} \tag{2.1}$$

The position and velocity of the spacecraft are defined with respect to the body in which the spacecraft was orbiting, and the pointing state is defined with Euler angles. The estimated position and velocity are defined in the asteroid-fixed inertial frame, and the rotation between the asteroid-centered body-fixed frame and the inertial frame is assumed to be known.

The angles in $\underline{\theta} = [\theta_1,\ \theta_2,\ \theta_3]$ represent a rotation from the nominal pointing direction of the spacecraft, $\hat{u}^*$ to the corrected pointing, $\hat{u}$. The angles correspond to a $3-1-2$ rotation, where the angle subscripts correspond to their axis of rotation. The error rotation is represented in Equation 2.2:

$$\hat{u} = C_2(\theta_2)C_1(\theta_1)C_3(\theta_3)\hat{u}^* \tag{2.2}$$

The x- and y-axes of sensor frame are defined as the flash lidar pixel array, and the $+z$-axis is the axis from which the nominal pointing is defined. The error rotation represents a twist rotation about the boresight direction (z-axis), and then a rotation about the x- and y-axes of pixel array.

The spacecraft dynamics are described by the velocity and acceleration of the spacecraft. The velocity, $\mathbf{v}$ describes the derivative of the position, $\mathbf{r}$, and the acceleration, $\mathbf{a}$ describes the derivative of the velocity, $\mathbf{v}$. The estimated pointing state describes an off-nominal pointing error for simplicity, and the pointing angle dynamics are set to zero.

$$\dot{X} = F(X) = \begin{bmatrix} \mathbf{v} \\ \mathbf{a} \\ 0 \end{bmatrix} \tag{2.3}$$

The acceleration, $\mathbf{a}$ includes the gravity of the central body described by spherical harmonics, solar radiation pressure from the sun, and third body effects from the sun. These equations of motion are described in more depth in Section 2.2.

Flash lidar measurements are the sole measurements used for relative navigation in this thesis. The flash lidar measurements are taken from the spacecraft and directed at the small body. These measurements are used in the estimation filters to produce updates to the estimation state, X. The analytical formulation of the flash lidar images are outlined in more detail in Section 2.3 and the measurement partial derivatives are outlined in Section 2.6.

2.2 Small Body Dynamical Environment

The typically irregular shape and low mass of small bodies leads to varying gravity dynamics dependent on the body-relative position of an orbiting object. The small mass leads to a small gravitational parameter, and an orbiting spacecraft is easily effected by other dynamical forces such as the sun, solar radiation pressure, and other planets. A comet or active asteroid may have outgassing as well that may impart an additional acceleration or drag on the orbiting spacecraft and may effect the comet's heliocentric ephemeris.

While ground-based observations provide information on the physical shape and rotation of a small body, typically this information is insufficient to fully design a small body mission until the spacecraft arrives at and further characterizes the small body. [11] Therefore, many physical parameters are estimated along with the spacecraft position and velocity when navigating around a small body. These parameters may include gravity field coefficients, the orientation and spin axis of the small body, the location of landmarks on the surface, and non-gravitational accelerations due to the spacecraft and outgassing from the small body. [77] [64] [11] This work does not estimate the dynamical parameters of the small body environment, but acknowledges that this is performed in typical small body missions.

The dynamics modeled for this research involve the main contributing sources of dynamical perturbations. The dynamical models include modeling the gravity field with a spherical harmonics expansion, modeling solar radiation pressure on the spacecraft as a cannonball model, and including third-body effects from the sun. The following sections detail the analytical expressions used for the accelerations, a comparison of these accelerations, and a discussion on the stability of small body orbits.

2.2.1 Spherical Harmonic Gravity Modeling

Small bodies tend to have irregular, non-spherical shape, and modeling their gravity as a point mass is insufficient when orbiting such a body. The gravity of a small body is therefore typically modeled with such methods as a spherical harmonic expansion [62] or a polyhedron model [111], which take into account the body-relative position of the spacecraft. Modeling the gravity field with a spherical harmonic expansion is typically seen in small body missions [77, 64, 18], as it is simple to estimate the coefficients in the estimation state to update the dynamics.

Equation 2.4 shows the formulation for the spherical harmonics expansion, where $C_{l,m}$ and $S_{l,m}$ are the spherical harmonic coefficients, and l and m are the degree and order, respectively. It is of note that this formulation assumes the body-fixed coordinate system aligns with the center of mass of the body, and hence the coefficients $C_{1,0}$, $C_{1,1}$, and $S_{1,1}$ are all zero.

$$U = \frac{\mu}{r} \sum_{l=2}^{\infty} \sum_{m=0}^{l} \left(\frac{R_s}{r} \right)^l P_{l,m}(\sin\phi) \left(C_{l,m} \cos(m\lambda) + S_{l,m} \sin(m\lambda) \right) \tag{2.4}$$

Equation 2.4 shows simply the spherical harmonics expansion of the potential due to the variations in the gravity field. The total potential due to the gravity of the body is defined by adding in the point mass gravity to this potential, and this is shown in Equation 2.5.

$$U = \frac{\mu}{r} + \frac{\mu}{r} \sum_{l=2}^{\infty} \sum_{m=0}^{l} \left(\frac{R_s}{r} \right)^l P_{l,m}(\sin\phi) \left(C_{l,m} \cos(m\lambda) + S_{l,m} \sin(m\lambda) \right) \tag{2.5}$$

Here, μ is the gravitational parameter, r is the position magnitude with respect to the center of mass, R_s is the reference radius of the small body, ϕ and λ are the latitude and longitude, respectively, and $P_{l,m}(\ )$ are the Legendre polynomials.[50, 108]

As the degree and order increase, the value of the corresponding coefficients usually decreases, and therefore to avoid machine precision round-off errors, the coefficients are usually normalized. Equation 2.6 defines this normalization.[108]

$$\begin{gathered} \Pi_{l,m} = \sqrt{\frac{(l+m)!}{(l-m)!k(2l+1)}} \\ \bar{S}_{l,m} = \Pi_{l,m} S_{l,m} \quad \bar{C}_{l,m} = \Pi_{l,m} C_{l,m} \quad \bar{P}_{l,m} = \frac{P_{l,m}}{\Pi_{l,m}} \\ k = 1 \text{ if } m = 0 \quad k = 2 \text{ if } m \neq 0 \end{gathered} \tag{2.6}$$

Equation 2.5 describes the potential on the spacecraft, therefore the acceleration due to the gravity is the derivative of this potential with respect to the position of the spacecraft, where $\mathbf{r}$, is the planet-fixed position of the spacecraft:

$$\mathbf{a}_{SH} = \frac{\partial U}{\partial \mathbf{r}} \tag{2.7}$$

The gravitational acceleration is implemented with the formulation in Reference [50]. This formulation defines the latitude and longitude coordinates in terms of a body-fixed cartesian frame, and provides further simplification of the coefficients and Legendre polynomials in order to improve performance of numerically integration.

2.2.2 Solar Radiation Pressure

The solar radiation pressure (SRP) acceleration is modeled as a cannonball model in Equation 2.8,

$$\mathbf{a}_{SRP} = -p_{AU} C_R \left(\frac{A}{m}\right) \frac{\mathbf{r}_{\odot/sc}}{r^3_{\odot/sc}} (1\text{AU})^2 \tag{2.8}$$

where, $p_{AU} = (1357\ \text{W/m}^2)/(3\text{E}8\ \text{m/s})$, C_R is the coefficient of reflectivity set to 1.2, the area to mass ratio, $\frac{A}{m} = 0.01\ \text{m}^2/\text{kg}$, $\mathbf{r}_{\odot/sc}$ is the vector from the spacecraft to the sun, and $r_{\odot/sc}$ is its magnitude. [108]

2.2.3 Sun Third Body Effects

The third body perturbations from the sun are expressed in Equation 2.9,

$$\mathbf{a}_3 = \mu_\odot \left(\frac{\mathbf{r}_{sc/\odot}}{r^3_{sc/\odot}} - \frac{\mathbf{r}_{ast/\odot}}{r^3_{ast/\odot}} \right) \tag{2.9}$$

where $\mu_\odot$ is the gravitational parameter of the sun, $\mathbf{r}_{sc/\odot}$ is the vector from the sun to the spacecraft with $r_{sc/\odot}$ as its magnitude, and $\mathbf{r}_{ast/\odot}$ is the vector from the sun to the asteroid, with $r_{ast/\odot}$ as its magnitude. The sun is held fixed with respect to the asteroid since the simulations are over a short enough period of time that this movement is negligible.

2.2.4 Estimation Dynamics

The estimation state at a maximum involves the position, $\mathbf{r}$, velocity, $\mathbf{v}$, and pointing error, $\underline{\theta}$, of the spacecraft. The equations of motion of this state are defined as:

$$\dot{X} = F(X) = \begin{bmatrix} \mathbf{v} \\ \mathbf{a} \\ 0 \end{bmatrix} \tag{2.10}$$

where

$$\begin{aligned} \mathbf{v} &= \dot{\mathbf{r}} \\ \mathbf{a} &= \ddot{\mathbf{r}} = \mathbf{a}_{SH} + \mathbf{a}_{SRP} + \mathbf{a}_3 \end{aligned} \tag{2.11}$$

The acceleration is the derivative of the velocity, and is the addition of the three contributing sources: gravity ($\mathbf{a}_{SH}$), SRP ($\mathbf{a}_{SRP}$), and third body effects from the sun ($\mathbf{a}_3$). The velocity is the derivative of the position, and the derivative of the pointing error derivative is zero.

2.2.5 Comparison of Accelerations

Figure 2.1 compares the effects of the different accelerations while the spacecraft is in a 1 km circular terminator orbit about the asteroid, Itokawa. The spherical harmonics include up to degree and order four. The accelerations due to the central body are the main contributing factor to the spacecraft motion, while the perturbations due to SRP and the spherical harmonics are close to the same order. One can notice the variations in acceleration from the spherical harmonics perturbations as the spacecraft orbits through different latitudes and longitudes of Itokawa. These small perturbations are usually negligible when a spacecraft is orbiting a larger body, but play a larger role in the small body environment.

2.2.6 Small Body Terminator Orbits

Orbits about small bodies have a tendency to become unstable due to the non-uniform gravity field and perturbations from SRP. It has been proven however, that stable orbits exist with the correct selection of the eccentricity and inclination of the orbit. If the direction of the SRP force, $\hat{\mathbf{d}}$ (away from the sun) is not rotating, frozen orbits exist for an eccentricity of $e = 0$ or when the angular momentum vector of the orbit, $\mathbf{h}$, is along the direction of $\hat{\mathbf{d}}$. In addition, an orbit that is initialized in the plane perpendicular to $\hat{\mathbf{d}}$ will remain circular if $\hat{\mathbf{d}}$ is not rotating.

For the case when the direction of the SRP force is rotating, a frozen orbit exists in the plane perpendicular to the sun line, also known as the terminator plane. Furthermore, as the SRP perturbation becomes larger, the orbit becomes more circular. This makes these orbits preferred in situations with strong perturbations. Additional care must be taken to prevent escape in regards to the perturbations due to the oblateness of the body, the rotation rate of the body, and the semi-major axis of orbit. Although, once the orbit pole of an asteroid is determined, it is possible

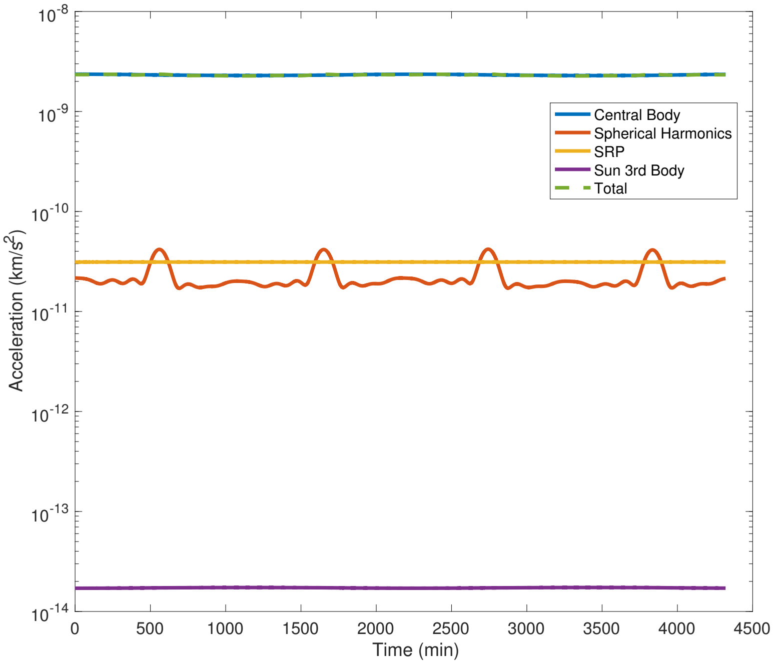


Figure 2.1: Accelerations in a 1 km circular terminator orbit around Itokawa for 3 days (approximately 2 orbital periods)

to specify when this additional care is necessary for the terminator orbits with respect to their interaction with the gravity field of the asteroid. [97]

A circular orbit in the plane parallel to the sun terminator plane is often used as a stable orbit for proximity operations, and will serve as a “home base” orbit for the OSIRIS-REx mission. [55] While these stable orbits exist, accurate and consistent OD must still be performed while orbiting a small body in order to ensure the safety of the spacecraft.

2.3 Flash Lidar Measurements

A flash lidar instrument has the ability to return the light intensity of an object in its field of view (FOV), or the range to that object. In this work, the flash lidar measurements are considered to the be the observed ranges of the target in the FOV with errors based on the observed range.

2.3.1 Analytical Representation

A flash lidar image may be considered as different vectors originating at a focal point, directed through the sensor frame, toward a target scene. The vector magnitude of these beams is the range and the vector direction corresponds to an image pixel. Figure 2.2 illustrates the asteroid body-fixed vector diagram of one range measurement, $\boldsymbol{\rho} = \rho\hat{u}$, and the sensor plane at the spacecraft. In this problem, the magnitude of $\boldsymbol{\rho}$ was measured from the sensor plane to the surface of the asteroid. In the diagram, $\mathbf{r}_{sc}$ is the position of the spacecraft with respect to the asteroid body-fixed frame, $\mathbf{s}_{xy}$ is the position of the pixel on the sensor plane, **fp** is the focal point, and $\mathbf{r}_A$ is the vector from the asteroid center to the asteroid surface where the lidar beam intersects the surface.

Equation 2.12 expresses the range measurement as a function of the asteroid surface radius and spacecraft position. This equation removes the focal point sensor geometry, and assumes that the range vector originates at the spacecraft state, $\mathbf{r}_{sc}$. The value ρ is treated as the reported range, and the pointing vector $\hat{u}$ is treated as the pointing direction associated with a specific pixel.

$$\rho\hat{u} = \mathbf{r}_A - \mathbf{r}_{sc} \tag{2.12}$$

An example of a flash lidar image is shown in Figure 2.3(a), viewing the north pole of Itokawa. Figure 2.3(a) is an image; the axes on the plots represent the pixel number, and the color bar represents the range in kilometers. The asteroid shape models used in this work are described as facet/vertex models. An example of the shape model of Itokawa is illustrated in Figure 2.3(b). The computed flash lidar measurements are calculated with a ray-tracing technique that is explained in more detail in Section 2.3.2.

2.3.2 Generation of Flash Lidar Images

To generate the computed flash lidar measurements, ray-tracing is used with each lidar beam. The shape of the asteroid is represented as a facet/vertex model, and each lidar beam was represented by a pointing direction, $\hat{u}$ in the flash lidar array. Given the individual pointing directions of the lidar beams and the facet/vertex model of the asteroid, a range for each beam is computed

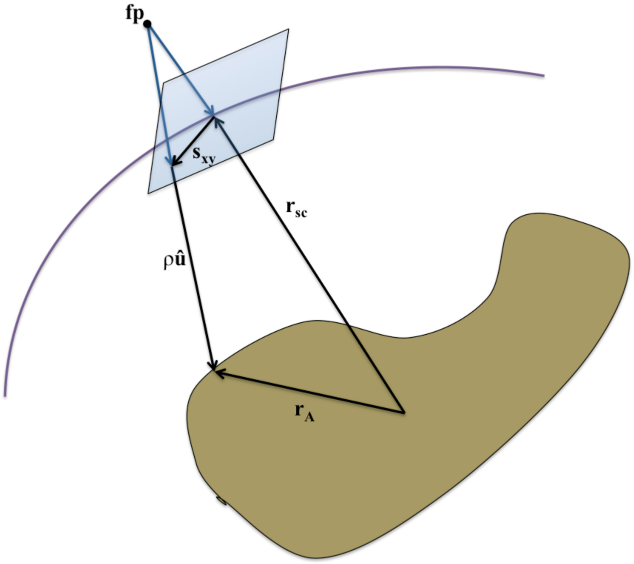
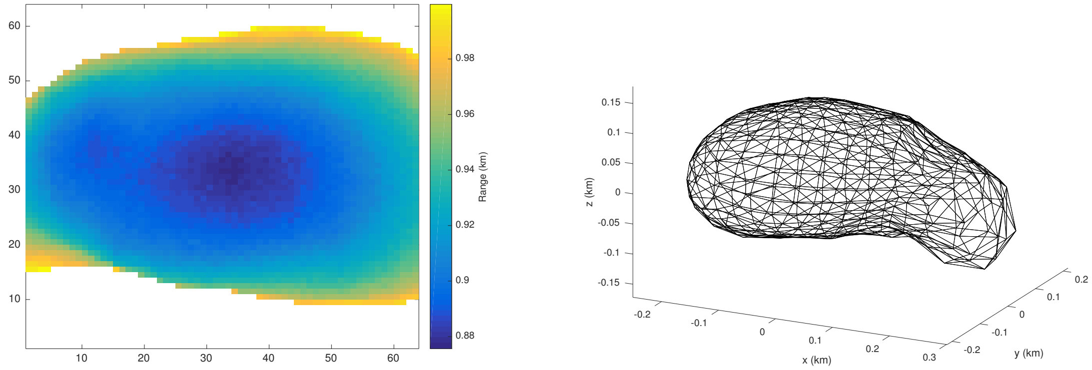


Figure 2.2: Flash lidar vector diagram depicting one lidar vector to the asteroid surface that returns a range measurement.



(a) Flash LIDAR measurement of Itokawa

(b) Facet/vertex model of Itokawa

Figure 2.3: Depiction of (a) a sample flash lidar measurement of Itokawa and (b) a facet/vertex shape model of Itokawa.

from the origin of the spacecraft to the closest intersection of that lidar beam with the shape model.

The pointing direction of each lidar beam is defined in the sensor frame from the focal point of the lidar array to its specified pixel location in the array (see Figure 2.2). The individual pointing vector is rotated from the sensor frame to the asteroid body-fixed frame, and offset by the asteroid body-fixed position of the spacecraft.

Each facet in the shape model is tested to determine distance from the spacecraft to the plane of the facet, and point where the lidar beam intersects that plane. Equation 2.13 describes the range value, ρ from the spacecraft to the plane of interest, where V_1 is a point on that plane (a vertex in the model), ${}^{BF}\mathbf{r}$ is the position of the spacecraft in the asteroid-centered body-fixed frame, $\hat{n}$ is the normal vector to the facet of interest, and $\hat{u}$ is the pointing direction. Equation 2.14 describes the point, p where the lidar beam intersects the plane.

$$\rho = \frac{(V_1 -^{BF} \mathbf{r}) \cdot \hat{n}}{\hat{u} \cdot \hat{n}} \tag{2.13}$$

$$\mathbf{p} = \rho \hat{u} +^{BF} \mathbf{r} \tag{2.14}$$

Once the range, ρ and point of intersection to the plane of interest, $\mathbf{p}$ is computed, it is determined whether the point, $\mathbf{p}$ is within the bounds of the facet. Given the three vertices that compose the facet, V_1, V_2, V_3, and the point of intersection, $\mathbf{p}$, the following equations are used to determine if $\mathbf{p}$ is within the facet.

The following vectors are defined as the differences between the vertices and point of intersection, and the matrix, M is composed of two of these difference vectors ($\mathbf{v}_1$ and $\mathbf{v}_2$) and the surface normal vector, $\hat{n}$.

$$\begin{aligned} \mathbf{v}_0 &= V_3 - V_1 \\ \mathbf{v}_1 &= V_2 - V_1 \\ \mathbf{v}_2 &= \mathbf{p} - V_1 \end{aligned} \tag{2.15}$$

$$M = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_0 & \hat{n} \end{bmatrix} \tag{2.16}$$

The two vectors, $\mathbf{a}_1$ and $\mathbf{a}_2$ are defined from the sub-determinants of M divided by the determinant of M:

$$\mathbf{a}_1 = \begin{bmatrix} \dfrac{\begin{vmatrix} \mathbf{v}_0(2) & \hat{n}(2) \\ \mathbf{v}_0(3) & \hat{n}(3) \end{vmatrix}}{|M|} & \dfrac{\begin{vmatrix} \mathbf{v}_0(1) & \hat{n}(1) \\ \mathbf{v}_0(3) & \hat{n}(3) \end{vmatrix}}{|M|} & \dfrac{\begin{vmatrix} \mathbf{v}_0(1) & \hat{n}(1) \\ \mathbf{v}_0(2) & \hat{n}(2) \end{vmatrix}}{|M|} \end{bmatrix} \tag{2.17}$$

$$\mathbf{a}_2 = \begin{bmatrix} \dfrac{\begin{vmatrix} \mathbf{v}_1(2) & \hat{n}(2) \\ \mathbf{v}_1(3) & \hat{n}(3) \end{vmatrix}}{|M|} & \dfrac{\begin{vmatrix} \mathbf{v}_1(1) & \hat{n}(1) \\ \mathbf{v}_1(3) & \hat{n}(3) \end{vmatrix}}{|M|} & \dfrac{\begin{vmatrix} \mathbf{v}_1(1) & \hat{n}(1) \\ \mathbf{v}_1(2) & \hat{n}(2) \end{vmatrix}}{|M|} \end{bmatrix} \tag{2.18}$$

The remaining difference vector ($\mathbf{v}_2$) is dotted with both of the vectors, $\mathbf{a}_1$ and $\mathbf{a}_2$:

$$\begin{aligned} \nu_1 &= \mathbf{a}_1 \cdot \mathbf{v}_2 \\ \nu_2 &= \mathbf{a}_2 \cdot \mathbf{v}_2 \end{aligned} \tag{2.19}$$

If the resulting values of ν_1 and ν_2 are positive, and the sum of $\nu_1 + \nu_2 \leq 1$, then the point $\mathbf{p}$ is inside the facet.

If $\mathbf{p}$ is determined as inside the facet of interest, the range ρ is recorded as the range for that lidar beam pixel. If more than one range, ρ is returned for one lidar beam, the smallest range is recorded, as that would be the facet that the lidar beam would encounter first.

2.4 Optical Navigation Measurements

The measurements for optical navigation (OpNav) are "landmarks" that are tied to the surface of the body. The landmark locations for the OpNav measurements are defined as vectors from the center of the body to the surface of the body. It is assumed that a small body characterization phase has been completed, a shape model has been defined, and the body-fixed position vectors of these landmarks are known.

During typical OpNav operations, the incoming optical images are used to create a 3D shape model of the small body through stereophotoclinometry (SPC). This involves predicting the position of the spacecraft, the pointing of the camera, the direction of the sun, and the albedo of the surface. Once the shape model is created, SPC is used to create landmark maps (L-maps) on the surface to define landmark locations. A correlation process occurs between the landmark locations and the optical images to determine the body-fixed locations of the landmarks. Once the landmark locations are determined, the locations are passed to the OD team to use in the navigation filters. It was assumed that this process had previously been completed, and the landmark locations were available to perform OD.

The landmarks are determined to be in view of the camera through three metrics: (1) The line of sight vector (LOS) from the focal point to the landmark is within the field of view (FOV), (2) The landmark is sunlit, and (3) The landmark is not behind the asteroid, which is determined through a facet search with the LOS vector. The landmark is considered to be sunlit if the angle between the landmark vector, ${}^{I}LM$ and the vector from the landmark to the sun is less than 90°. To determine if a landmark is behind the asteroid, a facet search of the shape model is conducted similar to the ray-tracing algorithm presented in Section 2.3.2. Using the ray-tracing technique, the range of the LOS vector and facet it hit are recorded. If the LOS vector encounters a facet from the shape model before it encounters its corresponding landmark, this landmark is considered behind the asteroid and out of view.

The landmark vectors, ${}^{BF}LM$, are rotated from the body-fixed frame to the inertial frame, ${}^{I}LM$, based on the current time step ($C_{BF/ACI} = C_3(\theta_0 + t_c)$), and the residuals are computed in the inertial frame.

The sample/line (s, l) coordinates of the landmarks in the image frame are determined from the LOS vector. This formulation is adapted from Owen (2011)[84]. The LOS vector is rotated from the inertial frame to the camera frame and projected on to the camera plane through Equation 2.20, where $(x,\ y)$ is the projected LOS vector and fl is the focal length:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{fl}{{}^{Cam}LOS_3} \begin{pmatrix} {}^{Cam}LOS_1 \\ {}^{Cam}LOS_2 \end{pmatrix} \tag{2.20}$$

The projected LOS vector is converted into sample/line coordinates with Equation 2.21:

$$\begin{pmatrix} s \\ l \end{pmatrix} = K \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} s_0 \\ l_0 \end{pmatrix} \tag{2.21}$$

The origin of the sample/line coordinate system, (s_0, l_0), is defined as $(0, 0)$ and $K = I(1\text{px/km})$. The conversion kilometer per pixel is defined by Equation 2.22, where fl is the focal length and n_{px} is the total number of pixels across one axis. With the sensor model, $fl = 10$ m, $FOV = 20°$, and $n_{px} = 64 \times 64$, this is approximately 5.51E-5 km/px.

$$fl\frac{\tan(FOV/2)}{n_{px}/2}\text{km/px} \tag{2.22}$$

2.5 Estimation Filtering Methods

Linear estimation methods involve linearizing the state dynamics about a reference state and providing a linear update to the estimation state. The measurements taken of the estimation state are usually a nonlinear combination of the estimation state, and therefore the measurements are linearized through partial derivatives as well. The unscented Kalman filter (UKF) is a nonlinear filter and avoids computing these partial derivatives. The UKF estimates a state by propagating a statistically determined set of states through the nonlinear dynamics and computes a measurement for each of these states.

The following sections outline the linearization process of the state dynamics and measurements with respect to the estimation state, X. The estimation filters include a least-squares solution and a minimum variance estimator. The minimum variance estimator is used in the sequential Kalman filter, which is outlined below along with the extended Kalman filter (EKF). An unscented Kalman filter (UKF) and an iterative least-squares (LS) filter are presented below as well. The simulations in the following chapters compared the performance of the EKF, UKF, and LS filter methods.

2.5.1 Linearization

The estimation state is linearized around a reference trajectory, and the deviation from the reference trajectory is the solution to the estimation procedures. The reference state is defined as X^*, the truth state is defined as X, and the deviation from the truth state is defined as $\mathbf{x}$.

Therefore, the truth state is defined as:

$$X = X^* + \mathbf{x} \tag{2.23}$$

A Taylor series expansion is used to linearize the dynamics, $\dot{X}(t) = F(X,t)$, around the reference trajectory, X^*. With higher order terms ignored, the Taylor series expansion is:

$$F(X,t) = F(X^*,t) + \left(\frac{\partial F(X,t)}{\partial X(t)}\right)^* (X(t) - X^*(t)) \tag{2.24}$$

The derivative of the dynamics with respect to the state is defined as the A matrix, and is evaluated along the reference state:

$$A(t) = \left(\frac{\partial F(X,t)}{\partial X(t)}\right)^* \tag{2.25}$$

The deviation in the dynamics can then be written in state space form, where $\dot{\mathbf{x}}(t) = F(X,t) - F(X^*,t)$ and $\mathbf{x} = X(t) - X^*(t)$:

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x} \tag{2.26}$$

The observations are linearized with a Taylor series expansion around the measurements evaluated along the reference trajectory, $G(X^*,t)$. The error in the truth observations, Y, and the computed measurements is represented by $\boldsymbol{\epsilon}$, and is assumed to be Gaussian white noise.

$$Y = G(X^*,t) + \left(\frac{\partial G(X^*,t)}{\partial X(t)}\right) (X(t) - X^*(t)) + \boldsymbol{\epsilon} \tag{2.27}$$

The measurement mapping matrix between the observations and the estimation state at time t is defined as $\tilde{H}$, and is evaluated with the reference trajectory:

$$\tilde{H} = \frac{\partial G(X^*,t)}{\partial X(t)} \tag{2.28}$$

The observation deviation is defined as $\mathbf{y} = Y - G(X^*,t)$, and the observation deviation equation becomes:

$$\mathbf{y} = \tilde{H}\mathbf{x} + \boldsymbol{\epsilon} \tag{2.29}$$

The state transition matrix (STM), $\phi(t_k, t_{k-1})$ is used to propagate the state deviations from time t_{k-1} to t_k:

$$\mathbf{x}(t_k) = \phi(t_k, t_{k-1})\mathbf{x}(t_{k-1}) \tag{2.30}$$

The STM is propagated through the state dynamics with the A matrix:

$$\dot{\phi}(t, t_k) = A(t)\phi(t, t_k) \tag{2.31}$$

Measurement mapping matrices produced at different times may be related to a single epoch time with the STM. This is useful if measurements are taken at various times throughout the propagation because one may use the information from all of these observations to the estimate the state at an epoch time. The combination of the multiple measurement mapping matrices, $\tilde{H}$ related to an epoch time is defined as [103]:

$$H = \begin{bmatrix} \tilde{H}_0\phi(t_0, t_k) \\ \vdots \\ \tilde{H}_l\phi(t_l, t_k) \end{bmatrix} \tag{2.32}$$

2.5.2 Least-Squares Solution

The method of least-squares is used to minimize a cost function, $J(\mathbf{x})$ based on the errors of the observations:

$$J(\mathbf{x}) = \frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} \tag{2.33}$$

By substituting in the observation deviation equation, the cost function becomes:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{y} - H\mathbf{x})^T(\mathbf{y} - H\mathbf{x}) \tag{2.34}$$

To minimize this cost function, the derivative of J with respect to the state deviation, $\mathbf{x}$ is set equal to zero, and the second derivative is ensured to be positive. Following this math procedure, the best estimate of the state deviation is [103]:

$$\hat{\mathbf{x}} = (H^T H)^{-1} H^T \mathbf{y} \tag{2.35}$$

2.5.3 Minimum Variance Estimator with *a priori* Information

The minimum variance estimator is the basis of the Kalman filtering techniques explained in this chapter. Unlike the least-squares estimation solution, this estimator includes information on the uncertainty of the measurements, as well as *a priori* information on the estimation state and state covariance.

The minimum variance estimator assumes that the observation errors are random, Gaussian white noise with a specified covariance, and requires the estimator to be linear, unbiased, and minimize the variance. The observational errors, $\boldsymbol{\epsilon}$, are assumed to be of a Gaussian distribution with zero mean and covariance R:

$$E[\boldsymbol{\epsilon}] = 0,\ E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = R \tag{2.36}$$

The observational errors are assumed not to be time-correlated, defining R as a diagonal matrix.

With these assumptions and requirements, and using the measurement mapping matrix, H, observation uncertainty covariance, R, and difference in the observed and computed measurements, $\mathbf{y}$, it follows that the best estimate of the state deviation at time t_k is:

$$\hat{\mathbf{x}}_k = (H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} \mathbf{y}_k \tag{2.37}$$

with a minimum covariance:

$$P_k = (H_k^T R_k^{-1} H_k)^{-1} \tag{2.38}$$

The inclusion of an *a priori* state, $\bar{\mathbf{x}}_k$ and *a priori* covariance, $\bar{P}_k^{-1}$ is folded into the best estimate of the state and its covariance:

$$\hat{\mathbf{x}}_k = (\tilde{H}_k^T R_k^{-1} \tilde{H}_k + \bar{P}_k^{-1})^{-1} (\tilde{H}_k^T R_k^{-1} \mathbf{y}_k + \bar{P}_k^{-1} \bar{\mathbf{x}}_k) \tag{2.39}$$

$$P_k = (\tilde{H}_k^T R_k^{-1} \tilde{H}_k + \bar{P}_k^{-1})^{-1} \tag{2.40}$$

In Equation 2.39 and 2.40, the measurement mapping matrix is evaluated by $\tilde{H}_k$, which is evaluated at time t_k, instead of multiple mappings to an epoch time as is with H_k. [103]

2.5.4 Sequential Kalman Filter

In the sequential Kalman filter, the measurements are processed at each observation time, as opposed to mapping multiple measurements to one specific observation epoch. A similar approach to the minimum variance method above is applied, with the addition of the computation of a Kalman gain, K_k. An advantage to this approach is that the matrix to be inverted is the same dimensionality as the number of observations at that time step, instead of the number of parameters estimated. Therefore, if observations are processed one at a time, then only scalar operations are necessary.

To initialize the filter, the best estimate and minimum variance are propagated from the previous time, t_{k-1} to the next observation time, t_k by the STM. The propagated state and covariance then becomes the *a priori* state, $\bar{\mathbf{x}}_k$, and covariance, $\bar{P}_k$ at time t_k:

$$\bar{\mathbf{x}}_k = \phi(t_k, t_{k-1})\hat{\mathbf{x}}_{k-1} \tag{2.41}$$

$$\bar{P}_k = \phi(t_k, t_{k-1})P_{k-1}\phi^T(t_k, t_{k-1}) \tag{2.42}$$

A computed measurement is made with the reference trajectory state, and compared to the truth observation. The computed measurement, $G(X_k^*)$, is subtracted from the observed measurement, Y_k, to form the observation deviation vector: $\mathbf{y}_k = Y_k - G(X_k^*)$. The measurement mapping matrix, $\tilde{H}_k$ (Equation 2.28) is calculated with the computed measurement and the reference trajectory state.

The Kalman gain is defined from the measurement mapping matrix, the *a priori* covariance, and the measurement uncertainty matrix:

$$K_k = \bar{P}_k\tilde{H}_k^T(\tilde{H}_k\bar{P}_k\tilde{H}_k^T + R_k)^{-1} \tag{2.43}$$

The state deviation update and the covariance are derived with the Kalman gain:

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + K_k(\mathbf{y}_k - \tilde{H}_k\bar{\mathbf{x}}_k) \tag{2.44}$$

$$P_k = (I - K_k\tilde{H}_k)\bar{P}_k \tag{2.45}$$

where I is the identity matrix.

An alternative form of the covariance matrix is the Joseph formulation of the covariance:

$$P_k = (I - K_k\tilde{H}_k)\bar{P}_k(I - K_k\tilde{H}_k)^T + K_kR_kK_k^T \tag{2.46}$$

This form always yields a symmetric result of the covariance. With Equation 2.45, the covariance can lose its symmetric properties if the *a priori* covariance is large, and the measurements are very precise, causing the covariance to shrink quickly. Therefore, due to its stability, the Joseph formulation is used in the subsequent simulations.

For the conventional Kalman filter, the state deviation solution at each time step is recorded. For the next observation time, the reference trajectory state, the previous state deviation and the covariance are propagated forward, a state deviation and covariance solution are recorded, and the process is repeated. The result is a recorded state deviation and covariance matrix at each observation time. The reference trajectory is not effected by the state deviation solutions, and continues to be used in the state propagation and computation of the measurements. [103]

2.5.4.1 Extended Kalman Filter

The Extended Kalman filter (EKF) follows the same procedures and math formulations as the conventional Kalman filter, except the reference trajectory state is updated with the state deviation solution at each time step. The updated reference trajectory state is used in the propagation step for the next observation, representing the best estimate of the reference trajectory. Therefore:

$$(X_k^*)^{EKF} = X_k^* + \hat{\mathbf{x}}_k \tag{2.47}$$

By updating the reference trajectory, the state deviation is reset to zero: $\hat{\mathbf{x}}_k = 0$, and the *a priori* state at the next observation time is zero: $\bar{\mathbf{x}}_{k+1} = 0$. [103]

2.5.5 State Noise Compensation (Process Noise)

As a large number of observations are processed, the elements of the covariance asymptotically approach zero. At each time step, more information is added, and more uncertainty is subtracted from the covariance (Equation 2.45). This results in the Kalman gain trending toward zero, and the state deviation update equaling the *a priori* state deviation (Equation 2.44). When this happens, the filter becomes insensitive to new information and observations, and filter saturation occurs.

Modeling errors can occur in the dynamical model or in the linearization of the nonlinear dynamics. These errors, combined with filter saturation, can lead to filter divergence as more observations are processed. The addition of process noise to the acceleration equations can compensate for these dynamical errors, and increases the uncertainty in the covariance at each time step. This results in a lower limit for the covariance to prevent filter saturation.

For the state defined as the position and velocity: $X = [\mathbf{r},\ \mathbf{v}]^T$, the dynamics of the system can be written as:

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) \tag{2.48}$$

where, $\mathbf{x}(t)$ is the state deviation. The function $A(t) = \left(\frac{\partial F(X,t)}{\partial X(t)}\right)^*$, the function $\mathbf{u}(t)$ is assumed to be white Gaussian process noise, and $B(t)$ maps the process noise into the state dynamics. The function $\mathbf{u}(t)$ has the properties:

$$E[\mathbf{u}(t)] = 0,\ E[\mathbf{u}(t_i)\mathbf{u}^T(t_j)] = Q(t_i)\delta_{ij} \tag{2.49}$$

The matrix, $Q(t_i)$ is the process noise covariance matrix, and δ_{ij} is the Dirac Delta function. The matrix $Q(t_i)$ is assumed to be a constant diagonal matrix.

The state deviation and covariance are propagated with the STM, and the process noise is propagated with the matrix, $\Gamma(t_{k+1}, t_k)$. The *a priori* state deviation and covariance at time t_{k+1} are then augmented to include process noise [103]:

$$\bar{\mathbf{x}}_{k+1} = \phi(t_{k+1}, t_k)\hat{\mathbf{x}}_k + \Gamma(t_{k+1}, t_k)\mathbf{u}_k \tag{2.50}$$

$$\bar{P}_{k+1} = \phi(t_{k+1}, t_k)P_k\phi^T(t_{k+1}, t_k) + \Gamma(t_{k+1}, t_k)Q_k\Gamma^T(t_{k+1}, t_k) \tag{2.51}$$

The matrix, $\Gamma(t_{k+1}, t_k)$ is defined as the process noise transition matrix:

$$\Gamma(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \tau)B(\tau)d\tau \tag{2.52}$$

From the definition of the state, $X = [\mathbf{r},\ \mathbf{v}]^T$, the equations of motion are the velocity, $\mathbf{v}$ and acceleration, $\mathbf{a}$: $\dot{X} = [\mathbf{v},\ \mathbf{a}]$. It is assumed that the process noise was only added to the acceleration;

therefore, from Equation 2.48, the process noise mapping matrix results in:

$$B(\tau) = \begin{bmatrix} 0_{3\times3} \\ I_{3\times3} \end{bmatrix} \tag{2.53}$$

The integral of the process noise transition matrix is rewritten as:

$$\Gamma(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \begin{bmatrix} \phi_2(\tau) \\ \phi_4(\tau) \end{bmatrix} d\tau \tag{2.54}$$

where,

$$\begin{bmatrix} \phi_2(\tau) \\ \phi_4(\tau) \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \mathbf{r}(t_{k+1})}{\partial \mathbf{v}(\tau)} \\ \dfrac{\partial \mathbf{v}(t_{k+1})}{\partial \mathbf{v}(\tau)} \end{bmatrix} \tag{2.55}$$

If the velocity is assumed to be linear between measurements, and the measurements are dense, the state and dynamics may be approximated as linear:

$$\begin{gathered} X(t_{k+1}) = \dot{X}(\tau)(t_{k+1} - \tau) \\ \frac{\partial X(t_{k+1})}{\partial \dot{X}(\tau)} = t_{k+1} - \tau \end{gathered} \tag{2.56}$$

It is further assumed that the time difference between measurements is small (from the dense measurement tracking assumption), and the velocity components negligibly effect each other. With these assumptions, the function inside the integral for $\Gamma(t_{k+1}, t_k)$ may be further approximated as:

$$\begin{bmatrix} \phi_2(\tau) \\ \phi_4(\tau) \end{bmatrix} = \left[\begin{array}{ccc} t_{k+1} - \tau & 0 & 0 \\ 0 & t_{k+1} - \tau & 0 \\ 0 & 0 & t_{k+1} - \tau \\ \hline & I_{3\times3} & \end{array}\right] \tag{2.57}$$

This leads to the approximation of the process noise transition matrix as:

$$\Gamma(t_{k+1}, t_k) = \begin{bmatrix} \dfrac{\Delta t^2}{2} & 0 & 0 \\ 0 & \dfrac{\Delta t^2}{2} & 0 \\ 0 & 0 & \dfrac{\Delta t^2}{2} \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} = \Delta t \begin{bmatrix} \dfrac{\Delta t}{2} I_{3\times3} \\ I_{3\times3} \end{bmatrix} \tag{2.58}$$

where $\Delta t = t_{k+1} - t_k$, and I is the identity matrix.

In the presented simulations, this formulation for process noise is added to the covariance, as in Equation 2.51. If a gap occurs in the measurements that is above a certain threshold, process noise is not added to the measurements.

2.5.6 Sequential Consider Kalman Filter

The Kalman filter assumes that the estimator is unbiased and the measurement errors are accurately modeled as Gaussian white noise. In practice, this is not always the case as unmodeled errors in the dynamical or measurement model can occur and cause biases in the state estimates. These unmodeled errors are treated as biases in the estimates, and may be neglected, directly estimated, or considered. At times, directly estimating these errors may be computationally expensive or the parameters may be poorly observable. Considering these parameters, either in an analysis or a filter, does not estimate the parameters directly but mathematically inflates the state covariance to include the uncertainty of the consider parameters.

The consider filter, also referred to as the Schmidt-Kalman filter, was first developed by Schmidt (1966)[98], and has seen further derivations by Jazwinski (1970)[57] and Tapley *et al.* (2004) [103]. Several formulations and comparisons of these formulations have been studied [115, 54, 113]. The following formulation follows the derivation in Tapley *et al.* (2004) [103].

The consider parameters are added to the estimation state:

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{c} \end{bmatrix} \tag{2.59}$$

where $\mathbf{x}$ is the estimated state deviation ($\mathbf{x} = X - X^*$), and $\mathbf{c}$ is the consider state deviation($\mathbf{c} = C - C^*$). The *a priori* estimates of $\mathbf{x}$ and $\mathbf{c}$ are given by:

$$\bar{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta} \tag{2.60}$$

$$\bar{\mathbf{c}} = \mathbf{c} + \boldsymbol{\beta} \tag{2.61}$$

where, $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$ have the properties:

$$E[\boldsymbol{\eta}] = 0,\ E[\boldsymbol{\eta}\boldsymbol{\eta}^T] = \bar{P}_x \tag{2.62}$$

$$E[\boldsymbol{\beta}] = 0,\ E[\boldsymbol{\beta}\boldsymbol{\beta}^T] = \bar{P}_{cc} \tag{2.63}$$

$$E[\boldsymbol{\eta}\boldsymbol{\beta}^T] = \bar{P}_{xc} = 0 \tag{2.64}$$

The last equation assumes there is no initial correlation between the state errors and the consider parameter errors. The measurement residual equation is given by:

$$\begin{aligned} \mathbf{y} =& H\mathbf{z} + \boldsymbol{\epsilon} \\ =& H_x\mathbf{x} + H_c\mathbf{c} + \boldsymbol{\epsilon} \end{aligned} \tag{2.65}$$

where $\boldsymbol{\epsilon} \sim N(0, R)$.

The measurement mapping matrices are defined for the estimated state, X and consider state, C:

$$\tilde{H}_{x_k} = \left.\frac{\partial G(X_k, t_k)}{\partial X}\right|_* \tag{2.66}$$

$$\tilde{H}_{c_k} = \left.\frac{\partial G(X_k, t_k)}{\partial C}\right|_* \tag{2.67}$$

The measurement mapping matrices are used to define the information matrices, Λ_{xx_k} and Λ_{xc_k} at time t_k:

$$\Lambda_{xx_k} = \bar{P}_{x_k} + \tilde{H}_{x_k}^T R_k^{-1} \tilde{H}_{x_k} \tag{2.68}$$

$$\Lambda_{xc_k} = \bar{P}_{xc_k} + \tilde{H}_{x_k}^T R_k^{-1} \tilde{H}_{c_k} \tag{2.69}$$

The sensitivity matrix is defined as: $S_{xc} = -P_x\Lambda_{xc}$. Substituting in the information matrices and assuming that $\bar{P}_{xc} = 0$, the sensitivity matrix at time t_k becomes:

$$S_{xc_k} = -P_{x_k} \tilde{H}_{x_k}^T R_k^{-1} \tilde{H}_{c_k} \tag{2.70}$$

The sensitivity matrix defines how a state update that has considered parameters varies with respect to those considered parameters. The sensitivity matrix also maps the consider parameter covariance into the state space.

2.5.6.1 Consider Covariance Analysis

In the consider covariance analysis, a consider state update and consider covariance are computed that include the effects of the sensitivity matrix. The state update with considered parameters is:

$$\hat{\mathbf{x}}_c = \hat{\mathbf{x}} + S_{xc}\bar{\mathbf{c}} \tag{2.71}$$

The consider covariance is defined as $P_c = E[(\hat{\mathbf{x}}_c - \hat{\mathbf{x}})(\hat{\mathbf{x}}_c - \hat{\mathbf{x}})^T]$ and may be written:

$$P_c = P_x + S_{xc}\bar{P}_{cc}S_{xc}^T \tag{2.72}$$

These are calculated with the covariance, P_x and the state update, $\hat{\mathbf{x}}$ that do not involve the effects of the consider parameters. The formulation of the consider covariance is now systematically inflated to include the effects of the consider parameters.

In the sequential consider filter, the time update for the state covariance, $\bar{P}$, the state update $\bar{x}$, and the Kalman gain, K_k remain the same:

$$\bar{\mathbf{x}}_k = \phi(t_k, t_{k-1})\hat{\mathbf{x}}_{k-1} \tag{2.73}$$

$$\bar{P}_k = \phi(t_k, t_{k-1})P_{k-1}\phi(t_k, t_{k-1})^T \tag{2.74}$$

The sensitivity matrix may be propagated in the sequential filter, given a previous sensitivity matrix, S_{k-1}. The time update for the sensitivity matrix is:

$$\bar{S}_k = \phi(t_k, t_{k-1})S_{k-1} + \theta(t_k, t_{k-1}) \tag{2.75}$$

where $\phi(t_k, t_{k-1})$ is the state transition matrix, and $\theta(t, t_k) = \dfrac{\partial X(t)}{\partial C(t_k)}$. The time update for the consider state deviation and the consider covariance are as follows:

$$\bar{\mathbf{x}}_{c_k} = \bar{\mathbf{x}}_k + \bar{S}_k\bar{\mathbf{c}} \tag{2.76}$$

$$\bar{P}_{c_k} = \bar{P}_{x_k} + \bar{S}_k\bar{P}_{cc}\bar{S}_k^T \tag{2.77}$$

The cross covariance of the deviations in the consider state and consider parameters is defined as $P_{xc} = E[(\hat{\mathbf{x}}_c - \mathbf{x})\boldsymbol{\beta}^T]$ and its time update is:

$$\bar{P}_{xc_k} = \bar{S}_k\bar{P}_{cc} \tag{2.78}$$

For the measurement update, the covariance update and state update remain the same:

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + K_k(\mathbf{y}_k - \tilde{H}_{x_k}\bar{\mathbf{x}}_k) \tag{2.79}$$

$$P_k = (I - K_k\tilde{H}_{x_k})\bar{P}_k \tag{2.80}$$

The sensitivity matrix is updated as:

$$S_k = (I - K_k\tilde{H}_{x_k})\bar{S}_k - K_k\tilde{H}_{c_k} \tag{2.81}$$

The consider state update and the consider covariance are updated with the sensitivity matrix:

$$\hat{\mathbf{x}}_{c_k} = \hat{\mathbf{x}}_k + S_k\bar{c} \tag{2.82}$$

$$P_{c_k} = P_k + S_k\bar{P}_{cc}S_k^T \tag{2.83}$$

as is the cross covariance of the deviations:

$$P_{xc_k} = S_k\bar{P}_{cc} \tag{2.84}$$

To initialize the filter, it is noted that the consider parameters do not effect the *a priori* value of the consider state at time t_0, hence: $\bar{\mathbf{x}}_{c_0} = \bar{\mathbf{x}}_0$. It follows from the time update of $\bar{\mathbf{x}}_{c_0}$ that $\bar{S}_0 = 0$, and the initial sensitivity matrix at time t_0 is:

$$S_0 = -K_0\tilde{H}_{c_0} \tag{2.85}$$

In the consider covariance analysis, the consider state and the consider covariance are calculated separately and recorded separately from the nominal Kalman filter state and covariance equations. The value of $\bar{\mathbf{c}}$ must be known and is passed into the estimation of the consider state. If $\bar{\mathbf{c}} = 0$, then the consider state update is the same as the state update.

2.5.6.2 Consider Covariance Filter

The consider covariance filter derives a consider state update that does explicitly include the value of $\bar{\mathbf{c}}$, yet captures its effects in the consider state update. In this way, $\bar{\mathbf{c}}$ does not need to be known *a priori*, and the state update includes the effects of the consider parameters.

In the consider covariance filter, a new Kalman gain is computed based on the consider covariance:

$$K_{c_k} = \bar{P}_{c_k}\tilde{H}_{x_k}^T\left(\tilde{H}_{x_k}\bar{P}_{c_k}\tilde{H}_{x_k}^T + R_k\right)^{-1} \tag{2.86}$$

This Kalman gain is used to compute the consider state update:

$$\hat{\mathbf{x}}_{c_k} = \bar{\mathbf{x}}_k + K_{c_k}(\mathbf{y}_k - \tilde{H}_{x_k}\hat{x}_k) \tag{2.87}$$

The remainder of the time update and measurement update equations remain the same as in the consider covariance analysis. The difference between the two algorithms is the computation of the additional Kalman gain, K_{c_k} and the computation of the consider state, $\hat{\mathbf{x}}_{c_k}$.

2.5.7 Unscented Kalman Filter

The unscented Kalman filter (UKF) differs from the previously discussed filters as it does not linearize around a reference trajectory, does not require the computation of Jacobian matrices, and the probability distribution is not restricted to a Gaussian distribution. For an estimation state of n components, the UKF derives $2n+1$ sigma points from the reference state, and propagates these points through the nonlinear model dynamics. A measurement is computed for each of these propagated sigma points and compared to the observed measurement. A weighted average of the differences between the observed and computed measurements provides a covariance of the estimated state. From this process, second order accuracy of the probability distribution is provided for all nonlinear systems, and third order accuracy is provided for Gaussian distributions. [59]

The sigma points that are propagated through the nonlinear dynamics represent the probability distribution around the mean. Various weighting schemes have been investigated for the sigma point distribution. [110, 60, 59] These weights effect the spread of the sigma points around the mean, as well as the weighted averages used to calculate the state update and covariance. The method presented here uses the following weights:

$$\begin{aligned} W_0^m &= 1 - \frac{n}{3\alpha^2} \\ W_0^c &= W_0^m + 1 - \alpha^2 + \beta \\ W_i &= \frac{1}{6\alpha^2} \end{aligned} \tag{2.88}$$

where α and β are tuning parameters. The value of α is defined $1\text{E}-4 \leq \alpha \leq 1$, and to minimize errors under a Gaussian distribution assumption, $\beta = 2$. [110]

Given an estimated state, $\hat{X}_{k-1}$ and covariance, P_{k-1} at t_{k-1}, the sigma points (χ) at t_{k-1} can be represented by Equation 2.89, where $\gamma = \alpha\sqrt{3}$.

$$\chi_{k-1} = \begin{bmatrix} \hat{X}_{k-1} & \hat{X}_{k-1} + \gamma\sqrt{P_{k-1}} & \hat{X}_{k-1} - \gamma\sqrt{P_{k-1}} \end{bmatrix}_{n\times(2n+1)} \tag{2.89}$$

These sigma points are then propagated through the nonlinear dynamics:

$$\chi_{k/k-1} = F(\chi_{k-1}, t) \tag{2.90}$$

A time update is computed from the propagated sigma points, the defined weights, and the process noise covariance matrix, Q_k:

$$\begin{aligned} \bar{X}_k &= \sum_{i=0}^{2n} W_i^m \chi_{i,k/k-1} \\ \bar{P}_k &= Q_k + \sum_{i=0}^{2n} W_i^c (\chi_{i,k/k-1} - \bar{X}_k)(\chi_{i,k/k-1} - \bar{X}_k)^T \end{aligned} \tag{2.91}$$

From this weighted average, new sigma points are computed:

$$\chi_k = \begin{bmatrix} \bar{X}_k & \bar{X}_k + \gamma\sqrt{\bar{P}_k} & \bar{X}_k - \gamma\sqrt{\bar{P}_k} \end{bmatrix}_{n\times(2n+1)} \tag{2.92}$$

A measurement is computed with each of the recomputed sigma points, χ_k:

$$G_k = G(\chi_k, t_k) \tag{2.93}$$

and combined for a weighted average of the measurements:

$$\bar{y}_k = \sum_{i=0}^{2n} W_i^m G_k \tag{2.94}$$

The innovation (P_{yy}) and cross-correlation (P_{xy}) covariances are computed with the defined weights, the computed measurements, the weighted average of the measurements, the measurement uncertainty matrix, R_k, and the sigma points:

$$P_{yy} = R_k + \sum_{i=0}^{2n} W_i^c (G_{i,k} - \bar{y}_k)(G_{i,k} - \bar{y}_k)^T \tag{2.95}$$

$$P_{xy} = \sum_{i=0}^{2n} W_i^c (\chi_{i,k} - \bar{X}_k)(G_{i,k} - \bar{y}_k)^T \tag{2.96}$$

The Kalman gain is determined from the innovation and cross-correlation matrices:

$$K_k = P_{xy} P_{yy}^{-1} \tag{2.97}$$

The state estimate and covariance matrix are computed with the Kalman gain:

$$\hat{X}_k = \bar{X}_k + K_k (Y_k - \bar{y}_k) \tag{2.98}$$

$$P_k = \bar{P}_k - K_k P_{yy} K_k^T \tag{2.99}$$

where Y_k is the observations.

The best estimate of the state is now considered $\hat{X}_k$. This best estimate is used to formulate the sigma points at the next time step, t_{k+1} that are propagated through the nonlinear dynamics. The difference between this method and the linearized filters is that the state is directly estimated. With the previous filters, the estimation state described a deviation from a reference trajectory.

2.5.8 Iterative Least-Squares Filter

One of the measurement processing techniques investigated is an iterative least-squares (LS) algorithm within an EKF. The iterative LS algorithm produces an estimate of the position and pointing of the spacecraft at one observation time with one flash lidar image. This direct estimate is passed into an EKF to propagate the state, estimate the velocity, and incorporate *a priori* information. This is similar to the current OpNav procedures: an OpNav team uses optical images to produce an estimated position and pointing of the spacecraft and passes that information to an

OD team for trajectory estimation. This algorithm parallels the iterative Kalman filter (IKF) [47], and a comparison is provided below.

To estimate the position and pointing of the spacecraft, a cost function is derived from the observational errors, minimized, and iterated until convergence. The sum of the square of the differences between the observed and computed range measurements of each pixel is minimized, and is expressed as the cost function, J in Equation 2.100.

$$J = \sum_{i=1}^{p} (\rho_{iobs} - \rho_{icomp})^2 \tag{2.100}$$

The ranges ρ_i are from the p number of overlapping pixels in the observed and computed flash lidar images. This can be rewritten in terms of an observed measurement, Y_i, and computed measurement, $G_i(X)$.

$$J = \sum_{i=1}^{p} (Y_i - G_i(X))^2 \tag{2.101}$$

The computed measurement, $G_i(X)$ is expanded about the nominal state, X^*, with a Taylor series expansion and higher order terms are ignored:

$$G_i(X) = G_i(X^*) + \tilde{H}_i \delta X \tag{2.102}$$

In Equation 2.102, $\tilde{H}_i = \left. \frac{\partial G(X)_i}{\partial X} \right|_*$ is evaluated along the nominal state, X^*, and $\delta X = (X - X^*)$.

The pre-fit residuals and observation errors are defined in Equations 2.103 and 2.104, respectively, and combing these equations with Equation 2.102, results in Equation 2.105.

$$y_i = Y_i - G_i(X^*) \tag{2.103}$$

$$\epsilon_i = Y_i - G_i(X) \tag{2.104}$$

$$y_i = \tilde{H}_i \delta X + \epsilon_i \tag{2.105}$$

Equation 2.105 is regularly seen in the filter estimation framework [103], and relates the observation deviation vector, $\mathbf{y}$ to the state deviation vector, δX. By substituting Equation 2.104 into Equation 2.101, and substituting that result into Equation 2.105, the cost function, J can be rewritten as:

$$J = (\mathbf{y} - \tilde{H} \delta X)^T (\mathbf{y} - \tilde{H} \delta X) \tag{2.106}$$

where $\mathbf{y}$ is a vector of all of the observation deviations, and $\tilde{H}$ is the measurement mapping matrix. Minimizing Equation 2.106 can be solved through the method of least squares by solving for the deviation, δX in Equation 2.107, with a covariance matrix, P in Equation 2.108. [103]

$$\delta X = (\tilde{H}^T \tilde{H})^{-1} \tilde{H}^T \mathbf{y} \tag{2.107}$$

$$P = (\tilde{H}^T \tilde{H})^{-1} \tag{2.108}$$

Within in the LS algorithm, a "least-squares state" (X_{LS}) is defined either as the position, $\mathbf{r}$, or the position and pointing, $\underline{\theta}$, of the spacecraft. Studying Equation 2.12, the position and pointing are the two states dependent on the returned ranges, ρ_i, and therefore the LS state consists of these states.

$$X_{LS} = \begin{bmatrix} \mathbf{r} \\ \\ \underline{\theta} \end{bmatrix} \tag{2.109}$$

The state update and covariance equations include the measurement uncertainty of the lidar beams, R_ρ, and are computed with Equations 2.110 and 2.111, where $\mathbf{y}$ is defined in Equation 2.103.

$$\delta X = (\tilde{H}_{LS}^T R_\rho^{-1} \tilde{H}_{LS})^{-1} \tilde{H}_{LS}^T R_\rho^{-1} \mathbf{y} \tag{2.110}$$

$$P_{LS} = (\tilde{H}_{LS}^T R_\rho^{-1} \tilde{H}_{LS})^{-1} \tag{2.111}$$

After an update, δX, is produced, it is added to the state, X_{LS}. A new computed image is created, and the algorithm is repeated until a convergence requirement was met.

2.5.8.1 Convergence Criteria

The least-squares algorithm is terminated based on the cost function, J, or a maximum number of iterations. The algorithm is considered converged if the criteria in Equations 2.112 and 2.113 are simultaneously met.

$$\sqrt{J/p} < 1.1 * \texttt{rms}(\texttt{diag}(R_\rho)) \tag{2.112}$$

$$|J_{t-1} - J_t| / J_t < 1\% \tag{2.113}$$

Equation 2.112 represents $\sqrt{J/p}$ dropping below a tolerance based on the uncertainty of the observed range values. The criteria, $\sqrt{J/p}$, scales the cost function such that it is the same order of magnitude as the RMS value of the range uncertainties. Once Equation 2.112 is satisfied, Equation 2.113 is checked until the value of J is within 1% of its previous value. At that point, the algorithm is considered to be stalled out and converged.

If J stalls out by the criteria in Equation 2.113, yet it is not below the tolerance set in Equation 2.112, it is assumed that J had reached a local minima, but not the global minima. To counter this, the current update of X_{LS} is altered before the next algorithm iteration in hopes that the optimization state would move out of the local minima. If this occurred, the state update δX is multiplied by -1 for that iteration, and added to the state, X_{LS}: $X_{LS_t} = X_{LS_{t-1}} - \delta X_t$. While this does not provide a solution for every case, it is a simple solution that opposes the direction the optimization state is headed toward the local minima. A maximum of 10 iterations is set if the algorithm does not converge.

2.5.8.2 Range Registration

To compare the observed and computed ranges of the flash lidar images, the corresponding pixels of each image array are paired together. The flash lidar image consists of a grid of points that describes the point cloud, and the range registration method between the point clouds simply matches the observed pixel grid with the computed pixel grid.

If an individual pixel does not have an observed range, or does not have a computed range, a comparison can not be made, and that pixel is not used. It is assumed that the pixels that do not have both an observed and computed range are on the perimeter of the image. The measurements may contain thousands of range points, and therefore excluding pairs on the boundaries of the image is assumed to be acceptable.

2.5.8.3 Direct Measurement Filter

The result from the LS algorithm is a position error or a position and pointing error estimate directly, and this is fed into an outer EKF to incorporate the *a priori* covariance, estimate the velocity, and propagate the state. The following equations outline the scenario of estimating the position and pointing state. If only the position is estimated, the pointing state was simply removed from the measurement matrices.

When estimating the position and pointing, the computed measurement, G_k at time t_k, is the current estimate of the state:

$$G_k(X^*) = \begin{bmatrix} \mathbf{r}^* \\ \underline{\theta}^* \end{bmatrix} \tag{2.114}$$

The observed measurement, Y_k, is computed with the observed flash lidar image and the LS algorithm. The resulting observed measurement is the current state estimate plus the deviation from the LS algorithm:

$$Y_k = \begin{bmatrix} \mathbf{r}^* + \delta\mathbf{r} \\ \underline{\theta}^* + \delta\underline{\theta} \end{bmatrix} \tag{2.115}$$

The observation deviation equation in the outer EKF is simply the deviation from the estimated state based on the LS algorithm:

$$\mathbf{y}_k = Y_k - G_k(X^*) = \begin{bmatrix} \delta\mathbf{r} \\ \delta\underline{\theta} \end{bmatrix} \tag{2.116}$$

From the direct measurement of the state, the measurement mapping matrix, $\tilde{H}_k$ is a combination of zeros and identity matrices, I. The measurement mapping matrix when only estimating the position is Equation 2.117, and the measurement mapping matrix when estimating position and pointing is Equation 2.118.

$$\tilde{H}_k = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{2.117}$$

$$\tilde{H}_k = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3}, \end{bmatrix} \tag{2.118}$$

The measurement uncertainty matrix, R_k is defined as the outputted covariance matrix from the least-squares algorithm, P_{LS} from Equation 2.111. The Kalman gain is computed with Equation 2.43 and the state update and covariance are computed with Equations 2.44 and 2.46, respectively. [103]

2.5.8.4 Comparison to the Iterative Kalman Filter

The iterative Kalman filter (IKF) is similar to the EKF in that it is a sequential filter, yet it iterates the state update with the current observation at each time step. [8, 47]

Provided the time-updated state, $\bar{\mathbf{X}}_k$ and covariance, $\bar{P}_k$ at a time t_k, the IKF is initialized as:

$$X_{T0} = \bar{\mathbf{X}}_k, \ P_{T0} = \bar{P}_k \tag{2.119}$$

where X_{T0} and P_{T0} are iterated at that time step. The state estimate is then iterated with the equation:

$$X_{T(i+1)} = \bar{\mathbf{X}}_k + K_{Ti}\left(Y_i - G(X_{Ti}) - \tilde{H}|_{X_{Ti}}(\bar{\mathbf{X}}_k - X_{Ti})\right) \tag{2.120}$$

and the covariance becomes:

$$P_{T(i+1)} = (I - K_{Ti}\tilde{H}|_{X_{Ti}})\bar{P}_k \tag{2.121}$$

The value of the Kalman gain is determined as:

$$K_{Ti} = \bar{P}_k\tilde{H}|_{X_{Ti}}^T(\tilde{H}|_{X_{Ti}}\bar{P}_k\tilde{H}|_{X_{Ti}}^T + R_k)^{-1} \tag{2.122}$$

where the expression $\tilde{H}|_{X_{Ti}}$ indicates that the derivative of the measurement is evaluated at the current state within the IKF iterations.

The main difference between the presented iterative LS algorithm and the IKF is the inclusion of the *a priori* covariance matrix, $\bar{P}_k$ in the IKF. The LS algorithm does not employ *a priori* information on the state to produce its state estimate. The IKF reduces to the EKF formulation if the state is not iterated. The state update for the EKF is the minimum variance estimator with *a priori* information (Equation 2.39), while the state update in the LS algorithm is the minimum variance estimator without *a priori* state information (Equation 2.37). The LS filter presented iterates the state update that is dependent on the measurement before applying *a priori* information.

2.5.8.5 Comparison to the Iterative Point Cloud Algorithm

The conventional ICP algorithm aligns two three-dimensional point clouds of a geometric object by solving for the translation and rotation of the measured point cloud to match that of the model point cloud. The typical formulation is presented in Besl and McKay [10], and is solved through the minimization of the mean square objective function that is reproduced in Equation 2.123.

$$f(\vec{q}) = \frac{1}{N_p} \sum_{i=1}^{N_p} ||\vec{x}_i - \mathbf{R}(\vec{q}_R)\vec{p}_i - \vec{q}_T||^2 \tag{2.123}$$

The vector $\vec{q}$ represents the registration state vector, $\vec{q} = [\vec{q}_R \mid \vec{q}_T]^T$, where $\vec{q}_R$ is a unit quaternion rotation vector, and $\vec{q}_T$ is the translation vector. The matrix $\mathbf{R}(\vec{q}_R)$ is the 3×3 rotation matrix created from the quaternion vector, and acts on the point $\vec{p}_i$. The point $\vec{x}_i$ is from the model point cloud and corresponds to the point $\vec{p}_i$ from the measured point cloud, and N_p is the total number of points to be aligned. Besl and Mckay[10] present a quaternion-based algorithm to directly solve for the rotation and translation vectors of $\vec{q}$, and the process is iterated until an error metric criteria is met.

A similar formulation is presented in Chen and Medioni [22]. The difference of the distances of the corresponding points is mapped to the normal vector of the surface at the model point, and is commonly known as the point-to-plane ICP method. Low[70] linearized the point-to-plane algorithm with small angle approximations and reduced the problem to a simple least-squares formulation of $\mathbf{Ax} - \mathbf{b}$, where $\mathbf{x}$ represented the rotation angles and translation vector.

The least-squares algorithm presented above similarly reduces the minimization of a cost function to a least-squares formulation. However, instead of solving for a rotation and translation of the point cloud vectors, as is done in these ICP algorithms, the cost function (Equation 2.100) is expressed in terms of the state of interest: the spacecraft position and pointing (Equation 2.109).

In addition, from the expression in Equation 2.12, the partial derivatives of the lidar beams with respect to the state are derived, and the first order nonlinear effects of the system are able to be included.

Typical ICP formulations require a registration algorithm to assign the corresponding points of the two point clouds. This includes such methods as closest point, normal shooting, and projection, and may involve additional computational methods such as a *k*-*d* tree.[93] The method presented uses the grid of pixels produced with the flash lidar image to pair corresponding pixels, removing the need for additional registration algorithms.

Furthermore, the orientation of the sensor grid in the sensor frame is assumed to be known, and thus the directions of the grid vectors are known in the sensor frame. In the cost function, the magnitudes of the observed and computed vectors (ρ_i) are simply differenced, and information on the direction of the beams is folded into the measurement partial derivatives. This reduces the number of measurements to process (one scalar value vs. a three-component vector for each pixel), and reduces computational time.

2.6 Realization of Flash Lidar Measurements in the Estimation Filters

This section details the specifics of processing the flash lidar measurements in the estimation filters. In each filtering technique, the range measurements aer compared by pairing the corresponding pixels in the observed and computed image. If a pixel did not return a range value in either the observed or computed image, this pixel is not used in the state estimation. This places a requirement on the measurement processing that an overlap must occur between the observed and computed measurements. This requirement places constraints on the magnitude of the initial state errors, effects processing measurements in the UKF, and led to the development of a pre-processing algorithm.

The partial derivatives with respect to one lidar beam are defined with respect to the spacecraft position and pointing. Since the flash lidar image is produced instantaneously at one observation time, only the position and pointing may be determined from this one image. The velocity state is determined over time as more observations are accumulated. Therefore, the partial derivatives presented are of one lidar beam with respect to the position, $\mathbf{r}$ (Section 2.6.2) and pointing error $\underline{\theta}$ (Section 2.6.3).

The full flash lidar image is processed by combining the individual measurement mapping matrices for each flash lidar pixel at one observation time. It is assumed that calibration of the flash lidar pixels has occurred, and each pixel of a flash lidar image is treated as an independent lidar beam. Therefore, the measurement mapping matrix, $\tilde{H}$, is defined as the accumulation of the measurement mapping matrices of the individual lidar beams, $\tilde{H}_i$. This is mathematically presented in Section 2.6.4.

With the partial derivatives defined, the information content related to the position and pointing error may be extracted. This is achieved through the Fisher Information Matrix, which is detailed in Section 2.6.5. The analytical analysis presented is used in Chapter 6 to support how subsets of pixels are selected in order to maximize the information content. In addition, this section compares the information content when using the EKF and LS filter, and shows the information gained is equivalent.

2.6.1 Flash Lidar Measurement Processing

In each of the estimation methods, the pixels of the flash lidar image are treated as individual measurements, and the corresponding pixels in the observed image and computed image are compared for the measurement residuals. If a particular pixel does not return a range in the observed image or in the computed image, a comparison can not be made, and a measurement residual is not computed for this pixel. It is assumed that if this occurred, these pixels are on the boundaries of the images, and there are enough pixels within the overlap of the observed and computed image to calculate the measurement residuals. Furthermore, this requires that the difference in the truth and estimated state must not be excessive so that an overlap can occur.

The observation deviation vector, $\mathbf{y}_k$, has a variable number of measurements used at each time step. In the filter, the corresponding pixels that returned an altimetry measurement in both the observed and computed image are differenced, and the differences populated the observation deviation vector. The number of measurements used per observation, p, is dynamic.

With the UKF, $2n+1$ flash lidar images are computed for each of the corresponding sigma points, and these images are compared to the observed image to provide a state estimate update. In order for a comparison of an individual pixel to be made, a overlap needed to occur between the observed image and each of the sigma point images. From the definition of the sigma points, this required that the state covariance was small enough such that the spread in the sigma points did not prevent a measurement overlap.

2.6.2 Relationship of Spacecraft Position to One Lidar Beam

The derivative of the range returned from a lidar beam, ρ with respect to the spacecraft position, $\mathbf{r}$ is derived from Equation 2.12. The resulting partial derivative can be expressed in terms of the pointing direction ($\hat{u}$) of the lidar beam and the surface variations where the lidar beam hit the asteroid ($\hat{n}$). The following derivation describes the relationship between the returned range of one lidar beam and the spacecraft position. The lidar beams in a flash lidar image are considered independent, and the combination of these independent beams comprise the full flash lidar image.

The first order Taylor series expansion of the range magnitude, ρ_i, with respect to the spacecraft state, X, is expressed in Equation 2.124, where ρ_{iobs} is the observation, ρ_i^* is the range evaluated along the reference trajectory, and ϵ_i is the error in the measurement, assumed to be Gaussian white noise, $E[\epsilon_i \epsilon_i^T] = \sigma_{\rho_i}^2$. This is the same formulation as in Equation 2.27.

$$\rho_{iobs} = \rho_i^* + \left.\frac{\partial \rho_i}{\partial X}\right|_* (X - X^*) + \epsilon_i \tag{2.124}$$

The state, X is defined as the position of the spacecraft in the body-fixed frame: $X = \mathbf{r}$.

From Equation 2.12, the range ρ is isolated by dotting $\hat{u}$ with both sides:

$$\rho_i = \hat{u}_i^T \mathbf{r}_A - \hat{u}_i^T \mathbf{r} \tag{2.125}$$

where $\mathbf{r}_A$ is vector from the center of the asteroid to the lidar intersection point on the surface. The vector $\mathbf{r}_A$ depends on the position of the spacecraft, the pointing direction, and the time.

$$\mathbf{r}_A = f(\mathbf{r},\ \hat{u}_i,\ t) \tag{2.126}$$

The derivative of the range with respect to the spacecraft position is expressed Equation 2.127, where I is the identity matrix.

$$\frac{\partial \rho_i}{\partial \mathbf{r}} = \hat{u}_i^T \frac{\partial \mathbf{r}_A}{\partial \mathbf{r}} - \hat{u}_i^T I \tag{2.127}$$

The expression in Equation 2.127 was multiplied by the deviation, $\Delta\mathbf{r} = (\mathbf{r} - \mathbf{r}^*)$, and evaluated along the nominal trajectory:

$$\left.\frac{\partial \rho}{\partial \mathbf{r}}\right|_* \Delta\mathbf{r} = \hat{u}^T \left.\frac{\partial \mathbf{r}_A}{\partial \mathbf{r}}\right|_* \Delta\mathbf{r} - \hat{u}^T \Delta\mathbf{r} \tag{2.128}$$

The Taylor series expansion of the surface vector, $\mathbf{r}_A$ is examined:

$$\mathbf{r}_A = \mathbf{r}_A^* + \left.\frac{\partial \mathbf{r}_A}{\partial \mathbf{r}}\right|_* (\mathbf{r} - \mathbf{r}^*) \tag{2.129}$$

and the expression $\Delta\mathbf{r}_A^*$ is defined as:

$$\Delta\mathbf{r}_A^* = \left.\frac{\partial \mathbf{r}_A}{\partial \mathbf{r}}\right|_* \Delta\mathbf{r} \tag{2.130}$$

Substituting the expression for $\Delta\mathbf{r}_A^*$ into Equation 2.128, it becomes:

$$\left.\frac{\partial \rho}{\partial \mathbf{r}}\right|_* \Delta\mathbf{r} = \hat{u}^T \Delta\mathbf{r}_A^* - \hat{u}^T \Delta\mathbf{r} \tag{2.131}$$

The expression in Equation 2.131 is the same as seen in Equation 2.124. Equation 2.124 is rewritten as:

$$\rho_{iobs} - \rho_i^* = \hat{u}_i^T \Delta\mathbf{r}_A^* - \hat{u}_i^T \Delta\mathbf{r} + \epsilon_i \tag{2.132}$$

From its definition in Equation 2.28, the measurement mapping matrix for one lidar beam, $\tilde{H}_i$ equals: $\tilde{H}_i = \hat{u}_i^T \Delta\mathbf{r}_A^* - \hat{u}_i^T \Delta\mathbf{r}$. The expression $\Delta\mathbf{r}$ is the deviation in the estimation state, so now one must define the relationship of the surface variations, $\Delta\mathbf{r}_A^*$ to the position deviation.

2.6.2.1 Relating the Spacecraft Position to the Surface Vector

To relate the perturbations in the surface variations and the spacecraft position, the geometry between $\mathbf{r}$ and $\mathbf{r}_A$ is evaluated in Figure 2.4. This figure illustrates the vector relationships between the spacecraft state, $\mathbf{r}$, the vector from the center of the asteroid to the surface, $\mathbf{r}_A$, the lidar beam vector, $\rho\hat{u}$, and their deviations, $\Delta\mathbf{r}_{sc}$ and $\Delta\mathbf{r}_A$. The states, $\mathbf{r}_{sc1}$ and $\mathbf{r}_{A1}$, are the resulting states from the deviations, and $x\hat{u}$ is the lidar beam vector at the new position, $\mathbf{r}_{sc1}$. This diagram assumes the the pointing direction has not changed, only the spacecraft position.

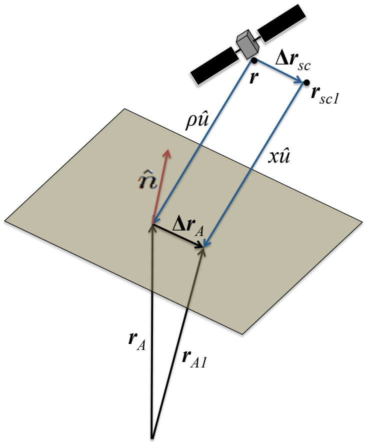


Figure 2.4: Vector diagram depicting a perturbation in the spacecraft position, $\Delta\mathbf{r}_{sc}$, and the resulting perturbation in the surface vector, $\Delta\mathbf{r}_A$ along the surface.

It is assumed that when the spacecraft is perturbed, the intersection point of the lidar beam at $\mathbf{r}_{A1}$ has the same surface normal, $\hat{n}$, as the original intersection point, $\mathbf{r}_A$. When considering a faceted shape model of the asteroid surface, this is equivalent to the lidar beam hitting the same facet as the nominal state after it is perturbed. If the surface is curved, this is equivalent to assuming the lidar beam hit the plane tangent to surface at the nominal intersection point.

The following vectors are defined from the diagram:

$$\Delta\mathbf{r}_A = \mathbf{r}_{A1} - \mathbf{r}_A \tag{2.133}$$

$$\Delta\mathbf{r}_{sc} = \mathbf{r}_{sc1} - \mathbf{r} \tag{2.134}$$

$$\mathbf{r} + \rho\hat{u} = \quad \mathbf{r}_A \tag{2.135}$$

$$\mathbf{r}_{sc1} + x\hat{u} = \quad \mathbf{r}_{A1} \tag{2.136}$$

The above vector equations are dotted with the normal vector to the surface, $\hat{n}$, and the ranges ρ and x are isolated.

$$\mathbf{r} \cdot \hat{n} + \rho\hat{u} \cdot \hat{n} = \qquad \mathbf{r}_A \cdot \hat{n} \tag{2.137}$$

$$\rho = \quad \frac{\mathbf{r}_A \cdot \hat{n} - \mathbf{r} \cdot \hat{n}}{\hat{u} \cdot \hat{n}} \tag{2.138}$$

$$\mathbf{r}_{sc1} \cdot \hat{n} + x\hat{u} \cdot \hat{n} = \qquad \mathbf{r}_{A1} \cdot \hat{n} \tag{2.139}$$

$$x = \quad \frac{\mathbf{r}_{A1} \cdot \hat{n} - \mathbf{r}_{sc1} \cdot \hat{n}}{\hat{u} \cdot \hat{n}} \tag{2.140}$$

It is assumed that as the spacecraft position is perturbed, $\Delta\mathbf{r}_{sc}$, the new intersection point to the surface, $\mathbf{r}_{A1}$ has the same surface normal vector as $\mathbf{r}_A$. The assumption that $\mathbf{r}_{A1}$ and $\mathbf{r}_A$ have the same surface normal vector, $\hat{n}$ leads to:

$$\Delta\mathbf{r}_A \cdot \hat{n} \quad = 0 \tag{2.141}$$

The expression for $\Delta\mathbf{r}_A$ is evaluated with the assumption that $\Delta\mathbf{r}_A \cdot \hat{n} = 0$:

$$\begin{aligned} \Delta\mathbf{r}_A &= \mathbf{r}_{A1} - \mathbf{r}_A \\ &= \mathbf{r}_{sc1} + x\hat{u} - (\mathbf{r} + \rho\hat{u}) \\ &= \mathbf{r}_{sc1} - \mathbf{r} + \frac{\mathbf{r}_{A1} \cdot \hat{n} - \mathbf{r}_{sc1} \cdot \hat{n}}{\hat{u} \cdot \hat{n}}\hat{u} - \frac{\mathbf{r}_A \cdot \hat{n} - \mathbf{r} \cdot \hat{n}}{\hat{u} \cdot \hat{n}}\hat{u} \\ &= \Delta\mathbf{r}_{sc} + \frac{(\mathbf{r}_{A1} - \mathbf{r}_A) \cdot \hat{n}}{\hat{u} \cdot \hat{n}}\hat{u} + \frac{(-\mathbf{r}_{sc1} + \mathbf{r}) \cdot \hat{n}}{\hat{u} \cdot \hat{n}}\hat{u} \end{aligned} \tag{2.142}$$

The resulting relationship is:

$$\Delta\mathbf{r}_A = \Delta\mathbf{r}_{sc} + \frac{(-\Delta\mathbf{r}_{sc}) \cdot \hat{n}}{\hat{u} \cdot \hat{n}}\hat{u} \tag{2.143}$$

2.6.2.2 Defining the Measurement Mapping Matrix

The expression in Equation 2.143 is substituted into Equation 2.132 (where $\Delta\mathbf{r}_{sc} = \Delta\mathbf{r}$) :

$$\begin{aligned} \rho_{iobs} - \rho_i^* &= \hat{u}_i^T \left(\Delta\mathbf{r} + \frac{(-\Delta\mathbf{r}) \cdot \hat{n}_i}{\hat{u}_i \cdot \hat{n}_i}\hat{u}_i \right) - \hat{u}_i^T \Delta\mathbf{r} + \epsilon_i \\ &= \hat{u}_i^T \Delta\mathbf{r} + \frac{(-\Delta\mathbf{r}) \cdot \hat{n}_i}{\hat{u}_i \cdot \hat{n}_i}\hat{u}_i^T \hat{u}_i - \hat{u}_i^T \Delta\mathbf{r} + \epsilon_i \end{aligned} \tag{2.144}$$

The result of the observation deviation vector is:

$$\rho_{iobs} - \rho_i^* = \frac{-\hat{n}_i^T}{\hat{u}_i \cdot \hat{n}_i}\Delta\mathbf{r} + \epsilon_i \tag{2.145}$$

Therefore, the measurement partial matrix for the i^{th} range observation at one time step with respect to the spacecraft position, $\tilde{H}_i$, can be expressed as Equation 2.146:

$$^{\mathbf{r}}\tilde{H}_i = \frac{\partial \rho_i}{\partial \mathbf{r}} = \frac{-\hat{n}_i^T}{\hat{u}_i \cdot \hat{n}_i} \tag{2.146}$$

This relationship provides a better understanding of how the position information is effected by the lidar beam pointing direction and the surface variations of its target. This is also consistent with a similar result found in Bayard et al. [7]

2.6.3 Relationship of Spacecraft Pointing to One Lidar Beam

The pointing state in the estimation state is defined as an off-nominal error with Euler angles: $\underline{\theta} = [\theta_1,\ \theta_2,\ \theta_3]$. The rotation is expressed in Equation 2.2, and reproduced here:

$$\hat{u} = C_2(\theta_2)C_1(\theta_1)C_3(\theta_3)\hat{u}^* \tag{2.2}$$

This rotation matrix created a rotation first about the boresight direction, and then about the x- then y-axes that represent the pixel/line directions of the sensor frame.

The measurement partial derivatives for a single range measurement, ρ_i, with respect to the Euler angle, θ_n, is described by Equation 2.147,

$$^{\theta_n}\tilde{H}_i = \frac{\partial \rho_i}{\partial \theta_n} = \frac{-\rho_{icomp}}{\hat{u}_i \cdot \hat{n}_i}(M_{\theta_n}\hat{u}_i^*)^T\hat{n}_i \tag{2.147}$$

where,

$$M_{\theta_1} = \begin{bmatrix} -\sin\theta_2\sin\theta_3\cos\theta_1 & \sin\theta_2\cos\theta_3\cos\theta_1 & \sin\theta_2\sin\theta_1 \\ \sin\theta_3\sin\theta_1 & -\cos\theta_3\sin\theta_1 & \cos\theta_1 \\ \cos\theta_2\sin\theta_3\cos\theta_1 & -\cos\theta_2\cos\theta_3\cos\theta_1 & -\cos\theta_2\sin\theta_1 \end{bmatrix} \tag{2.148}$$

$$M_{\theta_2} = \begin{bmatrix} -\cos\theta_3\sin\theta_2 - \sin\theta_1\sin\theta_3\cos\theta_2 & -\sin\theta_3\sin\theta_2 + \sin\theta_1\cos\theta_3\cos\theta_2 & -\cos\theta_1\cos\theta_2 \\ 0 & 0 & 0 \\ \cos\theta_3\cos\theta_2 - \sin\theta_1\sin\theta_3\sin\theta_2 & \sin\theta_3\cos\theta_2 + \sin\theta_1\cos\theta_3\sin\theta_2 & -\cos\theta_1\sin\theta_2 \end{bmatrix} \tag{2.149}$$

$$M_{\theta_3} = \begin{bmatrix} -\cos\theta_2\sin\theta_3 - \sin\theta_2\sin\theta_1\cos\theta_3 & \cos\theta_2\cos\theta_3 - \sin\theta_2\sin\theta_1\sin\theta_3 & 0 \\ -\cos\theta_1\cos\theta_3 & -\cos\theta_2\sin\theta_3 & 0 \\ -\sin\theta_2\sin\theta_3 + \cos\theta_2\sin\theta_1\cos\theta_3 & \sin\theta_2\cos\theta_3 + \cos\theta_2\sin\theta_1\sin\theta_3 & 0 \end{bmatrix} \tag{2.150}$$

In Equations 2.147, the pointing direction $\hat{u}_i = f(\hat{u}_i^*, \underline{\theta})$ from Equation 2.2, and $\hat{n}_i$ is the normal vector to the surface where the lidar beam intersects the surface. In Equation 2.147, ρ_{icomp} is the computed range, and M_{θ_n} is the derivative matrix for the corresponding angles (Equations 2.148, 2.149, and 2.150).

2.6.3.1 Derivation of Partial Derivatives for the Pointing State

The pointing direction, $\hat{u}$ is defined as a function of $\hat{u}^*$, such that $\hat{u}^*$ is the nominal pointing direction, and $\hat{u}$ is the corrected pointing direction based on θ_1, θ_2, and θ_3 (Equation 2.2). It follows that $\hat{u}$ can be expanded as such:

$$\hat{u} = \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1^* \\ u_2^* \\ u_3^* \end{bmatrix} \tag{2.151}$$

$$\hat{u} = \begin{bmatrix} (\cos\theta_2\cos\theta_3 - \sin\theta_2\sin\theta_1\sin\theta_3)u_1^* + (\cos\theta_2\sin\theta_3 + \sin\theta_2\sin\theta_1\cos\theta_3)u_2^* - \sin\theta_2\cos\theta_1 u_3^* \\ -\cos\theta_1\sin\theta_3 u_1^* + \cos\theta_1\cos\theta_3 u_2^* + \sin\theta_1 u_3^* \\ (\sin\theta_2\cos\theta_3 + \cos\theta_2\sin\theta_1\sin\theta_3)u_1^* + (\sin\theta_2\sin\theta_3 - \cos\theta_2\sin\theta_1\cos\theta_3)u_2^* + \cos\theta_2\cos\theta_1 u_3^* \end{bmatrix} \tag{2.152}$$

The returned range, ρ_i, is defined as that from Equation 2.13:

$$\rho_i = \frac{(V - \mathbf{r}) \cdot \hat{n}_i}{\hat{u}_i \cdot \hat{n}_i} \tag{2.153}$$

where V is a point on the plane that the pointing vector $\rho\hat{u}$ intersects, $\mathbf{r}$ is the position of the spacecraft, $\hat{n}_i$ is the normal to that plane, and $\hat{u}_i = f(\hat{u}_i^*, \theta_1, \theta_2, \theta_3)$.

The derivative of the range measurement, ρ_i is taken with respect to the angle, θ_n:

$$\begin{aligned}\frac{\partial \rho_i}{\partial \theta_n} =& \frac{\partial}{\partial \theta_n}\left(\frac{(V-\mathbf{r})\cdot\hat{n}_i}{\hat{u}_i\cdot\hat{n}_i}\right)\\ =& \left(\frac{-(V-\mathbf{r})\cdot\hat{n}_i}{(\hat{u}_i\cdot\hat{n}_i)^2}\right)\frac{\partial(\hat{u}_i\cdot\hat{n}_i)}{\partial\theta_n}\end{aligned} \tag{2.154}$$

This equation is applied to each angle, while only needing to define $\frac{\partial(\hat{u}_i\cdot\hat{n}_i)}{\partial\theta_n}$ for the specific Euler angle. The computed range is defined as:

$$\rho_{icomp} = \frac{(V-\mathbf{r})\cdot\hat{n}_i}{\hat{u}_i\cdot\hat{n}_i} \tag{2.155}$$

The partial derivative of the pointing is rewritten as:

$$\frac{\partial \rho_i}{\partial \theta_n} = \frac{-\rho_{icomp}}{\hat{u}_i\cdot\hat{n}_i}\frac{\partial(\hat{u}_i\cdot\hat{n}_i)}{\partial\theta_n} \tag{2.156}$$

The definitions of the partial derivatives $\frac{\partial(\hat{u}_i\cdot\hat{n}_i)}{\partial\theta_n}$ for each Euler angle are as follows. For θ_1, the partial derivative may be written in equation form:

$$\begin{aligned}\frac{\partial(\hat{u}\cdot\hat{n})}{\partial\theta_1} =& (-\sin\theta_2\sin\theta_3\cos\theta_1 u_1^* + \sin\theta_2\cos\theta_3\cos\theta_1 u_2^* + \sin\theta_2\sin\theta_1 u_3^*)n_1\\ &+ (\sin\theta_3\sin\theta_1 u_1^* - \cos\theta_3\sin\theta_1 u_2^* + \cos\theta_1 u_3^*)n_2\\ &+ (\cos\theta_2\sin\theta_3\cos\theta_1 u_1^* - \cos\theta_2\cos\theta_3\cos\theta_1 u_2^* - \cos\theta_2\sin\theta_1 u_3^*)n_3\end{aligned} \tag{2.157}$$

or in matrix form:

$$\frac{\partial(\hat{u}\cdot\hat{n})}{\partial\theta_1} = \left(\begin{bmatrix} -\sin\theta_2\sin\theta_3\cos\theta_1 & \sin\theta_2\cos\theta_3\cos\theta_1 & \sin\theta_2\sin\theta_1 \\ \sin\theta_3\sin\theta_1 & -\cos\theta_3\sin\theta_1 & \cos\theta_1 \\ \cos\theta_2\sin\theta_3\cos\theta_1 & -\cos\theta_2\cos\theta_3\cos\theta_1 & -\cos\theta_2\sin\theta_1 \end{bmatrix}\begin{bmatrix} u_1^* \\ u_2^* \\ u_3^* \end{bmatrix}\right)^T \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \tag{2.158}$$

Writing the partial derivative in matrix form allows the expression to be reduced:

$$\frac{\partial(\hat{u}\cdot\hat{n})}{\partial\theta_1} = (M_{\theta_1}\hat{u}_i^*)^T\hat{n}_i \tag{2.159}$$

where M_{θ_1} is defined in Equation 2.148. The partial derivatives for θ_2 and θ_3 in equation form are:

$$\begin{aligned}\frac{\partial(\hat{u}\cdot\hat{n})}{\partial\theta_2} =&((-\cos\theta_3\sin\theta_2 - \sin\theta_1\sin\theta_3\cos\theta_2)u_1^* \\ &+ (-\sin\theta_3\sin\theta_2 + \sin\theta_1\cos\theta_3\cos\theta_3)u_2^* - \cos\theta_1\cos\theta_2 u_3^*)n_1 \\ &+ ((\cos\theta_3\cos\theta_2 - \sin\theta_1\sin\theta_3\sin\theta_2)u_1^* \\ &+ (\sin\theta_3\cos\theta_2 + \sin\theta_1\cos\theta_3\sin\theta_2)u_2^* - \cos\theta_1\sin\theta_2 u_3^*)n_3\end{aligned} \tag{2.160}$$

$$\begin{aligned}\frac{\partial(\hat{u}\cdot\hat{n})}{\partial\theta_3} =&((-\cos\theta_2\sin\theta_3 - \sin\theta_2\sin\theta_1\cos\theta_3)u_1^* + (\cos\theta_2\cos\theta_3 - \sin\theta_2\sin\theta_1\sin\theta_3)u_2^*)n_1 \\ &+ (-\cos\theta_1\cos\theta_3 u_1^* - \cos\theta_1\sin\theta_3 u_2^*)n_2 \\ &+ ((-\sin\theta_2\sin\theta_3 + \cos\theta_2\sin\theta_1\cos\theta_3)u_1^* + (\sin\theta_2\cos\theta_3 + \cos\theta_2\sin\theta_1\sin\theta_3)u_2^*)n_3\end{aligned} \tag{2.161}$$

It follows that $\frac{\partial(\hat{u}\cdot\hat{n})}{\partial\theta_n}$ can take the general form:

$$\frac{\partial(\hat{u}\cdot\hat{n})}{\partial\theta_n} = (M_{\theta_n}\hat{u}_i^*)^T\hat{n}_i \tag{2.162}$$

where the appropriate M_{θ_n} matrices are defined in Equations 2.148, 2.149, and 2.150.

2.6.4 Processing the Full Flash Lidar Image

The measurement mapping matrix for that with respect to position is defined as ${}^{\mathbf{r}}\tilde{H}_i$, and that with respect to the pointing state is defined as ${}^{\underline{\theta}}\tilde{H}_i$. The measurement mapping matrix for one lidar beam with respect to the full state of position, velocity, and pointing, is defined as a combination of these matrices:

$$\tilde{H}_i = \begin{bmatrix} {}^{\mathbf{r}}\tilde{H}_i & 0_{1\times3} & {}^{\underline{\theta}}\tilde{H}_i \end{bmatrix}_{1\times9} \tag{2.163}$$

Information on the velocity state is not acquired from a single flash lidar image, but is accumulated over time.

To process the full flash lidar image, the lidar beams are treated as independent measurements. Therefore, the measurement mapping matrix for the full flash lidar image at time t_k with

p usable measurements is the combination of the individual $\tilde{H}_i$ matrices:

$$\tilde{H}_k = \begin{bmatrix} \tilde{H}_1 \\ \vdots \\ \tilde{H}_i \\ \vdots \\ \tilde{H}_p \end{bmatrix}_{p\times 9} \tag{2.164}$$

In the LS filter, the position and pointing states populate the LS state (Equation 2.109). The measurement mapping matrix for the LS algorithm is defined as:

$$\tilde{H}_{LSi} = \begin{bmatrix} {}^{\mathbf{r}}\tilde{H}_i & {}^{\underline{\theta}}\tilde{H}_i \end{bmatrix}_{1\times 6} \tag{2.165}$$

The measurement matrix for the full flash lidar image, $\tilde{H}_{LS}$ is formed by combining the individual $\tilde{H}_{LSi}$ matrices, forming a $p \times 6$ matrix:

$$\tilde{H}_{LS} = \begin{bmatrix} \tilde{H}_{LS1} \\ \vdots \\ \tilde{H}_{LSi} \\ \vdots \\ \tilde{H}_{LSp} \end{bmatrix}_{p\times 6} \tag{2.166}$$

2.6.5 Information Content

The information content contained in a lidar measurement can be determined through the Fisher Information Matrix (FIM). Regardless of the estimation filter used, analyzing this information can provide insight into how to best direct the lidar beams in any 3D lidar system.

The Cramér-Rao Lower Bound, P^* is the best achievable covariance for an unbiased estimator [104]. This limit is defined in Equation 2.167, where P is the covariance of an unbiased estimator of unknown deterministic variables, and Λ is defined as the FIM.

$$P \geq P^* = \Lambda^{-1} \tag{2.167}$$

A recursive formulation of the FIM at time t_k with acquired measurements over time is defined in Equation 2.168, where ϕ_{k-1} is the state transition matrix, H_k is the measurement partial matrix mapped to one time ($H_k = \tilde{H}_{k-1}\phi_{k-1}$), and R_k is the measurement uncertainty matrix.

$$\Lambda_k = (\phi_{k-1}P^*_{k-1}\phi^T_{k-1})^{-1} + H_k^T R_k^{-1} H_k \tag{2.168}$$

The FIM for the spacecraft position with an *a priori* position covariance matrix, P_0, for one range measurement at one time step can be expressed with $\tilde{H}_i$ and the uncertainty of the measurement, R_i.

$$\Lambda_i = P_0^{-1} + \tilde{H}_i^T R_i^{-1} \tilde{H}_i \tag{2.169}$$

2.6.5.1 Measurement Uncertainty

The measurement error of the lidar beams, $\boldsymbol{\epsilon}$ is considered to be from a Gaussian distribution of zero mean with covariance R_ρ:

$$E[\boldsymbol{\epsilon}] = 0,\ E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = R \tag{2.170}$$

The uncertainty of a single lidar beam is defined as σ_{ρ_i}. Since each lidar beam is considered an independent measurement, the matrix R is defined as a diagonal matrix:

$$R_\rho = \texttt{diag}[\sigma^2_{\rho 1}, \cdots, \sigma^2_{\rho i}, \cdots, \sigma^2_{\rho p}] \tag{2.171}$$

2.6.5.2 Information with Respect to Position

For a single lidar beam, the inverse of the uncertainty matrix, R_i^{-1}, is simply $1/\sigma^2_{\rho i}$. With this, and the definition of the measurement mapping matrix with respect to position, ${}^{\mathbf{r}}\tilde{H}_i$ in Equation 2.146, the FIM for one lidar beam becomes:

$$\Lambda_i = P_0^{-1} + \frac{1}{\sigma^2_{\rho i}} \frac{\hat{n}_i \hat{n}_i^T}{(\hat{u}_i \cdot \hat{n}_i)^2} \tag{2.172}$$

For p independent lidar beams at one time step, the FIM becomes a sum of the final term above, and is expressed in Equation 2.173. The variable i represents an individual lidar beam, and therefore an individual $\hat{u}_i$ and $\hat{n}_i$ are calculated for each pixel within the measurement image.

$$\Lambda_p = P_0^{-1} + \sum_{i=1}^{p} \frac{1}{\sigma_{\rho i}^2} \frac{\hat{n}_i \hat{n}_i^T}{(\hat{u}_i \cdot \hat{n}_i)^2} \tag{2.173}$$

To quantify the total amount of information content contained in p lidar beams, the determinant of the FIM is examined. This is known as the D-optimality condition, and is seen in other work [75, 42]. For one or two lidar beams, the determinant of the FIM becomes zero and no information on the state is available without *a priori* covariance knowledge, P_0. For $p \geq 3$, the determinant of the FIM is nonzero with and without an *a priori* covariance. The determinant of the FIM for $p = 3$ without an *a priori* covariance is expressed in Equation 2.174.

$$|\Lambda_3| = \frac{1}{\sigma_{\rho 1}^2 \sigma_{\rho 2}^2 \sigma_{\rho 3}^2} \left(\frac{\hat{n}_1}{\hat{u}_1 \cdot \hat{n}_1} \cdot \left(\frac{\hat{n}_3}{\hat{u}_3 \cdot \hat{n}_3} \times \frac{\hat{n}_2}{\hat{u}_2 \cdot \hat{n}_2} \right) \right)^2 \tag{2.174}$$

In order to maximize the total measurement information for three beams, the corresponding surface normals should all be perpendicular to each other: $\hat{n}_1 \perp \hat{n}_2 \perp \hat{n}_3$. In addition, the pointing directions, $\hat{u}_i$ should be almost perpendicular with their corresponding $\hat{n}_i$ vectors to reduce the denominators in the expression.

Analyzing Equation 2.174, one must direct three beams at three non-parallel target planes order for the determinant of the information matrix to be nonzero. Mathematically, if $\hat{n}_2$ is parallel to $\hat{n}_3$, the cross product will be zero and therefore $|\Lambda_3| = 0$. Moreover, if $\hat{n}_1$ is parallel to $\hat{n}_3$, then $\hat{n}_1$ will be perpendicular to the cross product of $(\hat{n}_3 \times \hat{n}_2)$; therefore, the dot product of $\hat{n}_1 \cdot (\hat{n}_3 \times \hat{n}_2) = 0$, and $|\Lambda_3| = 0$. Equivalently, if all three beams are directed at one flat wall, or at only two flat walls, the full position state cannot be determined.

2.6.5.3 Information with Respect to Position and Pointing

Following a similar procedure, the FIM is determined for the position and pointing state for one lidar beam at one time step:

$$\Lambda_i = \bar{P}_0^{-1} + \begin{bmatrix} {}^{\mathrm{r}}\tilde{H}_i \\ {}^{\underline{\theta}}\tilde{H}_i \end{bmatrix} \frac{1}{\sigma_{\rho i}^2} \begin{bmatrix} {}^{\mathrm{r}}\tilde{H}_i & {}^{\underline{\theta}}\tilde{H}_i \end{bmatrix} \tag{2.175}$$

Substituting in the expressions for ${}^{\mathbf{r}}\tilde{H}_i$ (Equation 2.146) and ${}^{\underline{\theta}}\tilde{H}_i$ (Equation 2.147), the FIM becomes:

$$\Lambda_i = \bar{P}_0^{-1} + \frac{1}{\sigma_{\rho i}^2}\begin{bmatrix} \dfrac{\hat{n}_i\hat{n}_i^T}{(\hat{u}_i\cdot\hat{n}_i)^2} & \dfrac{\rho_{ic}}{(\hat{u}_i\cdot\hat{n}_i)^2}\begin{bmatrix} M_{\theta_1}\hat{u}_i^* & M_{\theta_2}\hat{u}_i^* & M_{\theta_3}\hat{u}_i^* \end{bmatrix} \\ \dfrac{\rho_{ic}}{(\hat{u}_i\cdot\hat{n}_i)^2}\begin{bmatrix} M_{\theta_1}\hat{u}_i^* \\ M_{\theta_2}\hat{u}_i^* \\ M_{\theta_3}\hat{u}_i^* \end{bmatrix}\hat{n}_i\hat{n}_i^T & \dfrac{\rho_{ic}}{(\hat{u}_i\cdot\hat{n}_i)^2}\begin{bmatrix} M_{\theta_1}\hat{u}_i^* \\ M_{\theta_2}\hat{u}_i^* \\ M_{\theta_3}\hat{u}_i^* \end{bmatrix}\hat{n}_i\hat{n}_i^T\begin{bmatrix} M_{\theta_1}\hat{u}_i^* & M_{\theta_2}\hat{u}_i^* & M_{\theta_3}\hat{u}_i^* \end{bmatrix} \end{bmatrix} \tag{2.176}$$

Combining the $\tilde{H}_i$ matrices for all of the lidar beams in on flash lidar image, the FIM becomes:

$$\Lambda_p = \bar{P}_0^{-1} + \begin{bmatrix} \displaystyle\sum_{i=1}^{p}\frac{1}{\sigma_{\rho i}^2}\frac{\hat{n}_i\hat{n}_i^T}{(\hat{u}_i\cdot\hat{n}_i)^2} & \displaystyle\sum_{i=1}^{p}\frac{1}{\sigma_{\rho i}^2}\frac{\rho_{ic}}{(\hat{u}_i\cdot\hat{n}_i)^2}\begin{bmatrix} M_{\theta_1}\hat{u}_i^* & M_{\theta_2}\hat{u}_i^* & M_{\theta_3}\hat{u}_i^* \end{bmatrix} \\ \displaystyle\sum_{i=1}^{p}\frac{1}{\sigma_{\rho i}^2}\frac{\rho_{ic}}{(\hat{u}_i\cdot\hat{n}_i)^2}\begin{bmatrix} M_{\theta_1}\hat{u}_i^* \\ M_{\theta_2}\hat{u}_i^* \\ M_{\theta_3}\hat{u}_i^* \end{bmatrix}\hat{n}_i\hat{n}_i^T & \displaystyle\sum_{i=1}^{p}\frac{1}{\sigma_{\rho i}^2}\frac{\rho_{ic}}{(\hat{u}_i\cdot\hat{n}_i)^2}\begin{bmatrix} M_{\theta_1}\hat{u}_i^* \\ M_{\theta_2}\hat{u}_i^* \\ M_{\theta_3}\hat{u}_i^* \end{bmatrix}\hat{n}_i\hat{n}_i^T\begin{bmatrix} M_{\theta_1}\hat{u}_i^* & M_{\theta_2}\hat{u}_i^* & M_{\theta_3}\hat{u}_i^* \end{bmatrix} \end{bmatrix} \tag{2.177}$$

One can notice the term $\dfrac{\hat{n}_i\hat{n}_i^T}{(\hat{u}_i\cdot\hat{n}_i)^2}$, from the position information is common in each of the matrix terms. This implies that the same criteria to maximize the FIM with respect to position applies to the FIM involving position and pointing as well.

2.6.5.4 Information Content of Estimation Methods

The information gained at one observation time for the different filtering methods may also be determined through the FIM. The FIM at time, t_k is defined as:

$$\Lambda_k = \bar{P}_0^{-1} + \tilde{H}_k^T R_k^{-1}\tilde{H}_k \tag{2.178}$$

In the EKF, the observation deviation vector, $\mathbf{y}_k$, directly uses the differences between the observed (Y_k) and computed (G_k) ranges:

$$\mathbf{y}_k = Y_k - G_k = \begin{bmatrix} \rho_{1obs} - \rho_{1comp} \\ \vdots \\ \rho_{iobs} - \rho_{icomp} \\ \vdots \\ \rho_{pobs} - \rho_{pcomp} \end{bmatrix}_{p\times 1} \tag{2.179}$$

The measurement mapping matrix for one pixel with the EKF is presented in Equation 2.180, and Equation 2.181 represents the full flash lidar image with p range-returns at one measurement time.

$$\tilde{H}_i = \begin{bmatrix} {}^{\mathbf{r}}\tilde{H}_i & 0_{1\times 3} & {}^{\underline{\theta}}\tilde{H}_i \end{bmatrix}_{1\times 9} \tag{2.180}$$

$$\tilde{H}_{\mathbf{p}} = \begin{bmatrix} {}^{\mathbf{r}}\tilde{H}_{\mathbf{p}} & 0_{\mathbf{p}} & {}^{\underline{\theta}}\tilde{H}_{\mathbf{p}} \end{bmatrix}_{p\times 9} \tag{2.181}$$

The measurement uncertainty for the lidar beams is defined as R_ρ from Equation 2.171, and $\tilde{H}_{\mathbf{p}}$ and R_ρ are passed into Equation 2.178 to calculate the information. The FIM for the EKF at one time step results in:

$$\Lambda_k = \bar{P}_0^{-1} + \begin{bmatrix} {}^{\mathbf{r}}\tilde{H}_{\mathbf{p}}^T(R_\rho^{-1}){}^{\mathbf{r}}\tilde{H}_{\mathbf{p}} & 0_{3\times 3} & {}^{\mathbf{r}}\tilde{H}_{\mathbf{p}}^T(R_\rho^{-1}){}^{\underline{\theta}}\tilde{H}_{\mathbf{p}} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ {}^{\underline{\theta}}\tilde{H}_{\mathbf{p}}^T(R_\rho^{-1}){}^{\mathbf{r}}\tilde{H}_{\mathbf{p}} & 0_{3\times 3} & {}^{\underline{\theta}}\tilde{H}_{\mathbf{p}}^T(R_\rho^{-1}){}^{\underline{\theta}}\tilde{H}_{\mathbf{p}} \end{bmatrix} \tag{2.182}$$

In the LS filter, the estimate of the position and pointing are treated as the measurements in the observation deviation vector. The measurement mapping matrix, $\tilde{H}_k$ is defined as a combination of zeros and identity matrices from Equation 2.118, and the measurement uncertainty matrix is defined as the covariance from the LS algorithm: $R_k = P_{LS}$. Passing these definitions into the information matrix, the information for the LS method is defined as:

$$\Lambda_k = \bar{P}_0^{-1} + \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & I_{3\times 3}, \end{bmatrix}^T P_{LS}^{-1} \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & I_{3\times 3}, \end{bmatrix} \tag{2.183}$$

The LS covariance matrix, P_{LS} (Equation 2.111), can be rewritten from the definition of $\tilde{H}_{LS}$ in Equation 2.166:

$$P_{LS}^{-1} = \begin{bmatrix} {}^{\mathbf{r}}\tilde{H}_{\mathbf{p}}^{T} \\ {}^{\theta}\tilde{H}_{\mathbf{p}}^{T} \end{bmatrix} R_{\rho}^{-1} \begin{bmatrix} {}^{\mathbf{r}}\tilde{H}_{\mathbf{p}} & {}^{\theta}\tilde{H}_{\mathbf{p}} \end{bmatrix} \tag{2.184}$$

Substituting Equation 2.184 into Equation 2.183 results in the same expression for the FIM as in Equation 2.182.

Therefore, the information content is the same for one flash lidar image for both the EKF and LS filter. The difference between the methods is in the iterative procedure the LS filter employs at each time step before passing the estimated state to the propagation. The LS filter uses all of the available information from the observation and produces a state error that aligns with the filter covariance.

Chapter 3

Orbit Determination Simulations

Once a spacecraft arrives at a small body, and the characterization phase is complete, the spacecraft typically inserts into a stable terminator orbit for the scientific observations. Orbit determination (OD) of the spacecraft must be consistent to ensure the accuracy of the spacecraft state for science applications as well as spacecraft safety. The frequency that the navigation observations are captured may be reduced to every few hours for OD operations and to accommodate other scientific instruments onboard.[55]

This chapter compares the performance of three navigation filters, the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the iterated least-squares (LS) filter in a variety of orbits around Itokawa and Bennu with the flash lidar images. An optical navigation (OpNav) simulation is investigated around Itokawa for comparison with the flash lidar measurements. The measurements for the OpNav simulations are defined as landmarks tied to the surface of Itokawa, and the measurement residuals are the difference in the landmark locations in the image plane.

Terminator orbits around Itokawa and Bennu encompass the majority of the simulations since this is a common small body spacecraft orbit. Other orbits include a descent orbit with an initial impulsive maneuver that travels to the surface of Itokawa, and an unstable eccentric orbit around Itokawa. The effect of the observation frequency is also investigated, comparing the performance of capturing measurements every 2 hours or every 10 minutes. The addition of process noise is investigated as well with the UKF and LS filters in the Itokawa and Bennu terminator orbits.

These simulations assume the shape model is known, and the spin state and axis of the asteroid are known. Gaussian white noise is applied to the individual range measurements in the flash lidar image, and the uncertainty is based on the observed range. The first set of simulations estimate the position and velocity, and assume that the spacecraft pointing is known. The second set of simulations apply an off-nominal pointing bias to the sensor pointing, and estimate this error along with the position and velocity.

Section 3.1 overviews the orbits, and Section 3.2 presents the simulation setup. Section 3.3 presents simulations that estimate the position and velocity in the Itokawa terminator orbit with an operations measurement frequency (with and without process noise) and a high measurement frequency (with process noise). It also presents results from the Bennu terminator orbit with an operations measurement frequency with and without process noise. Section 3.4 compares the filter performance of the flash lidar images with an optical navigation simulation. Section 3.5 presents a navigation simulation in an descent orbit around Itokawa while varying the initial state errors and observation frequencies. Section 3.6 estimates the position, velocity, and pointing bias of the spacecraft for the Itokawa terminator orbit with an operations measurement frequency (with and without process noise), and a high measurement frequency (with process noise). It also includes OD for the Bennu terminator orbit for an operations measurement frequency with and without process noise, and an unstable eccentric orbit around Itokawa with a high measurement frequency. Section 3.7 summarizes the simulation results of the terminator orbits around Itokawa and Bennu, and the Itokawa eccentric orbit.

3.1 Orbits

A circular orbit in the sun-terminator plane (a terminator orbit) is stable around around a small body due to the forces of solar radiation pressure (SRP). Small body mission profiles include this orbit due to its stable properties. One-kilometer circular terminator orbits around the asteroid Itokawa and around the asteroid Bennu are the focus for the majority of the OD simulations presented.

In addition, a descent orbit to the surface of Itokawa is investigated. This orbit is initiated in a circular terminator orbit, an impulsive maneuver is applied, and the orbit descends to the surface across the sunlit side of Itokawa. An eccentric orbit around Itokawa is also investigated. This eccentric orbit is unstable, and escapes after propagation for 30 days. The orbit also travels beyond the limit of the flash lidar instrument (> 1 km), and a gap in the observations occurrs.

This section outlines the different orbits used in the OD simulations, and the numerical propagation used.

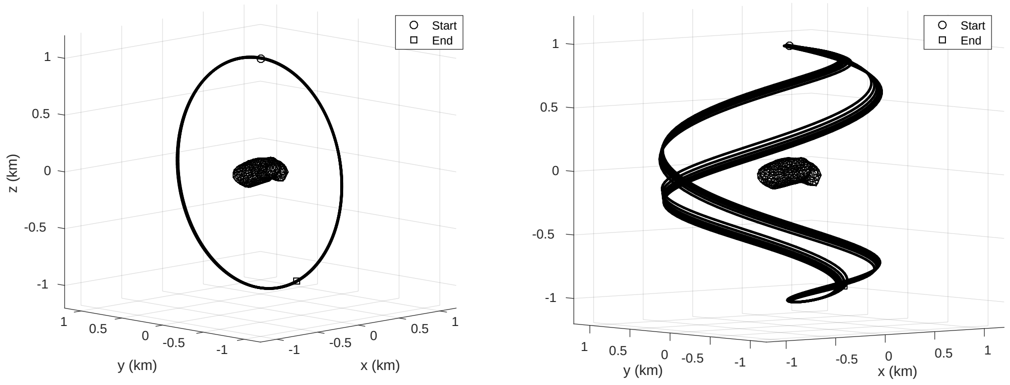
3.1.1 Dynamical Propagation

The spacecraft is propagated with the dynamics outlined in Section 2.2. The dynamical forces include gravity modeling with spherical harmonics up to degree and order four for both the Itokawa [94] and Bennu [24] orbits, solar radiation pressure (SRP), and third body effects from the sun. A cannonball model is used for the SRP modeling with an area-to-mass ratio of 0.01 m^2/kg, and a coefficient of reflectivity, $C_R = 1.2$. The sun is held constant with respect to Itokawa when modeling the third-body perturbations due to the short nature of the simulations.

The orbits are numerically propagated with an RK78 integrator. This is a variable step Runge-Kutta integrator that adjusts the step size by comparing the results of the 7th and 8th order integrators. An initial guess for the step size is 1E-7 s, and the relative tolerance between the 7th and 8th order integrators is set to 1E-14.

3.1.2 Itokawa Terminator Orbit

The circular terminator orbit had a semi-major axis of one kilometer and inclination of 90°. Figure 3.1 illustrates this orbit in the inertial (Figure 3.1(a)) and body-fixed (Figure 3.1(b)) frames for 10 days. The terminator orbit is stable due to the effects of SRP, and is a typical orbit a spacecraft would enter for operations around a small body. As shown in Figure 3.1, the orbit repeats itself for several revolutions.



(a) Inertial frame orbit

(b) Body-Fixed frame orbit

Figure 3.1: Nominal terminator orbit about Itokawa propagated for 10 days

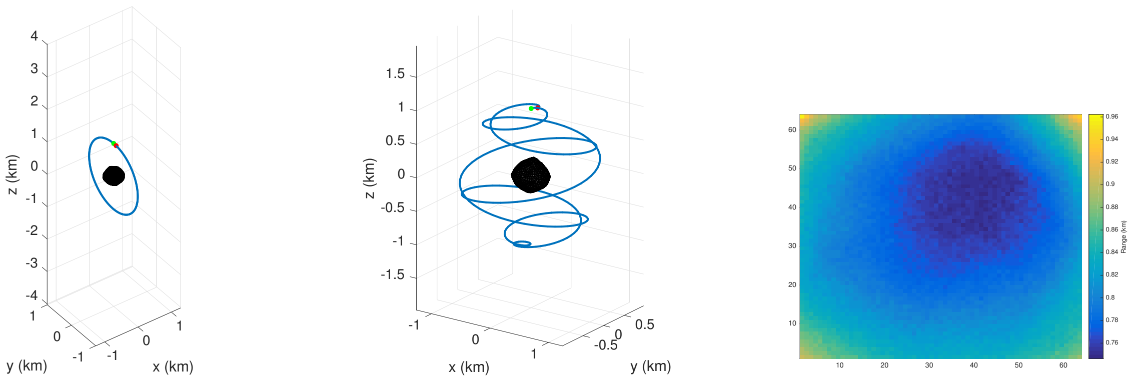
3.1.3 Bennu Terminator Orbit

The circular terminator orbit around Bennu had a semi-major axis of 1 km, and an inclination of 90°. Figure 3.2(a) shows inertial frame view and Figure 3.2(b) shows the body-fixed view of this orbit around Bennu. The orbit shown is propagated for one day or approximately one orbital period.

Figure 3.2(c) shows the observed flash lidar measurement at 100 minutes into the nominal orbit. The images for Bennu in Figure 3.2(c) and for Itokawa in Figure 2.3(a) were both taken with the same sensor setup, in a 1 km circular orbit. The main differences between the measurements is that the surface encompasses the entirety of the image when imaging Bennu, producing more range measurements per observation time.

3.1.4 Itokawa Eccentric Orbit

The eccentric orbit has a semi-major axis of 800 m, inclination of 60°, and eccentricity of 0.5. Figure 3.3 shows this orbit for 30 hours in the inertial (Figure 3.3(a)) and body-fixed (Figure 3.3(b)) frames. The orbit stays in the terminator plane and does not travel behind the asteroid,



(a) Inertial Frame (b) Body-Fixed Frame (c) Flash LIDAR measurement

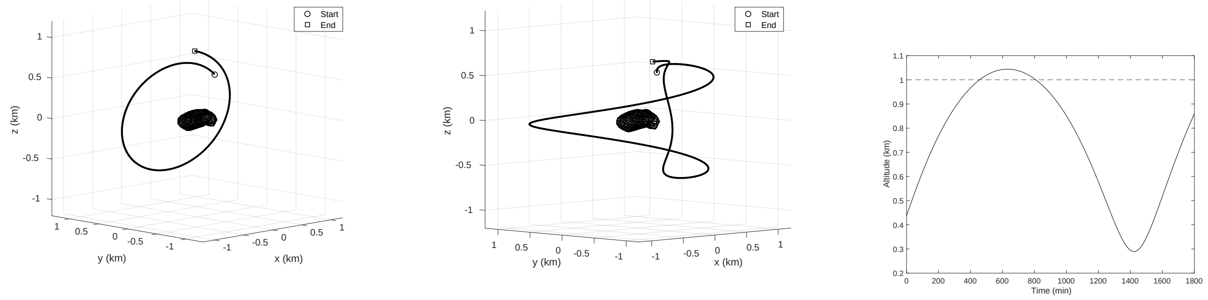
Figure 3.2: Bennu nominal circular terminator orbit for one day in the asteroid-centered inertial (ACI) frame (a), and the body-fixed frame (BF) (b), and a sample flash LIDAR measurement (c).

and shadowing does not need to be taken into account in the propagation. As shown in Figure 3.3(a), the orbit does not repeat due to the small body gravity perturbations, and the spacecraft is accelerated as it traveled closer to the body, as shown in Figure 3.3(b). This orbit is not stable, and the spacecraft escapes when it is propagated for 30 days.

Figure 3.3(c) represents the altitude of the spacecraft in the eccentric orbit with respect to reference radius of 162 m of Itokawa. The lowest altitude occurrs around $t = 23$ hours at approximately 290 m. The straight line at one kilometer in Figure 3.3(c) represents the upper limit of the flash lidar instrument, thus the spacecraft traveled outside the flash lidar capabilities briefly, and a data gap occurs in the navigation simulations.

3.1.5 Itokawa Descent Orbit

The descent orbit around Itokawa initiates in a terminator orbit, an impulsive maneuver is applied, and the spacecraft travels in front of the asteroid such that it is seeing the sunlit side of the asteroid as it was descending. Figure 3.4 represents the descent orbit (blue) and the initial terminator orbit (cyan) in the body-fixed frame. The spacecraft arrives at the point of interest (POI) at the end of the descent orbit that is described in Table 3.1. When the spacecraft arrives at the POI, it is at 30° with respect to the terminator plane, and approximately 12 m above the



(a) Inertial frame orbit (b) Body-Fixed frame orbit (c) Altitude

Figure 3.3: Eccentric orbit about Itokawa propagated for 30 hours

surface of Itokawa. The total time of flight of the descent trajectory is approximately 420 minutes. The dynamics for the descent orbit are the same as described in Section 2.2, and do not change when the spacecraft is close to the surface.

Table 3.1: Point of interest state, and the final state of the descent orbit.

State	Value
Latitude	0°
Longitude	0°
Radius	300 m

3.2 Simulation Setup

The OD simulations assumed that characterization phase of the small body mission had been completed, and a well-defined shape model was available. It was assumed that the spin state and axis of the asteroid, as well as the dynamical environment of the asteroid were known.

A faceted shape model of Itokawa and Bennu were used for the measurement computation. For the nominal simulations in this chapter, the Itokawa shape model contained approximately 800 facets and was based on the implicitly connected quadrilateral (ICQ) format. In the descent orbit simulation, an Itokawa shape model with approximately 12,000 facets was used for the measurements. The shape model with 800 facets is referred to as the FV-8 model and the shape model with 12,000 facets is referred to as the FV-32 model. [44] The Bennu shape model had approximately

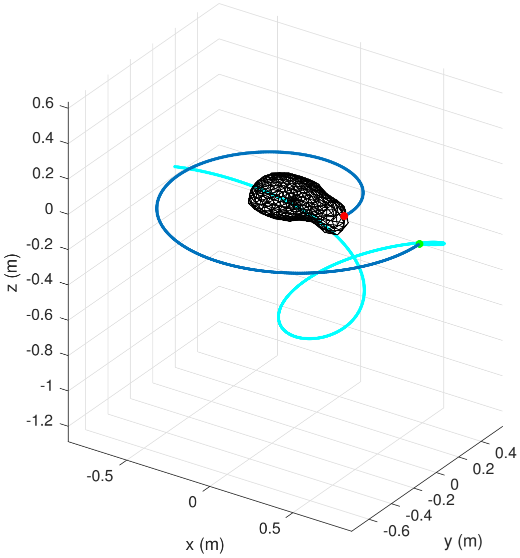


Figure 3.4: Descent orbit (blue) from its intital terminator orbit (cyan) in the body-fixed frame about Itokawa

2700 facets [81], and was constructed from ground-based observations [82].

3.2.1 Sensor Model

The flash lidar measurements were simulated as an array of vectors originating at a focal point, and the observed range, ρ was measured from the sensor plane at the spacecraft state to the intersection of the asteroid (Figure 2.2). The sensor was comprised of a grid map of pixels with a 20° total field of view (FOV) in the $x-$ and $y-$directions, and a focal length of 10 meters. While flash lidar instruments can have resolutions of 256×256, a 64×64 resolution was used for computational purposes. The $z-$axis of the sensor frame connected the focal point to the center of the pixel array, and the $x-$ and $y-$axes were the pixel/line directions.

The measurement uncertainty of each of the lidar measurements had a baseline uncertainty of 1 cm with a variable uncertainty based on the observed range, ρ_{iobs}:

$$3\sigma_{\rho_i} = 1 \text{ cm} + 0.01\rho_{iobs} \tag{3.1}$$

This was based on a baseline model of uncertainty on lidar beams seen in the literature [9]. The measurement uncertainty matrix, R_ρ, was defined as a diagonal matrix in Equation 2.171, and calculated from observed range (Equation 3.1).

$$R_\rho = \texttt{diag}[\sigma_{\rho 1}^2, \cdots, \sigma_{\rho i}^2, \cdots, \sigma_{\rho p}^2] \tag{2.171}$$

A maximum value of 1 km was applied to the ranges, and is reflective of the upper limit of a flash lidar instrument's capabilities when the full array is illuminated. Longer ranges are achieved with different flash lidar instrument models, or different illumination patterns. If a small portion of the flash lidar array illuminated, more power is directed through those pixels, and a larger range limit is achieved.

3.2.2 Frames

The asteroid body-fixed (BF) frame was defined with the x-direction along the longest axis, the z-direction as the spin axis, and the y-direction completing the right-hand system. The asteroid-centered inertial frame (ACI) was defined with the x-direction away from the sun, the z-direction as the spin axis, and the y-direction completing the system. With this definition, there was a simple a rotation about the z-axis to rotate between the body-fixed and inertial frames. Figure 3.5 illustrates these coordinate frames on the asteroid Itokawa.

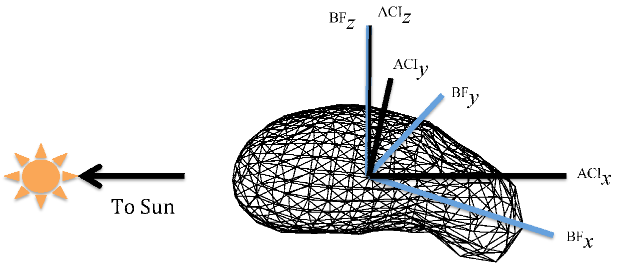


Figure 3.5: Body-fixed (BF) and asteroid-centered inertial (ACI) frames defined on the asteroid Itokawa.

The orbit determination results are presented in the radial, in-track, cross-track (RIC) frame, where the R-direction is directed radially from the center of the asteroid to the spacecraft state, the C-direction is along the orbit's angular momentum vector, and the I-direction completes the system.

The sensor frame (SF) was oriented such that the $+z$-direction was pointed radially at the center of the asteroid, the $+y$-direction was pointed along the positive orbit normal, and the $+x$-direction was directed along the negative orbit in-track direction.

3.2.3 Initial Covariance and Process Noise

For consistency across the nominal OD simulations, the same *a priori* covariance and same level of process noise was used. Simulations were run with and without the addition of process noise. The 1σ value in the *a priori* covariance for position was 10 m, and for velocity was 1 mm/s. The *a priori* covariance, $\bar{P}_0$, is shown in Equation 3.2, and the process noise added had a magnitude along each axis of 1E−9 m/s^2 (Equation 3.3). (For reference, the acceleration of the SRP and the J2 term of the Itokawa terminator orbit were both on the order of 1E−8 m/s^2.)

$$\begin{aligned} \bar{P}_0 &= \texttt{diag}[\sigma_{\mathbf{r}}^2,\ \sigma_{\mathbf{v}}^2] \\ \sigma_{\mathbf{r}} = 10\text{ m},\ \sigma_{\mathbf{v}} &= 1\text{ mm/s} \end{aligned} \tag{3.2}$$

$$\begin{aligned} Q &= \texttt{diag}[\sigma_Q^2,\ \sigma_Q^2,\ \sigma_Q^2] \\ \sigma_Q &= 1\text{E} - 9\text{ m/s}^2 \end{aligned} \tag{3.3}$$

3.3 Orbit Determination Estimating Position and Velocity

The following simulations estimated the position and velocity of the spacecraft, and the pointing of the spacecraft was assumed to be known. Errors were present in the initial state based on the *a priori* covariance in Equation 3.2. Gaussian white noise was added to the individual ranges based on the measurement uncertainty in Equation 3.1.

$$X = \begin{bmatrix} \mathbf{r} \\ \\ \mathbf{v} \end{bmatrix} \tag{3.4}$$

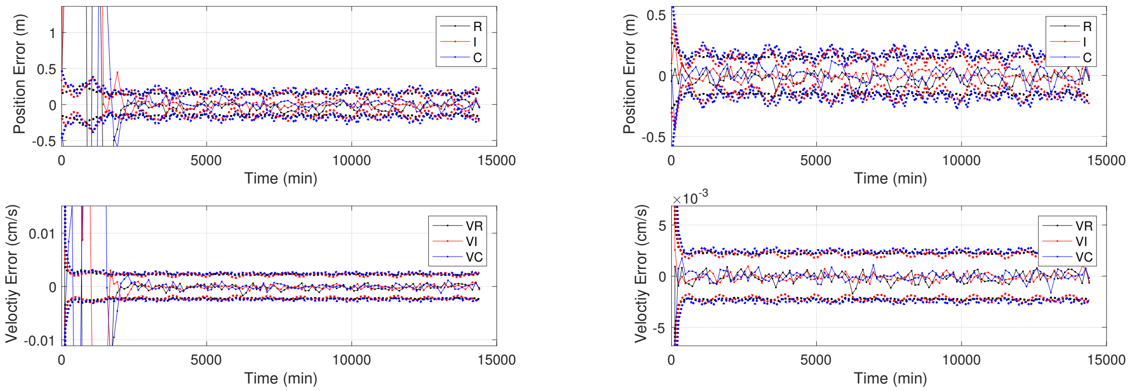
The Itokawa terminator orbit and Bennu terminator orbit were used in these OD simulations. In the Itokawa terminator orbit, a simulation involved observations every two hours for ten days, and another involved observations every 10 minutes for 30 hours. Simulations were also run with the 2 hour observation frequency without process noise to analyze its effect. The Bennu terminator orbit acquired observations every two hours for ten days, and also investigated the effect of process noise.

3.3.1 Itokawa Terminator Orbit Results

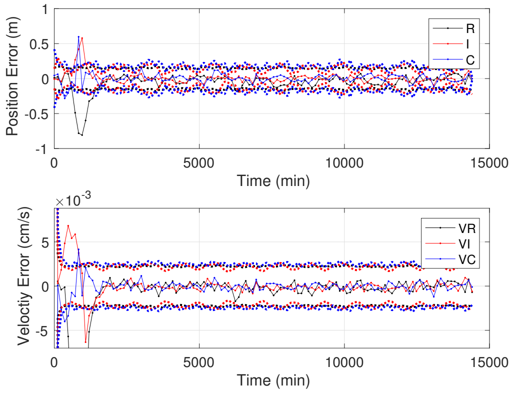
Two observation frequencies were investigated: an operations measurement frequency and a high measurement frequency. The operations measurement frequency acquired measurements every two hours for ten days. This is a common observation frequency seen in small body missions to ensure the navigation solution remains up-to-date while accomplishing scientific goals of the other onboard instruments. The high measurement frequency acquired measurements every ten minutes for 30 hours. This measurement frequency would be typical if the spacecraft was in uncertain dynamics or an unstable orbit, and the OD state required a higher frequency of observations to remain up-to-date.

3.3.1.1 Operations Measurement Frequency

The observations were taken every two hours to simulate the typical observation frequency of optical navigation measurements, and the observation arc was ten days. This resulted in a total number of observations of 121. Figure 3.6 compares the performance of the EKF (a), UKF (b) and LS filter (c). Figure 3.6 illustrates the errors in position and velocity with the 3σ covariance bounds in the RIC frame. Each filter began with the same *a priori* state, and processed the same measurements. Table 3.2 presents the root-mean-squared (RMS) errors of the magnitude of the



(a) EKF (b) UKF



(c) LS

Figure 3.6: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame with process noise for the Itokawa terminator orbit with measurements every 2 hours for 10 days.

position and velocity errors over the course of the simulation and after the first 5 observations when the filter has settled (†). The table also presents the run time of each of the simulations. It is of note that the errors presented average the errors over the entirety of the simulation. The errors in the first few observations were sometimes one or two orders of magnitude larger than those near the end of the simulation.

Comparing the simulations, the EKF resulted in the largest errors. The initial position errors reached approximately 50 m at the beginning of the simulation, and this is present in the RMS of the position errors before with the inclusion of the first five observations. The UKF and LS resulted in comparable state errors at centimeter-level accuracy in position, and the LS ran slightly faster

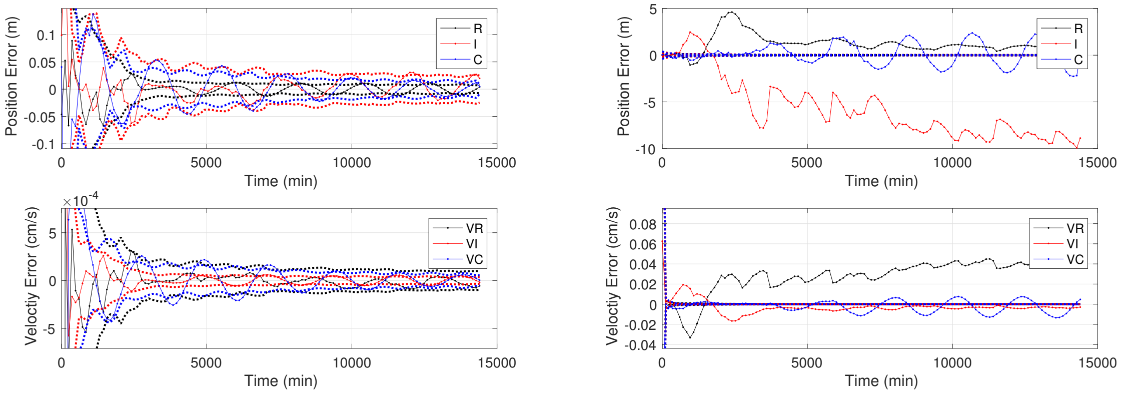
Table 3.2: RMS errors of the magnitude of the position and velocity error over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Itokawa terminator orbit with measurements every 2 hours for 10 days and process noise added.

Filter	$\lvert \mathbf{e_r} \rvert$ (m)	$\lvert \mathbf{e_v} \rvert$ (cm/s)	$\lvert \mathbf{e_r} \rvert$† (m)	$\lvert \mathbf{e_v} \rvert$† (cm/s)	Run Time (min)
EKF	14.00	0.0746	3.84	0.041	3.2
UKF	0.102	0.0211	0.0877	7.66E-4	9.7
LS	0.185	0.0216	0.185	4.54E-3	8.2

than the UKF. The errors in the LS filter reached slightly above the covariance bounds near the beginning of the simulation before reaching a steady state within the covariance bounds. The UKF resulted in errors within the covariance bounds for the entirety of the simulation. The process noise added is seen in the covariance bounds of the velocity errors, and the covariance remains open such that the errors resemble noise.

Figure 3.7 compares the performance of the UKF (a) and LS filter (b) without process noise added. The same *a priori* state and measurements were used as in the simulation with process noise (Figure 3.6). Table 3.3 presents the RMS errors of the magnitude of the position and velocity errors over the course of this simulation, after the first 5 observations (†), as well as the run times of the filters.

Without process noise added, the errors in the UKF simulation were the same order of magni-



(a) UKF

(b) LS

Figure 3.7: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame without process noise for the Itokawa terminator orbit with measurements every 2 hours for 10 days.

Table 3.3: RMS errors of the magnitude of the position and velocity error over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Itokawa terminator orbit with measurements every 2 hours for 10 days and process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	Run Time (min)
UKF	0.0682	0.0211	0.0419	2.23E-4	9.9
LS	6.84	0.0398	6.99	0.0344	14.3

tude smaller in position and velocity compared to the simulation with process noise. Oscillations are seen in Figure 3.7(a) in the in-track and cross-track position errors. This was due to observational issues in these directions due to the orientation of the orbit in the terminator plane. Conversely, the errors in the LS filter were an order of magnitude larger in position and velocity when process noise was not added. The LS filter did not diverge, yet the errors were not bounded in the covariance bounds, and the position errors averaged about 7 m.

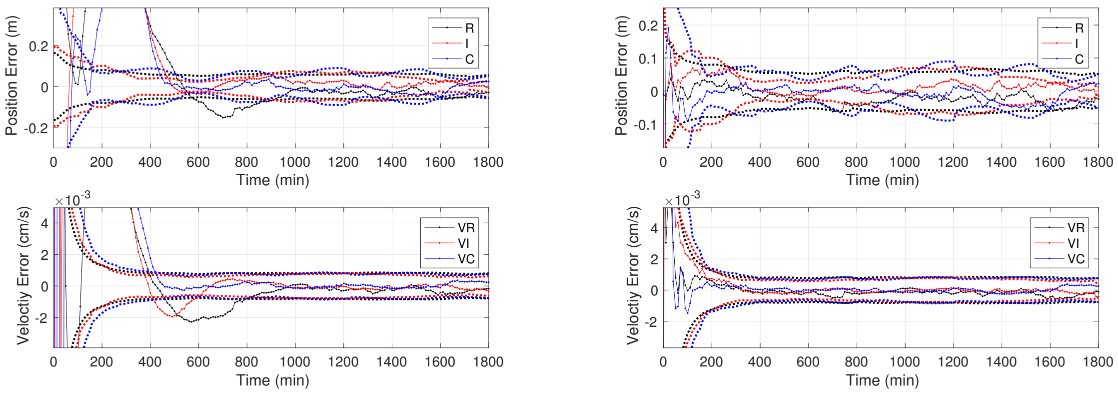
The post-fit residuals for both simulations, after the first measurements, were bounded between ± 10 m. This is consistent with the noise level of the range observations, which was about 9 m (3σ) at the altitude of the spacecraft. The state knowledge dropped below this noise level, however, due to the variety of geometries in each observation.

3.3.1.2 High Measurement Frequency

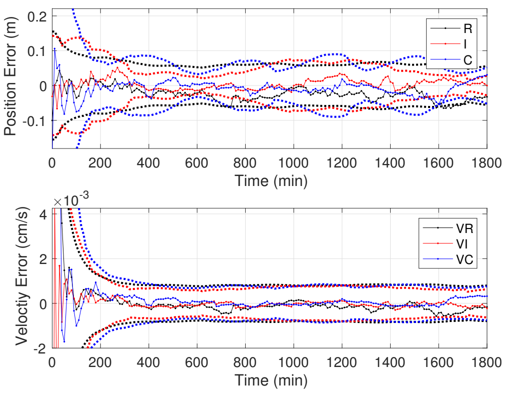
The following simulations were taken in the Itokawa terminator orbit at a frequency of 10 minutes for 30 hours for 181 total observations. This simulates a high measurement frequency situation, and more information is gathered over a shorter period of time. The initial state involved errors from the *a priori* covariance in Equation 3.2, and process noise added with Equation 3.3. The same *a priori* state and the same measurements were used in each of the filters.

Figure 3.8 illustrates the position and velocity state errors and the 3σ covariance bounds in the RIC frame for the EKF (a), UKF (b), and LS filter (c). Table 3.4 presents the RMS errors of the magnitude of the position and velocity errors over the course of the simulation and the errors without the first 5 observations (†).

The UKF and LS filter resulted in comparable errors after the first five observations, with



(a) EKF (b) UKF



(c) LS

Figure 3.8: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame with process noise for the Itokawa terminator orbit with measurements every 10 minutes for 30 hours.

the LS filter running faster. The errors with the EKF were two orders of magnitude larger than those from the UKF and LS filter, and ran the fastest of the three filters. The errors from the LS filter when the first 5 observations were included were similar to when these observations were excluded, showing consistency across the simulation. With the UKF, the errors were larger in the beginning of the simulation, showing that the filter needed more observations to converge.

Comparing the RMS errors of the different observation frequencies, the errors are comparable across the three filters after the first 5 observations. The run time of the filters also followed the same trend between the two simulations: the EKF was the fastest, the UKF was the slowest, and the LS filter was in the middle, yet closer to the UKF run time.

Table 3.4: RMS errors of the magnitude of the position and velocity error over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Itokawa terminator orbit with measurements every 10 minutes for 30 hours and process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	Run Time (min)
EKF	1.01	0.0316	0.775	0.0237	5.9
UKF	0.0859	0.0147	0.0468	8.26E-4	14.7
LS	0.0453	8.44E-3	0.0405	4.68E-4	10.9

3.3.2 Bennu Terminator Orbit Results

The terminator orbit from Section 3.1 around Bennu was investigated for OD. The shape of Bennu is much closer to spherical, as opposed to the shape of Itokawa, which is more unique and elongated. In addition, the flash lidar images of Bennu in this orbit occupy the entirety of the array for the majority of the observations.

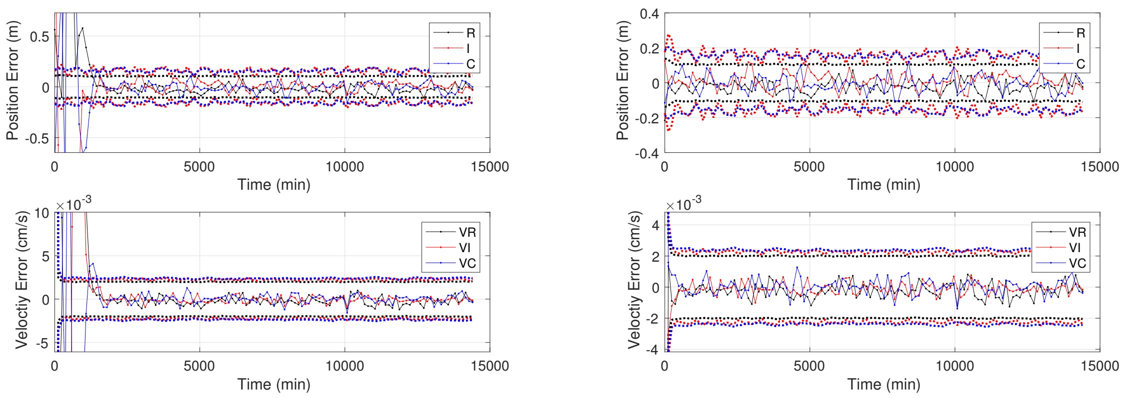
Measurements were taken every 2 hours for 10 days to represent a measurement frequency seen in operations. The *a priori* covariance was defined from Equation 3.2, and process noise was added with Equation 3.3. The *a priori* state included errors based on the *a priori* covariance, and each filter simulation employed the same initial state.

Figure 3.9 presents the state errors in the RIC frame and their 3σ covariance bounds for the EKF (a), UKF (b), and LS filter (c). Table 3.5 extracts the RMS errors of the magnitude of the position and velocity errors for the entire simulation, the errors excluding the first 5 observations (†), and the run time of the fiters.

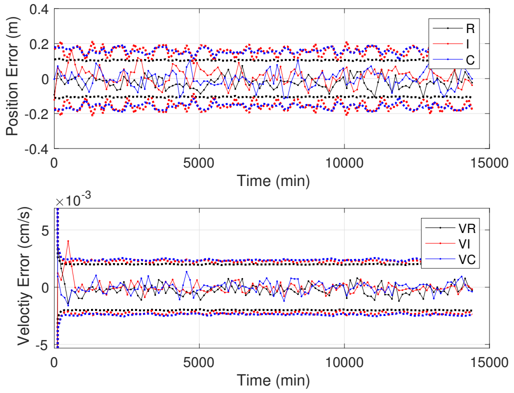
Table 3.5: RMS errors of the magnitude of the position and velocity error over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Bennu terminator orbit with measurements every 2 hours for 10 days and process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	Run Time (min)
EKF	1.74	0.0197	1.21	0.0127	7.2
UKF	0.0805	0.0110	0.0772	7.89E-4	30.4
LS	0.0781	0.0110	0.0772	7.92E-4	26.4

As with the Itokawa terminator orbit, the UKF and LS filter performed the best. The errors from these filters reached centimeter-level accuracy in position, and resulted in almost identical



(a) EKF (b) UKF



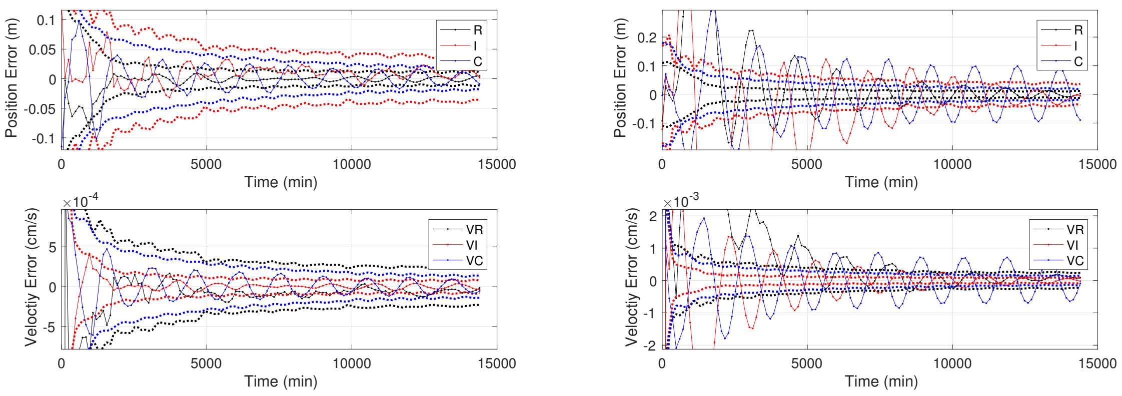
(c) LS

Figure 3.9: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame with process noise for the Bennu terminator orbit with measurements every 2 hours for 10 days.

RMS errors when excluding the first 5 observations. The position errors from the EKF averaged to meter-level accuracy and for the majority of the simulation, achieved centimeter-level accuracy and were within the covariance bounds.

The magnitude of the errors in the Bennu terminator orbit were an order of magnitude smaller than those in the Itokawa terminator orbit (Table 3.2), when comparing the errors for the entirety of the simulations. This is likely due to the magnitude of measurements processed in each simulation. The flash lidar images in the Bennu orbit contained more altimetry measurements at each observation time. Excluding the initial observations, the errors in the Bennu orbit and Itokawa orbit were comparable across the different filters.

Simulations were run with the UKF and LS filter without process noise added. The same measurements, *a priori* covariance, and *a priori* state were used as with the simulations from Figure 3.9. Figure 3.10 illustrates the position and velocity errors in the RIC frame and the 3σ covariance bounds for the UKF (a) and the LS filter (b). Table 3.6 represents the RMS of the magnitude of the position and velocity errors for the complete simulation, the RMS errors when excluding the first 5 observations, and the time to run the filters.



(a) UKF (b) LS

Figure 3.10: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame without process noise for the Bennu terminator orbit with measurements every 2 hours for 10 days.

In the UKF (Figure 3.10(a)), oscillations were present in the in-track and cross-track directions as was seen with the Itokawa terminator orbit (Figure 3.7(a)). This orbit was also circular in the terminator plane, and was subject to the same sinusoidal effect of the phasing between the in-track and cross-track directions. The LS filter did not produce errors with in the 3σ covariance bounds, and oscillations were seen in all three directions.

With process noise added, the UKF errors resembled noise as the covariance was larger due to the additional process noise. The averaged errors were lower in the UKF when process noise was not added. The averaged errors with the LS filter without process noise present were an order of magnitude larger than when process noise was added.

Table 3.6: RMS errors of the magnitude of the position and velocity error over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Bennu terminator orbit with measurements every 2 hours for 10 days and without process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	Run Time (min)
UKF	0.0438	0.0109	0.0343	2.53E-4	30.9
LS	0.307	0.0112	0.312	2.18E-3	20.9

3.4 Optical Navigation in the Itokawa Terminator Orbit

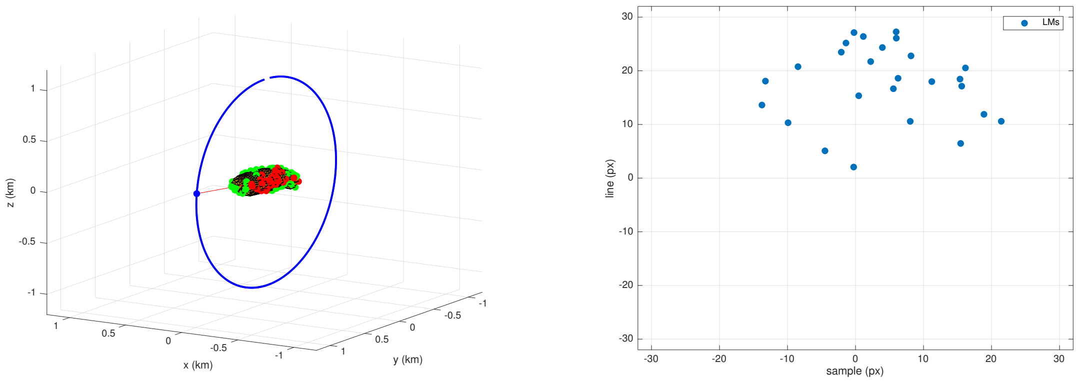
A simulation with optical navigation (OpNav) was performed with the UKF in the Itokawa terminator orbit for comparison with the flash lidar measurements.

One hundred vertices were randomly selected from the FV-8 Itokawa facet model and were treated as the landmark location measurements. The landmarks were considered to be useable to the spacecraft if the landmark was within the camera FOV, sunlit, and not behind the asteroid. The landmarks were calculated with the procedures outlined in Section 2.4.

Figure 3.11 depicts the useable landmarks in the asteroid inertial frame at one state in the Itokawa terminator orbit. The selected state was at 500 minutes into the nominal terminator orbit about Itokawa. In Figure 3.11(a), the red markers on Itokawa represent the useable landmarks, and the green markers represent the remaining non-useable landmarks. The blue marker is the spacecraft position in the orbit, and the red line is its pointing direction, pointing radially at the center of the body. Figure 3.11(b) represents the image view of the sample/line locations of the useable landmarks, and this was the image used in the filter.

The green landmarks that appear to be within the FOV of the spacecraft are not sunlit, and therefore are not considered to be useable. The sun direction in Figure 3.11(a) is in the $-x-$direction, and illuminates the landmarks on that side. The sample/line image view represents this concentration of landmarks and is related to the orientation of the camera sensor frame. The $+y-$axis of the sensor frame was aligned with the normal vector of the orbit, and the orbit normal was always in the direction of the sun.

The measurement uncertainty on each sample/line coordinate was defined with a baseline



(a) Global View

(b) Image View

Figure 3.11: Useable landmarks at one state in the Itokawa terminator orbit. (a) Global view of the useable landmarks (red) and non-useable landmarks (green) in the ACI frame, and (b) Image view of the sample/line locations of the visible landmarks.

value of 0.25 pixels, and an additional term related to its distance from the center of the coordinate system. Equation 3.5 represents the measurement uncertainty applied, where s and l are the sample/line location, and n_{px} is the total number of pixels across one axis:

$$1\sigma = 0.25 + 0.1\frac{\sqrt{s^2 + l^2}}{n_{px}/2}\text{px} \tag{3.5}$$

This was based off the uncertainty model in Owen (2011) [84], and the second term in Equation 3.5 reflects a degradation of the measurement as it gets closer to the edge of the image.

3.4.1 OpNav State Estimation

The position and velocity state of the spacecraft was estimated with a UKF. The set up of the camera sensor, dynamical assumptions, and observation frequency for the OpNav simulation were the same as the flash lidar simulation in order to compare the results as close as possible.

Similar to the flash lidar sensor, the camera sensor model had a total FOV of 20°, the pixel resolution was 64 × 64, and the focal length was 10 m. The sensor frame was oriented such that the camera bore-sight ($+z-$axis) was pointed radially at the center of asteroid, and the $+y-$axis was pointed along the orbit normal. The $x-$axis was considered the sample coordinate axis, and the $y-$axis was considered the line coordinate axis. It was assumed that the dynamics, the shape model, the spin state and axis of Itokawa, and the spacecraft pointing were known.

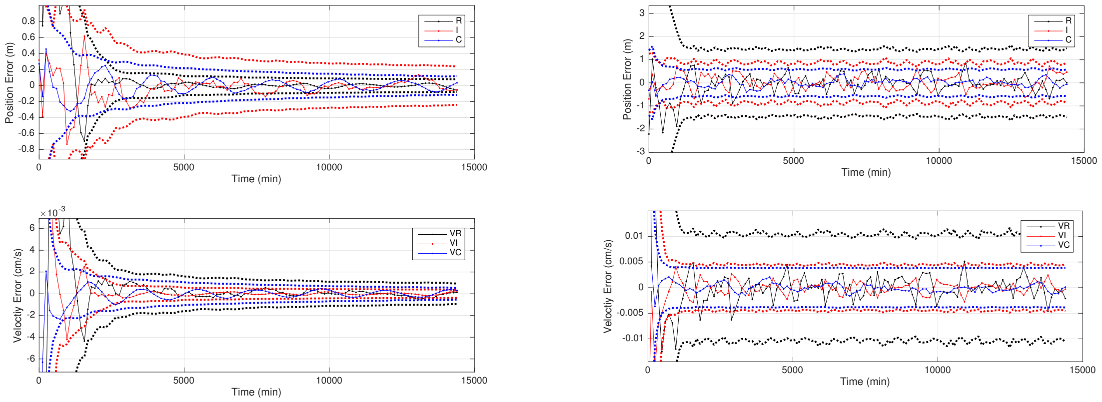
The measurements were taken every 120 minutes for ten days, and the *a priori* filter state included errors based on the *a priori* covariance matrix in Equation 3.2. Two simulations were performed: without process noise and with process noise added along each axis with magnitudes of 1E−9 m/s^2 (Equation 3.3).

It was assumed that each landmark was correctly identified within the filter, and there were no false correlations of the landmarks within the noise. To reduce processing time in the simulations, only the landmarks that were in the observed image were computed in the simulated image.

Figures 3.12(a) and 3.12(b) show the resulting errors in position and velocity in the RIC frame and the 3σ covariance bounds for the simulations without and with process noise, respectively. Without process noise (Figure 3.12(a)) the position errors reach approximately 10 cm, with a larger uncertainty in the in-track direction, and the velocity errors reach approximately 0.01 mm/s. A sinusoidal pattern is present in the in-track and cross-track directions, similar to the flash lidar case, and implies that this phenomenon is not specific to one measurement type. When process noise was added, the maximum position errors were on the order of 50 cm, with the largest uncertainty in the radial direction. The maximum velocity errors were on the order of 0.03 mm/s, with the radial direction also having the highest uncertainty.

3.4.2 Flash Lidar and OpNav Comparison

Table 3.7 presents the RMS of the magnitude of the position and velocity error for the flash lidar and OpNav UKF simulations with and without process noise. The flash lidar results are the same as in Tables 3.2 and 3.3 and are repeated here for comparison. The results show that with almost equivalent simulations, the navigation accuracy with flash lidar performs as good or better than with OpNav. The position errors for the flash lidar simulations were an order of magnitude smaller than those in the OpNav simulations, and the velocity errors an order of magnitude smaller when excluding the first 5 measurements.



(a) Without process noise

(b) With process noise

Figure 3.12: Errors and 3σ covariance bounds in the RIC frame for the OpNav landmark measurements in the Itokawa terminator orbit

With process noise added in the OpNav simulation, the largest uncertainty was found in the radial direction (Figure 3.12(b)). As the OpNav landmarks traverse across the field of view, more information is gained in the cross-track and in-track directions, while the landmarks move little in the radial direction, causing a higher uncertainty.

In the flash lidar simulation without process noise (Figure 3.7(a)), the cross-track errors touched the 3σ covariance bounds throughout the simulation, while in the OpNav simulation (Figure 3.12(a)), the errors remained inside 3σ covariance bounds for the entirety of the simulation. In the flash lidar case, the covariance shrank quickly, down to approximately 10 cm in position by the second measurement. The uncertainty in the OpNav case was an order of magnitude larger by the second measurement, at 1 m, and allowed the errors to be bounded.

3.4.3 Flash Lidar vs. OpNav Discussion

The difference in state errors was likely due to the number of measurements used at each time step. The flash lidar measurements contained approximately 2000 measurements in each image while the OpNav measurements usually had between 15-20 visible landmarks in each image. The lighting conditions played a factor in this simulation; as you can see from Figure 3.11, not all of

Table 3.7: RMS errors of the magnitude of the position and velocity error over the full simulation, and excluding the first 5 observations (†), comparing the OpNav and flash lidar measurements with (w/ PN) and without process noise (w/o PN). This was is in the Itokawa terminator orbit with measurements every 2 hours for 10 days.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)
Flash Lidar UKF (w/o PN)	0.0682	0.0211	0.0419	2.23E-4
OpNav UKF (w/o PN)	0.746	0.0141	0.326	1.92E-3
Flash Lidar UKF (w/ PN)	0.102	0.0211	0.0877	7.66E-4
OpNav UKF (w/ PN)	0.621	0.0160	0.538	2.98E-3

the landmarks in the FOV of the spacecraft could be used because they were shaded from the sun. One hundred landmarks were used within the defined shape model, and more landmarks on the surface could provide a better state solution. For example, the Rosetta mission contained about 1600 landmarks to define the surface [73], yet the average useable landmarks per image were on the order of 50. In addition, the flash lidar observations in these simulations were taken only once every two hours, while the instrument has potential to take measurements with a frequency of up to 30 Hz. In that case, the flash lidar would rapidly provide more measurements than the landmark sample/line measurements.

One drawback to the large number of observations per time step in the flash lidar images, was that it caused the covariance to shrink quickly. With the flash lidar measurements, the covariance shrank faster than the state estimation errors and faster than the covariance in the OpNav simulation. While saturation did not occur in this case, the filter has potential to saturate due the accuracy and large number of measurements per time step.

One aspect of this OpNav simulation that was not included but affects the landmark measurement processing is a scale factor attached to the z-position of the landmarks. Without an altitude attached to the landmark positions with respect to the spacecraft, multiple solutions of the relative position of the spacecraft exist due to the two-dimensional nature of the optical images. The optical measurements only include directional information, while the flash lidar measurements include both directional and range information.

An advantage the flash lidar measurements have over the OpNav landmark measurements, is that the flash lidar measurements do not have to be correlated with the shape model. To use the OpNav landmark vectors on the surface, an iterative process occurs between the spacecraft state, spacecraft pointing, OpNav images, and state of the asteroid to create a landmark map. This iteration process, however, can take hours to days to produce a solution for the navigation, and the current practice is to perform this on the ground [73, 13, 71].

The flash lidar measurements only need to be computed with the shape model onboard the spacecraft and differenced by pixel. This requires much less processing power and ground operations than correlating the OpNav images, and can be more readily moved onboard the spacecraft to encourage autonomous navigation. This method only requires an onboard shape model to create measurement residuals, and shape models can be produced from ground observations before arriving at an asteroid. [24]

3.5 Orbit Determination in the Itokawa Descent Orbit

The Itokawa descent orbit involved an initial circular terminator orbit, an impulsive maneuver, and a 420 minute descent to the surface across the sunlit side of Itokawa. The position and velocity of the spacecraft were estimated with a UKF, flash lidar measurements, and the sensor model as described in Section 3.2.1. This simulation used the FV-32 faceted shape model of Itokawa, instead of the FV-8 model, that contained approximately 12,000 facets. The increase in the number of facets allowed the flash lidar measurements to see more detail of the surface as the spacecraft was descending.

3.5.1 Without Initial State Errors

The first simulation presented assumed the *a priori* filter state did not contain errors, and only included errors in the returned ranges based on Equation 3.1. The *a priori* covariance was set to Equation 3.2, process noise was added from Equation 3.3. The observation frequency was every five minutes for the duration of the orbit (420 minutes).

The state errors and 3σ covariance bounds in the RIC frame are shown in Figure 3.13.

The covariance bounds dropped to centimeter-level in position after the second measurement, and the filter was able to converge onto the correct state without the errors diverging outside of the covariance bounds.

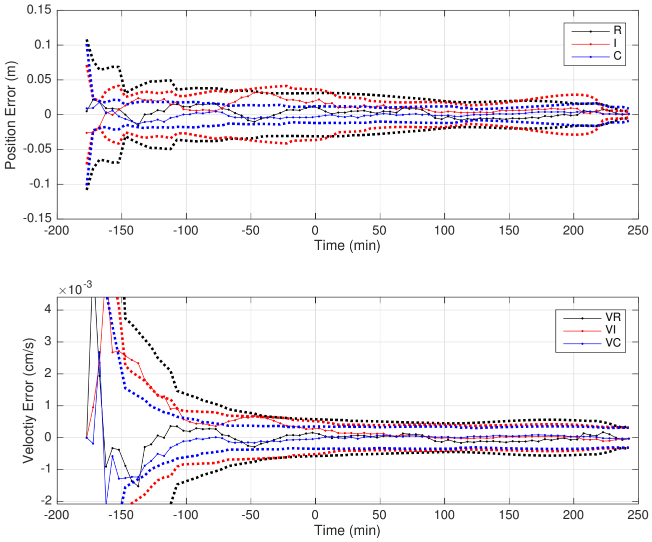


Figure 3.13: Errors and 3σ covariance bounds in the RIC frame for the Itokawa descent orbit with no initial state errors.

3.5.2 Initial State Errors Based on the Terminator Orbit Estimation

The second simulation involved errors in the *a priori* state based on state estimation and maneuver errors. An OD simulation was performed of the initial terminator orbit with the UKF and the FV-32 facet model. In the terminator orbit simulation, errors were applied to the *a priori* filter state based on the *a priori* covariance in Equation 3.2, process noise was added from Equation 3.3, and variations on the range measurements were present from Equation 3.1. Observations were taken every 5 minutes for 0.5 days before the maneuver for the descent orbit.

The final estimated state of the terminator orbit, $\hat{X}_{ft}$, was used to determine the initial state of the descent orbit. The initial position was the same as in $\hat{X}_{ft}$ and involved an error on the order of 1 mm in each direction. The initial velocity was the final estimated velocity from the terminator orbit, plus an instantaneous maneuver, $\Delta\mathbf{v}_{1e}$. The $\Delta\mathbf{v}_{1e}$ maneuver included an error in its magnitude based on the true maneuver, $\Delta\mathbf{v}_1$, and a small directional error. The error in the magnitude was calculated with Equation 3.6, where $\sigma_{\Delta\mathbf{v}} = 0.05|\Delta\mathbf{v}_1|$.

$$|\Delta\mathbf{v}_{1e}| = (|\Delta\mathbf{v}_1| + \sigma_{\Delta\mathbf{v}}) \tag{3.6}$$

The magnitude error on $\Delta\mathbf{v}_{1e}$ was normally distributed by 5% of the magnitude of $\Delta\mathbf{v}_1$. The directional error was pointed off of the nominal pointing direction by an angle θ_e, with a twist. The error angle, θ_e was zero-mean Gaussian noise with $1\sigma = 0.05°$, or approximately 0.87 mrad. The *a priori* velocity state, $\mathbf{v}_0$ for the descent orbit was:

$$\mathbf{v}_0 = \hat{\mathbf{v}}_{ft} + \Delta\mathbf{v}_{1e} \tag{3.7}$$

The *a priori* covariance, $\bar{P}_0$, was a diagonal matrix shown in Equation 3.8. The descent position covariance, $\sigma_\mathbf{r}$, was inflated to 10 times that of the ending position covariances from the terminator orbit estimation, σ_{Rft}, while the descent velocity covariance, $\sigma_\mathbf{v}$ was reset to 3 mm/s. (For reference, 5% of $|\Delta\mathbf{v}_1|$ was approximately 3 mm/s.)

$$\begin{gathered} \bar{P}_0 = \mathtt{diag}[\sigma_\mathbf{r}^2,\ \sigma_\mathbf{v}^2] \\ \sigma_\mathbf{r} = 10\sigma_{Rft},\ \sigma_\mathbf{v} = 3 \text{ mm/s} \end{gathered} \tag{3.8}$$

In the filter, process noise was added uniformly in each direction, similar to Equation 3.3, but increased to $\sigma_Q = 1\text{E}-7$ m/s^2. While lower values of process noise were evaluated, this increased value was found to work best in this new trajectory. Variations on the range measurements based on Equation 3.1 were applied, and observations were taken every five minutes with the FV-32 shape model for the truth and filter measurements.

Figure 3.14 shows the filter results in the RIC frame for this simulation. With the errors applied to the $\Delta\mathbf{v}$ maneuver, the estimation errors were not inside of the covariance bounds at the very beginning of the simulation. However, after approximately 18 observations, the errors converged inside of the covariance bounds.

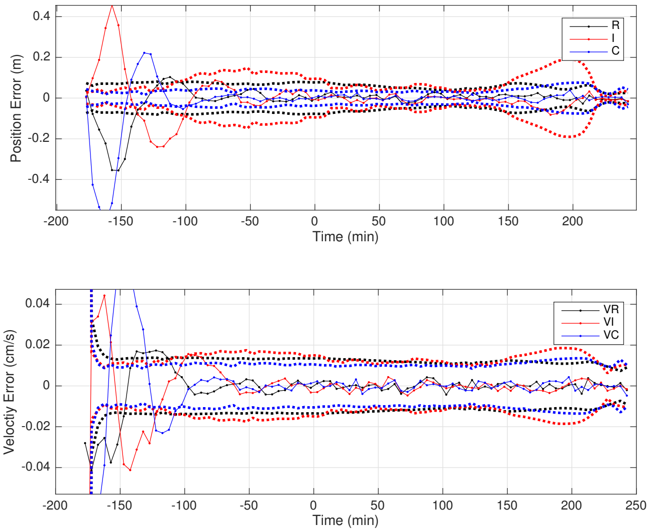


Figure 3.14: Errors and 3σ covariance bounds in the RIC frame for the Itokawa descent orbit with initial state errors.

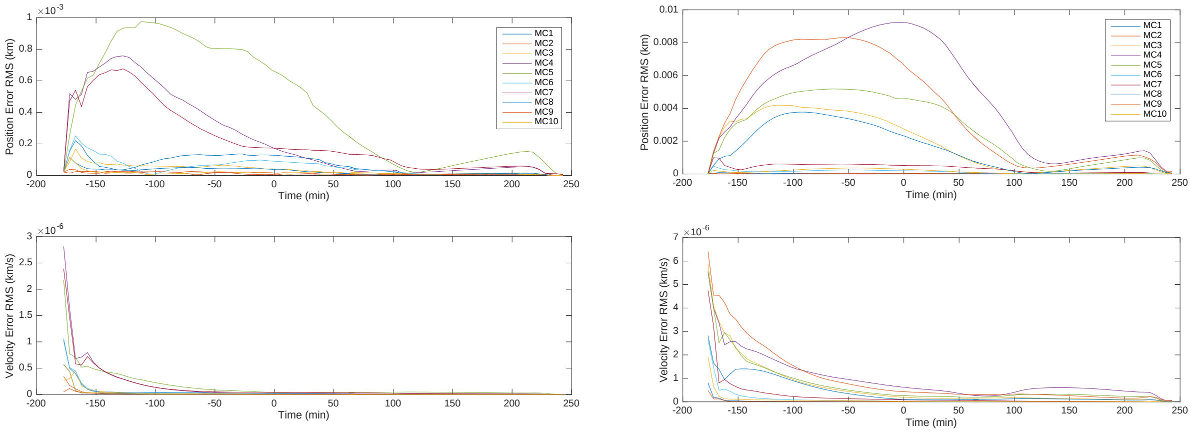
3.5.3 Monte Carlo Simulations with Varying Maneuver Errors

In order to investigate the effect of the $\Delta\mathbf{v}$ error in the filter, two Monte Carlo (MC) simulations were performed with a normally distributed error of 2% and 10% of the magnitude of $\Delta\mathbf{v}_1$. This corresponds to $\sigma_{\Delta\mathbf{v}} = 0.02|\Delta\mathbf{v}_1|$, and $\sigma_{\Delta\mathbf{v}} = 0.1|\Delta\mathbf{v}_1|$ in Equation 3.6. Each simulation involved ten runs, and the results in Figure 3.15 show the RMS errors in position and velocity at each time for each of the MC runs.

In each of the figures, the MC runs with high velocity errors correspond to high position errors, as expected. The largest position error in 2% Error simulation reached approximately 1 m, and for the 10% Error simulations, the position error reached approximately 10 m. The errors in each simulation converged to zero by the end of the trajectory.

3.5.4 Altered Observation Frequency

A new observation scheme was next investigated by taking observations at a higher frequency directly after the maneuver, and then resuming a lower observation frequency for the remainder of



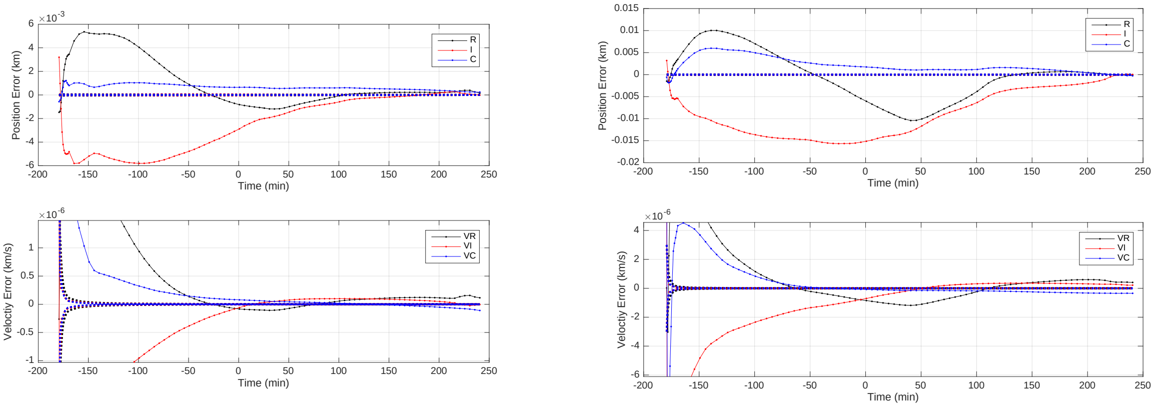
(a) 2% $|\Delta\mathbf{v}|$ Error (b) 10% $|\Delta\mathbf{v}|$ Error

Figure 3.15: RMS errors in position and velocity over the time of the descent orbit trajectory for each of the MC runs.

the trajectory. With the flash lidar measurements, information on the velocity is not obtained until the second measurement. With the previous uniform observation frequency the second observation was five minutes after the maneuver. The theory was that by increasing the observation frequency after the maneuver, the error in the $|\Delta\mathbf{v}|$ can be resolved quicker and reduce the overall errors.

The observation frequency after the maneuver (at the beginning of the observation arc) was increased to an observation every one minute for ten minutes, and then every five minutes for the remainder of the trajectory. Two initial states were tested with this observation scheme: MC 7 from the 2% error simulation and MC 5 from the 10% error simulation. The velocity error in MC 7 from the 2% Error simulation was approximately 0.2 cm/s, and the velocity error in MC 5 from the 10% Error simulation was approximately 0.55 cm/s. These two cases were chosen because the RMS errors were larger in comparison to the other MC runs, but were not at the extremes. The filter simulations had the exact same setup, except for the altered observation frequency.

Figure 3.16 shows the filter results from these two simulations. Figure 3.16(a) shows MC 7 with a 2% $|\Delta\mathbf{v}|$ error and Figure 3.16(b) shows MC 5 with a 10% $|\Delta\mathbf{v}|$ error. The position and velocity errors were outside the covariance bounds for almost the entirety of the simulation.



(a) MC 7, 2% $|\Delta \mathbf{v}|$ Error

(b) MC 5, 10% $|\Delta \mathbf{v}|$ Error

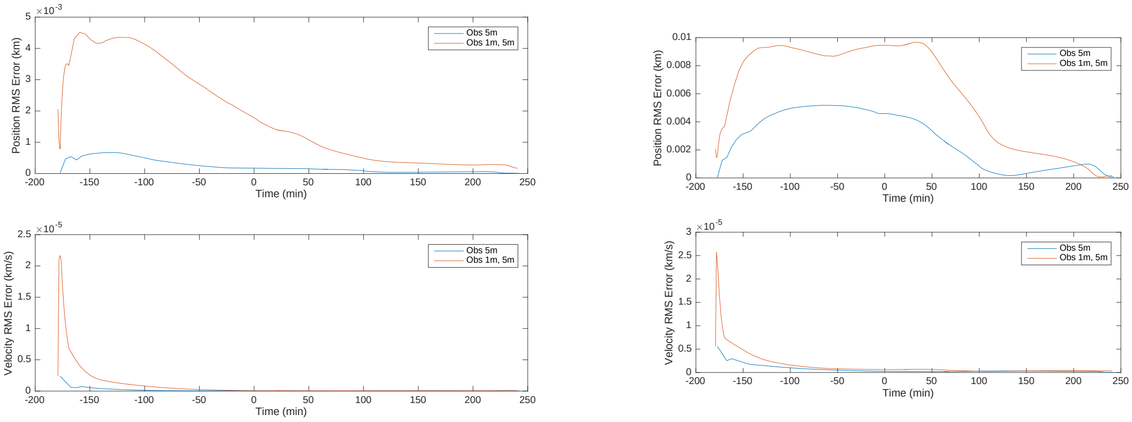
Figure 3.16: Errors and 3 σ covariance bounds in the RIC frame with the altered measurement frequency: observations every one minute for ten minutes, then every 5 minutes. This was applied to two initial states from the MC simulations, (a) MC 7 with 2% $|\Delta \mathbf{v}|$ error and (b) MC 5 with 10% $|\Delta \mathbf{v}|$ error.

To compare the performance of the two observation schemes, Figure 3.17 illustrates the RMS errors over time in position and velocity for the uniform and altered observation frequencies for these two MC runs. The altered frequency performed worse and had larger errors in position and velocity when compared to the uniform frequency. Due to the increased measurement frequency, the covariance decreased quickly at the beginning of the simulation and saturated the filter. Therefore the state was not updated accurately.

3.5.5 Landing and Descent Discussion

Using flash lidar measurements in a descent orbit situation can provide accurate results when the process noise is adequately tuned. Using a higher observation frequency to solve for errors in the descent maneuver decreased the filter performance. With the higher observation frequency in the UKF, the covariance decreased quickly and the filter saturated.

As opposed to the terminator orbits, the sensor is closer to the body and therefore sees a



(a) MC7, 2% $|\mathbf{\Delta v}|$ Error

(b) MC5, 10% $|\mathbf{\Delta v}|$ Error

Figure 3.17: RMS errors in position and velocity over time for when using a uniform observation frequency of 5 minutes, and an altered scheme of a frequency of 1 minute for 10 minutes, and then every 5 minutes.

smaller portion of the surface and possibly less variations in the surface. In addition, the spacecraft is moving quicker when it is closer to the body and the observed and computed measurements could involve greater discrepancies. This orbit required additional process noise, when compared to the terminator orbit, to prevent the covariance from shrinking too quickly and account for the initial errors.

When compared to the OpNav techniques, flash lidar measurements show promise for descent and landing trajectories. Within OpNav, the number of landmarks decreases as the spacecraft travels closer to the body, and different landmark maps must be created and used. The flash lidar measurements, on the other hand, are able to use the same shape model to calculate the measurement residuals without creating new landmark maps. The onboard shape model, however, does play a factor in the precision of the navigation when using flash lidar. Since the sensor sees a smaller portion of the asteroid at one time, a higher fidelity shape model should be used.

3.6 Orbit Determination Estimating Position, Velocity, and Pointing Error

The estimation state for the OD simulations was expanded to include estimating an off-nominal pointing error in addition to the position and velocity. Estimating the spacecraft pointing is achievable with OpNav because the landmark measurements are estimated in the plane of the camera. Estimating the spacecraft pointing with a single lidar beam or a scanning lidar may be difficult since only one altimetry measurement is used with a single lidar and the direction is constantly changing with a scanning lidar. With the flash lidar instrument, the direction of the pointing is measured in the sensor plane and includes an altimetry measurement.

Three filters were compared: the EKF, UKF, and iterative LS filter. The spacecraft position, $\mathbf{r}$, and velocity, $\mathbf{v}$, were estimated in the ACI frame, and an off-nominal pointing bias, $\underline{\theta}$, defined from Equation 2.2 was estimated. The filter estimation state, X, is shown in Equation 2.1.

$$X = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \underline{\theta} \end{bmatrix} \tag{2.1}$$

The pointing direction, $\hat{u}$ from Equation 2.2 was defined as the z-axis of the sensor frame. The individual pointing directions of the flash lidar array were defined from this coordinate system and fixed with respect to the z-axis. The rotation from in Equation 2.2 was applied to the z-axis of the sensor frame, and the individual pixel pointing directions were defined from this rotated axis.

3.6.1 Filter Setup and Error Modeling

The three measurement processing methods were investigated in the Itokawa terminator oribt, the Bennu terminator orbit, and the Itokawa eccentric orbit. The observations in the terminator orbit around Itokawa and Bennu were taken every two hours for 10 days, for a total of 121 observations. This simulated an operations environment, in which the spacecraft is in a stable orbit, and navigation images are taken approximately every two hours in order to perform OD operations.

In the eccentric orbit, observations were taken every 10 minutes for 30 hours, totaling 181 observations. Since the eccentric orbit is not stable and travels closer to the body, navigation images for OD would likely be taken at a higher frequency to ensure the OD solution remained up-to-date. This observation frequency was also investigated in the Itokawa terminator orbit.

It was assumed that the dynamics of the system, the shape model of the body, and the spin state and axis of Itokawa were known. For the pointing, it was assumed that the rotation matrix between the nominal sensor frame and the asteroid body-fixed frame was known, and the state estimated an off-nominal pointing error.

The error modeling included Gaussian white noise on the truth individual range measurements based on Equation 3.1, and errors in the *a priori* state. In the truth measurements, a constant bias of 0.5° was applied to each pointing angle, θ_n, at each observation time, and was equivalent to a constant off-nominal pointing of about 12 mrad.

The *a priori* state error was based on the *a priori* covariance in Equation 3.9 for the position and velocity, and the *a priori* state for each angle was set to zero. For each orbit, the same *a priori* state was used in the three filter simulations. The 1σ value in the *a priori* covariance for position was 10 m, for velocity was 1 mm/s, and for the angels was 2° (Equation 3.9).

$$\begin{gathered} \bar{P}_0 = \texttt{diag}[\sigma_{\mathbf{r}}^2,\ \sigma_{\mathbf{v}}^2,\ \sigma_{\underline{\theta}}^2] \\ \sigma_{\mathbf{r}} = 10 \text{ m},\ \sigma_{\mathbf{v}} = 1 \text{ mm/s},\ \sigma_{\theta} = 2^\circ \end{gathered} \tag{3.9}$$

Process noise was added on the order of 1E-9 m/s^2 (Equation 3.3), which is slightly below the accelerations due to SRP and the J2 term of Itokawa, which were on the order of 1E-8 m/s^2.

3.6.2 Itokawa Terminator Orbit Results

Three OD scenarios were run in the Itokawa terminator orbit while estimating the position, velocity, and pointing error of the spacecraft. The first two captured measurements every two hours for ten days to emulate a small body mission operations environment, and were run with and without process noise added. The third simulation acquired measurements every 10 minutes for 30 hours, and this would be considered a high observation frequency for a small body mission.

3.6.2.1 Operations Measurement Frequency

Observations were acquired every two hours for ten days, as is common in a small body mission during nominal operations. This totaled 121 observations. Figures 3.18(a), 3.18(b), and 3.18(c) present the state errors and the 3σ covariance bounds in the RIC frame for the EKF, UKF, and LS filter, respectively. Table 3.8 presents the RMS of the magnitude of the position and velocity errors over the entire simulation, and after the first five observations. The angle, α in Table 3.8 was defined as the total angle offset between the estimated pointing direction and the truth pointing direction. The run times of each filter are also provided in Table 3.8.

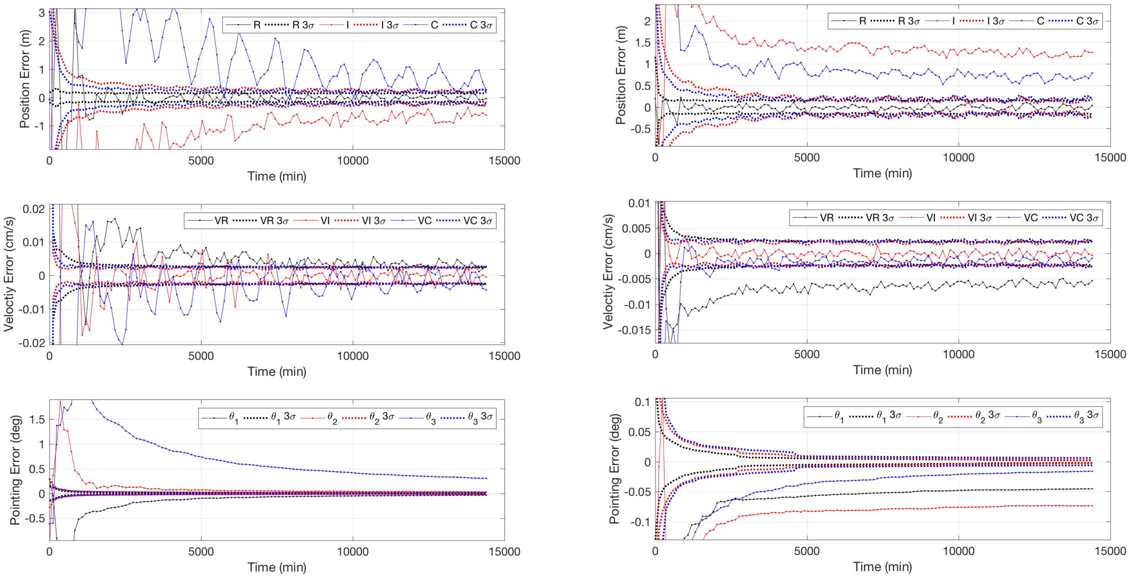
Table 3.8: RMS errors of the magnitude of the position and velocity errors and total pointing error angle, α over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Itokawa terminator orbit, estimating a pointing bias, with measurements every 2 hours for 10 days and process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	α (deg)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
EKF	8.82	0.0482	0.44	4.05	0.0289	0.31	3.5
UKF	2.61	0.0281	0.15	1.79	7.83E-3	0.11	11.4
LS	0.243	0.0266	0.013	0.137	8.79E-4	6.48E-3	6.7

While all of the simulations converged toward the truth, the LS filter was the only simulation where all of the state errors remained inside the covariance bounds for the entirety of the simulation. The radial position errors in the EKF and UKF fell within the covariance bounds, while the error in the pointing angles did not for these simulations. The pointing of the flash lidar sensor was defined as pointing radially at the asteroid with a small pointing error. Therefore, the majority of the information was contained in the radial position direction. The averaged errors from the LS filter were the smallest of the three filters, and ran in almost half the time as the UKF.

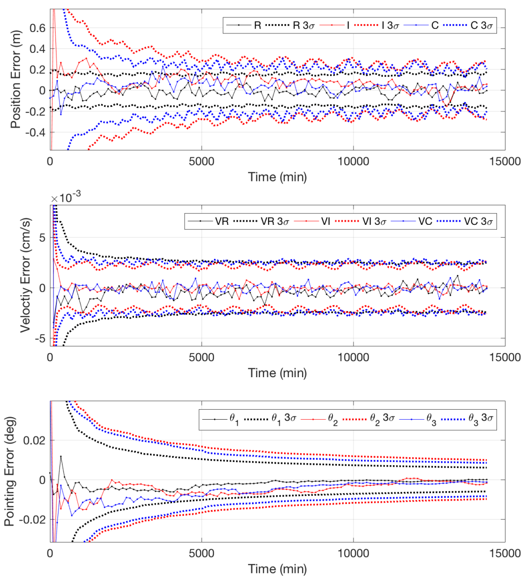
The same measurements and *a priori* state were investigated with the UKF and LS filter, but without adding process noise. Figure 3.19(a) illustrates the state errors in the RIC frame and the 3σ covariance bounds for the UKF, and Figure 3.19(b) illustrates the state errors in the RIC frame and the 3σ covariance bounds for the LS filter. Table 3.9 shows the magnitude of the position and velocity errors and the total pointing angle offset from the truth, α, for the entire simulations and excluding the first five observations.

Without process noise added, the UKF resulted in similar errors to when process noise was



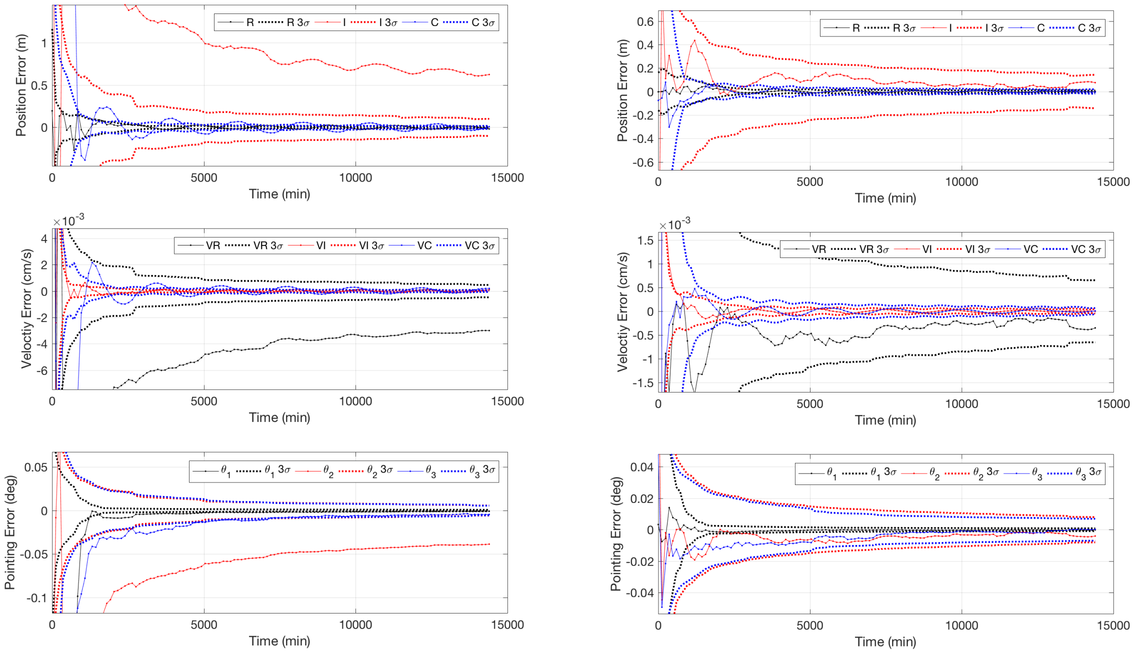
(a) EKF

(b) UKF



(c) LS

Figure 3.18: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame with process noise for the Itokawa terminator orbit with measurements every 2 hours for 10 days.



(a) UKF (b) LS

Figure 3.19: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame without process noise for the Itokawa terminator orbit with measurements every 2 hours for 10 days.

added. In contrast to when process noise was present, the cross-track position error with the UKF fell within the covariance bounds. This trend was the same for the θ_1 pointing angle.

The LS filter also resulted in similar errors with and without process noise, yet resulted in a longer run time without process noise present. Similar to the simulation with process noise, the state errors when process noise was not present were inside their respective covariance bounds for the entirety of the simulation.

3.6.2.2 High Observation Frequency

An observation frequency of 10 minutes for 30 hours was investigated in the Itokawa terminator orbit while estimating pointing. The *a priori* state errors were based off of Equation 3.9, and process noise was added from Equation 3.3. The *a priori* state errors were based on the *a priori* covariance for the position and velocity state, and the initial pointing angles were set to zero. The

Table 3.9: RMS errors of the magnitude of the position and velocity errors and total pointing error angle, α over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Itokawa terminator orbit, estimating a pointing bias, with measurements every 2 hours for 10 days and without process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	α (deg)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
UKF	2.31	0.0278	0.13	1.29	6.41E-3	0.083	12.9
LS	0.234	0.0266	0.013	0.115	5.00E-4	6.10E-3	9.4

same *a priori* state was used in each of the three estimation filters, the EKF, UKF, and LS filter.

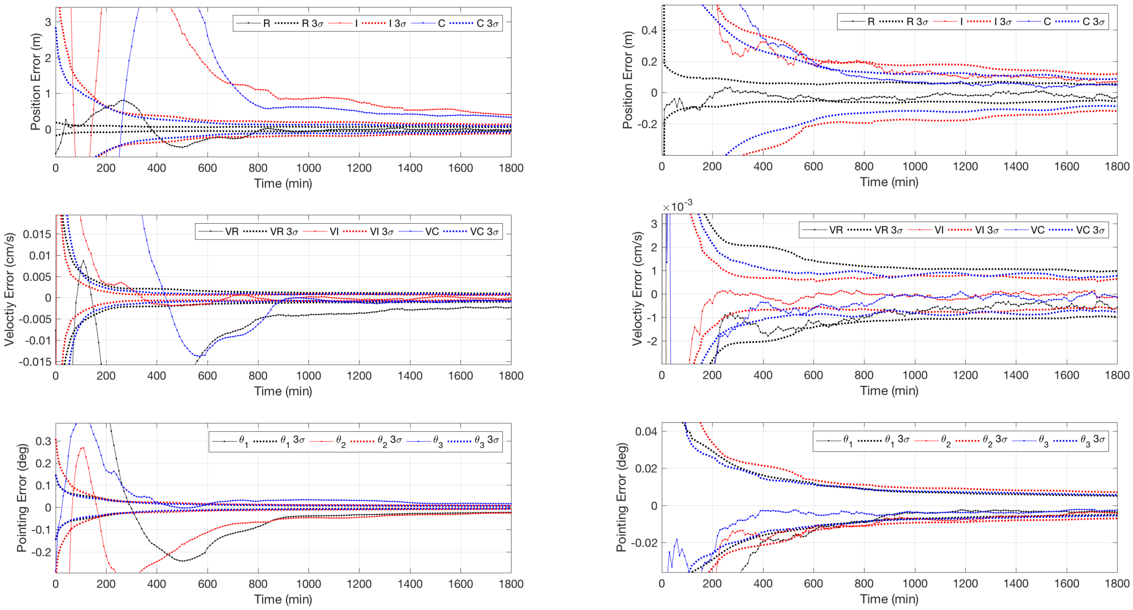
Figure 3.20 presents the state errors and 3σ covariance bounds in the RIC frame for the position and velocity, and the pointing angles with the EKF (a), UKF (b), and LS filter (c). Table 3.10 presents the RMS of the magnitude of the position and velocity errors, and the pointing offset error angle, α. These are presented for the entire simulation and excluding the first five observations, and the filter run time is presented as well.

Table 3.10: RMS errors of the magnitude of the position and velocity errors and total pointing error angle, α over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Itokawa terminator orbit, estimating a pointing bias, with measurements every 10 minutes for 30 hours and with process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	α (deg)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
EKF	14.30	0.0620	0.72	7.85	0.0523	0.41	7.2
UKF	1.79	0.0174	0.099	0.942	4.51E-3	0.055	19.2
LS	0.408	0.0131	0.023	0.352	1.69E-3	0.021	13.0

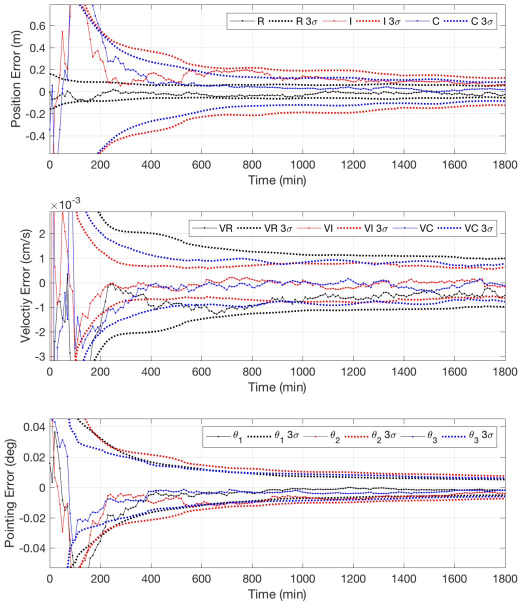
The EKF resulted in errors outside of the 3σ covariance bounds, and produced the largest averaged errors of the three filters. The majority of the state errors in the UKF and LS filters were within their corresponding 3σ covariance bounds. The LS filter generated the smallest averaged errors that were comparable to the errors from the UKF, and ran faster than the UKF.

As opposed to when only position and velocity were estimated, the UKF and LS filters performed differently between the two observation frequencies. With the higher frequency, the state errors from the UKF fell within their covariance bounds as opposed to using the operations frequency. This resulted in smaller overall errors for the high measurement frequency UKF com-



(a) EKF

(b) UKF



(c) LS

Figure 3.20: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame with process noise for the Itokawa terminator orbit with measurements every 10 minutes for 30 hours.

pared to the low measurement frequency UKF. The EKF and LS filter produced comparable errors between the two observation frequencies, with the lower frequency performing better in each case.

3.6.3 Bennu Terminator Orbit Results

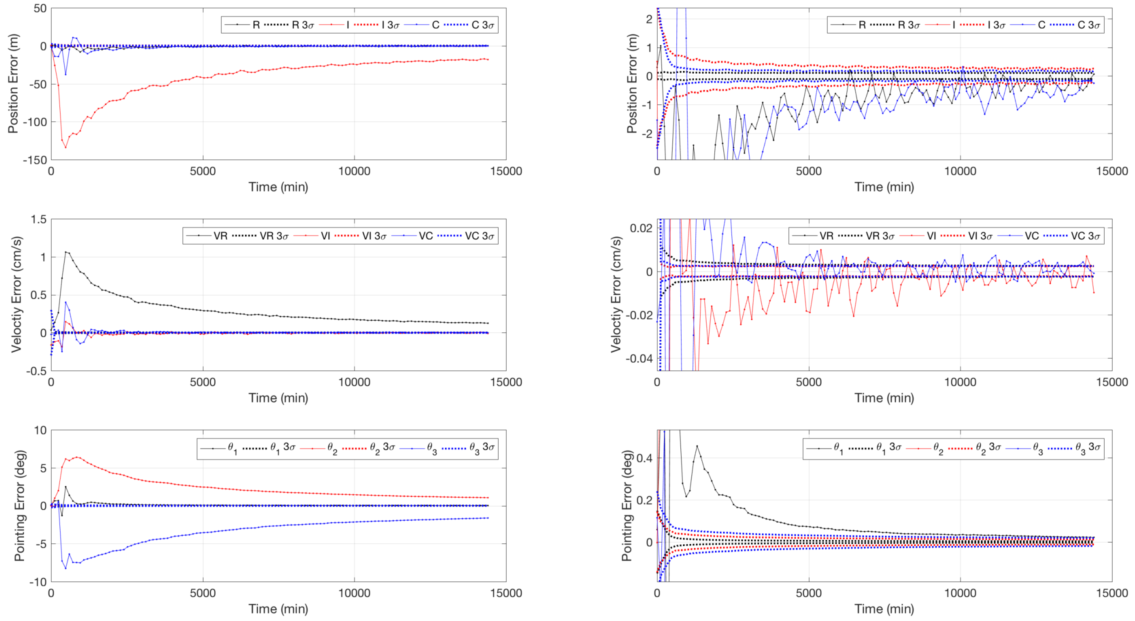
OD in the Bennu terminator orbit was investigated with an observation frequency of 2 hours for 10 days with the EKF, UKF, and LS filter. As with the simulations that only estimated the position and velocity, the flash lidar images of Bennu cover the full image array for the majority of the observations. This increases the number of measurements processed at each time step when compared to the Itokawa orbit measurements.

The *a priori* state errors were based on the covariance from Equation 3.9 for the position and velocity, and the initial pointing angles were set to zero. The same *a priori* state and measurements were used in each of the three filters. The *a priori* covariance was that from Equation 3.9 and process noise was added from Equation 3.3.

Errors in the measurements were the same as with the Itokawa terminator orbit. Gaussian white noise was added to the truth individual range measurements from Equation 3.1, and a constant pointing bias of 0.5° was applied to each pointing angle (total offset was approximately 12 mrad) in the truth measurements.

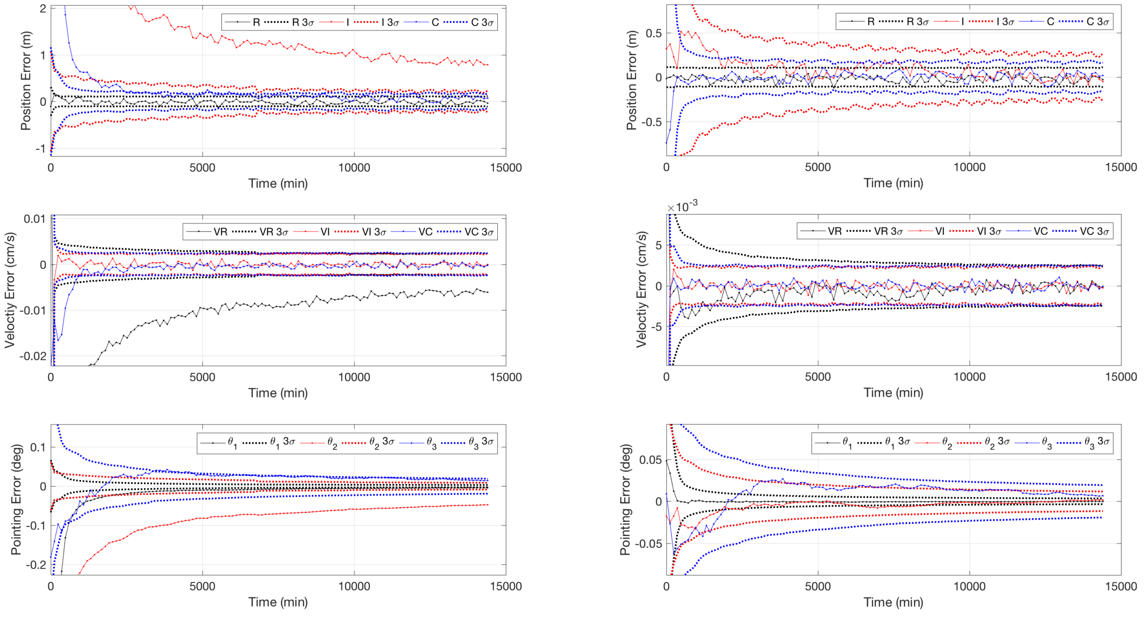
Figure 3.21 presents two views of the EKF results (Figures 3.21(a) and 3.21(b)), the UKF results (Figure 3.21(c)) and the LS filter results (Figure 3.21(d)). The results show the state errors (solid) in the RIC frame for the position and velocity state, and the pointing angle state, and their 3σ covariance bounds (dashed). Table 3.11 presents the RMS of the magnitude of the errors for the entire simulation, excluding the first five measurements, and the run time of the filters. The errors are presented as the magnitude of the position and velocity, and the pointing angle offset from the true pointing, α.

The LS filter resulted in the lowest state errors of the three filters, and the state errors were within the covariance bounds for the entirety of the simulation. The state errors with the EKF filter were orders of magnitude greater than the covariance bounds, and reached over 100 m in



(a) EKF View 1

(b) EKF View 2



(c) UKF

(d) LS

Figure 3.21: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame with process noise for the Bennu terminator orbit with measurements every 2 hours for 10 days.

Table 3.11: RMS errors of the magnitude of the position and velocity errors and total pointing error angle, α over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Bennu terminator orbit, estimating a pointing bias, with measurements every 2 hours for 10 days and with process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	α (deg)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
EKF	48.68	0.359	2.76	46.24	0.341	2.69	8.2
UKF	2.20	0.0202	0.13	1.65	0.0117	0.098	40.3
LS	0.193	0.0152	0.011	0.161	1.36E-3	8.38E-3	21.3

position.

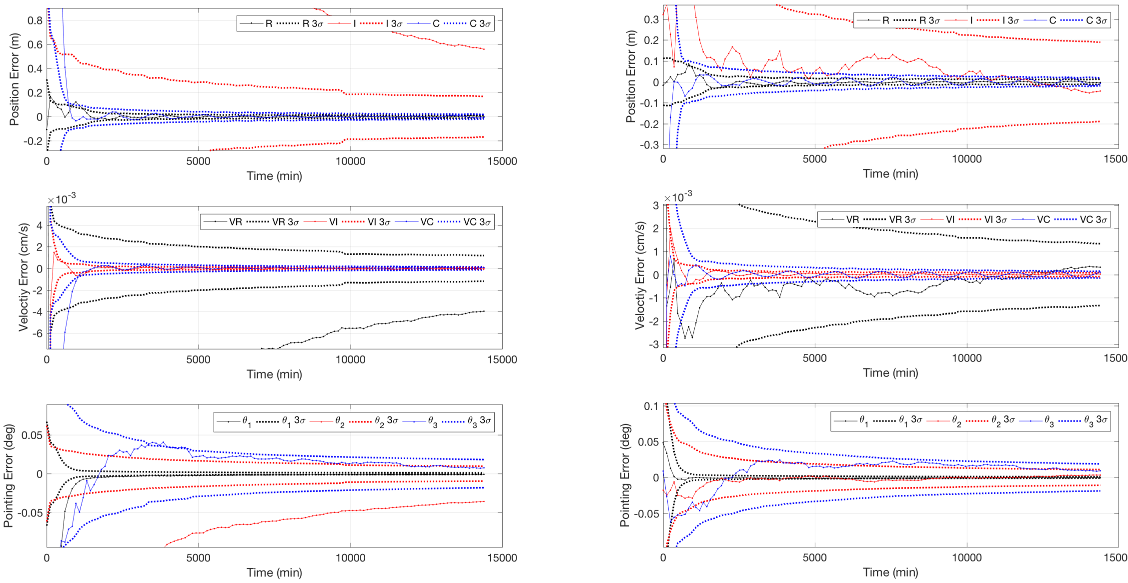
The UKF simulation ran in almost twice the time as the LS simulation and produced errors an order of magnitude larger. The in-track position error and the radial velocity error were outside of their respective covariance bounds. This was a similar trend seen in the UKF simulations with the same observation frequency in the Itokawa terminator orbit. Observability issues may be present due to the orbit geometry and the observation frequency.

The UKF and LS filter were investigated without the addition of process noise in the Bennu terminator orbit. The same measurements and *a priori* state were used as the previous simulations, and the only difference was the exclusion of additional process noise.

Figure 3.22 presents the results of the UKF (a) and LS filter (b) in the Bennu terminator orbit without process noise. The figures present the state errors (solid) in the RIC frame for the position and the velocity, and the pointing state, and their corresponding 3σ covariance bounds (dashed). Table 3.12 presents the magnitude of the errors over the course of the simulation, excluding the first five measurements, and the run time of the filters.

Table 3.12: RMS errors of the magnitude of the position and velocity errors and total pointing error angle, α over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Bennu terminator orbit, estimating a pointing bias, with measurements every 2 hours for 10 days and without process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	α (deg)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
UKF	2.08	0.0197	0.13	1.51	0.0107	0.090	44.6
LS	0.159	0.0152	9.27E-3	0.118	7.43E-4	6.47E-3	21.2



(a) UKF

(b) LS

Figure 3.22: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame without process noise for the Bennu terminator orbit with measurements every 2 hours for 10 days.

In the filter results, the same trend is seen in the UKF with and without process noise. The in-track position direction and the radial velocity direction were outside of their covariance bounds, and these directions contained the greatest uncertainty with the UKF and LS filter. Oscillation patterns were seen in the UKF in the cross-track position and velocity directions, similar to those seen in the Itokawa terminator orbit, and when only estimating the position and velocity.

The errors produced from the LS filter were within their covariance bounds for the entirety of the simulation. This was different from only estimating the position and velocity with the LS filter and without process noise, when the errors were not contained within their covariance bounds. The addition of the pointing angles in the estimation state created more uncertainty, and the position and velocity covariance bounds were larger when estimating pointing than without.

3.6.4 Itokawa Eccentric Orbit Results

OD in an eccentric orbit about Itokawa was investigated with the EKF, UKF, and LS filters. Observations were taken every 10 minutes for 30 hours. This orbit was unstable, as the spacecraft escaped after 30 days of propagation. In a small body mission, if the spacecraft was in such an orbit, observations would be required at a frequency higher than every two hours to maintain the accuracy of the navigation solution.

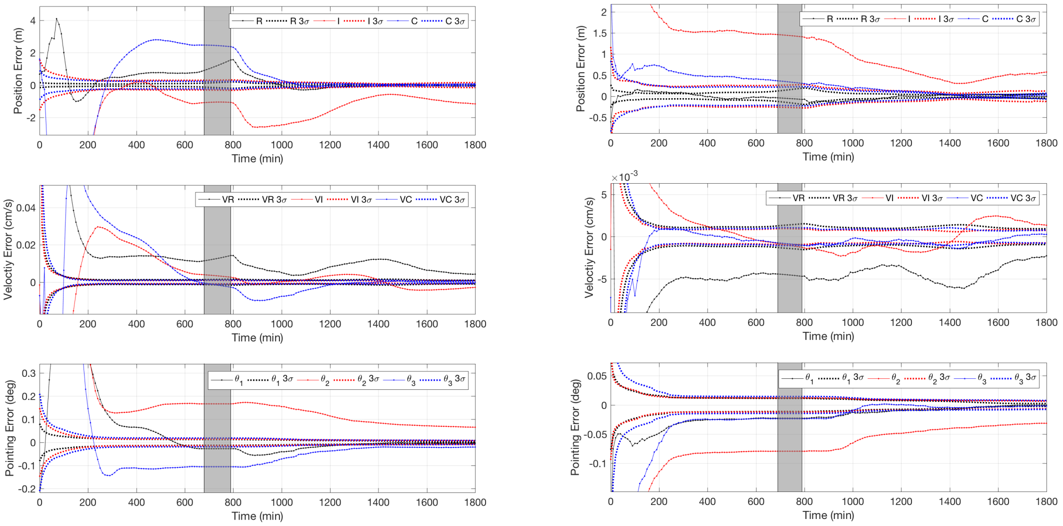
Figures 3.23(a), 3.23(b), and 3.23(c) show the state errors and 3σ covariance bounds in the eccentric orbit for the EKF, UKF, and LS filter, respectively. The gray patch in the figures represents a data gap. The eccentric orbit travels outside of the bounds of the flash lidar instrument ($>$ 1 km), and a gap in the measurements occurs. Table 3.13 presents the RMS of the magnitude of the errors over every time step, excluding the first five observations, and the run time of the filters.

Table 3.13: RMS errors of the magnitude of the position and velocity errors and total pointing error angle, α over the full simulation, excluding the first 5 observations (†), and the run time of the filter. This is in the Bennu terminator orbit, estimating a pointing bias, with measurements every 2 hours for 10 days and with process noise added.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	α (deg)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
EKF	7.01	0.0643	0.48	6.28	0.0507	0.40	6.5
UKF	1.63	0.0194	0.12	1.44	7.45E-3	0.096	18.1
LS	0.131	0.0145	9.09E-3	0.112	7.50E-4	6.22E-3	15.1

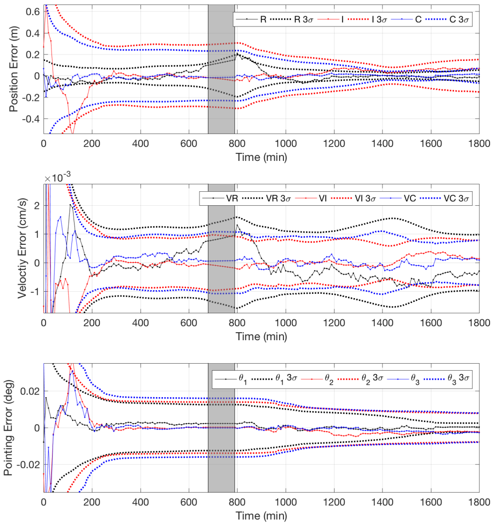
Similar to the Itokawa terminator orbit, the EKF and UKF simulations converged toward the truth, yet the filter saturated in both simulations and the errors were outside of the 3σ covariance bounds. With the LS filter, the state errors were smaller and within the covariance bounds for the majority of the simulation.

In the eccentric orbit, a data gap occurred between 670 minutes and 790 minutes because the spacecraft traveled outside of the one kilometer limit of the flash lidar instrument, as shown in Figure 3.3(c). Figure 3.24 shows the post-fit residuals of LS method, with the data gap, and one can notice the post-fit residuals worsen as the altitude increases. In the observed measurement before the data gap (at 660 minutes), the image of Itokawa was approximately only 15 pixels in diameter. This still provided over 200 range measurements, but was less accurate than the thousands of range measurements used in the previous observations.



(a) EKF

(b) UKF



(c) LS

Figure 3.23: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame without process noise for the Itokawa eccentric orbit with measurements every 10 minutes for 30 hours. The grey patch represents the data gap.

During the data gap in the filter, the estimated state and covariance were simply propagated until a useable observation became available. This caused a slight increase in the covariance bounds, shown in Figure 3.23(c), but due to the high observation frequency and small duration of the gap, the increase remained small.

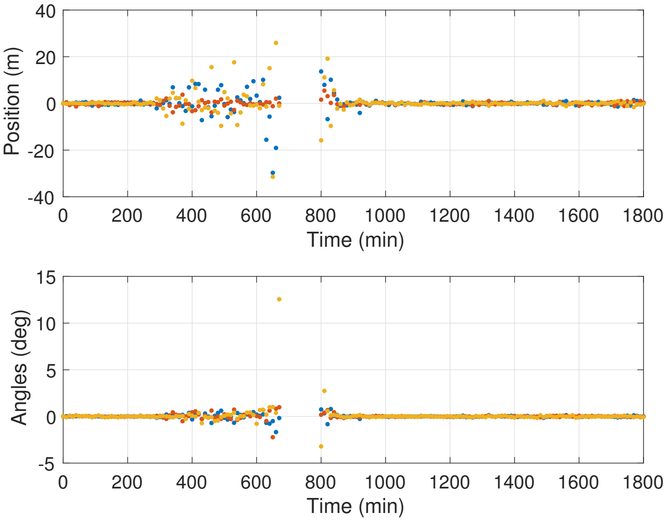


Figure 3.24: Post-fit residuals for the LS simulation; as the spacecraft traveled above 1 km in altitude, a measurement gap occurred due to the upper limit of the flash lidar instrument.

3.7 Summary of Filter Results

Table 3.14 summarizes the filter results for the terminator orbits around Itokawa and Bennu when estimating the position and velocity. The consistency of the errors refers to whether the filter errors for each component were within the 3σ covariance bounds for the majority of the observation arc.

Across the simulations with the Itokawa and Bennu terminator orbits, the iterative LS filter results in comparable errors to the UKF, and runs in a shorter amount of time. With the EKF, once the filter has converged after the initial observations, the resulting errors are within the covariance bounds for both the Itokawa and Bennu terminator orbits. The same level of accuracy is achieved with the high frequency measurements and the operations frequency measurements in the Itokawa

Table 3.14: Summary of RMS errors of the magnitude of the position and velocity error over the full simulation, the consistency of the filter errors, and the run time of the filter. Presented for when estimating the position and velocity.

Orbit	Obs Freq	PN added	Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	Consistent Errors?	Run Time (min)
Itokawa terminator orbit	2 hours, 10 days	✓	EKF	14.00	0.0746	✓	3.2
		✓	UKF	0.102	0.0211	✓	9.7
		✓	LS	0.185	0.0216	✓	8.2
			UKF	0.0682	0.0211	✓	9.9
			LS	6.84	0.0398		14.3
	10 min, 30 hours	✓	EKF	1.01	0.0316		5.9
		✓	UKF	0.0859	0.0147	✓	14.7
		✓	LS	0.0453	8.44E-3	✓	10.9
Bennu terminator orbit	2 hours, 10 days	✓	EKF	1.74	0.0197	✓	7.2
		✓	UKF	0.0805	0.0110	✓	30.4
		✓	LS	0.0781	0.0110	✓	26.4
			UKF	0.0438	0.0109	✓	30.9
			LS	0.307	0.0112		20.9
Itokawa terminator orbit OpNav	2 hours, 10 days	✓	UKF	0.621	0.0160	✓	–
			UKF	0.746	0.0141	✓	–

terminator orbit, as each simulation has a similar number of total observations. The LS filter also produces similar errors with and without the first five observations when compared to the EKF and UKF. This implies that the LS filter requires less time to converge on a solution when initializing the filter.

Without process noise added, the LS filter does not achieve errors within the 3σ covariance bounds. Analyzing the results from the inner LS algorithm, the majority of the observations converge on a position state. The estimated position state is passed to the outer EKF to perform the full state update and covariance calculations. Without process noise added, the covariance in the outer EKF is smaller, and as a result the state update, $\hat{x}$, is smaller. Therefore, even if the inner LS algorithm converges on the truth state, the outer EKF is unable to appropriately update the state estimate, resulting in larger state errors. This implies that the LS filter performs better when process noise is included in the filter, which is typical in OD operations.

The larger number of altimetry measurements in each observation in the Bennu orbit leads to quicker convergence of the filters when compared to the Itokawa terminator orbit simulations. However, the filter simulations with the Bennu orbit takes two to three times longer to complete than the Itokawa orbit simulations.

Table 3.15 summarizes the filter results for the terminator orbits around Itokawa and Bennu and the Itokawa eccentric orbit when estimating the position, velocity, and pointing bias. The consistency of the errors refers to whether the filter errors for each component were within the 3σ covariance bounds for the majority of the observation arc.

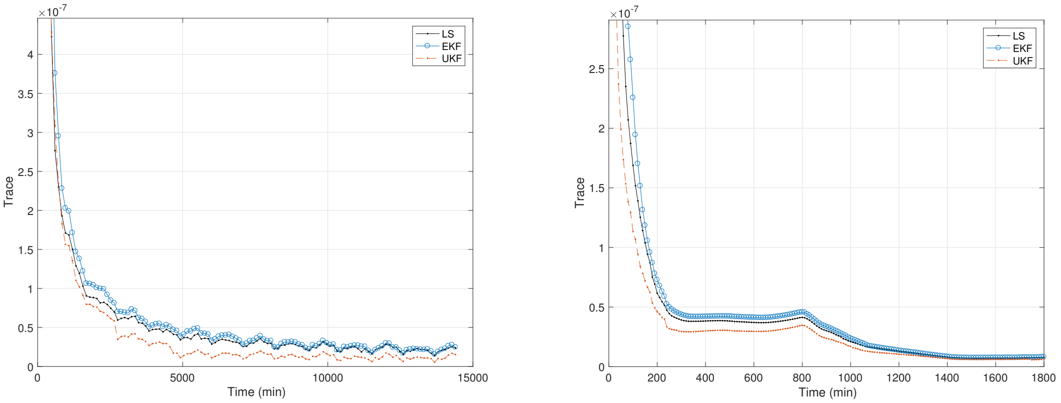
Table 3.15: Summary of RMS errors of the magnitude of the position and velocity errors and total pointing error angle, α over the full simulation, the consistency of the filter errors, and the run time of the filter. Presented for when estimating the position, velocity, and pointing bias.

Orbit	Obs Freq	PN added	Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	α (deg)	Consistent Errors?	Run Time (min)
Itokawa terminator orbit	2 hours, 10 days	✓	EKF	8.82	0.0482	0.44		3.5
		✓	UKF	2.61	0.0281	0.15		11.4
		✓	LS	0.243	0.0266	0.013	✓	6.7
			UKF	2.31	0.0278	0.13		12.9
			LS	0.234	0.0266	0.013	✓	9.4
	10 min, 30 hours	✓	EKF	14.30	0.0620	0.72		7.2
		✓	UKF	1.79	0.0174	0.099	✓	19.2
		✓	LS	0.408	0.0131	0.023	✓	13.0
Bennu terminator orbit	2 hours, 10 days	✓	EKF	48.68	0.359	2.76		8.2
		✓	UKF	2.20	0.0202	0.13		40.3
		✓	LS	0.193	0.0152	0.011	✓	21.3
			UKF	2.08	0.0197	0.13		44.6
			LS	0.159	0.0152	9.27E-3	✓	21.2
Itokawa eccentric orbit	10 min, 30 hours	✓	EKF	7.01	0.0643	0.48		6.5
		✓	UKF	1.63	0.0194	0.12		18.1
		✓	LS	0.131	0.0145	9.09E-3	✓	15.1

The LS filter outperforms the EKF and UKF in each of the terminator orbit simulations and in the eccentric orbit simulation. The LS filter errors are the smallest of the three filters, and the errors are within the 3σ covariance bounds. The UKF encounters observability issues with the in-track position and radial velocity directions in the Itokawa and Bennu terminator orbits with observation frequencies of two hours. Although, when the observation frequency is increased to every 10 minutes, the UKF errors fall within the 3σ covariance bounds. The higher observation frequency allows the pointing bias to be resolved quicker while the evolution of the dynamics is slower. This allows the pointing to be resolved before gaining more information on the position and velocity state.

3.7.1 Discussion on Filter Iterations

Figure 3.25 illustrates the trace of the covariance for the three filter simulations in the Itokawa terminator orbit with observations every 2 hours (Figure 3.25(a)) and the Itokawa eccentric orbit with observations every 10 minutes (Figure 3.25(b)). Minimal differences are seen between the resulting uncertainties in both orbits. The trace of the covariance for the UKF is the smallest across the simulations, however it follows the same trends as the traces of the covariances for the EKF and LS simulations. This shows that the primary difference between the filtering techniques is in the resulting state errors. The additional iterative procedure in the LS filter produces a more accurate state estimate, and the errors are within the resulting covariance bounds.



(a) Itokawa terminator orbit

(b) Itokawa eccentric orbit

Figure 3.25: Trace of the covariance for the three filter simulations in the Itokawa terminator orbit with observations every 2 hours and the Itokawa eccentric orbit with observations every 10 minutes.

The filter saturation with the EKF and UKF is due to the preciseness of the lidar measurements and the nonlinear measurement function. Solutions to this problem found in the literature suggest iterating the state update, including the second order effects of the measurement function, or artificially inflating the measurement uncertainty matrix.[33, 116] Following the procedure in Zanetti (2010) [116], if the *a priori* covariance, $\bar{P}$, mapped with the measurement partial derivatives in $\tilde{H}$, is much larger than the measurement error covariance, R, an underestimation of the *a posteriori* state covariance can occur. This implies that the estimated state has less uncertainty than it actually does. A mathematical form of this underestimation is expressed in Equation 3.10, where P_k^- is the *a priori* covariance and P_k^+ is the *a posteriori* covariance at time t_k. [116]

$$\begin{aligned} \texttt{if:} \quad & \tilde{H}_k P_k^- \tilde{H}_k^T \gg R_k \\ \texttt{then:} \quad & \tilde{H}_k P_k^+ \tilde{H}_k^T \simeq R_k \end{aligned} \tag{3.10}$$

The conditions in Equation 3.10 are satisfied in both of the EKF orbit scenarios, causing an underestimation of the state covariance and filter saturation. The proposed solution[116] is to inflate the measurement error matrix, R, systematically to prevent filter saturation.

Due to the multitude of lidar measurements at each observation time, the proposed solution in this work is to implement the iterative LS filter to solve for the position and pointing at each time step. Before moving to the next observation, the linear state update is iterated until convergence, similar to applying the Gauss-Newton method. This used the all of available information to produce a state update, as is suggested in Denham (1966)[33], without artificially adding uncertainty to the range measurements.

3.7.2 Discussion on Estimating the Pointing

When pointing is included in the estimation state, the LS filter produced the smallest errors, and errors within the resulting covariance bounds, consistently across the simulations. The LS and UKF generate comparable errors, yet the LS filter consistently runs faster. The UKF creates $2n+1$ computes states at each observation step. When pointing is included in the estimation state, the UKF creates 19 computed measurements at each observation step. The LS algorithm has a maximum iteration limit of ten, and computes a maximum of ten measurements per observation step. This number is less if the LS algorithm converged before this maximum iteration limit.

The UKF consistently produces errors outside the covariance bounds in the in-track position and radial velocity directions, for the terminator and eccentric orbit around Itokawa and the terminator orbit about Bennu. In the terminator orbit about Itokawa, these issues are resolved when the observation frequency was increased. This allowed the pointing to be resolved before the evolution of the dynamics was too great.

Chapter 4

Estimation Robustness

A degree of uncertainty still exists after the characterization phase of a small mission due to errors in the dynamical and measurement mathematical models, as well as in the observed measurements. An estimation filter must account for this uncertainty and be robust to errors between the truth system and the mathematical models.

This chapter investigates the limits of the estimation filters investigated in Chapter 3. Robustness studies include varying the errors in the initial state with a nominal uncertainty to investigate how the state errors compare to the resulting covariance bounds. The robustness of the UKF and the LS filter are investigated with larger than nominal initial state errors to study the upper limit of the initial filter state that would ensure convergence.

Robustness to measurement mismodeling involves applying shape modeling errors and random pointing errors. A low fidelity shape model of the asteroid Itokawa is employed in the filter and a high fidelity model is used to create the truth measurements. Using a lower fidelity shape model onboard decreases the processing power of the filter, yet introduces measurement model errors. In order to account for the measurement errors, investigations include increasing the measurement uncertainty matrix, and increasing the covariance by implementing a consider filter. Errors in the pointing involve a random pointing offset error at each observation time to emulate pointing jitter. The iterated LS filter is applied in this case to solve for the pointing jitter.

4.1 Robustness to Nominal Initial State Errors

Thee Monte Carlo simulations of 100 cases each were conducted with the EKF, UKF, and LS filter to investigate how the estimation state errors match to the filter covariance bounds. The filters estimated the spacecraft position, velocity, and pointing error (Equation 2.1) in the Itokawa terminator orbit with observations every two hours for ten days. It was assumed that the shape model was well known, and the dynamics, spin state and axis of Itokawa were known. The same measurements were used in each of the filter cases, and included white noise on each altimetry measurement in the image array based on Equation 3.1.

The initial position and velocity state were sampled from the uncertainty of the *a priori* covariance in Equation 3.9, and each initial pointing angle was set to zero. Process noise was added on the order of 1 E-9 m/s^2 from Equation 3.3.

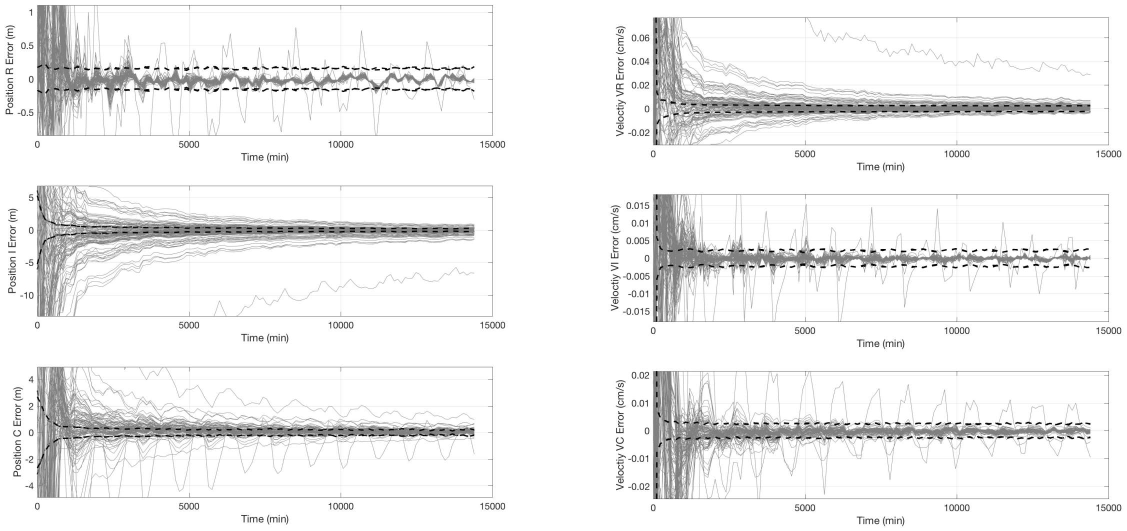
$$\begin{gathered}\bar{P}_0 = \texttt{diag}[\sigma_{\mathbf{r}}^2,\ \sigma_{\mathbf{v}}^2,\ \sigma_{\underline{\theta}}^2] \\ \sigma_{\mathbf{r}} = 10\ \text{m},\ \sigma_{\mathbf{v}} = 1\ \text{mm/s},\ \sigma_{\theta} = 2^\circ\end{gathered} \tag{3.9}$$

$$\begin{gathered}Q = \texttt{diag}[\sigma_Q^2,\ \sigma_Q^2,\ \sigma_Q^2] \\ \sigma_Q = 1\text{E} - 9\ \text{m/s}^2\end{gathered} \tag{3.3}$$

The Monte Carlo filter results for the EKF are presented in Figure 4.1 for the position, velocity and pointing errors. In the EKF, all of the filter runs converge toward the truth values. Once the filter has processed enough observations, the majority of the runs contain errors that fall within the 3σ bounds for the radial position, and the in-track and cross-track velocity components. It is shown that the covariance for each component reduces quickly at the beginning of the simulation, below the errors. This filter saturation prevents the state estimate from larger updates towards convergence.

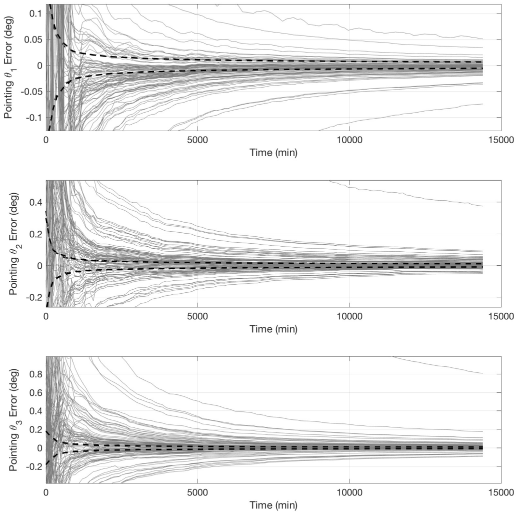
The position, velocity, and pointing errors and 3σ covariance bounds with the UKF are presented in Figure 4.2. Overall, each of the component errors from the UKF followed a similar pattern. The radial position and in-track velocity remained inside of their respective covariance bounds, while the other component errors reached a steady state either just above or below their covariance bounds.

The LS filter results of state errors and their 3σ covariance bounds are shown in Figure 4.3.



(a) Position

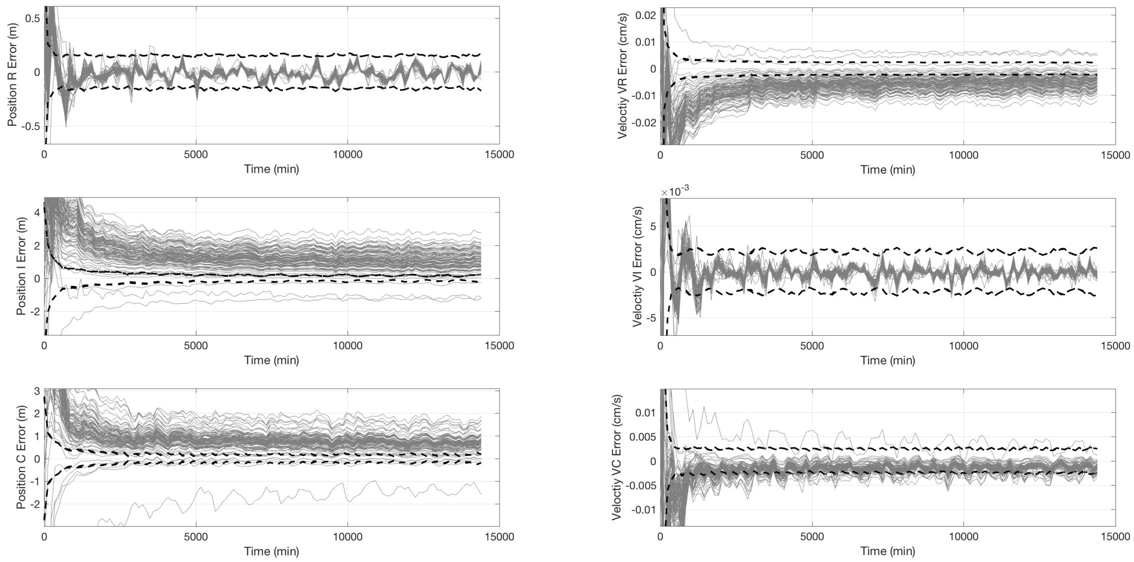
(b) Velocity



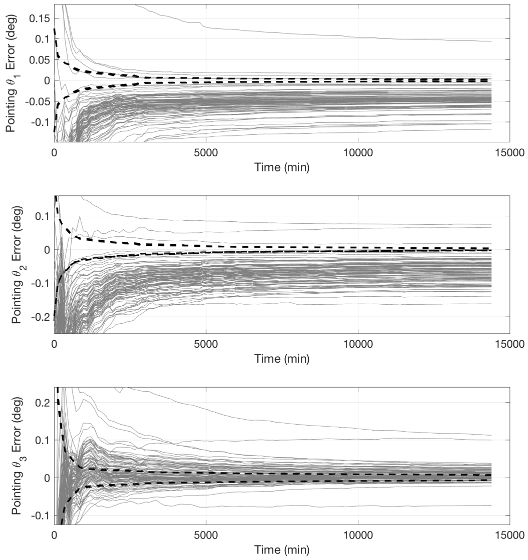
(c) Pointing

Figure 4.1: Monte Carlo errors (solid gray) and 3σ covariance bounds (dashed black) in the RIC frame with the EKF for the Itokawa terminator orbit with measurements every 2 hours for 10 days.

Once settled, each component error is almost identical through the end of the filter, and each one falls within their respective 3σ covariance bounds. This implies that the LS filter is very predictable



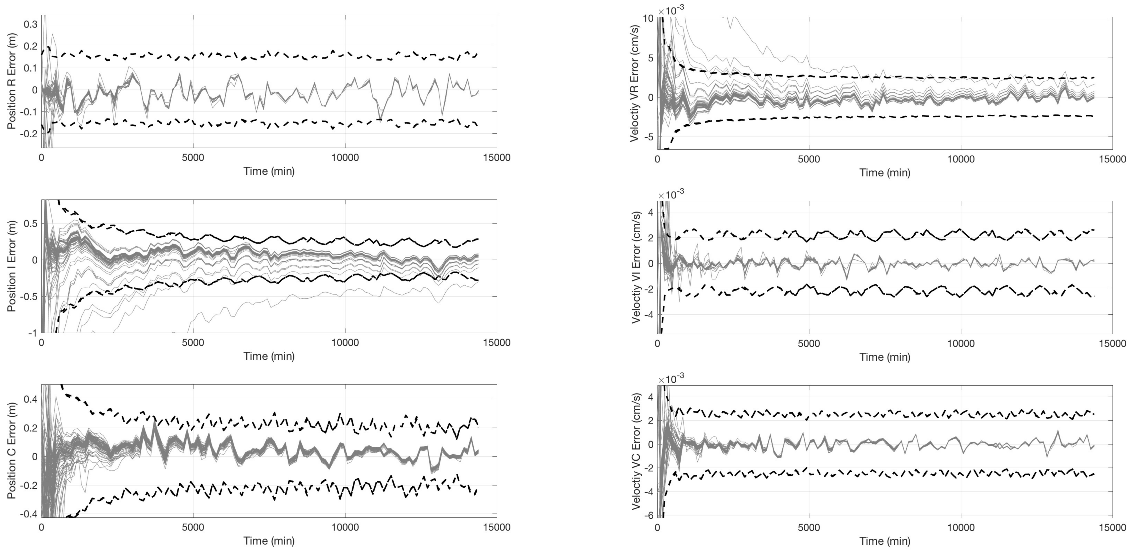
(a) Position (b) Velocity



(c) Pointing

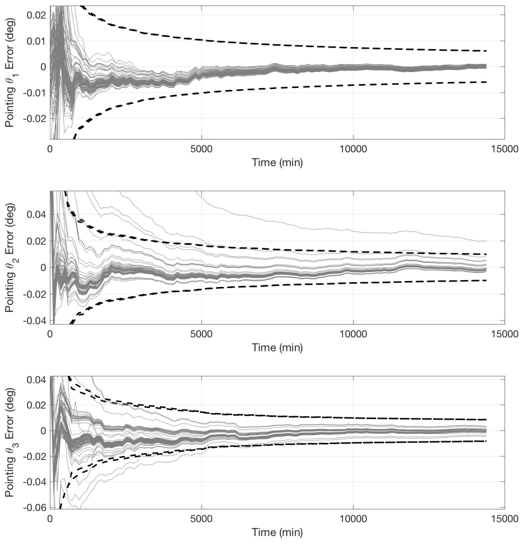
Figure 4.2: Monte Carlo errors (solid gray) and 3σ covariance bounds (dashed black) in the RIC frame with the UKF for the Itokawa terminator orbit with measurements every 2 hours for 10 days.

and behaves almost identically for the various initial state errors. Due to the iterative nature of the LS filter, the algorithm was able to converge on the same or similar state with the same observations



(a) Position

(b) Velocity



(c) Pointing

Figure 4.3: Monte Carlo errors (solid gray) and 3σ covariance bounds (dashed black) in the RIC frame with the LS filter for the Itokawa terminator orbit with measurements every 2 hours for 10 days.

in each of the filter runs.

Table 4.1 presents the RMS of the magnitude of the position and velocity errors taken at the end of each filter simulation. The RMS of the total pointing error from truth, α, evaluated at the end of each filter simulations is also presented, along with the average run time of the filters. The errors from the LS filter are at least an order of magnitude smaller than those from the EKF and UKF in position, velocity and pointing. The EKF errors were smaller than the UKF errors. This was most likely due to the EKF continuously converging in its filter runs, while the UKF reached a steady state in the majority of its filter runs.

Table 4.1: RMS errors of the ending magnitude of the position and velocity errors and total pointing error angle, α, and the average run time of the filter for the filter Monte Carlo simulation. This is in the Itokawa terminator orbit, estimating a pointing bias, with measurements every 2 hours for 10 days and with process noise added.

Filter	$\lvert\mathbf{e}_{\mathbf{r}_f}\rvert$ (m)	$\lvert e_{\mathbf{v}_f}\rvert$ (cm/s)	α_f (deg)	Average Run Time (min)
EKF	0.831	3.66E-3	0.048	3.8
UKF	1.49	5.42E-3	0.088	12.3
LS	0.0729	2.70E-4	2.83E-3	7.6

The iterations employed in the LS filter allowed the filter to converge on the correct state within the resulting covariance bounds for each state component and for each Monte Carlo run except one. The large number of altimetry measurements processed at one time step, in combination with their preciseness leads to a sharp decrease in the filter uncertainty. This causes filter saturation, and a sequential filter does not produce large enough updates to the estimation state because the state uncertainty is low. The iterations within the LS algorithm apply multiple linear updates to converge the position and pointing state at each time step. The result is a state estimate with errors from the truth that are within the 3σ filter covariance bounds.

4.2 Initial State Error Robustness with the Unscented Kalman Filter

The UKF creates $2n + 1$ sigma points at each observation time and propagates those states through the nonlinear dynamics. After the propagation, the filter computes flash lidar images for each of these sigma points, and compares those images to the observed image. Since each pixel in the flash lidar image is treated as an individual measurement, a particular pixel must contain a range-return in the observed image and each of the computed images in order for a comparison to be made. This requires that an overlap is present between the observed image and each sigma point computed image.

Monte Carlo simulations were conducted to investigate the limitations of this pixel-matching measurement technique. Three Monte Carlo (MC) simulations of 50 cases each studied the robustness to initial filter state errors and the sensitivity to the value of α in the UKF. These simulations estimated the position and velocity of the spacecraft in the circular terminator orbit about Itokawa, while varying the UKF α parameter in Equation 2.89, and the initial state errors.

The values of α chosen were $\alpha = 1,\ 0.6$, and 0.4. Nominally, a value of $\alpha = 1$ was used in the UKF simulations. The lower values of α reduced the spread of the UKF sigma points from the mean, and therefore effected the overlap of the observed image and the sigma point computed images.

The *a priori* filter state for each MC case was randomly sampled from a normal distribution based on the *a priori* covariance in Equation 4.1. The state was normally varied from the truth with $1\sigma = 100$ m in position and $1\sigma = 10$ mm/s in velocity.

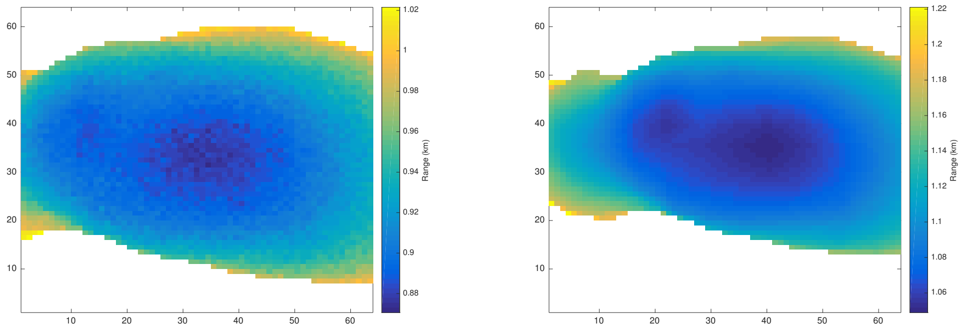
$$\begin{gathered} \bar{P}_0 = \texttt{diag}[\sigma_{\mathbf{r}}^2,\ \sigma_{\mathbf{v}}^2] \\ \sigma_R = 100 \text{ m},\ \sigma_V = 10 \text{ mm/s} \end{gathered} \tag{4.1}$$

This variation in the filter state was larger than in the nominal Itokawa terminator case in Sections 3.3 and 4.1 in order to further test the filter robustness. For the three MC simulations, only the value of α changed, and the same *a priori* state was used for each corresponding MC case.

4.2.1 Measurement Overlaps

The flash lidar measurements were processed such that each pixel in the image was treated as an independent observation. If one pixel did not contain a range-return in either the observed image or any of the sigma point computed images, that pixel could not be used. This required at least one range-return pixel overlap between the observed image and each of the sigma point computed images in order to process that observation.

To further explain this overlap, the observed and sigma point images from MC case 17 are presented in Figures 4.4 and 4.5. Figure 4.4(a) shows the observed image at the first time step, and Figure 4.4(b) shows the computed image from the *a priori* state, $\bar{X}_0$.



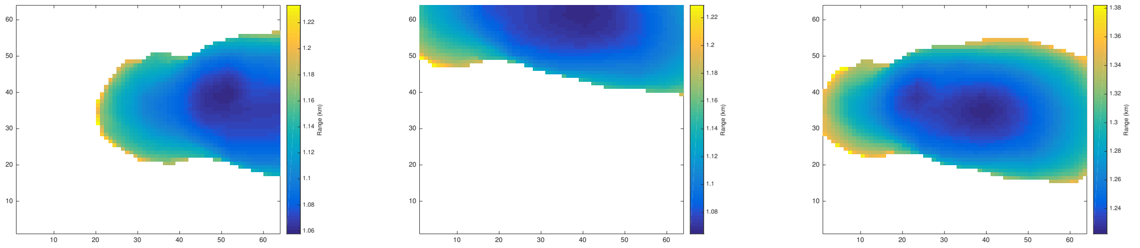
(a) Observed Measurement

(b) $\bar{X}_0$ Computed Measurement

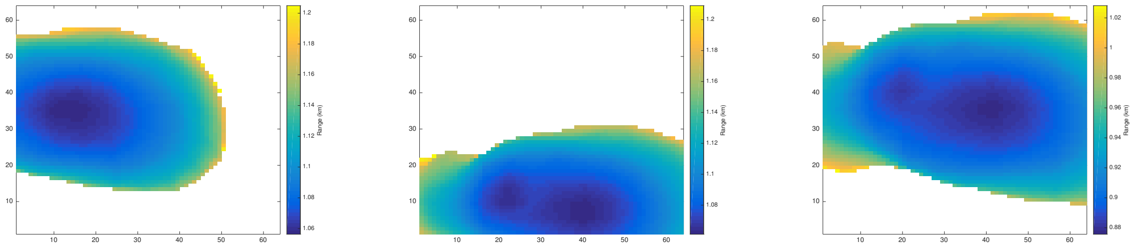
Figure 4.4: Observed image and computed image from the *a priori* state, $\bar{X}_0$ at $t = 0$ of MC case 17.

Figure 4.5 illustrates the computed images from the sigma points that were perturbed in position for MC case 17 with $\alpha = 1$. Since a flash lidar measurement only depends on the position of the spacecraft, the sigma points that were perturbed in velocity produced the same image as the *a priori* state, $\bar{X}_0$, in Figure 4.4(b). When the observed image (Figure 4.4(a)), the $\bar{X}_0$ computed image (Figure 4.4(b)), and each of the sigma point computed images (Figure 4.5) were superimposed on top of one another to compare corresponding pixels, the result was a blank image. With $\alpha = 1$ in this MC case, no individual pixel observed a range in the all of the images, and this observation at $t = 0$ was unusable.

When the value of α was reduced to 0.6 and 0.4 for MC case 17, an overlap did occur. Figures 4.6(a) and 4.6(b) show the overlap of the observed image and each sigma point computed image for $\alpha = 0.6$ and $\alpha = 0.4$, respectively, at $t = 0$. For these α values, the observed image and $\bar{X}_0$ computed image were the same as those in Figures 4.4(a) and 4.4(b), however the perturbed sigma

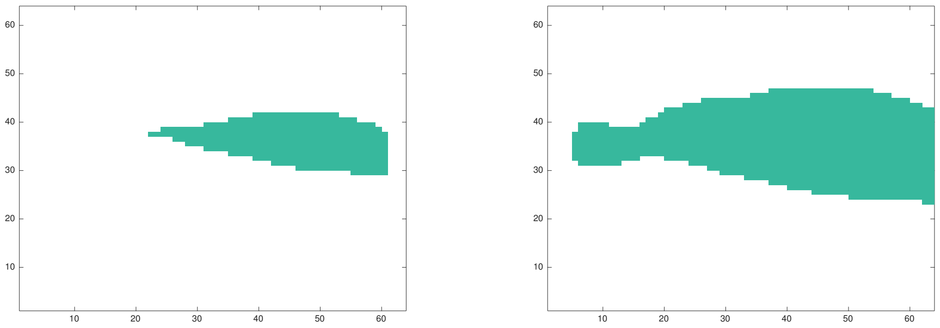


(a) $+x$ $\chi-$point (b) $+y$ $\chi-$point (c) $+z$ $\chi-$point



(d) $-x$ $\chi-$point (e) $-y$ $\chi-$point (f) $-z$ $\chi-$point

Figure 4.5: Computed images from the perturbed sigma points ($\chi-$points) in position for MC run 17 with $\alpha = 1$ at $t = 0$.



(a) $\alpha = 0.6$ (b) $\alpha = 0.4$

Figure 4.6: Overlap of the observed image and each sigma point computed image for MC case 17 at $t = 0$ with α values of $\alpha = 0.6$ and $\alpha = 0.4$.

points were closer to the *a priori* state, $\bar{X}_0$, and created an overlap in their computed images. The smaller the α value, the smaller the spread of the perturbed sigma points from $\bar{X}_0$, and the larger the overlap in images, as evidenced by the larger area in Figure 4.6(b) as opposed to Figure 4.6(a).

4.2.2 Robustness to Initial State Errors

The MC simulations involved the same nominal Itokawa terminator orbit (Section 3.1), and a similar filter set up as the OD simulation (Section 3.3). Dissimilar to the OD simulation in Section 3.3, the observation frequency was every 5 minutes for an observation arc of half a day. The *a priori* covariance was that in Equation 4.1, and, as explained above, initial state errors were applied based on this covariance. Process noise was added uniformly to each axis with a magnitude of 1 E-9 m/s^2 (Equation 3.3) to slightly inflate the covariance. The sensor was set up as explained in Section 3.2.1, with the exception that the upper limit of 1 km on the flash lidar ranges was removed. It was assumed that if the spacecraft could see the asteroid, it was able to take flash lidar measurements of the surface.

4.2.2.1 Usability of Observations

Within the observation arc, the usability of an observation at any time step depended on its state estimate, $\hat{X}_t$, covariance, P_t, and value of α. It was found that for particular *a priori* states, the filter was either able to process all of the observations, some of the observations, none of the observations, or did not finish. If an observation was unable to be processed, this was because an overlap was not present between the observed and computed images. The simulations that were able to process some of the observations had one or multiple observation arcs with unusable measurements.

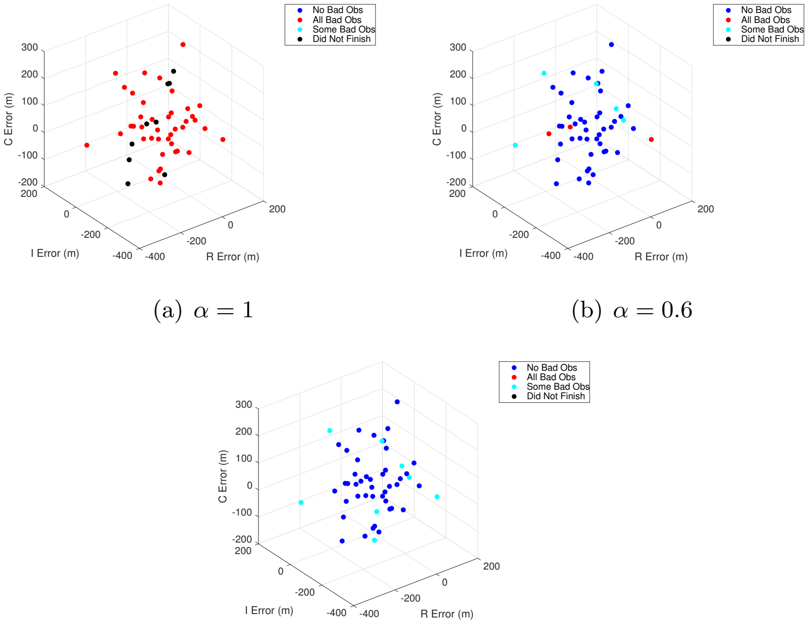
The simulations that did not finish were not able to process any observations, and the integration within the filter was terminated early. In these simulations, since no observations were able to be processed, the state covariance continued to grow and was never reduced with a state estimate update. This caused the position of the sigma points to grow farther away from the mean. Eventually, one of the perturbed sigma points was placed within the asteroid, and resulted in early termination of the integration. Within a UKF, the calculation of all of the sigma points is required to determine a state estimate update; therefore, without the use of one of the sigma points, the filter was not able to produce a state update and was terminated.

Figure 4.7 illustrates the errors in the *a priori* state for the 50 cases in each of the three MC simulations. The colors correspond to whether the filter was able to process all of the observations (*No Bad Obs*, blue), some of the observations (*Some Bad Obs*, cyan), none of the observations (*All Bad Obs*, red), or did not finish (*Did Not Finish*, black).

With $\alpha = 1$ (Figure 4.7(a)), every case contained either all bad observations, or did not finish. From the MC cases presented, only the $\alpha = 1$ simulation contained cases that did not finish. When the α value was reduced to 0.6 (Figure 4.7(c)), the majority of the cases contained no bad or some bad observations, with only three cases that contained all bad observations. With $\alpha = 0.4$ (Figure 4.7(b)), all of the cases were able to process some or all of the observations. The cases that involved some bad observations with $\alpha = 0.6$, also involved some bad observations with $\alpha = 0.4$, placing significance on the initial state errors.

Further analysis is shown in Figure 4.8 that illustrates the initial position errors in the in-track (I) and cross-track (C) frame for each of the three α values. The initial errors in the radial direction were not presented because they did not show a correlation with the usability of the observations. When an *a priori* state error was present in the radial direction, the asteroid was still centered in the image frame and a measurement overlap was still able to occur. Comparing the $\alpha = 0.6$ and $\alpha = 0.4$ simulations (Figures 4.8(b) and 4.8(c), respectively), one can see that the cases that resulted in all bad or some bad observations were correlated with larger initial errors in the in-track direction rather than the cross-track direction. These cases involved initial in-track state errors that were greater than ±100 m.

The correlation of in-track and cross-track errors was due to the orientation of the terminator orbit with respect to Itokawa. The orbit in-track direction was along the short axis of Itokawa, while the cross-track direction was along the long axis. Therefore, an initial state perturbation along the in-track direction of this orbit viewed less of Itokawa since it was perturbed along its shorter axis, and created less of of an image overlap, if any. On the other hand, a perturbation in the cross-track direction was perturbed along the long axis of Itokawa, and created a larger image

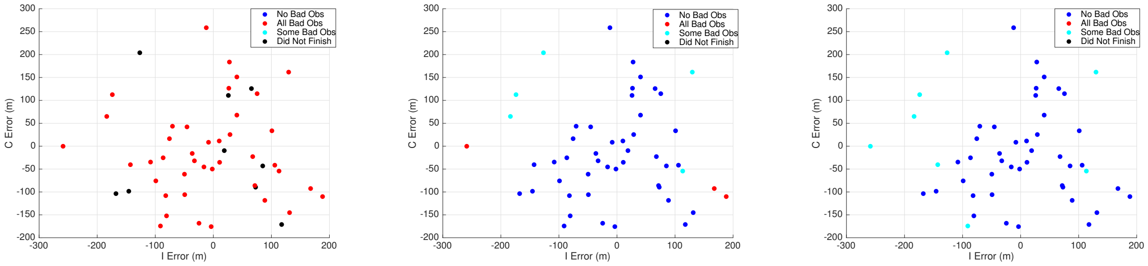


(a) $\alpha = 1$

(b) $\alpha = 0.6$

(c) $\alpha = 0.4$

Figure 4.7: Position errors in the *a priori* state in the RIC frame for the MC simulations with varying α values, and the usability of each case: no bad observations (blue), all bad observations (red), some bad observations (cyan), or did not finish (black).



(a) $\alpha = 1$

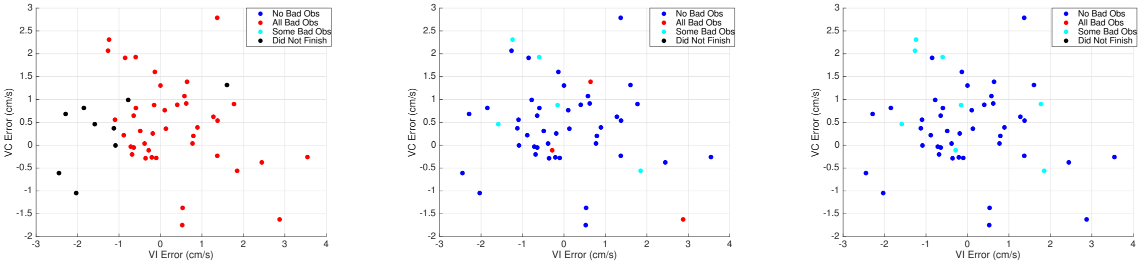
(b) $\alpha = 0.6$

(c) $\alpha = 0.4$

Figure 4.8: Position errors in the *a priori* state in the IC frame for the MC simulations with varying α values, and the usability of each case: no bad observations (blue), all bad observations (red), some bad observations (cyan), or did not finish (black).

overlap.

The initial velocity errors in the in-track (I) and cross-track (C) frame are shown in Figure 4.9 for each of the three α values. Unlike the initial position errors, less of a correlation is seen between the initial velocity errors and the usability of the observations. However, in Figure 4.9(a), the cases that did not finish were more prevalent in the negative in-track direction, which means the initial state was slower in this direction than the truth state. The cases that did not finish involved a perturbed sigma point that was inside the asteroid before failing. This correlation implies that the initial velocity was slower in the in-track direction, causing the estimated state to travel too close to the asteroid surface, and resulting in early termination.



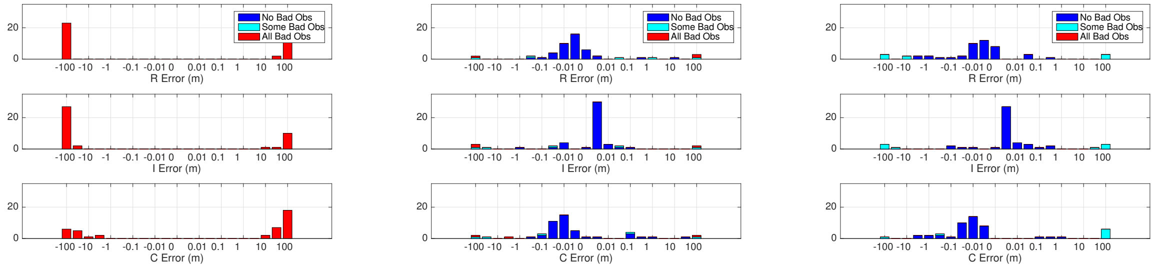
(a) $\alpha = 1$ (b) $\alpha = 0.6$ (c) $\alpha = 0.4$

Figure 4.9: Velocity errors in the *a priori* state in the IC frame for the MC simulations with varying α values, and the usability of each case: no bad observations (blue), all bad observations (red), some bad observations (cyan), or did not finish (black).

4.2.2.2 Resulting Final Errors

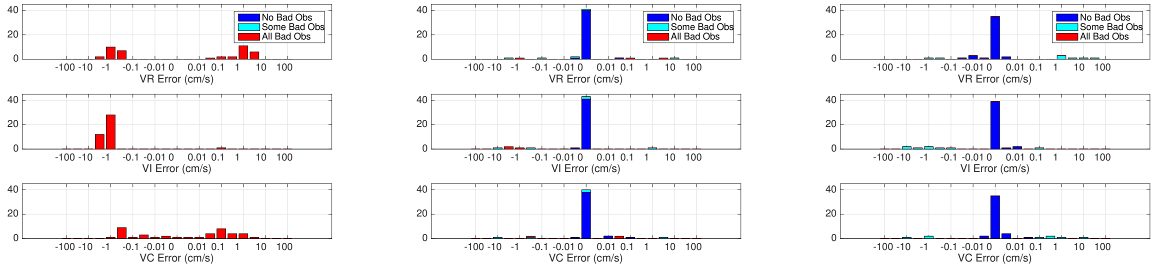
Figure 4.10 shows the histograms of the position errors at the final time, and Figure 4.11 illustrates the histograms for the velocity errors at the final time for the cases that finished in the three MC simulations. Both of the histograms are presented on a log-scale, with the position in meters and the velocity in cm/s.

In the simulations for $\alpha = 0.6$ and $\alpha = 0.4$, the majority of the cases that contained no bad observations resulted in position errors within centimeter accuracy, with the maximum error reaching up to 10 m. The resulting velocity errors were below 0.01 cm/s, with the maximum error



(a) $\alpha = 1$ (b) $\alpha = 0.6$ (c) $\alpha = 0.4$

Figure 4.10: Histograms of the position errors at the final time in each RIC direction for the MC simulations with varying α values, and the usability of the cases.



(a) $\alpha = 1$ (b) $\alpha = 0.6$ (c) $\alpha = 0.4$

Figure 4.11: Histograms of the velocity errors at the final time in each RIC direction for the MC simulations with varying α values, and the usability of the cases.

reaching 0.1 cm/s. For those cases with all bad observations (in $\alpha = 1$ and $\alpha = 0.6$), the filter simply propagated the estimated state for 0.5 days.

4.2.3 Effects of Measurement Processing

With this current methodology, if the initial spacecraft position was within 100 m of the truth, and $\alpha = 0.4$, the filter converged on the truth within meter-level accuracy. However, if the α value was increased to 1, measurement overlaps did not occur, and the filter was not able to converge. This procedure illustrates that even with 100 m of deviation in the initial position, the value of α can be adjusted in the UKF in order to encourage convergence. This upper limit on the initial position and velocity state would be sufficient for routine spacecraft operations.

This investigation also addressed the question of how to process these three-dimensional images within a filter. The pixel-matching measurement approach required an overlap in the observed image and each sigma point computed image, and can exclude useful information from the filter. For example, in Figure 4.5, each $\chi-$points can view the asteroid, but since there is not an overlap of all of the measurements, the filter cannot process the observation.

4.3 Initial State Error Robustness with the Iterative Least Squares Algorithm

In the LS algorithm, the estimated state was iterated until a convergence criteria was met or the maximum number of iterations was reached. The LS algorithm does not use *a priori* information from the filter, and only uses the observed image and estimated state. Three Monte Carlo simulations studied the limit on the initial position and pointing state of the spacecraft in order to ensure convergence with one observation. Each simulation was conducted with a single state, estimated the position and pointing of the spacecraft at one time step, and no estimation filtering was performed. The truth state contained errors in all six state components of the position and pointing angles based on a normal distribution with specified one-sigma values.

One simulation sampled the position errors from a distribution of $1\sigma_{\mathbf{r}} = 50$ m and the pointing errors from a distribution of $1\sigma_{\theta_n} = 0.5^\circ$, a second sampled from a distribution of $1\sigma_{\mathbf{r}} = 10$ m in position and $1\sigma_{\theta_n} = 3^\circ$ in pointing, and a third sampled errors from a distribution with $1\sigma_{\mathbf{r}} = 100$ m in position and $1\sigma_{\theta_n} = 0.5^\circ$ in pointing. A fourth simulation held the pointing state constant at its truth value, and varied the position state with $1\sigma_{\mathbf{r}} = 100$ m.

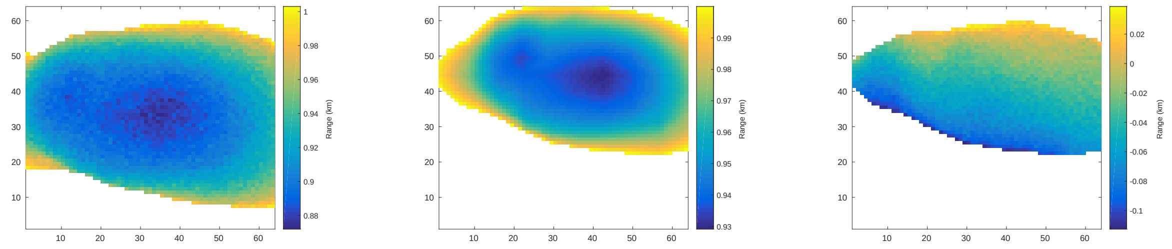
One hundred variations of the initial state were passed into the least-squares algorithm and each were iterated until the convergence criteria was met. The difference in this MC simulation as opposed to the filter formulation was that the maximum number of iterations was increased to 40. The shape model of Itokawa was assumed to be known, as well as the asteroid's attitude, and no dynamics were present in these simulations. The errors in each simulation involved errors in the initial state and white noise on the returned altimetry measurements based on Equation 3.1.

4.3.1 Moderate Position and Pointing Uncertainty

One hundred initial state errors were sampled from a normal distribution with $1\sigma_{\mathbf{r}} = 50$ m in position and $1\sigma_{\theta_n} = 0.5°$ pointing:

$$1\sigma_{\mathbf{r}} = 50 \text{ m}, \; 1\sigma_{\theta_n} = 0.5° \tag{4.2}$$

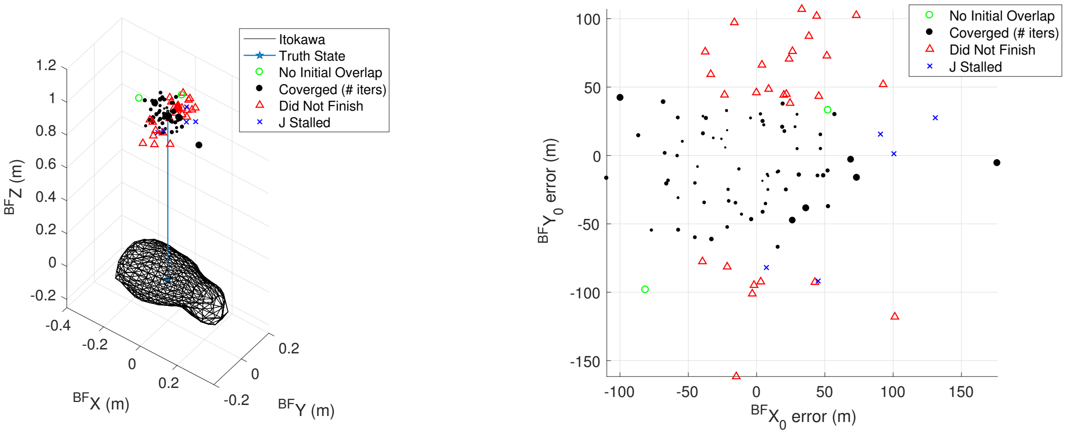
The truth state was chosen as pointed radially above the north pole of Itokawa in the circular terminator orbit. The observed measurement from the truth state is shown in Figure 4.12(a). For illustration purposes, Figure 4.12(b) shows the computed measurement image when the one-sigma values in Equation 4.2 were added to the truth state. Figure 4.12(c) shows the difference of the computed image subtracted from the observed truth image. The pixels in Figure 4.12(c) represent the useable data points for comparing these two images.



(a) Truth observation image (b) Perturbed observation image (c) Difference of (truth-perturbed)

Figure 4.12: Example observation images of the observed and perturbed flash lidar images for the LS MC.

Figure 4.13 illustrates the spread of the position initial conditions with respect to Itokawa (Figure 4.13(a)), the initial errors in the XY frame (Figure 4.13(b)), and the outcome of each run. The open green circle markers indicate there was not an initial overlap between the observed and computed image. The solid circle markers indicate that the run converged, and the larger the circle, the more iterations it took to converge. The triangle markers indicate that while an initial overlap occurred in the filter, the subsequent iterations did not produce an overlap and the algorithm was terminated. The cross markers indicate the cost function, J, had stalled out without dropping below the tolerance and was terminated at the maximum number of iterations.



(a) Initial perturbed states with respect to Itokawa (b) Initial Error in XY frame

Figure 4.13: Initial position errors and resulting LS MC outcomes for $1\sigma_{\mathbf{r}} = 50$ m, $1\sigma_{\theta_n} = 0.5°$.

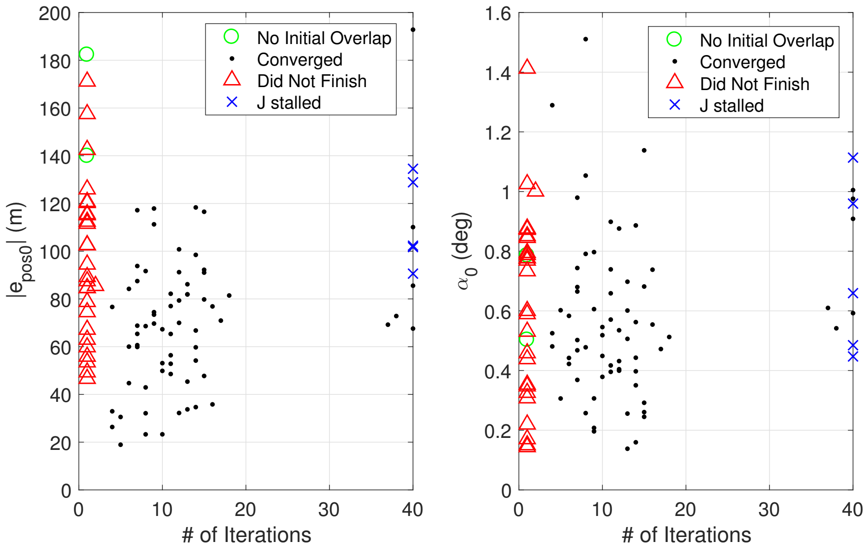


Figure 4.14: Magnitude of the initial position and pointing errors and their resulting LS MC outcomes for $1\sigma_{\mathbf{r}} = 50$ m, $1\sigma_{\theta_n} = 0.5°$.

Figure 4.14 presents the magnitude of the initial error versus the number of iterations. The magnitude of the initial errors was defined as the magnitude of the error in position, $|\mathbf{e_r}|$, and the total pointing offset angle from the truth, α. Table 4.2 presents the RMS of the final position error magnitude and total pointing angle offset for those cases that converged and did not reach the maximum number of iterations. Out of the cases that converged, the averaged ending position

Table 4.2: RMS of ending errors of the LS MC cases that converged for $1\sigma_{\mathbf{r}} = 50$ m, $1\sigma_{\theta_n} = 0.5°$

State	Ending Error
$\texttt{rms}(\lvert\mathbf{e}_{\mathbf{r}_f}\rvert)$	0.816 m
$\texttt{rms}(\alpha_f)$	0.866°

error magnitude was 0.8 m, and the pointing offset angle was 0.9°.

In this simulation, the position of the spacecraft played a larger role than the pointing in determining the outcome of the runs. One can notice a pattern in Figure 4.13(b). Runs with a position error greater than ±50 m in the y-direction did not produce an overlap of the observed and computed images. Studying the orientation of Itokawa at this time (Figure 4.13(a)), this is likely due to the short width of Itokawa in the y-direction. When analyzing the initial pointing angle errors and MC run outcomes, no reasonable pattern could be discerned.

4.3.2 Large Pointing Uncertainty

A similar Monte Carlo simulation varied the position with $1\sigma_{\mathbf{r}} = 10$ m, and the pointing with $1\sigma_{\theta_n} = 3°$.

$$1\sigma_{\mathbf{r}} = 10 \text{ m},\ 1\sigma_{\theta_n} = 3° \tag{4.3}$$

This position uncertainty was the nominal uncertainty on the OD simulations in Chapter 3. An uncertainty of 3° on each of the Euler angles in the pointing offset definition is equivalent to a total pointing offset of 4.24° or 74 mrad. This is a highly conservative pointing case, as most spacecraft pointing determination systems can be resolved to a few milliradians.

Figure 4.15 presents the magnitude of the initial errors versus the number of iterations for convergence, where $\lvert\mathbf{e}_{\mathbf{r}_0}\rvert$ is the magnitude of the initial position error, and α_0 is the total initial pointing offset from the truth. The open circles indicate there was not an initial overlap between the observed and computed image (which did not occur in this similation), the triangle markers indicate the case was terminated early, and the cross markers indicate that the cost function, J stalled out. Table 4.3 presents the RMS of the magnitude of the ending errors over the cases that converged and did not reach the maximum number of iterations. Figure 4.16 presents the

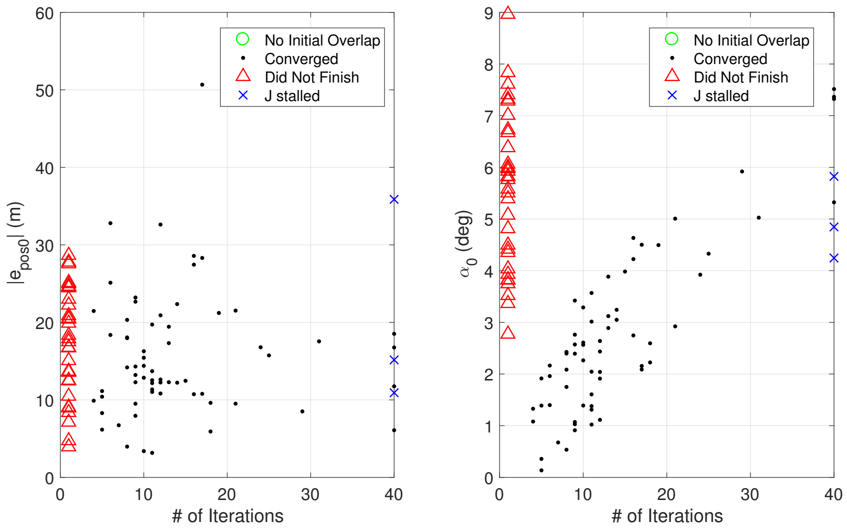
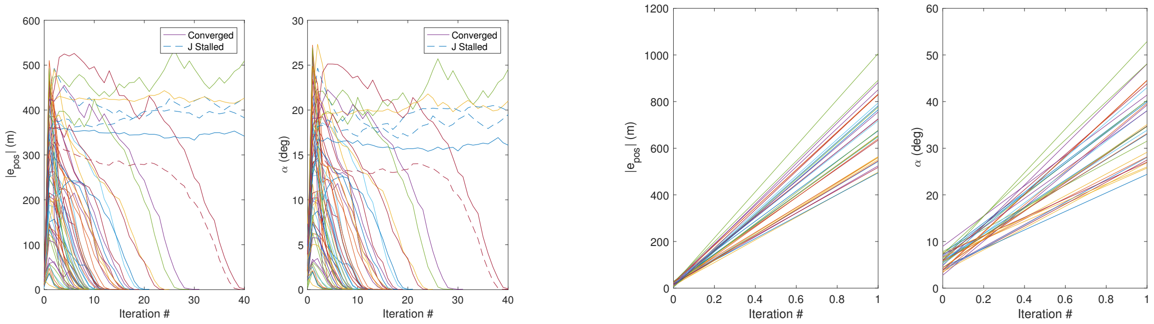


Figure 4.15: Magnitude of the initial position and pointing errors and their resulting LS MC outcomes for $1\sigma_{\mathbf{r}} = 10$ m, $1\sigma_{\theta_n} = 3^\circ$.

progression of the convergence for the cases that did finish (a), and those that did not finish (b). This is illustrated as the magnitude of the error from the truth ($|\mathbf{e_r}|$ for position and α for pointing) versus the iteration number.

A distinct positive trend is present in Figure 4.15 between the initial pointing error and the number of iterations for the cases that converged. The average errors for the cases that did converge resulted in a position error less than 1 m and a total offset in pointing error less than 1°.



(a) Cases that did finish

(b) Cases that did not finish

Figure 4.16: Convergence of the LS MC cases for $1\sigma_{\mathbf{r}} = 10$ m, $1\sigma_{\theta_n} = 3^\circ$.

Table 4.3: RMS of ending errors of the LS MC cases that converged for $1\sigma_{\mathbf{r}} = 10$ m, $1\sigma_{\theta_n} = 3^\circ$

State	Ending Error
$\texttt{rms}(\lvert\mathbf{e}_{\mathbf{r}_f}\rvert)$	0.924 m
$\texttt{rms}(\alpha_f)$	0.97°

The cases that did not finish all contained an initial pointing offset error larger than 3°. For the cases that did not finish in Figure 4.16(a), the first iteration reached a pointing offset error greater than 20°, and a position errors of greater than 500 m. This aligns with the sensor model because the full FOV of the flash lidar instrument was 20°. The convergence progression for those cases that did finish (Figure 4.16(b)) all increased in error with the first iteration of the LS algorithm. Yet, the first iteration errors for these cases were below 500 m in position and below 25° in pointing.

4.3.3 Large Position Uncertainty

A third Monte Carlo simulation varied the position with $1\sigma = 100$ m, and the pointing with $1\sigma = 0.5^\circ$. This is a large variation in the position, and a nominal variation in the pointing.

$$1\sigma_{\mathbf{r}} = 100 \text{ m},\ 1\sigma_{\theta_n} = 0.5^\circ \tag{4.4}$$

The truth state for this simulation was toward the $+x$ body-fixed axis of Itokawa, and pointing still directed radially at Itokawa. Figure 4.17 illustrates the truth state and the spread of the initial position states with their MC outcomes.

Figure 4.18 illustrates the magnitude of the initial position error and total initial pointing offset from truth versus the number of iterations toward convergence. The open circles represent the cases that did not contain an initial overlap in the observed and computed images, and the triangles represent the cases that contained an initial overlap but were terminated within the LS algorithm. Table 4.4 presents the RMS of the magnitude of the final errors for those cases that converged and did not reach the maximum number of iterations.

The cases that did not finish, contained over 100 m in initial position error. For the cases

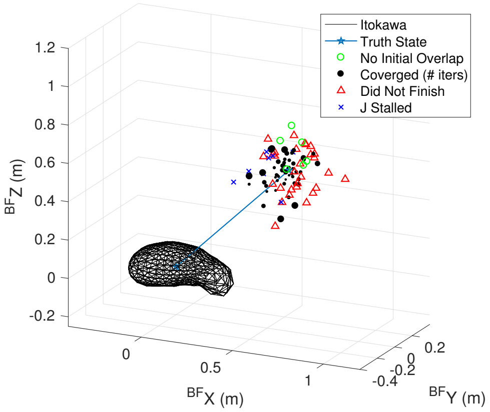


Figure 4.17: Initial position errors and resulting LS MC outcomes for $1\sigma_{\mathbf{r}} = 100$ m, $1\sigma_{\theta_n} = 0.5°$.

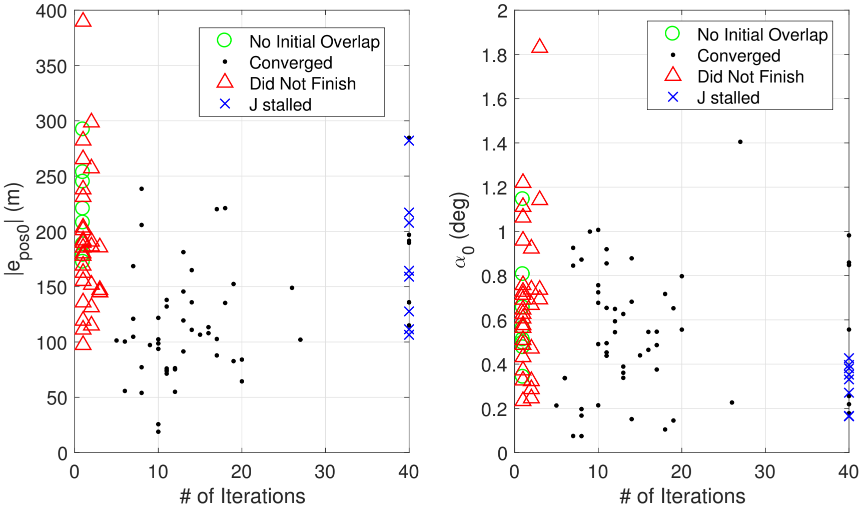


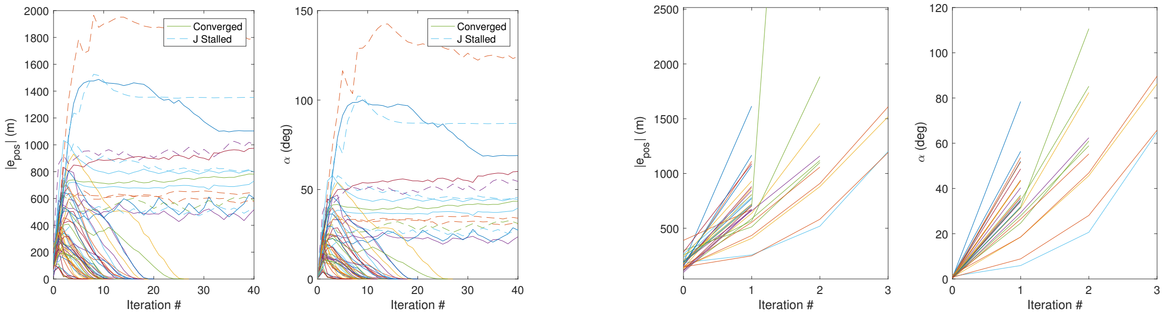
Figure 4.18: Magnitude of the initial position and pointing errors and their resulting LS MC outcomes for $1\sigma_{\mathbf{r}} = 100$ m, $1\sigma_{\theta_n} = 0.5°$.

that did converge, the ending position magnitude error was 0.5 m in position and 0.5° in pointing.

Figure 4.19 illustrates the progression of the convergence for the cases that did finish (a), and those that did not (b). For those cases that did not finish, the errors within the iteration grew to over 500 m in position and over 30° in pointing error. When this occurred, an overlap between the observed and computed measurement was no longer available and the algorithm was terminated. However, these error limits were not strictly tied to the convergence outcome of the case. Errors

Table 4.4: RMS of ending errors of the LS MC cases that converged for $1\sigma_{\mathbf{r}} = 100$ m, $1\sigma_{\theta_n} = 0.5°$

State	Ending Error
$\texttt{rms}(\lvert\mathbf{e}_{\mathbf{r}_f}\rvert)$	0.539 m
$\texttt{rms}(\alpha_f)$	0.50°



(a) Cases that did finish

(b) Cases that did not finish

Figure 4.19: Convergence of the LS MC cases for $1\sigma_{\mathbf{r}} = 100$ m, $1\sigma_{\theta_n} = 0.5°$.

reached beyond these limits to up to 800 m in position and 50° in pointing in some of the cases that were able to converge.

When estimating the position and pointing, the LS algorithm has two states to adjust in order to minimize the cost function. As shown, the LS algorithm adjusts the pointing state farther than necessary and can cause the asteroid to be out of view of the sensor. This occurs when the position error is large and the pointing is included in the estimation state,.

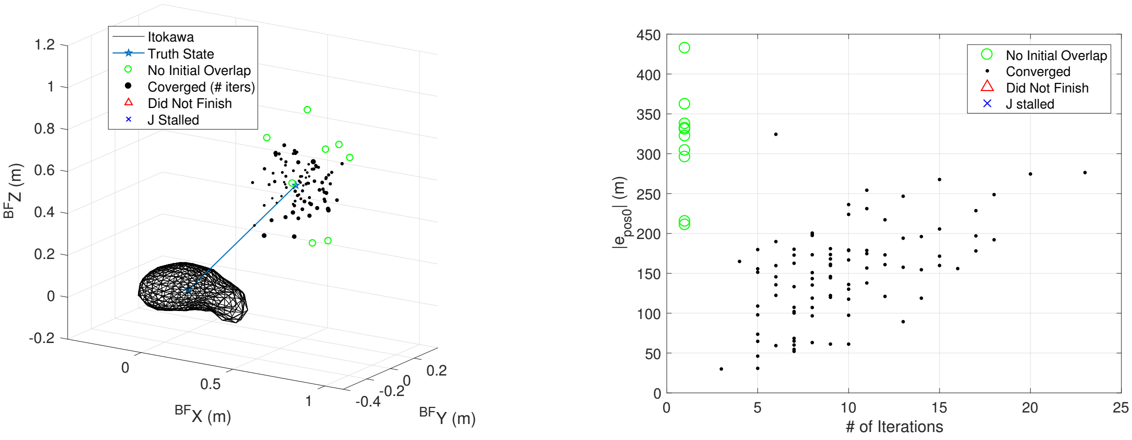
4.3.4 Position Only Iterations

This Monte Carlo simulation only varied the position and held the pointing constant. The position was varied in each component by sampling errors from a normal distribution with $1\sigma = 100$ m.

$$1\sigma_{\mathbf{r}} = 100 \text{ m} \tag{4.5}$$

In the LS algorithm, only the position was included in the estimation state, and therefore iterations were only performed with the spacecraft position.

The spread of the position states are shown in Figure 4.20(a), and the magnitude of the initial errors versus the number of iterations is shown in Figure 4.20(b). For the cases that converged, the RMS of the magnitude of the ending position error was 0.058 m.



(a) Body-fixed view

(b) Magnitude of errors

Figure 4.20: Initial position errors and resulting LS MC outcomes for position-only variations with $1\sigma_{\mathbf{r}} = 100$ m.

A distinct trend is present in Figure 4.20(b) between the initial position error and the number of iterations for convergence. Unlike the cases that estimated the pointing, this simulation did not contain any cases that stalled out or were terminated early. Each case either converged or did not have an initial overlap in observed and computed images to initiate the filter. The cases that did not have an initial overlap of observed and computed images contained position errors larger than 200 m and were on the periphery of the perturbed states.

4.3.5 Discussion

These simulations tested variations of one state with the LS algorithm. When the position and pointing were estimated in the state, the LS algorithm was terminated early if the position error was large. For example, the cases that did not finish yet had an initial image overlap in Figure 4.13 were on the periphery of the perturbed states. When this occurred, the iterations within the LS algorithm produced states that were no longer in view of the asteroid and an overlap did not occur between the observed and computed measurements.

When only the position was estimated and the pointing was held constant, every case converged if an initial overlap in the observed and computed images existed. This implies that better performance could be achieved if only the position is estimated first in the LS algorithm, and then after convergence, adding pointing into the estimation state. If it was known that the spacecraft position errors were large, and the pointing errors were small, this sequential approach would be beneficial.

These simulations were also performed with just one observation, and without the outer Kalman filter used for state propagation. Filter simulations showed that even if the least-squares algorithm did not converge on the truth state for one observation, the overall filter could still converge. The different orientations of the asteroid, and the propagation of the spacecraft provided additional information in the Kalman filter to aid in convergence.

This least-squares algorithm shows promise to solve for pointing errors at each observation time step, such as if the spacecraft pointing experiences jitter. The Monte Carlo simulation that only slightly varied the position, but varied the pointing with $1\sigma = 3^\circ$, saw convergence in all of its runs if the pointing error was less than 3°. This shows promise for solving for a constant pointing bias, as well as observation-dependent pointing errors within a Kalman filter.

4.4 Shape Modeling Errors

In practice, perfect knowledge of the asteroid shape may be unknown, and errors may be present in the shape model even at the highest fidelity. From an autonomous navigation standpoint, a low fidelity shape model may be used onboard the spacecraft to reduce the computational load. Since the overall goal of this research is to advance navigation autonomy, the robustness of the filtering methods presented was tested by using a low fidelity shape model in the filter, and using a high fidelity shape model to produce the truth measurements.

The resolutions of the Itokawa shape models studied here are described as FV-8, FV-16, and FV-32.[44] The FV-32 model was the highest fidelity model of the three with approximately 12,000 facets, while the FV-8 model was the lowest fidelity model with approximately 800 facets (shown in Figure 2.3(b)). The lower fidelity shape models were formed by removing every other vertex in the next-higher-fidelity model; for example, the FV-16 model was created by removing every other vertex in the FV-32 model. It is noted that these shape models were readily available [44], and no attempt was made to create a more optimal lower fidelity model.

OD investigations studied the EKF, UKF, and LS filter with these shape modeling errors. To account for the errors, one study increased the uncertainty in the measurement uncertainty matrix, R, and one study implemented a consider filter. Increasing the measurement uncertainty is simple for onboard navigation, and has been investigated for small body missions [9]. Consider covariance analysis have been applied to orbit determination for including sensor biases [54], in recursive implementations [117], and for an asteroid rendezvous [113].

4.4.1 Shape Model Differences

The measured differences in the shape models were computed by projecting the vertices of the higher resolution shape model onto the corresponding facet of the lower resolution shape model. Due to how the lower resolution shape models were formed, it was simple to match the high resolution facets within one low resolution facet. For example, four facets from the FV-32 model corresponded to one facet in the FV-16 model, and 16 facets in the FV-32 model corresponded to one facet in the FV-8 model. An example of the FV-8/FV32 comparison is shown in Figure 4.21.

The vertices from the high resolution shape model were projected onto the facet of the low resolution model along the normal vector of the low resolution facet. These distances were recorded as the measured differences of the shape models. If a difference was negative, this meant the high resolution vertex protruded beyond the low resolution model, and vice versa.

Figure 4.22 illustrates the differences between the FV-8 and FV-32 models, where Figure 4.22(a) represents the measured differences as explained above, and Figures 4.22(b) and 4.22(c) represent different views of the overlap of the FV-8 and FV-32 shape models. In Figure 4.22(a), the colored axis was bounded between 6 and -20 meters, and in Figures 4.22(b) and 4.22(c), the dark color shows where the FV-8 model has a larger radius, and the light color shows where the

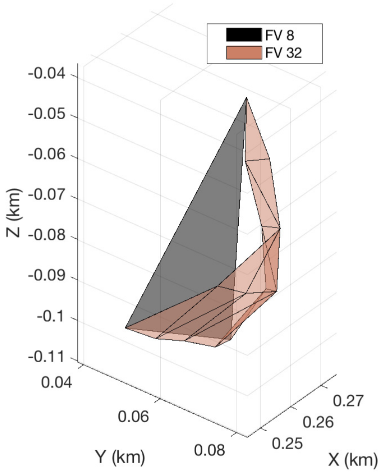
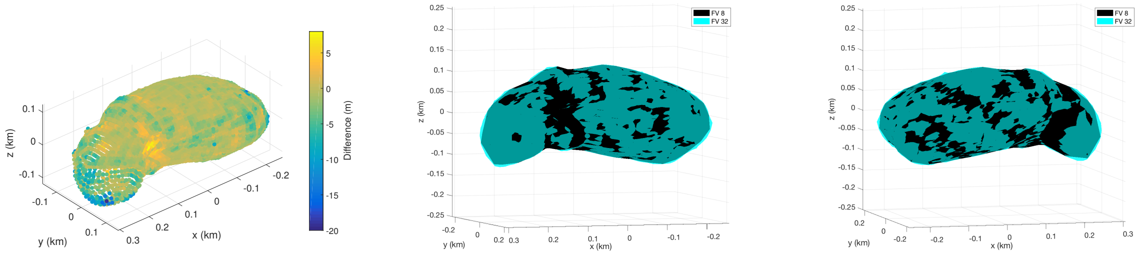


Figure 4.21: Corresponding facets from the FV-32 model within one facet of the FV-8 model.

FV-32 model has a larger radius.

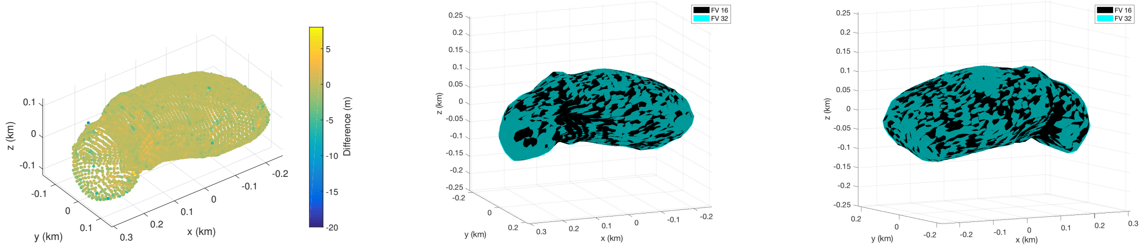
One can notice that the FV-32 model has a larger radius than the FV-8 model along the boundaries of the Itokawa image. This shows up in the negative measured differences in Figure 4.22(a), and in the light color border in Figures 4.22(b) and 4.22(c). Due to do how the FV-8 model was created, some features were cut out around the outer edges of Itokawa, creating a smaller radius here, as well as around the neck of Itokawa, creating a larger radius here.

Figure 4.23 illustrates the differences between the FV-16 and FV-32 models, where Figure 4.23(a) represents the measured differences bounded between 6 and -20 meters, and Figures 4.23(b)



(a) Measured differences (b) Overlap differences, $+^{BF}y$ view (c) Overlap differences, $-^{BF}y$ view

Figure 4.22: Differences in FV-8 and FV-32 facet/vertex models of Itokawa



(a) Measured differences (b) Overlap differences, $+^{BF}y$ view (c) Overlap differences, $-^{BF}y$ view

Figure 4.23: Differences in FV-16 and FV-32 facet/vertex models of Itokawa

and 4.23(c) show the overlap between the FV-16 (dark color) and FV-32 (light color) models. In this comparison, the measured differences and overlaps were not as severe as the FV-8/FV-32 comparison and were more spread out over the body.

Table 4.5 shows the mean, standard deviation, and maximum absolute value of the measured differences in the FV-8/FV-32 and FV-16/FV-32 comparisons. The statistics in Table 4.5 support the assumption that the low resolution models had a smaller radius since the mean of the differences in both comparisons was negative. This implied that the differences in the shape models were biased. This was not ideal since it is desired for any errors introduced into the state estimation filter to have a zero-mean and normal distribution.

Table 4.5: Mean, standard deviation, and maximum absolute value of the measured differences between the shape model fidelities.

Comparison	Average Difference (m)	Standard Deviation (m)	Maximum Difference (m)
FV-8/FV-32	-0.61	1.88	20.29
FV-16/FV-32	-0.100	0.713	11.98

4.4.2 Orbit Determination Robustness

Shape modeling errors were introduced into the filter simulations to test the robustness of the flash lidar measurements. A high fidelity shape model of Itokawa was used to create the truth measurements, and a lower fidelity model was used in the filter. The OD was performed in the circular terminator orbit around Itokawa with measurements every two hours for ten days. It was assumed that the spin state and spin axis Itokawa were known, as well as the dynamics.

The errors introduced in these simulations included Gaussian white noise on the altimetry measurements from Equation 3.1, and shape modeling differences between the onboard filter shape model and the truth shape model. The measurement uncertainty matrix was implemented as in Equation 2.171, but with an additional uncertainty added to each range measurement of $1\sigma_{shp}$ m, in Equation 4.6.

$$R_{shp} = \texttt{diag}\left[[(\sigma_{\rho_i} + \sigma_{shp})^2]_{p\times 1}\right] \tag{4.6}$$

Two values of σ_{shp} were investigated: $\sigma_{shp} = 2$ m and $\sigma_{shp} = 6$ m. From Table 4.5, the average difference was -0.6 m for the FV-8/FV-32 comparison, with a standard deviation of about 2 m. Using $\sigma_{shp} = 2$ m reflects the shape modeling errors based on Table 4.5, and using $\sigma_{shp} = 6$ m reflects a conservative approach to accounting for the shape modeling errors. Increasing the uncertainty of the range measurements based on the shape uncertainty is a simple approach that does not require the formulation of a new filter. This supports the autonomous navigation goal of this research, and a similar approach is used for the OSIRIS-REx lidar-guidance algorithm. [9]

It is noted that by increasing the measurement noise with Equation 4.6, the tolerance for convergence in the LS filter (Equation 2.112) was also increased. This was suitable since errors were known to be present in the measurements, and the iterations of the LS algorithm might not be able to reach the same tolerance as with the known shape model. The LS algorithm was able to continue to iterate on a solution once it was below the tolerance, and terminated once J stalled out from Equation 2.113.

The OD simulations estimated the spacecraft position, velocity, and pointing, and used the same *a priori* covariance as in Equation 3.9. Process noise was added as in Equation 3.3. Each simulation employed the same *a priori* state error in position and velocity, taken from a normal distribution based on Equation 3.9, and the *a priori* pointing angles were set to zero.

For comparison, Table 4.6 presents the magnitude of the state errors excluding the first five measurements for the FV-8/FV-32 case without the addition of measurement noise.

Table 4.6: RMS values of the magnitude of the position and velocity errors, the total angular pointing offset from nominal, α, excluding the first five observations (†), and the total run time. This is in the Itokawa terminator orbit with measurements every 2 hours for 10 days for the FV-8/FV-32 case without additional measurement noise.

Filter	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
EKF	2.10	0.0155	0.064	4.1
UKF	4.23	0.0262	0.215	12.3
LS	2.07	0.0139	0.085	29.6

Figure 4.24 presents the EKF (a), UKF (b), and LS filter (c) results when using the FV-8 shape model in the estimation filter, the FV-32 model for the truth measurements, and a shape uncertainty of $\sigma_{shp} = 2$ m. Table 4.7 presents the RMS of the magnitude of the position, velocity, and pointing errors, averaged without the first five measurements, and the run time of the filter.

Table 4.7: RMS errors of the magnitude of the position and velocity error and the total angular pointing offset from nominal, α excluding the first five observations (†), and the run time of the filter. This is in the Itokawa terminator orbit with measurements every 2 hours for 10 days for the FV-8/FV-32 case and $\sigma_{shp} = 2$ m.

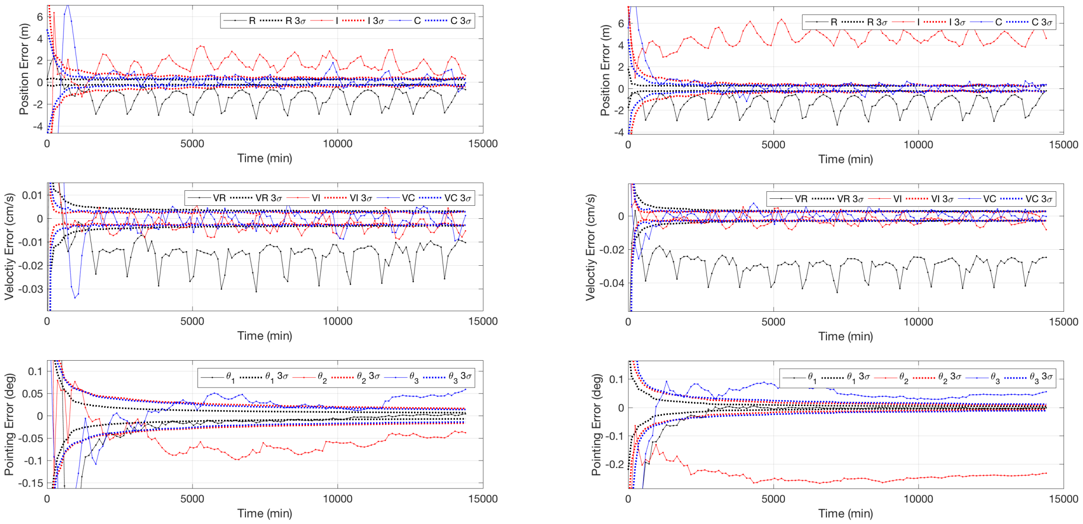
Filter	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
EKF	2.62	0.0177	0.085	4.1
UKF	5.00	0.0305	0.244	12.5
LS	2.19	0.0154	0.078	9.4

The filter results, using a FV-32 model for the truth measurements, the FV-8 model in the filter, and increasing the measurement uncertainty with $\sigma_{shp} = 6$ m, are presented next. Figure 4.25(a) shows the state errors and 3σ covariance bounds for the EKF, Figure 4.25(b) shows the state errors and 3σ covariance bounds for the UKF, and Figure 4.25(c) shows the state errors and 3σ covariance bounds for the LS filter. The magnitude of errors for this simulation, along with additional cases for the FV-16/FV-32 case are presented in Table 4.8.

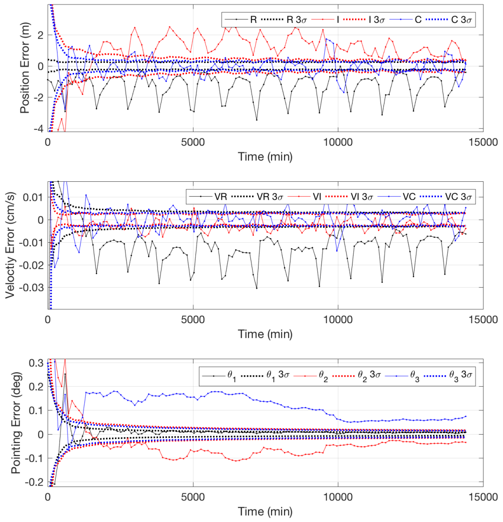
All six simulations showed similar patterns in their error results: the errors in the radial and in-track position directions, and the radial velocity direction plateaued outside their covariance bounds. Peaks occurred in the radial and in-track position errors at the same observation times.

Increasing the measurement uncertainty from $\sigma_{shp} = 2$ m to $\sigma_{shp} = 6$ m did not cause a significant change in the filters, and the state errors were of the same order of magnitude.

The same filter simulations, with the same *a priori* state and $\sigma_{shp} = 6$ were run with the FV-16



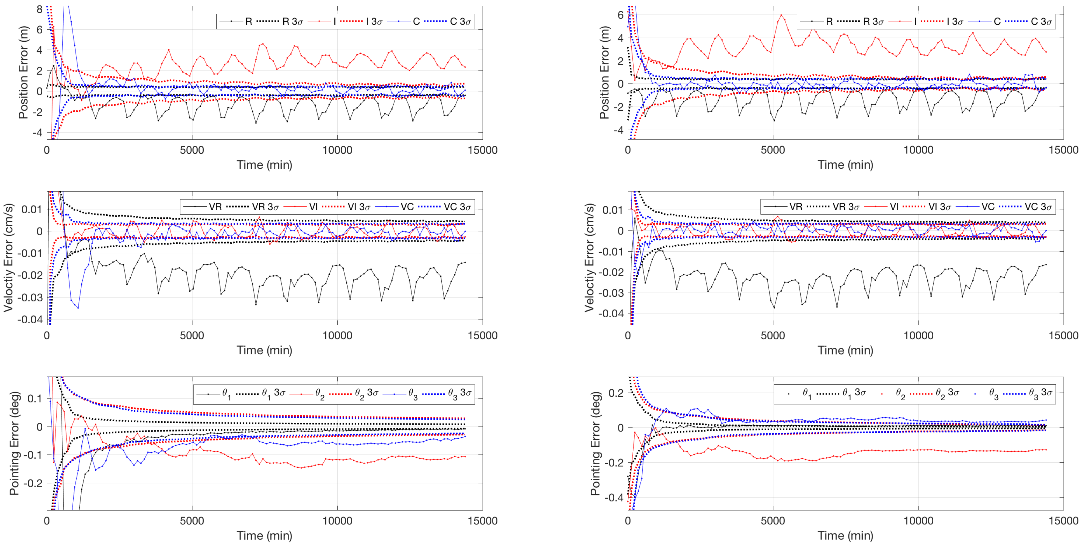
(a) EKF (b) UKF



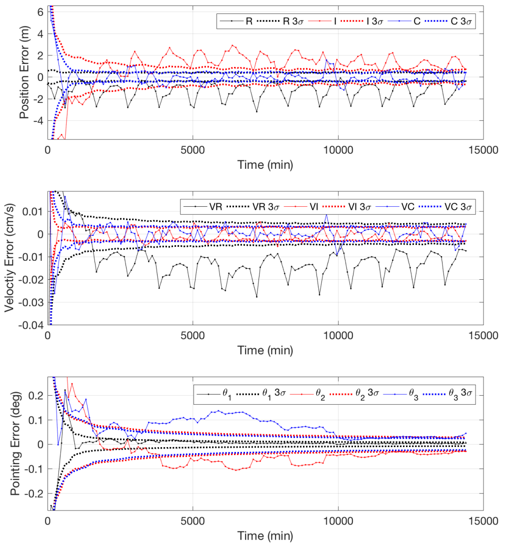
(c) LS

Figure 4.24: Errors in the radial (R), in-track (I), and cross-track (C) frame (solid) and the 3σ covariance bounds (dashed) with FV-8/FV-32 shape modeling errors with $\sigma_{shp} = 2$ m.

model as the onboard shape model, and using the FV-32 model to produce the truth measurements. The EKF, UKF, and LS filter for the FV-16/FV-32 case all showed a similar pattern of peaks in the radial errors at specific observation times. The errors were not as severe, and were inside the 3σ



(a) EKF (b) UKF



(c) LS

Figure 4.25: Errors in the radial (R), in-track (I), and cross-track (C) frame (solid) and the 3σ covariance bounds (dashed) with FV-8/FV-32 shape modeling errors with $\sigma_{shp} = 6$ m.

covariance bounds with the LS filter, yet slightly outside of the covariance bounds with the EKF and UKF. While the EKF and UKF errors were on the same order of magnitude, the covariance bounds in the UKF simulation were smaller than those in the EKF, resulting in more errors falling outside of its covariance bounds.

4.4.2.1 Filter Analysis

Each of the filters for the FV-8/FV-32 case and the FV-16/FV-32 case saw an oscillating state error at the same observation times in position and velocity. Figure 4.26 depicts the estimated position of the spacecraft in the BF frame from the FV-8/FV-32 UKF ($\sigma_{shp} = 6$ m) simulation when the peaks in the radial errors occurred. It reveals that these periodic errors occurred when the spacecraft viewed a particular area of Itokawa. This area can be classified as the negative $x-$coordinates of Itokawa in the BF frame.

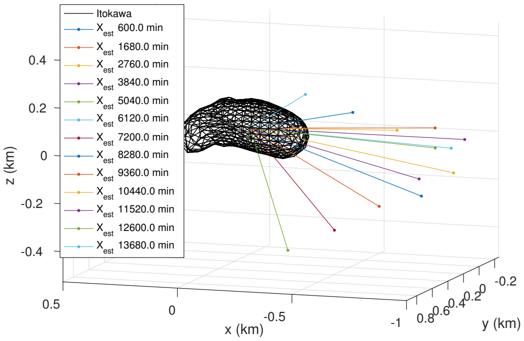


Figure 4.26: Estimated spacecraft position in the Itokawa BF frame from the FV-8/FV-32 UKF simulation when the radial error peaks occurred in Figure 4.25(b).

Figure 4.27 shows two histograms of the measured differences between the FV-8 and FV-32 models, divided between the positive and negative BF x-coordinates of Itokawa. While both histograms were skewed toward negative differences, the differences associated with the negative BF x-coordinates involved a greater skew. This implies that the overall radius of the FV-8 model

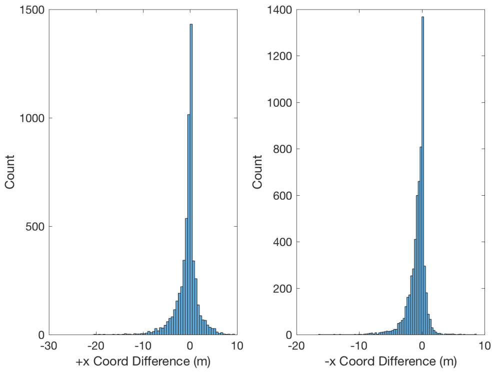


Figure 4.27: Histograms of $+x$ and $-x$ shape model differences for the FV-8/FV-32 comparison of the Itokawa shape model

was less than that of the FV-32 model in this area of Itokawa.

Within the filter, the average pre-fit residuals revealed that the computed measurements were greater than the observed measurements at these observation times. The filter adjusted the spacecraft position closer to the asteroid, and resulted in the negative radial errors seen in the filter results. When the spacecraft was close to the body, it traveled faster and caused larger in-track position errors as a secondary effect.

Table 4.8 presents the RMS values of the magnitude of the position and velocity errors, the total off-nominal offset pointing error (α), and the total run time for the EKF, UKF, and LS filters. These are presented for the FV-8/FV-32 case and the FV-16/FV-32 case with $\sigma_{shp} = 6$ m, and the FV-32 case that did not include additional measurement uncertainty. In the FV-32 case, the FV-32 model was used for the onboard shape model and to produce the truth measurements.

The RMS errors for the FV-16/FV-32 case were slightly less than those for the FV-8/FV-32 case for each component error. Across all of the shape model comparison cases, the LS filter performed the best of the three filters in terms of higher accuracy and lower computational times.

The accuracies when using the FV-32 shape model for the onboard measurements versus a lower fidelity onboard model were on average an order of magnitude less. The computational time when using the FV-32 model, however, was increased by an order of magnitude. It is noted

Table 4.8: RMS values of the magnitude of the position and velocity errors, the total angular pointing offset from nominal, α, excluding the first five observations (†), and the total run time. This is in the Itokawa terminator orbit with measurements every 2 hours for 10 days for the FV-8/FV-32 case and $\sigma_{shp} = 6$ m.

Filter/Truth Shape Model	Filter	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
FV-8 / FV-32	EKF	3.43	0.0215	0.137	3.5
FV-8 / FV-32	UKF	3.68	0.0236	0.142	11.5
FV-8 / FV-32	LS	2.28	0.0148	0.084	9.7
FV-16 / FV-32	EKF	2.22	0.0114	0.134	6.1
FV-16 / FV-32	UKF	1.82	0.0101	0.103	39.8
FV-16 / FV-32	LS	0.62	2.84E-3	0.033	11.1
FV-32	EKF	1.03	5.56E-3	0.060	16.5
FV-32	UKF	1.69	7.64E-3	0.103	152.6
FV-32	LS	0.12	8.68E-4	4.66E-3	25.4

that with a higher resolution instrument, such as 256 × 256, the differences in the shape model resolutions may be more pronounced, and possibly result in smaller errors.

Overall, with the LS filter, a similar level of accuracy and an increase in computation speed is achieved when using an onboard shape model that is of lower fidelity than the truth shape model.

4.4.3 Orbit Determination with the Consider Filter

When introducing the shape modeling errors, the onboard lower fidelity shape had a smaller radius than the high fidelity truth shape. This caused subsequent errors in the estimation state. With the lower fidelity shape model in the filter, errors were introduced on the individual range measurements in the image array. Estimating an error on the individual range measurements would cause observability issues of the state since the errors are linearly dependent on the range. It was assumed that calibration procedures had been performed of the flash lidar instrument and there were no biases on the pixels due to the sensor. The biases in the state errors were assumed to be solely from the asteroid shape mismodeling.

To compensate for the state errors due to the shape mismodeling, a consider covariance analysis and consider filter were implemented. A constant bias was assumed to be present on each of the range measurements at one time step, and this was the consider parameter. The consider state, C was defined as a range bias, b:

$$C = [b] \tag{4.7}$$

This bias appears in the measurement model as an addition to the measured range, ρ:

$$\rho_i = \hat{u}_i^T(\mathbf{r}_A - \mathbf{r}) + b \tag{4.8}$$

The measurement mapping matrix for the bias parameter was:

$$\tilde{H}_c = \frac{\partial \rho_i}{\partial b} = 1 \tag{4.9}$$

The covariance matrix of the consider parameter was simply the uncertainty:

$$\bar{P}_{cc} = \sigma_b^2 \tag{4.10}$$

The mapping matrix, $\theta(t, t_k)$ is defined as the change in the state with respect to the change in the consider parameters over time. The consider parameter does not change the state dynamics, and only effects the observations. Therefore, the matrix $\theta(t, t_k) = 0$, and:

$$\bar{S}_k = \phi(t_k, t_{k-1})S_{k-1} \tag{4.11}$$

4.4.3.1 Consider Analysis and Filter Results

A consider analysis and consider filter were run with an uncertainty based off of the shape model statistics and a conservative shape uncertainty. The FV-8/FV-32 shape model comparison in the circular terminator orbit around Itokawa with measurements every two hours for ten days was used as the basis of this simulation. This was the exact same filter simulation as used as in Section 4.4.2. The difference was that a consider covariance and consider update were produced alongside the filter simulation.

Two cases were studied with the consider filter. In the first, the value of $\bar{\mathbf{c}}$ and its uncertainty, σ_b was defined from the errors computed when differencing the low and high fidelity shape models. For the FV-8/FV-32 case, these values were: $\bar{\mathbf{c}} = -0.6$ m, and $1\sigma_b = 2$ m. The second case used a more conservative in the shape model uncertainty with $\bar{\mathbf{c}} = -4$ m, and $1\sigma_b = 6$ m. The mean of the pre-fit residuals at the occurrence of the radial error peaks in the FV-8/FV-32 EKF simulations were all close to 4 m, and therefore, this mean was chosen to conservatively include the errors.

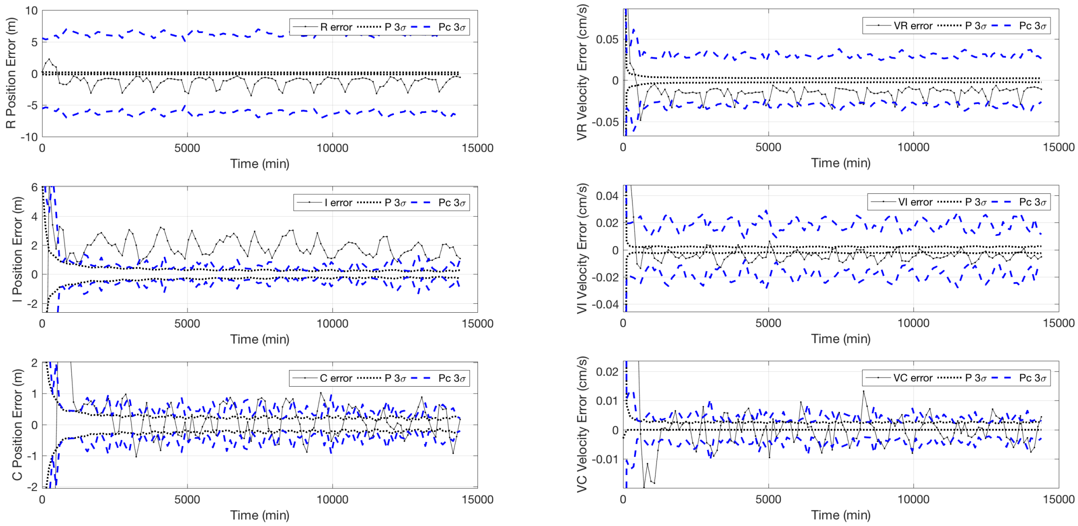
The consider covariance P_c is the same for the consider analysis and consider filter algorithms. In the consider analysis, the consider state update depends on the *a priori* value of the consider parameters, $\bar{\mathbf{c}}$, while in the consider filter, the consider state update is calculated with a consider Kalman gain and the consider covariance. In the presented results, the consider covariance, P_c is presented with the state error and covariance from the filter. To determine error in the consider state, the consider state update, $\hat{\mathbf{x}}_{c_k}$ was added to the time update of the state, $X_k^* + \bar{\mathbf{x}}_k$, and differenced with the truth state.

Figure 4.28 presents the state error and the 3σ state covariance bounds from the estimation filter, and the 3σ consider covariance bounds, P_c for $1\sigma_b = 2$ m. Figure 4.29 presents errors in the filter state, the consider analysis (CA) state, and the consider filter (CF) state for $\sigma_b = 2$ m and $\bar{\mathbf{c}} = -0.6$ m.

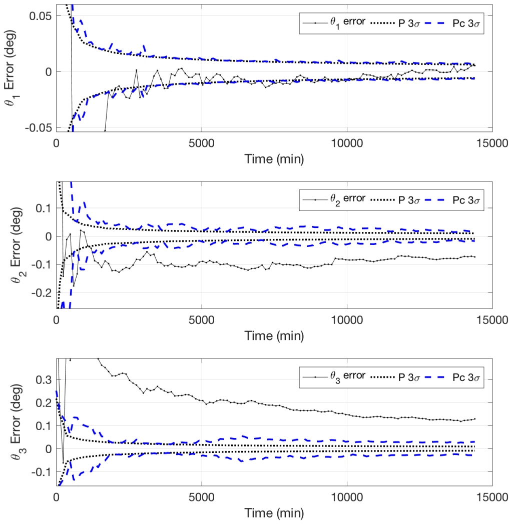
The consider covariance captured all of the state errors, with the exception of the in-track position direction, and θ_2 and θ_3 in the pointing state. The consider state updates were all consistent with the filter state estimates, and all followed the same trend. The consider filter, which does not depend on $\bar{\mathbf{c}}$, produced slightly greater errors in the estimation state, while the consider analysis state update resulted in slightly less errors.

Figure 4.30 presents the consider covariance 3σ bounds with the filter state error and state covariance bounds when using the conservative $\sigma_b = 6$ m in the consider analysis. Figure 4.31 illustrates the errors in the filter state and the consider state when evaluated from the consider analysis (CA) and consider filter (CF) with $\bar{\mathbf{c}} = -4$ m and $\sigma_b = 6$ m.

When the consider parameter uncertainty was increased to $\sigma_b = 6$ m, the consider covariance in the radial direction reached ±20 m. This captured the state errors from the filter, but the RMS of those errors were less than 3 m (Table 4.6). The in-track position consider covariance was also inflated, but still did not encapsulate the state errors in the in-track position direction. The same was the case for the pointing angles; the consider covariance was inflated, yet did not capture the

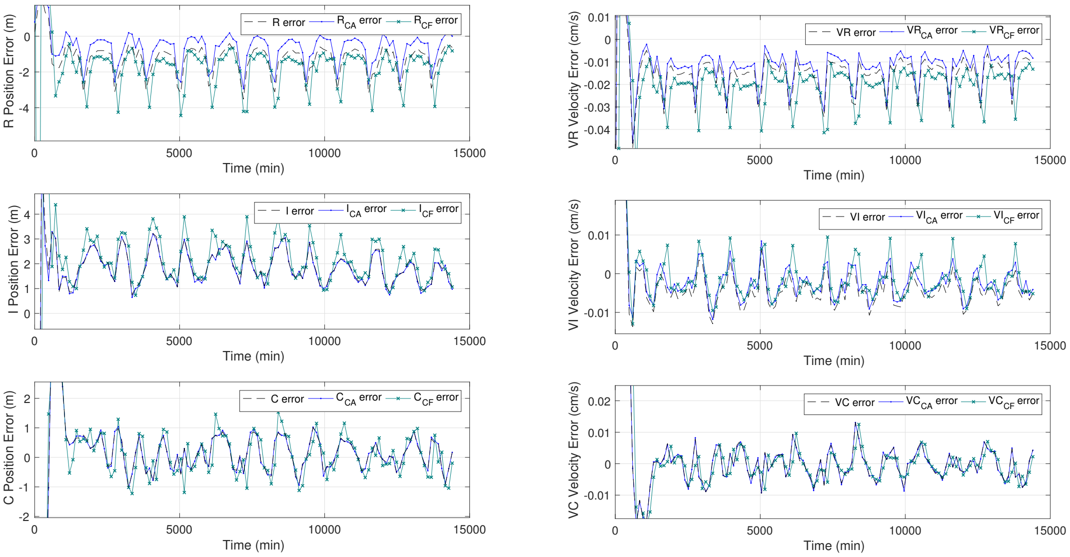


(a) Position (b) Velocity



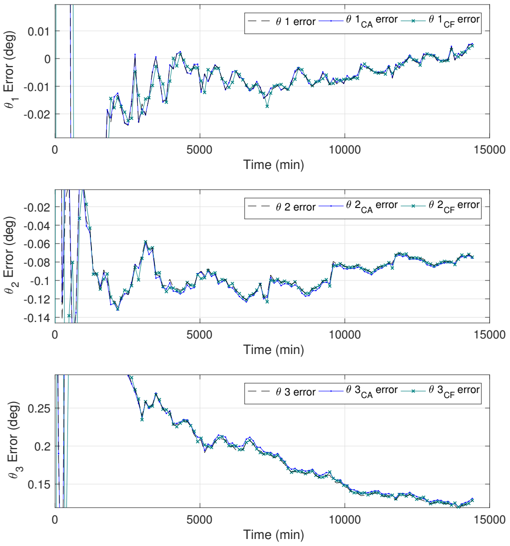
(c) Pointing

Figure 4.28: State errors in the RIC frame (black solid), the 3σ state covariance bounds (black dashed), and the 3σ consider covariance bounds (P_c, blue dashed) with FV-8/FV-32 shape modeling errors with $\sigma_b = 2$ m.



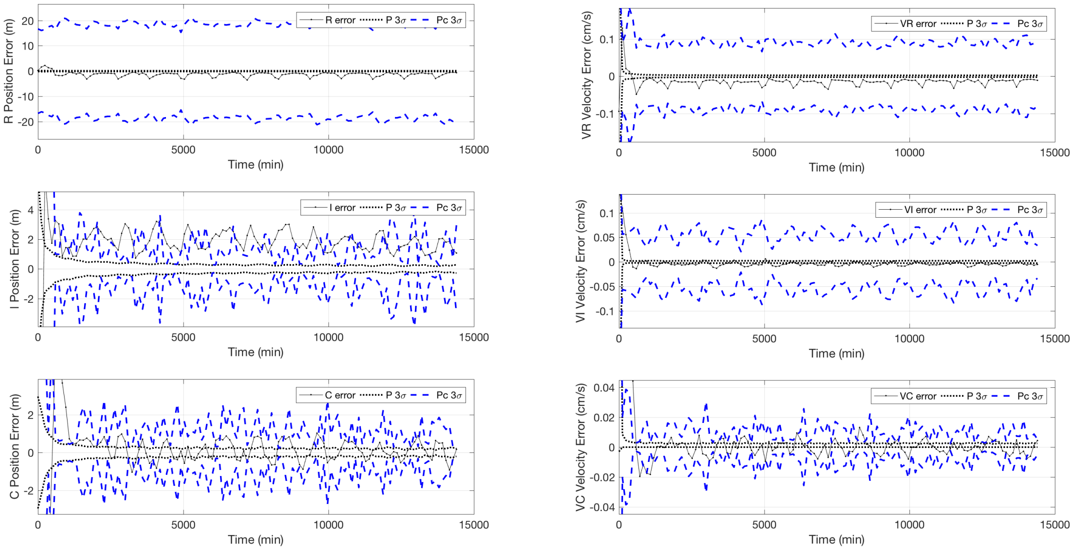
(a) Position

(b) Velocity



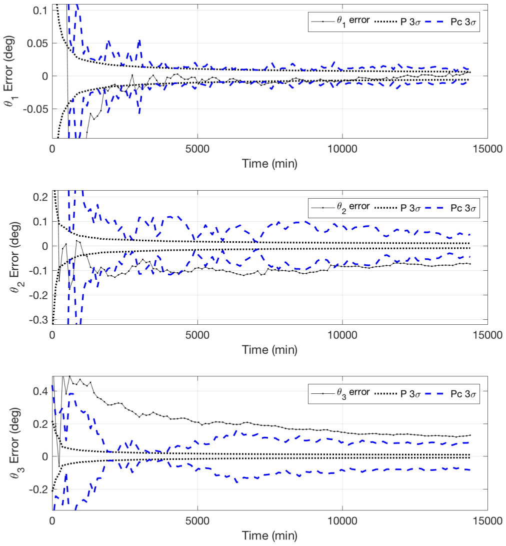
(c) Pointing

Figure 4.29: Errors of the filter state (black dashed), consider state from the analysis (CA, blue), and consider state from the filter (CF, teal) with FV-8/FV-32 shape modeling errors with $\sigma_b = 2$ m.



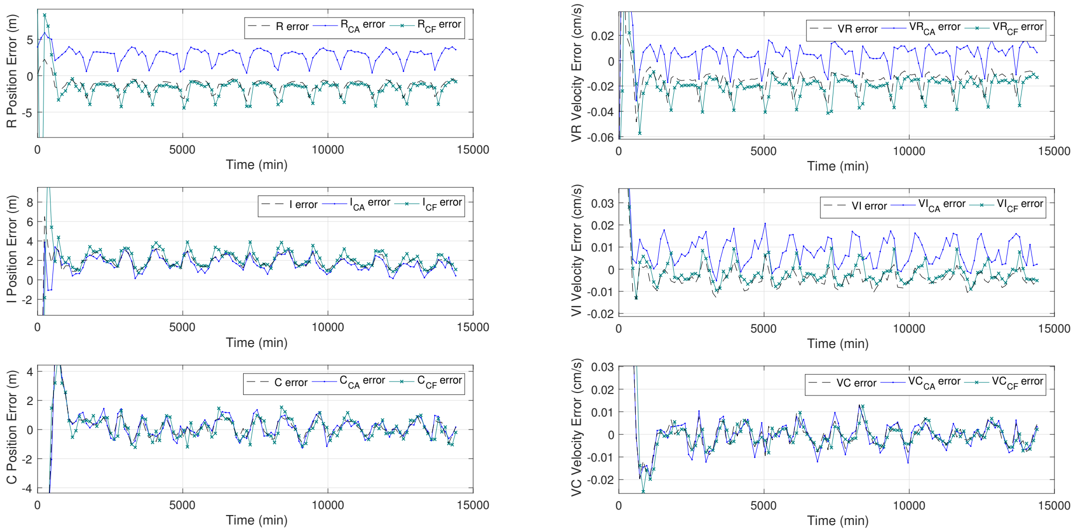
(a) Position

(b) Velocity



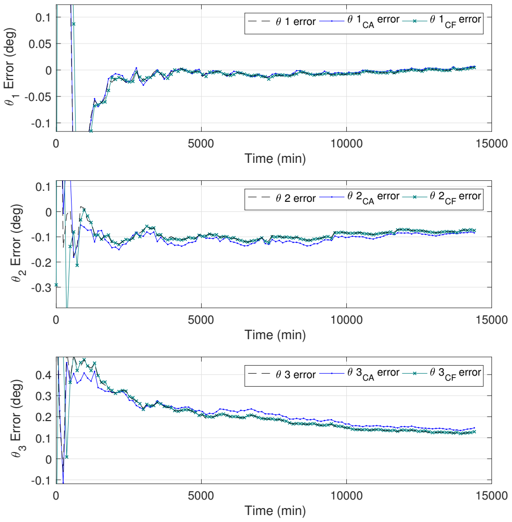
(c) Pointing

Figure 4.30: State errors in the RIC frame (black solid), the 3σ state covariance bounds (black dashed), and the 3σ consider covariance bounds (P_c, blue dashed) with FV-8/FV-32 shape modeling errors with $\sigma_b = 6$ m.



(a) Position

(b) Velocity



(c) Pointing

Figure 4.31: Errors of the filter state (black dashed), consider state from the analysis (CA, blue), and consider state from the filter (CF, teal) with FV-8/FV-32 shape modeling errors with $\bar{\mathbf{c}} = -4$ and $\sigma_b = 6$ m.

errors in θ_2 and θ_3. Increasing $\bar{\mathbf{c}} = -4$ m distinguished the consider state updates. The consider analysis state update that depended on $\bar{\mathbf{c}}$ shifted the radial position and velocity errors upwards toward the positive errors. However, the shift was too great, and the errors were of the same magnitude but positive instead of negative. The consider state update in the other components did not see significant changes.

4.4.3.2 Discussion

A consider filter is designed to account for dynamical or measurement parameters that contain uncertainty, yet are not directly estimated. These filters are best suited for parameters that may not be observable or would increase the computational load of the filter. With shape modeling errors present, the estimation state incurred errors that were not captured by the filter covariance.

A consider covariance analysis and filter are implemented by assuming a range bias, b with uncertainty σ_b as the consider parameter. With the bias uncertainty based off the measured shape model differences, the consider covariance captures the filter state errors in all but three state components. When the bias uncertainty is increased for a conservative approach, the 3σ consider covariance bounds are six times larger than the filter state errors in the radial position, and do not capture the errors in the same three components. The state errors with the consider state update for the consider analysis and consider filter algorithms are similar to the filter state errors for both σ_b cases.

The consider covariance would be suitable in this situation to capture the majority of the state errors due to shape mismodeling, yet the filter parameters would need further investigation and tuning to adequately capture all of the state errors.

4.5 Pointing Jitter with the Least Squares Filter

The LS filter did not employ *a priori* information to estimate the position and pointing of the spacecraft at each time step, and the filter can easily solve for different state errors at each observation time as evidenced from Section 4.3. This is ideal if the spacecraft experiences pointing jitter, where a random unknown error in the pointing might be present at each observation time.

This study investigated applying a random pointing error at each observation time and implementing the LS filter for OD. The terminator orbit with observations every 2 hours for 10 days was used as the nominal orbit. In the truth measurements, a random pointing error was applied to each of the pointing angles of $\underline{\theta}$, based on a normal distribution with $1\sigma = 0.5°$. If each angle, θ_n, was set 0.5°, this was equivalent to a total off-nominal offset of 12 mrad. With these errors implemented on the truth measurements, the total off-nominal offset ranged from 0.08° to 1.6°, a fairly conservative range of angles.

The same filter setup was employed as in Section 3.2.1, with *a priori* position and velocity errors based on the *a priori* covariance in Equation 3.9, and process noise added based on Equation 3.3. The range measurements had white noise applied based on Equation 3.1, and in this simulation, the shape model for the truth and computed measurements was the FV-8 model. The *a priori* angles, θ_n, were reset to zero at each observation time, and the *a priori* covariance for the angles was reset to the original *a priori* covariance in Equation 3.9 at each observation time.

Figure 4.32 illustrates the state errors and 3σ covariance bounds when using the LS filter with pointing jitter. The covariance bounds in this simulation were slightly larger compared to the pointing bias simulations, and plateaued around 1 m in position and 0.1° in pointing. This was due to the reset of the pointing angle covariance at each time step. The errors were still fairly small, reaching sub-meter accuracy in position, and below 0.1° in pointing, and the LS filter converged on the correct pointing state within the resulting uncertainty. These results are promising for using the LS filter not only to estimate a pointing bias, but also random pointing errors present at each time step.

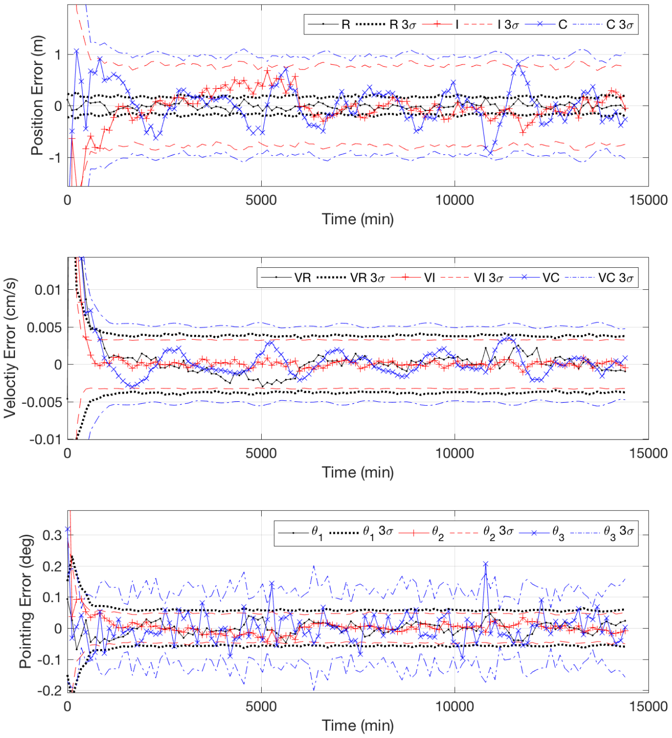


Figure 4.32: Errors in the radial (R), in-track (I), and cross-track (C) frame (solid) and the 3σ covariance bounds (dashed) when using the LS filter with pointing jitter errors.

Chapter 5

Filter Initialization

To initialize the estimation filter, one must have an *a priori* guess of the estimation state. Using the image properties of the flash lidar measurement, one can determine a rough estimate of the spacecraft state, and use this as the *a priori* state in the estimation filter. This is applicable if the spacecraft has exited safe mode. With this approach, it can autonomously determine an educated guess at its position and pointing with respect to the the asteroid. Through two consecutive position estimates, the spacecraft can determine an estimated velocity, and use this for the *a priori* filter state.

It is assumed in this scenario that the spacecraft has performed a pre-determined search pattern of capturing images with the goal of capturing an image of the asteroid. Once the asteroid is in view, the spacecraft determines its inertial pointing direction with onboard star tackers. It is further assumed that a shape model was available to the spacecraft. From knowledge of the asteroid shape, it is assumed that the spacecraft is able to estimate the relative attitude of the asteroid by predicting the spin state from the shape model and an onboard clock.

A pre-processing algorithm is presented in this section to estimate the spacecraft position and pointing at a single observation time using only the flash lidar observation at that time, knowledge of the inertial spacecraft pointing, and knowledge of the asteroid attitude. Monte Carlo simulations are performed to test this algorithm with a variety of position and pointing states. The pre-processing algorithm is performed on two observations to estimate the spacecraft position and pointing, and the two subsequent position estimates are used to estimate the spacecraft velocity.

The estimated position, velocity, and pointing are the *a priori* state in the filter that further refined the spacecraft state estimation.

5.1 Edge Detection Algorithm

The image properties of a flash lidar measurement allow features to be extracted and matched between the observed and computed images. One of those features is the edges of the asteroid in the returned flash lidar array. Edge detection algorithms are seen in spacecraft relative navigation such as fitting craters to an ellipse in optical images [23], or fitting the limb of a planet to an ellipse while on approach to a planet [28]. Determining the edges may be performed with various algorithms. The ones investigated here include using a fast Fourier transform (FFT), image shifting, and a neighbor search.

Fast Fourier Transform Fast Fourier transforms (FFTs) are commonly used to determine features in an image and correlate two images that maybe shifted or rotated with respect to each other. FFTs are also commonly used for image processing and correlation, and have been investigated for spacecraft relative terrain navigation for landing.[78] High frequencies in an image correspond to sharp changes, such as edges or corners, while low frequencies correspond to gradual changes or shading. A high-pass FFT was applied to the flash lidar image to determine the edges of the asteroid by only allowing the high frequencies in the picture to be detected. This algorithm used a high-pass filter greater than or equal to the 10th harmonic. The image was inputted into the high-pass filter FFT, and then the inverse FFT (IFFT) algorithm. The edges were detected by recording pixels that were greater than 50% times the maximum value in the filtered image. Figures 5.1(a) and 5.2(a) show the resulting edges for two measurements in the circular terminator orbit around Itokawa.

Image Shifting This approach shifted the image up, down, left, and right, by one pixel index with respect to the original image. The pixels in each flash lidar image without a range-return were set to zero, and the pixels in each shifted image were differenced from the original image. A tolerance was set at 50% of the maximum difference between the shifted and original image, and if a pixel returned a difference greater than this tolerance, it was considered an edge pixel. The results are shown in Figure 5.1(b) and Figure 5.2(b) for two different observations of Itokawa. The advantage to this technique over the FFT is that the pixels on the boundary of the image were also included as edges and the edges were more continuous.

Neighbor Search The Neighbor Search algorithm looped through each pixel that produced a range-return and tested its neighbors for if the pixel produced a range-return, produced no-return, or if it was on the boundary of the image. If a pixel's neighbor did not produce a range-return or was on the boundary of the image, this pixel was recorded as an edge. This resulted in the same edge pixels as the Image Shifting algorithm, as shown in Figures 5.1(b) and 5.2(b).

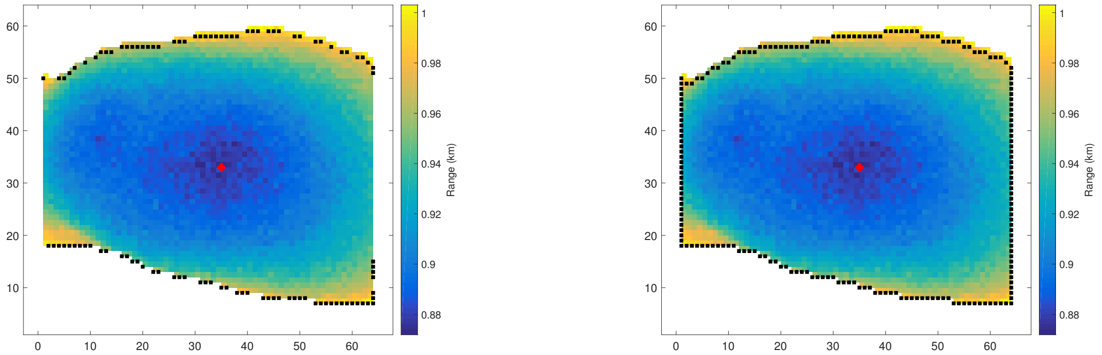
Table 5.1 shows the different run times in milliseconds for the three algorithms for five different measurements in the circular terminator orbit around Itokawa. The Neighbor Search algorithm consistently was the fastest algorithm across the different images, and therefore this algorithm was used in this study.

Table 5.1: Algorithm run times of the edge detection algorithms for different measurements from the Itokawa terminator orbit.

Measurement	FFT (ms)	Image Shift (ms)	Neighbor Search (ms)
A	0.356	0.188	0.185
B	0.355	0.185	0.182
C	0.362	0.195	0.165
D	0.362	0.191	0.160
E	0.373	0.193	0.137

5.2 Pre-Processing Algorithm

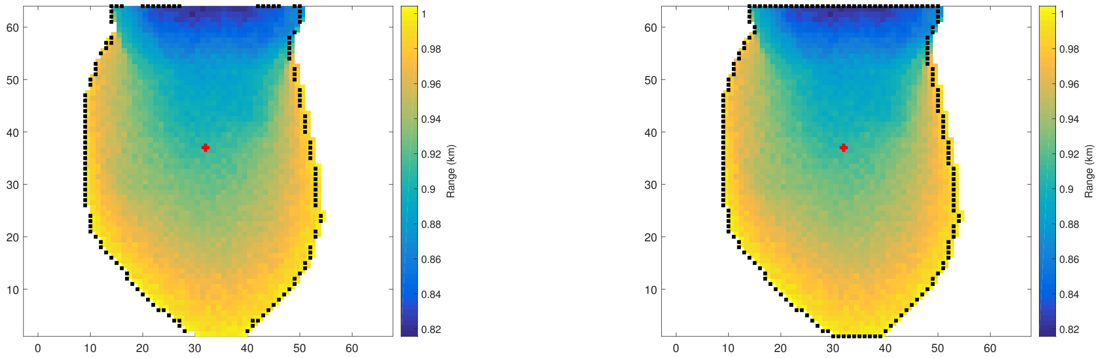
The pre-processing algorithm presented was designed to be run prior to the estimation filter. It estimates an *a priori* position and pointing state at one observation time to be fed into the estimation filter. This was performed without *a priori* information on the spacecraft position. To begin the pre-processing algorithm, it was assumed that the spacecraft would perform a pointing search pattern until an image of the asteroid was captured. With this image captured, the inertial pointing of the spacecraft could be determined by onboard star trackers, and provide the rotation



(a) FFT

(b) Image Shifting and Neighbor Search

Figure 5.1: Edge pixels and center pixel for measurement A as determined by the FFT algorithm (a) and the Image Shifting and Neighbor Search algorithms (b).



(a) FFT

(b) Image Shifting and Neighbor Search

Figure 5.2: Edge pixels and center pixel for measurement C as determined by the FFT algorithm (a) and the Image Shifting and Neighbor Search algorithms (b).

matrix $C_{ACI/SF}$ between the flash lidar sensor frame and the inertial frame. This further assumed that the relative orientation of the instruments on the spacecraft was known, and that the center of the asteroid in inertial space and the asteroid-center inertial frame (ACI) was defined. With an observation of the asteroid in the sensor field of view (FOV), and the inertial pointing of the spacecraft defined, the pre-processing algorithm may commence.

The steps of the pre-processing algorithm are outlined as follows. An initial position estimate was determined from the inertial spacecraft pointing, the range-return of the center pixel of the

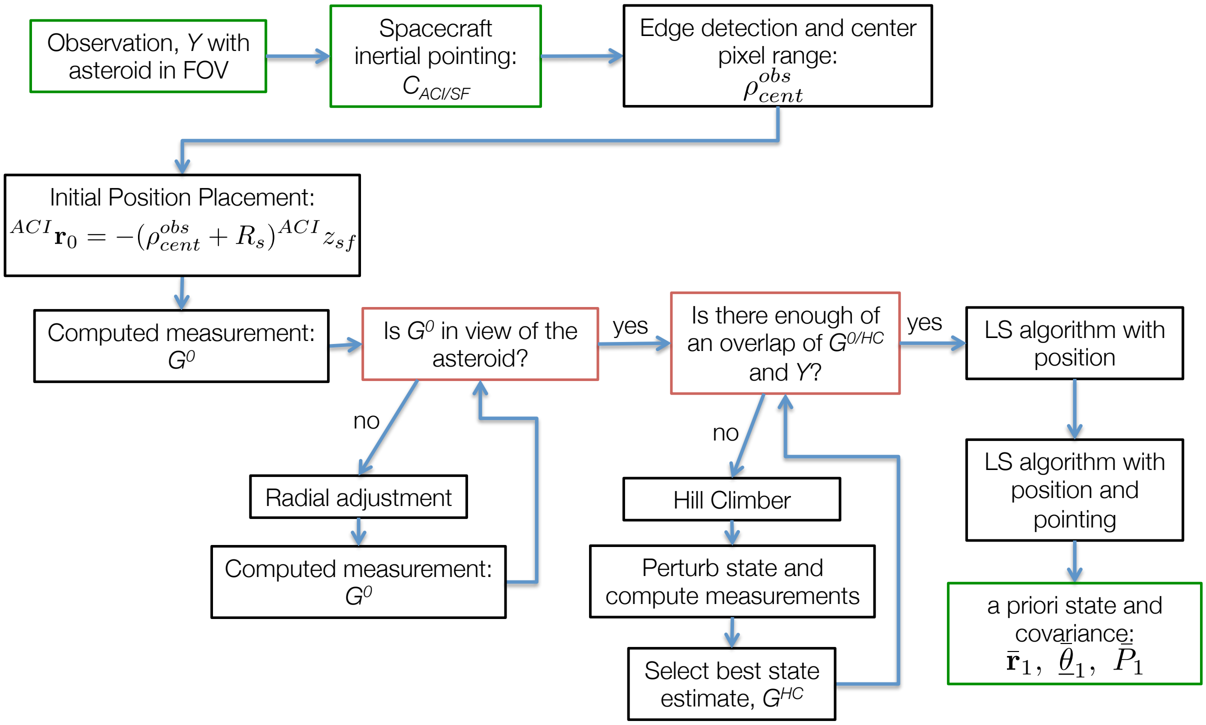


Figure 5.3: Pre-Processing algorithm flow chart

observed image, Y, and the reference radius of the asteroid. If the computed measurement, G^0 from this first state estimate was empty, the spacecraft may be beyond the limit of the flash lidar instrument. If this occurred, the radial position of the spacecraft was incrementally decreased until the computed image was populated. The next step tested if there was an overlap between the observed and computed images, and if a specified minimum number of pixels occurred in this overlap. If this was not the case, a hill climber optimization algorithm was performed for at least two iterations. Once an overlap occurred, the LS algorithm commenced while only estimating the position. After convergence, the LS algorithm was applied again, and estimated the position and pointing. A flow chart of this algorithm is presented in Figure 5.3.

5.2.1 Initial Position Placement

With the knowledge of the spacecraft pointing in the inertial frame, and that the asteroid was in the field of view, it was assumed that the boresight direction of the sensor frame, $+^{SF}z$ was directed at the asteroid. The boresight direction of the sensor frame was rotated into the ACI frame by the rotation matrix $C_{ACI/SF}$:

$$^{ACI}z_{sf} = C_{ACI/SF}(^{SF}z) \tag{5.1}$$

Next, the Neighbors Search edge detection algorithm was implemented to determine the center pixel of the observed image. The returned range value of this center pixel, ρ_{cent}^{obs} was recorded as the predicted altitude of the spacecraft.

The initial guess of the spacecraft position vector was set with the direction as the opposite of the inertial pointing direction, $^{ACI}z_{sf}$ and the magnitude as the reference radius of the asteroid, R_s plus the range of the center pixel, ρ_{cent}^{obs}:

$$^{ACI}\mathbf{r}_0 = -(\rho_{cent}^{obs} + R_s)^{ACI}z_{sf} \tag{5.2}$$

The sensor pointing vector, $^{ACI}z_{sf}$ was defined as pointed from the spacecraft to the asteroid; therefore, the position vector was multiplied by -1 to define the spacecraft direction from the asteroid to the spacecraft.

5.2.2 Radial Adjustment

With the initial position placement of the spacecraft at $\mathbf{r}_0$, a computed image, G^0 was generated. If the computed measurement, G^0 did not produce a measurement return, the spacecraft may be placed outside the limits of the flash lidar instrument. If this happened, the reference radius,

R_s was reduced by 25% until a measurement was returned:

$$\begin{aligned}
&r = R_s \\
&\texttt{while isempty}(G^0): \\
&\qquad r = 0.75r \\
&\qquad \mathbf{r}_0 = -(\rho_{cent}^{obs} + r')^{ACI}\hat{u} \\
&\qquad G^0 = G(\mathbf{r}_0)
\end{aligned} \tag{5.3}$$

5.2.3 Hill Climber Algorithm

The hill-climber algorithm perturbed each element of the initial position forward and backward (Δx), and calculated a cost function, M, for each of these perturbations. The state that resulted in the lowest cost function was saved and this process was repeated. This was a simplified optimization technique that did not require differential equations and moved the state towards the lowest cost. In addition, an initial overlap in the images was not required. The process was iterated for a set number of iterations or until no states were found with a better cost function. The algorithm described here also used simulated annealing, where the perturbation added to the state was reduced at each iteration. Simulated annealing assumes that the state is approaching a local minimum.

The advantage of this algorithm was the flexibility in the cost function because of the ability to define different metrics and their weights. The metrics chosen included the number of overlapping pixels, n_{ol}, the differences in ranges of select pixels, $|\rho_{obs} - \rho_{comp}|$, the area of the image, A, and difference between pixels with and without a return, δ_{px}. The cost function, M, is shown in Equation 5.4 with weights defined in Table 5.2.

$$\begin{aligned}
M = &- w_{ol} n_{ol} \\
&+ w_\rho |\rho_{obs}[1, \cdots, n] - \rho_{comp}[1, \cdots, n]| \\
&+ w_A |A_{obs} - A_{comp}| \\
&+ w_\delta \delta_{px}
\end{aligned} \tag{5.4}$$

Table 5.2: Weights of the metrics used in the hill climber cost function.

Weight	Value
w_{ol}	1
w_{ρ}	100
w_A	0.1
w_{δ}	0.1

The metrics were defined as:

- n_{ol}: The number of overlapping pixels in the observed and computed images

  This was the most important metric, as an overlap of the observed and computed images must occur in order to process the measurements in the filters. This metric was subtracted in the cost function, to further reward an overlap in pixels, and the weight was an order of magnitude larger than the other two metrics that were measured in number of pixels.

- $|\rho_{obs} - \rho_{comp}|$: The difference in ranges of the selected pixels (km)

  The selected pixels were a fixed cross that was centered at the center pixel (calculated from the edge detection algorithm), and expanded to every other pixel for 10 pixels in the x and y directions. If one of these selected pixels in the observed image did not provide a range-return, that pixel was not used in the calculated metric. This used n number of ranges in the metric calculations, with a maximum number of 22 ranges used. If one of these pixels in the computed image did not have a range-return, the difference was assigned a value of 100 km to increase the cost function. This metric helped refine the altitude of the spacecraft, as well as the transverse components.

- A: The area of the image (number of pixels)

  This was the total number of pixels that returned a range value. This metric helps refine the altitude of the spacecraft.

- δ_{px}: The pixel differential (number of pixels)

> To determine the difference between the pixels with and without a return, the two images were overlaid and the metric recorded the pixels that had a range-return in one image and a no-return in the other. For example, the observed image could occupy the left portion of the image, and the computed image could occupy the right portion of the image. These images could provide the same area, yet the pixel differential would be large as the images might not overlap.

Implementing simulated annealing, the perturbation of the computed state, Δx, was reduced after each iteration by a percentage, ζ (Equation 5.6). The initial perturbation, Δx_0, in each component was equal to half of the field of view at that altitude (Equation 5.5).

$$\Delta x_0 = \rho_{cent}^{obs} \tan(10^\circ) \tag{5.5}$$

$$\Delta x = \Delta x \cdot \zeta, \qquad \zeta = 60\% \tag{5.6}$$

The hill climber was terminated once an overlap of the specified minimum number of pixels, n_{ol}^{min} had occurred, the tolerance was met, or the maximum number of iterations was met. The minimum number of overlap pixel was set to 10% of the total number of pixels in the array and rounded to the nearest integer. For the 64×64 array, $n_{ol}^{min} = 410$. If the current cost differenced with the previous cost was less than 0.1, the cost function was considered to be stalled out and terminated, and a maximum number of 20 iterations was allowed.

$$n_{ol}^{min} = 0.1(n_{px} \times n_{px}) \tag{5.7}$$

5.2.4 Progression of the Pre-Processing Algorithm

The goal of the pre-processing algorithm was to provide an *a priori* state to the estimation filter. Once an overlap in the measurements occurs, the estimation filters investigated can converge on a state within sub-meter accuracy.

The hill climber algorithm may be applied regardless of whether an overlap of the initial computed image and observed image occurred. The hill climber algorithm rewards a state that returns an image with the asteroid in view, while the LS algorithm uses the partial derivatives of the measurements with respect to the state to update the state estimate. Because of this, the LS algorithm has potential to move the state out of view of the asteroid and terminate the algorithm. Therefore, the hill climber was applied first to produce enough of an overlap between the observed and computed images. The minimum number of pixels for the image overlap added additional robustness to the algorithm.

The hill climber was more computationally expensive than the LS algorithm. The hill climber produced six different measurements to compare at each iteration of the algorithm, while the LS algorithm only computed one image at each iteration. Therefore, the hill climber was only used until enough of an overlap occurred to reduce the computation time.

As seen in Section 4.3, the LS algorithm was more robust to large position errors when only the position was estimated and the pointing was held fixed. The position was resolved first by only estimating the position with the LS algorithm, and then the position and pointing were resolved with the LS algorithm.

5.3 Single State Testing

A Monte Carlo (MC) simulation of 50 cases was performed to test the pre-processing algorithm in orbits around Itokawa and Bennu. These orbits were considered the “lost-in-space” orbits. It was assumed that if the spacecraft was trying to estimate its current state, its orbit might not be exactly circular or exactly in the terminator plane. It was assumed that the spacecraft was a safe distance away from the asteroid at 1 km away. The lost-in-space orbits created involved a slight inclination and eccentricity from the nominal circular terminator orbits, and involved an off-radial pointing direction determined from a normal distribution of $1\sigma = 3^{\circ}$. This off-radial pointing was assumed to be known with the $C_{ACI/SF}$ rotation matrix.

5.3.1 Itokawa

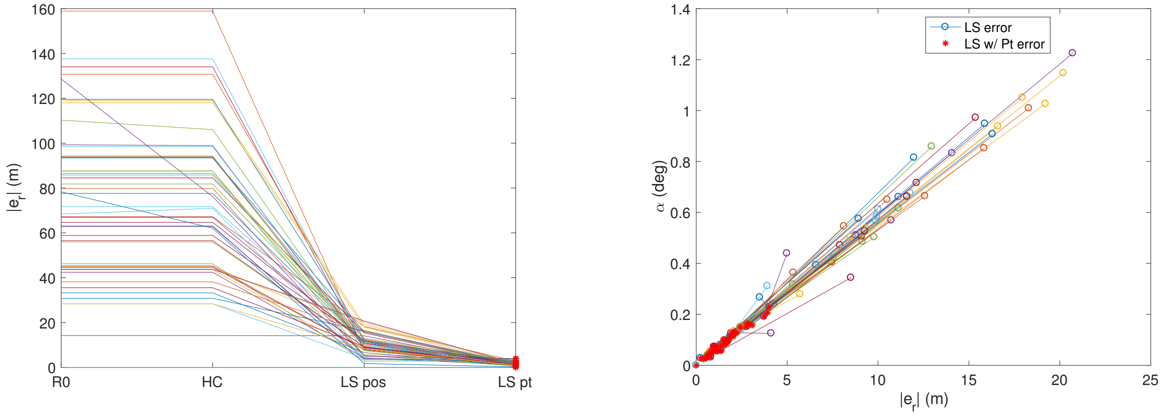
For the Itokawa orbit, the states were sampled from an orbit that had a semi-major axis, $a = 1$ km, eccentricity, $e = 0.1$, and inclination $i = 75°$. A random pointing offset error was applied to each of the observations with $1\sigma_{\theta_n} = 0.5°$. The reference radius of Itokawa was $R_s = 162$ m.

The ending errors are shown in Table 5.3, and the progression of the errors is shown in Figure 5.4. Figure 5.4(a) presents the progression of the position error through the steps of the pre-processing algorithm. In the figure, if a case contains the same position error in the $R0$ and hill climber (HC) steps, the hill climber step did not occur and the position from $\mathbf{r}_0$ was used in the LS with position step. Figure 5.4(b) presents the magnitude of the position error and the pointing error, α between the LS with position and LS with position and pointing step.

Table 5.3: RMS of the magnitude of the position error and the total pointing offset error, α of the pre-processing algorithm MC simulation with Itokawa observations.

State	Ending Error
$\texttt{rms}(\lvert\mathbf{e_r}\rvert)$	1.84 m
$\texttt{rms}(\alpha)$	0.10°

From Figure 5.4(a) only a few cases involved the hill climber step because the observed image



(a) Progression of position errors

(b) LS with position and LS with position and pointing errors

Figure 5.4: Evolution of position and pointing errors in the pre-processing algorithm when sampling from the Itokawa lost-in-space orbit.

and computed image from $\mathbf{r}_0$ did not contain an image overlap. The position errors reached tens of meters in magnitude at the LS with position step (*LS pos* in the figure), and reached meter-level accuracy after the LS with position and pointing step (*LS pt* in the figure). From Figure 5.4(b), the correction from the LS with position and pointing step decreased the position error to meter-level accuracy and the pointing error to below 0.1° in pointing.

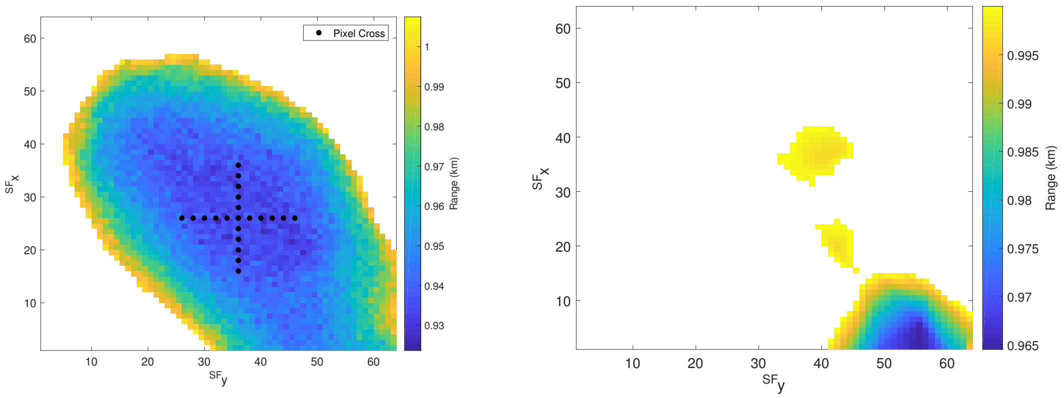
Figure 5.5 illustrates a sample of an observation and evolution of the computed measurements in the pre-processing algorithm for one of the MC cases from the Itokawa lost-in-space orbit. This was one of the more extreme cases and involved using the hill climber step. The image computed from $\mathbf{r}_0$ occupies only a fraction of the image array and does not provide enough of a pixel overlap to commence the LS with position step. However, the pre-processing algorithm produces a measurement similar to the observation and the ending error for this case was $|\mathbf{e_r}| = 2.90$ m and $\alpha = 0.16°$.

An outlier did occur in this simulation that involved ending errors of 992 m in position magnitude and $\alpha = 43.1°$. (The figures do not include this outlier.) In this case, the two LS algorithms updated the position and pointing state away from the truth. The final computed image in this case was an empty image array. In practice, a simple check of the final computed image with the observed image, such as with the metrics from the hill climber, could be applied to catch this error.

5.3.2 Bennu

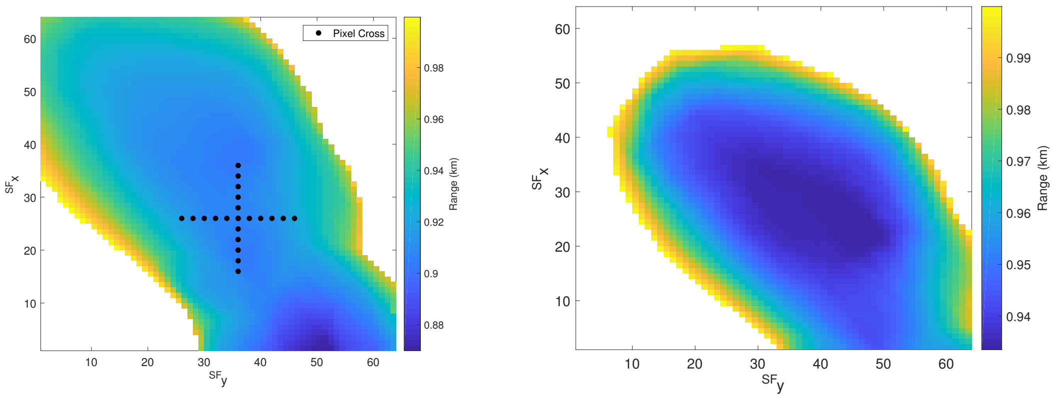
In the Bennu orbit, states were sampled from an orbit that has a semi-major axis of $a = 1$ km, eccentricity, $e = 0.3$, and inclination $i = 60°$. A random pointing offset error was applied to each of observation with $1\sigma_{\theta_n} = 0.5°$ in each Euler angle, θ_n. The reference radius for Bennu was $R_s = 246$ m.

Table 5.4 displays the RMS of the magnitude of the ending errors in position, $|\mathbf{e_r}|$ and in total pointing offset angle, α. Figure 5.6 demonstrates the evolution of the magnitude of the position error for the pre-processing steps (Figure 5.6(a)) and the difference in position and pointing error



(a) Observation

(b) G^0 from $\mathbf{r}_0$



(c) G after hill climber

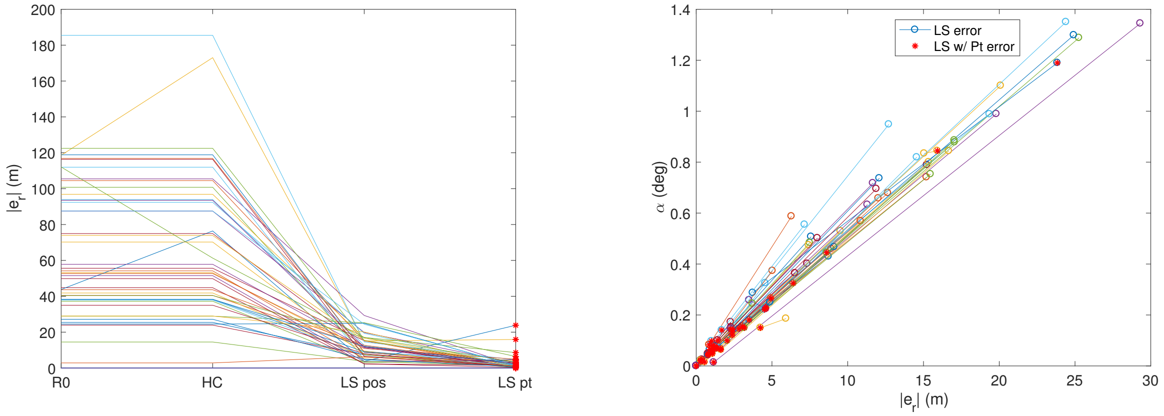
(d) Final G from $\bar{\mathbf{r}}_0$

Figure 5.5: Sample of an initial observation and the evolution of computed measurements, G through the steps of the pre-processing algorithm for an Itokawa observation.

magnitude between the LS with position step and LS with position and pointing step (Figure 5.6(b)). Similar to the Itokawa observations, the LS with position step (*LS pos*) solved for the position error to tens of meters in error magnitude (Figure 5.6(a)). Once this state was passed into the LS with position and pointing step (*LS pt*), the position errors reached meter-level accuracy, and a few cases increased slightly in position error. Figure 5.6(a) displays that the error in position

Table 5.4: RMS of the magnitude of the position error and the total pointing offset error, α of the pre-processing algorithm MC simulation with Bennu observations.

State	Ending Error
$\texttt{rms}(\lvert\mathbf{e_r}\rvert)$	4.77 m
$\texttt{rms}(\alpha)$	0.24°



(a) Progression of position errors

(b) Errors from the LS with position and the LS with position and pointing

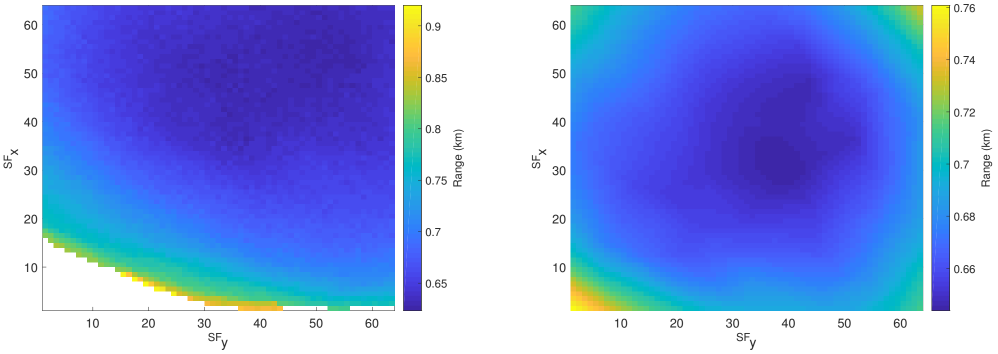
Figure 5.6: Evolution of position and pointing errors in the pre-processing algorithm when sampling from the Bennu lost-in-space orbit.

decreased to meter-level accuracy for the majority of the cases between the LS with position step and the LS with position and pointing step. The pointing errors were decreased to below 0.2° for the majority of the cases.

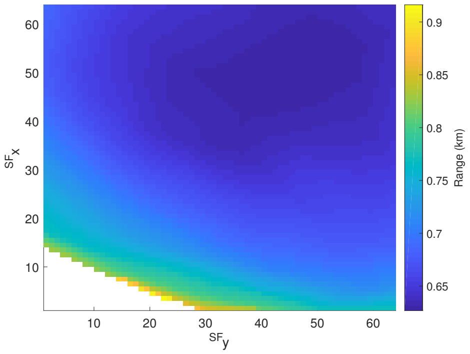
Figure 5.7 displays an observation from the Bennu lost-in-space orbit that fills almost the entirety of the image array, and the evolution of the computed measurements. The observation and initial computed measurement contained enough of an image overlap and were directly passed into the LS with position step. After convergence of the pre-processing algorithm, the ending errors in this case were $|\mathbf{e_r}| = 0.31$ m and $\alpha = 0.016°$.

5.3.3 Analysis

The pre-processing algorithm for the Itokawa and Bennu observations produced meter-level accuracy in position and below 0.3° in total pointing error. The goal of this algorithm was to provide an initial position and pointing estimate for the filter, and thus an accurate state estimate was achieved. As observations are accumulated over time, more information becomes available, and the estimation state is futher refined in the estimation filter.



(a) Observation (b) G^0 from $\mathbf{r}_0$



(c) Final G from $\bar{\mathbf{r}}_0$

Figure 5.7: Sample of an initial observation and the evolution of computed measurements, G through the steps of the pre-processing algorithm for a Bennu observation.

5.4 Initializing the Filter

With the pre-processing step, one can solve for the position and pointing with the flash lidar image. This provided information on the position and pointing state at one instant in time, but information on the velocity must be accumulated over time. The approach outlined here relied on the asteroid occupying the camera field of view, and taking a high frequency of measurements (one every second for 30 seconds) with the inertial pointing fixed. From these high frequency measurements, the pre-processing script was applied to two observation times, and two position and pointing states were determined. An estimate of the velocity was determined from the two positions, and thus an *a priori* state was formed to start the filter.

The procedure for initializing the filter is illustrated in Figure 5.8. Once an observation with the asteroid in the FOV was acquired, high frequency observations were acquired once every second for 30 seconds. The pre-processing algorithm was applied for the first observation at t_1, and for a second observation at t_2 from the high frequency observations. The result was two *a priori* states: $[\bar{\mathbf{r}}_1,\ \underline{\bar{\theta}}_1]$ and $[\bar{\mathbf{r}}_2,\ \underline{\bar{\theta}}_2]$, and their corresponding covariance matrices: $\bar{P}_1$ and $\bar{P}_2$ that were determined from the LS algorithm. From the two position estimates, an *a priori* velocity, $\mathbf{v}_1$ was determined and its corresponding covariance, $\bar{P}_{\mathbf{v}_1}$. The LS filter was then applied with $[\bar{\mathbf{r}}_1,\ \mathbf{v}_1,\ \underline{\bar{\theta}}_1]$ as the *a priori* state with *a priori* covariance determined from $\bar{P}_1$ and $\bar{P}_{\mathbf{v}_1}$.

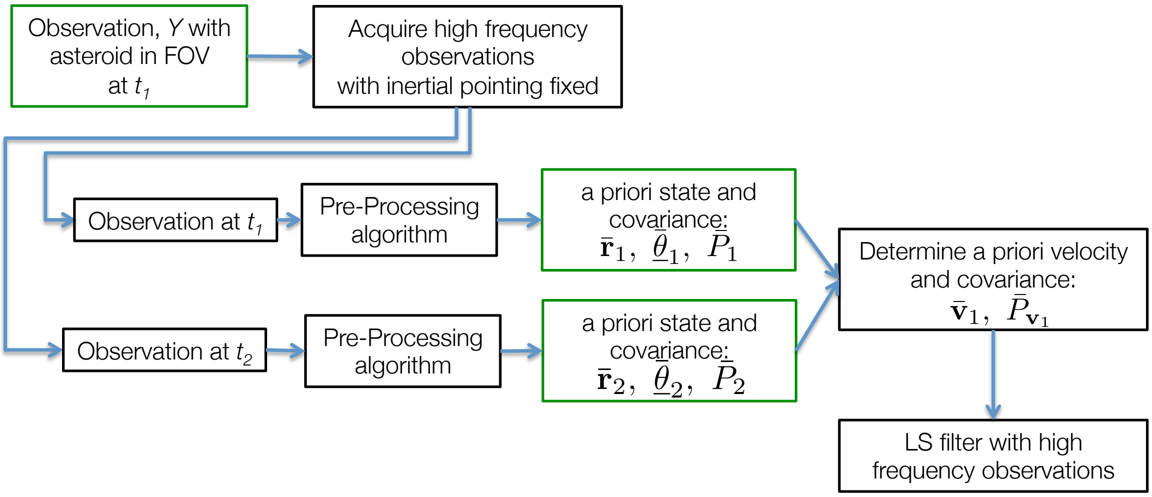


Figure 5.8: Flow chart of steps to initialize the filter with the pre-processing algorithm.

The *a priori* velocity was determined from a linear difference of the position vectors divided by the time:

$$\bar{\mathbf{v}}_1 = \frac{\bar{\mathbf{r}}_2 - \bar{\mathbf{r}}_1}{t_2 - t_1} \tag{5.8}$$

The covariance for the velocity was determined by substituting Equation 5.8 into the definition of the covariance, $P = E[(\hat{x} - x)(\hat{x} - x)^T]$:

$$\begin{aligned} P_{\mathbf{v}_1} =& E[(\hat{\mathbf{v}}_1 - \mathbf{v}_1)(\hat{\mathbf{v}}_1 - \mathbf{v}_1)^T] \\ =& E\left[\left(\frac{\hat{\mathbf{r}}_2 - \hat{\mathbf{r}}_1}{t_2 - t_1} - \frac{\mathbf{r}_2 - \mathbf{r}_1}{t_2 - t_1}\right)\left(\frac{\hat{\mathbf{r}}_2 - \hat{\mathbf{r}}_1}{t_2 - t_1} - \frac{\mathbf{r}_2 - \mathbf{r}_1}{t_2 - t_1}\right)^T\right] \end{aligned} \tag{5.9}$$

Evaluating the expected values, and assuming that $\mathbf{r}_1$ and $\mathbf{r}_2$ were independent (i.e. $E[(\hat{\mathbf{r}}_2 - \mathbf{r}_2)(\hat{\mathbf{r}}_1 -$

$\mathbf{r}_1)^T] = 0)$, the resulting covariance was:

$$\begin{aligned} P_{\mathbf{v}_1} =& \frac{1}{(t_2 - t_1)^2} \left(E[(\hat{\mathbf{r}_2} - \mathbf{r}_2)(\hat{\mathbf{r}_2} - \mathbf{r}_2)^T] + E[(\hat{\mathbf{r}_1} - \mathbf{r}_1)(\hat{\mathbf{r}_1} - \mathbf{r}_1)^T] \right) \\ =& \frac{1}{(t_2 - t_1)^2} (P_{\mathbf{r}_1} + P_{\mathbf{r}_2}) \end{aligned} \tag{5.10}$$

The *a priori* covariance for the velocity at t_1 was defined as:

$$\bar{P}_{\mathbf{v}_1} = \frac{1}{(t_2 - t_1)^2} (\bar{P}_{\mathbf{r}_1} + \bar{P}_{\mathbf{r}_2}) \tag{5.11}$$

The *a priori* covariance in the filter was a diagonal matrix composed of the diagonal components from the respective covariance matrices of $\bar{P}_{\mathbf{r}_1}$, $\bar{P}_{\mathbf{v}_1}$, and $\bar{P}_{\underline{\theta}_1}$. The values of each covariance component were multiplied by 10 to inflate them for the filter initialization.

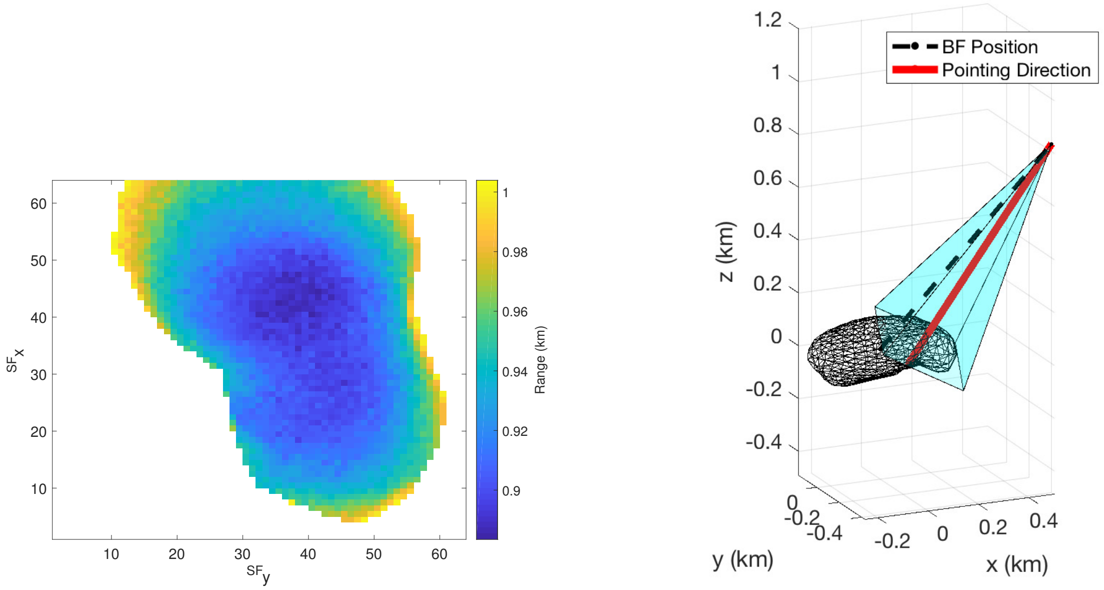
5.4.1 Filter Results

A truth state was sampled from the Itokawa lost-in-space orbit that contained an off-radial pointing direction and the asteroid in its FOV. The truth measurements were then created by propagating this truth state and capturing measurements every one second for 30 seconds while keeping the inertial pointing fixed. The ranges contained white Gaussian noise based on Equation 3.1, and an off-nominal pointing bias was applied to each Euler angle of 0.5°. The *a priori* state and covariance were determined from the pre-processing algorithm and the steps outlined above. Process noise was added based on Equation 3.3, and the LS filter was used for the state estimation.

The truth observation sampled from the Itokawa orbit is shown in Figure 5.9, and contained an off-radial pointing direction of 6.5°. In this simulation the observation at t_2 was set at 5 second after the observation at t_1. Table 5.5 displays the errors for position and pointing at $\bar{\mathbf{r}}_1$ and $\bar{\mathbf{r}}_2$, and the error in the *a priori* velocity. Each position state was resolved to about 1 m in accuracy, and

Table 5.5: Progression of state errors through the filter initialization steps.

State	$\lvert\mathbf{e_r}\rvert$ (m)	α (deg)	$\lvert\mathbf{e_v}\rvert$ (cm/s)
$\bar{\mathbf{r}}_1$	1.05	0.050	–
$\bar{\mathbf{r}}_2$	1.64	0.085	–
$\bar{\mathbf{v}}_1$	–	–	66.45



(a) Observation at t_1

(b) Body-fixed position and pointing at t_1

Figure 5.9: Observation and truth position and pointing state at t_1 to initialize the filter.

less than 0.1° accuracy in pointing. While the initial velocity error was significant, the covariance bounds were inflated to capture this error.

Figure 5.10 illustrates the state errors in the RIC frame (solid lines) and their respective 3σ covariance bounds. Table 5.6 illustrates the RMS of the magnitude of the state errors through the entire simulation and excluding the first five observations.

Table 5.6: RMS errors of the magnitude of the position and velocity errors and total pointing error angle, α over the full simulation, excluding the first 5 observations (†), and the run time of the filter. T.

Filter	$\lvert\mathbf{e_r}\rvert$ (m)	$\lvert\mathbf{e_v}\rvert$ (cm/s)	α (deg)	$\lvert\mathbf{e_r}\rvert$† (m)	$\lvert\mathbf{e_v}\rvert$† (cm/s)	α† (deg)	Run Time (min)
LS	0.879	14.57	0.050	0.738	3.20	0.044	1.7

The errors for the position and pointing were below 1 m and 0.1° for the position and pointing, respectively for the entirety of the simulation. The state errors for the position and pointing were also within the covariance bounds during the simulation. The velocity errors were larger and required a few observations to be resolved. By the end of the observation arc, the velocity

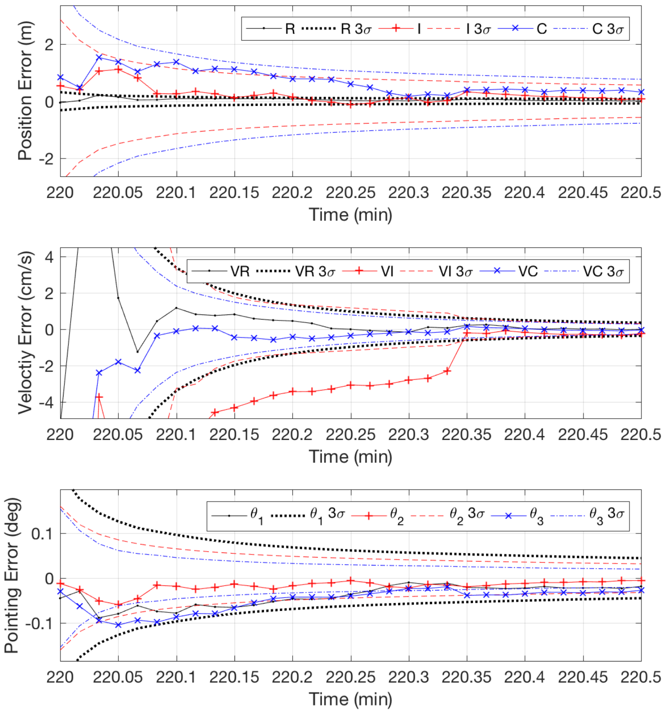


Figure 5.10: Errors (solid lines) and 3σ covariance bounds (dashed lines) in the RIC frame with process noise for the filter initialization simulation with an Itokawa observation.

component errors were within their respective covariance bounds and the ending magnitude of the velocity error was $|\mathbf{e_v}| = 0.25$ cm/s.

5.4.2 Discussion

The pre-processing algorithm and the filter initialization procedure have a number of tuning parameters. In the filter initialization, the observation frequency and duration of the high frequency observations may be adjusted. The two position states to determine the velocity may be adjusted, and the velocity may be determined with a more sophisticated method, such as a Lambert solver. Determining the velocity was studied by using a Lambert solver, yet because the time span was small, it produced similar results to the linear approximation.

Chapter 6

Increasing Filter Speed

In each of the previous simulations, the full flash lidar array is processed, and each pixel in that array is considered an individual measurement. Using the 64×64 flash lidar array entails that a maximum of 4,096 range measurements may be processed at each observation time. With the Itokawa measurements, this is usually around 2,000 range measurements per observation because the asteroid does not occupy the full image array. However, with Bennu, the full image array is populated in the majority of the terminator orbit measurements, and approximately 4,000 range measurements are processed at each time step.

This section looks to reduce the number of range measurements processed at each time step. This entails a smaller measurement mapping matrix in the filter, as well as fewer computations of the ray-tracing algorithm for the computed measurements. This is performed by selecting pixels that maximize the information returned. Thus, a reduced number of measurements is selected that maximizes the information content in each observation and processed in the filter.

6.1 Information of the Flash Lidar Image

The information of the observation was determined from the Fisher Information Matrix (FIM) that was overviewed in Section 2.6.5, and is reproduced here at time t_k:

$$\Lambda_k = \bar{P}_k^{-1} + \tilde{H}_k^T R_k^{-1} \tilde{H}_k \tag{6.1}$$

where $\bar{P}_k = \phi_{k-1} P_{k-1} \phi_{k-1}^T$ and is the *a priori* covariance matrix, $\tilde{H}_k$ is the measurement mapping matrix and R_k^{-1} is the measurement uncertainty matrix. Without *a priori* covariance knowledge, the *a priori* covariance term $\bar{P}_k$ simply drops out and the FIM becomes the second term in the Equation 6.1:

$$\Lambda_k = \tilde{H}_k^T R_k^{-1} \tilde{H}_k \tag{6.2}$$

This work studied the FIM without *a priori* information, and only considered the information contained in one observation at one time. Equation 6.2 was therefore the basis of the FIM studies.

The determinant of the FIM was used to quantify the amount of information contained in each observation, and is known as the D-optimality condition:

$$|\Lambda_k| = |\tilde{H}_k^T R_k^{-1} \tilde{H}_k| \tag{6.3}$$

This quantity has been used in other work [75, 42] also to quantify the information content. The derivation of the information for a flash lidar image was presented in Section 2.6.5, and an overview of relevant equations is reproduced here.

The partial derivative of one range measurement, ρ_i with respect to the state X at one time step. t_k was defined as the measurement mapping matrix, $\tilde{H}_i$. The measurement mapping matrix for the full flash lidar array with p range measurements was the combination of the individual $\tilde{H}_i$ matrices:

$$\tilde{H}_k = \begin{bmatrix} \tilde{H}_1 \\ \vdots \\ \tilde{H}_i \\ \vdots \\ \tilde{H}_p \end{bmatrix}_{p\times 9} \tag{2.164}$$

The uncertainty of the range measurements were defined as σ_{ρ_i} and populated the measurement uncertainty matrix, R_ρ. Each pixel was treated as an individual range measurement, creating a diagonal measurement uncertainty matrix:

$$R_\rho = \mathtt{diag}[\sigma_{\rho 1}^2, \cdots, \sigma_{\rho i}^2, \cdots, \sigma_{\rho p}^2] \tag{2.171}$$

The inverse of the measurement uncertainty matrix, R_ρ^{-1} was a diagonal of the values, $1/\sigma_{\rho i}^2$.

6.1.1 Position Only

For one lidar beam, the measurement mapping matrix with respect to position was:

$$^{\mathbf{r}}\tilde{H}_i = \frac{\partial \rho_i}{\partial \mathbf{r}} = \frac{-\hat{n}_i^T}{\hat{u}_i \cdot \hat{n}_i} \tag{2.146}$$

The vector, $\hat{n}_i$ is the surface normal vector where the altimetry measurement intersects the body, and $\hat{u}_i$ is the pointing direction. The information of the full flash lidar array with respect to position and without *a priori* information is: The combination of

$$^{\mathbf{r}}\Lambda_p = \sum_{i=1}^{p} \frac{1}{\sigma_{\rho i}^2} \frac{\hat{n}_i \hat{n}_i^T}{(\hat{u}_i \cdot \hat{n}_i)^2} \tag{2.173}$$

6.1.2 Position and Pointing

The partial derivative of the range measurement, ρ_i with respect to pointing was:

$$^{\theta_n}\tilde{H}_i = \frac{\partial \rho_i}{\partial \theta_n} = \frac{-\rho_{ic}}{\hat{u}_i \cdot \hat{n}_i} (M_{\theta_n} \hat{u}_i^*)^T \hat{n}_i \tag{2.147}$$

where M_{θ_n} was defined in Equations 2.148, 2.149, and 2.150 for the corresponding Euler angle. The FIM with respect to position and pointing for the full flash lidar image was:

$$\Lambda_i = \bar{P}_0^{-1} + \begin{bmatrix} \sum_{i=1}^{p} \frac{1}{\sigma_{\rho i}^2} \frac{\hat{n}_i \hat{n}_i^T}{(\hat{u}_i \cdot \hat{n}_i)^2} & \sum_{i=1}^{p} \frac{1}{\sigma_{\rho i}^2} \frac{\rho_{ic}}{(\hat{u}_i \cdot \hat{n}_i)^2} \begin{bmatrix} M_{\theta_1} \hat{u}_i^* & M_{\theta_2} \hat{u}_i^* & M_{\theta_3} \hat{u}_i^* \end{bmatrix} \\ \sum_{i=1}^{p} \frac{1}{\sigma_{\rho i}^2} \frac{\rho_{ic}}{(\hat{u}_i \cdot \hat{n}_i)^2} \begin{bmatrix} M_{\theta_1} \hat{u}_i^* \\ M_{\theta_2} \hat{u}_i^* \\ M_{\theta_3} \hat{u}_i^* \end{bmatrix} \hat{n}_i \hat{n}_i^T & \sum_{i=1}^{p} \frac{1}{\sigma_{\rho i}^2} \frac{\rho_{ic}}{(\hat{u}_i \cdot \hat{n}_i)^2} \begin{bmatrix} M_{\theta_1} \hat{u}_i^* \\ M_{\theta_2} \hat{u}_i^* \\ M_{\theta_3} \hat{u}_i^* \end{bmatrix} \hat{n}_i \hat{n}_i^T \begin{bmatrix} M_{\theta_1} \hat{u}_i^* & M_{\theta_2} \hat{u}_i^* & M_{\theta_3} \hat{u}_i^* \end{bmatrix} \end{bmatrix} \tag{2.177}$$

6.2 Maximum Information Pixel Subsets of a Single Image

This analysis studied the pixel combinations that maximized the information of one observation image. The determinant of the FIM was computed for different combinations of pixels within one observation image. The top 100 combinations were saved, and the locations of these pixel combinations were recorded. This was performed for maximizing the information for position only and for position and pointing. The observation chosen to perform this analysis was taken of Itokawa, when the spacecraft was approximately 1 km away from Itokawa.

6.2.1 Maximizing the Position Information

Three pixels were required to resolve the position state. This image contained $p = 2535$ pixels, and each pixel in the image was combined in a three-pixel combintaion to calculate the determinant of the FIM from Equation 2.173. This involved ${}_{2535}C_3 = 2.71\text{E}9$ combinations. The 100 three-pixel combinations that returned the largest value of the determinant of the FIM were recorded. These were considered the top three-pixel combinations that maximized the information content in this image.

Figure 6.1 illustrates the locations of the pixels that composed these top three-pixel combinations. Some pixels were included in more than one combination, and Figure 6.2 illustrates a histogram of the pixel indices that occurred in the top three-pixel combinations.

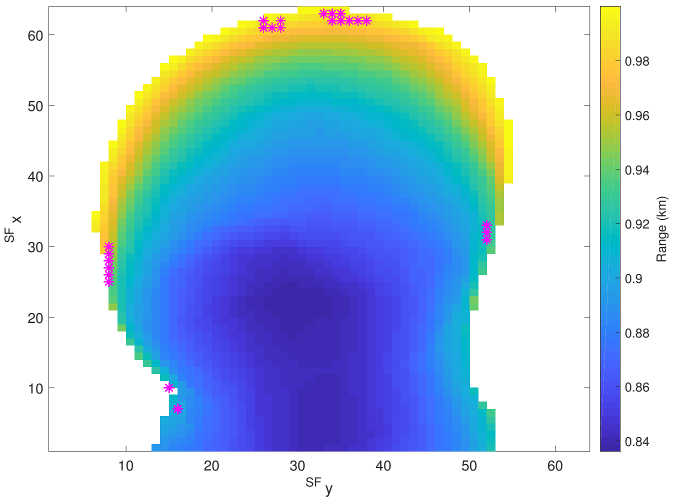


Figure 6.1: Pixels that composed the top three-pixel combinations for maximizing the position information.

The pixels that composed the three-pixel combinations with the most information occurred on the edges of the image. Only 24 pixels of the image appeared in the top information combinations.

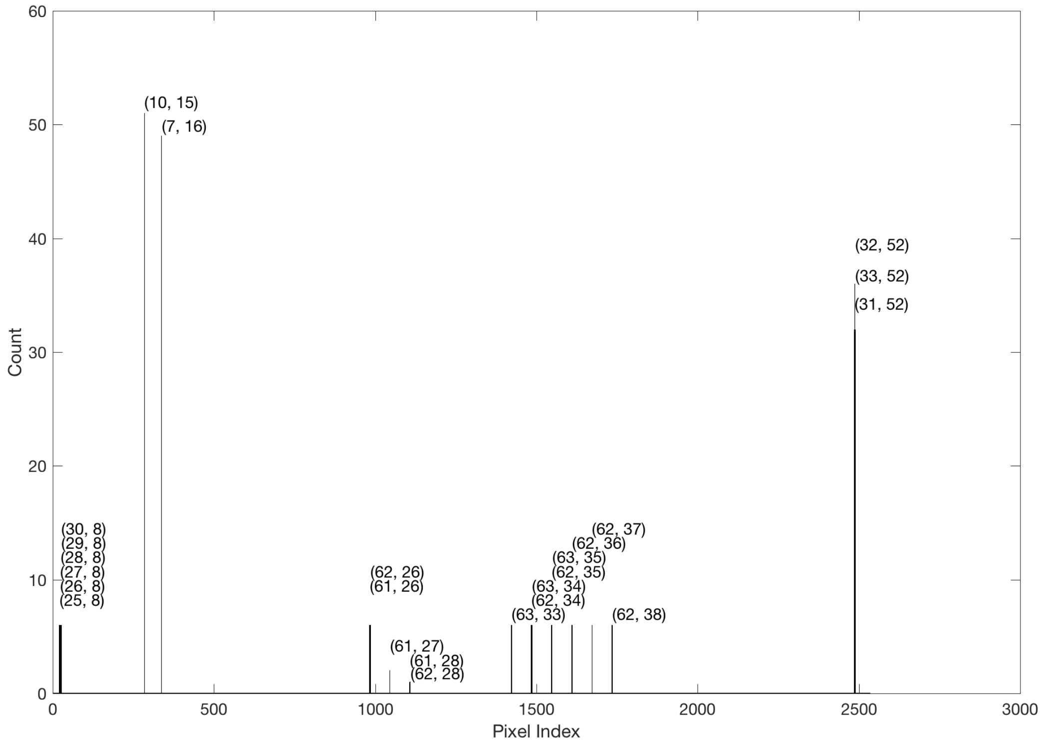


Figure 6.2: Histogram of pixels that occurred in a top three-pixel combination with the bins labeled as the pixel coordinates in the sensor frame (SF) for maximizing the position information.

From Figure 6.2, those pixels that occurred in the most combinations were the pixels at $(10, 15)$ and $(7, 16)$ in the sensor frame array (Figure 6.1).

Following the analytical analysis in Section 2.6.5, the maximum information occurs with the pointing direction of the lidar beam and the surface normal of the facet it intersects are perpendicular to each other. In addition, the information is further maximized when the surface normal vectors from the corresponding lidar beams are all perpendicular to each other. This idealized situation, however, is difficult to realize in practice. For three lidar beams that originate from one point and are directed at a concave body, the maximum information is achieved when the pointing directions of the lidar beams are spread toward the limb of the surface. This theory is supported by the numerical results in Figure 6.1 and previous work [75].

6.2.2 Maximizing Pointing and Position Information

Determining the position and pointing state required at least six pixels. Testing each six-pixel combination in an image would require testing ${}_pC_6$ number of combinations. For the image above, $p = 2535$; therefore, ${}_{2535}C_6 = 3.7\text{E}17$ number of combinations would be required for testing the full image. To reduce the computational load of this analysis, 200 pixels from the image were randomly selected to be tested. With 200 pixels, the number of combinations was ${}_{200}C_6 = 8.24\text{E}10$.

The analysis cycled through the randomly selected pixels and determined the determinant of the FIM from Equation 2.177 for a six-pixel combination. The top 100 combinations that contained the largest value of the determinant of the FIM were recorded.

Figure 6.3 illustrates the 200 randomly selected pixels in the image (black) and the pixels that composed the top six-pixel combinations. Figure 6.4 illustrates the histogram of the pixels that were involved in the top six-pixel combinations, and their indices in the sensor frame (SF) in Figure 6.3.

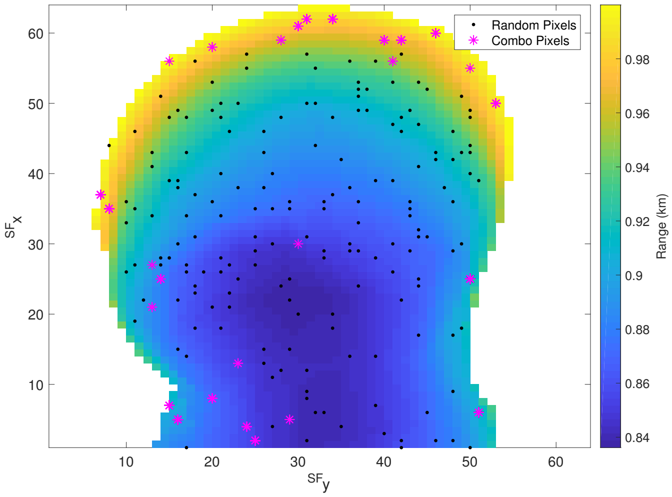


Figure 6.3: Pixels that composed the top six-pixel combinations (pink) out of the 200 randomly selected pixels (black) for maximizing the position and pointing information.

Similar to maximizing only the position information, the pixels that composed the combinations with the most information occurred near the edges of the image. One pixel in the center of

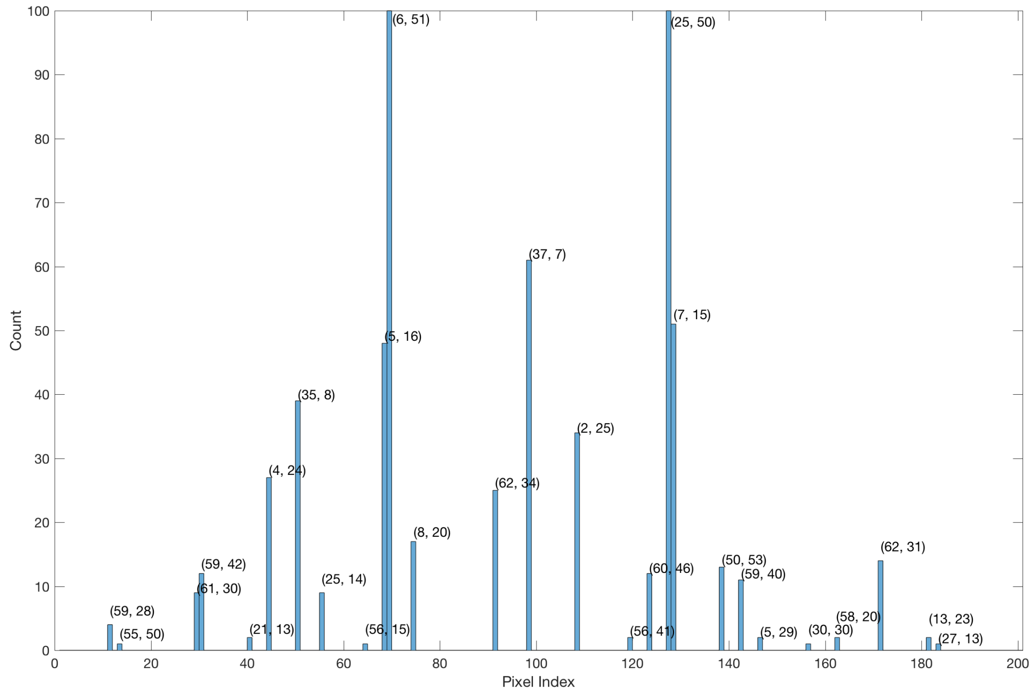


Figure 6.4: Histogram of pixels that occurred in a top six-pixel combination with the bins labeled as the pixel coordinates in the sensor frame (SF) for maximizing the position and pointing information.

the image was involved in a top combination, yet only occurred in one instance. From Figure 6.4, two pixels, $(6, 51)$ and $(25, 50)$ occurred in all of the top combinations, and both were located on the right edge of the image.

6.3 Pre-Determined Pixel Patterns

The optimization analysis of determining the combination of pixels that maximize the information for each observation becomes tedious. The goal of this study is to reduce the computational load of processing a full flash lidar array in the filter by optimally reducing the number of measurements to process. From the numerical analysis of one observation image, the pixels that contain the most information on the position and pointing state occur near the edges of the image. Instead of performing an optimization step to determine the maximum-information set of pixels at each observation, patterns within the flash lidar array are pre-determined based on this numerical analysis.

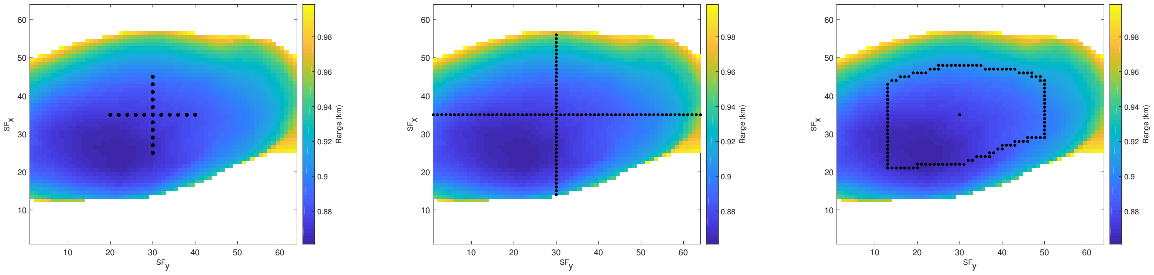
Various patterns are tested for their information content and measurement computation time, and analyzed for robustness. Choosing to process only the edge pixels of an observation image may provide maximum information, however, this pattern may not be robust because the same pixels may not be present in the computed image. The pre-determined pixel patterns include a cross pattern for robustness, a limb pattern based on the edge pixels in the observed image, and a random selection of the pixels that contain a range-return in the observed image.

6.3.1 Definition of Pixel Patterns

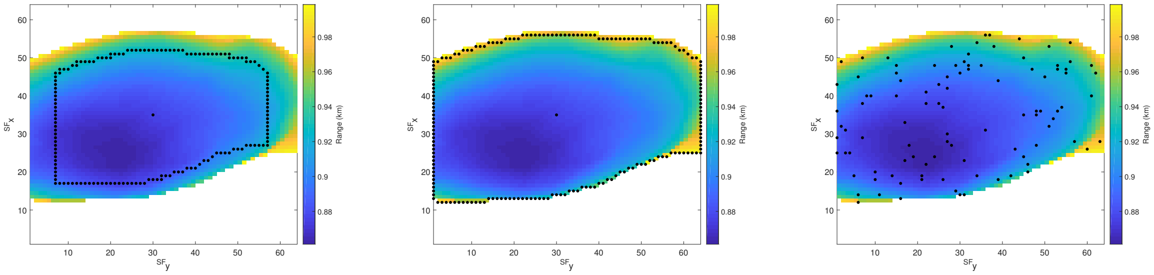
Six overall patterns were investigated, and each one included the center pixel in the pattern, with the exception of the random selection. The center pixel was determined from the Neighbor Search edge detection algorithm in Section 5.1. An observation from the Itokawa terminator orbit was used to display the patterns, and the pre-determined patterns are illustrated in Figure 6.5.

Two cross patterns were investigated: one that expanded from the center pixel to every other pixel for ten pixels in the ${}^{SF}x$ and ${}^{SF}y$ directions (*Cross*, Figure 6.5(a)), and one that spanned the entirety of the ${}^{SF}x$ and ${}^{SF}y$ directions and was centered on the center pixel (*Full Cross*, Figure 6.5(b)). Because the cross patterns were centered in the image, they were considered to be robust to producing an image overlap of the selected pixels.

Three patterns involved the limb of the asteroid and were determined from the edge pixels of the image. The ${}^{SF}x$ and ${}^{SF}y$ coordinates of the edge pixels were subtracted from the those of the center pixel. The difference was multiplied by 60% (60% *Limb*, Figure 6.5(c)) and by 80% (80% Limb, Figure 6.5(d)), and rounded to the nearest integer. The pixels were not double-counted if this subtraction produced a pixel index more than once. The edge pixels also were investigated as a pattern (*Edges*, Figure 6.5(e)). These limb patterns were chosen to maximize the information by expanding the pointing directions toward the limb. Selecting the pixels at 60% and 80% of the difference from the center to the limb increased the probability of an overlap of the selected pixels



(a) Cross (b) Full Cross (c) 60% Limb



(d) 80% Limb (e) Edges (f) Random

Figure 6.5: Pre-determined pixel patterns

in the observed and computed images. By only using the edge pixels, this introduced a higher probability that those selected pixels would not be available in both the observed and computed images.

A random sample pattern also was investigated. This sampled 100 pixels from those pixels that produced a range-return in the observed image. Figure 6.5(f) illustrates an example of the randomly selected pixels, but it is noted that a new random sample was created each time the observation was processed.

6.3.2 Comparison of Pixel Patterns

Figure 6.6 compares the information content of each of the patterns in Figure 6.5, as well as the maximum information from the optimal six-pixel combination from the numerical simulation, and the information of the full image. The information of each pattern was determined with the determinant of the FIM from Equation 6.3. The figure presents the information on a log-scale and the number of pixel used to calculate the information.

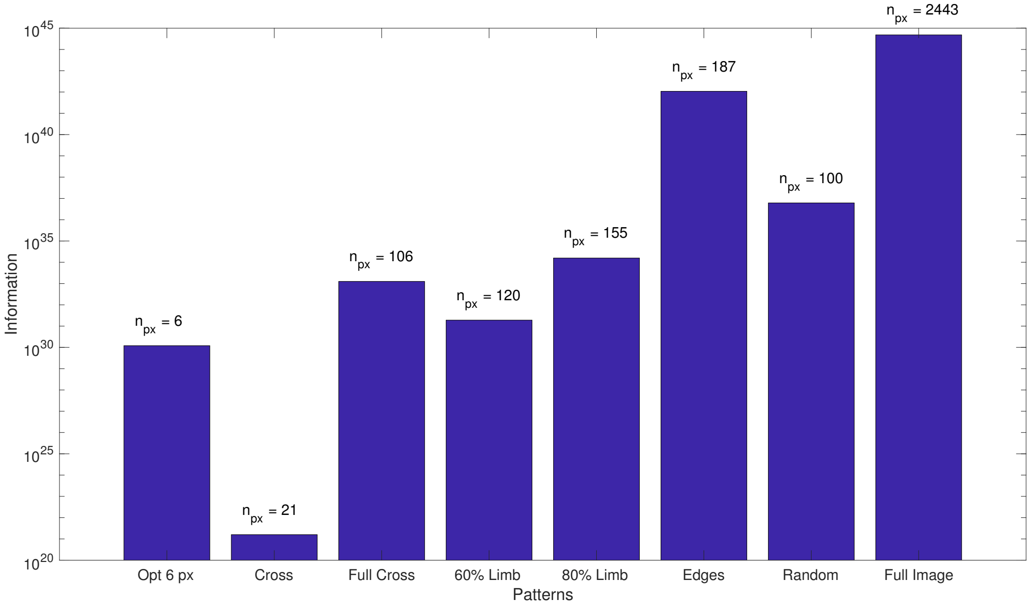


Figure 6.6: Information content of the pre-determined patterns in Figure 6.5, the optimal six-pixel combination, and the full image, as well as the number of pixels used for each calculation.

Out of the pre-determined patterns, the Edges pattern contained the most information. The Random pattern involved the next highest information, followed by the 80% Limb pattern, yet the Random pattern involved a lower number of pixels than the 80% Limb pattern. This was likely due to the few pixels that were selected near the limb of the asteroid in the Random pattern that increased its information content.

Table 6.1 displays the time of to read an observation, determines the pixel pattern (through edge detection and the center pixel), and calculate the computed measurement for each of the pre-determined patterns and the full image.

The Cross pattern required the least amount of time to process, yet also contained the least amount of information. The Random pattern involved the second to lowest processing time and contained the second highest information content of the pre-determined patterns. The Edges pattern and the 80% Limb pattern contained comparable processing time, and the Full Cross and 60% Limb patterns required comparable processing time.

Table 6.1: Timing of processing the observation to determine the pixel pattern, and calculating the computed measurement in milliseconds.

Pattern	Processing Time (ms)
Cross	2.24
Full Cross	8.42
60% Limb	8.97
80% Limb	11.08
Edges	12.70
Random	6.94
Full Image	250.24

6.4 Filtering with Pre-Determined Pixel Patterns

Analyzing the information content and the processing time required of the different patterns, four patterns were tested with the three estimation filters: Full Cross, 80% Limb, Edges, and Random patterns. The Full Cross and Random patterns were most likely to contain the selected pixels in both the observed and computed images because they involved pixels near the center of the image. The Edges pattern contained the most information, yet the pixels selected in the observed image may not be present in the computed image. The 80% Limb pattern attempts to increase the robustness to this by selecting pixels just inside the edges while still maximizing the information.

The patterns were applied for OD in the 1 km circular terminator orbit around Itokawa with measurements every 10 minutes for 30 hours. The estimated state involved the position, velocity, and off-nominal pointing error of the spacecraft. Errors included *a priori* state errors based on the covariance in Equation 3.9 for position and velocity, an off-nominal pointing bias of 0.5° on each Euler angle, and white Gaussian noise on the range measurements based on Equation 3.1. The *a priori* covariance was that in Equation 3.9 and process noise was added from Equation 3.3. The *a priori* Euler angle state was set to zero, and the same *a priori* state and measurements were used in each simulation.

6.4.1 Results

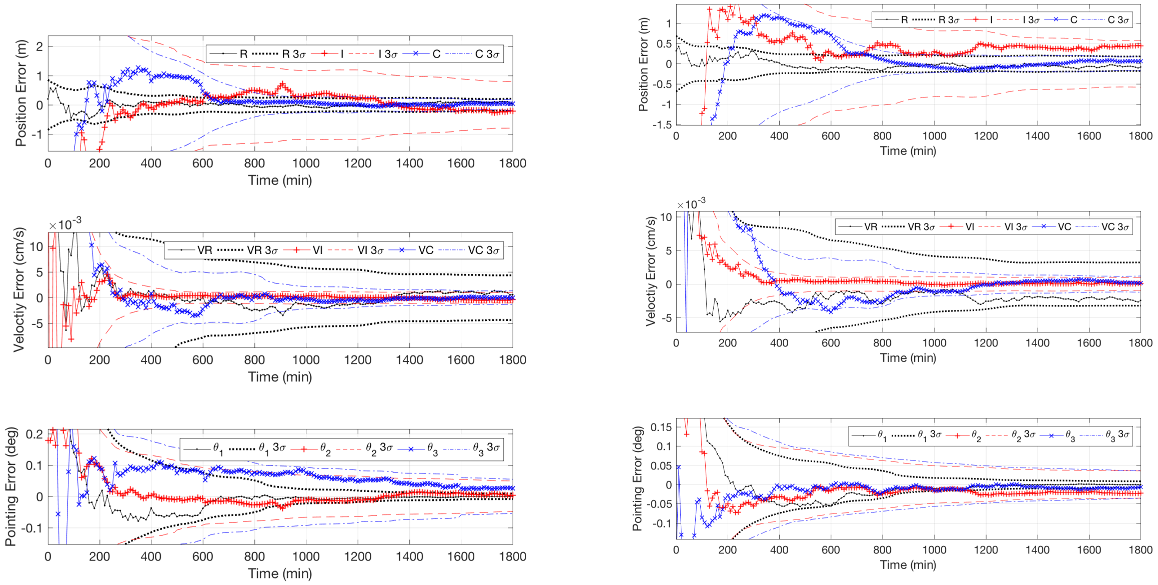
The state errors and 3σ covariance bounds for each of the four pixel patterns are presented in Figure 6.7 for the EKF, Figure 6.8 for the UKF, and Figure 6.9 for the LS filter. Table 6.2 displays the RMS of the magnitude of the errors and filter run time for each pixel pattern with the three estimation filters, as well as the full flash lidar image for comparison.

Table 6.2: RMS of the magnitude of the position and velocity errors, and the total pointing offset error of the different patterns and filtering methods for the entirety of the simulation. This was performed in the Itokawa terminator orbit with measurements every 10 minutes for 30 hours.

Pattern	Filter	$\lvert\mathbf{e}\rvert_{\mathbf{r}}$ (m)	$\lvert\mathbf{e}\rvert_{\mathbf{v}}$ (cm/s)	$\alpha(deg)$	Run Time (min)
Full Cross	EKF	2.44	0.0282	0.12	0.47
	UKF	2.69	0.0272	0.16	0.63
	LS	2.48	0.0232	0.13	0.40
80% Limb	EKF	2.24	0.0268	0.11	0.54
	UKF	1.42	0.0202	0.06	0.83
	LS	1.34	0.0216	0.07	0.47
Edges	EKF	10.80	0.0509	0.57	0.59
	UKF	1.63	0.0225	0.09	0.94
	LS	0.72	0.0131	0.04	0.73
Random	EKF	5.98	0.0318	0.33	0.47
	UKF	1.35	0.0232	0.07	0.61
	LS	2.00	0.0196	0.10	0.39
Full Image	EKF	6.20	0.0365	0.32	7.31
	UKF	1.85	0.0167	0.10	18.56
	LS	0.31	0.0127	0.02	11.67

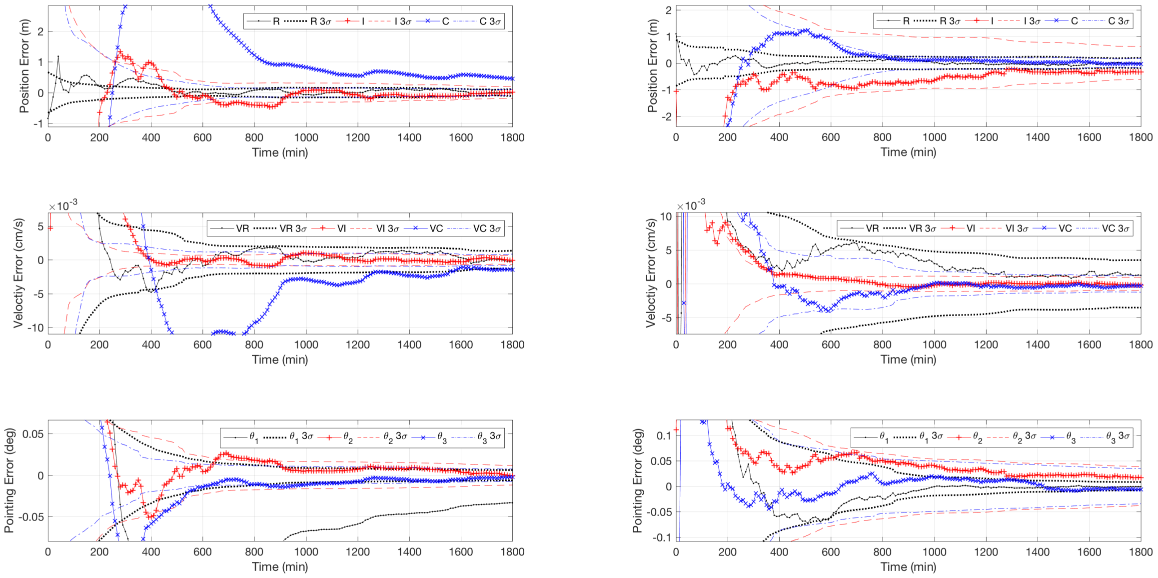
In each of the pixel pattern filter simulation, the state errors are bounded by the 3σ covariance bounds. The one exception is for the Edges pattern in the EKF. Due to the reduced number of measurements processed per observation step, the covariance does not shrink before the state errors were reduced and filter saturation did not occur.

With each of the patterns, the UKF requires the longest run time of the three filters. The run times of the EKF and LS filter are comparable in each of the pattern simulations. In terms of accuracy, the errors from the LS filter and the UKF are comparable, and are less than those from the EKF across all of the patterns except the Full Cross pattern. With the Full Cross pattern, the accuracy of the three filters is similar.



(a) Full Cross

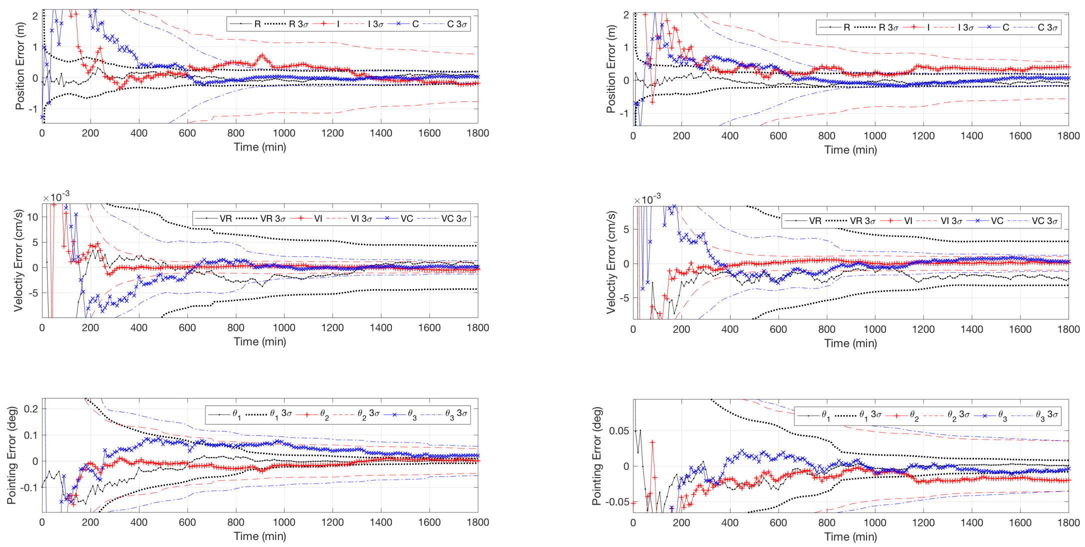
(b) 80% Limb



(c) Edges

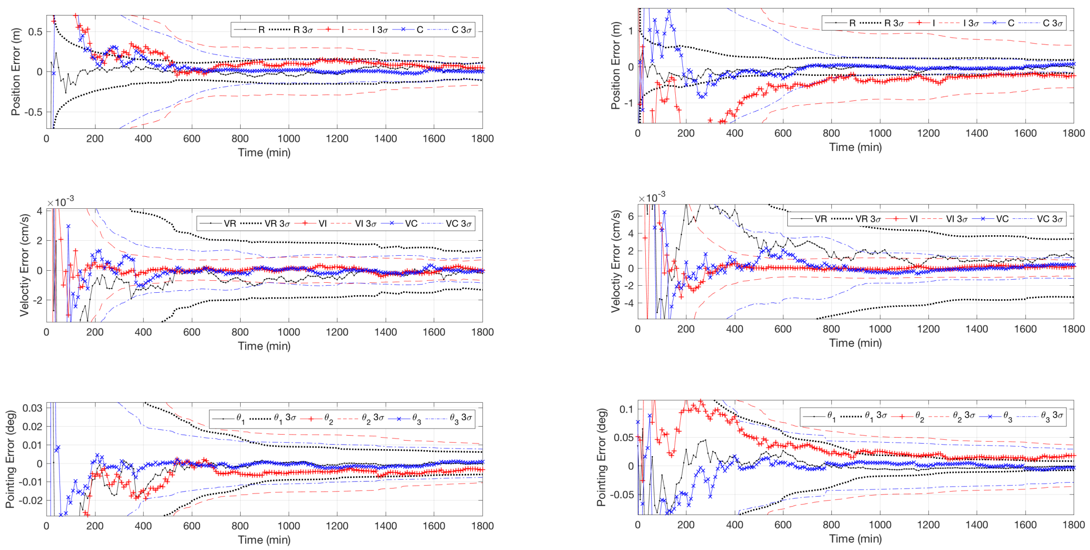
(d) Random

Figure 6.7: State errors in the RIC frame and 3σ covariance bounds in the Itokawa terminator orbit with measurements every 10 minutes for 30 hours, with the EKF.



(a) Full Cross

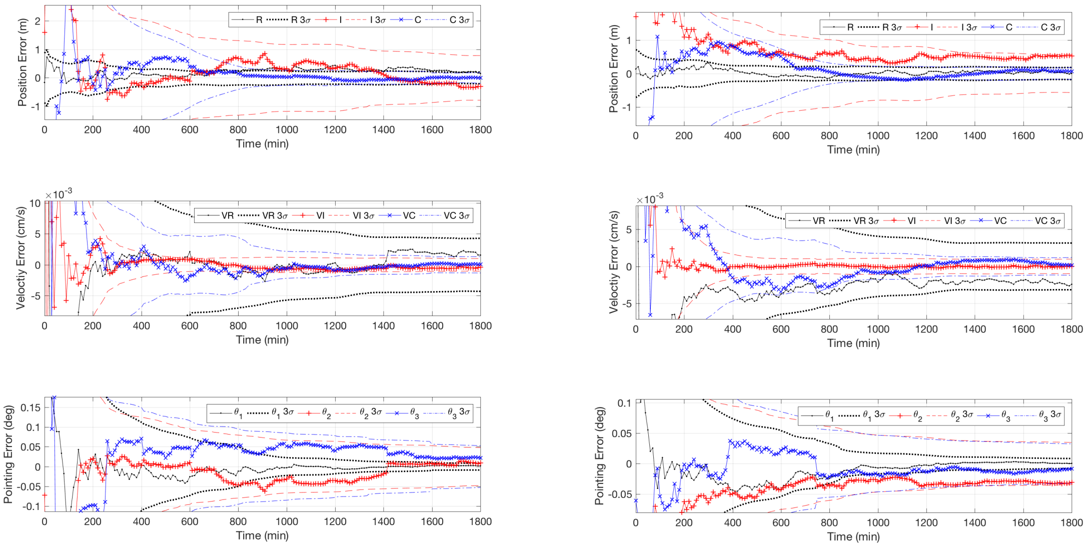
(b) 80% Limb



(c) Edges

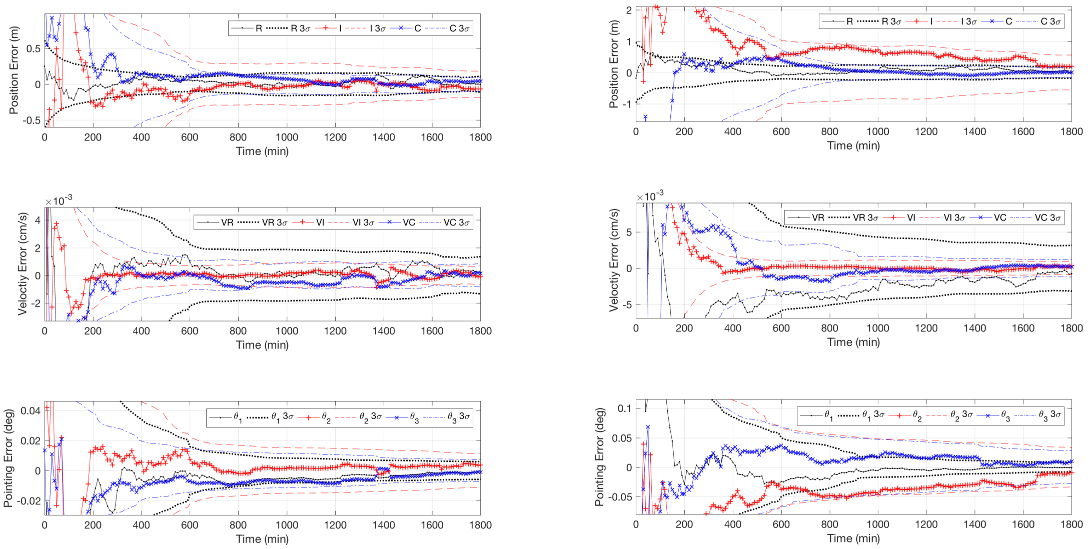
(d) Random

Figure 6.8: State errors in the RIC frame and 3σ covariance bounds in the Itokawa terminator orbit with measurements every 10 minutes for 30 hours, with the UKF.



(a) Full Cross

(b) 80% Limb



(c) Edges

(d) Random

Figure 6.9: State errors in the RIC frame and 3σ covariance bounds in the Itokawa terminator orbit with measurements every 10 minutes for 30 hours, with the LS filter.

The Edges pattern with the LS filter produces the lowest errors of the pixel patterns, yet had a slightly longer run time than the other LS filter simulations. This was mostly likely due to extra iterations performed in the LS filter to align the edge pixels. The 80% Limb and Random patterns with the LS filter also produce small errors and complete in a shorter amount of time. These patterns are more robust, and do not require many iterations within the LS filter. The 80% Limb pattern and Full Cross pattern with the EKF result in comparable errors and run times to their respective LS filters. Implementing an EKF with one of these patterns would be simpler than implementing an iterative LS filter, and could be further optimized for timing.

Table 6.3 displays the magnitude of the ending errors of each of the pixel patterns with each of the estimation filters. When using the same measurements and *a priori* state, every pattern performed as well as using the full image in terms of accuracy by the end of the simulation.

Table 6.3: Final RMS of the magnitude of the position and velocity errors, and the total pointing offset error of the different patterns and filtering methods. This was performed in the Itokawa terminator orbit with measurements every 10 minutes for 30 hours.

Pattern	Filter	$\|\mathbf{e}\|_{\mathbf{r}_f}$ (m)	$\|\mathbf{e}\|_{\mathbf{v}_f}$ (cm/s)	$\alpha_f(deg)$
Full Cross	EKF	0.22	6.99E-04	3.8E-03
	UKF	0.18	5.61E-04	1.6E-03
	LS	0.34	1.62E-03	9.2E-03
80% Limb	EKF	0.45	2.44E-03	0.023
	UKF	0.41	2.25E-03	0.020
	LS	0.53	2.42E-03	0.031
Edges	EKF	0.45	1.53E-03	0.033
	UKF	0.05	2.30E-04	3.4E-03
	LS	0.08	2.60E-04	3.8E-03
Random	EKF	0.33	1.30E-03	0.017
	UKF	0.27	1.23E-03	0.018
	LS	0.20	8.28E-04	9.6E-03
Full Image	EKF	0.14	8.31E-04	0.011
	UKF	0.11	8.84E-04	8.0E-03
	LS	0.13	7.90E-04	7.6E-03

6.4.2 Discussion

Reducing the number of pixels processed in the filter significantly reduces the computational load when compared to using the full flash lidar image. The computational load is reduced in terms of calculating the computed measurement through fewer calls of the ray-tracing algorithm, and in terms of handling smaller matrices within the filter. Selecting a subset of pixels that maximizes the information achieves the same level of accuracy as when using the full flash lidar image.

A significant result from this study is that filter saturation does not occur due to processing fewer measurements at each time step. With the full flash lidar image, filter saturation is seen with the EKF and UKF due to the magnitude of measurements processed at each time step as well as their high precision. With fewer measurements, filter saturation does not occur and it is possible to achieve a comparable level of accuracy with the EKF. Implementing an EKF in an onboard filter would be simpler than implementing an iterative LS filter or UKF.

The 80% Limb and Full Cross patterns consistently run quicker and produce meter-level errors in position across the three estimation filters. Implementing one of these patterns with an EKF would significantly reduce the computational power while providing accurate state estimates. An advantage to using a Full Cross, circle, or random pattern, is that these patterns could be pre-determined without knowledge of the observed or computed images.

The Edges pattern with the LS filter results in the lowest state errors compared to the other patterns, and achieves similar accuracy to the full image. This simulation runs slightly longer than the other LS filter simulations due to additional iterations performed to align the edge pixels. While the edge pixels contain the most information, using an edge pattern might not be as practical with the lidar sensor. As the lidar beam is directed toward the limb of the asteroid, the angle of incidence is increased, and the intensity of the power returned is reduced (Figure 1.2(b)). In addition, the edge pixels may not be present in the observed and computed images, and possibly reduce the number of measurements by half. Patterns such as the 60% Limb or 80% Limb would reduce the angle of incidence to ensure more power is returned while still maximizing the information.

Chapter 7

Conclusion

This dissertation analyzes the use of flash lidar measurements as a relative navigation source in proximity to a small body. It presents a framework that begins with only knowledge of the inertial spacecraft pointing and attitude of the asteroid, sequentially estimates the spacecraft state in a small body orbit, and provides the ability to reduce the number of measurements processed without sacrificing accuracy. The presented work iterates the spacecraft state at each observation time to provide an accurate estimate, and provides bounded state errors when shape modeling errors are present. Reduction of the processing power is achieved by implementing a low fidelity shape model in the filter, as well as reducing the number of altimetry measurements to process. This analysis provides groundwork for implementing a flash lidar instrument as a basis for an autonomous spacecraft navigation system.

Flash lidar and other three-dimensional lidar imaging systems have been studied for spacecraft-to-spacecraft relative navigation, planetary landings, and hazard avoidance. The current basis of knowledge for small body relative navigation has primarily focused on using optical images and single lidar beams for relative navigation. This dissertation addresses this gap in the research by using flash lidar measurements relative to a small body to perform orbit determination.

Flash lidar measurements perform as well or better than using landmark-based navigation from optical images, which is the current state-of-the-art for relative small body navigation. The investigations involve comparing three estimation filters: an extended Kalman filter (EKF), an unscented Kalman filter (UKF), and an iterated least-squares (LS) filter. The estimated state includes estimating the position and velocity, as well as resolving an off-nominal pointing bias. The LS filter results in the lowest state errors of the three filters when processing the full flash lidar image in the nominal orbit determination simulations. The state iterations performed at each time step in this filter combat the filter saturation seen with the EKF and UKF.

Robustness studies analyze the effect of initial state errors, shape modeling errors, and pointing jitter. With initial state errors present, the LS filter is the only filter to produce state errors within the filter covariance bounds. With the UKF, the tuning parameter α effects the potential for an overlap of the observed image and the computed sigma point images, and effects the usability of the observation. If an overlap occurs, the UKF is able to reach sub-meter accuracy in position. When only estimating the position, the LS algorithm is able to converge on a position state to centimeter-level accuracy if an overlap occurs between the observed and computed images. When the position and pointing are estimated, the LS algorithm has potential to terminate early if the position error is too large.

Introducing a low fidelity shape model in the estimation filters does not cause filter divergence, yet the state errors are outside of their respective covariance bounds. The majority of the state errors are captured when a sequential consider covariance analysis is implemented. A random error is applied to the spacecraft pointing at each observation time, emulating pointing jitter, and the LS filter is able to accurately resolve the pointing jitter.

Through knowledge of the spacecraft inertial pointing and asteroid attitude, an *a priori* spacecraft state is determined for filter initialization. A pre-processing algorithm is presented that uses the image properties of the flash lidar image, and performs iterations that reward an overlap in the observed and computed images. Once this overlap occurs, the LS algorithm targets the relative spacecraft position within a few meters and the spacecraft pointing to less than a degree. With two subsequent position estimates, the velocity is linearly calculated, and an *a priori* state is formed to initialize the filter.

Further reduction of the processing power of the navigation filter is achieved by optimally reducing the number of altimetry measurements used at each time step. The pixels in the flash lidar image that provide the maximum information of the spacecraft state occur near the limb of the body. Directing lidar measurements near the limb of the body increases the angle of incidence, which reduces the lidar intensity returned, and was accounted for by an upper range limit. Using this knowledge, pre-determined pixel patterns are developed that reduced the total number of measurements processed per time step by an order of magnitude. With these pre-determined pixel patterns, the same level of accuracy is achieved as with using the full flash lidar image by the end of the observation arc. Additionally, filter saturation does not occur when using these optimal pixel patterns, and a simpler filter such as the EKF achieves similar accuracy to the UKF and LS filter.

Future work includes further testing of the limitations of the flash lidar measurements as a source for relative navigation. This involves increasing the errors between the truth and measurement models with mismodeled dynamics and instrument errors. The number of parameters in the estimation state also may be investigated, such as estimating the spacecraft pointing, the coefficient of reflectivity involved in solar radiation pressure, and gravity coefficients with flash lidar measurements. An autonomous system must be robust to many error sources while maintaining computational efficiency. Tradeoffs may be performed between the number of parameters estimated to account for the error sources, the magnitude of the error sources, and correctly characterizing the state covariance.

Robustness studies may be expanded to the pre-processing algorithm and the optimal pixel selection. Processing fewer measurements per observation and with a lower fideltiy shape model onboard further increases computational efficiency. The ray-tracing algorithm for computing the flash lidar measurements also may be evaluated to reduce the onboard processing power. One technique that is seen in practice is to smartly compute the ray-tracing algorithm for a smaller set of localized facets.

A parallel research topic to state estimation is creating a map that the filter uses for its estimation. In this application, that would be building a shape model of the asteroid through lidar measurements. Future research may be directed at autonomously building this shape model and then using it for orbit determination. Using a model-based approach in the navigation seamlessly connects the autonomous shape building and the navigation.

Other investigations may include studying the differences of other three-dimensional image sensors, such as a scanning lidar or multiple beam lidar, and determining the differences between the algorithms needed. Experiments may also be investigated with these navigation algorithms to test their applicability with hardware.

Bibliography

[1] Matthew Abrahamson and Shyam Bhakskaran. LIDAR and Optical-Based Autonomous Navigation For Small Body Proximity Operations. In AAS/AIAA Space Flight Mechanics Meeting, Santa Fe, NM, January 2014.

[2] Farhad Aghili, Marcin Kuryllo, Galina Okouneva, and Chad English. Fault-tolerant position/attitude estimation of free-floating space objects using a laser range sensor. IEEE Sensors Journal, 11(1):176–185, 2011.

[3] Farzin Amzajerdian, Diego F. Pierrottet, Larry B. Petway, Glenn D. Hines, and Vincent E. Roback. Lidar Systems for Precision Navigation and Safe Landing on Planetary Bodies. Proc. SPIE 8192, International Symposium on Photoelectronic Detection and Imaging 2011: Laser Sensing and Imaging; and Biological and Medical Applications of Photonics Sensing and Imaging, 819202, August 19 2011.

[4] Farzin Amzajerdian, Michael Vanek, Larry Petway, Diego Pierrottet, George Busch, and Alexander Bulyshev. Utilization of 3-D Imaging Flash Lidar Technology for Autonomous Safe Landing on Planetary Bodies. Proc. SPIE 7608, Quantum Sensing and Nanophotonic Devices VII, 760828, January 22 2010.

[5] Franz Andert, Nikolaus Ammann, and Bolko Maass. Lidar-Aided Camera Feature Tracking and Visual SLAM for Spacecraft Low-Orbit Navigation and Planetary Landing. Advances in Aerospace Guidance, Navigation and Control, 2015.

[6] Abraham Bachrach, Samuel Prentice, Ruijie He, and Nicholas Roy. RANGE — Robust Autonomous Navigation in GPS-Denied Environments. Journal of Field Robotics, 28(5):644–666, 2011.

[7] D.S. Bayard, P.B. Brugarolas, and S.B. Broschart. A Faceted Shape Model Approach to Altimetry and Velocimetry for Irregularly Shaped Bodies. In AIAA Guidance, Navigation and Control Conference and Exhibit, number 2008-6833 in AIAA, Honolulu, Hawaii, August 2008.

[8] Bradley M. Bell and Frederick W. Cathey. The Iterated Kalman Filter Update as a Gauss-Newton Method. IEEE TRANSACTIONS ON AUTOMATIC CONTROL, 38(2):294–297, 1993.

[9] Kevin Berry, Brian Sutter, Alex May, and Ken Williams. OSIRIS-REx Touch-And-Go (TAG) Mission Design and Analysis ANALYSIS. In Proceedings of the 36th Annual AAS Guidance and Control Conference, number 13-095 in AAS, 2013.

[10] P J Besl and N D Mckay. A Method for Registration of 3-D Shapes. SPIE, 1611(Sensor Fusion IV):586–606, 1992.

[11] Shyam Bhaskaran, Stephen Broschart, Don Han, Nick Mastrodemos, Ian Roundhill, Brian Rush, Jonathon Smith, and David Surovik. ROSETTA NAVIGATION AT COMET CHURYUMOV-GERASIMENKO. In AAS Guidance and Control Conference, number 15-122 in AAS, Breckenridge, CO, Feburary 2015.

[12] Shyam Bhaskaran and Brian Kennedy. Closed loop terminal guidance navigation for a kinetic impactor spacecraft. Acta Astronautica, 103:322–332, oct 2014.

[13] Shyam Bhaskaran, Shyamskaran, Sumita Nandi, Stephen Broschart, Mark Wallace, L. Alberto Cangahuala, and Corwin Olson. Small Body Landings Using Autonmous Onboard Optical Navigation. The Journal of the Astronautical Sciences, 58(3):409–427, September 2011.

[14] Jens Biele, Stephan Ulamec, Michael Maibaum, Reinhard Roll, Lars Witte, Eric Jurado, Pablo Muñoz, Walter Arnold, Hans-ulrich Auster, Carlos Casas, Claudia Faber, Cinzia Fantinati, Felix Finke, Hans-herbert Fischer, Koen Geurts, Carsten Güttler, Philip Heinisch, Alain Herique, Stubbe Hviid, Günter Kargl, Martin Knapmeyer, Jörg Knollenberg, Wlodek Kofman, Norbert Kömle, Ekkehard Kührt, Valentina Lommatsch, Stefano Mottola, Ramon Pardo De Santayana, Emile Remetean, Frank Scholten, Klaus J. Seidensticker, Holger Sierks, and Tilman Spohn. The landing(s) of Philae and inferences about comet surface mechanical properties. Science, 349(6247), 2015.

[15] B G Boone, K Strohbehn, B E Kluga, K C Baldwin, J R Bruzzi, N P Paschilidis, C W Drabenstadt, and P Wilson. Dual-band spacecraft sensor suite for lunar and small-body landing. Laser Radar Technology and Applications XII, Proc. of SPIE, 6550:1–14, 2007.

[16] J J Bordi, J K Miller, B G Williams, R S Nerem, and F J Pelletier. The Impact of Altimeter Range Observations on Near Navigation. In AAS Astrodynamics Specialist Conference, number 2000-4423, pages 486–496. AIAA, 2000.

[17] Jack P Brazzel, Fred D. Clark, and Peter T Spehar. RPOP Enhancements to Support the Space Shuttle R-Bar Pitch Maneuver for Tile Inspection. In AIAA Guidance, Navigation, and Control Conference and Exhibit, pages 15–18, August 2005.

[18] Stephen Broschart, Shyam Bhaskaran, Julie Bellerose, Ann Dietrich, Dongsuk Han, Robert Haw, Nickolaos Mastrodemos, William M. Owen, Brian Rush, and David Surovik. Shadow navigation support at jpl for the rosetta landing on comet 67p/churyumov-gerasimenko. In 26th International Symposium on Space Flight Dynamics ISSFD, number ISSFD-2017-096, June 2017.

[19] Stephen B Broschart. Close Proximity Spacecraft Maneuvers Near Irregularly Shaped Small-bodies : Hoveri
PhD thesis, The University of Michigan, October 2006.

[20] Stephen B Broschart, Daniel J Scheeres, and Ann Arbor. Spacecraft Descent and Translation in the Small-Body Fixed Frame. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, number August, pages 1–14, 2004.

[21] Alexander Bulyshev, Diego Pierrottet, Farzin Amzajerdian, George Busch, Michael Vanek, and Robert Reisse. Processing of 3-dimensional flash lidar terrain images generated from an airborne platform. SPIE 7329: Three-Dimensional Imaging, Visualization, and Display, 7329:73290I–73290I–9, 2009.

[22] Yang Chen and Gérard Medioni. Object Modeling by Reg strat ion of Multiple Range Images. In IEEE International Conference on Robotics and Automation, pages 2724–2729. IEEE, April 1991.

[23] Yang Cheng, Andrew E. Johnson, Larry H. Matthies, and Clark F. Olson. Optical landmark detection for spacecraft navigation. Advances in the Astronautical Sciences, 114 III:1785–1803, 2003.

[24] Steven R. Chesley, Davide Farnocchia, Michael C. Nolan, David Vokrouhlický, Paul W. Chodas, Andrea Milani, Federica Spoto, Benjamin Rozitis, Lance a M Benner, William F. Bottke, Michael W. Busch, Joshua P. Emery, Ellen S. Howell, Dante S. Lauretta, Jean Luc Margot, and Patrick a. Taylor. Orbit and bulk density of the OSIRIS-REx target Asteroid (101955) Bennu. Icarus, 235:5–22, June 2014.

[25] Steve Chien, Rob Sherwood, Daniel Tran, Rebecca Castano, Benjamin Cichy, Ashley Davies, Gregg Rabideau, Nghia Tang, Michael Burl, Dan Mandl, Stuart Frye, Jerry Hengemihle, Jeff D Agostino, Robert Bote, Bruce Trout, Seth Shulman, Stephen Ungar, Jim Van Gaasbeck, Darrell Boyer, Control Systems, Michael Griffin, and Hsiao-hua Burke Mit. Autonomous Science on the EO-1 Mission. In International Symposium on Artificial Intelligence Robotics and Automation in Space (i-SAIRAS), Nara, Japan, May 2003.

[26] John Christian, Shane B. Robinson, Christopher N. D'Souza, and Jose P. Ruiz. Cooperative Relative Navigation of Spacecraft Using Flash Light Detection and Ranging Sensors. Journal of Guidance, Control, and Dynamics, 37(2):452–465, 2014.

[27] John A. Christian. Optical Navigation Using Planet's Centroid and Apparent Diameter in Image. Journal of Guidance, Control, and Dynamics, 38(2), 2015.

[28] John A. Christian. Accurate Planetary Limb Localization for Image-Based Spacecraft Navigation. Journal of Spacecraft and Rockets, 54(3), 2017.

[29] John A. Christian and Scott Cryan. A Survey of LIDAR Technology and its Use in Spacecraft Relative Navigation. In Guidance, Navigation, and Control and Co-located Conferences, pages 1–7, 2013.

[30] Rex Craig and Patrick Earhart. Optical sensors for space proximity operations. In International Quantum Electronics Conference, page PThB2, Boulder, CO, May 2009. Optical Society of America.

[31] M. G. Daly, O. S. Barnouin, C. Dickinson, J. Seabrook, C.L. Johnson, G. Cunningham, T. Haltigin, D. Gaudreau, C. Brunet, I. Aslam, A. Taylor, E. B. Bierhaus, W. Boynton, M. Nolan, and D. S. Lauretta. The OSIRIS-REx Laser Altimeter. In Lunar and Planetary Science XLVIII (2017), pages 11–12, 2017.

[32] J. de Lafontaine, A Ulitsky, J W Tripp, R Richards, M Daly, and C Sallaberger. LAPS : THE DEVELOPMENT OF A SCANNING LIDAR SYSTEM WITH GNC FOR AUTONOMOUS HAZARD AVOIDANCE AND. Proceedings of SPIE, 5418:81–93, 2004.

[33] Walter F. Denham and Samuel Pines. Sequential Estimation When Measurement Function Nonlinearity Is Comparable to Measurement Error. AIAA Journal, 4(6):1071–1076, 1966.

[34] C. S. Dickinson, M. Daly, O. Barnouin, B. Bierhaus, D. Gaudreau, J. Tripp, M. Ilnicki, and A. Hildebrand. AN OVERVIEW OF THE OSIRIS REX LASER ALTIMETER - OLA. In 43rd Lunar and Planetary Science Conference, pages 1–2, 2012.

[35] Ann Dietrich and Jay W McMahon. Asteroid proximity navigation using flash lidar. In AIAA/AAS Astrodynamics Specialist Conference, San Diego, CA, August 2014.

[36] Ann Dietrich and Jay W McMahon. Autonomous observation planning with flash lidar around small bodies. In AIAA/AAS Astrodynamics Specialist Conference, number 15-736 in AAS, Vail, CO, August 2015.

[37] Ann Dietrich and Jay W McMahon. Error sensitivies for flash lidar based relative navigation around small bodes. In AAS Guidance and Control Conference, number AAS 16-017, Breckenridge, CO, February 2016.

[38] Ann Dietrich and Jay W McMahon. Orbit determination using flash light detection and ranging (lidar) around small bodies. Journal of Guidance, Control, and Dynamics, Accessed Dec 30, 2016 2016.

[39] D Dunham, RobertW. Farquhar, James V. McAdams, Mark Holdridge, Robert Nelson, Karl Whittenburg, Peter Antreasian, Steven Chesley, Clifford Helfrich, William M. Owen, Bobby Williams, Joseph Veverka, and Ann Harch. Implementation of the First Asteroid Landing. Icarus, 159(2):433–438, 2002.

[40] Tara a. Estlin, Benjamin J. Bornstein, Daniel M. Gaines, Robert C. Anderson, David R. Thompson, Michael Burl, Rebecca Castaño, and Michele Judd. AEGIS Automated Science Targeting for the MER Opportunity Rover. ACM Transactions on Intelligent Systems and Technology, 3(3):1–19, May 2012.

[41] Angel Flores-abad, Ou Ma, Khanh Pham, and Steve Ulrich. Progress in Aerospace Sciences A review of space robotics technologies for on-orbit servicing. Progress in Aerospace Sciences, 68:1–26, 2014.

[42] Eric W. Frew. Sensitivity of Cooperative Target Geolocalization to Orbit Coordination. Journal of Guidance, Control, and Dynamics, 31(4):1028–1040, 2008.

[43] A Fujiwara, J Kawaguchi, D K Yeomans, M Abe, T Mukai, T Okada, J Saito, H Yano, M Yoshikawa, D J Scheeres, O Barnouin-Jha, a F Cheng, H Demura, R W Gaskell, N Hirata, H Ikeda, T Kominato, H Miyamoto, a M Nakamura, R Nakamura, S Sasaki, and K Uesugi. The rubble-pile asteroid Itokawa as observed by Hayabusa. Science, 312(5778):1330–4, 2006.

[44] R. Gaskell, J. Saito, M. Ishiguro, T. Kubota, T. Hashimoto, N. Hirata, S. Abe, O. Barnouin-Jha, and D. Scheeres. Gaskell itokawa shape model v1.0. hay-a-amica-5-itokawashape-v1.0. NASA Planetary Data System, 2008.

[45] R W Gaskell, O S Barnouin-Jha, Daniel J Scheeres, A S Konopliv, T Mukai, S Abe, J Saito, M Ishiguro, T Kubota, T Hashimoto, J Kawaguchi, M Yoshikawa, Ken'ichi Shirakawa, T Kominato, N Hirata, and H Demura. Characterizing and navigating small bodies with imaging data. Meteoritics & Planetary Science, 43(6):1049–1061, September 2008.

[46] Erann Gat. On three-layer architectures. Artificial Intelligence and Mobile Robots, David Kortenkamp, R. Peter Bonnasso, and Robin Murphy, eds., AAAI Press, 1998.

[47] A Gelb. Applied Optimal Estimation. MIT Press, Cambridge, Mass., 1974.

[48] G. Gonzalez Peytavi, T.P. Andert, A. Probst, D. Sanchez Morales, and B. Eissfeller. 3D Imaging for Autonomous Navigation About Small Bodies. In AIAA/AAS Astrodynamics Specialist Conference AIAA, number September, pages 1–12, 2016.

[49] John L Goodman. History of Space Shuttle Rendezvous and Proximity Operations. Journal of Spacecraft and Rockets, 43(5):944–959, 2006.

[50] Robert G Gottlieb. Fast Gravity, Gravity Partials, Normalized Gravity, Gravity Gradient Torque and Magnetic Field : Derivation, Code and Data. NASA Contractor Report 188243, McDonnell Douglas Space Systems - Houston Division, 1993.

[51] Slawomir Grzonka, Giorgio Grisetti, and Wolfram Burgard. A Fully Autonomous Indoor Quadrotor. IEEE TRANSACTIONS ON ROBOTICS, 28(1):90–100, 2012.

[52] Alan W. Harris and Germano D'Abramo. The population of near-Earth asteroids. Icarus, 257:302–312, 2015.

[53] Ulrich Hillenbrand, Roberto Lampariello, Ulrich Hillenbrand, and Roberto Lampariello De. Motion and Parameter Estimation of a Free-Floating Space Object from Range Data for Motion Prediction. In Proceedings of i-SAIRAS, 2005.

[54] Michael E. Hough. Orbit Determination with Improved Covariance Fidelity, Including Sensor Measurement Biases. JOURNAL OF GUIDANCE,CONTROL, AND DYNAMICS, 34(3), 2011.

[55] Coralie D Jackman, Derek S Nelson, Leilah K Mccarthy, Tiffany J Finley, Andrew J Liounis, Kenneth M Getzandanner, Peter G Antreasian, and Michael C Moreau. OPTICAL NAVIGATION CONCEPT OF OPERATIONS FOR THE OSIRIS-REX MISSION. In AAS/AIAA Spaceflight Mechanics Meeting, pages 1–18, San Antonio, TX, 2017.

[56] R. Jaumann, D. a. Williams, D. L. Buczkowski, R. a. Yingst, F. Preusker, H. Hiesinger, N. Schmedemann, T. Kneissl, J. B. Vincent, D. T. Blewett, B. J. Buratti, U. Carsenty, B. W. Denevi, M. C. De Sanctis, W. B. Garry, H. U. Keller, E. Kersten, K. Krohn, J.-Y. Li, S. Marchi, K. D. Matz, T. B. McCord, H. Y. McSween, S. C. Mest, D. W. Mittlefehldt, S. Mottola, a. Nathues, G. Neukum, D. P. O'Brien, C. M. Pieters, T. H. Prettyman, C. a. Raymond, T. Roatsch, C. T. Russell, P. Schenk, B. E. Schmidt, F. Scholten, K. Stephan, M. V. Sykes, P. Tricarico, R. Wagner, M. T. Zuber, and H. Sierks. Vesta's Shape and Morphology. Science, 336(6082):687–690, 2012.

[57] Andrew Jazwinski. Stocastic Processes and Filtering Theory. Academic Press, New York, 1970.

[58] a E Johnson, Adnan Ansar, and L H Matthies. A general approach to terrain relative navigation for planetary landing. In Proc. 2007 AIAA Infotech at Aerospace Conference, pages 7–10, May 2007.

[59] Simon Julier and Jeffrey Uhlmann. Unscented Filtering and Non Linear Estimation. Proceedings of the IEEE, 92(3):401–422, 2004.

[60] Simon J. Julier. The scaled unscented transformation. In American Control Conference, 2002. Proceedings of the 2002, pages 4555–4559. IEEE, 2002.

[61] Alireza G Kashani, Michael J Olsen, Christopher E Parrish, and Nicholas Wilson. A Review of LIDAR Radiometric Processing: From Ad Hoc Intensity Correction to Rigorous Radiometric Calibration. Sensors, 15(11):28099–28128, 2015.

[62] William M. Kaula. Theory of Satellite Geodesy: Applications of Satellites to Geodesy, pages 1–7. Dover Publications, Inc, 1966.

[63] Jed M Kelsey, Jeffrey Byrne, Martin Cosgrove, Sanjeev Seereeram, and Raman K Mehra. Vision-Based Relative Pose Estimation for Autonomous Rendezvous And Docking. In Aerospace Conference, 2006 IEEE. IEEE, 2006.

[64] a. S. Konopliv, S. W. Asmar, R. S. Park, B. G. Bills, F. Centinello, a. B. Chamberlin, a. Ermakov, R. W. Gaskell, N. Rambaux, C. a. Raymond, C. T. Russell, D. E. Smith, P. Tricarico, and M. T. Zuber. The Vesta gravity field, spin pole and rotation period, landmark positions, and ephemeris from the Dawn tracking and optical data. Icarus, 240:103–117, 2013.

[65] DG Kubitschek, Nickolaos Mastrodemos, Robert Werner, Brian Kennedy, Stephen P. Synnott, George Null, Shyam Bhaskaran, Joseph Riedel, and Andrew T. Vaughan. Deep Impact Autonomous Navigation: the trials of targeting the unknown. In AAS Guidance and Control Conference, number AAS 06-081, Breckenridge, CO, February 2006.

[66] T. Kubota, T. Hashimoto, M. Uo, M. Maruya, and K. Baba. Maneuver strategy for station keeping and global mapping around an asteroid. In Space- flight Mechanics 2001: Advances in the Astronautical Sciences, number AAS Paper 01-156, pages 769–779, 2001.

[67] Takashi Kubota, Tatsuaki Hashimoto, Jun'ichiro Kawaguchi, Masashi Uo, and Ken'ichi Shirakawa. Guidance and navigation of hayabusa spacecraft for asteroid exploration and sample return mission. 2006 SICE-ICASE International Joint Conference, pages 2793–2796, 2006.

[68] Yangming Li and Edwin B. Olson. Extracting general-purpose features from LIDAR data. Proceedings - IEEE International Conference on Robotics and Automation, pages 1388–1393, 2010.

[69] David A Lorenz, Courtney Mario, Michael Daly, Ryan Olds, Mark E Perry, Alexander May, and Eric E Palmer. Lessons Learned from OSIRIS-REx Autonomous Navigation Using Natural Feature Tracking. In Aerospace Conference, 2017 IEEE, 2017.

[70] Kok-Lim Low. Linear Least-Squares Optimization for Point-to-plane ICP Surface Registration. Technical Report TR04-004, Department of Computer Science, University of North Carolina at Chapel Hill, February 2004.

[71] Courtney Mario and Chris Debrunner. ROBUSTNESS AND PERFORMANCE IMPACTS OF OPTICAL-BASED FEATURE TRACKING TO OSIRIS-REX ASTEROID SAMPLE COLLECTION MISSION. In AAS Guidance and Control Conference, number AAS 16-087, Breckenridge, CO, February 2016.

[72] Nickolaos Mastrodemos, Brian Rush, Drew Vaughan, and Bill Owen. Optical navigation for dawn at vesta. In 21st AAS/AIAA Space Flight Mechanics Meeting, number AAS 11-222, New Orleans, Louisiana, Feburary 2011.

[73] Nickolaos Mastrodemos, Brian P Rush, and William M Owen. OPTICAL NAVIGATION FOR THE ROSETTA MISSION. In AAS Guidance and Control Conference, number AAS 15-123, Breckenridge, CO, Feburary 2015.

[74] Nikos Mastrodemos, Daniel G. Kubitschek, and Stephen P. Synnott. Autonomous Navigation for the Deep Impact Mission Encounter with Comet Tempel 1. Space Science Reviews, 117(1-2):95–121, March 2005.

[75] Jay W McMahon, Daniel J Scheeres, and Kevin Berry. Asteroid Proximity Navigation using Direct Altimetry Measurements. In AAS/AIAA Space Flight Mechanics Meeting, number AAS 14-354, Santa Fe, NM, January 2014.

[76] James K Miller, Connie J Weekst, and Lincoln J Woodj. Orbit Determination Strategy and Accuracy for a Comet Rendezvous Mission. Journal of Guidance, Control, and Dynamics, 13(5), 1990.

[77] J.K. Miller, a.S. Konopliv, P.G. Antreasian, J.J. Bordi, S. Chesley, C.E. Helfrich, W.M. Owen, T.C. Wang, B.G. Williams, D.K. Yeomans, and D.J. Scheeres. Determination of Shape, Gravity, and Rotational State of Asteroid 433 Eros. Icarus, 155(1):3–17, 2002.

[78] Aanastasios I. Mourikis, Nikolas Trawny, S.I. Roumeliotis, a.E. Johnson, and Larry H. Matthies. Vision-Aided Inertial Navigation for Precise Planetary Landing: Analysis and Experiments. Robotics: Science and Systems, 2007.

[79] N Muscettola, GA Dorais, C Fry, R Levinson, and C Plaunt. IDEA : Planning at the Core of Autonomous Agents. American Association for Artificial Intelligence, 2002.

[80] NASA. "sts-127, a porch in space", sts-127 press kit, June 2009.

[81] M.C. Nolan, C. Magri, E.S. Howell, L.A.M. Benner, J.D. Giorgini, C.W. Hergenrother, R.S. Hudson, D.S. Lauretta, J.L. Margot, S.J. Ostro, and D.J. Scheeres. Asteroid (101955) bennu shape model v1.0. ear-a-i0037-5-bennushape-v1.0. NASA Planetary Data System, 2013.

[82] Michael C Nolan, Christopher Magri, Ellen S Howell, Lance A M Benner, Jon D Giorgini, Carl W Hergenrother, R Scott Hudson, Dante S Lauretta, Jean-luc Margot, Steven J Ostro, and Daniel J Scheeres. Shape model and surface properties of the OSIRIS-REx target Asteroid (101955) Bennu from radar and lightcurve observations. ICARUS, 226(1):629–640, 2013.

[83] Edwin B Olson. Real-Time Correlative Scan Matching. In IEEE International Conference on Robotics and Automation, pages 4387–4393, Kobe, Japan, 2009.

[84] William M Owen. METHODS OF OPTICAL NAVIGATION . In AAS Spaceflight Mechanics Conference, number AAS 11-215, New Orleans, Louisiana, February 2011.

[85] Antoine Petit, Eric Marchand, and Keyvan Kanani. Tracking complex targets for space rendezvous and debris removal applications. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 4483–4488. IEEE, 2012.

[86] Diego F Pierrottet, Farzin Amzajerdian, Byron L Meadows, Robert Estes, and Anna M Noe. Characterization of 3-D imaging lidar for hazard avoidance and autonomous landing on the Moon. SPIE, 6550:655008–655009, 2007.

[87] Didier Pinard, Stéphane Reynaud, Patrick Delpy, and Stein E Strandmoe. Accurate and autonomous navigation for the ATV Navigation précise et autonome pour l ' ATV . Aerospace Science and Technology, 11:490–498, 2007.

[88] Vincent E Roback, Farzin Amzajerdian, Alexander E Bulyshev, Paul F Brewster, W Bruce, and Nasa Langley. 3D Flash Lidar Performance in Flight Testing on the Morpheus Autonomous , Rocket-Propelled Lander to a Lunar-Like Hazard Field. Proc. of SPIE, 9832(983209):1–20, 2016.

[89] Reuben Rohrschneider, Carl Weimer, James Masciarelli, Michael Lieber, Carl Adkins, and Jeanette L. Domber. Vision Navigation Sensor (VNS) with Adaptive Electronically Steerable Flash LIDAR (ESFL). In AIAA Guidance, Navigation, and Control Conference, number AIAA 2016-2096, San Diego, CA, January 2016.

[90] Reuben R. Rohrschneider. Terrain Relative Navigation Using Crater Identification in Surface Topography Data. In AIAA Guidance, Navigation, and Control Conference, number AIAA 2011-6490, pages 1–13, Reston, VA, August 2011.

[91] Reuben R Rohrschneider and William Tandy. Pose Determination Using Only 3D Range Images from the STORRM Mission. In 36th ANNUAL AAS GUIDANCE AND CONTROL CONFERENCE, number AAS 13-104, Breckenridge, CO, February 2013.

[92] Stéphane Ruel, Tim Luu, and Andrew Berube. Space Shuttle Testing of the TriDAR 3D Rendezvous and Docking Sensor. Journal of Field Robotics, 29(4):535–553, 2012.

[93] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In International Conference on 3-D Digital Imaging and Modeling, 3DIM, pages 145–152, 2001.

[94] D J Scheeres, S Broschart, S J Ostro, and Et Al. The Dynamical Environment About Asteroid 25143 Itokawa : Target of the Hayabusa Mission. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, (August 2004):1–12, 2004.

[95] D J Scheeres, R Gaskell, S Abe, O Barnouin-Jha, T Hashimoto, J Kawaguchi, T Kubota, J Saito, M Yoshikawa, N Hirata, T Mukai, M Ishiguro, T Kominato, K Shirakawa, and M Uo. The actual dynamical environment about Itokawa. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, number AIAA 2006-6661, Keystone, CO, August 2006.

[96] Daniel J. Scheeres. Interactions Between Ground-Based and Autonomous Navigation for Precision Landing at Small Solar-System Bodies. Telecommunications and Data Acquisition Progress Reports, 42, 1998.

[97] Daniel J Scheeres. Orbit Mechanics About Asteroids and Comets. Journal of Guidance, Control, and Dynamics, 35(3):987–997, 2012.

[98] Stanley F Schimdt. Applications of state space methods to navigation problems. Advances in Control Systems, 3:293–340, 1966.

[99] D E Smith, M T Zuber, S C Solomon, R J Phillips, J W Head, J B Garvin, W B Banerdt, D O Muhleman, G H Pettengill, G a Neumann, F G Lemoine, J B Abshire, O Aharonson, C D Brown, S a Hauck, a B Ivanov, P J McGovern, H J Zwally, and T C Duxbury. The global topography of Mars and implications for surface evolution. Science (New York, N.Y.), 284(5419):1495–1503, 1999.

[100] David E. Smith, Maria T. Zuber, Gregory a. Neumann, Frank G. Lemoine, Erwan Mazarico, Mark H. Torrence, Jan F. McGarry, David D. Rowlands, James W. Head, Thomas H. Duxbury, Oded Aharonson, Paul G. Lucey, Mark S. Robinson, Olivier S. Barnouin, John F. Cavanaugh, Xiaoli Sun, Peter Liiva, Dan Dan Mao, James C. Smith, and Arlin E. Bartels. Initial observations from the Lunar Orbiter Laser Altimeter (LOLA). Geophysical Research Letters, 37(18):1–6, 2010.

[101] Yu Takahashi and D. J. Scheeres. Small body surface gravity fields via spherical harmonic expansions. Celestial Mechanics and Dynamical Astronomy, 119(2):169–206, 2014.

[102] Yu Takahashi, D. J. Scheeres, and Robert a. Werner. Surface Gravity Fields for Asteroids and Comets. Journal of Guidance, Control, and Dynamics, 36(2):362–374, 2013.

[103] Byron D. Tapley, Bob E. Schutz, and George H. Born. Statistical Orbit Determination, pages 11,13, 160–165, 183–190, 199–205, 220–230. Elsevier Inc., Burlington, MA, 2004.

[104] James H Taylor. The Cramer-Rao Estimation Error Lower Bound Computation for Deterministic Nonlinear Systems. IEEE TRANSACTIONS ON AUTOMATIC CONTROL, AC-24(2):343–344, 1979.

[105] Julie K Thienel, Nasa Goddard, and Space Flight. Hubble Space Telescope Angular Velocity Estimation During the Robotic Servicing Mission. Journal of Guidance Control and Dynamics, 30(1):29–34, 2007.

[106] Charles Toth, Dorota a. Grejner-Brzezinska, and Young-Jin Lee. Terrain-based navigation: Trajectory recovery from LiDAR data. In 2008 IEEE/ION Position, Location and Navigation Symposium, pages 760–765, 2008.

[107] Tristan Tzschichholz, Lei Ma, and Klaus Schilling. Model-based spacecraft pose estimation and motion prediction using a photonic mixer device camera. Acta Astronautica, 68(7-8):1156–1167, 2011.

[108] D.A. Vallado. Fundamentals of Astrodynamics and Applications, page 543. Space Technology Library. Springer Netherlands, 2001.

[109] Kiri L. Wagstaff, David R. Thompson, Brian D. Bue, and Thomas J. Fuchs. Autonomous Real-Time Detection of Plumes and Jets From Moons and Comets. The Astrophysical Journal, 794(1):43, September 2014.

[110] E.A. Wan and R Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), pages 153–158. Ieee, 2000.

[111] Robert A. Werner and Daniel J. Scheeres. Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 castalia. Celestial Mechanics and Dynamical Astronomy, 65(3):313–344, 1996.

[112] Tim Winkler, Brian Kaplinger, and Bong Wie. Optical navigation and fuel-efficient orbit control around an irregular-shaped asteroid. In AIAA Guidance, Navigation, and Control (GNC) Conference, Boston, MA, August 2013.

[113] Drew P Woodbury and John L Junkins. On the Consider Kalman Filter. In AIAA Guidance, Navigation, and Control Conference, August 2010.

[114] Hajime Yano, T Kubota, H Miyamoto, T Okada, D Scheeres, Y Takagi, K Yoshida, M Abe, S Abe, O. Barnouin-Jha, A Fujiwara, S Hasegawa, T Hashimoto, M Ishiguro, M Kato, J Kawaguchi, T Mukai, J Saito, S Sasaki, and M Yoshikawa. Touchdown of the Hayabusa Spacecraft at the Muses Sea on Itokawa. Science, 312(5778):1350–1353, 2005.

[115] Renato Zanetti and Robert H. Bishop. Kalman Filters with Uncompensated Biases. JOURNAL OF GUIDANCE,CONTROL, AND DYNAMICS, 35(1):327–330, 2012.

[116] Renato Zanetti, Kyle J. DeMars, and Robert H. Bishop. Underweighting Nonlinear Measurements. Journal of Guidance, Control, and Dynamics, 33(5):1670–1675, 2010.

[117] Renato Zanetti and Chirs D'Souza. RECURSIVE IMPLEMENTATIONS OF THE CONSIDER FILTER. In AAS Jer-Nan Juang Astrodynamics, pages 1–17, June 2012.

[118] M T Zuber, D E Smith, S C Solomon, D O Muhleman, J W Head, J B Garvin, J B Abshire, and J L Bufton. The Mars-Observer Laser Altimeter Investigation. Journal of Geophysical Research-Planets, 97(E5):7781–7797, 1992.