Computationally-Efficient Visual-Inertial Simultaneous Localization and Mapping for Spaceflight Navigation

by

Matthew W. Givens

B.S., University of Utah, 2017

M.S., Utah State University, 2019

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

2023

Givens, Matthew W. (Ph.D., Aerospace Engineering Sciences)

Computationally-Efficient Visual-Inertial Simultaneous Localization and Mapping for Spaceflight Navigation

Thesis directed by Prof. Dr. Jay McMahon

This thesis represents an investigation into the application to spaceflight of the estimation techniques developed to solve the well-known robotics problem of Simultaneous Localization and Mapping (SLAM). This subject has been thoroughly studied in the context of ground and aerial robotics but its study for use in the spaceflight domain, where the dynamical, measurement, and computational challenges are often very different than in terrestrial applications, is less common. Further, the wide body of extant robotics research into SLAM can be difficult to approach and understand for space navigators with a more classical estimation background because of the differences in terminology and assumptions that roboticists have utilized over time. This work offers an overview of the development of SLAM in robotics and how it has been applied in that field, as well as an accessible approach to the problem for researchers with an aerospace background. This also leads into the development of a novel visual-inertial (VI) SLAM algorithm designed to achieve constant-time exploration and mapping while still integrating the full nonlinear dynamics of the space environment, handling the high update rates of inertial measurement units (IMUs), and incorporating the measurement information produced by a camera sensor. This algorithm is applied to 2D and 3D simulated and real datasets to demonstrate its capability to quickly generate accurate state estimates of both spacecraft and environmental variables.

Dedication

To all those who sacrificed so that I could succeed, both before and during my brief stay on this planet

Acknowledgements

After an extended educational career that anyone in my family will tell you is defined by wandering, and admittedly sometimes wanderlust, an immediate irony is present in the fact that I found my academic and career interests in engineering and navigation. Coming to college, I was unsure of everything except that I never wanted to do math again. Architecture, music, history, and videography were my interests and if you told my younger self that he would be writing a gigantic, math-laden dissertation to obtain a PhD in engineering, he would not have believed you. While I maintain all of these interests to the present, sometimes to a fault, my calling was eventually found in dreams of space exploration and that path eventually led to the present monograph.

The patience and grace of my family cannot be emphasized or overstated enough. I would be nothing without the guidance and support of my parents, Al and Flossie, and the undying love and dedication of my wife, Christine. Even through the storms of the COVID-19 pandemic, working from home, and the slow and tragic death of her father, Mike DeDen, she has stayed by my side and continued to support my dreams.

I also could not have asked for a better advisor and friend in Dr. Jay McMahon. Even when I decided to careen off into the unfamiliar oceans of computer vision and robotics, he knew which questions to ask and how to keep me on course. The friends and labmates I have been fortunate to interact with over my college career, while in ORCCA and otherwise, have also been outstanding and I want to especially thank my optical navigation buddies Jacopo Villa and Koundinya Kuppa, as well as my Canadian compadre Spencer Boone, for all of the good times and good conversations in Boulder and beyond.

Contents

Tables

Figures

Chapter 1

Introduction

Philosophers and psychologists have spilled much ink attempting to answer the fundamental question of "who am I?"[1] while chemists and biologists have devised convincing answers to the question of "what am I?" and psychiatrists probe the sometimes-banal question of "how am I?" By contrast, relatively few words have been written about a perhaps equally fundamental human question: "*where* am I?" This is the intellectual realm of the navigator and his craft. The related questions of "where did I come from?" and "where will I end up?", in the literal sense, are fundamentally the same question as before, separated only by the uncompromising arrow of time.

Navigation, like Reason, can be divided into two fundamentally different regimes: inductive and deductive. The navigator's parlance for these two approaches is *dead reckoning* for the former and *postion fixing* for the latter. Both are typically essential to come to a satisfactory navigational solution during a sea voyage or, as is the more pertinent case in the present work, for a spacecraft traversing the void. Dead reckoning is the process of propagating prior positional knowledge forward in time using some model of the intervening motion to make an informed guess of the current or future position. Without perfect knowledge of the motion of the craft, dead reckoning inevitably results in errors accumulating and drift in the solution from the truth over time. Position fixing is done through measurements which, when thoughtfully taken with respect to fixed references with known attributes, provide a means of correction and are thus essential to stay the course.

An important tool for the navigator is the *map*, a document or database which encompasses

[1] Theologians also address this question, although many in the Augustinian tradition prefer the frame of "whose am I?"

knowledge of the environment that exists prior to the navigational act. Given some prior knowledge of a craft's position, the map can provide additional knowledge by comparison with measurements. In modern robotical terms this is more widely labeled *localization*. In the event that an *a priori* map is unavailable, and that we wish to generate a map of the environment while contemporaneously navigating through it, the processes of localization and mapping become a joint problem: the most likely solution to one corresponds to the most likely solution of the other. This observation was the genesis of *Simultaneous Localization and Mapping* (SLAM), a field of study that has become a mainstay of robotics research in the last 40 years.[38] If the measurements used in the SLAM process are only taken relative to the robotic agent and not with respect to some more general, fixed frame of reference, the map generated is also only relative and subject to drift. The internal consistency of the map can be corrected through *loop closure*, when the agent returns to previously-mapped location and uses the re-encounter to correct the intervening accumulated drift.

The use of *visual* measurements to environmental landmarks in SLAM, such as to image features provided by a digital camera sensor, echoes other fields of research such as Structure from Motion (SfM) and the more traditional Bundle Adjustment (BA) problem of photogrammetry. These three fields, which have been united under the banner of Bayesian estimation,[105, 127, 116] are now notionally separated by their respective application domains: SLAM in mobile robotics, SfM in object reconstruction, and photogrammetry in space and aerial photography. Visual SLAM (V-SLAM) can also be differentiated from visual odometry (VO) by its retention of a map in memory for use in *loop closure*. Where VO "forgets" the map, or does not generate a map in the first place, V-SLAM builds a map and retains it for future use. In this way, a VO solution with only relative measurements could be seen as a simplification of a comparable V-SLAM solution and, as the name implies, another form of dead reckoning.

The motivation for this simplification is often computational: retaining an arbitrarily large map in computer memory comes with a cost. Beyond simply requiring the ever-growing map information to be retained in storage, the map also induces correlations between estimated quantities that must be accounted for in the navigation process in order to provide a consistent solution. With often-limited computational resources available onboard a robot or vehicle, the algorithmic *time complexity* of SLAM has been one of the major topics of research in the field up to the present day. For space systems, the limitations of radiation-hardened computer hardware present an even more acute challenge.

1.1 Brief Historical View of Robotic SLAM

Any complete SLAM algorithm can be divided into two general phases, generally referred to as the "frontend" and "backend", that, while typically coupled, can largely be treated separately. The "frontend" refers to the process of taking sensor data and distilling it into useful measurements of the environment that the "backend" can use to solve the estimation problem. These must operate in concert, with information flowing in both directions, to enable the efficient computation of a reliable solution. With the exception of the discussion in Section 1.2, the present work is largely concerned with investigating the backend portion of the SLAM problem, specifically in a formulation that assumes that visual measurements from a camera sensor are being processed to unmoving, world-fixed landmarks. The following section has thus been written from that perspective. However, it must be stated that much of the innovation in SLAM in the last decade has come on the frontend side of the problem, with point clouds, direct methods, dense methods, machine learning, and other innovations coming onto the scene.[98]

SLAM in dynamic environments, where the assumption that landmarks are static in the "world-fixed" frame is relaxed,[110] is also another, very active field of research that is not broached in this thesis. For most space applications, the relative geometry of the object being mapped is typically static but must be coordinatized with respect to a rotating frame. SLAM methodology could also be used for localization and tracking with respect to discrete, moving targets but this is also beyond the scope of this work.

1.1.1 Extended Kalman Filter

As recounted by Reference [38], the SLAM problem in its current form has its origins in the 1980s when probabilistic methods were first being introduced to robotics. In the subsequent years, as computer processing power increased, the first wave of SLAM research was typically characterized by approaches that assumed applications in ground robotics as well as the availability of both range and bearing measurements. As the community began to realize the importance of *jointly* estimating the robot's states, the landmarks, and the statistical correlations between all quantities of interest, the Extended Kalman Filter (EKF), which is the extension of the classical Kalman Filter to nonlinear systems via linearization of the dynamics and measurement models,[16] began to find wide application in SLAM research. The EKF and its variants "roll-up" all prior motion and measurement information into a single *state vector* that represents the current best estimate of the states with their assumed-Gaussian uncertainties. This *sequential* algorithm has previously been used, and is still used, for all manner of other navigation and tracking problems.

In a general sense, estimating only the most-current state vector in the EKF algorithm typically results in computational savings over a conceivable alternative algorithm, such as a naïve nonlinear *least squares* or *batch* implementation, that considers the entire state history when computing the solution. However, in EKF-SLAM the growing size of the state vector as new landmarks are added to it makes approximations and other tricks necessary to maintain computational tractability. This is because the computation of the EKF measurement update requires operations over the entire system covariance matrix, resulting in a computational cost that is cubic in the number of landmark states n, $\mathcal{O}(n^3)$[26, 107]. Realizing that the positive definite structure of the covariance matrix inherently contains redundancy, and that only certain parts of the matrix are affected by the update, this issue can be partially attenuated by utilizing more strategic, alternative formulations[38, 107], which can achieve $\mathcal{O}(n^2)$ or even $\mathcal{O}(n)$ complexity, or by clever submapping techniques.[42, 7] Even so, the unbounded nature of the EKF-SLAM state vector remains a challenging limitation to the algorithm's applicability to real problems.[89, 107]

A further drawback is that generic EKF-SLAM has been shown to be inherently *inconsistent.*[66] As defined in Reference [9], an estimator is consistent if the mean is unbiased and the true covariance is as predicted by the filter. More recent work has concluded that EKF-SLAM's tendency to be overconfident in its estimates stems from a mismatch between the observable subspaces of the underlying nonlinear problem and the linearized version.[60] This mismatch causes the estimator to "gain" spurious information about the global attitude, leading to overconfidence that cascades to other states. Proposed modifications to the EKF to alleviate this discrepancy include the the First Estimates Jacobian EKF (FEJ-EKF) and Observability Constrained EKF (OC-EKF)[72][59], as well as the idea of a *robocentric* formulation wherein the agent is always placed at the origin of the frame and the landmark locations must be updated in time.[57] In visual SLAM, improved consistency can also be obtained by using special landmark parameterizations such as the Unified Inverse Depth Model [22] and Anchored Homogenous Points (AHP)[106] that ensure linear-Gaussian behavior of landmark states at all depths. This will be discussed in detail in Chapter 3 of this thesis.

1.1.2 Particle Filtering SLAM Algorithms

The assumption that the process and measurement models of a sequential filter must be Gaussian can be an onerous limitation in some cases. While relaxing this assumption completely is impossible analytically, Monte Carlo methods, also known as particle filters, can be formulated to approximate non-Gaussianity by the use of many randomly-drawn samples in the state space.[26] The quality of the approximation naturally increases with the number of samples or particles being considered but this leads to a proportional increase in the computational complexity of the algorithm. Coupled with the multiplying influence of the size of the state space, particle filtering quickly succumbs to a so-called "curse of dimensionality".[27] Since landmark-based SLAM typically involves an unbounded state vector size, with hundreds or thousands of landmark states, the simplest particle filter applied to SLAM will quickly become hopelessly intractable.

A more considered approach is the one utilized by Reference [82], which involves the use of a

Rao-Blackwellized particle filter. This algorithm, dubbed FAST-SLAM, represents one of the only truly successful applications of particle filtering methods in the SLAM problem. The main idea is to treat each of m particles as individual estimates of the robot path and, by the observation that knowing the true robot path yields independent estimates of the landmarks, assigning each of n landmarks its own Kalman filter conditioned on the path estimate. Using a carefully-considered data tree-based data structure, they show that the computational complexity can be reduced to $\mathcal{O}(m \log n)$, which is significantly faster than most EKF-SLAM formulations.

1.1.3 Nonlinear Least Squares Algorithms

Around the same time that EKF-SLAM was reaching maturity, and after the seminal work of Lu and Milos[78], research interest shifted to efficient least squares solutions for SLAM[53, 73, 67] and this line of reasoning proved to be highly successful. These estimators leverage the fact that the discrete-time Jacobian matrix describing the linearized relationships between all states is always sparse and can be intuitively interpreted through the application of graphical models.[30] This matrix can be used to derive a first-order approximation of the system Hessian matrix, and is thus typically referred to as the "Hessian" in the SLAM literature. The Hessian is used to solve the normal equations to obtain the least squares estimate of the deviations for all states at all times about a nonlinear reference trajectory. The reference trajectory can then be iterated to convergence to achieve the Maximum A-Posteriori (MAP) solution to the so-called *full SLAM problem.* A more detailed dscussion of the mathematics of this approach is provided in Section 2.3 of this thesis.

While a naïve batch implementation would incur a large computational penalty as a result of solving for many thousands of states simultaneously, careful consideration and construction of the system Hessian matrix can be leveraged to create computationally efficient algorithms.[105, 53, 67] Many algorithms exist in this category and a summary of these and subsequent developments is beyond the scope of this work. The interested reader can find more information in recent literature review papers such as References [17] and [98].

While batch algorithms have largely taken the lead in SLAM research, sequential filtering approaches have persisted and matured in the Visual-Inertial Navigation System (VINS) application domain,[47] where computational limitations can be acute and where difficulties arise from the use of the high-rate sensor data outputs of inertial measurement units (IMUs). These sensors often produce information at rates at or greater than 100 Hz, which in graph-based formulation can introduce hundreds of new nodes on the graph per second. This can become cumbersome quickly and requires careful "pre-integration" strategies to handle efficiently.[45]

1.2 Visual SLAM

The use of cameras for visual SLAM is, in general, significantly more complicated than "range and bearing" measurements but with the considerable benefits of the sensor package being lightweight, compact, low-power, relatively inexpensive, and passive with respect to the environment given good lighting conditions. A measurement of a given image point using a camera amounts to a 2D projection of a semi-infinite ray from the camera's projection center to a distinct point in $\mathbb{R}^3$. The direction of this ray is rather well-constrained by the camera's sensor while the range or depth, in the absence of any other information, is entirely ambiguous, making Structure from Motion (SfM) or Visual SLAM type of "angles-only" or "bearings-only" estimation problem. *Stereo* cameras resolve the depth ambiguity by comparing simultaneous views of the same scene while *monocular* systems can use multiple views across time to a similar effect. Naturally, the ability of stereo configurations to precisely resolve the depth ambiguity rapidly diminishes as the baseline between the two cameras is shortened.[105]

As with range and bearing measurements, the first visual SLAM algorithms were formulated to track individual, visually-identified *features* over successive images. These distinct 2D image features are taken to be projections of 3D *landmarks* that are typically assumed static in a world-fixed reference frame. This type of visual SLAM has been successfully implemented in real-time using both EKF[29] and batch backends.[87] Importantly, landmark-based V-SLAM relies on robust feature descriptors that can be generated and matched between successive frames. Classic examples are Scale Invariant Feature Transform (SIFT)[77] and the binary Oriented FAST and Rotated

BRIEF (ORB)[99] which have each been used extensively in visual SLAM and have implementations in MATLAB and OpenCV. These descriptors were developed with terrestrial applications in mind and the appropriateness of different feature descriptors for spaceborne (small body) visual estimation problems has been assessed in Reference [85]. Reference [126] describes a shadow-based feature descriptor that may be more appropriate for small body missions. Once a set of landmarks has been estimated, it can be used to generate a shape model.[37]

Visual features are not the only method of information extraction from camera images. Direct methods, which directly compare pixel intensity changes over sequences of images, have proven to be a very successful alternative using both EKF[13] and batch[41, 40, 46] backends. These approaches have the added benefit that the data association problem, which can be an onerous limitation for landmark-based approaches, is solved simply and robustly. The application of these techniques to space problems has been rarely demonstrated up to the present time.

1.3 SLAM in Spaceflight

Statistical estimation has a storied history in space and defense applications and, for many years, these are the domains in which the most estimation research was being conducted. The fundamental theory of estimation is perhaps best elucidated by Peter Maybeck's classic three-part textbook,[80] which has proven to be influential and useful to this day. As the more general application and study of robotic systems has become ubiquitous in the 21st century, most novel research on estimation problems has shifted in origin to more application-agnostic disciplines such as computer science. While some earlier applications of SLAM in aerospace problems generated novel estimation architectures,[86, 105] more recent work in aerospace research has sought to adapt proven, open source solutions from computer science to aerospace problems.[113, 74, 34] Thus, most novel SLAM research in aerospace applications has shifted from the backend estimation problem to more application-specific sub-problems and frontend considerations.

Restricting our attention to the spaceflight domain, we envison three possible application scenarios of SLAM: Rendezvous and Proximity Operations(RPO), small body navigation and mapping, and planetary landing and hazard mapping. These three scenarios, while offering unique challenges and possibly requiring different assumptions, are fundamentally only separated by the scale of the body being mapped. Nearly all of the extant SLAM research in the spaceflight domain utilizes either visual measurements or Light Detection and Ranging (LIDAR) measurements or both. The three spaceflight applications of SLAM are depicted notionally in the diagram of Figure 1.1.

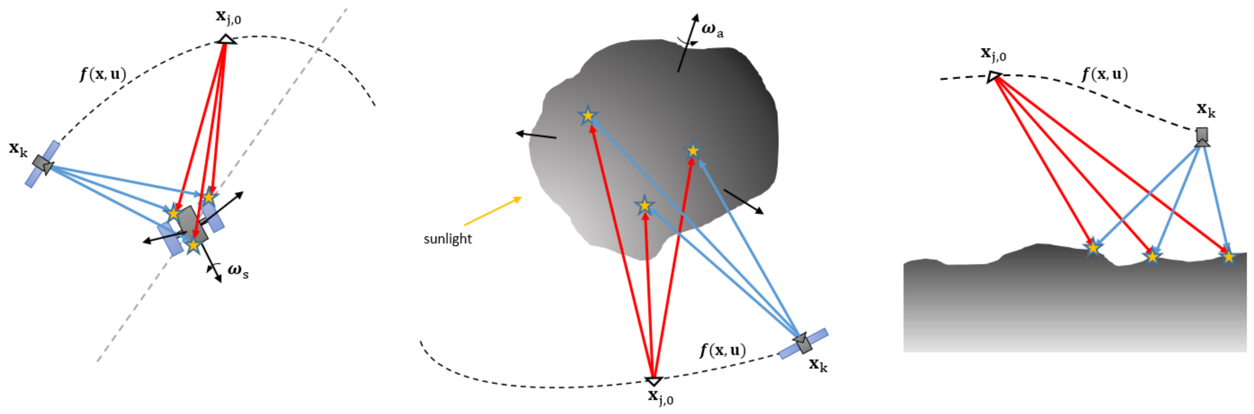


Figure 1.1: Illustration of the various application domains of SLAM in spaceflight problems. Left: Rendezvous and Proximity Operations (RPO). Center: small body relative navigation and mapping. Right: Planetary landing and hazard mapping.

Rendezvous and Proximity Operations (RPO) with respect to a previously-unknown and possibly-uncooperative target spacecraft is an obvious use case for SLAM, where decisions about guidance and control must be made in real time based on geometric information captured and processed onboard an autonomous or manned agent. If the geometry of the target spacecraft is known before the mission, the estimation problem no longer requires SLAM but is instead one of pose estimation exclusively, also called "model-matching."[91] Viewing conditions due to the presence of the Earth can be challenging, both due its daytime presence in images and the eclipse conditions that it can impose. Spacecraft geometry, with its sharp angles and edges, tends to be amenable to corner feature detectors and other traditional computer vision techniques. An uncontrolled target may have a highly-unstable spin state, possibly making estimation of the target's inertia tensor necessary. Machine learning approaches have recently been formulated for model-matching but are still underexplored for SLAM in RPO scenarios.[111]

Many deep space missions are now visiting small bodies such as comets and asteroids. These bodies are typically only roughly characterized before they are visited by the spacecraft, necessestating a rather long, iterative ground-in-the-loop process of relative navigation and shape modeling.[90] The current process utilizes stereophotoclinometry (SPC), which estimates a set of local topological landmark maps (L-maps) using human-identified surface features that are matched across multiple frames. This information is then combined with estimated camera poses and illumination data to estimate a global shape model of the body. While SPC is very effective, SLAM offers a path to an onboard, real-time capable alternative.

Small body scenarios are challenging for SLAM because properties such as the body's gravitational parameter and mass distribution, pole orientation, rotation rate, and surface topography must be estimated before close proximity operations or landing can be attempted. Given that these objects are small on a celestial scale, care must be taken to model small perturbative forces such as solar radiation pressure (SRP) and the gravitational influence of other solar system bodies. Small bodies, despite the name, can come in sizes that span multiple orders of magnitude, from tens of meters to hundreds of kilometers across, and the surface appearance in images can likewise vary substantially between sizes and types of bodies. Craters may or may not be present and lighting conditions are constantly changing. The rotation state of the body may or may not be stable about the principal axes on the mission timescale.

A third SLAM application in spaceflight is mapping and navigation with respect to large celestial objects such as the Moon, especially during a powered descent or final landing phase. While parameters such as the inertial pole direction and rotation rate must be accounted for in the dynamics, they are typically less uncertain than for small bodies. However, landing problems can involve significant scale changes and thrusting forces, possibly necessitating the inclusion of an Inertial Measurement Unit (IMU) in the formulation. Upon final descent, the detection and mapping of hazards with respect to the spacecraft at or near a targeted landing site may be crucial to avoidance and ultimately mission success.

In Table 1.1, we have collected citations to extant SLAM work in these three spaceflight applications and categorized them based on measurement type (Visual/LIDAR) and backend estimation architecture. These references only include work that has jointly estimated both the spacecraft or lander states and some representation of the environment, disqualifying closely-related subjects like Visual Odometry,[21] velocimetry,[64] and other forms of Terrain-Relative Navigation (TRN)[65] that do not explicitly retain or estimate a map representation.

Table 1.1: Existing SLAM Research in Spaceflight Applications

	Rendezvous & Proximity Operations		Small Bodies		Planetary	
	Visual	LIDAR	Visual	LIDAR	Visual	LIDAR
KF	[112],[103],[62],[63],[15]	[76]	[32],[5],[52],[51]		[125],[3],[49],[51]	
Batch	[124],[69],[120],[36],[74],[70]		[32],[95],[96],[8],[97],[94],[35],[37],[39],[34]	[11]	[5],[113],[114]	[5],[104],[79]
Other/Hybrid	[6]	[102],[101]	[24],[23],[25],[118]	[88]		

Following the discussions of the previous sections, we can outline some unique features of the space environment that can make the visual SLAM problem even more challenging than in many terrestrial applications.

- **Complex dynamics** - Even the simplest spacecraft dynamics models for position and velocity are nonlinearly dependent on the position of the spacecraft. Though approximations for the State Transition Matrix (STM) can be formulated, they degrade sharply as the evaluation time interval is increased. The most robust way to capture its behavior, particularly in strongly-perturbed environments, is through numerical integration. The computational behavior of popular open-source SLAM algorithms may not be as-advertised if this is not taken into account.

- **Rotating Frames** - While landmarks can be considered static with respect to whatever body they inhabit, the body itself may be rotating. If this rotation is not stable about the principal axis of the body, and the inertia tensor is not known *a priori*, it may be necessary to estimate it jointly with the other states. On the flip side, if a body is known to be a principal-axis rotator, the observability of the system may be improved due to the presence of this inertially-stable reference.

- **Constrained computational resources** - The space environment is harsh and, unless a dedicated computer is given to the SLAM routine, many other necessary spacecraft functions may be assigned to the same radiation-hardened computer. These computers are typically far behind the state-of-the-art in terms of processing capabilities.[14]
- **Challenging lighting conditions** - The main source of optical illumination in space is the Sun and without the Earth's atmosphere, scenes can be harshly lit with extremely high contrast. In addition, shadows on spacecraft and small bodies often change significantly between images, necessitating robust approaches to feature or template matching and outlier rejection.
- **Lack of corner features** - Many feature descriptors, such as Harris corners and Difference of Gaussians, are based on finding and matching sharp corner-like features in images. For larger small bodies and some moons, many images may not exhibit any sharp edge or corner structures.

On the other hand, as will be discussed in Chapter 3, the availability of mature star tracking technology makes the estimation of attitude unnecessary many space problems, possibly simplifying the underlying estimation problem significantly and side-stepping the attitude inconsistency issues of EKF-SLAM. In addition, camera calibration during spaceflight may be more straightforward than terrestrial problems since starfields can be used for precise calibration after launch.[20]

1.4 Thesis Statement

As the burgeoning space economy has begun to generate ever more robotic and manned space missions to a diverse array of celestial targets, and as ground-based navigation resources become more constrained as a result, autonomous capabilities of spacecraft systems will move to the forefront of importance. SLAM is one of the key technologies that will enable these systems and its application to space problems is still in its infancy when compared to its use in robotics. The motivation of this thesis is to understand what has been done in SLAM in robotics, the assumptions and limitations of those techniques, and the ways in which their application in the spaceflight domain must be carefully considered. In addition, the theoretical contributions of this thesis can be summarized as:

- In Chapter 3, we present a new view-based inverse depth measurement model based on homogeneous coordinates that better captures the directional uncertainty in the initial direction from the anchor pose to an observed visual feature. This comes at the cost of two extra states per feature.

- Chapter 4 presents a novel square root information filter design for visual and visual-inertial odometry applications that represents the first known use of an inverse depth model within an information filter. Because of the presented augmentation scheme, the algorithm can jointly estimate both the state of the vehicle and environmental landmarks with constant time complexity.

- Chapter 5 refines the square root information filter algorithm by introducing a "mean reconciliation" step. The former allows for subsequent relinearizations to be accounted-for in previously-estimated states with little extra computational effort, improving the consistency of the solution and providing a smooth trajectory at the end.

- Chapter 5 also introduces an efficient "relocalization" or "loop closure" routine, which is a novel application of the "Schmidt" or "consider" update in order to process new information from previously-tracked landmarks.

1.5 Thesis Overview

Chapter 2 describes the landmark-based EKF and batch SLAM approaches in more mathematical detail and discusses their tradeoffs with respect to their application to space problems. The systems theory and models used in the later chapters of this thesis are described in Chapter 3 before a novel visual-inertial odometry algorithm is proposed in Chapter 4 and demonstrated on simulated data. Chapter 5 builds on the previous algorithm and presents Inverse Incremental Pose Augmentation SLAM (IIPA-SLAM), a new algorithm with improved capabilities that combines elements of both EKF and batch SLAM approaches, and applies it to both simulated and real datasets.

1.6 Associated Publications

The following list contains the conference and journal publications directly associated with the work in this thesis.

Conference Presentations

- Matthew Givens and Jay McMahon. *Nearly Constant-Time SLAM-Based Terrain Relative Navigation for Landing on an Uncharted World.* 2021 American Astronautical Society Spaceflight Mechanics Meeting. Big Sky, Montana. (virtual) 2021.

- Matthew Givens and Jay McMahon. *Square Root, Sequential Visual Odometry for Constant-Time Navigation and Mapping.* AIAA SCITECH 2022 Forum. San Diego, California. 2022.

- Matthew Givens and Jay McMahon. *Square Root Extended Information Filter Visual Odometry Applied to Blue Origin Deorbit, Descent, and Landing Dataset.* 3rd Space Imaging Workshop. Atlanta, Georgia. 2022.

- Matthew Givens, Jacopo Villa, and Jay McMahon. *Computationally Efficient Visual-Inertial SLAM for Asteroid-Relative Navigation.* Austin, Texas. 2023 American Astronautical Society Spaceflight Mechanics Meeting. 2023.

Journal Articles

- Matthew Givens and Jay McMahon. *Square-Root Extended Information Filter for Visual-Inertial Odometry for Planetary Landing.* Journal of Guidance, Control, and Dynamics. 2023.

- Matthew Givens and Jay McMahon. *Computationally-Efficient Asteroid-Relative Visual SLAM Using Inverse Incremental Pose Augmentation.* Journal of Guidance, Control, and Dynamics. 2023-2024. (in progress)

Other work by the author while at CU Boulder:

Conference Presentations

- Matthew Givens and Calvin Coopmans. *Exploring the Use of Reverse Thrust in a Dynamic UAS Landing Maneuver using Kinodynamic RRT*. 2020 International Conference on Unmanned Aircraft Systems (ICUAS). Athens, Greece. (virtual) 2020.

- Matthew Givens and Jay McMahon. *Unscented Kalman Filter Using Modified Spherical Coordinates for Passive Spacecraft Angles-only Relative Navigation.* 2022 AAS Astrodynamics Specialist Conference. Charlotte, North Carolina. 2022.

Journal Articles

- Matthew Givens and Jay McMahon. *Assessing Modified Spherical Coordinates for Sequential Angles-Only Relative Spacecraft Navigation* . Journal of Guidance, Control, and Dynamics. 2023. (under review)

Chapter 2

Formulating the SLAM Problem

This chapter provides mathematical foundations for working with nonlinear systems in a SLAM context and serve to contextualize the succeeding sections of the present work. Linear systems theory provides powerful tools for the description and analysis of linear systems. However, even the most simplified forms of both the SLAM problem and spacecraft navigation involve nonlinear dynamics and measurement models and so these results must necessarily be extended. We then describe the EKF and batch approaches to SLAM before offering a discussion of the tradeoffs between the two within the context of spaceflight applications.

2.1 Linearized System Models

Properly, the state $\mathbf{x}$ at some time t evolves according to the nonlinear stochastic differential equation

$$\mathbf{dx}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)\mathrm{d}t + \mathbf{G}(t)\mathbf{d}\boldsymbol{\beta}(t) \tag{2.1}$$

where $\mathbf{u}$ is some deterministic control input which is assumed to be known and $\boldsymbol{\beta}$ is a zero-mean Brownian vector process with strength $\mathbf{Q}(t)$ mapped into the state space by the time-varying matrix $\mathbf{G}(t)$.[80] Less formally, this can be rewritten in a "white noise" notation as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2.2}$$

where the Gaussian white noise $\mathbf{w}(t)$ is hypothetically interpreted as the derivative of $\boldsymbol{\beta}(t)$ with covariance kernel

$$\mathbf{E}[\mathbf{w}(t)\mathbf{w}(t+\tau)^T] = \mathbf{Q}(t)\delta(\tau) \tag{2.3}$$

where $\mathbf{E}[\cdot]$ is the expectation operator and $\delta(\cdot)$ is the Dirac delta function such that $\mathbf{w}(t)$ is uncorrelated with itself except at $t = \tau$, hence it is "white". Measurements are taken at discrete times t_k according to the nonlinear measurement model

$$\mathbf{z}(t_k) = \mathbf{h}(\mathbf{x}(t_k), t_k) + \mathbf{v}(t_k) \tag{2.4}$$

with $\mathbf{v}$ being zero-mean, white Gaussian noise with covariance $\mathbf{V}(t_k)$.

A Taylor series expansion can be utilized to linearize 2.2 about a nonlinear reference $(\tilde{\cdot})$ state,

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \approx \mathbf{f}(\tilde{\mathbf{x}}, \mathbf{u}(t), t) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}(t)} (\mathbf{x}(t) - \tilde{\mathbf{x}}(t)) \tag{2.5}$$

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) - \mathbf{f}(\tilde{\mathbf{x}}, \mathbf{u}(t), t) = \delta\dot{\mathbf{x}}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\tilde{\mathbf{x}}(t)} \delta\mathbf{x}(t) \tag{2.6}$$

so that the new model for the state deviations about the reference follows

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t, \tilde{\mathbf{x}}(t))\delta\mathbf{x} + \mathbf{G}(t)\mathbf{w} \tag{2.7}$$

where $\mathbf{F}$ is the Jacobian matrix of partial derivatives of $\mathbf{f}$ with respect to $\mathbf{x}$ in the preceeding equations. The measurement equation can likewise be linearized according to

$$\mathbf{z}(t_k) \approx \mathbf{h}(\tilde{\mathbf{x}}(t_k)) + \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\tilde{\mathbf{x}}(t_k)} \delta\mathbf{x}(t_k) + \mathbf{v}(t_k) \tag{2.8}$$

$$\delta\mathbf{z}(t_k) = \mathbf{H}\delta\mathbf{x}(t_k) + \mathbf{v}(t_k) \tag{2.9}$$

where $\mathbf{H}$ is the Jacobian of $\mathbf{h}$ with respect to $\mathbf{x}$. Given new information from a processed measurement, the nonlinear state could be corrected according to the additive relationship,

$$\tilde{\mathbf{x}}(t_k)^+ = \tilde{\mathbf{x}}(t_k)^- + \delta\tilde{\mathbf{x}}(t_k) \tag{2.10}$$

In the case of attitude estimation, where an additive state deviation relationship can be problematic, a multiplicative deviation formulation is commonly used with a reference unit quaternion to parameterize the global rotation. This specific case is addressed more closely in Section 3.2.1.

Concepts from linear estimation can now be applied to this linearized system centered about the nonlinear reference state. The mean of the reference state itself can be propagated without knowledge of the noise according to the expected value of Equation 2.2 and, given the additive relationship to the state deviations in Equation 2.7 and under the assumption of Gaussianity, its covariance matrix, which is the second central moment of the state deviations, evolves according to the matrix differential equation

$$\dot{\mathbf{P}}(t) = \mathbf{F}(t, \tilde{\mathbf{x}}(t))\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(t, \tilde{\mathbf{x}}(t))^T + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}(t)^T \tag{2.11}$$

Equivalently, the solutions to Equations 2.2 and 2.11 between t_k and t_{k+1} take the integral forms

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \int_{t_k}^{t_{k+1}} \mathbf{f}(\tilde{\mathbf{x}}(\tau), \mathbf{u}(\tau), \tau)d\tau = \boldsymbol{\phi}_{k+1,k} \tag{2.12}$$

$$\mathbf{P}_{k+1} = \boldsymbol{\Phi}(t_{k+1}, t_k)\mathbf{P}_k\boldsymbol{\Phi}(t_{k+1}, t_k)^T + \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(\tau, t_k)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}(\tau)^T\boldsymbol{\Phi}(\tau, t_k)^T d\tau \tag{2.13}$$

where the notation has been simplified such that a subscript k denotes a quantity's value at t_k. $\boldsymbol{\phi}_{k+1,k}$ is known as the *solution flow* or *flow function.* In a standard EKF, numerical integration of Equation 2.11 is typically more convenient than computation of Equation 2.13, which involves numerically integrating the state transition matrix (STM) according to

$$\dot{\boldsymbol{\Phi}}(t) = \mathbf{F}(t, \tilde{\mathbf{x}}(t))\boldsymbol{\Phi}(t) \tag{2.14}$$

with initial condition

$$\boldsymbol{\Phi}(t_k, t_k) = \mathbf{I}_{n\times n} \tag{2.15}$$

and the second, integral term according to[80]

$$\dot{\tilde{\mathbf{Q}}}(t) = \mathbf{F}(t, \tilde{\mathbf{x}}(t))\tilde{\mathbf{Q}}(t) + \tilde{\mathbf{Q}}(t)\mathbf{F}(t, \tilde{\mathbf{x}}(t))^T + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}(t)^T \tag{2.16}$$

with initial condition

$$\tilde{\mathbf{Q}}(t_k, t_k) = \mathbf{0}_{n\times n} \tag{2.17}$$

where n here is the size of the state. Worth noting is that the solution to Equation 2.16 can come out to be non-positive definite if the numerical integration scheme used is not accurate enough over the integration interval. For the batch and information estimators, the integral forms will be necessary for the derivations to come.

Various approximations exist for these relationships that are useful in some circumstances. Given a "short" time interval, the state transition matrix can be approximated to first order via

$$\mathbf{\Phi}(t_{k+1}, t_k) \approx \mathbf{I} + \mathbf{F}(t_k, \tilde{\mathbf{x}}_k)\Delta t \tag{2.18}$$

where $\mathbf{I}$ is the identity matrix and $\Delta t = t_{k+1} - t_k$. The validity of this and other, more sophisticated approximations, as well as what "short" means in this context, is discussed in Reference [16]. The second term in Equation 2.13 can be approximated as

$$\int_{t_k}^{t_{k+1}} \mathbf{\Phi}(\tau, t_k)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}(\tau)^T\mathbf{\Phi}(\tau, t_k)^T d\tau \approx \left(\int_{t_k}^{t_{k+1}} \mathbf{\Phi}(\tau, t_k)\mathbf{G}(\tau)d\tau\right)\mathbf{Q}_k\left(\int_{t_k}^{t_{k+1}} \mathbf{\Phi}(\tau, t_k)\mathbf{G}(\tau)d\tau)\right)^T \tag{2.19}$$

or, more succinctly,

$$\tilde{\mathbf{Q}}(t_{k+1}, t_k) = \mathbf{\Gamma}(t_{k+1}, t_k)\mathbf{Q}_k\mathbf{\Gamma}(t_{k+1}, t_k)^T \tag{2.20}$$

where

$$\mathbf{\Gamma}(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(\tau, t_k)\mathbf{G}(\tau)d\tau \tag{2.21}$$

This relies on a zero-order hold assumption which assumes that $\mathbf{w}$ is constant over a "small" time interval. References [119] and [16] provide useful discussions of this approximation and its consequences. Whether approximated or integrated using Equation 2.16, the process noise covariance must be handled carefully to ensure consistent application.

Given the previous derivations, two possible approaches for computing the nonlinear reference trajectory and state deviations become apparent. The first is the approach of the *linearized* Kalman Filter (LKF), also confusingly dubbed the *classical* or *conventional* Kalman Filter (CKF) in some sources, where the full reference trajectory is integrated forward over some time interval and the state deviations for all measurement times computed along this reference. Computation of the deviations can be computed by either a sequential method, as in the LKF, or all at once using a batch least squares method.[119] Given a complete solution, the reference trajectory can be reintegrated and the process repeated until convergence. The LKF and batch produce the same solution at all times if the LKF solution at the final time is *smoothed* backward to the initial time.

The other approach is the one of the Extended Kalman Filter (EKF) which is to relinearize about a *new* nominal trajectory after each measurement is processed. The solution of the EKF is therefore fundamentally different from the LKF and batch methods: it does not result in a consistent, smooth trajectory over time but instead prioritizes the most recent state estimate, resulting in an often jagged-looking state estimate history. A smooth trajectory can be obtained by integrating or smoothing the final solution backward in time. Unlike the LKF and batch, the EKF solution is not typically iterated over the trajectory since the relinearization happens instead in a sequential fashion. Another result of this strategy is that the state deviations themselves need not be mapped between measurement times using Equation 2.7 because the new prior best estimate at the time of an update is always zero, enabling the EKF to utilize the differential form of the covariance propagation in Equation 2.11 instead of using the integral form in Equation 2.13 to compute the STM. Another result is that the EKF typically converges to the best estimate faster than the LKF or batch given a single pass through of the data, although in practice it can be smart to initialize the EKF with an LKF for a few measurements for improved stability.[119] EKF measurement updates can also be iterated since a new nominal state is obtained when each measurement is processed, typically referred to as an IEKF or IKF,[10] and this generally improves the algorithm's robustness to more pronounced nonlinearities. These observations will have important implications in the succeeding sections.

2.2 Extended Kalman Filter SLAM

The following section will give mathematical definition to EKF-SLAM concepts and illustrate some opportunities and drawbacks in this methodology. The interested reader can find a more complete Bayesian derivation of landmark-based SLAM in Reference [127]. Notionally, the SLAM problem is to estimate a joint probability density function that is defined over the states of a robot or vehicle ("agent") and some parameterization of the environment. Assuming that these quantities have a multivariate Gaussian distribution, this can be represented using a mean vector and covariance matrix,

$$\boldsymbol{\mu} = \begin{bmatrix} \mathbf{x} \\ \mathbf{m} \end{bmatrix}_{n\times 1} \qquad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xm} \\ \mathbf{P}_{mx} & \mathbf{P}_{mm} \end{bmatrix}_{n\times n} \tag{2.22}$$

where $\mathbf{x}$ represents the agent's *most recent* position and any other dynamic states of interest, and $\mathbf{m}$ encompasses the landmark position states.

Using the building blocks from the previous section, the standard EKF [16] propagates the prior mean and covariance by

$$\hat{\boldsymbol{\mu}}_{k+1}^{-} = \mathbf{f}(\hat{\boldsymbol{\mu}}_k, t) \tag{2.23}$$

$$\mathbf{P}_{k+1}^{-} = \boldsymbol{\Phi}(t_{k+1}, t_k)\mathbf{P}_k\boldsymbol{\Phi}(t_{k+1}, t_k)^T + \boldsymbol{\Gamma}(t_{k+1}, t_k)\mathbf{Q}_k\boldsymbol{\Gamma}(t_{k+1}, t_k)^T \tag{2.24}$$

where $(+)$ and $(-)$ notation denotes quantities *prior* and *posterior* to the measurement update. As described previously, Equation 2.11 could also be used for the covariance propagation. The posterior mean and covariance are obtained via the standard Kalman update relationships

$$\hat{\boldsymbol{\mu}}_k^{+} = \hat{\boldsymbol{\mu}}_k^{-} + \mathbf{K}(\mathbf{z}_k - \hat{\mathbf{z}}_k) \tag{2.25}$$

$$\mathbf{P}_k^{+} = (\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{-} \tag{2.26}$$

where

$$\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{2.27}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T + \mathbf{V}_k \tag{2.28}$$

where $\mathbf{K}_k$ is called the "Kalman gain" and $\mathbf{S}_k$ is the "innovation covariance" that combines the state covariance, projected into measurement space by $\mathbf{H}_k$, and the measurement covariance $\mathbf{V}_k$.

If the state ordering from equation 2.22 is used, and assuming all of the landmarks are static in a "world-fixed" reference frame, then the state transition matrix in equation 2.24 takes the form

$$\boldsymbol{\Phi}(t_{k+1}, t_k) = \begin{bmatrix} \boldsymbol{\Phi}_{xx}(t_{k+1}, t_k) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{dim(\mathbf{m})} \end{bmatrix} \tag{2.29}$$

which implies that this operation can be efficiently computed regardless of number of landmark states. On the other hand, Equation 2.26 requires multiplying $\mathbf{K}_k$ by $\mathbf{H}_k$ before subtraction from $\mathbf{I}_n$, an operation that scales quadratically with the size of the state.[38]

One solution to the time complexity of EKF-SLAM is to simply marginalize out old landmark states, leading to a type of visual-inertial odometry (VIO) that estimates the states of landmarks only while they are "active" in the state vector. These techniques, while they solve the tractability issue and can provide useful odometry for the agent, do not generate consistent solutions over the whole map because new information is not fed back into the correlations with older landmarks. This can make loop closure, which is the process of rediscovering older landmarks and using this information to correct the drift accumulated in the intervening time, problematic due to overconfidence.

One very successful VIO algorithm is the Multi-State Constraint Kalman Filter (MSCKF) which uses EKF methodology alongside a clever nullspace projection measurement model.[86] The main idea is to process all of the measurements to a given visual landmark obtained over multiple images at once. By processing full tracks instead of individual measurements as they appear, the data association problem is simplified and the results are shown to be more accurate and consistent than a more generic EKF VIO solution.[75] The MSCKF has subsequently been extended to include "SLAM features" whose positions are estimated by the filter but then subsequently marginalized from the system.[129]

2.2.1 Sparse Extended Information Filters

Eventually, EKF-SLAM researchers noticed that the canonical form of the covariance matrix, which corresponds to its inverse and is also commonly known as the information matrix or precision matrix, is *naturally sparse* in contrast to the *dense* structure of the covariance matrix.[121] Sparsity represents the relative number of zero to nonzero entries present in a matrix. Large sparse matrices have the significant benefit that they can be stored and accessed much more efficiently than standard matrices by converting the nonzero entries to ordered lists of numbers.

Intuitively, the sparsity of the information matrix arises from the fact that the inverse of a large variance, which represents a low certainty, is a small number. Infinite uncertainty corresponds to zero information. Therefore, entries in a large matrix that are only very weakly correlated will have large values in a covariance matrix but very small values in an information matrix. This structure is convenient for SLAM because new landmarks to be estimated will only be weakly correlated directly to older landmarks that were previously estimated when the agent was in a different location.

Another useful property of the information filter approach is that, in contrast to the standard Kalman formulation, the measurement update does not require computing a "Kalman Gain" as in Equation 2.26 and the new information is purely additive. The uncertainties and their relationships that are parameterized by the information matrix can therefore be very-efficiently updated. This has made the information filter very useful in distributed estimation problems.[18] On the other hand, Reference [127] points out that this advantage comes at the expense of a more costly state propagation routine, in large part because the mean of the distribution must be recovered for the linearization of the dynamics and measurement models.

Using these insights, a new class of EKF-SLAM algorithms called *Sparse Information Filters* was developed. By carefully zeroing weak correlations in the information matrix and leveraging the additive nature of the information matrix measurement update, the Sparse Extended Information Filter (SEIF) [121] was able to achieve constant time complexity using sparse matrix libraries. However, state propagation in the SEIF still requires recovering the mean of of the agent's states and any active landmark states, necessitating either a very-costly inversion of the typically very-large information matrix or an approximation thereof. In addition, the statistical *d-separation* technique proposed for strategically deleting correlations and maintaining sparsity, termed "sparsification", invariably results in significant inconsistency issues in the filter solution. This drawback is downplayed in the original work but we believe this cannot be ignored, at least not in precision aerospace applications.

Subsequent variations on the SEIF, notably the Exactly Sparse Delayed State Filter (ESDF)[44] and Exactly Sparse Extended Information Filter (ESEIF)[127], improved the viability of the information filter concept in different ways. The latter approach marginalizes the robot's states out of the overall probability distribution and requires a "relocalization" function, which amounts to an inverse measurement model, to emplace the agent back into the probability distribution. This maintains the sparsity of the information matrix but requires a robust relocalization function which can become onerous with limited sensor configurations and in 3D problems.[49] The ESDF is not a feature-based algorithm and retains exact sparsity through the observation that state augmentation obeys the Markov Property: *state augmentation of a propagated state does not induce any correlations other than to the prior state from which it was generated.* Despite these improvements, both the ESDF and ESEIF still require approximate inversions to the information matrix like the original SEIF. Additionally, the numerical properties of the information filter are now known to be poor for visual-inertial navigation.[129]

2.2.2 Square Root Information Filtering

The idea of using the square root of the covariance or the square root of the information matrix is not new to aerospace applications, having been developed in the 1960s for use on the Apollo missions due to its increased accuracy on finite-word-length computers over standard Kalman filtering methods.[12][81] Indeed, Bierman's classic text on the subject[12] convincingly advocates that square root methods should *always* be used instead of the standard Kalman Filter because of their superior numerical stability. Despite this, the use of the square root information matrix (SRI) is not common in the filtering-based SLAM literature, possibly due to its reputation for having higher computational demands in general. Two noteworthy exceptions are square-root inverse sliding window filter (SR-ISWF)[129] and Resource-aware Inverse Schmidt Estimator (RISE-)SLAM[68] which are closely related and represent the closest point-of-comparison to the present work.

An interesting feature of the SRI matrix is that it maps directly to a Bayes Network, a graphical model that is very useful for inference problems.[31] This special, directed structure of this graph is leveraged in our work. For a covariance matrix, a non-unique square root information matrix $\mathbf{R}$ can be defined as

$$\mathbf{P} = \mathbf{\Lambda}^{-1} = \mathbf{R}^{-1}\mathbf{R}^{-T} \tag{2.30}$$

where $\mathbf{\Lambda}$ is the information matrix and $\mathbf{R}$ is obtained by its Cholesky factorization. The analogue of the mean of the multivariate distribution is the square root information vector,

$$\mathbf{b} = \mathbf{R}\boldsymbol{\mu} \tag{2.31}$$

which amounts to a weighting of the mean $\boldsymbol{\mu}$ by $\mathbf{R}$. Taking the matrix $\mathbf{R}$ to be upper-triangular, note that the marginal distribution of some set of N states, starting from the lower-right entry of the matrix, is nothing more than the $N \times N$ sub-block of the matrix in question. In practice, this means that the marginal distribution can be very cheaply extracted if the desired states are located in the lower-right corner of the matrix. To prove this fact, consider the well-known marginalization operation of the standard information matrix,[43] where, for example, we wish to extract the marginal distribution of $\mathbf{\Lambda}_{22}$ via the Schur complement:

$$\mathbf{\Lambda}_m = \mathbf{\Lambda}_{22} - \mathbf{\Lambda}_{21}\mathbf{\Lambda}_{11}^{-1}\mathbf{\Lambda}_{12} \tag{2.32}$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_{11} & \mathbf{\Lambda}_{12} \\ \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_{22} \end{bmatrix}. \tag{2.33}$$

In contrast to the state ordering in Equation 2.22, suppose that the landmark states, which could be numerous, are contained in the $\mathbf{\Lambda}_{11}$ block of the matrix and the most recent agent states are contained in the $\mathbf{\Lambda}_{22}$ block. In this case, a costly inversion of $\mathbf{\Lambda}_{11}$ is evidently required to recover the associated mean. Instead, here we explicitly write the components of the upper triangular square root information matrix as

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_2 \end{bmatrix} \tag{2.34}$$

and carry out the multiplication to yield an equivalent definition of $\mathbf{\Lambda}$:

$$\mathbf{R}^T\mathbf{R} = \begin{bmatrix} \mathbf{R}_1^T\mathbf{R}_1 & \mathbf{R}_1^T\mathbf{R}_{12} \\ \mathbf{R}_{12}^T\mathbf{R}_1 & \mathbf{R}_{12}^T\mathbf{R}_{12} + \mathbf{R}_2^T\mathbf{R}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{\Lambda}_{11} & \mathbf{\Lambda}_{12} \\ \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_{22} \end{bmatrix}. \tag{2.35}$$

Substituting the corresponding terms into Eq. 2.32 from above and simplifying yields

$$\mathbf{\Lambda}_m = \mathbf{R}_{12}^T\mathbf{R}_{12} + \mathbf{R}_2^T\mathbf{R}_2 - (\mathbf{R}_{12}^T\mathbf{R}_1)(\mathbf{R}_1^T\mathbf{R}_1)^{-1}(\mathbf{R}_1^T\mathbf{R}_{12}) \tag{2.36}$$

$$\mathbf{\Lambda}_m = \mathbf{R}_2^T\mathbf{R}_2 \tag{2.37}$$

and by inspection, the desired square root of the marginal information $\mathbf{\Lambda}_m$ up to a sign ambiguity is simply

$$\mathbf{R}_m = \mathbf{R}_2. \tag{2.38}$$

A similar approach can be used to prove that the information vector of the marginal distribution is simply

$$\mathbf{b}_m = \mathbf{b}_2 \tag{2.39}$$

from which the mean can then be computed very quickly via back-substitution using Equation 2.31 because $\mathbf{R}_2$ is upper-triangular.

Thus, the marginal distribution of states beginning at the lower right corner of the upper-triangular square root information matrix can be recovered very cheaply without approximation. Furthermore, if desired, the mean and covariance of the entire distribution can also be very efficiently calculated via back-substitution operations on the entire square root matrix and vector, albeit not in a constant-time fashion as the matrix grows in size. Following the method shown above, the interested reader will find that there is no simple analytical solution for finding the marginal distribution of subsets of states that exclude the lower right entries in the matrix unless the subset in question is entirely uncorrelated with them. This property, along with the observation that, like the information filter, augmented states follow the Markov Property, will be leveraged in the algorithm presented in Chapter 4 of this thesis.

A similar understanding of the square root information matrix and its properties vis-à-vis SLAM is presented in the detailed report in Reference [130] and a subsequent publication by the same authors.[68] Their algorithm, Resource-aware Inverse Schmidt Estimator (RISE-) SLAM, is based on the MSCKF and takes advantage of the structure of the square root as well as the idea of the Schmidt update, classically formulated for the Kalman filter.[81] The Schmidt update "considers" the known uncertainty of all states involved but only updates a subset of these states with new information. This idea is commonly used in spacecraft navigation to consider, for example, the uncertainties in the ephemerides of celestial bodies or the imprecision of instrument locations. RISE-SLAM utilizes a novel approximation to the Schmidt update, in a square root inverse formulation, to enable computationally efficient, consistent relocalization given observations of previously-estimated landmarks. These landmarks are not estimated further but the new information is used to update the agent states. Their formulation for the Schmidt update in square root information space assumes the state ordering in Equation 2.22 and they rely on frequent re-ordering operations, and the required re-factorizations, to leverage it. Inspired by this approach, we take a slightly different tack in Chapter 5 to accomplish the same goal without re-ordering the state.

2.3 Batch Maximum A Posteriori Solution

As described in Chapter 1, much of the research in SLAM at the time of this writing has moved on from the EKF to graph-based batch solvers. An excellent primer on the development of this theory is given by Reference [30] and the key mathematical developments therein are recounted here. That work eventually spawned the more sophisticated iSAM2 solver and the open source GTSAM library[31] which are used widely today. Other groups also developed their own understandings of this approach in parallel,[53] notably spawning the open source g^{2}o library[73] and successful algorithms such as ORB-SLAM.[87] These algorithms draw parallels to classical *bundle adjustment* (BA) from the older field of photogrammetry as well as Structure from Motion (SfM). A helpful unifying perspective on these topics is given by the thesis in Reference [105].

A model for the joint probablity distribution over the set of all agent states X, landmark states L, and measurements Z, assuming that these are conditionally independent from each other, can be written as

$$P(X, L, Z) = P(\mathbf{x}_0) \prod_{k=1}^{K} P(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \prod_{m=1}^{M} P(\mathbf{z}_m, \mathbf{x}_{k_m}, \mathbf{l}_{i_m}) \tag{2.40}$$

where $P(\mathbf{x}_0)$ is a prior on the initial state $P(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ are transition densities associated with the motion (dynamics) model, and $P(\mathbf{z}_m, \mathbf{x}_{k_m}, \mathbf{l}_{i_m})$ are densities associated with measurements to individual landmarks $\mathbf{l}_i$. As before, the control term $\mathbf{u}_k$ is considered to be deterministic and known. Since the measurements are not unknowns, the previous equation is equivalent to $P(X, L|Z)$.

The Maximum A Posteriori (MAP) estimate for the unknowns $\mathbf{\Theta} = X \cup L$, is found by maximizing the joint probability or minimizing its negative log likelihood,

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\operatorname{argmax}} P(X, L, Z) = -\underset{\mathbf{\Theta}}{\operatorname{argmin}} \Big(\log P(X, L, Z) \Big) \tag{2.41}$$

The nonlinear dynamics model from Section 2.1 can be written in a discretized form as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, t_{k-1}) + \mathbf{\Gamma}_{k-1}\mathbf{w}_{k-1} \tag{2.42}$$

and the nonlinear measurement model is again

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, t_k) + \mathbf{v}_k \tag{2.43}$$

These can each be associated with the Gaussian models in Equation 2.40 according to the definition of the Gaussian distribution,

$$P(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \propto \exp -\frac{1}{2} \|\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, t_{k-1}) - \mathbf{x}_k\|^2_{\tilde{\mathbf{Q}}_k} \tag{2.44}$$

$$P(\mathbf{z}_m, \mathbf{x}_{k_m}, \mathbf{l}_{i_m}) \propto \exp -\frac{1}{2} \|\mathbf{h}(\mathbf{x}_k, t_k) - \mathbf{z}_k\|^2_{\mathbf{V}_k} \tag{2.45}$$

where the proportionality comes from the fact that the scalar coefficient can be neglected without loss of generality because it is independent of $\mathbf{\Theta}$. The Mahalanobis distance of a vector quantity $\mathbf{v}$ weighted by a covariance matrix $\mathbf{\Sigma}$, follows

$$\|\mathbf{v}\|^2_{\mathbf{\Sigma}} = \mathbf{v}^T \mathbf{\Sigma}^{-1} \mathbf{v} \tag{2.46}$$

and will be important for this and other subsequent derivations. Returning to Equation 2.41, we can now make the arguments more explicit,

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \left[\|\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, t_{k-1}) - \mathbf{x}_k\|^2_{\tilde{\mathbf{Q}}_k} + \|\mathbf{h}(\mathbf{x}_k, t_k) - \mathbf{z}_k\|^2_{\mathbf{V}_k} \right] \tag{2.47}$$

where $\mathbf{x}_0$ is assumed to be given. Since this equation is clearly nonlinear, it can be linearized and solved iteratively. Using the same linearization approaches as in Section 2.1, the equivalent linearized model is

$$\boldsymbol{\delta\Theta}^* = \underset{\boldsymbol{\delta\Theta}}{\operatorname{argmin}} \left[\|\boldsymbol{\Phi}(t_k, t_{k-1})\delta\mathbf{x}_{k-1} + \mathbf{x}_k - \mathbf{a}_k\|^2_{\tilde{\mathbf{Q}}_k} + \|\mathbf{H}_{m_k}\delta\mathbf{x}_k + \mathbf{J}_{i_k}\delta\mathbf{l}_i - \mathbf{r}_k\|^2_{\mathbf{V}_k} \right] \tag{2.48}$$

where the measurement Jacobians have been broken up into state-related terms $\mathbf{H}_{m_k}$ and landmark-related components $\mathbf{J}_{i_k}$ and where $\mathbf{a}_k$ is an "odometry prediction error" and $\mathbf{r}_m$ is the measurement residuals,

$$\mathbf{a}_k = \mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, t_{k-1}) \tag{2.49}$$

$$\mathbf{r}_m = \mathbf{z}_k - \hat{\mathbf{z}}_k \tag{2.50}$$

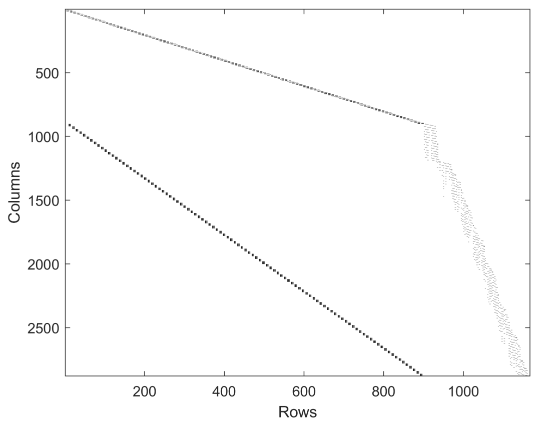
Using the definition of the Mahalanobis norm in Equation 2.46, the two terms in the cost function in Equation 2.48 can be combined so that the full cost function becomes

$$\boldsymbol{\delta\Theta}^* = \underset{\boldsymbol{\delta\Theta}}{\operatorname{argmin}} \|\mathbf{A}\boldsymbol{\delta\Theta} - \mathbf{b}\|^2 \tag{2.51}$$

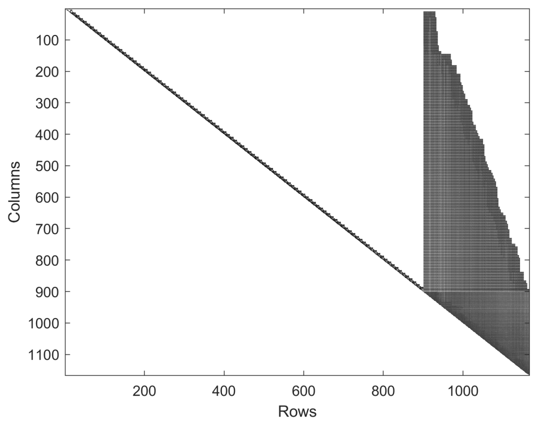
which is a linear equation that can be solved via the normal equation of standard least squares,

$$(\mathbf{A}^T\mathbf{A})\boldsymbol{\delta\Theta}^* = \mathbf{A}^T\mathbf{b} \tag{2.52}$$

where $\mathbf{a}$ and $\mathbf{r}$ have been collected into $\mathbf{b}$. $\mathbf{A}^T\mathbf{A}$ is commonly called the "Hessian" even though it actually represents a linearized approximation to the system Hessian at the linearization points. It can also be understood to be the Fisher Information Matrix of the system. Importantly, the matrix $\mathbf{A}$ is always very sparse and its structure is described by a factor graph.[31] An example of the $\mathbf{A}$ matrix and the equivalent $\mathbf{R}$ matrix for an asteroid relative SLAM problem is shown in Figure 2.1.



(a) **A** matrix of batch solution to an asteroid SLAM problem. The upper left block contains the state transition Jacobians related to the vehicle dynamics and the lower half contains the measurement Jacobians relating the vehicle (left) and landmark (right) states.



(b) Corresponding **R** matrix to 2.1a without any re-ordering. The measurement rows have been eliminated.

Figure 2.1: **A** and **R** of a batch SLAM solution.

The large least squares system in Equation 2.52 can either be solved via Cholesky factorization of $(\mathbf{A}^T\mathbf{A})$ or by using an equivalent square root found via QR factorization of **A**,

$$\mathbf{A} = \mathbf{Q}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix} \tag{2.53}$$

where **Q** is an orthonormal matrix and **R** is an upper-triangular square root information matrix. The transformed **b** vector can then be computed via

$$\mathbf{Q}^T\mathbf{b} = \begin{bmatrix}\mathbf{d}\\ \mathbf{e}\end{bmatrix} \tag{2.54}$$

and the system solved via back-substitution

$$\boldsymbol{\delta\Theta}^* = \mathbf{R}^{-1}\mathbf{d} \tag{2.55}$$

It turns out that this **R** matrix, which represents the square root of the posterior distribution over all states, as well as the vector term **d** can be efficiently updated in an incremental fashion as in Incremental Smoothing and Mapping (iSAM/iSAM2)[67] using Givens (no relation) rotations. This

QR factorization can be interpreted through the lens of graphical models as a *variable elimination* routine, converting the *factor graph* describing the structure of $\mathbf{A}$ to a *Bayes network* in $\mathbf{R}$ by eliminating the measurement rows, as seen in Figure 2.1b.

The sparsity of the $\mathbf{R}$ matrix is heavily influenced by the state ordering used in the QR factorization and thus both square root smoothing and mapping ($\sqrt{\text{SAM}}$)[30] and its progeny iSAM2 utilize a heuristic algorithm called *colamd* to obtain a good variable ordering before computing the factorization. iSAM2 can operate with constant time complexity until a loop closure is encountered. Once a loop closure is processed and the state ordering updated with *colamd*, it returns to constant time complexity. iSAM2 also introduces "fluid relinearization" to relinearize parts of the trajectory without recomputing all of the Jacobians in $\mathbf{A}$. Given an *analytical* form of the STM, Equation 2.48 can be iterated efficiently to convergence.

Since iSAM2 was developed, the capabilities of batch SLAM algorithms have been expanded in many ways. Much of the field has now embraced concepts from Lie theory to better represent the dynamics and kinematics of rigid body rotations and transformations.[109] This has paralleled developments that enable the efficient use of high-rate inertial measurement units using "pre-integration" factors[45] which can otherwise be problematic for batch methods. Pre-integration methods combine many IMU measurements over a given time interval into a single analytic factor that does not depend on the linearization point at the start of the interval, removing the need to re-integrate the equations of motion when the initial linearization point is altered by the optimization.

Another development has been that of "trajectory estimation" instead of "localization", which is to recast the previous mathematics in terms of continuous-time dynamics, represented with some style of basis functions such as Gaussian processes, to produce a smooth trajectory that can be queried at any time of interest and better enables the processing of asynchronous measurements.[4]

2.4 Tradeoffs

In the interest of obtaining the most computationally efficient SLAM algorithm possible, one that would be suitable for an implementation onboard a resource-constrained spacecraft system, it is worth reflecting on the tradeoffs associated with the EKF and Batch approaches. Unlike many robotics problems, even the simplest spacecraft dynamics have a transcendental relationship with time and *do not* yield an analytical form of the STM unless approximations are made which degrade significantly as the time interval over which the STM is calculated increases. For a batch algorithm, this would necessitate re-integration of the trajectory in order to obtain new STMs to be added to the Hessian to enable iteration of the solution to convergence. This precludes the constant time complexity of algorithms like iSAM2, which then have linear time complexity with the number of landmarks.[4] That said, if a good initial linearization is assumed, the initial forward solution without iteration would likely still be acceptable. An alternative might be to break up the large time interval with small intervals with valid STM approximations, but this may present challenges similar to those of inertial measurement units. A spacecraft dynamics "pre-integration" factor, which recasts the factor nodes in a way that is agnostic to their initial conditions so that they can be efficiently processed, may be an interesting topic of future research but this is beyond the scope of the present work.

While Reference [115] famously asked "Why Filter?" and answered by arguing that batch approaches are typically superior in terms of accuracy and computational cost, a caveat that "filtering may have a niche in systems with low processing resources" was given. An additional comment by the same authors in Reference [116] concerned the existence of the relatively underexplored "middle-ground" between EKF and batch approaches, where a filtering based solution could conceivably be constructed with a more batch-like structure in order to leverage the benefits of both paradigms. One example of this is the sliding-window approach of Reference **??** where the global MAP solution can be approximated by only considering a sliding window of the most recent states. In fact, the EKF can be seen as a simplification of this scheme where only the most recent state is maintained. Another, more recent example of a hybrid approach is RISE-SLAM[68] which operates as a sequential filter but constructs a square root information matrix that is used for global bundle adjustment and shows performance on-par with state-of-the-art batch solvers.

The main advantage of an EKF-like approach is that it is inherently "forward-looking" due to its style of linearization, never having to revisit older states and observations. If constructed carefully, the solution generated at the final time should also be slightly more accurate than a single iteration of the LKF or batch approaches.[119] With these observations in mind, and with the recent successes of hybrid algorithms that inhabit this middle-ground such as RISE-SLAM and others, we believe that further research into computationally efficient filtering-based SLAM solutions remains motivated, especially for spaceflight applications.

Chapter 3

Sensors and Models

This chapter outlines the development of the measurement and dynamical models that are used in the other chapters of this thesis. While many types of sensors exist and can be adapted for use in SLAM, the focus of this work has been on combining inertial and visual information exclusively, the former being provided by an inertial measurement unit (IMU) and the latter by a single camera periodically taking optical images. This choice was motivated by considering the very few possible information sources available to a given autonomous spacecraft at any time. While it is true that in proximity scenarios that laser-based sensors such as rangefinders and LIDAR may prove very useful, the power and size requirements of such sensors grow considerably as the relevant relative distance to a target increases. That said, if these sensors are available in addition to the IMU and camera sensors assumed here, their information can be incorporated into the estimation process to improve the results. This is beyond the scope of this work.

One exception we make is the assumption that an attitude determination and control system provides a reliable inertial attitude solution that can either be used as a strong prior or as an input into the SLAM estimator. Attitude Determination and Control Systems (ADCS) based on the tight integration of low-drift gyroscopes and star tracker sensors are mature and compact, generating very precise estimates of a spacecraft's attitude at high temporal rates. This fact has been noted in aerospace research before and enabled new geometric techniques that decouple the attitude from the full 6 degree-of-freedom problem.[71]

3.1 Spacecraft Dynamics

Orbital mechanics is filled with elegant mathematical tools to aid in the analysis and comprehension of complex orbital motion. For the purposes of precision navigation, where a littany of perturbing forces act on the spacecraft at all times, tools such as classical orbital elements and the circular restricted three-body problem (CR3BP) tend to be less appropriate due to the strong assumptions they make about these perturbations. Additionally, casting some measurement types into these parameters can be challenging. In Earth orbits, relative orbital element approaches have been demonstrated to be effective in certain problems such as angles-only navigation.[117]

Instead, here we utilize numerical integration of the Cartesian equations of motion. For a system with a large, central body in an inertial reference frame I, the relative point-mass gravitational acceleration acting on a second body of negligible mass is

$$ {}^{I}\mathbf{f}_{kep} = -\mu \frac{{}^{I}\mathbf{p}}{p^3} \tag{3.1} $$

where $\mu = GM$ is the gravitational parameter of the first (central) body and $\mathbf{p}$ is the position of the second body with respect to the first. Other force models we consider in this thesis are the 3rd body gravitational perturbation,

$$ {}^{I}\mathbf{f}_{3B} = \mu_3 \left(\frac{{}^{I}\mathbf{r}}{r^3} - \frac{{}^{I}\mathbf{R}}{R^3} \right) \tag{3.2} $$

where ${}^{I}\mathbf{R}$ is the position of the perturbing body with mass parameter μ_3 with respect to the central body and $\mathbf{r} = \mathbf{R} - \mathbf{p}$. The acceleration imparted on a spacecraft by solar radiation pressure (SRP) can be captured approximately by the "Cannonball model",

$$ {}^{I}\mathbf{f}_{SRP} = -\eta S \frac{{}^{I}\mathbf{r}_s}{r_s^3} \tag{3.3} $$

where ${}^{I}\mathbf{r}_s$ is the position of the sun with respect to the spacecraft. The factor S contains terms assumed known *a priori* encoding the Sun's radiation output and the shape of the spacecraft

$$ S = \frac{\gamma R_{\odot}^2 T_s^4}{c} C_R \frac{A}{m} \tag{3.4} $$

where γ is the Stefan-Boltzmann constant, $R_{\odot}$ is the radius of the Sun, T_s is the surface temperature of the Sun, c is the speed of light, C_R is the coefficient of reflectivity of the spacecraft, and A/m is the spacecraft area-to-mass ratio. The parameter η is a scale factor to be estimated instead of any of the linearly-dependent terms in S. These and other spacecraft acceleration models are described in Reference [83].

With these dynamical models, the equations of motion can be formulated for central-body relative position and velocity using Cowell's method,

$$^{I}\dot{\mathbf{p}} = {}^{I}\mathbf{v} \tag{3.5}$$

$$^{I}\dot{\mathbf{v}} = {}^{I}\mathbf{f}_{kep} + {}^{I}\mathbf{f}_{SRP} + \sum_{i} {}^{I}\mathbf{f}_{i,3B} \tag{3.6}$$

where we have allowed for multiple perturbing third bodies using the summation. Other accelerations such as non-spherical gravity and drag could be included in the same way but these have been neglected in the cases considered here due to the vanishing magnitudes of their influences.

For landmark-based sequential SLAM, it is typically desirable to have the landmarks be stationary in whatever frame they are being estimated in such that Equation 2.29 and its computational benefits remain valid. For terrestrial problems, a simple North-East-Down (NED) frame with a constant gravity assumption suffices for this purpose. However, for space problems in general we need to utilize a body-fixed rotating frame M. Assuming that the asteroid or planet is rotating on its principal axis, the previous equations can be recast into such a frame as

$$^{M}\dot{\mathbf{p}} = {}^{M}\mathbf{v} \tag{3.7}$$

$$^{M}\dot{\mathbf{v}} = {}^{M}_{I}\mathbf{C}\,{}^{I}\dot{\mathbf{v}} - [\boldsymbol{\omega}_M \times]^2\mathbf{p} - 2[\boldsymbol{\omega}_M \times]\mathbf{v} \tag{3.8}$$

where the additional Euler and Coriolis terms are apparent. It is worth noting that a non-spherical gravity model may be more easily computed in this rotating frame, so any rotation of that component could be neglected. Indeed, it is a design choice to represent any of the gravitational models in the inertial frame in our problems, one that merely affects the form of the necessary Jacobians.

3.2 Inertial Measurement Units

The term "inertial measurement unit" has, in the 21st century, largely come to denote a small strapdown sensor package that includes two types of sensors: linear accelerometers and rate gyroscopes. The former measures non-gravitational accelerations along a single dimension while the latter measures the instantaneous angular rate about a single axis. Three of each sensor are arranged in aligned orthogonal triads such that the outputs of each sensor can be easily combined and integrated to provide six independent measurements of the sensor's motion in space.[48] Through sconing and culling integrals, the details of which are beyond the scope of this work, these outputs can be turned into measurements of velocity change and attitude change over a given integration period.

This is a modern-day form of classical dead reckoning techniques: starting from a known state, it is possible to predict future states by summing up small observed rates and changes over time. As any mariner knows, the solution will always drift over time as unavoidable errors are accumulated. For IMUs, as with many things in life, the quality and stability of the integrated solution tends to scale with the sensor's price. When fused with other measurement systems such as from the Global Positioning System (GPS), the sensor package is typically referred to as an Inertial Navigation System (INS). An excellent primer on these topics has been provided online by the INS manufacturer VectorNav.[1]

With the advent of micro-electromechanical sensor (MEMS) technology, leading to tiny sensors that can be integrated directly onto circuit boards, IMUs have been used in everything from cell phones to commercial drones to TV remotes, to name a few. The outputs of these small IMUs tend to be very noisy and thus do not provide reliable estimates of motion and orientation for very long unless fused with complimentary sensor information as in an INS. In terrestrial applications, the normal force which opposes gravity can be drawn upon as a stable, reliable downward reference. This is how your phone knows when you turn it sideways and want it to switch the screen to landscape orientation to watch a video.

In space, the only accelerations that even the most sensitive IMU will ever encounter come from transient events such as maneuver thrust forces and drag during atmospheric re-entry. The former case is interesting since contemporary orbit determination practice often only estimates

[1] `https://www.vectornav.com/resources/inertial-navigation-primer`

maneuvers after they have been performed. There's no fundamental reason why an autonomous spacecraft could not use the measured acceleration provided by an onboard IMU as an input to a navigation solution or SLAM estimator. Indeed, in the case of landing on a large celestial body, the integration of an IMU is essential to the success of a mission and has been incorporated into the descent systems of recent Mars landers.[64]

The inclusion of an IMU acceleration measurement in the formulation for space problems is therefore situation-dependent. If image measurements are sparse and no large maneuvers are being executed, there is no reason to include accelerometers and the often-unneccessary uncertainty they introduce into the SLAM solution. However, if large forces are present, such as in cases of landing or otherwise aggressive maneuvering, the inclusion of the accelerometers in the equations of motion is necessary. Similarly, as previously discussed, the inclusion of gyroscope measurements and attitude estimation into a tightly-coupled SLAM solution is not strictly necessary given a high-quality ADCS solution but would be essential in some cases.

3.2.1 IMU Model Replacement

In the situation that a full IMU model is to be included in an EKF formulation, the standard method of incorporating this information is not to treat it as high-rate measurements but as a known, noisy input. This is in contrast to other measurement types and is referred to under the general name "model replacement." Here, we model the noisy IMU accelerometer and gyroscope inputs in the IMU body frame B as

$$^{B}\tilde{\mathbf{f}} = {}^{B}\mathbf{f} + {}^{B}\mathbf{w}_f \tag{3.9}$$

$$^{B}\tilde{\boldsymbol{\omega}} = {}^{B}\boldsymbol{\omega} + {}^{B}\mathbf{w}_\omega \tag{3.10}$$

where $\mathbf{f}$ and $\boldsymbol{\omega}$ are the "clean" acceleration and angular rate signals corrupted by additive, white Gaussian noise. More sophisticated signal models are commonly used in practice, with biases, misalignments, and scale factor errors explicitly modeled and estimated. Also, coning and sculling can be neglected if we assume that the signals are being integrated at high rates.

When incorporating quaternion kinematics, as will be done in this section, it is common to formulate the system using explicitly-defined error state dynamics.[123, 108] For most states of interest, such as position and velocity, the error states are defined in a simple, additive way,

$$\mathbf{x} = \hat{\mathbf{x}} + \delta\mathbf{x} \tag{3.11}$$

where $\mathbf{x}$ is the "true" state and $\hat{\mathbf{x}}$ is the "estimated" state that differs from the true state by the error.

Since attitude is to be estimated, an attitude parameterization must be chosen. All three-parameter attitude sets are known to have singularities that must be avoided during operation. This is typified by the infamous "gimbal lock" associated with Euler angles. Some aerospace practitioners, usually ones somehow associated with the University of Colorado Boulder, prefer "Modified Roderigues Parameters" (MRPs) as an elegant three-parameter choice that requires switching to a "shadow set" to avoid its singularity.[100] However, the choice of most GNC engineers and roboticists at the present time is to utilize unit quaternions, also known as Euler parameters, which offer a 4×1 parameter set obeying a unit norm constraint that has no singularities. While this could be an ideal tradeoff, a single extra state for no singularities, the details of quaternion kinematics require more complexity in the error state definition.

To make things even more difficult, multiple conventions for quaternions exist that, if mixed and not reconciled, can lead to erroneous results. Following the nomenclature in Reference [108], we choose here to adopt the "JPL" or "Breckenridge" convention for the quaternion which is further developed in Reference [123]. Namely, we define a unit quaternion as

$${}^{B}_{I}q = \begin{bmatrix} q_x \\ q_y \\ q_z \\ q_w \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ q_w \end{bmatrix} \tag{3.12}$$

where the fourth element is the "scalar" part and the rotation that is represented is from the I frame to the B frame. This is in contrast to the "Hamiltonian" convention that is more common in robotics and, ironically, in certain departments at JPL. The Hamiltonian convention, which Reference [108] prefers, treats the first element as the scalar part and represents rotations from the B frame to the I frame. These references provide all of the necessary definitions for quaternion conversions and multiplications, so these details will not be provided here.

Following the recommendations of authors in the visual-inertial navigation literature,[58] and in contrast to Reference [123], we choose here to define the multiplicative quaternion error *with respect to the fixed frame of reference* I,

$$ {}^{B}_{I}q = {}^{B}_{\hat{I}}q \otimes {}^{\hat{I}}_{I}\delta q = {}^{B}_{\hat{I}}q \otimes \begin{bmatrix} \hat{\mathbf{k}} \sin(\delta\psi/2) \\ \cos(\delta\psi/2) \end{bmatrix} \tag{3.13} $$

where $\delta\psi$ is a rotation about the Euler axis $\hat{\mathbf{k}}$. For small $\delta\boldsymbol{\psi}$,

$$ {}^{\hat{I}}_{I}\delta q \approx \begin{bmatrix} \delta\boldsymbol{\psi}/2 \\ 1 \end{bmatrix} \tag{3.14} $$

where $\delta\boldsymbol{\psi} = \delta\psi\hat{\mathbf{k}}$. Given that the quaternion to DCM conversion is

$$ {}^{B}_{I}\mathbf{C} = (2q_w^2 - 1)\mathbf{I}_3 - 2q_w[\mathbf{q}\times] + 2\mathbf{q}\mathbf{q}^T \tag{3.15} $$

where $[\mathbf{q}\times]$ is the skew-symmetric cross product matrix of $\mathbf{q}$, the following useful small-angle relationship can be derived by using the preceding error definition and ignoring second order terms:

$$ {}^{I}_{\hat{I}}\mathbf{C} \approx \mathbf{I}_3 - [\delta\boldsymbol{\psi}\times] = (\mathbf{I}_3 + [\delta\boldsymbol{\psi}\times])^T \tag{3.16} $$

This is utilized extensively in the measurement model derivatives in Appendix A.

With the error states now defined, we proceed by writing the “true” dynamics of the quaternion as

$$ {}^{B}_{I}\dot{q} = \frac{1}{2} \begin{bmatrix} {}^{B}\boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes {}^{B}_{I}q \tag{3.17} $$

and the position and velocity dynamics in the rotating M frame as

$$ {}^{M}\dot{\mathbf{p}} = {}^{M}\boldsymbol{v} \tag{3.18} $$

$$ {}^{M}\dot{\mathbf{v}} = {}^{M}\boldsymbol{f} + {}^{M}_{B}\mathbf{C}^{B}\mathbf{f} + {}^{M}_{I}\mathbf{C}^{I}\mathbf{w}_{v} \tag{3.19} $$

where the dynamics on the right-hand-size of Equation 3.8 have been collected into ${}^{M}\boldsymbol{f}$ and an acceleration process noise term $\mathbf{w}_v \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_v)$ has been added in the inertial frame. The relationship between the rotating frame and inertial frame is assumed to be perfectly known. The expected value of these dynamics yields the navigation propagation equations

$$ {}^{B}_{I}\dot{\hat{q}} = \frac{1}{2}\begin{bmatrix} {}^{B}\tilde{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \otimes {}^{B}_{I}\hat{q} \tag{3.20} $$

$$ {}^{M}\dot{\hat{\mathbf{p}}} = {}^{M}\hat{\boldsymbol{v}} \tag{3.21} $$

$$ {}^{M}\dot{\hat{\mathbf{v}}} = {}^{M}\hat{\boldsymbol{f}} + {}^{M}_{\hat{B}}\mathbf{C}^{B}\tilde{\mathbf{f}} \tag{3.22} $$

Equations 3.11 and 3.13 can be differentiated and Equations 3.17-3.22 inserted to yield expressions for the error state model equations. After extensive simplification and by ignoring second-order terms, the final linear equations can be found:

$$ \delta\dot{\psi} = -{}^{\hat{I}}_{B}\mathbf{C}\mathbf{w}_{\omega} \tag{3.23} $$

$$ {}^{M}\delta\dot{\mathbf{p}} = {}^{M}\hat{\boldsymbol{v}} \tag{3.24} $$

$$ {}^{M}\delta\dot{\mathbf{v}} = -{}^{M}_{\hat{B}}\mathbf{C}[{}^{B}\tilde{\mathbf{f}}\times]\delta\boldsymbol{\psi} + \left(\frac{\partial \boldsymbol{f}}{\partial {}^{M}\mathbf{p}} - [\boldsymbol{\omega}_M\times]^2\right){}^{M}\delta\mathbf{p} - 2[\boldsymbol{\omega}_M\times]^{M}\delta\mathbf{v} - {}^{M}_{\hat{B}}\mathbf{C}\mathbf{w}_f - {}^{M}_{I}\mathbf{C}\mathbf{w}_v \tag{3.25} $$

where a Taylor series expansion was used to approximate the difference

$$ {}^{M}\boldsymbol{f} - {}^{M}\hat{\boldsymbol{f}} \approx \frac{\partial \boldsymbol{f}}{\partial {}^{M}\mathbf{p}}{}^{M}\delta\mathbf{p} \tag{3.26} $$

and where the dynamics Jacobian follows

$$ \frac{\partial \boldsymbol{f}}{\partial {}^{M}\mathbf{p}} = \frac{\mu}{p^3}\left(\frac{3\mathbf{p}\mathbf{p}^T}{p^2} - \mathbf{I}_3\right) + \frac{\mu_{3B}}{r^3}\left(\frac{3\mathbf{r}\mathbf{r}^T}{r^2} - \mathbf{I}_3\right) + \frac{\eta S}{r_s^3}\left(\mathbf{I}_3 - \frac{3\mathbf{r}_s\mathbf{r}_s^T}{r_s^3}\right) \tag{3.27} $$

The error state system in state space form is

$$ \delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x} + \mathbf{G}\mathbf{w} \tag{3.28} $$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ -{}^{M}_{\hat{B}}\mathbf{C}[{}^{B}\tilde{\mathbf{f}}\times] & \frac{\partial \boldsymbol{f}}{\partial {}^{M}\mathbf{p}} - [\boldsymbol{\omega}_M\times]^2 & -2[\boldsymbol{\omega}_M\times] \end{bmatrix} \tag{3.29}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & -{}^{\hat{I}}_{B}\mathbf{C} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -{}^{M}_{I}\mathbf{C} & -{}^{M}_{\hat{B}}\mathbf{C} & \mathbf{0} \end{bmatrix} \tag{3.30}$$

which assumes the noise vector is defined as

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_v \\ \mathbf{w}_f \\ \mathbf{w}_\omega \end{bmatrix} \tag{3.31}$$

When no accelerometer input is expected, it suffices to simply set any terms involving $\tilde{\mathbf{f}}$ and $\mathbf{w}_f$ to zero and remove the relevant rows and columns from the matrices.

3.3 Visual Measurement Models

This section gives an outline of our approach to incorporating visual information in a sequential estimation framework. We present the standard pinhole camera model and then describe the various extant inverse depth measurement models in the literature. Finally, we introduce a new inverse depth model, the Anchored Homogenous Bundles (AHB) model, and discuss some practical considerations.

3.3.1 Pinhole Camera Model

The standard model of a conventional camera image is the pinhole model, a simplification of the true physical process where light rays from a scene are focused at a single point and sensed by either a photosensitive film or, more commonly in the 21st century, by a digital camera sensor. This allows for a two-dimensional projection of the three-dimensional illuminated scene to be encoded geometrically and subsequently processed. More details on the simplifications and justifications of the pinhole model are provided in Reference [19].

A raw camera image will never, in general, preserve the 3D parallel lines of an observed scene due to distortions introduced by the camera lens, severely attenuating the usefulness of a naive pinhole model. However, well-known camera calibration methods can be used to compensate for these distortions and produce images that follow the pinhole perspective projection very closely.[132] In the case of spaceflight, any starfield provides an excellent reference that can be used to determine these parameters very precisely.[19] Given this, we proceed by only considering the pinhole camera model in the subsequent derivations.

When a distinct point $\mathbf{l}_i$ in a world-fixed frame M is observed by the camera center $\mathbf{p}$, which we assume is coincident with the spacecraft center of navigation until Section 3.3.7, the ray from the camera center $\mathbf{p}$ to the image feature is given by

$$^{C}\mathbf{h}_i = {}^{C}_{M}\mathbf{C}({}^{M}\mathbf{l}_i - {}^{M}\mathbf{p}) = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \tag{3.32}$$

where ${}^{C}_{M}\mathbf{C}$ is a direction cosine matrix mapping vectors from M to C. The scale of this vector is ambiguous, so its projection on the image plane $z = 1$ is

$$^{C}\mathbf{u}_i = \frac{^{C}\mathbf{h}_i}{^{C}h_{i,z}} = \begin{bmatrix} x_i/z_i \\ y_i/z_i \\ 1 \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} \tag{3.33}$$

which can be scaled to image coordinates to model via the camera intrinsics matrix $\mathbf{K}$ to yield the measurement model

$$^{C}\mathbf{z}_i = \mathbf{K}\,{}^{C}\mathbf{u}_i + \boldsymbol{v}_i \tag{3.34}$$

with

$$\mathbf{K} = \begin{bmatrix} f/d_x & 0 & c_x \\ 0 & f/d_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.35}$$

where the focal length f and pixel pitch d are in world units and the center of the image is located at (c_x, c_y). Zero mean, white Gaussian noise $\boldsymbol{v}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{V})$ is added and can be specified in units of pixels.

3.3.2 Depth Estimation

Equation 3.32 represents the simplest and most intuitive model of the vector between the camera and a 3D visual feature. Given at least two spatially separated images with a known baseline and which each yield a measurement to the same feature, the 3D location of the feature can be calculated in a process known as *triangulation.* An extensive treatment of the triangulation, and the closely related problem of *resection*, is given in Reference [55]. However, if Gaussian errors are assumed in the individual measurements in two images, as is commonly done, and hence the difference or *disparity* between the measurements is also Gaussian, the probability distribution of the triangulated point is not well characterized by a Gaussian spatial distribution,[54] particularly at low parallax.[84] The reason why this is the case can be seen by examining the so-called stereo projection function[105]

$$d = f\frac{b}{x_2} \tag{3.36}$$

where the disparity d is represented as a function of the focal length, the baseline $b = x_1 - x_0$, and the depth of the given point x_2. Clearly, the disparity is nonlinearly related to the depth and this effect is strongly tied to the parallax ratio b/x_2.

Repeating an experiment from Reference [105], 100,000 random samples can be drawn from a Gaussian distribution over disparity with a mean of 1 pixel and a standard deviation of 0.3 pixels. As shown in Figure 3.1, the true depth of the triangulated point is at 51.08 meters with a baseline between viewpoints of 1 meter. The result is the classic “heavy tail” distribution with a biased mean value of 58.1026 meters.

The bias present in these calculations makes visual feature initialization problematic in a sequential VSLAM architecture. If a feature is initialized immediately, its non-Gaussian behavior will violate the assumptions that are typically baked into Kalman Filter algorithms. Given mea-

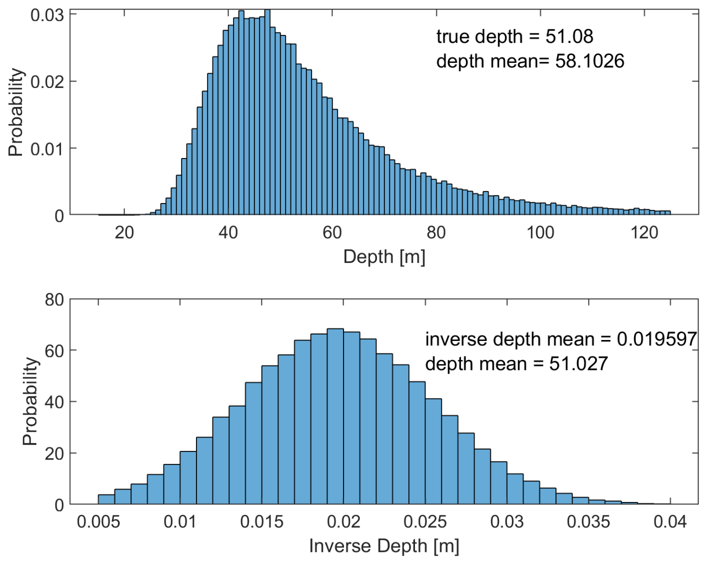


Figure 3.1: Comparison of depth and inverse depth distributions given Gaussian errors in disparity.

surements over more than two frames, a visual feature could be more-reliably triangulated and then inserted into the state vector, a process known as *delayed initialization.* While this yields better feature estimates, it also throws away valuable correlation information that could have been used to improve the other state estimates while the feature was being initialized. A better delayed approach is the one taken by the Multi-State Constraint Kalman Filter (MSCKF) and variants[72], where features are tracked without being inserted into the state vector at all but their correlation information is retained and ultimately applied to the state via a nullspace projection scheme.

3.3.3 Inverse Depth Model

Returning to Equation 3.32, we note that the disparity is linearly related to the inverse of the depth,

$$d = fbx_2^{-1} \tag{3.37}$$

which implies that the *inverse depth* probability distribution will be Gaussian if the disparity is Gaussian. Using the same 100,000 samples as in the previous section, we see exactly that in the lower half of Figure 3.1. The inverse depth distribution is Gaussian and has the same kurtosis as the initial distribution. Taking the mean of this distribution and inverting it yields an unbiased estimate for the depth of the visual feature.

The linearity of the inverse of the depth has motivated researchers to develop efficient methods to use the inverse depth to parameterize visual features in sequential filtering schemes. Long before SLAM was a hot topic of research, the use of the "reciprocal of the range" was noted in the Target Motion Analysis literature.[2] Later, for SLAM, Montiel et al. [84] developed the "Unified Inverse Depth Model" and showed how to mechanize it within an EKF-SLAM architecture. This works by introducing a copy of the camera position when a visual feature is first identified (the "anchor") and estimating the inverse depth along a ray in space from that position to the feature, parameterized with a pair of world-referenced spherical angles. Mathematically, Equation 3.32 is rewritten as

$$^{C}\mathbf{h}_i = {}^{C}_{M}\mathbf{C}[\rho_i({}^{M}\mathbf{p}_j - {}^{M}\mathbf{p}) + {}^{M}\mathbf{m}(\phi_i, \theta_i)] \tag{3.38}$$

where ρ_i is the inverse of the depth, $\mathbf{p}_j$ is the camera anchor position at the time the feature is first observed, and (θ, ϕ) are the spherical angles that are used to form a direction vector

$$ {}^{M}\mathbf{m}(\phi_i, \theta_i) = \begin{bmatrix} \cos\phi_i \sin\theta_i \\ -\sin\phi_i \\ \cos\phi_i \cos\phi_i \end{bmatrix} \tag{3.39} $$

At the cost of over-parameterizing the state of each feature, this model results in vastly improved estimation performance for features at all positive depths including infinity ($\rho = 0$). Further, if no prior information about a given feature's depth is known, it can safely be initialized at infinite depth and will not contribute to the position estimates of the robot until multiple observations are processed and the inverse depth is constrained by geometry. The authors also offered a useful linearity index in order to determine when individual features were behaving linearly enough to be converted back to a standard 3×1 representation via

$$ {}^{M}\mathbf{l}_i = {}^{M}\mathbf{p}_j + \frac{1}{\rho_i} {}^{M}\mathbf{m}(\phi_i, \theta_i) \tag{3.40} $$

and its Jacobians. This and many more details are usefully provided in Reference [22]

As pointed out by Imre et al.[61], the original inverse depth model does not enforce a "common origin" constraint because each feature is assigned its own anchor position. If multiple features are initialized at once, they should share a common anchor position. Doing so increases the consistency of the algorithm.[61]

3.3.4 Anchored Homogenous Points

The original inverse depth model was adopted by many researchers in the succeeding years and alternative formulations were explored with the goal of improving its consistency and computational characteristics. Notably, Solà [106] describes two alternatives to the original: Homogenous Points (HP) and Anchored Homogenous Points (AHP). Homogenous points in the general sense are well-known in computer vision and, in the case that the camera position is assumed to be known very-precisely, Solà's HP parameterization allows for the camera anchor position for each feature to be dropped from the state vector entirely. While increasing computational efficiency, this assumption decreases the consistency and accuracy of the estimator.

Alternatively, the AHP formulation is the same as the original inverse depth model except that the direction vector is parameterized in Cartesian coordinates,

$$ {}^{C}\mathbf{h}_i = {}^{C}_{M}\mathbf{C}[\rho_i({}^{M}\mathbf{p}_j - {}^{M}\mathbf{p}) + {}^{M}\mathbf{m}_i] \tag{3.41} $$

where $\mathbf{m}_i$ must be a unit vector if ρ_i is the true inverse depth. If $\mathbf{m}_i$ is not a unit vector, ρ_i is still a valid inverse depth but it is scaled linearly by some constant related to $\|\mathbf{m}_i\|$. The drawback of using a Cartesian representation is that it requires another state in the state vector, bringing the total states per feature to 7×1 or $3n_j + 4n_i$ when enforcing the common origin constraint. The benefit is an increase in consistency of the solution.[106]

3.3.5 Inverse Depth Bundles

The contribution of Pietzsch[92] was to re-coordinatize the feature direction vectors in their respective anchor camera frames instead of the fixed frame. This requires the estimator to retain and estimate anchor *poses*, containing both the camera position and attitude, instead of just anchor positions. With the additional rotation,

$$ {}^{C}\mathbf{h}_i = {}^{C}_{M}\mathbf{C}[\rho_i({}^{M}\mathbf{p}_j - {}^{M}\mathbf{p}) + {}^{M}_{C_j}\mathbf{C}^{C_j}\mathbf{m}_i] \tag{3.42} $$

This is described as a *view-based* model and has been leveraged in other successful visual SLAM algorithms since its inception.[13] Additionally, Pietzsch proposes the inverse depth bundles (IDB) model which makes the assumption that the initial measurement to each feature can be seen as having no error and therefore its direction vector in the original camera frame can be treated as deterministic. This assumption allows for the direction vector to be saved and not estimated, resulting in significant computational advantage due to the reduced state size. An analysis of the costs and benefits of this assumption is provided in the dissertation in Reference [93]. However, while this assumption may be justified using the warped patch-feature representation used in that work, our experience has shown that it results in accuracy and consistency losses when using standard feature descriptors in spacecraft problems.

Another benefit of the view-based model, one that has not been noted in the literature, is that the individual feature states are not correlated to the rest of the state vector until the *second* measurement to each feature is processed. This has three benefits: unlike the Unified Inverse Depth model and the AHP model, no Jacobians need be computed when a feature is initialized in the state. The states and their covariances can simply be inserted into the state vector. Second, this effectively delays any decisions about whether or not a feature should be tracked by one image *without any loss of information.* Finally, because features are not correlated to the rest of the state until the second measurement, a good inverse depth prior can be triangulated using the first two measurements before the second measurement is processed by the filter without double-counting information.

3.3.6 Anchored Homogenous Bundles

If high-precision estimates of visual feature locations are desired, the deterministic $\mathbf{m}$ assumption of the IDB model is not justified. Instead, we take advantage of the view-based representation but model the direction of the visual feature in the initial frame as a normalized homogenous vector,

$$^{C_j}\mathbf{m}_i = \frac{^{C_j}\mathbf{s}_i}{s_i} \tag{3.43}$$

where

$$\boldsymbol{s}_i = \begin{bmatrix} s_{i,x} \\ s_{i,y} \\ 1 \end{bmatrix} \tag{3.44}$$

and $s_i = \|\mathbf{s}_i\|$. Since $\mathbf{m}_i$ is a 3×1 unit vector with only rank 2, the uncertainty is ill-defined in the camera frame. However, these two independent quantities are precisely the first two elements of $\mathbf{s}_i$, so by factoring out its norm we obtain

$$s_i{}^{C}\mathbf{h}_i = {}^{C}_{M}\mathbf{C}[s_i\rho_i({}^{M}\mathbf{p}_j - {}^{M}\mathbf{p}) + {}^{M}_{C_j}\mathbf{C}^{C_j}\mathbf{s}_i] \tag{3.45}$$

Returning to Equation 3.33, we can see that the norm actually cancels out of the projection relationship:

$$^{C}\mathbf{u}_i = \frac{s_i {}^{C}\mathbf{h}_i}{s_i {}^{C}h_{i,z}} \tag{3.46}$$

This introduces two new states into the state vector in addition to the inverse depth and the camera pose so that a bundle of n features initializd with pose $(\mathbf{p}_j, {}^{M}_{C_j}q)$ takes the form

$$\mathbf{x}_j = \begin{bmatrix} \rho_1 \\ s_{1,x} \\ s_{1,y} \\ \vdots \\ \rho_n \\ s_{n,x} \\ s_{n,y} \\ {}^{M}_{C_j}q \\ \mathbf{p}_j \end{bmatrix} \tag{3.47}$$

While it costs more computationally than IDB, the uncertainty of the initial direction is treated properly with the addition of the two components of $\mathbf{s}_i$. Recovery of an individual landmark estimate must incorporate the estimated s_i factor to scale ρ_i,

$$^{M}\mathbf{l}_i = {}^{M}\mathbf{p}_j + {}^{M}_{C_j}\mathbf{C}\frac{^{C_j}\mathbf{s}_i}{s_i \rho_i} \tag{3.48}$$

The geometry of the AHB representation is shown in Figure 3.2.

As a view-based representation, the useful initialization properties of the IDB model are retained with AHB and the uncertainty of the homogenous direction vector itself is very straightforward to obtain from the initial measurement. At initialization, the Equation 3.45 simplifies to

$$^{C}\mathbf{h}_i = \frac{^{C}\mathbf{s}_i}{s_i} \tag{3.49}$$

and, given Equation 3.46, the first measurement can be modeled as

$$^{C}\mathbf{z}_i = \mathbf{K}{}^{C}\mathbf{s}_i + \boldsymbol{v}_i \tag{3.50}$$

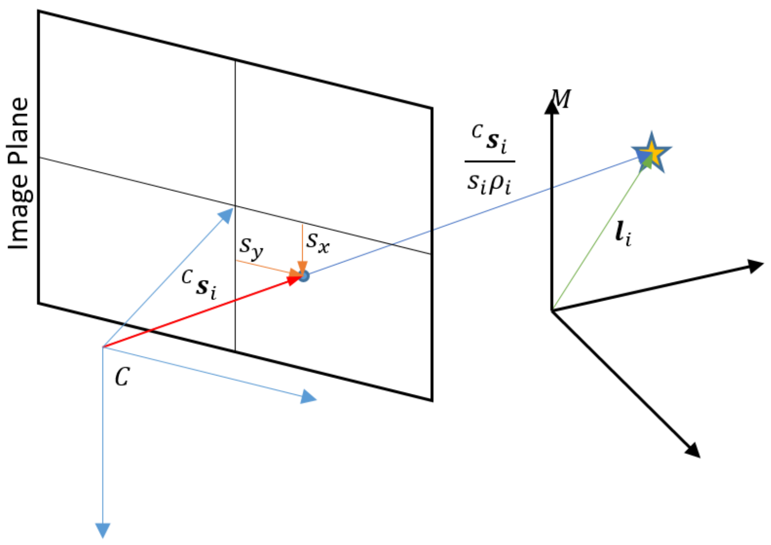


Figure 3.2: Geometry of Anchored Homogeneous Bundles model.

The mean and covariance of the $\mathbf{s}_i$ vector, assuming a realization of $\mathbf{z}_i$ is given at initialization, are then

$$\mathbf{E}[\mathbf{s}_i] = \mathbf{E}[\mathbf{K}^{-1}(\mathbf{z}_i - \boldsymbol{v}_i)] = \mathbf{K}^{-1}\mathbf{z}_i \tag{3.51}$$

$$\mathbf{E}[(\mathbf{s}_i - \mathbf{E}[\mathbf{s}_i])(\mathbf{s}_i - \mathbf{E}[\mathbf{s}_i])^T] = \mathbf{K}^{-1}\mathbf{V}_i\mathbf{K}^{-T} \tag{3.52}$$

3.3.7 Incorporation of Camera Alignment and Offset

Often, the camera sensor will not be mounted at the center of navigation (CON) and will also not be aligned with the body frame of the robot or spacecraft. The previous models can be straightforwardly modified to incorporate the presence of a separate body frame by defining

$${}^{M}\mathbf{p}_{j,c} = {}^{M}\mathbf{p}_j + {}^{M}_{B_j}\mathbf{C}^{B_j}\mathbf{d}_c \tag{3.53}$$

$${}^{M}\mathbf{p}_c = {}^{M}\mathbf{p} + {}^{M}_{B}\mathbf{C}^{B}\mathbf{d}_c \tag{3.54}$$

for both the anchor and current poses and inserting these relationships into the measurement model. For AHB,

$$s_i{}^{C}\mathbf{h}_i = {}^{C}_{M}\mathbf{C}[s_i\rho({}^{M}\mathbf{p}_{j,c} - {}^{M}\mathbf{p}_c) + {}^{M}_{C_j}\mathbf{C}^{C_j}\mathbf{s}_i] \tag{3.55}$$

This adds some complexity to the necessary Jacobians depending on what quantities are to be estimated. The geometry of this augmented measurement model is shown in Figure 3.3.

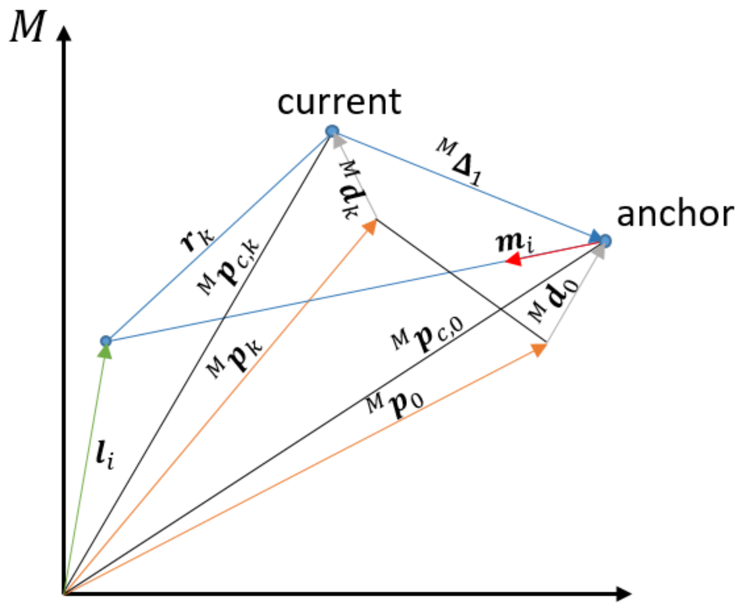


Figure 3.3: Geometry of inverse depth model including lever arms.

Chapter 4

Square Root Extended Information Filter

The EKF algorithm proposed by the authors of the original Unified Inverse Depth[84] retained copies of the pose of the camera for every feature. However, in addition to the increased computational burden caused by maintaining this redundancy in the state, a major problem arises when attempting to invert the resulting covariance matrix. As reported in Reference [56], the resulting covariance matrix is rank-deficient by construction. The problem stems from instantiating copies of the position state in the state vector, which upon initialization results in perfect correlation between the current position and the position copies as well as perfect correlation between the copies themselves. This naturally presents a significant problem to any prospective inverse depth filter operating on the information matrix or its square root, both of which require positive definitness at all times.

Even with this apparent drawback, the benefits of this inverse depth model motivate an attempt to reconcile it with sparse information filters which, due to their targeted handling of computational complexity, address the most significant drawback identified in Reference [22]. That author even mentions sparse information filters by name as a possible avenue of improvement, a challenge that has seemingly gone unanswered until now.

At this point two problems can be solved with the same modification. Following the insights of Reference [44], fill-in of the information matrix can be prevented by properly augmenting the propagated pose while retaining the previous pose instead of marginalizing it out. The new pose is only correlated with the previous pose and not with the prior map, as per the Markov property.

The retention of a pose at a specific time can be made to serve another purpose: to store it for use in the inverse depth parametrization. Multiple features initialized at any time can, in this way, be tied to a single "parent" pose without causing rank deficiency. This methodology elegantly adapts the inverse depth paradigm to the information filter while only adding some extra "bookkeeping", and we believe this represents a novel contribution of this work. Additionally, it allows for a reduction in the total number of states in the state vector as multiple features can be initialized to a single parent pose instead of many.

4.1 SREIF Mechanics

The algorithm presented here is a Square Root Extended Information Filter and will be referred to as the SREIF. Using the observation from Section 2.2.2 that the square root matrix is easy to invert starting from the lower right, we find that it is advantageous to keep the state of the robot or vehicle at the end of the square root matrix and continuously add map states to "permanent addresses" above it. This configuration has the added benefit that the map states do not need to be moved via a potentially-costly copy operation at any point during execution. The SLAM state vector at any time k, which includes all of the previously-tracked "passive" map states $\mathbf{m}_p$ as well as the currently-tracked "active" map states $\mathbf{m}_a$ and the spacecraft states, is

$$\boldsymbol{\mu}_k = \begin{bmatrix} \mathbf{m}_p \\ \mathbf{m}_a \\ \mathbf{x}_k \end{bmatrix} \tag{4.1}$$

where, as the map grows in size, the agent navigation states are constantly moved downward in the stack.

The measurement update is a straightforward application of the standard approaches found in the square root filtering literature.[12, 119] The algorithm makes use of QR-factorizations to avoid explicitly taking square roots after initialization. Due to the need to retain certain poses in the state, both for the inverse depth parametrization and to retain sparsity, the time propagation can be separated into two cases. The first routine is called when the previous measurement update has added features into the state vector, necessitating the retention of the previous pose for the inverse depth model. This process is here labeled “Propagation via Augmentation” because a new pose is augmented to the state vector via the state transition model and follows an alternative square root propagation formulation similar to that found in [129]. Alternatively, if no new features were added to the state at the previous time, the robot state is updated in the standard way, dubbed “Propagation via Marginalization” because the previous state is fully marginalized out of the distribution after propagation. The algorithm is provided in Algorithm 1 for additional clarity and easy reference.

Algorithm 1 VIO Square Root Extended Information Filter

```
Require: $\mathbf{P}_0$,$\boldsymbol{\mu}_0$ (a priori robot estimate)
 1: Initialization:  Compute initial square root using Cholesky factorization $\mathbf{R}_0$,$\mathbf{b}_0 \leftarrow \mathbf{P}_0$,$\boldsymbol{\mu}_0$
 2: NEW_FEATURE $\leftarrow 0$
 3: procedure ((p)erform VIO)
 4:     if NEW_FEATURE then
 5:         Propagate state via augmentation using QR factorization $\mathbf{R}_k^- \leftarrow \mathbf{R}_{k-1}$ (Eq. 4.14)
 6:         NEW_FEATURE $\leftarrow 0$
 7:     else
 8:         Propagate state via marginalization using QR factorization $\mathbf{R}_k^- \leftarrow \mathbf{R}_{k-1}$ (Eq. 4.25)
 9:     Recover mean of robot and local map states $\mathbf{x}_k^-$,$\boldsymbol{\mu}_{l,k}^-$
10:     for Available vision measurements $\mathbf{z}_i$ do
11:         if New feature identified then
12:             NEW_FEATURE $\leftarrow 1$
13:             Initialize feature $\rightarrow \mathbf{R}_{i,0}$,$\boldsymbol{l}_{i,0}$
14:         else
15:             Compute measurement update via QR factorization $\mathbf{R}_k^+$,$\delta\mathbf{b}_k^+$ (Eq. 4.9)
16:         Recover mean of robot and active map states $\mathbf{x}_k^-$,$\boldsymbol{\mu}_{l,k}^-$
```

4.1.1 Measurement Update

Following the notational conventions established in Chapter 2, the data equation for the linearized measurement assumes that the residual of the estimate differs from the true value by the measurement noise,

$$\mathbf{r}_i = \mathbf{H}\delta\boldsymbol{\mu}_k + \boldsymbol{\nu}. \tag{4.2}$$

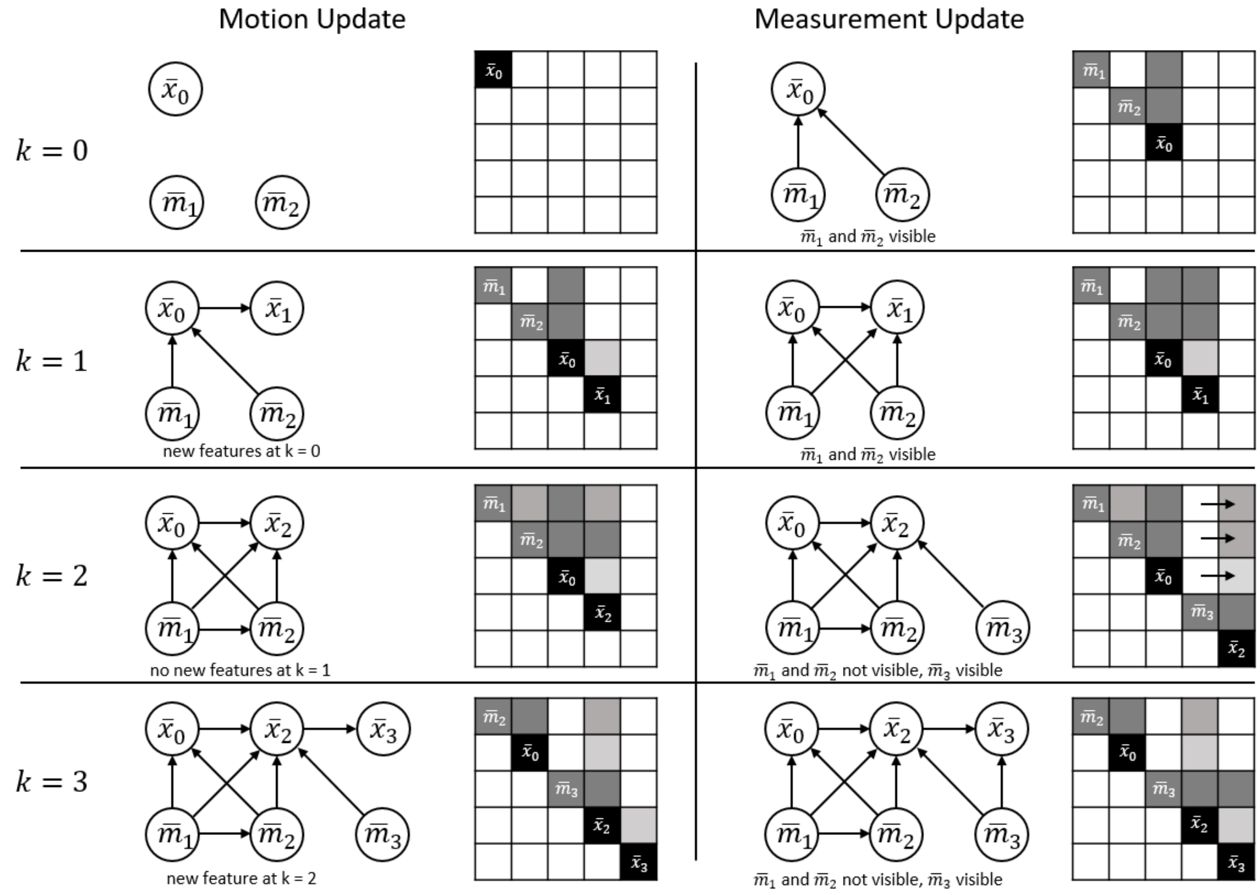


Figure 4.1: Diagram showing the Bayes Network and associated fill-in pattern of the square root information matrix over a discrete time sequence. Initially, at $k = 0$, two landmarks are observed by the vehicle. At time $k = 1$, the same landmarks are observed again, resulting in the relevant off-diagonal terms filling in. At time $k = 2$, propagation via marginalization results in fill-in and then only $\mathbf{m}_3$ is observed and the state is moved down the stack. Finally, since a new feature was added to the matrix, the pose at $k = 2$ is retained at $k = 3$ and the only fill-in comes from another measurement of $\mathbf{m}_3$. $\mathbf{m}_1$ is still retained but not shown in this diagram for brevity.

Utilizing the square root of the inverse of the measurement covariance matrix, Eq. 4.2 is "pre-whitened" as

$$\mathbf{R}_C\mathbf{r}_i = \mathbf{R}_C\mathbf{H}\delta\boldsymbol{\mu}_k + \mathbf{R}_C\boldsymbol{\nu} \tag{4.3}$$

$$\tilde{\mathbf{r}}_i = \tilde{\mathbf{H}}\delta\boldsymbol{\mu}_k + \tilde{\boldsymbol{\nu}} \tag{4.4}$$

where $\tilde{(\cdot)}$ quantities are the pre-whitened ones and the measurement covariance is related to $\mathbf{R}_C$ via

$$\mathbf{V}_i = \mathbf{R}_C^{-1}\mathbf{R}_C^{-T}. \tag{4.5}$$

A cost function can now be written as

$$\mathcal{J}_m(\delta\boldsymbol{\mu}_k) = ||\tilde{\mathbf{H}}\delta\boldsymbol{\mu}_k - \tilde{\mathbf{r}}_i||^2 \tag{4.6}$$

The prior information must be accounted for with its own cost function, taking the form

$$\mathcal{J}_k(\delta\boldsymbol{\mu}_k) = ||\mathbf{R}_k^- \delta\boldsymbol{\mu}_k - \delta\hat{\mathbf{b}}_k^-||^2. \tag{4.7}$$

Summing the cost functions and rewriting the equation yields

$$\mathcal{J}(\delta\boldsymbol{\mu}_k) = \left\| \begin{bmatrix} \mathbf{R}_k^- \\ \tilde{\mathbf{H}} \end{bmatrix} \delta\boldsymbol{\mu}_k - \begin{bmatrix} \delta\hat{\mathbf{b}}_k^- \\ \tilde{\mathbf{r}}_i \end{bmatrix} \right\|^2 \tag{4.8}$$

which can be minimized via QR-factorization so that

$$\begin{bmatrix} \mathbf{R}_k^- & \delta\hat{\mathbf{b}}_k^- \\ \tilde{\mathbf{H}} & \tilde{\mathbf{r}}_i \end{bmatrix} = \boldsymbol{Q} \begin{bmatrix} \mathbf{R}_k^+ & \delta\hat{\mathbf{b}}_k^+ \\ \mathbf{0} & \mathbf{e} \end{bmatrix} \tag{4.9}$$

where $\tilde{\mathbf{r}}_i$ is the whitened residual of the i-th measurement and the matrix $\boldsymbol{Q}$ does not need to be saved. Note that in the extended formulation, $\delta\hat{\mathbf{b}}_k^-$ can be safely set to zero as there is no prior estimate of the deviation from the nominal before the update. By constructing $\tilde{\mathbf{H}}$ in Eq. 4.9 to include all of the available feature measurements at a given time, a single QR-factorization starting from the first active feature in question to the end can be computed quite efficiently. This constitutes a "windowed" update of the full SRI matrix. An added benefit of the very fast and exact mean-recovery capability of the presented algorithm is that an iterated measurement update may also be feasible. This is addressed in the Appendix.

4.1.2 Time Propagation via Augmentation

The development in this section follows a similar approach as in Reference [129] and, interestingly but not surprisingly, strongly parallels the batch formulation described of Section 2.3. Expanding Equation 2.12 to include both the nonlinear and linearized deviation terms from Equation 2.7, we obtain

$$\hat{\boldsymbol{\mu}}(t_k) + \delta\boldsymbol{\mu}(t_k) = \boldsymbol{\phi}(t_k; \hat{\boldsymbol{\mu}}(t_{k-1}), t_{k-1}) + \boldsymbol{\Phi}(t_k, t_{k-1})\delta\boldsymbol{\mu}(t_{k-1}) + \boldsymbol{\Gamma}(t_k, t_{k-1})\mathbf{w}(t_{k-1}) \tag{4.10}$$

Dropping the verbose indices and rearranging results in a cost function that contains both the current and previous state

$$\mathcal{J}_k(\delta\mathbf{x}_{k-1}, \delta\mathbf{x}_k) = \left\| \begin{bmatrix} \boldsymbol{\Phi}_x & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_{k-1} \\ \delta\mathbf{x}_k \end{bmatrix} + (\boldsymbol{\phi}_{x;k,k-1} - \hat{\mathbf{x}}_k) \right\|^2_{\tilde{\mathbf{Q}}_k} \tag{4.11}$$

which is a weighted 2-norm by

$$\tilde{\mathbf{Q}}_k = \boldsymbol{\Gamma}_x \mathbf{Q}_k \boldsymbol{\Gamma}_x^T. \tag{4.12}$$

Additionally, since the augmentation only affects the previous state, the scope is narrowed to including just the robot states $\mathbf{x}$ instead of the entire mean $\boldsymbol{\mu}$ reflected in the "x" subscripts on matrices. Using a prior information cost function of the same form as in the measurement update, representing all of the prior information up to the current time, the total cost function can be composed and rewritten as

$$\mathcal{J}_k^{\ominus}(\delta\mathbf{x}_k, \delta\mathbf{x}_{k-1}) = \left\| \begin{bmatrix} \mathbf{R}_{k-1} & \mathbf{0} \\ -\tilde{\mathbf{Q}}_k^{-1/2}\boldsymbol{\Phi}_x & \tilde{\mathbf{Q}}_k^{-1/2} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_{k-1} \\ \delta\mathbf{x}_k \end{bmatrix} - \begin{bmatrix} \delta\hat{\mathbf{b}}_{k-1} \\ \tilde{\mathbf{Q}}_k^{-1/2}(\boldsymbol{\phi}_{x;k,k-1} - \hat{\mathbf{x}}_k) \end{bmatrix} \right\|^2. \tag{4.13}$$

A QR-factorization can be applied to solve for the optimal values of $\delta\mathbf{x}$,

$$\begin{bmatrix} \mathbf{R}_{k-1} & \mathbf{0} & \delta\hat{\mathbf{b}}_{k-1} \\ -\tilde{\mathbf{Q}}_k^{-1/2}\boldsymbol{\Phi}_x & \tilde{\mathbf{Q}}_k^{-1/2} & \tilde{\mathbf{Q}}_k^{-1/2}(\boldsymbol{\phi}_{x;k,k-1} - \hat{\mathbf{x}}_k) \end{bmatrix} = \boldsymbol{Q} \begin{bmatrix} \mathbf{R}_{k-1}^- & \mathbf{R}_{k-1,k}^- & \delta\hat{\mathbf{b}}_{k-1}^- \\ \mathbf{0} & \mathbf{R}_k^- & \delta\hat{\mathbf{b}}_k^- \end{bmatrix} \tag{4.14}$$

where the only affected portion of the $\mathbf{R}_{k-1}$ matrix is the sub-block corresponding to the new robot states. Since $\hat{\mathbf{x}}_{k-1}$ is computed by numerical integration using the best estimate at t_{k-1}, we can consider it equal to $\boldsymbol{\phi}_{x;k,k-1}$ and set the lower-right term on the right-hand side to be zero. Because of the structure of these matrices and the Markov property, the augmented states are conditionally independent of the entire map, following the same result as for the full information matrix as emphasized in Reference [43].

4.1.3 Time Propagation via Marginalization

We now wish to propagate the robot states forward in time and concurrently marginalize out the preceeding pose without requiring or inducing any additional permutations of the square root matrix. The approach taken here amounts to the same as is used by Reference [119] for the standard square root information filter (SRIF) with process noise. Since marginalization is implicitly involved, this operation "spreads" information [127] to correlations between the active map and robot states, the active portion of the map must be accounted for in the QR-factorization. The active map consists of all features that are being actively tracked at the current time and the parent poses that the features have been assigned. To account for these states, which have no dynamics by construction, the the state transition matrices are defined for both the map states and the robot states as

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{I}_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Phi}_x \end{bmatrix} \tag{4.15}$$

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{0}_m \\ \mathbf{0}_p \\ \mathbf{\Gamma}_x \end{bmatrix} \tag{4.16}$$

and where the size of the identity and zero matrices reflect the size of the passive and active maps. Returning to Eq. 4.10 and dropping the extra terms gives the generic state propagation equation

$$\delta\boldsymbol{\mu}_k = \mathbf{\Phi}\delta\boldsymbol{\mu}_{k-1} + \mathbf{\Gamma}\delta\mathbf{u}_{k-1} \tag{4.17}$$

where $\mathbf{w}_{k-1}$ has been replaced with a more general control term that incorporates it with a known input $\delta\boldsymbol{\alpha}_{k-1}$,

$$\delta\mathbf{u}_{k-1} = \delta\boldsymbol{\alpha}_{k-1} + \mathbf{w}_{k-1}. \tag{4.18}$$

This can be whitened by the square root of the discrete-time process noise,

$$\mathbf{Q}_k^{-1/2}\delta\mathbf{u}_{k-1} = \mathbf{Q}_k^{-1/2}\delta\boldsymbol{\alpha}_{k-1} + \mathbf{Q}_k^{-1/2}\mathbf{w}_{k-1} \tag{4.19}$$

$$\delta\tilde{\mathbf{u}}_{k-1} = \delta\tilde{\boldsymbol{\alpha}}_{k-1} + \tilde{\mathbf{w}}_{k-1}. \tag{4.20}$$

The true state at the previous time can be obtained from Eq. 4.17 as

$$\delta\boldsymbol{\mu}_{k-1} = \mathbf{\Phi}^{-1}[\delta\boldsymbol{\mu}_k - \mathbf{\Gamma}\delta\mathbf{u}_{k-1}] \tag{4.21}$$

and the prior information defined as before as

$$\delta\hat{\mathbf{b}}_{k-1} = \mathbf{R}_{k-1}\delta\boldsymbol{\mu}_{k-1} + \tilde{\boldsymbol{\eta}}_{k-1}. \tag{4.22}$$

Substituting Eq. 4.21 into the data equation in Eq. 4.22 yields

$$\delta\hat{\mathbf{b}}_{k-1} = \tilde{\mathbf{R}}_{k-1}\delta\boldsymbol{\mu}_k - \tilde{\mathbf{R}}_{k-1}\boldsymbol{\Gamma}\delta\mathbf{u}_{k-1} + \tilde{\boldsymbol{\eta}}_{k-1}.$$

where

$$\tilde{\mathbf{R}}_{k-1} = \mathbf{R}_{k-1}\boldsymbol{\Phi}^{-1}.$$

The cost function can now be created as the sum of the whitened error terms,

$$\mathcal{J}_p = \left\|\tilde{\boldsymbol{\eta}}_{k-1}\right\|^2 + \left\|\tilde{\mathbf{w}}_{k-1}\right\|^2 = \left\|\tilde{\mathbf{R}}_{k-1}\delta\boldsymbol{\mu}_k - \tilde{\mathbf{R}}_{k-1}\boldsymbol{\Gamma}\delta\mathbf{u}_{k-1} - \delta\hat{\mathbf{b}}_{k-1}\right\|^2 + \left\|\mathbf{Q}_k^{-1/2}\delta\mathbf{u}_{k-1} - \delta\tilde{\boldsymbol{\alpha}}_{k-1}\right\|^2 \tag{4.23}$$

$$\mathcal{J} = \left\| \begin{bmatrix} \mathbf{Q}_k^{-1/2} & \mathbf{0} \\ \tilde{\mathbf{R}}_{k-1}\boldsymbol{\Gamma} & \tilde{\mathbf{R}}_{k-1} \end{bmatrix} \begin{bmatrix} \delta\mathbf{u}_{k-1} \\ \delta\boldsymbol{\mu}_k \end{bmatrix} - \begin{bmatrix} \delta\tilde{\boldsymbol{\alpha}}_{k-1} \\ \delta\hat{\mathbf{b}}_{k-1} \end{bmatrix} \right\|^2 \tag{4.24}$$

This can now be minimized as before to find the desired result

$$\begin{bmatrix} \mathbf{Q}_k^{-1/2} & \mathbf{0} & \delta\tilde{\boldsymbol{\alpha}}_{k-1} \\ \tilde{\mathbf{R}}_{k-1}\boldsymbol{\Gamma} & \tilde{\mathbf{R}}_{k-1} & \delta\hat{\mathbf{b}}_{k-1} \end{bmatrix} = \boldsymbol{Q} \begin{bmatrix} \mathbf{R}_{\boldsymbol{\alpha}}^- & \mathbf{R}_{\boldsymbol{\alpha},\mu}^- & \delta\tilde{\boldsymbol{\alpha}}_k \\ \mathbf{0} & \mathbf{R}_k^- & \delta\hat{\mathbf{b}}_k^- \end{bmatrix} \tag{4.25}$$

where again, since we are dealing with error states, $\delta\tilde{\boldsymbol{\alpha}}_{k-1}$ and $\delta\hat{\mathbf{b}}_{k-1}$ can be set to zero. The relevant propagated quantities can be easily extracted from the lower right entries on the right-hand side of Eq. 4.25. This factorization must be computed for the combined subset of states that includes the active map states m and robot states x.

By following the strategies outlined above, the given algorithm provides constant-time visual-inertial odometry without explicit marginalization of past landmarks.

4.2 2D Linear Scenario

In order to assess the proposed algorithm's performance, a simulation using simplified two-dimensional dynamics and one-dimensional camera measurements was created. At the time of this work, the algorithm was laboriously called the "Square Root Extended Sparse Extended Information Filter" (SRESEIF) even though in this 2D scenario with linear dynamics it is in fact just a form of a linear SRIF. The dynamics of the robot are given by

$$\mathbf{x}_k^- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \bar{\mathrm{v}}_{k-1} + \bar{w}_{k-1} \tag{4.26}$$

where the state encompasses the robot position and the velocity input is treated as known with zero mean, white Gaussian process noise $\bar{w} \sim \mathcal{N}(\bar{0}, \mathbf{Q}_{k-1})$. A random map of features was generated and detections were pre-computed along a true trajectory using the one-dimensional, fully-calibrated camera measurement model given by

$$z_i = [0\ 1]\frac{{}^C\mathbf{h}_i}{{}^C\mathbf{h}_{i,z}} + \nu \tag{4.27}$$

where ${}^W\mathbf{h}_i$ is the relative direction of the i-th landmark in the world frame, ${}^C_W\mathbf{C}$ is the direction cosine matrix between the world and camera frames which is assumed to be known, and $\nu \sim \mathcal{N}(0, \sigma_\nu^2)$. The inverse depth measurement model, which requires the camera position $\hat{x}_{k_0}$ at which each landmark was first observed as well as the bearing ϕ_i and inverse depth ρ_i, is adapted to this scenario by defining

$$\mathbf{m}(\phi_i) = \begin{bmatrix} \sin\phi_i \\ \cos\phi_i \end{bmatrix} \tag{4.28}$$

and modeling the feature location as

$${}^W\mathbf{l}_i = \rho_i(\mathbf{x}_{k_0} - \mathbf{x}_k) + \mathbf{m}(\hat{\phi}_i) \tag{4.29}$$

A discussion on feature initialization for the inverse depth model is included in the Appendix. In addition to the SRESEIF implementation given in the previous sections, an EKF visual odometry algorithm with the inverse depth measurement model was implemented for direct comparison. In both filters, the number of detections at any one time step was limited to 15 so that computational comparisons can be made on the basis of overall state size alone.

Given the simulation parameters in Table 4.1, an example output of the filters is shown in Fig. 4.2 where the robot was given a velocity profile that resulted in a circular trajectory.

Table 4.1: 2D Simulation Parameters

Parameter	**Symbol**	**Value**	**Units**
Number of landmarks	m	200	
Initial position variance	σ_r^2	10	m^2
Process noise variance	σ_w^2	0.01	m^2
Measurement noise variance	σ_ν^2	0.001^2	m^2
Inverse depth prior mean	$\rho_{i,0}$	0.05	1/m
Inverse depth prior variance	σ_c^2	0.01	$1/\text{m}^2$
Maximum detections at any time	$n_{\max}$	15	
1D camera field of view	ψ	60	deg

The landmark positions are well-estimated with 3σ ellipses that consistently encompass their true locations. Landmarks with larger ellipses are generally more recent or farther away from the camera position. It is noted that the final uncertainty of the landmark positions, which is typically a symmetric Gaussian distribution, can never be less than the initial uncertainty of the robot position, which is expected given that no other external measurement sources are provided to the estimator.

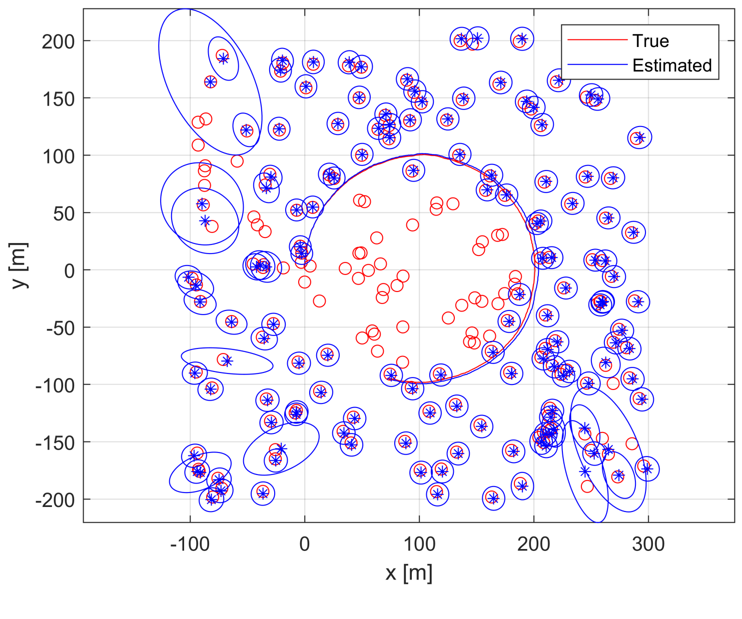


Figure 4.2: Trajectory and map estimated by robot traversing a circle using only odometry information and 1D camera measurements. 3σ uncertainty ellipses about each feature location are provided.

Fig. 4.3 shows the typical state error and associated 3σ position uncertainty for the EKF and SRESEIF given that there is an initial random error of magnitude σ_r in both dimensions. The variances are numerically indistinguishable but the state errors exhibit some slight differences. Possibly owing to the well-known numerical stability of square root filtering paradigm, the SRESEIF obtains a slight edge in terms of accuracy over the EKF. Results from 100 Monte Carlo runs of the SRESEIF filter using randomized landmark locations and with varying noise values and initial conditions are given in Fig. 4.4. These data show that the filter is as robust to initial perturbations as would be expected of any EKF-based solution.

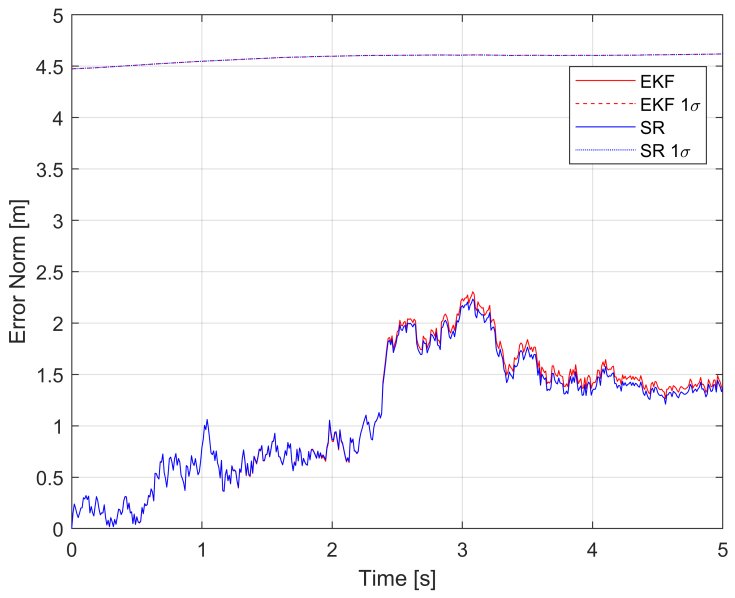


Figure 4.3: 2-norm of state errors over time along with associated 1σ uncertainties. The SRESEIF shows small accuracy gains likely due to the inherent stability of square root formulations.

The amount of time taken at each time step to execute each filter's main loop was saved alongside the size of the state vector at each time. This test was run 20 times for the same map and the data averaged over those trials so that a more reliable estimate of the actual time required for execution could be obtained. The EKF exhibits the classic quadratic behavior with respect to state size. The SRESEIF, on the other hand, maintains a relatively constant-time complexity as intended.

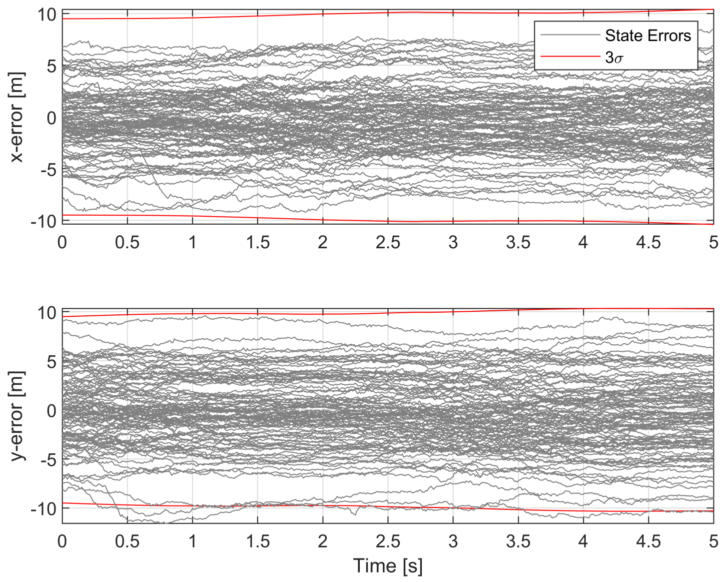


Figure 4.4: Monte Carlo plot of state errors from 100 runs of the SRESEIF with randomized maps and initial deviations.

4.3 3D TRN Scenario

Naturally, most VIO and VI-SLAM implementations in the literature are formulated for motion models tailored to operation on Earth, specifically for applications such as autonomous cars and unmanned aerial vehicles. These implementations typically assume that there is a constant gravity vector acting on the vehicle which has a magnitude that is either known or estimated alongside the other states. In the moon landing scenario considered here, we must account for the changing direction and magnitude of the gravity vector in the dynamic model as well as the fact that the celestial body itself is rotating. With that in mind, four reference frames are leveraged here: moon-centered inertial (MCI), moon-centered-moon-fixed (MCMF), the vehicle body frame, and the camera frame. The moon-centered inertial frame I, analogous to a standard ECI frame, is fixed to the center of the celestial body in question and its basis vectors point in constant inertial directions. The moon-centered-moon-fixed frame M, like the standard ECEF definition, has the same origin as the inertial frame but rotates with the planet or moon such that surface features remain static. The vehicle body frame B is a unit triad of basis vectors that is fixed to the spacecraft body and assumed here to align exactly with the spacecraft's IMU. Finally, the camera frame C has its origin at the optical center of the camera with the z-direction of the frame pointing through the boresight. This geometry is depicted diagrammatically in Figure 4.5.

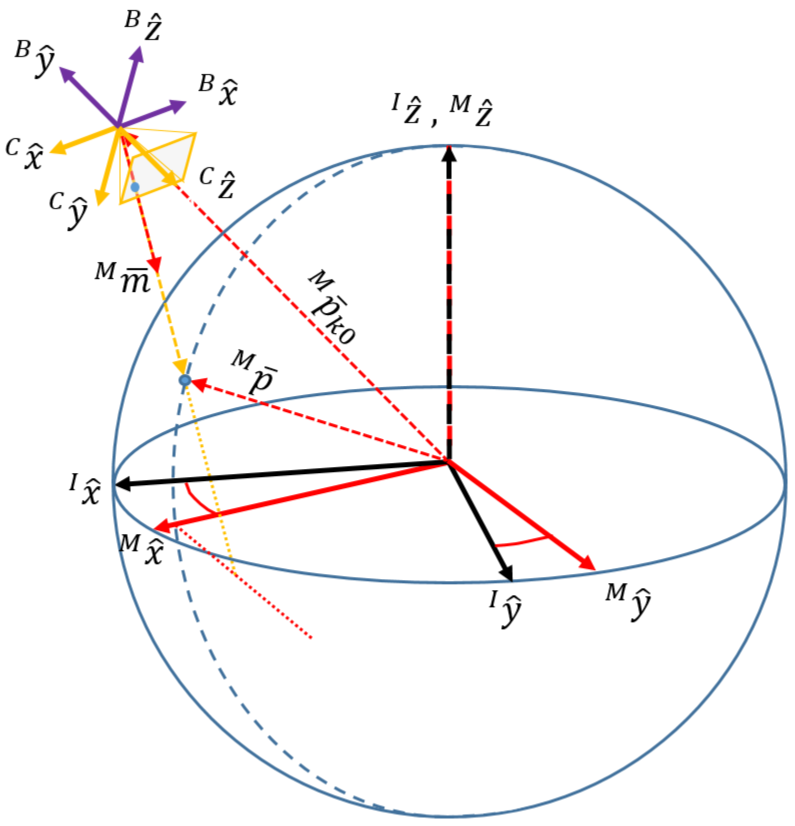


Figure 4.5: Reference frame geometry of the TRN problem: the I superscript denotes the Moon-Centered-Inertial frame, M denotes the Moon-Centered-Moon-Fixed frame, B is the vehicle body frame, and C is the vehicle camera frame.

The state vector used here is made up of the spacecraft states, which are always estimated and represent the vehicle's position and velocity, as well as the landmark states to be estimated. The navigation states of interest are the 6DOF pose and velocity,

$$\mathbf{x} = \begin{bmatrix} {}^M\mathbf{p} \\ {}^M\mathbf{v} \\ {}^B_I q \end{bmatrix} \tag{4.30}$$

where ${}^M\mathbf{p}$ is the position of the robot in MCMF frame, ${}^M\mathbf{v}$ is its velocity in the same frame, and

${}^{B}_{I}q$ is a quaternion which, using the JPL convention, parametrizes the attitude of the body frame with respect to the MCI frame. In a real operational context, a star tracker may be available for attitude estimation and would likely improve the results of the present filter.

All map states, instead of being represented as a 3×1 position vector, are represented using the inverse depth parametrization from Eq. 3.38. In the case of a camera or robot moving through a static environment on Earth, the angular quantites are related to a static frame such as East-North-Down (ENU) that is assumed inertial. In the present context of a planetary landing, these can most-easily be related to the MCMF frame because their true locations can be assumed to be static in that frame with respect to the rotating planet. This assumes that the rotation rate of the planet is known *a priori*, which is not typically an onerous assumption for larger celestial bodies.

4.3.1 Dynamic Models

The dynamics of the landing spacecraft largely follow the forms established in Section 3.1 and utilize IMU model replacement due to the presence of a significant active thrust force. This system can be written in state space form as

$$\delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x} + \mathbf{G}\mathbf{w} \tag{4.31}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{J}_g(\hat{\mathbf{p}}) - [\boldsymbol{\omega}_M \times]^2 & -2[\boldsymbol{\omega}_M \times] & -{}^{M}_{\hat{\mathbf{B}}}\mathbf{C}[\tilde{f}\times] \\ \mathbf{0} & \mathbf{0} & -[\tilde{\omega}\times] \end{bmatrix} \tag{4.32}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -{}^{M}_{\hat{\mathbf{B}}}\mathbf{C} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \tag{4.33}$$

Note that, in contrast to 3.1, a different state ordering and a different quaternion error definition were used for these results. The zero-mean, white Gaussian noise terms have been collected into vector $\mathbf{w}$,

$$\mathbf{w} = \begin{bmatrix} \mathbf{n}_f \\ \mathbf{n}_\omega \end{bmatrix} \tag{4.34}$$

which corresponds to the continuous IMU noise covariance matrix

$$\mathbf{Q} = \begin{bmatrix} \sigma_f^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_\omega^2\mathbf{I} \end{bmatrix} \tag{4.35}$$

Given the high sample rate of the IMU, an analytical approximations for the STM can be used. The state transition matrix between times t_k and t_{k+1} is approximated via

$$\mathbf{\Phi}_x(t_{k+1}, t_k) \approx \mathbf{I}_{9\times 9} + \mathbf{F}\Delta t + \frac{\mathbf{F}^2\Delta t}{2} \tag{4.36}$$

where Δt for the IMU is kept to a constant and acceptably-small 0.1 seconds. As noted in Section 3.1, the transition matrix for the noise can be calculated by

$$\mathbf{\Gamma}_x(t_{k+1}, t_k) \approx \int_{t_k}^{t_{k+1}} \mathbf{\Phi}_x(t_{k+1}, t_k)\mathbf{G}\mathrm{d}t\tau \tag{4.37}$$

over the same small Δt with a zero-order hold assumption. These approximations, while perfectly useful in a standard the EKF formulation, present a challenge for the SREIF because, due largely to the approximation in Equation 4.36, they result in

$$\tilde{\mathbf{Q}}_{k+1,k} = \mathbf{\Gamma}_x(t_{k+1}, t_k)\tilde{\mathbf{Q}}_k\mathbf{\Gamma}_x(t_{k+1}, t_k)^T \tag{4.38}$$

being not positive definite, causing a distinct problem for Eq. 4.14. In the current work, this issue was bypassed by adding another, very small ($\sigma_p \sim 10^{-10}$ m) Gaussian process noise term to $\mathbf{Q}$ corresponding to the position state in $\mathbf{G}$, e.g.

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -{}^{M}_{\hat{\mathbf{B}}}\mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \tag{4.39}$$

$$\mathbf{Q} = \begin{bmatrix} \sigma_p^2\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_f^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_\omega^2\mathbf{I} \end{bmatrix}. \tag{4.40}$$

4.3.2 Simulation Setup

The parameters used for the simulation are given in Table 4.2. The IMU specifications are based on a Collins Aerospace LITIS MEMS IMU [1] and the nadir-pointing camera is based on the Perseverance Rover's TRN camera. Using the true dynamics given in Eqs. 3.17 through 3.19, two types of lunar trajectories were generated. First, a purely orbital case was generated that has no thrust input and where the camera was purely nadir-pointing. Next, a notional powered descent landing trajectory was generated, also with a nadir-pointing camera. Integrating the generated acceleration and angular rate profiles forward in time generated the "truth" trajectory that could then be corrupted with noise as in Eqs. 3.21 through 3.20. In both cases, a random map of both mapped and opportunistic landmarks was generated beneath the camera. Mapped landmarks were generated at a lower rate and only when the vehicle was above an altitude of 30 kilometers to simulate a mission where only lower-resolution orbital maps were available prior to flight. An example of the gravity-turn landing trajectory on the surface of the moon is shown in Figure 4.9. Images were taken at a rate of 1 Hz and the IMU, which in reality can output data at 100 Hz, was instead sampled at 10 Hz to cut down on unneccessary computation for the purposes of this simulation.

The only *ad hoc* tuning parameters are the inverse depth prior mean ρ_{i0} and standard deviation σ_ρ, the former of which can theoretically be set to zero corresponding to an infinite range to the landmark. As long as the standard deviation is set such that zero (infinite range) is reasonably included in the distribution, the algorithm converges to a realistic estimate of the range to each landmark. However, we observed that a prior mean of zero does result in large pre-fit residuals at the first update after initialization, in which the filter compares the reprojection of a point at infinity to the new landmark measurement. If the imaging rate is kept constant, this implies that these initial residuals may get larger as the spacecraft gets closer to the surface and the disparity between successive views grows. This is clearly visible in in Figure 4.6, which shows one outlier

[1] `https://www.collinsaerospace.com/-/media/project/collinsaerospace/collinsaerospace-website/product-assets/marketing/l/litis/litis_ms_datasheet_a4_final_web.pdf`

for each initialized landmark in the image plane. Early on, the outliers are mostly in the v image frame dimension that corresponds to lateral motion. Later in the landing trajectory, the residuals are spread between both dimensions because the motion is more vertical, corresponding to motion in the camera depth direction.

With this in mind, the inverse depth prior mean was instead set to be the vehicle's current altitude above the surface,

$$\rho_{i,0} = \frac{1}{\|\hat{\mathbf{p}}_j\| - R_M} \tag{4.41}$$

where R_M is the nominal radius of the celestial body, which in this case is Luna. The algorithm was not very sensitive to the prior inverse depth standard deviation, so this value was set to a value of 0.01. Given that the inverse altitude was between 10^{-5} 1/m and 0.01 1/m, this allows for infinite range, and even negative ranges, to be accounted for in the prior distribution.

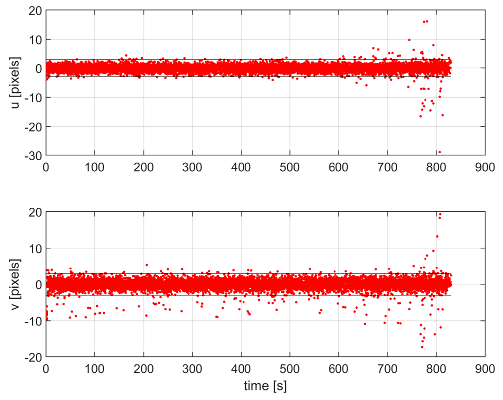


Figure 4.6: Image plane pre-fit residuals with inverse depth prior set to zero: outliers correspond to the first update after initialization for each landmark.

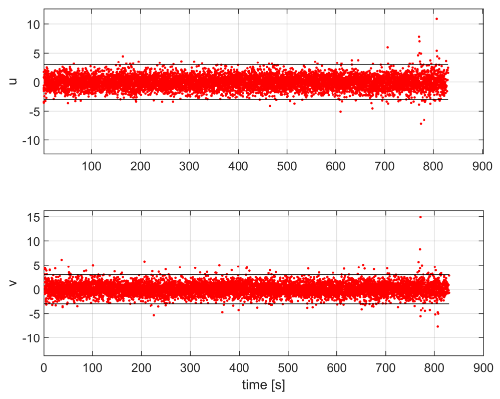


Figure 4.7: Image plane pre-fit residuals with inverse depth prior set to $1/r_k$: outliers correspond to the first update after initialization for each landmark.

4.4 Results

The position errors for both the orbital and landing trajectories with both ML and OL updates are plotted in Figures 4.11-4.12. Velocity and attitude errors for the same cases are likewise plotted in Figures 4.13-4.14 and 4.15-4.16 respectively. In all of these figures, the ensemble $\pm 3\sigma$ of

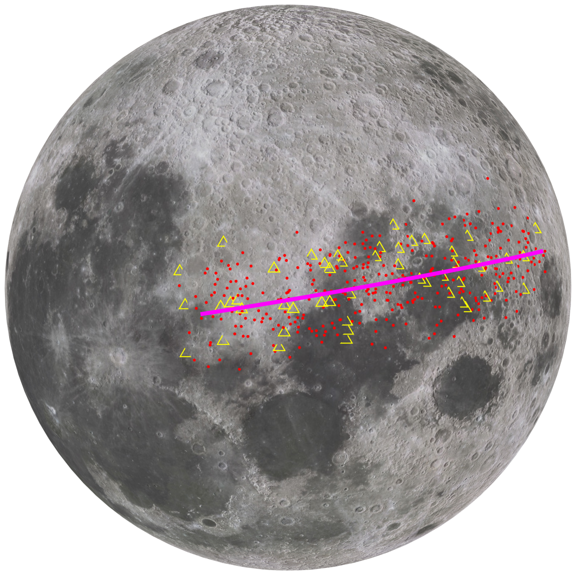


Figure 4.8: Orbital case: randomly generated landmark map on the lunar surface with both mapped landmarks (yellow) and opportunistic landmarks (red) beneath the orbital trajectory.

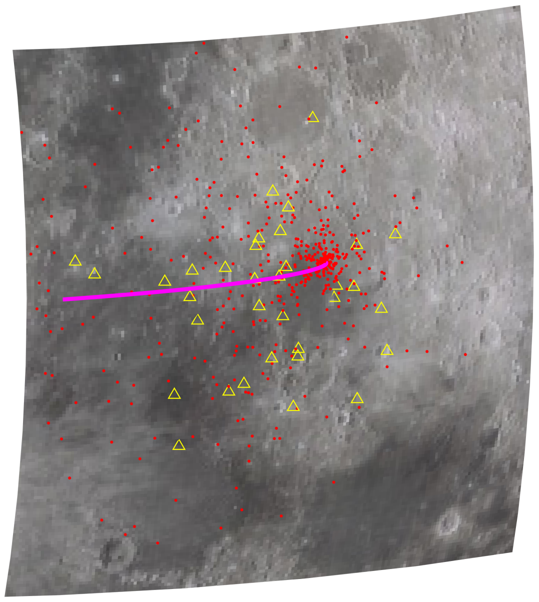


Figure 4.9: Powered descent case: randomly generated landmark map on the lunar surface with both mapped landmarks (yellow) and opportunistic landmarks (red) beneath the gravity-turn landing trajectory.

Table 4.2: 3D Simulation Parameters

Parameter	**Symbol**	**Value**	**Units**
Pixel noise standard dev.	σ_η	1	pixels
Camera focal length	f	5.8	mm
Camera pixel pitch	d_x, d_y	9.6	μm
Camera field of view	FOV	90	degrees
Camera frame rate	ω_c	1	Hz
Maximum number of OLs tracked simultaneously	n_{max}	15	-
Mapped landmark standard dev.	σ_m	10	m
Initial inverse depth mean	ρ_{i_0}	$1/r_k$	1/m
Initial inverse depth standard dev.	σ_ρ	$1/r_k$	1/m
Velocity random walk	σ_f	0.17	$\text{m/s}/\sqrt{\text{hr}}$
Angular random walk	σ_ω	0.05	$\text{rad}/\sqrt{\text{s}}$
Initial vehicle position standard dev.	σ_p	16.67	m
Initial vehicle velocity standard dev.	σ_v	0.67	m/s
Initial vehicle attitude standard dev.	σ_θ	0.01	rad

250 Monte Carlo trials and the average $\pm 3\sigma$ computed by the filter for these trials are provided. By comparing these two curves, we can note the overconfidence of the proposed filter in all of these

states. The mean of the errors, however, is close to zero which may indicate that consistency could be improved with *ad hoc* approaches such as measurement underweighting or covariance inflation. In fact, the filter is still overconfident when tracking only MLs, which may be why the authors of Reference [122] included covariance inflation and measurement iteration in their formulation.

For the purposes of comparison, we utilize the Normalized Estimation Error Squared metric (NEES) [9] in order to characterize the consistency of each estimator. If the estimator is consistent, meaning that the filter-predicted mean and covariance match the true mean and covariance, the scalar random variable

$$\epsilon_x = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T (\mathbf{P}_k^+)^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k) \tag{4.42}$$

should be χ^2 distributed with $n_{\mathbf{x}}$ degrees of freedom at every time k. For N Monte Carlo trials, the average NEES is simply

$$\bar{\epsilon}_x = \frac{1}{N} \sum_{i=1}^{N} \epsilon_{x,k}^i \tag{4.43}$$

Given a Type I error rate α, a filter can be deemed consistent if, at each time k,

$$\bar{\epsilon}_x \in [l_{\mathbf{x}}(\alpha, N), u_{\mathbf{x}}(\alpha, N)] \tag{4.44}$$

with probability $100(1-\alpha)$, where $[l_{\mathbf{x}}(\alpha, N), u_{\mathbf{x}}(\alpha, N)]$ are lower and upper tail bounds. We use $\alpha = 0.05$ in the results presented here.

Plotting the average Normalized Estimation Error Covariance (NEES) of the Monte Carlo trials in Figure 4.17 shows the filter inconsistency clearly but these results are very much in-line with those in the literature (see [106], [68]). Also noted in Reference [106] and in Section 3.3.4, these consistency results could potentially be improved by using an Anchored Homogenous Points (AHP) formulation which is very similar to the inverse depth model used here but instead parameterizes the ray in space using a unit vector instead of two spherical angles.

The utility of adding OL estimation to the TRN suite is evident in that the overall uncertainty for the position estimates is reduced from 133 meters $\pm 3\sigma$ to approximately 50 meters $\pm 3\sigma$. A comparison of the different measurement inclusion scenarios is given in Figure 4.18. This plot was created by normalizing the standard deviations of all states at all times by their respective initial standard deviations then summing the result at each time. As with ML estimation, tracking more features contemporaneously would decrease this result further. As expected for visual odometry, without including MLs the proposed algorithm results in a small amount of unavoidable drift over time on the order of 10^{-4} kilometers per kilometer.

The landmark errors resulting from the proposed algorithm flying the landing trajectory without any MLs is plotted in Figure 4.19 in their ENU frames. A bias is clearly evident in the vertical dimension but all observed landmarks are within the filter's $\pm 3\sigma$ estimates. This bias largely disappears if the errors are instead plotted by accounting for the spacecraft's final position estimate, as in Figure 4.20, implying that this bias is closely related to the drift of the spacecraft's position over time. This also indicates that the active landmark estimates would still be useful in a hazard avoidance algorithm because their geometry relative to the spacecraft at any point in time would be consistent with reality. These landmark errors all decrease substantially if MLs are included in any given trial, which is also expected.

The main benefit of this algorithm is computational. The structure of the square root information matrix allows us to retain all past information without explicit marginalization so that the current algorithm has a bounded computational requirement. Further, as is obvious in Figure 4.21, the square root information matrix is a more efficient way of storing the same information as the covariance matrix. The relative sparsity of the square root is not leveraged in the current framework but offers intruiging possibilities for future work.

In Figure 4.10, the quadratic complexity of the EKF VIO can be directly contrasted with the constant time complexity of the SREIF VIO. Loop closures in the SREIF are handled naturally, as in EKF-SLAM but, because the number of computations at any given time is governed by the size of the QR-factorization "window," their inclusion can compromise the computational efficiency of the algorithm as presented. This drawback will be addressed in Chapter 5.

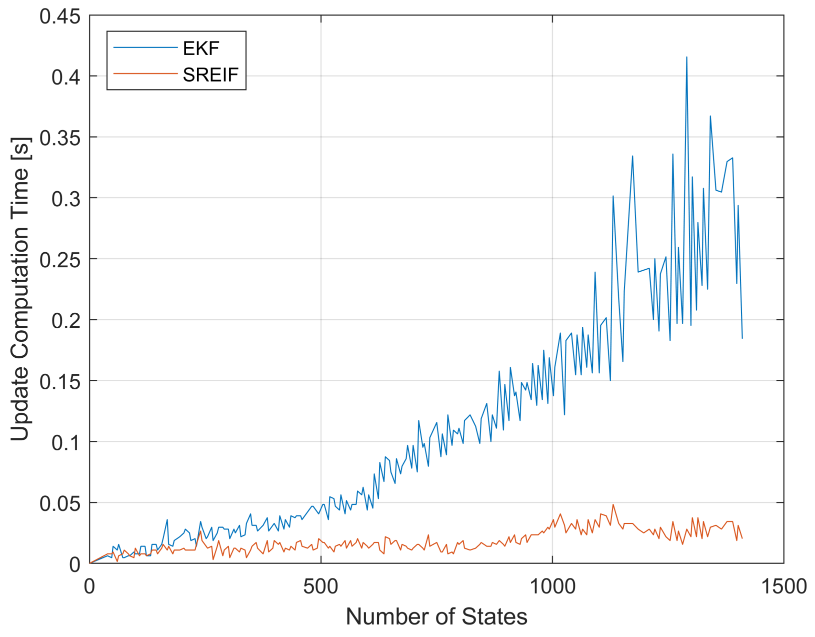


Figure 4.10: Computation time versus state size comparison between an inverse depth EKF-SLAM formulation and the presented SREIF algorithm using the 6DOF scenario presented herein. (updated from Reference [50])

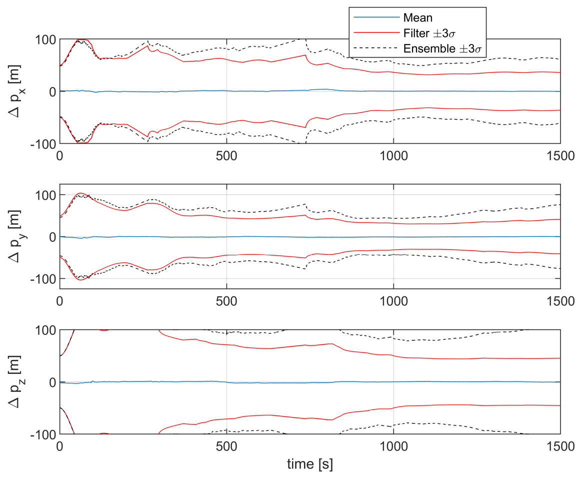


Figure 4.11: Orbital case: position errors with both OLs and MLs for 250 Monte Carlo trials in the MCMF frame.

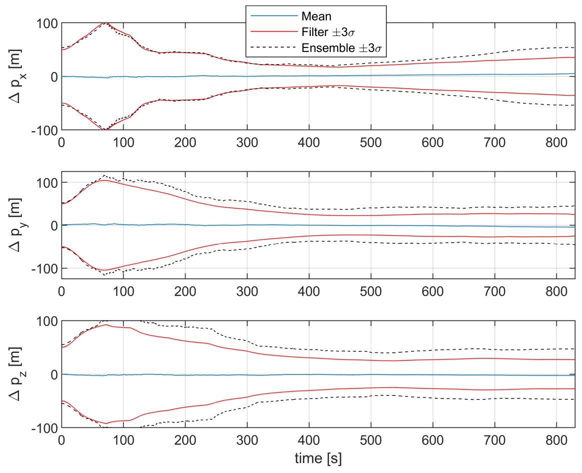


Figure 4.12: Powered descent case: position errors with both OLs and MLs for 250 Monte Carlo trials in the MCMF frame.

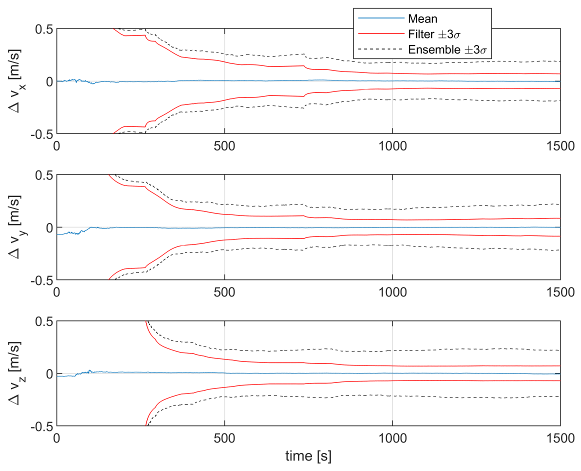


Figure 4.13: Orbital case: velocity errors with both OLs and MLs for 250 Monte Carlo trials in the MCMF frame.

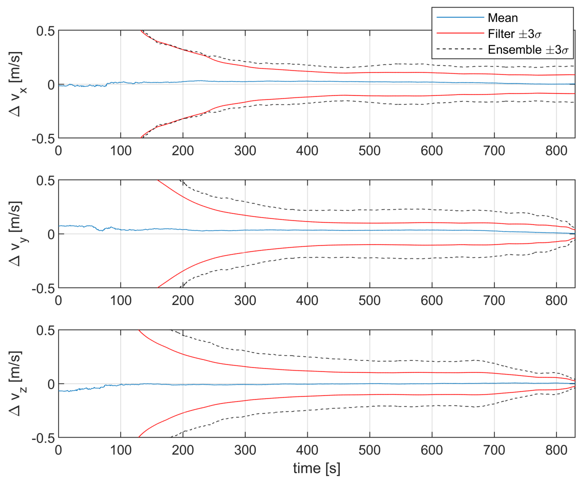


Figure 4.14: Powered descent case: velocity errors with both OLs and MLs for 250 Monte Carlo trials in the MCMF frame.

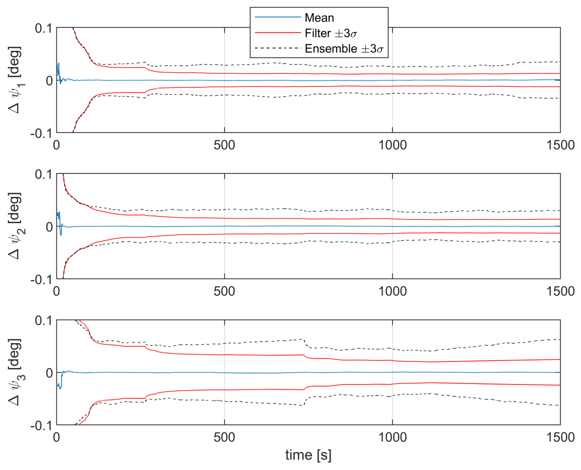


Figure 4.15: Orbital case: angular errors with both OLs and MLs for 250 Monte Carlo trials in the MCMF frame.

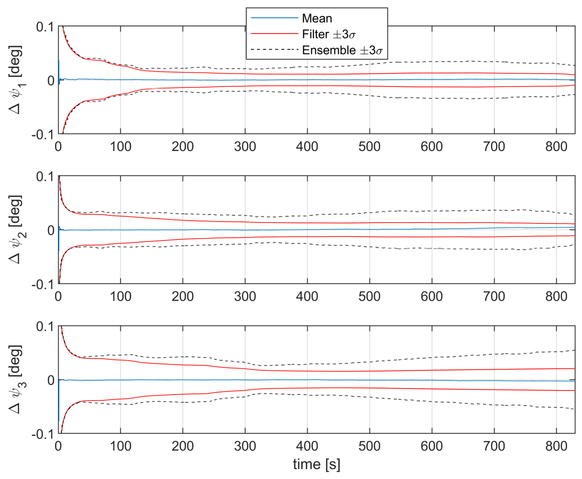


Figure 4.16: Powered descent case: angular errors with both OLs and MLs for 250 Monte Carlo trials in the MCMF frame.

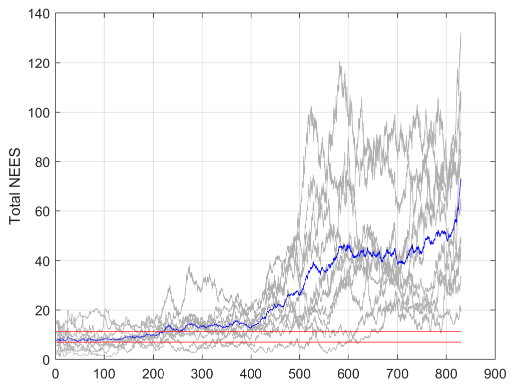


Figure 4.17: Normalized Estimation Error Squared (NEES) and Average NEES over time for 10 Monte Carlos in the case with no MLs.

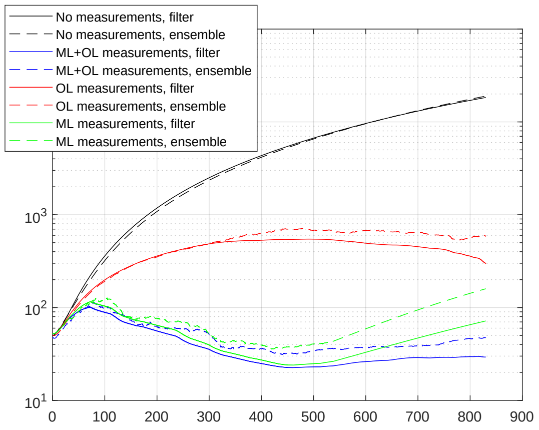


Figure 4.18: Comparison of standard deviation of errors resulting from 25 Monte Carlo trials for different measurement update cases.

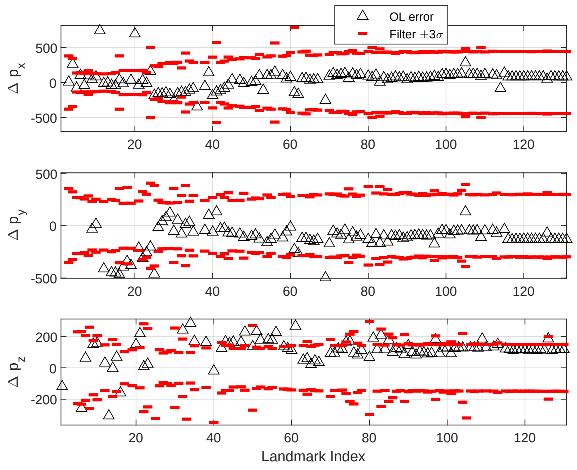


Figure 4.19: Opportunistic landmark errors for a single trial at the final time with corresponding $\pm 3\sigma$ values in landmark-relative ENU frames.

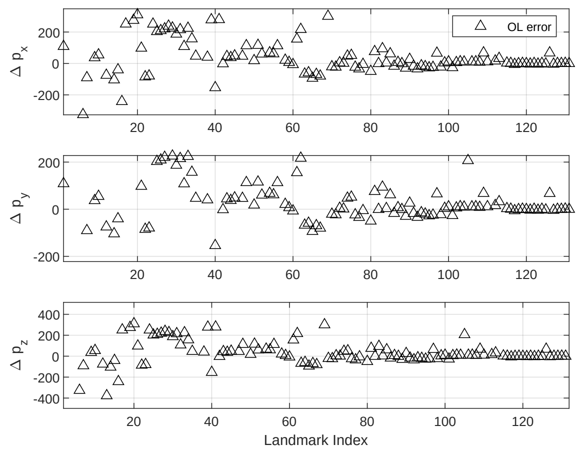


Figure 4.20: Opportunistic landmark errors for the same trial as in Figure 4.19, corrected for the final vehicle estimate, in landmark-relative ENU frames.

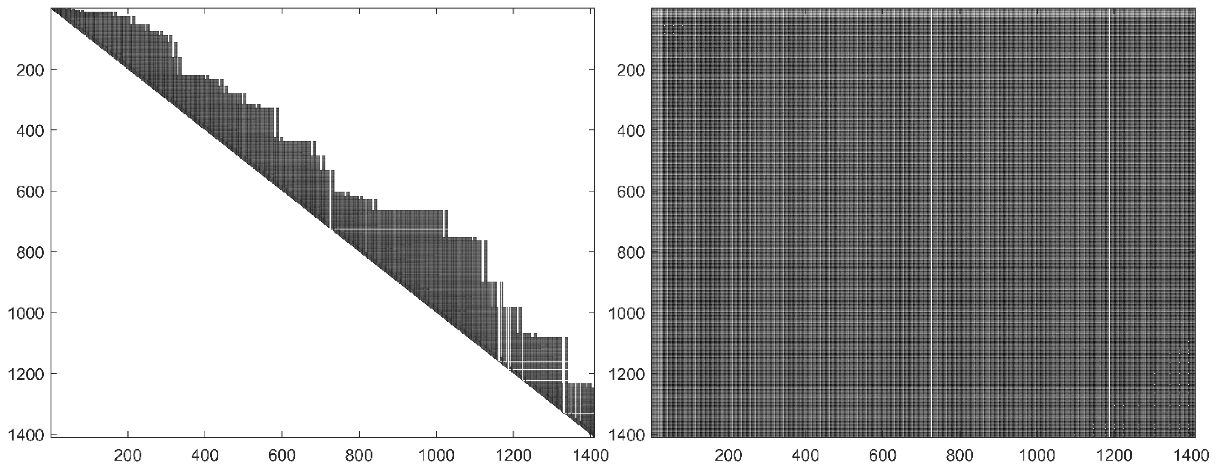


Figure 4.21: Comparison of a square root information matrix built by the SREIF and the associated covariance matrix with the same information. Darker entries represent larger values and whitespace represents zeros.

Chapter 5

Inverse Incremental Pose Augmentation SLAM

The SREIF of the previous section was shown to be a viable algorithm for visual-inertial odometry but with some apparent inconsistency issues and no strategy for the efficient computation of loop-closures. Building on this work, we present a significant evolutionary improvement which we dub "Inverse Incremental Pose Augmentation SLAM" or IIPA-SLAM (pronounced double IPA SLAM). The key improvements over the SREIF include the use of the novel Anchored Homogeneous Bundles (AHB) model previously described in Section 3.3.6, an efficient mean correction step for all states, and a new Schmidt-based relocalization strategy to incorporate information from previously-tracked landmarks. These developments and their mechanization in the new algorithm will be described in the succeeding sections and then demonstrated on both simulated and real datasets.

5.1 Filtering with Anchored Homogenous Bundles

As described in Section 3.3.6, the Anchored Homogenous Bundles (AHB) model allows us to represent a set of features initialized with the j-th anchor state using three states for the i-th feature: two that represent the direction to the feature in the camera frame as a homogenous vector $\mathbf{n}_i$ and one that represents the inverse of the depth ρ_i. While this represents more states than the Inverse Depth Bundles (IDB) model, and an equivalent number to our previous work using the Unified Inverse Depth model, it better captures the uncertainty of the first measurement and also retains the favorable initialization properties of the IDB model's "view-based" representation. Unlike the Unified model, when a feature is to be added to the state vector and estimated, it will be *independent* of all other states until another observation to said feature is processed.

Taking advantage of this property, we can make two important additions to the algorithm. The first is to realize that decisions about adding features to the state vector or not can be delayed by one observation *with no loss of information.* If a feature has been identified in two successive frames, it can simply be added to the state and square root information matrix just before the second observation is processed by the filter that correlates it to other states. In IIPA-SLAM, in addition to the n_{max} currently-tracked features, we maintain a pool of potential features from previous frames to draw from. When an active feature is deactivated, a new feature is selected from the pool for initialization. This feature will have already been seen in two frames and can thus be added immediately and processed in the same way as the already-active features. No special initialization Jacobians, as in Reference [22], are necessary.

This leads to a second benefit: it enables the use of *triangulation* to provide a good prior guess for the inverse depth of a given feature when initialized. We utilize the Direct Linear Transform (DLT) as described in Reference [55], where, based on two 2D observations of the same point, the 3D position of the i-th landmark can be calculated from

$$\begin{bmatrix} [{}^C\mathbf{h}_j \times]{}^C_M\mathbf{R}_j \\ [{}^C\mathbf{h}_k \times]{}^C_M\mathbf{R}_k \end{bmatrix} \mathbf{p}_{i,0} = \begin{bmatrix} [{}^C\mathbf{h}_j \times]{}^C_M\mathbf{R}_j {}^M\mathbf{p}_j \\ [{}^C\mathbf{h}_k \times]{}^C_M\mathbf{R}_k {}^M\mathbf{p}_k \end{bmatrix} \tag{5.1}$$

where ${}^C\mathbf{h}_j$ is the observation to the feature at anchor position ${}^M\mathbf{p}_j$ rotated into the appropriate camera frame by ${}^C_M\mathbf{R}_j$, with the same interpretation going for said variables at the time of initialization, t_k. The initial guess for the inverse depth can then be easily computed according to

$$\rho_{i,0} = \frac{1}{\|\mathbf{p}_{i,0} - \mathbf{p}_j\|} \tag{5.2}$$

As described previously, this estimate will be biased in depth but still represents a very good prior guess for the inverse depth model. As noted in Chapter 4, while the inverse depth model converges even with inverse depth priors set to 0, corresponding to infinite range, a good initial guess speeds up convergence significantly and thus makes measurement update iteration much less necessary to reach convergence. However, we note that the triangulation does assume that there is prior knowledge of the position of the spacecraft and the motion it experiences between observations.

5.2 Reconciling the Mean

An oversight of our previous work was that while the square root information matrix and vector retain a convenient, sparse pattern throughout operation when the previously-described augmentation strategy is used, the mean vector for all states is always in flux due to relinearization of the most-current active states. After a measurement update is computed at t_k, the corresponding error state correction is obtained via

$$\delta\boldsymbol{\mu}_k = (\mathbf{R}_k^+)^{-1}\delta\mathbf{b}^+ \tag{5.3}$$

where, as described previously, the augmentation scheme ensures that only the portions of $\mathbf{R}_k^+$ and $\delta\mathbf{b}^+$ that were affected by the measurements at t_k have a nonzero change. Indeed, $\mathbf{R}_k^+$ shows absolutely no change outside of the window of states being considered in the update. However, despite these properties, we observe that the result of Equation 5.3 typically contains nonzero elements in *all* entries of $\delta\boldsymbol{\mu}_k$ arising from the dense inverse of $\mathbf{R}_k^+$. The implication is that, while estimates of the active states carry forward all of the information of all of the prior measurements, subsequent relinearizations are not taken into account in the inactive states and their mean values are not automatically updated with new information.

If the only solution for the navigation states is desired, as in pure visual odometry, no further corrections need be made. The current best estimate of the active states is still the best estimate given the information up to the current time. On the other hand, if the best estimates of all inactive anchors and landmarks are desired too, or if a loop closure is to be computed, the estimates of the inactive states must be reconciled with the active ones before a consistent solution can be obtained. Throwing computational efficiency out the window, one possible solution would be to simply compute the entire $\delta\boldsymbol{\mu}_k$ vector at all measurement update times. Because the most recent states are being constantly relinearized, this requires computing any measurement update QR factorizations over the entire $\mathbf{R}_k$ matrix instead of just the windowed sub-block associated with the active states. Clearly, with an ever-growing SRI matrix, this would severely limit the computational performance of the algorithm.

Instead, we formulate a novel, efficient method for retaining the accumulated mean update information about inactive states that can later be applied to correct their estimates whenever an inversion of the full SRI matrix is executed. This improvement is also important for the consistency of the Schmidt-based relocalization routine described in Section 5.3.

5.2.1 Efficient Mean Correction

Equation 5.3 can be expanded into a partitioned form using block matrix inversion,

$$\delta\boldsymbol{\mu} = \begin{bmatrix} \mathbf{R}_1^{-1} & -\mathbf{R}_1^{-1}\mathbf{R}_{12}\mathbf{R}_2^{-1} \\ \mathbf{0} & \mathbf{R}_2^{-1} \end{bmatrix} \begin{bmatrix} \delta\mathbf{b}_1 \\ \delta\mathbf{b}_2 \end{bmatrix} \tag{5.4}$$

where the first row corresponds to inactive states and the second row corresponds to active states. The measurement update only involves the second row terms, and thus $\mathbf{R}_1^{-1}$ is unaffected. Carrying out the multiplication,

$$\delta\boldsymbol{\mu} = \begin{bmatrix} \mathbf{R}_1^{-1}\delta\mathbf{b}_1 - \mathbf{R}_1^{-1}\mathbf{R}_{12}\mathbf{R}_2^{-1}\delta\mathbf{b}_2 \\ \mathbf{R}_2^{-1}\delta\mathbf{b}_2 \end{bmatrix} \tag{5.5}$$

recall that, as a result of our augmentation strategy, $\delta\mathbf{b}_1$ is always zero, resulting in

$$\delta\boldsymbol{\mu} = \begin{bmatrix} -\mathbf{R}_1^{-1}\mathbf{R}_{12}\mathbf{R}_2^{-1}\delta\mathbf{b}_2 \\ \mathbf{R}_2^{-1}\delta\mathbf{b}_2 \end{bmatrix} = \begin{bmatrix} -\mathbf{R}_1^{-1}\mathbf{R}_{12}\delta\boldsymbol{\mu}_2 \\ \delta\boldsymbol{\mu}_2 \end{bmatrix} \tag{5.6}$$

Because of the structure of the SRI matrix as constructed, the rectangular sub-block $\mathbf{R}_{12}$ is typically very sparse and can be stored as a sparse matrix at the time of each update. $\delta\boldsymbol{\mu}_2$ is already being computed for the measurement update and is a single column with a bounded row dimension corresponding the the active states. The quantity

$$\delta\mathbf{b}_{12} = -\mathbf{R}_{12}\delta\boldsymbol{\mu}_2 \tag{5.7}$$

can thus be efficiently computed and saved using a sparse representation of $\mathbf{R}_{12}$ after each update. Assuming that the vehicle is progressively exploring a space and only re-encountering inactive features after some time, as we have assumed previously, this can be done in constant time. Elements of the entire mean vector could therefore be recursively computed after each update k by

$$\delta\boldsymbol{\mu}_k = \delta\boldsymbol{\mu}_{k-1} + \mathbf{R}_{1,k}^{-1}\delta\mathbf{b}_{12,k} \tag{5.8}$$

which would require the undesirable inversion of the growing $\mathbf{R}_{1,k}^{-1}$ matrix each time. In the previous equation, entries of the mean vector corresponding the the active states turn out to be zero since these have been updated already in the measurement update.

Recalling again that elements of $\mathbf{R}_1$ do not change after instantiation, its behavior vis-à-vis any $\delta\mathbf{b}_{12,k}$ is linear over all succeeding updates. Given this fact, the recursive equation could instead be computed just once with the initial $\delta\boldsymbol{\mu}_0 = \mathbf{0}$ after q updates as

$$\delta\boldsymbol{\mu}_k = \mathbf{R}_1^{-1}\sum_{k=1}^{q}\delta\mathbf{b}_{12,k} \tag{5.9}$$

where $\mathbf{R}_1^{-1}$ need only be obtained once for any reason, such as recovery of the full map or for the relocalization routine. Assuming that the uncertainty represented in $\mathbf{R}_1$ has been consistently represented, the corrections should be exact up to the linearization.

5.3 Schmidt-based Relocalization

In the SLAM literature, a "loop closure" occurs when a scene is re-observed by the robot after some intervening period of time and the trajectory and map must be updated to account for this new information. Because this necessarily affects large portions of the estimated map, as well as the current best-estimate of the robot position, the large amount of correlative terms introduced into the posterior covariance matrix must be handled deftly. iSAM2 utilizes a heuristic variable re-ordering approach whenever a loop is encountered in order to find a more sparse square root information matrix.[67]

Another conceivable approach is to treat previously-estimated landmarks as "mapped" with some known uncertainty, as we did with mapped landmarks in the planetary landing scenario, and to simply ignore the correlations between the navigation and landmark states. Experience has shown that using a marginalized covariance estimate of landmarks previously generated by the

SLAM algorithm results in overconfidence and inconsistency unless some form of *ad hoc* method, such as underweighting,[131] is utilized.

The mapped landmark idea is attractive because it can be cheaply computed, only requiring a full inverse of the **R** matrix once, and does not introduce correlations into the SRI matrix. However, in order to utilize previously-mapped landmarks consistently, we must turn to the classic Schmidt Kalman Filter (SKF),[128] also known as "consider" Kalman Filtering.[119] The idea behind these approaches is to consider all of the uncertainty information of the covariance matrix while only updating a subset of the estimated states. This has proven eminently useful in, for example, deep space navigation, where the position uncertainties of planets and asteroids can be "considered" without altering their states.

Schmidt-based algorithms in the square root inverse filtering domain have been recently developed rigorously by researchers at the University of Minnesota.[130] Their methodology was subsequently applied to a SLAM algorithm known as RISE-SLAM,[68] an MSCKF-based algorithm that represents the closest point-of-comparison to the present work in many ways. That work assumes that the map states are always contained in a large $\mathbf{R}_2$ portion of the overall upper-triangular matrix square root **R** and thus relies on plentiful re-ordering and re-factoring operations running on a separate thread to leverage their prior theoretical work on Schmidt-based inverse filtering. We adopt some of their terminology, specifically the delineation between "exploration-phase" and "relocalization-phase" landmarks, where the former refers to the first time a given landmark is tracked and estimated via SLAM. The latter refers to when a landmark is re-encountered and tracked without updating its states via Schmidt.

While the referenced report[130] was indespensable in the development of the following algorithm, we ultimately take a different tack to avoid any re-ordering operations on **R**. What results is a novel, Schmidt-based relocalization routine that, while not constant-time, represents a substantial improvement in terms of computational demand over both the full measurement update and a naive square root Schmidt update.

The cost function for a given measurement update can be written as

$$\mathcal{J} = \left\| \begin{bmatrix} \mathbf{R} \\ \mathbf{H} \end{bmatrix} \delta\mathbf{x} - \begin{bmatrix} \delta\mathbf{b} \\ \tilde{\mathbf{r}} \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_2 \\ \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{bmatrix} - \begin{bmatrix} \delta\mathbf{b}_1 \\ \delta\mathbf{b}_2 \\ \tilde{\mathbf{r}} \end{bmatrix} \right\|^2 \tag{5.10}$$

where the prior information is encompassed in the $\mathbf{R}$ terms and the $\mathbf{H}$ matrices represent the whitened measurement Jacobians with residuals $\tilde{\mathbf{r}}$. The 1 subscripts here denote the inactive, static part of the map while the 2 subscripts denote the active window of states. An visual example of this division is given in Figure 5.1 which illustrates the typical sparsity pattern of the submatrices.

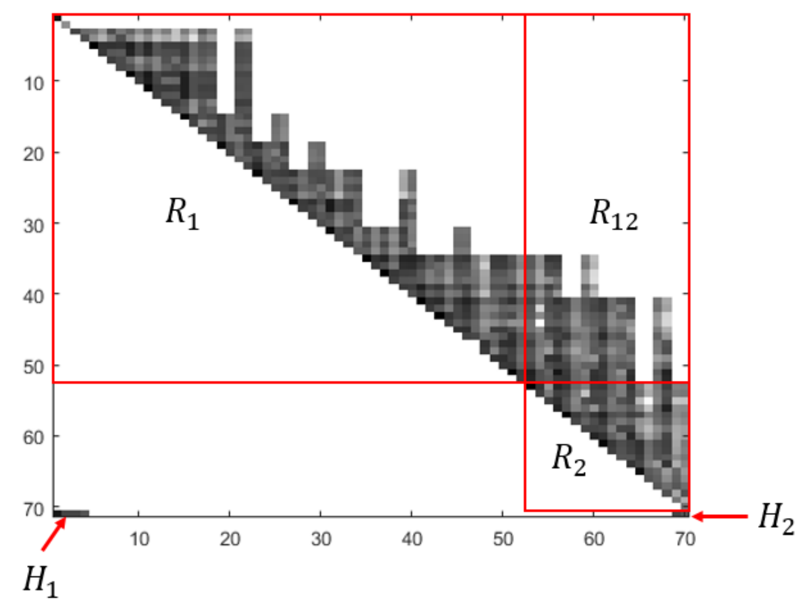


Figure 5.1: Example square root information matrix showing the block definitions when a relocalization feature is encountered.

Equation 5.10 could be equivalently written as separate $\delta\mathbf{x}_1$ and $\delta\mathbf{x}_2$ terms,

$$\mathcal{J} = \left\| \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \\ \mathbf{H}_1 \end{bmatrix} \delta\mathbf{x}_1 - \begin{bmatrix} \delta\mathbf{b}_1 \\ \tilde{\mathbf{r}} \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{R}_2 \\ \mathbf{H}_2 \end{bmatrix} \delta\mathbf{x}_2 - \begin{bmatrix} \delta\mathbf{b}_2 \\ \tilde{\mathbf{r}} \end{bmatrix} \right\|^2 \tag{5.11}$$

Following an approach that echoes the MSCKF measurement update,[86] the first matrix can be decomposed using a QR factorization:

$$\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \\ \mathbf{H}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1^{\oplus} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{5.12}$$

where $\mathbf{Q}_2$ is non-square and is associated with the nullspace of the matrix and the columns of $\mathbf{Q}_1$ interact with $\mathbf{R}_1^{\oplus}$. Substituting this back into the cost function yields

$$\mathcal{J} = \left\| \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1^{\oplus} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \delta\mathbf{x}_1 - \begin{bmatrix} \delta\mathbf{b}_1 \\ \tilde{\mathbf{r}} \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{R}_2 \\ \mathbf{H}_2 \end{bmatrix} \delta\mathbf{x}_2 - \begin{bmatrix} \delta\mathbf{b}_2 \\ \tilde{\mathbf{r}} \end{bmatrix} \right\|^2 \tag{5.13}$$

or

$$\mathcal{J} = \left\| \begin{bmatrix} \mathbf{Q}_1\mathbf{R}_1^{\oplus} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{bmatrix} - \begin{bmatrix} \delta\mathbf{b}_1 \\ \delta\mathbf{b}_2 \\ \tilde{\mathbf{r}} \end{bmatrix} \right\|^2 \tag{5.14}$$

If both terms in the above equation are multiplied $\mathbf{Q}_2^T$,

$$\mathcal{J} = \left\| \mathbf{Q}_2^T \begin{bmatrix} \mathbf{Q}_1\mathbf{R}_1^{\oplus} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{bmatrix} - \mathbf{Q}_2^T \begin{bmatrix} \delta\mathbf{b}_1 \\ \delta\mathbf{b}_2 \\ \tilde{\mathbf{r}} \end{bmatrix} \right\|^2 \tag{5.15}$$

$$\mathcal{J} = \left\| \mathbf{Q}_2^T \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{R}_2 \\ \mathbf{H}_2 \end{bmatrix} \delta\mathbf{x}_2 - \mathbf{Q}_2^T \begin{bmatrix} \delta\mathbf{b}_1 \\ \delta\mathbf{b}_2 \\ \tilde{\mathbf{r}} \end{bmatrix} \right\|^2 \tag{5.16}$$

the dependence on $\delta\mathbf{x}_1$ can be removed due to orthonormality. This new cost function can then be minimized to obtain the updated state and covariance information,

$$\mathbf{Q}_2^T \begin{bmatrix} \mathbf{R}_{12}^- & \delta\mathbf{b}_1^- \\ \mathbf{R}_2^- & \delta\mathbf{b}_2^- \\ \mathbf{H}_2^- & \tilde{\mathbf{r}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_2^{\ominus} & \delta\mathbf{b}_2^{\ominus} \\ \mathbf{H}_2^{\ominus} & \tilde{\mathbf{r}} \end{bmatrix} = \mathbf{Q}^+ \begin{bmatrix} \mathbf{R}_2^+ & \delta\mathbf{b}_2^+ \\ \mathbf{0} & \mathbf{e} \end{bmatrix} \tag{5.17}$$

where $\mathbf{R}_2^+$ and $\delta\mathbf{b}_2^+$ are updated information terms and the error state and covariance can be recovered as before.

The computation of the $\mathbf{Q}_1$ matrix, which is large and dense and includes all columns up to the active features, is required to form the necessary small, sparse $\mathbf{Q}_2$ matrix. This makes the naive computation of the Schmidt update just as, if not more, costly than the full update computation. Fortunately, since we know that the prior information $\mathbf{R}_1$ is not being updated, we can derive an alternative representation of the same operation. We can restate Equation 5.12, making the orthonormal matrix blocks explicit and absorbing the zero matrices into the upper triangular $\mathbf{R}$ blocks,

$$\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{H}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{R}_1^{\oplus} \\ \mathbf{0} \end{bmatrix} \tag{5.18}$$

and equivalently write

$$\begin{bmatrix} \mathbf{Q}_{11}^T & \mathbf{Q}_{21}^T \\ \mathbf{Q}_{12}^T & \mathbf{Q}_{22}^T \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{H}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^{\oplus} \\ \mathbf{0} \end{bmatrix} \tag{5.19}$$

from which the following relationship is evident:

$$\mathbf{Q}_{12}^T\mathbf{R}_1 + \mathbf{Q}_{22}^T\mathbf{H}_1 = \mathbf{0} \tag{5.20}$$

Solving for $\mathbf{Q}_{12}$ yields

$$\mathbf{Q}_{12} = -\mathbf{R}_1^{-T}\mathbf{H}_1^T\mathbf{Q}_{22} \tag{5.21}$$

which relies on $\mathbf{R}_1$ being invertible, which is always true. Another observation is that $\mathbf{Q}$ is orthonormal and satisfies the condition

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I} \tag{5.22}$$

Taking the first relationship in Equation 5.22, writing it in block form, and multiplying terms yields

$$\mathbf{Q}^T\mathbf{Q} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{11}^T & \mathbf{Q}_{21}^T \\ \mathbf{Q}_{12}^T & \mathbf{Q}_{22}^T \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{11}^T\mathbf{Q}_{11} + \mathbf{Q}_{21}^T\mathbf{Q}_{21} & \mathbf{Q}_{11}^T\mathbf{Q}_{12} + \mathbf{Q}_{21}^T\mathbf{Q}_{22} \\ \mathbf{Q}_{12}^T\mathbf{Q}_{11} + \mathbf{Q}_{22}^T\mathbf{Q}_{21} & \mathbf{Q}_{12}^T\mathbf{Q}_{12} + \mathbf{Q}_{22}^T\mathbf{Q}_{22} \end{bmatrix} \tag{5.23}$$

where the condition in the lower right can be extracted as

$$\mathbf{I} = \mathbf{Q}_{12}^T\mathbf{Q}_{12} + \mathbf{Q}_{22}^T\mathbf{Q}_{22} \tag{5.24}$$

Recalling the condition in Equation 5.21, we now have two equations involving both $\mathbf{Q}_{12}$ and $\mathbf{Q}_{22}$, the components of the desired $\mathbf{Q}_2$, that ultimately depend only on $\mathbf{R}_1$ and $\mathbf{H}_1$. The first condition can now be transposed and multiplied by itself to yield

$$\mathbf{Q}_{12}^T\mathbf{Q}_{12} = (-\mathbf{R}_1^{-T}\mathbf{H}_1^T\mathbf{Q}_{22})^T(-\mathbf{R}_1^{-T}\mathbf{H}_1^T\mathbf{Q}_{22}) \tag{5.25}$$

$$\mathbf{Q}_{12}^T\mathbf{Q}_{12} = \mathbf{Q}_{22}^T\mathbf{H}_1\mathbf{R}_1^{-1}\mathbf{R}_1^{-T}\mathbf{H}_1^T\mathbf{Q}_{22} \tag{5.26}$$

and equated to the second condition

$$\mathbf{I} - \mathbf{Q}_{22}^T\mathbf{Q}_{22} = \mathbf{Q}_{22}^T\mathbf{H}_1\mathbf{R}_1^{-1}\mathbf{R}_1^{-T}\mathbf{H}_1^T\mathbf{Q}_{22} \tag{5.27}$$

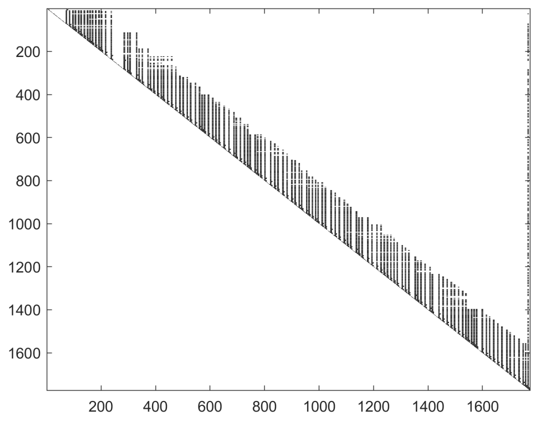
ultimately giving a closed-form expression for $\mathbf{Q}_{22}$,

$$\mathbf{Q}_{22}\mathbf{Q}_{22}^T = (\mathbf{H}_1\mathbf{R}_1^{-1}\mathbf{R}_1^{-T}\mathbf{H}_1^T + \mathbf{I})^{-1} = (\mathbf{H}_1\mathbf{P}_{11}\mathbf{H}_1^T + \mathbf{I})^{-1} \tag{5.28}$$

Interestingly, the product $\mathbf{R}_1^{-1}\mathbf{R}_1^{-T}$ is a sort of prior covariance matrix $\mathbf{P}_{11}$ and this equation represents something very similar to one that appears in the standard Kalman gain equation. By decomposing this equation and obtaining a solution for $\mathbf{Q}_{22}$, the result can be substituted into Equation 5.21 and the full nullspace projection matrix $[\mathbf{Q}_{12}, \mathbf{Q}_{22}]^T$ can be obtained without computing any QR factorization. $\mathbf{P}_1$ *is not* the full covariance matrix of the map. It comes from inverting $\mathbf{R}_1$ and multiplying that with its transpose whereas the full covariance matrix $\mathbf{P}$ comes from doing the same with the full $\mathbf{R}$ matrix.

The benefit of our relocalization approach is twofold: it reduces the measurement update to an SRI inversion and a small, bounded QR factorization instead of a very-large, growing QR factorization and also produces a posterior SRI that shows no fill-in between the exploration states and the relocalization states. The resulting SRI matrices from the full update and the Schmidt update are contrasted in Figure 5.2.

A final detail involves the extraction of $\mathbf{Q}_{22}$ from Equation 5.28. Since matrix square roots are not unique, we must determine which factorization is appropriate to obtain $\mathbf{Q}_{22}$. By inspection, we note that $\mathbf{Q}_{22}$ is necessarily symmetric and invertible and therefore is *not* the Cholesky factor


(a) Full measurement update.

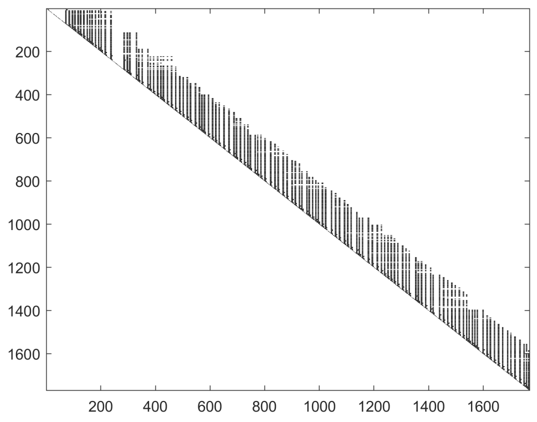

(b) Schmidt-based measurement update.

Figure 5.2: Comparison of the resulting square root information posterior after a relocalization feature is encountered and processed. Note the column of nonzero entries present on the right side of 5.2a.

of Equation 5.28. The correct factorization can be found using MATLAB's `sqrtm` function or, equivalently, by using either eigendecomposition or SVD factorization. For example, if

$$\sqrt{\mathbf{Q}_{22}\mathbf{Q}_{22}^T} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{5.29}$$

using SVD then,

$$\mathbf{Q}_{22} = \mathbf{U}\sqrt{\mathbf{\Sigma}}\mathbf{V}^T \tag{5.30}$$

With the previous discussion in mind, the Schmidt relocalization algorithm for updating the navigation states is provided as Algorithm 2.

5.3.1 Computational Considerations

The algorithm would be very slow if it was always computing QR factorizations over the entire state space. Instead, it is possible to only consider a subset of active states *and the states they are correlated to* at a given time in the factorization to achieve identical results. Referring back to Figure 5.1, the states that must be included can be visualized as any nonzero terms in $\mathbf{R}_{12}$. Determining which states these are can be nontrivial, and we have found that including past

Algorithm 2 Relocalize Navigation States

```
 1: Form full measurement update cost function $\mathcal{J}$
 2: if $f_{relocalize} \cap f_{update} \neq \varnothing$ then          ▷ Relocalization features detected
 3:     if flag = 0 then
 4:         Recover $\mathbf{S}$ from $\mathbf{R}$ up to highest relocalization landmark
 5:         flag = 1                                              ▷ Recover full map once
 6:     if $f_{deactivate} \neq \varnothing$ then
 7:         Update $\mathbf{S}_{12}$ via Equation 5.31
 8:     procedure RELOCALIZE($\mathbf{S}, \mathbf{H}$)
 9:         Partition into $\mathbf{S}_1$, $\mathbf{H}_1$ which encompass entire passive map
10:         Compute $\mathbf{A} \leftarrow \mathbf{H}_1\mathbf{S}_1$
11:         Compute $\mathbf{Q}_{22} \leftarrow \texttt{sqrtm}[(\mathbf{A}\mathbf{A}^T + \mathbf{I})^{-1}]$
12:         Compute $\mathbf{Q}_{12} \leftarrow -\mathbf{S}_1^T\mathbf{H}_1^T\mathbf{Q}_{22}$
13:         Form $\mathbf{Q}_2^T = [\mathbf{Q}_{12}^T, \mathbf{Q}_{22}^T]$
14:         Apply projection to update cost function, $\mathcal{J} \leftarrow \mathbf{Q}_2^T\mathcal{J}$
15: Minimize $\mathcal{J}$ via QR to obtain $\mathbf{R}^+$, $\delta\mathbf{b}^+$
```

anchors that were active when currently-active features were instantiated, along with the features those anchors were associated with, provides a well-sized update window with only a moderate increase in bookkeeping.

Because of the presence of $\mathbf{R}_1$ in Equation 5.21, an inversion and large matrix multiplication of the prior information will need to be computed. Because $\mathbf{R}_1$ is upper-triangular, this inversion can be done in $\mathcal{O}(n^2)$ time in the worst, dense case. This represents a large improvement over computing the full update via QR factorization, which can be $\mathcal{O}(n^3)$.[1]

Decomposing the inverse of $\mathbf{R}$ into a block form as in Equation 5.4, where $\mathbf{R}_1 \in \mathbb{R}^{m\times m}$ and $\mathbf{R}_2 \in \mathbb{R}^{p\times p}$, we again note the useful property $\mathbf{S}_1 = \mathbf{R}_1^{-1}$, which again implies that the typically-large inverse of $\mathbf{R}_1$ will not change in our formulation as more measurements are processed. $\mathbf{S}_1$ is dense and will progressively gain new, non-zero columns. Thus, an alternative approach to inverting $\mathbf{R}$ repeatedly might be to progressively update $\mathbf{S}_1$ with a newly-computed $\mathbf{S}_{12}$ matrix as poses are augmented and features are deactivated. In a similar way to the mean reconciliation routine in a previous section, the product

$$\mathbf{S}_{12} = -\mathbf{R}_1^{-1}\mathbf{R}_{12}\mathbf{R}_2^{-1} \tag{5.31}$$

could be computed to this end. If a standard matrix multiplication routine is assumed, this operation has approximately $\mathcal{O}(mmp)$ complexity and thus offers minimal advantage over the full $\mathcal{O}(n^2)$ inversion. However, both $-\mathbf{R}_1^{-1}$ and $\mathbf{R}_{12}$, can be considered and stored as sparse matrices, making the computation potentially more efficient. Their product, $\mathbf{S}_{12}$, is dense and so a lower bound on the time complexity of the multiplication in Equation 5.31 must be the number of elements in $\mathbf{S}_{12}$, or $\mathcal{O}(mp)$.

The complexity of sparse matrix multiplication is, in general, proportional to the number of nonzero elements in the columns of the constituent matrices. The sparse matrix multiplication algorithm utilized by the `CSparse` library, of which a similar algorithm used under-the-hood in MATLAB, has time complexity $\mathcal{O}(p + f + \text{nnz}(\mathbf{B}))$ where $\text{nnz}(\mathbf{B})$ is the number of nonzero entries in the second matrix in the multiplication $\mathbf{B}$ and f is the total number of floating point operations (*flops*).[28] In this case, the f term dominates and is proportional to the number of nonzero terms in the columns of each matrix in the multiplication, here denoted as a for R_1^{-1} and b for R_{12}. Taking this into account, the overall complexity of Equation 5.31 is approximately $\mathcal{O}(mabp^2)$ where m grows linearly and a, b, and p are bounded by constants by construction, making the overall multiplication linearly proportional to m and thus also linear in n. While no longer constant time, this represents a substantial improvement over the cubic complexity of the full QR factorization. When the algorithm goes back to "exploration" mode, where relocalization features are no longer being processed, the computational requirement returns to constant time.

5.4 IIPA-SLAM Algorithm Summary

IIPA-SLAM is a novel estimation algorithm that inhabits the "middle-ground" between visual EKF-SLAM and batch approaches, as mentioned in Section 2.4, drawing on lessons from each. We maintain and update a posterior distribution contained in the SRI matrix instead of a Hessian, and we process measurements only once, as in an EKF. However, we never marginalize states explicitly and we augment the dynamic states *after every image is processed*, in contrast to the SREIF of Chapter 4 and in-line with the batch approach as described in Chapter 2. The algorithm is then a sequential method for finding the EKF posterior over all states associated with image measurements and thus yields a discrete, "smooth" trajectory at these times with no extra smoothing at the end. Only an inversion of the SRI matrix, computed via back-substitution, is necessary to access the final mean. If the relocalization routine is not activated, the algorithm has constant time complexity except for the final inversion.

Figure 5.3 shows a representation of the Bayes Network of the algorithm, which corresponds to the structure of the SRI matrix, in comparison to a batch approach and the basic EKF-SLAM approach that marginalizes past poses. Importantly, while the batch and windowed filter approaches appear similar graphically, they differ computationally in that the windowed solution is re-linearized each time a measurement is processed. Also, whereas we always use a chronological variable ordering, the variable ordering of batch algorithms may be different or subject to many reorderings during computation.

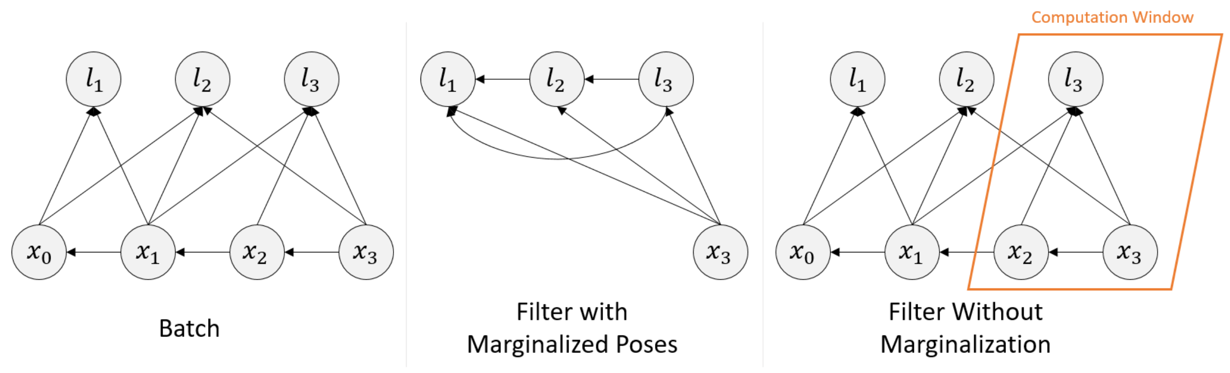

Computation Window
l_1
l_2
l_3
x_0
x_1
x_2
x_3
Batch
Filter with Marginalized Poses
Filter Without Marginalization

Figure 5.3: Bayes Network representations of various landmark-based SLAM solutions. The center figure is the fully-connected network that would result from marginalizing past poses out of the posterior, as in EKF-SLAM. The right figure shows the IIPA-SLAM approach: by utilizing the SRI matrix, we can only consider the latest states in the computation window to maintain sparsity and computational efficiency.

Another implication of the IIPA-SLAM SRI updating methodology is that if another type of measurement, such as from LIDAR or GPS, is processed at the most-current state, the information is automatically accounted-for in all past anchor states and landmarks. With no extra computational effort, the map and its uncertainty are updated and corrected with this new information because they are conditionally dependent on more recent states. Given these qualities, the present algorithm may have applications in other estimation problems outside of just SLAM.

IIPA-SLAM can also efficiently handle IMU measurements since the “Propagation via Marginalization” routine from the SREIF can still be utilized. Unlike the SREIF, however, the PvM routine here is never called immedately after an image is processed, avoiding the need to include the active map in the computation. This is beneficial because it avoids augmenting thousands of poses for IMU-propagated states and also prevents the inclusion of a large number of terms in the PvM routine, which would slow it down significantly in a real-time application.

Algorithm 3 IIPA-SLAM

```
Require: $\mathbf{P}_0$,$\boldsymbol{\mu}_0$ (a priori robot estimate)
 1: Initialization:  Compute initial square root using Cholesky factorization $\mathbf{R}_0 \leftarrow \mathbf{P}_0$
 2: IMAGE_FLAG $\leftarrow 0$
 3: procedure ((p)erform SLAM)
 4:     if IMAGE_FLAG then
 5:         Propagate state via augmentation using QR, $\mathbf{R}_k^-$,$\mathbf{x}_k^- \leftarrow \mathbf{R}_{k-1}$,$\mathbf{x}_{k-1}$ (Equation 4.14)
 6:         IMAGE_FLAG $\leftarrow 0$
 7:     else
 8:         Propagate state via marginalization using QR, $\mathbf{R}_k^-$,$\mathbf{x}_k^- \leftarrow \mathbf{R}_{k-1}$,$\mathbf{x}_{k-1}$ (Equation 4.25)
 9:     Recover mean of active states $\hat{\boldsymbol{\mu}}_{k,active}^-$
10:     if Image is to be processed then
11:         IMAGE_FLAG $\leftarrow 1$
12:         Identify features, associate to tracks
13:         Maintain pool of previously-detected, uninitialized features
14:         if $n_{active} \leq n_{max}$ then
15:             Initialize features $\rightarrow \rho_{i,0}$, $\mathbf{s}_{i,0}$, $\mathbf{R}_{i,0}$
16:         Scope update window
17:         Call Algorithm 2, if applicable                          ▷ Relocalization
18:         Compute measurement update over window via QR, $\mathbf{R}_k^+$,$\delta\mathbf{b}_k^+$ (Equation 4.9)
19:         Recover mean of robot and active map states $\mathbf{x}_k^-$,$\boldsymbol{\mu}_{l,k}^-$
20:         Compute and save $\delta\mathbf{b}_{12,k}$ (Equation 5.7)
21:     if Full $\mathbf{R}^{-1}$ was computed then
22:         Reconcile inactive states with current linearization (Equation 5.8)
```

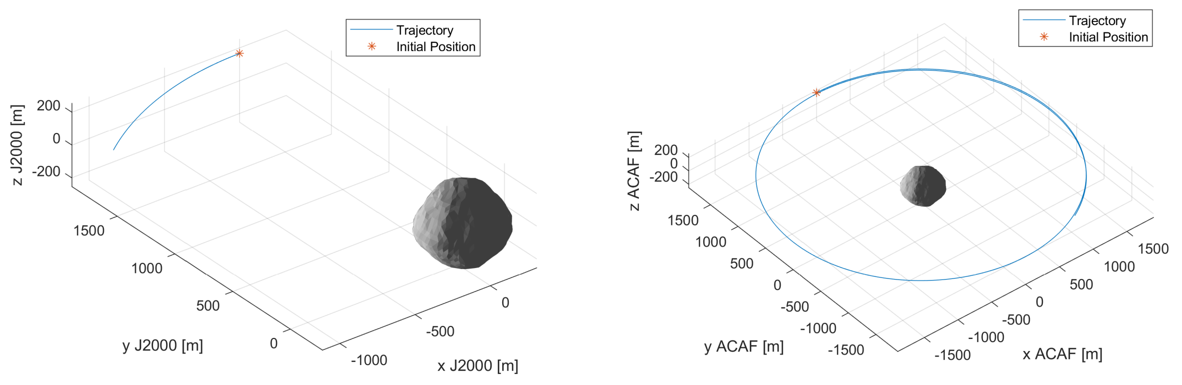
5.5 Asteroid-Relative Navigation and Mapping Scenarios

We imagine a scenario where a probe is orbiting a small rubble-pile asteroid similar to Bennu and taking images of the surface. Using Bennu as a reference, we simulate a nearly-circular orbit with a semi-major axis of 1.9 kilometers and inclination of 5 degrees and perturbations from SRP and the Sun's gravity. The resulting trajectory is visualized in both Asteroid-Centered Inertial (ACI) and Asteroid-Centered-Asteroid-Fixed (ACAF) frames in Figure 5.4. Using the camera parameters of the OSIRIS-REx SAMCAM, given in Table 5.1, this yields well-sized images of the body. Given the very-slow orbital motion in these cases, where the sidereal rotation of the body actually outpaces the orbital angular rate, images were only processed every 100 seconds. Two methods of generating image feature measurements are tested here, respectively labeled "Sandbox" and "Rendered Images" and each will be described in more detail below.

Table 5.1: Asteroid Navigation Simulation Parameters

Parameter	**Symbol**	**Value**	**Units**
Pixel noise standard dev.	σ_η	0.25	pixels
Camera focal length	f	24	mm
Camera pixel pitch	d_x, d_y	8.5	μm
Camera field of view	FOV	4	degrees
Image size (square)	n_{pix}	1037	pixels
Camera frame rate	ω_c	0.01	Hz
Maximum number of landmarks tracked simultaneously	n_{max}	20	-
Initial inverse depth standard dev.	σ_ρ	$5\rho_i$	1/m
Angular random walk	σ_ω	0.05	rad/$\sqrt{\text{s}}$
Initial vehicle position standard dev.	σ_p	50	m
Initial vehicle velocity standard dev.	σ_v	0.1	cm/s
Initial vehicle attitude standard dev.	σ_θ	20	arcsec

For the sandbox simulation, two cases are presented: one without estimating static parameters and one with estimating static parameters. In the former case, the navigation state vector consists of the inertial-to-spacecraft body attitude as well as the asteroid-fixed position and velocity following the conventions in Section 3.2.1. In the latter case, we treat the attitude as a known input and include various parameters in the state vector. The necessary Jacobians for including these parameters will be provided in the Appendix. For the rendered images case, only the non-parameter case is analyzed.



(a) Asteroid-Centered Inertial frame (b) Asteroid-Centered-Asteroid-Fixed Frame

Figure 5.4: Spacecraft trajectory around simulated Bennu-like asteroid. The asteroid's rotation rate exceeds the orbital angular rate, making the motion of the spacecraft appear retrograde.

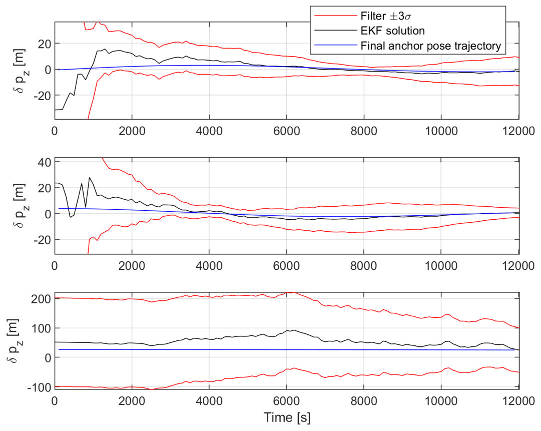
5.6 Bennu, Sandbox Simulation

We extracted the vertices of the low-resolution RADAR-based shape model of Bennu available online[1] and treated these as surface landmarks with precisely-known truth. Following the perfect pinhole model of Equation 3.34, 180 image measurements were generated with zero-mean, Gaussian random noise with a standard deviation of 0.25 pixels. Two sets of results are presented, with and without the estimation of asteroid parameters. In the first case, the state includes the inertial-to-spacecraft attitude as well as ACAF relative position and velocity vectors. The attitude is given a strong prior of ± 20 arcseconds, which is reasonable if the spacecraft has an active star tracker. In the parameter case, the attitude is treated as a perfectly-known input but we estimate the pole right ascension and declination as well as the rotation rate of the asteroid, the SRP scaling factor, and the Keplerian gravitational parameter.

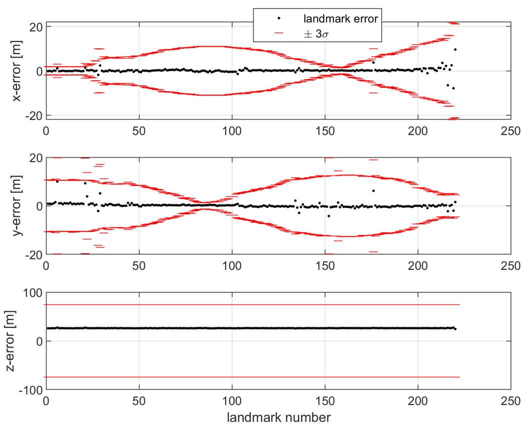
[1] `https://www.asteroidmission.org/updated-bennu-shape-model-3d-files/`

5.6.1 No Relocalization

We first process 120 of the 180 sandbox images that occur just before a relocalization feature is detected. The position and landmark errors and the filter-computed standard deviations of a single, randomly perturbed case are plotted in Figure 5.5. This case is illustrative because it shows that in the z-direction, which is very close to alignment with the orbit normal and the rotation pole of the simulated asteroid, the position is not well-estimated due to the lack of motion in that dimension and an initial bias persists. This lack of information is reflected in both the position errors of the spacecraft and the landmarks as well as their marginal variances. Another visualization of the asteroid and landmarks is given in Figure 5.6. This result is consistent with our expectation: without additional out-of-plane information, the estimator cannot converge on the true value in this dimension.



(a) Position errors and the filter-computed standard deviations compared with truth.



(b) Landmark errors and filter-computed standard devations.

Figure 5.5: Results from a single run of the algorithm with a random perturbation on all state components.

In Figure 5.5a, we also plot the "Final anchor pose trajectory", which represents the filter's final estimate of all of the prior anchor poses. This is retrieved by simply inverting the final SRI matrix and extracting the anchor poses from the state, which is done anyway in order to recover the landmark states in Cartesian coordinates as required by Equation 3.45. Unlike the jagged time

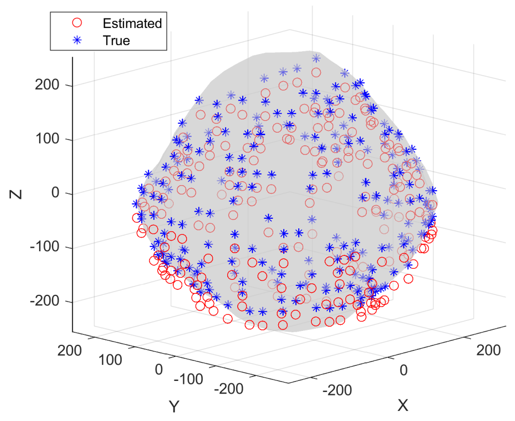


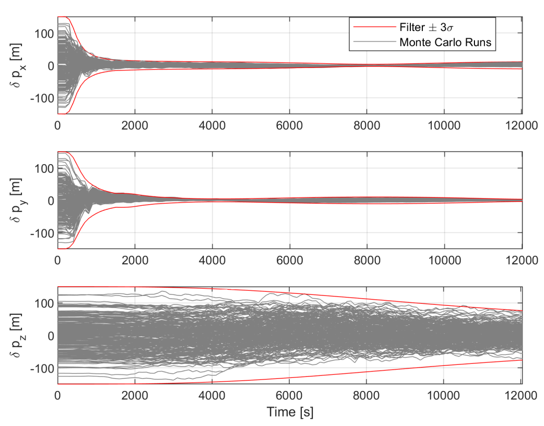
Figure 5.6: Landmark errors of single perturbed case plotted with Bennu shape model.

history of the state, which is a classic feature of EKF solutions, this trajectory is smooth.

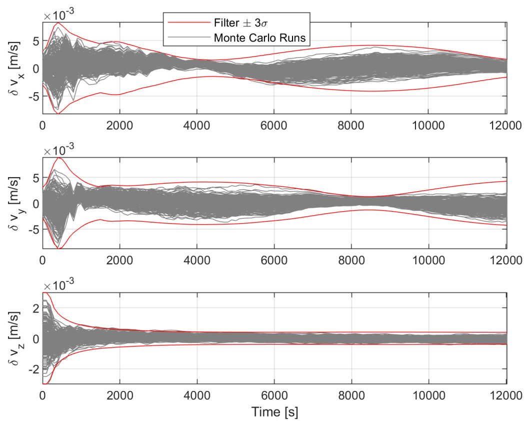
To rigorously assess the consistency of the algorithm in this idealized scenario, we turn again to Monte Carlo analysis and the average NEES test. 250 trials were run with randomized attitude, position, and velocity errors according to the standard deviations given in Table 5.1. Only one set of randomly-perturbed image measurements was used, so all trials utilize the same measurements at the same times. In Figures 5.7a-5.8a, the errors over the trajectory are plotted in a Radial-Tangential-Normal (RTN) frame along with the filter-computed standard deviations. The $\pm 3\sigma$ curves were computed along a zero-perturbation case and so represent the filter's knowledge of the uncertainty in the nominal case. The average time to compute one of Monte Carlo trial was 5.63 seconds on the test computer.

Applying the average NEES test over the 250 Monte Carlo trials with a 95% confidence interval and nine degrees of freedom yields upper and lower bounds of 9.45 and 8.56 respectively. Figure 5.9a plots the average NEES for the trajectory, showing that in this case the algorithm is largely consistent. Near the end of the trajectory, when measurements are no longer being processed, the filter is actually underconfident. Also provided in Figure 5.9b is the NEES broken into the attitude, position, and velocity components, each with three degrees of freedom.

The static parameters that we now add to the formulation are the asteroid's pole right

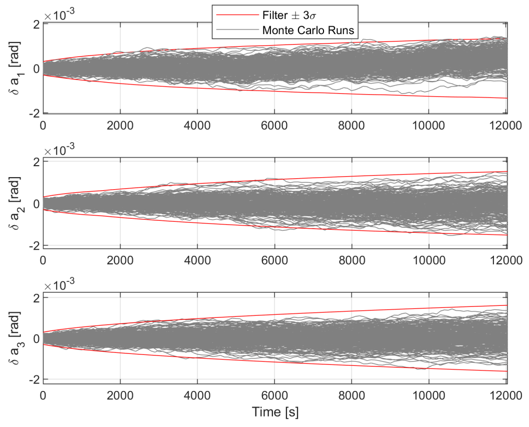


(a) Monte Carlo position errors and the filter-computed standard deviations compared with truth.

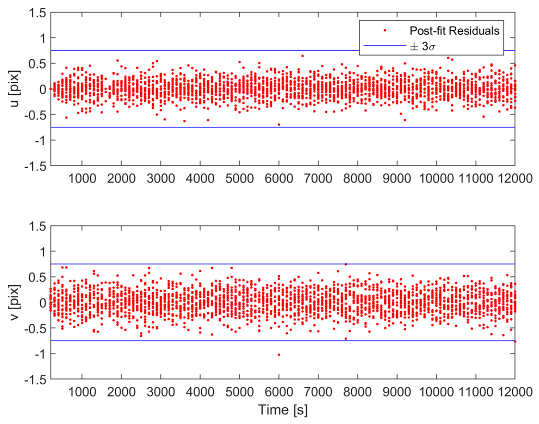


(b) Monte Carlo velocity errors and filter-computed standard devations.

Figure 5.7: Results from 250 trials of the algorithm with a random perturbation on all state components.



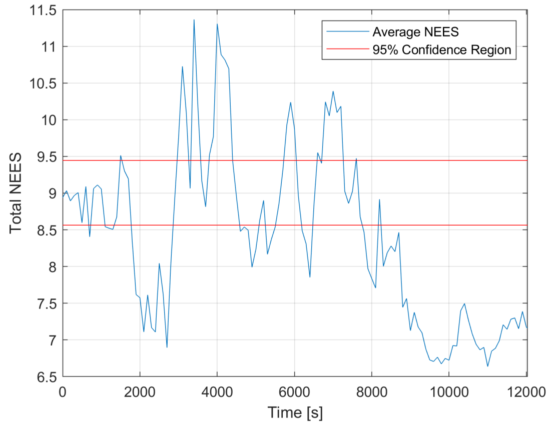
(a) Monte Carlo attitude errors and the filter-computed standard deviations compared with truth.



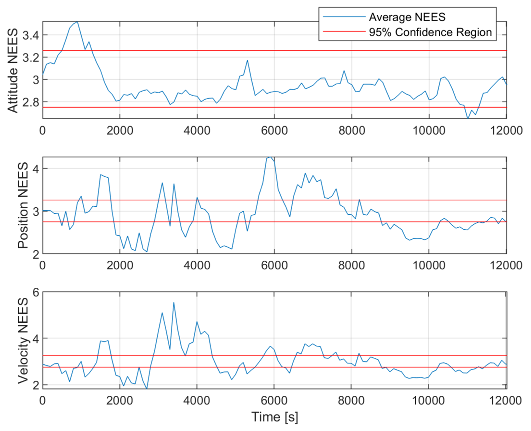
(b) Postfit residuals and measurement standard deviations.

Figure 5.8: Attitude errors and postfit residuals for the sandbox case from 250 trials of the algorithm with a random perturbation on all state components.

ascension α and declination ϕ, rotation rate ω_a, the SRP coefficient of the spacecraft η, and the



(a) Average NEES for sandbox case.



(b) Average NEES for sandbox case separated into state components.

Figure 5.9: NEES tests for case with no parameters and no relocalization.

gravitational parameter of the body μ. These quantities have dynamics governed by

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\phi} \\ \dot{\omega}_a \\ \dot{\eta} \\ \dot{\mu} \end{bmatrix} = \begin{bmatrix} w_\alpha \\ w_\phi \\ w_\omega \\ w_\eta \\ w_\mu \end{bmatrix} \tag{5.32}$$

where each w term represents a zero-mean, scalar Gaussian white noise sequence. We include these noisy inputs in order to ensure that $\mathbf{\Gamma}(t_{k+1}, t_k)\mathbf{Q}_k\mathbf{\Gamma}(t_{k+1}, t_k)^T$ in Equation 2.21 is not rank deficient. The constant-time process noise covariance matrix of these terms is set to be very small, e.g.

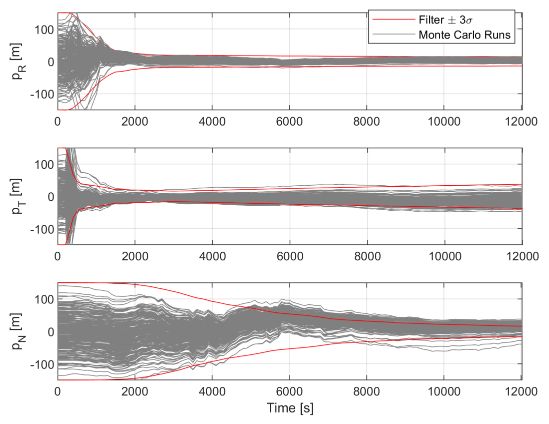
$$\mathbf{Q}_k = \text{diag}([(1 \times 10^{-10}\text{rad})^2, (1 \times 10^{-10}\text{rad})^2, (1 \times 10^{-10}\text{rad/s})^2, (1 \times 10^{-10})^2, (0.1 \times 10^{-10}\text{m}^3/\text{s}^2)^2]) \tag{5.33}$$

such that they have negligible influence. The initial state covariance for the parameters is

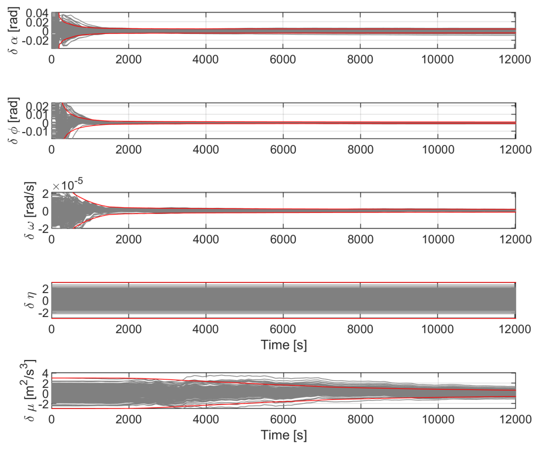
$$\mathbf{P}_0 = \text{diag}([(0.05\text{rad})^2, (0.05\text{rad})^2, (1 \times 10^{-5}\text{rad/s})^2, 1, (1\text{m}^3/\text{s}^2)^2]) \tag{5.34}$$

which is reasonable based on the expected accuracy of pre-mission fits of ground-based observational data.[33]

With no changes in the underlying formulation except for the addition of the new parameters, we observe in Figure 5.10 that the quantities are largely well-estimated over 250 Monte Carlo trials but that the algorithm is clearly overconfident. No information about the SRP scale factor is gained by the estimator and this is consistent with the ensemble of Monte Carlo trials. The overconfidence can be partially alleviated by underweighting the measurements, and doing so by setting the expected measurement standard deviation to 0.375 pixels 1σ results in Figures 5.11a and 5.11b. We observe that the strange behaviors seen in these plots can also be attenuated if the spacecraft trajectory is more inclined with respect to the asteroid's rotation plane than in these test cases, likely due to more favorable observability geometry.



(a) Monte Carlo position errors and the filter-computed standard deviations compared with truth for the with-parameters case.

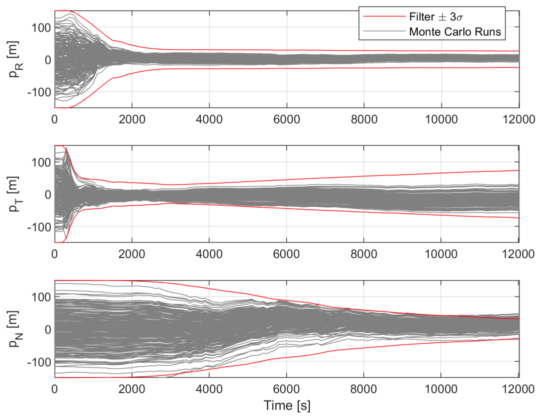


(b) Postfit residuals and measurement standard deviations.

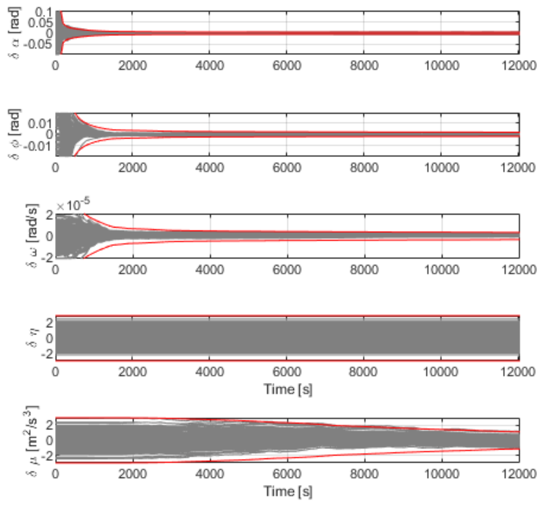
Figure 5.10: Parameter estimation errors and the filter-computed standard deviations compared with truth for the with-parameters case.

5.6.2 Relocalization

Continuing through the sandbox image sequence, the relocalization routine described in Section 5.3 begins working at image 122, at 122000 seconds, and continues processing images until image 180. During this process, new exploration features are still being initialized while relocaliza-



(a) Monte Carlo position errors and the filter-computed standard deviations compared with truth for the with-parameters case with underweighting.



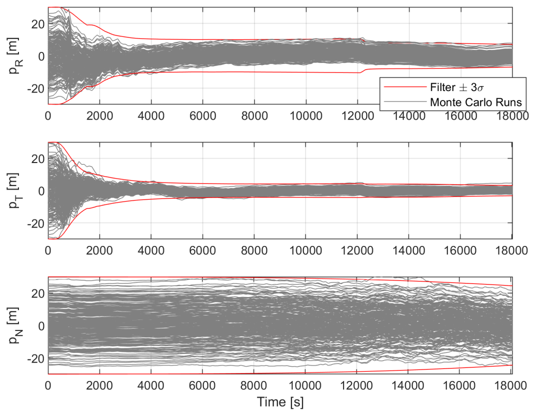
(b) Postfit residuals and measurement standard deviations.

Figure 5.11: Parameter estimation errors and the filter-computed standard deviations compared with truth for the with-parameters case with underweighting.

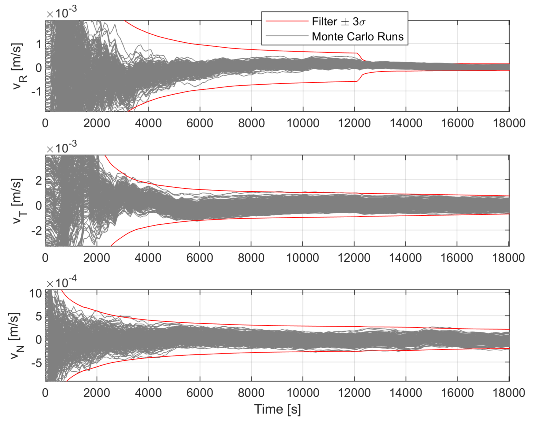
tion features are being considered. In Figures 5.12a-5.13b, we present the errors and average NEES results for this case. While we observe some unexplained behavior in the attitude states shown in Figure 5.13a, the position and velocity components seem well-behaved both before and during localization. This is reinforced by the NEES results in 5.13b.

5.7 Bennu, Rendered Images Simulation

With the same trajectory as the sandbox case, a sequence of 234 images were rendered using Blender, an open source 3D modelling and rendering tool, taking advantage of path-tracing techniques for increased photorealism. A similar pipeline was utilized in Reference [90]. The shape model used for rendering was a high-resolution Bennu model from the OSIRIS-REx mission and zero lens distortion was assumed. The albedo was assumed to be constant and equal to the overall average albedo of Bennu, about 4%. While this is unrealistic, albedo variations in real images would likely only increase the local variability of image regions and result in improved tracking

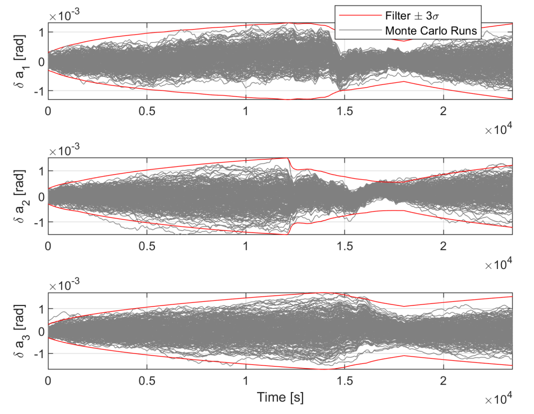


(a) Monte Carlo position errors and the filter-computed standard deviations compared with truth with relocalization.

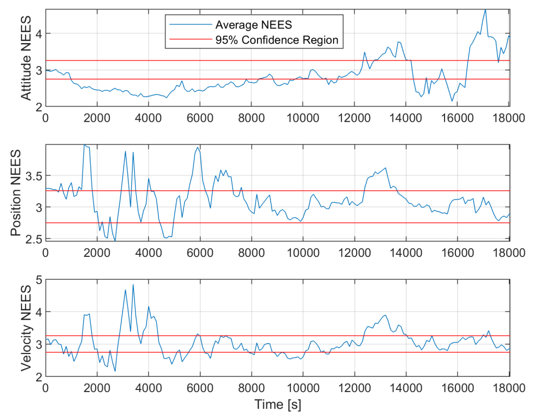


(b) Monte Carlo velocity errors and filter-computed standard devations with relocalization.

Figure 5.12: Results from 250 trials of the algorithm with a random perturbation on all state components with relocalization.



(a) Monte Carlo attitude errors and the filter-computed standard deviations compared with truth with relocalization.



(b) Average NEES for state components in sandbox case with relocalization.

Figure 5.13: Attitude errors and average NEES for the sandbox case from 250 trials of the algorithm with a random perturbation on all state components with relocalization.

performance. An example image used in the computations here is given in Figure 5.14.

For image processing, SIFT features[77] were extracted and matched using MATLAB's computer vision toolbox over the entire trajectory. A convex hull was computed over the asteroid

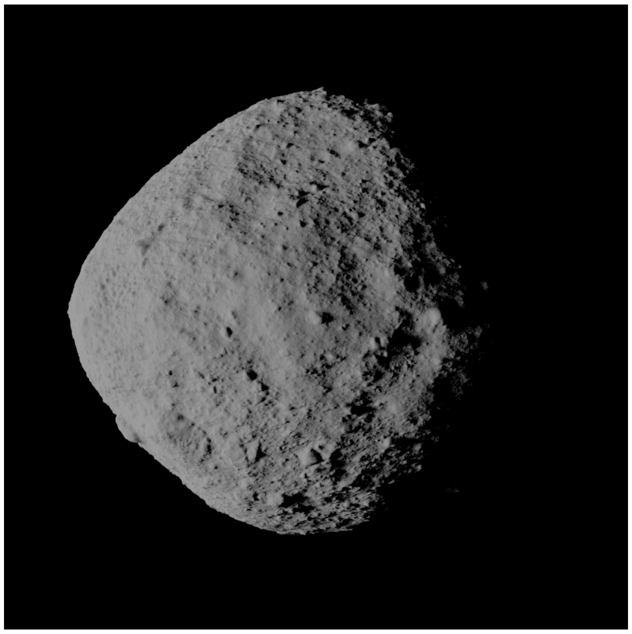
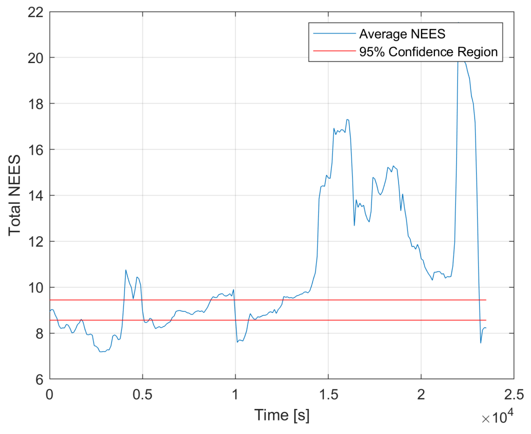


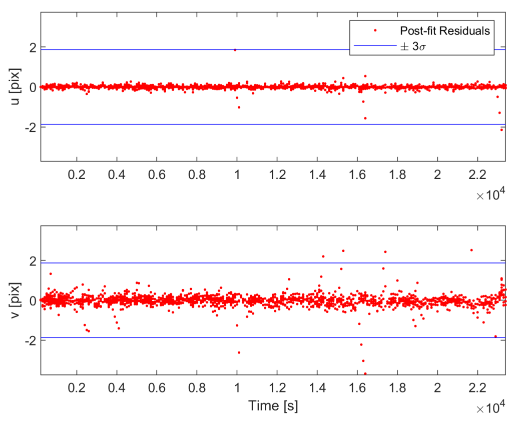
Figure 5.14: Image 10 of 234 rendered image sequence.

surface in each image so that a distance metric could be used to reject features too close to the limb of the asteroid. Given that the rendered images under consideration were generated using exactly the same trajectory as the sandbox images, and that the IIPA-SLAM algorithm under the hood is unchanged, the non-Gaussian structures appearing in the postfit residuals in Figure 5.15b are likely due discrepancies in the 3D location associated with the feature's projection in the image. This behavior is in line with results from recent work in small body feature tracking.[85] We attenuate this problem by underweighting the measurements for all time by setting $\sigma_m = 0.625$ pixels.

Despite the inconsistent feature tracks, we still obtain fairly consistent performance in terms of the average NEES as Figure 5.15a shows. To assess landmark estimation performance, we created a routine to compute the absolute error with respect to the reference shape model used in rendering the images. This works by finding the three closest vertices in the shape model to each estimated feature location, computing the outward normal vector of the associated triangle, and finding the distance between the triangle face and the given feature in that direction. These results are plotted in 3D in Figure 5.16 on top of the reference shape model and a histogram of the errors is provided



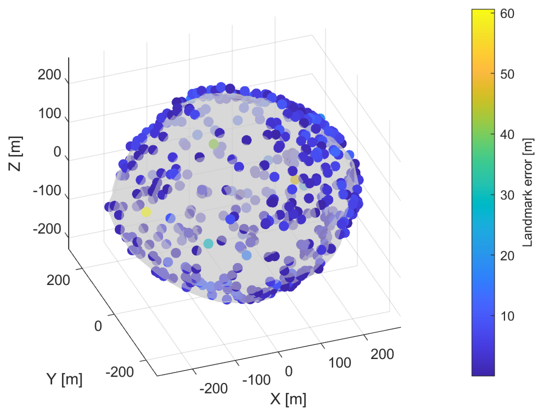
(a) Average NEES for simulated case.



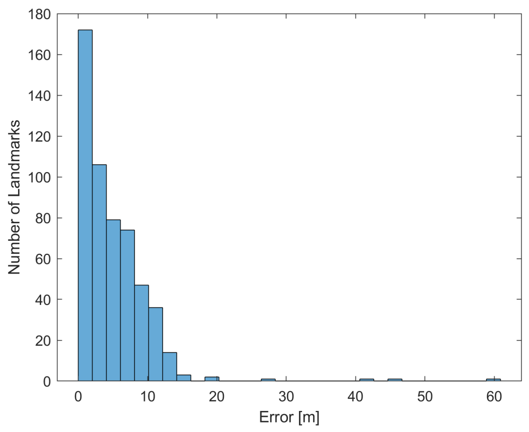
(b) Postfit residuals and measurement standard deviations for the simulated case.

Figure 5.15: Average NEES and postfit residuals of simulated Bennu case.

in Figure 5.16b.



(a) Landmark errors computed with respect to reference shape model.

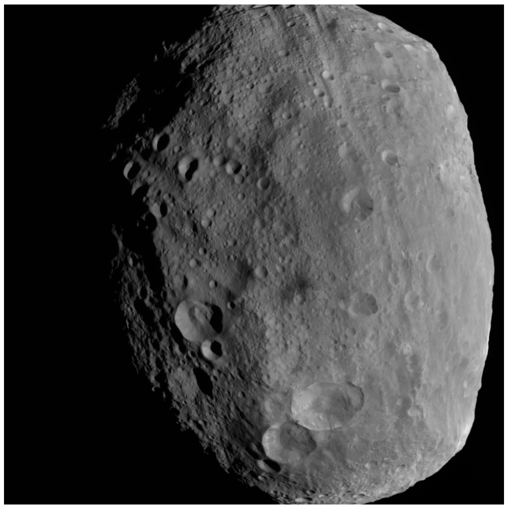


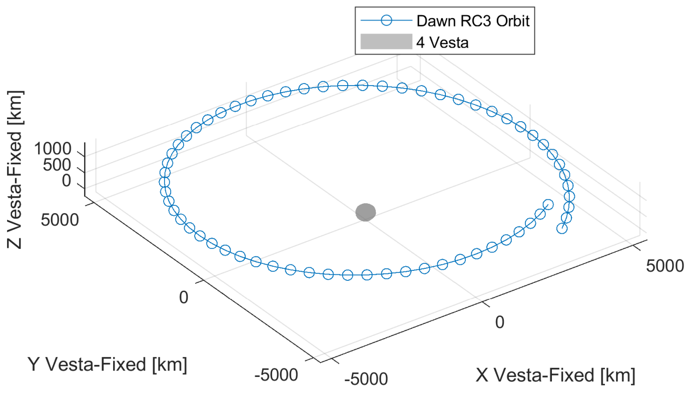
(b) Histogram of landmark errors in 5.16a.

Figure 5.16: Characteristic landmark estimation results in the simulated case.

5.8 4 Vesta, Real Data

Recently, Reference [34] presented the AstroSLAM adaptation of iSAM2 and applied it to publicly-available image data of the asteroid 4 Vesta that was captured during NASA's Dawn mission. In order to assess the performance of IIPA-SLAM in comparison to this incremental batch approach, we utilize the same data and present the results here. The Vesta RC3 phase of the mission, which put the spacecraft in a nearly-circular 5600 km orbit around the 500 km wide asteroid, yielded groups of images of the body separated by a nominal cadence of 0.033 seconds. Using the SPICE toolkit and the time tags of these images, we can visualize the reconstructed RC3 orbit in Figure 5.17b. An example image of this sequence is given in Figure 5.17a.


(a) First image from Dawn/Vesta sequence.



(b) Dawn's trajectory as reported in mission SPICE kernels.

Figure 5.17: Dawn's visit to Vesta.

We again model the spacecraft dynamics in a Vesta-fixed rotating frame. The third body influences of the Sun and Jupiter are included as well as the SRP following the models in Section 3.1. Numerical integration with an RK5 fixed-step integrator using an initial condition provided by the SPICE kernels shows a deviation of about 50 m and 0.5 cm/s over the 5.25 hour time span of the trajectory. The provided images are already calibrated, so no extra distortion correction is necessary. SIFT features are extracted and matched over the image set in the same way as in the previous simulated case. The remaining parameters used to generate the results are provided in Table 5.2 where we note that the prescribed initial position and attitude uncertainties are larger than those used in the AstroSLAM paper.

Table 5.2: Dawn/Vesta Scenario Parameters

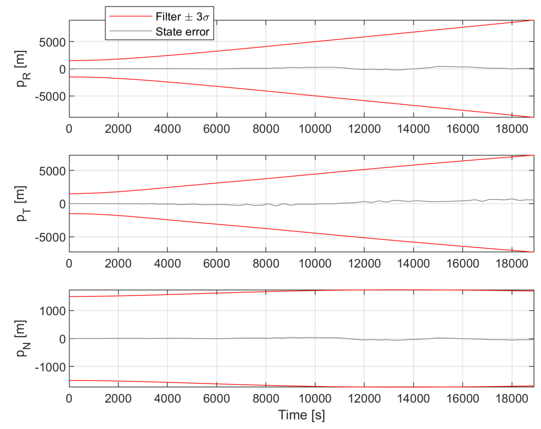
Parameter	**Symbol**	**Value**	**Units**
Pixel noise standard dev.	σ_η	0.2	pixels
Camera focal length	f	150	mm
Camera pixel pitch	d_x, d_y	13.98	μm
Camera field of view	FOV	5.47	degrees
Image size (square)	n_{pix}	1024	pixels
Camera frame rate	ω_c	0.0033	Hz
Maximum number of landmarks tracked simultaneously	n_{max}	20	-
Initial inverse depth standard dev.	σ_ρ	$5\rho_i$	1/m
Angular random walk	σ_ω	0.05	rad/$\sqrt{\text{s}}$
Initial vehicle position standard dev.	σ_p	500	m
Initial vehicle velocity standard dev.	σ_v	1	cm/s
Initial vehicle attitude standard dev.	σ_θ	20	arcsec

The results from processing this dataset with IIPA-SLAM are given in Figures 5.18 through 5.20. The position and velocity uncertainty of the spacecraft, which is captured by the $\pm 3\sigma$ lines in these figures, is not improved much by image measurements but that small corrections are largely consistent with respect to the SPICE reference. The results from AstroSLAM are very similar to the results presented here except IIPA-SLAM seemingly shows much better performance in the tangential (along-track) direction where AstroSLAM gives errors as large as 40 kilometers. IIPA-SLAM never deviates more than 500 meters in the tangential direction. Additionally, the 2-norm of IIPA-SLAM attitude error estimate never exceeds 0.03 degrees from the SPICE kernel truth, significantly less than AstroSLAM's 0.5 degrees for the same metric.

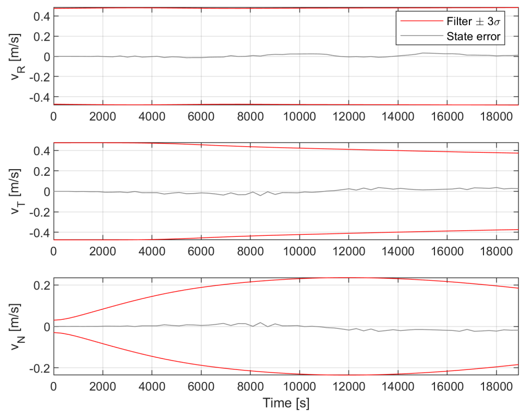
The landmark estimates in Figure 5.20, again computed with reference to the publicly-available shape model, are comparable with the AstroSLAM[34] results, with landmark errors distributed between 1-10 kilometers and biased toward errors of less than 2 kilometers. While the kilometer-scale landmark errors may seem high at first glance, a 9 kilometer error in landmark position represents less than 0.2 percent of the altitude of the spacecraft from the asteroid in this scenario. This is comparable or better than the 10-meter scale landmark errors in the Bennu case.

While we have not shown state error results for a full Monte Carlo simulation here, we note that we see robust and consistent performance when doing so. This can be seen in Figure 5.21

alongside the landmark errors with their uncertainty.

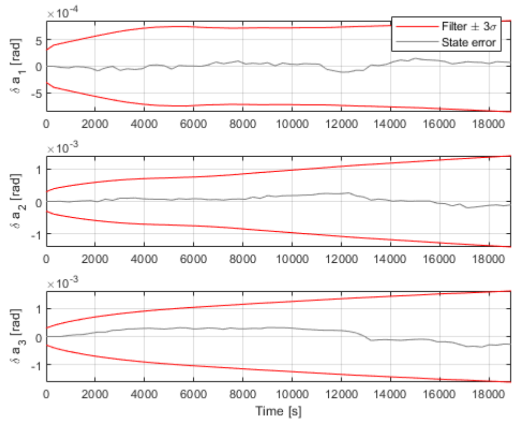


(a) Position errors and the filter-computed standard deviations in spacecraft RTN frame.

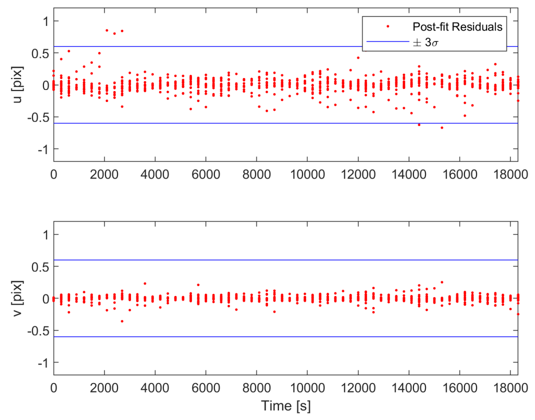


(b) Velocity errors and filter-computed standard devations in spacecraft RTN frame.

Figure 5.18: Position and velocity results for the Dawn/Vesta case.

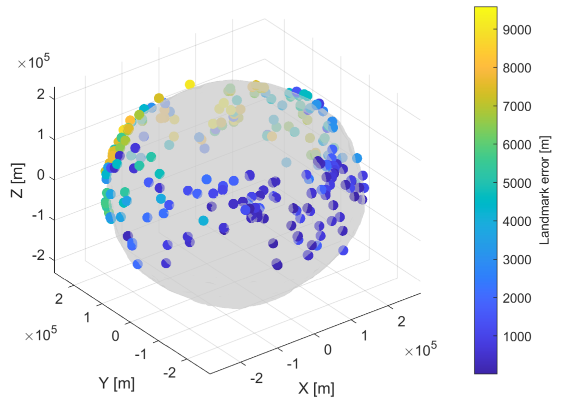


(a) Attitude errors and the filter-computed standard deviations.

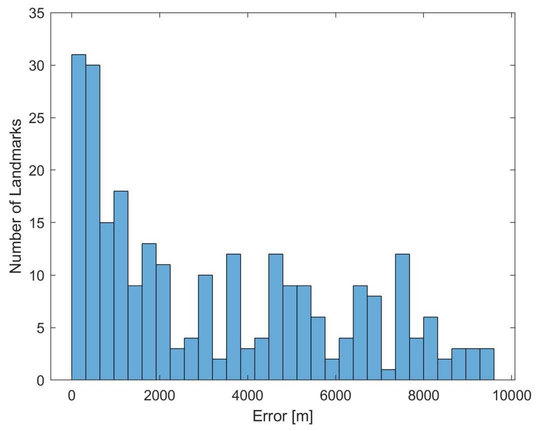


(b) Postfit residuals and measurement standard deviations.

Figure 5.19: Attitude results and postfit residuals for the Dawn/Vesta case.

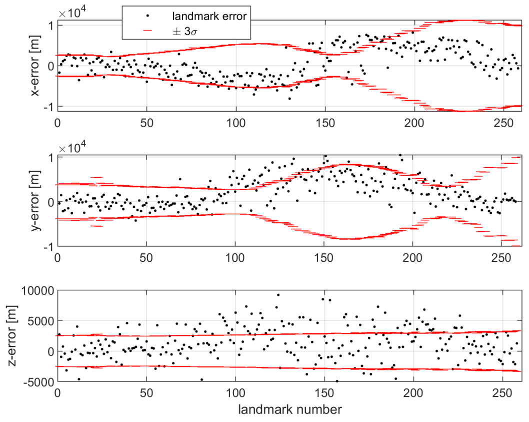


(a) Landmark errors computed with respect to reference shape model.

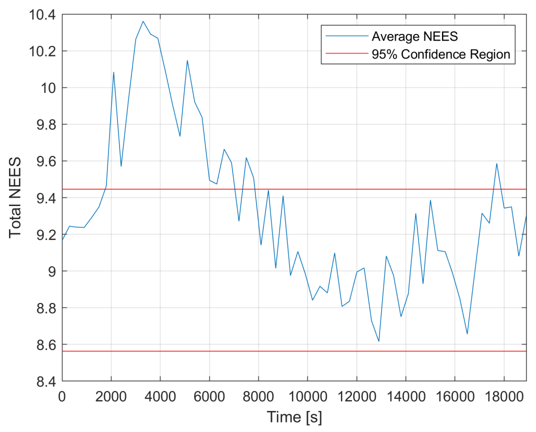


(b) Histogram of landmark errors in 5.16a.

Figure 5.20: Landmark estimation results in the Dawn/Vesta case.



(a) Landmark errors and filter-computed standard deviations computed with respect to reference shape model for a random Monte Carlo trial.



(b) Average NEES for Vesta case with 250 Monte Carlo trials.

Figure 5.21: Landmark errors and average NEES of Vesta case.

Chapter 6

Conclusion

In this work, we have offered a new perspective on Simultaneous Localization and Mapping (SLAM) in spaceflight and focused on understanding the underlying backend estimation techniques that have largely been investigated previously for terrestrial landmark-based SLAM. Combining insights from the sparse extended information filter (SEIF) family of algorithms and older factorized filtering concepts from space navigation, we developed a square root extended information filter (SREIF) for visual-inertial odometry that utilized the Unified Inverse Depth model to parameterize and estimate visual features. This new filter has constant time complexity under the assumptions that the number of features being tracked at one time and that the track length associated with each feature are bounded by constants. The SREIF was mechanized with an inertial measurement unit (IMU) in a model replacement mode and applied to a simulated planetary landing scenario.

Building on the SREIF, we presented an improved algorithm dubbed Inverse Incremental Pose Augmentation SLAM (IIPA-SLAM) that included a novel inverse depth measurement model, called Anchored Homogenous Bundles (AHB), as well as a new, computationally-efficient mean correction routine and a Schmidt-based relocalization algorithm. The AHB model adds two extra states per feature and requires that an anchor pose be retained for each group of features instead of just an anchor position. The benefit of this approach is that the direction to each feature can be estimated and the initialization of features in the state vector is trivial and can be delayed by one observation.

The mean reconciliation routine allows the best estimate of the posterior mean to be propagated backward to all prior poses and landmarks, improving the overall consistency of the map for little more than the computational cost of an upper triangular inversion. Since the IIPA-SLAM posterior with mean reconciliation, which is constructed by augmenting poses associated with measurements over time and never explicitly marginalizing them out, represents the same posterior that would be obtained from an Extended Kalman Filter (EKF) over the same state space, it may be useful for other estimation problems. Given new measurement information processed at the most-recent states, the *entire* posterior distribution is updated immediately with no extra computational effort.

Finally, the Schmidt-based relocalization algorithm allows the estimator to consider past landmark estimates without directly updating them or correlating them to the most recent state estimates, preserving the sparsity of the square root information matrix and allowing for faster relocalization when landmarks are re-encountered.

We applied IIPA-SLAM to both simulated and real asteroid datasets, obtaining largely consistent estimation performance when only estimating attitude, position, and velocity in the state vector. On real data taken by the Dawn spacecraft, the solution was comparable with other SLAM solutions with the same data. When asteroid parameters were included in the estimation routine, we obtained biased results that could be attenuated with underweighting. In another dataset with rendered images, we also observed deficiencies in the feature tracking performance of the Scale Invariant Feature Transform (SIFT) algorithm, resulting in non-Gaussian residuals.

SLAM is one of the key enabling technologies for future autonomous space exploration. This domain of human and robotic exploration offers new challenges to the methods and models for SLAM that have been developed over the last few decades by ground and aerial roboticists, including highly-nonlinear and detailed dynamics models, lack of known control inputs, large distances between vehicle and target, severe computational resource scarcity, and harsh lighting conditions. These and other problems will be solved by researchers and missions of exploration in the solar system.

Bibliography

[1] Numerical Methods for Solving Large Scale Eigenvalue Problems, Chapter 4.

[2] Vincent J. Aidala. Kalman Filter Behavior in Bearings-Only Tracking Applications. IEEE Transactions on Aerospace and Electronic Systems, AES-15(1):29–39, 1979.

[3] Nikolaus Ammann and Laura Garcia Mayo. Undelayed initialization of inverse depth parameterized landmarks in UKF-SLAM with error state formulation. IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, 2018-July:918–923, 2018.

[4] Sean Anderson, Timothy D. Barfoot, Chi Hay Tong, and Simo Särkkä. Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression. Autonomous Robots, 39(3):221–238, 2015.

[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] Sean Augenstein. Monocular Pose and Shape Estimation of Moving Targets, for Autonomous Rendezvous and Docking. PhD thesis, Stanford University, 2011.

[7] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. IEEE Robotics and Automation Magazine, 13(3):108–117, 2006.

[8] Francesca Baldini, Alexei Harvard, Soon Jo Chung, Issa Nesnas, and Shyamkumar Bhaskaran. Autonomous small body mapping and spacecraft navigation. In Proceedings of the International Astronautical Congress, IAC, number October, pages 1–5, 2018.

[9] Yaakov Bar-Shalom, X.-Rong Li, and Thiagalingam Kirubarajan. Estimation with Applications to Tracking and Navigation. Estimation with Applications to Tracking and Navigation, 2001.

[10] 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.

[11] Benjamin Bercovici and Jay W. McMahon. Robust autonomous small-body shape reconstruction and relative navigation using range images. Journal of Guidance, Control, and Dynamics, 42(7):1473–1488, 2019.

[12] Gerald J. Bierman. Factorization methods for discrete sequential estimation. Academic Press, 1977.

[13] Michael; Bloesch, Michael; Burri, Sammy; Omari, Marco; Hutter, and Roland Siegwart. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. The International Journal of Robotics Research, 36(10):1053–1072, 2017.

[14] Fredrik C. Bruhn, Nandinbaatar Tsog, Fabian Kunkel, Oskar Flordal, and Ian Troxel. Enabling radiation tolerant heterogeneous GPU-based onboard data processing in space. CEAS Space Journal, 12(4):551–564, 2020.

[15] Vincenzo Capuano, Kyunam Kim, Alexei Harvard, and Soon Jo Chung. Monocular-based pose determination of uncooperative space objects. Acta Astronautica, 166(September 2019):493–506, 2020.

[16] J Russell Carpenter and Christopher N D'souza. Navigation Filter Best Practices. Technical Report April, NASA, 2018.

[17] Chang Chen, Hua Zhu, Menggang Li, and Shaoze You. A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives. Robotics, 7(3), 2018.

[18] Chee Yee Chong, Kuo Chu Chang, and Shozo Mori. A Review of Forty Years of Distributed Estimation. In 2018 21st International Conference on Information Fusion, FUSION 2018, pages 70–77. ISIF, 2018.

[19] John A. Christian. A Tutorial on Horizon-Based Optical Navigation and Attitude Determination with Space Imaging Systems. IEEE Access, 9:19819–19853, 2021.

[20] John A. Christian, Lylia Benhacine, Jacob Hikes, and Christopher D'Souza. Geometric Calibration of the Orion Optical Navigation Camera using Star Field Images. Journal of the Astronautical Sciences, 63(4):335–353, 2016.

[21] John A. Christian, Lillian Hong, Paul McKee, Randall Christensen, and Timothy P. Crain. Image-Based Lunar Terrain Relative Navigation Without a Map: Measurements. Journal of Spacecraft and Rockets, pages 1–18, 2020.

[22] Javier Civera, Andrew J. Davison, and Jose Maria Martinez Montiel. Structure from Motion Using the Extended Kalman Filter, volume 75. 2012.

[23] C Cocaud and Takashi Kubota. SURF-Based SLAM Scheme using Octree Occupancy Grid for Autonomous Landing on Asteroids. In I-Sairis 2010, pages 275–282, 2010.

[24] Cedric Cocaud and Takashi Kubota. Autonomous Navigation Near Asteroid Based on Visual SLAM, 2009.

[25] Cedric Cocaud and Takashi Kubota. SLAM-based navigation scheme for pinpoint landing on small celestial body. Advanced Robotics, 26(15):1747–1770, 2012.

[26] Fred Daum. Nonlinear filters: Beyond the kalman filter. IEEE Aerospace and Electronic Systems Magazine, 20(8 II):57–68, 2005.

[27] Fred Daum and Jim Huang. Curse of dimensionality and particle filters. IEEE Aerospace Conference Proceedings, 4:1979–1993, 2003.

[28] Timothy A. Davis. Direct Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, Philadelphia, 2006.

[29] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Oliver Stasse. MonoSLAM: Real-Time Single Camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6):1052–1067, 2007.

[30] Frank Dellaert and Michael Kaess. Square Root SAM. International Journal of Robotics Research, 25(12):1181–1203, 2006.

[31] Frank Dellaert and Michael Kaess. Factor Graphs for Robot Perception. Foundations and Trends in Robotics, 6(1-2):1–139, 2017.

[32] Michel Delpech, Vincent Bissonnette, and Laurent Rastel. Vision Based Navigation For Proximity Operations Around Asteroid 99942 Apophis. In 25th International Symposium on Space Flight Dynamics ISSFD, number 1, 2015.

[33] P. Denchev, P. Magnusson, and Z. Donchev. Lightcurves of nine asteroids, with pole and sense of rotation of 42 Isis. Planetary and Space Science, 46(6-7):673–682, 1998.

[34] Mehregan Dor, Travis Driver, Kenneth Getzandanner, and Panagiotis Tsiotras. AstroSLAM: Autonomous Monocular Navigation in the Vicinity of a Celestial Small Body. Computer Vision and Pattern Recognition, 2022.

[35] Mehregan Dor, Katherine A. Skinner, Panagiotis Tsiotras, and Travis Driver. Visual SLAM for asteroid relative navigation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pages 2066–2075, 2021.

[36] Mehregan Dor and Panagiotis Tsiotras. ORB-SLAM applied to spacecraft non-cooperative rendezvous. In Space Flight Mechanics Meeting, 2018, number 210009, 2018.

[37] Travis Driver, Mehregan Dor, Katherine A. Skinner, and Panagiotis Tsiotras. Space Carving in Space: a Visual-Slam Approach To 3D Shape Reconstruction of a Small Celestial Body. 2021.

[38] Hugh Durrant-Whyte and Tim Bailey. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. IEEE Robotics & Automation Magazine, 13(2):99–110, 2006.

[39] Kawsihen Elankumaran and Andrew G. Dempster. Probabilistic approach to the autonomous navigation of distributed spacecraft near small celestial bodies. Acta Astronautica, 182(February):517–530, 2021.

[40] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(3):611–625, 2018.

[41] Jakob Engel, Jurgen Sturm, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In Proceedings of the IEEE International Conference on Computer Vision, pages 1449–1456, 2013.

[42] Carlos Estrada, José Neira, and Juan D. Tardós. Hierarchical SLAM: Real-time accurate mapping of large environments. IEEE Transactions on Robotics, 21(4):588–596, aug 2005.

[43] Ryan Eustice, Matthew Walter, and John Leonard. Sparse Extended Information Filters: Insights into Sparsification. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3281–3288, 2005.

[44] Ryan M. Eustice, Hanumant Singh, and John J. Leonard. Exactly sparse delayed-state filters. In Proceedings - IEEE International Conference on Robotics and Automation, volume 2005, pages 2417–2424, 2005.

[45] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. IEEE Transactions on Robotics, 33(1):1–21, 2016.

[46] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. IEEE Transactions on Robotics, 33(2):249–265, 2017.

[47] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS : A Research Platform for Visual-Inertial Estimation. In 2020 IEEE International Conference on Robotics and Automation, pages 4666–4672, Paris, France, 2020.

[48] Matthew Givens. Multiple IMU Sensor Fusion for SUAS Navigation and Photogrammetry. PhD thesis, Utah State University, 2019.

[49] Matthew W Givens and Jay W Mcmahon. Nearly Constant-Time SLAM-Based Terrain Relative Navigation for Landing on an Uncharted World. In 2021 American Astronautical Society Spaceflight Mechanics Meeting, Virtual, 2021.

[50] Matthew W Givens and Jay W Mcmahon. Square Root, Sequential Visual Odometry for Constant-Time Navigation and Mapping. In AIAA SCITECH 2022 Forum, San Diego, CA, jan 2022. American Institute of Aeronautics and Astronautics.

[51] Matthew W Givens, Jacopo Villa, and Jay W Mcmahon. Computationally-Efficient Visual-Inertial SLAM for Asteroid-Relative Navigation. In 2023 AAS Astrodynamics Specialist Conference, Austin, 2023.

[52] Chris R. Gnam, Timothy B. Chase, Karthik Dantu, and John L. Crassidis. Efficient Feature Matching and Mapping for Terrain Relative Navigation Using Hypothesis Gating. In AIAA SciTech Forum 2022, pages 1–13, 2022.

[53] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based SLAM. IEEE Intelligent Transportation Systems Magazine, 2(4):31–43, 2010.

[54] Richard Hartley and Andrew Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, second edition, 2004.

[55] Sébastien Henry and John A. Christian. Absolute Triangulation Algorithms for Space Exploration. Journal of Guidance, Control, and Dynamics, 46(1):21–46, 2023.

[56] Steven Holmes, Georg Klein, and David W. Murray. A Square Root Unscented Kalman Filter for visual monoSLAM. In Proceedings - IEEE International Conference on Robotics and Automation, pages 3710–3716, 2008.

[57] Zheng Huai and Guoquan Huang. Robocentric visual–inertial odometry. The International Journal of Robotics Research, jul 2019.

[58] Guoquan Huang, Michael Kaess, and John J. Leonard. Towards consistent visual-inertial navigation. Proceedings - IEEE International Conference on Robotics and Automation, pages 4926–4933, 2014.

[59] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. A First-Estimates Jacobian EKF for Improving SLAM Consistency. Springer Tracts in Advanced Robotics, 54:373–382, 2009.

[60] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Observability-based Rules for Designing Consistent EKF SLAM Estimators:. The International Journal of Robotics Research, 29(5):502–528, dec 2010.

[61] E. Imre, M. O. Berger, and N. Noury. Improved inverse-depth parameterization for monocular simultaneous localization and mapping. Proceedings - IEEE International Conference on Robotics and Automation, pages 381–386, 2009.

[62] Danil Ivanov, Mikhail Ovchinnikov, and Marianna Sakovich. Relative pose and inertia determination of unknown satellite using monocular vision. International Journal of Aerospace Engineering, 2018, 2018.

[63] Zeming Jin, Ling Wang, Hanhan Liu, Ronghua Du, and Xiang Zhang. Monocular-Based Pose Estimation of Non-Cooperative Space Targets Using EKF and EKPF. In Proceedings - 2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2020, pages 46–51, 2020.

[64] Andrew Johnson, Reg Willson, Yang Cheng, Jay Goguen, Chris Leger, Miguel Sanmartin, and Larry Matthies. Design through operation of an image-based velocity estimation system for mars landing. International Journal of Computer Vision, 74(3):319–341, 2007.

[65] Andrew E. Johnson and James F. Montgomery. Overview of Terrain Relative Navigation approaches for precise lunar landing. In IEEE Aerospace Conference Proceedings, 2008.

[66] S. J. Julier and J. K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In IEEE International Conference on Robotics and Automation, volume 4, pages 4238–4243, 2001.

[67] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J. Leonard, and Frank Dellaert. ISAM2: Incremental smoothing and mapping using the Bayes tree. International Journal of Robotics Research, 31(2):216–235, 2012.

[68] Tong Ke, Kejian J. Wu, and Stergios I. Roumeliotis. RISE-SLAM: A Resource-aware Inverse Schmidt Estimator for SLAM. In IEEE International Conference on Intelligent Robots and Systems, pages 354–361. Institute of Electrical and Electronics Engineers Inc., nov 2019.

[69] Scott J. Kelly. A Monocular SLAM method to Estimate Relative Pose During Satellite Proximity Operations. PhD thesis, United States Air Force Institute of Technology, 2015.

[70] B Kevers. Monocular-based Pose Estimation Of Unknown Uncooperative Targets. PhD thesis, Delft University of Technology, 2021.

[71] Kevin R Kobylka. Photometric and geometric methods for enhanced image-based spacecraft navigation. PhD thesis, Rensselaer Polytechnic Institute, 2021.

[72] Dimitrios G. Kottas, Joel A. Hesch, Sean L. Bowman, and Stergios I. Roumeliotis. On the Consistency of Vision-Aided Inertial Navigation. In Experimental Robotics: The 13th International Symposium on Experimental Robotics, pages 303–317, Heidelberg, 2013. Springer International Publishing.

[73] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In Proceedings - IEEE International Conference on Robotics and Automation, pages 3607–3613, 2011.

[74] Ting Lei, Xiao Feng Liu, Guo Ping Cai, Yun Meng Liu, and Pan Liu. Pose estimation of a noncooperative target based on monocular visual SLAM. International Journal of Aerospace Engineering, 2019, 2019.

[75] Mingyang Li and Anastasios I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry:. The International Journal of Robotics Research, 32(6):690–711, jun 2013.

[76] Matthew D. Lichter and Steven Dubowsky. State, shape, and parameter estimation of space objects from range images. In Proceedings - IEEE International Conference on Robotics and Automation, volume 2004, pages 2974–2979, 2004.

[77] David G Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, pages 91–110, 2004.

[78] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. Autonomous Robots, 4(4):333–349, 1997.

[79] Corey L. Marcus, Timothy P. Setterfield, Robert A. Hewitt, and Po Ting Chen. Landing Site Selection with a Variable-Resolution SLAM-Refined Map. In IEEE Aerospace Conference Proceedings, pages 1–12. IEEE, 2022.

[80] Peter Maybeck. Stochastic Models, Estimation and Control, Volume 2. Elsevier, Amsterdam, 1979.

[81] James Samuel McCabe. Multitarget Tracking and Terrain-Aided Navigation Using Square-Root Consider Filters. PhD thesis, 2018.

[82] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In Eighteenth national conference on Artificial intelligence, pages 593–598, 2002.

[83] Oliver Montenbruck and Eberhard Gill. Satellite Orbits. Springer-Verlag, Berlin, 4th printi edition, 2012.

[84] J. M.M. Montiel, Javier Civera, and Andrew J. Davison. Unified inverse depth parametrization for monocular SLAM. Robotics: Science and Systems, 2:81–88, 2007.

[85] Benjamin Morrell, Jacopo Villa, Saptarshi Bandyopadhyay, Daniel Lubey, and Benjamin Hockman. Automatic Feature Tracking on Small Bodies for Autonomous Approach. In ASCEND, 2020.

[86] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In Proceedings - IEEE International Conference on Robotics and Automation, pages 3565–3572, 2007.

[87] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. ORB-SLAM: a Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics, 31(5):1147–1163, feb 2015.

[88] David Nakath, Joachim Clemens, and Carsten Rachuy. Active Asteroid-SLAM: Active Graph SLAM with Landing Site Discovery in a Deep Space Proximity Operations Scenario. Journal of Intelligent and Robotic Systems: Theory and Applications, 99(2):303–333, 2020.

[89] Esha D. Nerurkar and Stergios I. Roumeliotis. Power-SLAM: A linear-complexity, consistent algorithm for SLAM. IEEE International Conference on Intelligent Robots and Systems, pages 636–643, 2007.

[90] Issa A.D. Nesnas, Benjamin J. Hockman, Saptarshi Bandopadhyay, Benjamin J. Morrell, Daniel P. Lubey, Jacopo Villa, David S. Bayard, Alan Osmundson, Benjamin Jarvis, Michele Bersani, and Shyam Bhaskaran. Autonomous Exploration of Small Bodies Toward Greater Autonomy for Deep Space Missions. Frontiers in Robotics and AI, 8, nov 2021.

[91] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. Pose estimation for spacecraft relative navigation using model-based algorithms. IEEE Transactions on Aerospace and Electronic Systems, 53(1):431–447, 2017.

[92] Tobias Pietzsch. Efficient feature parameterisation for visual SLAM using inverse depth bundles. In BMVC 2008 - Proceedings of the British Machine Vision Conference 2008, 2008.

[93] Tobias Pietzsch. Towards Dense Visual SLAM. PhD thesis, 2010.

[94] Arunkumar Rathinam. Small Body Gravimetry using SLAM-based Autonomous Navigation. PhD thesis, University of New South Wales, 2019.

[95] Arunkumar Rathinam and Andrew G Dempster. 3D reconstruction of an asteroid shape using visual SLAM for autonomous navigation. In 17th Australian Space Research Conference (ASRC), pages 87–96, 2017.

[96] Arunkumar Rathinam and Andrew G. Dempster. Monocular vision based simultaneous localization and mapping for close proximity navigation near an asteroid. Proceedings of the International Astronautical Congress, IAC, 11(September):6986–6991, 2017.

[97] Arunkumar Rathinam and Andrew G. Dempster. Vision based state estimation using a graph-SLAM approach for proximity operations near an asteroid. Proceedings of the International Astronautical Congress, IAC, 2018-Octob:1–5, 2018.

[98] David M. Rosen, Kevin J. Doherty, Antonio Terán Espinoza, and John J. Leonard. Advances in Inference and Representation for Simultaneous Localization and Mapping. Annual Review of Control, Robotics, and Autonomous Systems, 4(1):215–242, 2021.

[99] Ethan Rublee, Willow Garage, and Menlo Park. ORB : an efficient alternative to SIFT or SURF. pages 2564–2571, 2011.

[100] Hanspeter Schaub and John L. Junkins. Analytical Mechanics of Space Systems, Fourth Edition. Analytical Mechanics of Space Systems, Fourth Edition, apr 2018.

[101] Lauren Schlenker. Spacecraft Relative Navigation Using Random Finite Sets. PhD thesis, University of Minnesota, 2019.

[102] Lauren Schlenker, Mark Moretto, David Gaylor, and Richard Linares. Simultaneous localization and mapping for satellite rendezvous and proximity operations using random finite sets. volume 168, pages 3401–3419, 2019.

[103] Frank Schnitzer, Klaus Janschek, and Georg Willich. Experimental results for image-based geometrical reconstruction for spacecraft Rendezvous navigation with unknown and uncooperative target spacecraft. In IEEE International Conference on Intelligent Robots and Systems, pages 5040–5045. IEEE, 2012.

[104] Timothy P. Setterfield, Robert A. Hewitt, Po Ting Chen, Antonio Teran Espinoza, Nikolas Trawny, and Anup Katake. LiDAR-Inertial Based Navigation and Mapping for Precision Landing. In IEEE Aerospace Conference Proceedings, 2021.

[105] Gabe Sibley. Long Range Stereo Data-Fusion From Moving Platforms. PhD thesis, University of Southern California, 2007.

[106] Joan Solà. Consistency of the monocular EKF-SLAM algorithm for three different landmark parametrizations. In Proceedings - IEEE International Conference on Robotics and Automation, number 1, pages 3513–3518, 2010.

[107] Joan Solà. Simulataneous localization and mapping with the extended Kalman filter 'A very quick guide... with Matlab code!'. Technical report, 2014.

[108] Joan Solà. Quaternion Kinematics for the Error-State Kalman Filter. Technical report, Barcelona, Spain, 2016.

[109] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics. pages 1–17, 2018.

[110] Boyi Song, Xianfeng Yuan, Zhongmou Ying, Baojiang Yang, Yong Song, and Fengyu Zhou. DGM-VINS : Visual-Inertial SLAM for Complex Dynamic Environments with Joint Geometry Feature Extraction and Multiple Object Tracking. IEEE Transactions on Instrumentation and Measurement, PP:1, 2023.

[111] Jianing Song, Duarte Rondao, and Nabil Aouf. Deep learning-based spacecraft relative navigation methods: A survey. Acta Astronautica, 191:22–40, 2022.

[112] Arne Sonnenburg, Marcel Tkocz, and Klaus Janschek. EKF-SLAM based approach for spacecraft rendezvous navigation with unknown target spacecraft. In IFAC Proceedings Volumes (IFAC-PapersOnline), volume 18, pages 339–344. IFAC, 2010.

[113] Ted J. Steiner. Utility-based Map Reduction for Ground and Flight Vehicle Navigation. PhD thesis, 2015.

[114] Ted J. Steiner, Tye M. Brady, and Jeffrey A. Hoffman. Graph-based terrain relative navigation with optimal landmark database selection. In IEEE Aerospace Conference. IEEE, 2015.

[115] Hauke Strasdat, J. M.M. Montiel, and Andrew J. Davison. Real-time monocular SLAM: Why filter? Proceedings - IEEE International Conference on Robotics and Automation, pages 2657–2664, 2010.

[116] Hauke Strasdat, J. M.M. Montiel, and Andrew J. Davison. Visual SLAM: Why filter? Image and Vision Computing, 30(2):65–77, feb 2012.

[117] Joshua Sullivan, Sebastian Grimberg, and Simone D'Amico. Comprehensive survey and assessment of spacecraft relative motion dynamics models. Journal of Guidance, Control, and Dynamics, 40(8):1837–1859, 2017.

[118] Naoya Takeishi, Takehisa Yairi, Yuichi Tsuda, Fuyuto Terui, Naoko Ogawa, and Yuya Mimasu. Simultaneous estimation of shape and motion of an asteroid for automatic navigation. In Proceedings - IEEE International Conference on Robotics and Automation, pages 2861–2866. IEEE, 2015.

[119] Byron D. Tapley, Bob E. Schutz, and George H. Born. Statistical Orbit Determination. 2004.

[120] Dylan Thomas, Scott Kelly, and Jonathan Black. A monocular SLAM method for satellite proximity operations. In Proceedings of the American Control Conference, volume 2016-July, pages 4035–4040. American Automatic Control Council (AACC), 2016.

[121] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. International Journal of Robotics Research, 23(7-8):693–716, 2004.

[122] Nikolas Trawny, A.I. Mourikis, S.I. Roumeliotis, A. Johnson, J. Montgomery, Adnan Ansar, and L. Matthies. Coupled Vision and Inertial Navigation for Pin-Point Landing. In NASA Science and Technology Conference, pages 1–10, 2007.

[123] Nikolas Trawny and Stergios I Roumeliotis. Indirect Kalman Filter for 3D Attitude Estimation. Technical report, University of Minnesota, Minneapolis, MN, 2005.

[124] Brent Edward Tweddle. Computer vision-based localization and mapping of an unknown, uncooperative and spinning target for spacecraft proximity operations. PhD thesis, Massachussetts Institute of Technology, 2013.

[125] Corinne Vassallo, Wennie Tabib, and Kevin Peterson. Orbital SLAM. In Proceedings -2015 12th Conference on Computer and Robot Vision, CRV 2015, pages 305–312. IEEE, 2015.

[126] Jacopo Villa and Jay W McMahon. Robust Landmark and Hazard Detection on Small Body Surfaces Using Shadow Imagery. In 2022 AAS Astrodynamics Specialist Conference, number August, Charlotte, NC, 2022.

[127] Matthew R Walter. Sparse Bayesian Information Filters for Localization and Mapping. PhD thesis, Massachusetts Institute of Technology, 2008.

[128] Drew P. Woodbury. Accounting for parameter uncertainty in reduced-order static and dynamic systems. PhD thesis, Texas A&M University, 2011.

[129] Kejian J. Wu, Ahmed M. Ahmed, Georgios A. Georgiou, and Stergios I. Roumeliotis. A square root inverse filter for efficient vision-aided inertial navigation on mobile devices. Robotics: Science and Systems, 11, 2015.

[130] Kejian J Wu, Stergios I Roumeliotis, Kejian J Wu, and Stergios I Roumeliotis. Inverse Schmidt Estimators Multiple Autonomous Technical Report Number -2016-003 Inverse Schmidt Estimators. Technical Report 612, 2016.

[131] Renato Zanetti, Kyle J. Demars, and Robert H. Bishop. Underweighting nonlinear measurements. Journal of Guidance, Control, and Dynamics, 33(5):1670–1675, 2010.

[132] Zhengyou Zhang. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330–1334, 2000.

Appendix A

Measurement Model Derivatives

The Anchored Homogenous Bundles model with a camera offset included is given by

$$s_i {}^C\mathbf{h}_i = {}^C\mathbf{h}_{s,i} = {}^C_M\mathbf{C}[s_i\rho_i({}^M\mathbf{p}_{j,c} - {}^M\mathbf{p}_c) + {}^M_{C_j}\mathbf{C}^{C_j}\mathbf{s}_i] \tag{A.1}$$

where, as described in Section 3.3.7,

$${}^M\mathbf{p}_{j,c} = {}^M\mathbf{p}_j + {}^M_{B_j}\mathbf{C}^{B_j}\mathbf{d}_c \tag{A.2}$$

$${}^M\mathbf{p}_c = {}^M\mathbf{p} + {}^M_B\mathbf{C}^B\mathbf{d}_c \tag{A.3}$$

The derivatives of A.1 with respect to the landmark and both anchor and current attitude, position, velocity, and parameter states must be explicit in order to be used in a linearized filter. Without the parameters, the state vector considered here is

$$\mathbf{x} = \begin{bmatrix} \rho_i \\ \mathbf{s}_i \\ {}^B_I q_j \\ {}^M\mathbf{p}_j \\ {}^M\mathbf{v}_j \\ {}^B_I q \\ {}^M\mathbf{p} \\ {}^M\mathbf{v} \end{bmatrix} \tag{A.4}$$

where ρ_i is the i-th feature's inverse depth, $\mathbf{s}_i$ are the coordinates of the homogenous vector from the j-th anchor to the i-th feature, ${}^B_I q$ is the inertial-to-spacecraft-body quaternion (JPL convention),

${}^{M}\mathbf{p}$ and ${}^{M}\mathbf{v}$ are the asteroid-fixed position and velocity. Anchor states are shown with a j subscript while the current state has no subscript.

Immediately, the position, velocity, and inverse depth derivates can be extracted by inspection:

$$\frac{\partial^{C}\mathbf{h}_{s,i}}{\partial^{M}\mathbf{p}_j} = {}^{C}_{M}\mathbf{C} s_i \rho_i \tag{A.5}$$

$$\frac{\partial^{C}\mathbf{h}_{s,i}}{\partial^{M}\mathbf{p}} = -{}^{C}_{M}\mathbf{C} s_i \rho_i \tag{A.6}$$

$$\frac{\partial^{C}\mathbf{h}_{s,i}}{\partial^{M}\mathbf{v}_j} = \mathbf{0}_{3\times 3} \tag{A.7}$$

$$\frac{\partial^{C}\mathbf{h}_{s,i}}{\partial^{M}\mathbf{v}} = \mathbf{0}_{3\times 3} \tag{A.8}$$

$$\frac{\partial^{C}\mathbf{h}_{s,i}}{\partial \rho_i} = {}^{C}_{M}\mathbf{C} s_i ({}^{M}\mathbf{p}_{j,c} - {}^{M}\mathbf{p}_c) = {}^{C}_{M}\mathbf{C} s_i \boldsymbol{\Delta}_1 \tag{A.9}$$

The homogenous vector can be differentiated component-wise to yield

$$\frac{\partial^{C}\mathbf{h}_{s,i}}{\partial s_{i,x}} = {}^{C}_{M}\mathbf{C}\left[\frac{s_{i,x}}{s_i}\rho_i \boldsymbol{\Delta}_1 + {}^{M}_{C_j}\mathbf{C}[1,0,0]^T\right] \tag{A.10}$$

$$\frac{\partial^{C}\mathbf{h}_{s,i}}{\partial s_{i,y}} = {}^{C}_{M}\mathbf{C}\left[\frac{s_{i,y}}{s_i}\rho_i \boldsymbol{\Delta}_1 + {}^{M}_{C_j}\mathbf{C}[0,1,0]^T\right] \tag{A.11}$$

For the attitude states, we utilize the error state definition from Section 3.2.1 about the nominal quaternion state and differentiate with respect to the error state $\delta\boldsymbol{\psi}$. Rewriting Equation A.1 to isolate terms that are directly affected by $\delta\boldsymbol{\psi}$ yields

$$ {}^{C}\mathbf{h}_{s,i} = {}^{C}_{B}\mathbf{C}{}^{B}_{M}\mathbf{C}[s_i\rho_i({}^{M}\mathbf{p}_j + {}^{M}_{B_j}\mathbf{C}^{B_j}\mathbf{d}_c - {}^{M}\mathbf{p} - {}^{M}_{B}\mathbf{C}^{B}\mathbf{d}_c) + {}^{M}_{C_j}\mathbf{C}^{C_j}\mathbf{s}_i] \tag{A.12}$$

$$ {}^{C}\mathbf{h}_{s,i} = {}^{C}_{B}\mathbf{C}\left[s_i\rho_i{}^{B}_{M}\mathbf{C}\left({}^{M}\mathbf{p}_j + {}^{M}_{B_j}\mathbf{C}^{B_j} + \frac{1}{s_i\rho_i}{}^{M}_{C_j}\mathbf{C}^{C_j}\mathbf{s}_i{}^{B_j}\mathbf{d}_c - {}^{M}\mathbf{p}\right) - {}^{B}\mathbf{d}_c\right] \tag{A.13}$$

Subsuming the terms inside the parentheses into a $\boldsymbol{\Delta}_2$ shorthand, we obtain an expression that can be differentiated more easily:

$$ {}^{C}\mathbf{h}_{s,i} = {}^{C}_{B}\mathbf{C}[s_i\rho_i{}^{B}_{M}\mathbf{C}\boldsymbol{\Delta}_2 - {}^{B}\mathbf{d}_c] \tag{A.14}$$

More explicitly,

$$ {}^{C}\mathbf{h}_{s,i} = {}^{C}_{B}\mathbf{C}[s_i\rho_i{}^{B}_{I}\mathbf{C}{}^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2 - {}^{B}\mathbf{d}_c] \tag{A.15}$$

The derivative will only affect terms that interact with ${}^{B}_{I}\mathbf{C}$, so

$$\frac{\partial {}^{C}\mathbf{h}_{s,i}}{\partial \delta\boldsymbol{\psi}} = \frac{\partial}{\partial \delta\boldsymbol{\psi}} {}^{C}_{B}\mathbf{C}[s_i \rho_i {}^{B}_{I}\mathbf{C}^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2] \tag{A.16}$$

The full inertial-to-body rotation can be re-interpreted as

$${}^{B}_{I}\mathbf{C}^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2 = {}^{B}_{\hat{I}}\mathbf{C}^{\hat{I}}_{I}\mathbf{C}^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2 \approx {}^{B}_{\hat{I}}\mathbf{C}(\mathbf{I}_3 + [\delta\boldsymbol{\psi}\times])^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2 \tag{A.17}$$

by utilizing the approximation from Equation 3.16. Distributing and reversing the cross product yields

$${}^{B}_{\hat{I}}\mathbf{C}(\mathbf{I}_3 + [\delta\boldsymbol{\psi}\times])^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2 = {}^{B}_{\hat{I}}\mathbf{C}^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2 - {}^{B}_{\hat{I}}\mathbf{C}[{}^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2\times]\delta\boldsymbol{\psi} \tag{A.18}$$

so that the desired derivative is, finally,

$$\frac{\partial {}^{C}\mathbf{h}_{s,i}}{\partial \delta\boldsymbol{\psi}} = -{}^{C}_{B}\mathbf{C}^{B}_{\hat{I}}\mathbf{C}[{}^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2\times] \tag{A.19}$$

In a similar way, the derivative with respect to the anchor attitude $\delta\boldsymbol{\psi}_j$ can be derived and comes out to

$$\frac{\partial {}^{C}\mathbf{h}_{s,i}}{\partial \delta\boldsymbol{\psi}_j} = {}^{C}_{B}\mathbf{C}^{B}_{M}\mathbf{C}^{M}_{I_j}\mathbf{C}[{}^{B_j}_{\hat{I}_j}\mathbf{C}\boldsymbol{\Delta}_3\times] \tag{A.20}$$

where

$$\boldsymbol{\Delta}_3 = s_i \rho_i {}^{B_j}\mathbf{d}_c - {}^{B_j}_{C_j}\mathbf{C}\mathbf{s}_i \tag{A.21}$$

Chapter 5 also estimates asteroid right ascension α and declination ϕ with respect to inertial, asteroid rotation rate ω_a, gravitational parameter μ, and SRP scale factor η,

$$\mathbf{x}_p = \begin{bmatrix} \alpha \\ \phi \\ \omega_a \\ \eta \\ \mu \end{bmatrix} \tag{A.22}$$

The same logic applies as for the attitude derivatives,

$$\frac{\partial {}^{C}\mathbf{h}_{s,i}}{\partial \alpha} = \frac{\partial}{\partial \alpha}\left[{}^{B}_{I}\mathbf{C}^{I}_{M}\mathbf{C}\boldsymbol{\Delta}_2\right] = {}^{B}_{I}\mathbf{C}\frac{\partial}{\partial \alpha}\left[{}^{I}_{M}\mathbf{C}\right]\boldsymbol{\Delta}_2 \tag{A.23}$$

$$\frac{\partial^C \mathbf{h}_{s,i}}{\partial \phi} = \frac{\partial}{\partial \phi}\left[{}^B_I\mathbf{C}{}^I_M\mathbf{C}\boldsymbol{\Delta}_2\right] = {}^B_I\mathbf{C}\frac{\partial}{\partial \phi}\left[{}^I_M\mathbf{C}\right]\boldsymbol{\Delta}_2 \tag{A.24}$$

$$\frac{\partial^C \mathbf{h}_{s,i}}{\partial \omega_a} = \frac{\partial}{\partial \omega_a}\left[{}^B_I\mathbf{C}{}^I_M\mathbf{C}\boldsymbol{\Delta}_2\right] = {}^B_I\mathbf{C}\frac{\partial}{\partial \omega_a}\left[{}^I_M\mathbf{C}\right]\boldsymbol{\Delta}_2 \tag{A.25}$$

where the unresolved derivatives are the transposes of the ones derived for the dynamics. For the anchor pole parameters,

$$\frac{\partial^C \mathbf{h}_{s,i}}{\partial \alpha_j} = \frac{\partial}{\partial \alpha_j}\left[{}^B_M\mathbf{C}{}^M_{I_j}\mathbf{C}{}^{I_j}_{B_j}\mathbf{C}\boldsymbol{\Delta}_3\right] = {}^B_M\mathbf{C}\frac{\partial}{\partial \alpha_j}\left[{}^M_{I_j}\mathbf{C}\right]{}^{I_j}_{B_j}\mathbf{C}\boldsymbol{\Delta}_3 \tag{A.26}$$

$$\frac{\partial^C \mathbf{h}_{s,i}}{\partial \phi_j} = \frac{\partial}{\partial \phi_j}\left[{}^B_M\mathbf{C}{}^M_{I_j}\mathbf{C}{}^{I_j}_{B_j}\mathbf{C}\boldsymbol{\Delta}_3\right] = {}^B_M\mathbf{C}\frac{\partial}{\partial \phi_j}\left[{}^M_{I_j}\mathbf{C}_0\right]{}^{I_j}_{B_j}\mathbf{C}_0\boldsymbol{\Delta}_3 \tag{A.27}$$

$$\frac{\partial^C \mathbf{h}_{s,i}}{\partial \omega_{a,j}} = \frac{\partial}{\partial \omega_{a,j}}\left[{}^B_M\mathbf{C}{}^M_{I_j}\mathbf{C}{}^{I_j}_{B_j}\mathbf{C}\boldsymbol{\Delta}_3\right] = {}^B_M\mathbf{C}\frac{\partial}{\partial \omega_{a,j}}\left[{}^M_{I_j}\mathbf{C}\right]{}^{I_j}_{B_j}\mathbf{C}\boldsymbol{\Delta}_3 \tag{A.28}$$

The constituent heretofore unresolved derivatives are

$$\frac{\partial}{\partial \alpha}{}^M_I\mathbf{C} = \mathbf{C}_\theta\mathbf{C}_\phi\frac{\partial}{\partial \alpha}\mathbf{C}_\alpha = \mathbf{C}_\theta\mathbf{C}_\phi\begin{bmatrix} -\sin(90^o+\alpha) & \cos(90^o+\alpha) & 0 \\ -\cos(90^o+\alpha) & -\sin(90^o+\alpha) & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.29}$$

$$\frac{\partial}{\partial \phi}{}^M_I\mathbf{C} = \mathbf{C}_\theta\frac{\partial}{\partial \phi}\mathbf{C}_\phi\mathbf{C}_\alpha = \mathbf{C}_\theta\begin{bmatrix} 0 & 0 & 0 \\ 0 & \sin(90^o-\phi) & -\cos(90^o-\phi) \\ 0 & \cos(90^o-\phi) & \sin(90^o-\phi) \end{bmatrix}\mathbf{C}_\alpha \tag{A.30}$$

$$\frac{\partial}{\partial \omega}{}^M_I\mathbf{C} = \frac{\partial}{\partial \omega}\mathbf{C}_\theta\mathbf{C}_\phi\mathbf{C}_\alpha = \begin{bmatrix} -\sin\theta & \cos\theta & 0 \\ -\cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{C}_\phi\mathbf{C}_\alpha \tag{A.31}$$

Finally, the remaining AHB measurement model derivatives are zero,

$$\frac{\partial^C \mathbf{h}_{s,i}}{\partial \eta} = \frac{\partial^C \mathbf{h}_{s,i}}{\partial \mu} = \frac{\partial^C \mathbf{h}_{s,i}}{\partial \eta_j} = \frac{\partial^C \mathbf{h}_{s,i}}{\partial \mu_j} = \mathbf{0}_{3\times 1} \tag{A.32}$$

Lastly, the pinhole camera model Jacobian that maps $\mathbf{h}_{s,i}$ to image space and which can be used for all of the previous derivatives can be written as

$$\frac{\partial^C \mathbf{z}_i}{\partial \mathbf{x}} = \mathbf{K}\frac{1}{[0,0,1]^C\mathbf{h}_{s,i}}\left(\mathbf{I}_3 - \frac{\mathbf{h}_{s,i}}{[0,0,1]^C\mathbf{h}_{s,i}}[0,0,1]\right)\frac{\partial^C \mathbf{h}_{s,i}}{\partial \mathbf{x}} \tag{A.33}$$

Appendix B

SRIF Measurement Update Iteration

Recalling the section describing the SREIF measurement update, we seek to adapt it to be capable of measurement update iteration in the same sense as the Iterated Extended Kalman Filter (IEKF). According to Reference [10], the IEKF update can be seen as an application of the Gauss-Newton method that approximates the maximum likelihood estimate. This has been useful in other VIO algorithms to date (see Reference [13], for example) because it increases EKF robustness to nonlinearity at a relatively low cost in terms of algorithm complexity and computation. This was not actually used in the results in this thesis but no extant references were found that formulate an analogous update for a square root extended information filter, so this development is meant to be broad enough to be useful for future SREIF formulations in other problems.

A cost function containing the prior information for the i-th iteration can be constructed as

$$\mathcal{J}_{p,i} = \left\| \mathbf{R}_i^- \delta\mathbf{x}_i^- - \delta\mathbf{b}_i^- \right\|^2 \tag{B.1}$$

where, for the zeroth iteration only, $\delta\mathbf{b}_0^- = \mathbf{0}$. After linearizing the measurement model about the current best estimate and forming the residual,

$$\mathbf{H}_i = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_i} \tag{B.2}$$

$$\mathbf{r}_i = \mathbf{z}_i - \mathbf{h}(\hat{\mathbf{x}}_i) \tag{B.3}$$

A second cost function that encapsulates the measurement contribution can be constructed as

$$\mathcal{J}_{m,i} = \left\| \mathbf{H}_i \delta\mathbf{x}_i^- - \mathbf{r}_i \right\|_{\mathbf{V}}^2 \tag{B.4}$$

Breaking down the Mahalanobis norm and whitening the measurement terms, the two cost functions can be summed to yield

$$\mathcal{J}_i = \mathcal{J}_{p,i} + \mathcal{J}_{m,i} = \left\| \begin{bmatrix} \mathbf{R}_i^- \\ \mathbf{R}_V \mathbf{H}_i \end{bmatrix} \delta \mathbf{x}_i^- - \begin{bmatrix} \delta \mathbf{b}_i^- \\ \mathbf{R}_V \mathbf{r}_i \end{bmatrix} \right\|^2 \tag{B.5}$$

This equation can be minimized by QR factorization

$$\begin{bmatrix} \mathbf{R}_i^i & \delta \mathbf{b}_i^- \\ \mathbf{R}_V \mathbf{H}_i & \mathbf{R}_V \mathbf{r}_i \end{bmatrix} = \mathbf{Q}_i \begin{bmatrix} \mathbf{R}_i^+ & \delta \mathbf{b}_i^+ \\ \mathbf{0} & \mathbf{e}_i \end{bmatrix} \tag{B.6}$$

and the optimal linearized state correction can then be computed as

$$\delta \mathbf{x}_i^+ = (\mathbf{R}_i^+)^{-1} \delta \mathbf{b}_i^+ \tag{B.7}$$

This correction is added to the nonlinear state

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i^- + \delta \mathbf{x}_i^+ \tag{B.8}$$

and the measurement Jacobians for the next iteration can then be computed about this new linearization point. To avoid double-counting information, the prior must remain unchanged, requiring that

$$\hat{\mathbf{x}}_0 + \delta \mathbf{x}_0^- = \hat{\mathbf{x}}_i + \delta \mathbf{x}_i^- \tag{B.9}$$

is held constant[119] (note: $\delta \mathbf{x}_0^- = \mathbf{0}$). This means that for each iteration,

$$\hat{\mathbf{x}}_{i+1} + \delta \mathbf{x}_{i+1}^- = \hat{\mathbf{x}}_i + \delta \mathbf{x}_i^+ \tag{B.10}$$

which, when reconciled with Equation B.9, results in

$$\delta \mathbf{x}_{i+1}^- = \delta \mathbf{x}_i^- - \delta \mathbf{x}_i^+ \tag{B.11}$$

Additionally, the same $\mathbf{R}_i = \mathbf{R}_0$ is used for each iteration and the final SRI matrix is extracted from the last iteration. With these definitions, the prior SRI vector can be iterated as

$$\delta \mathbf{b}_{i+1}^- = \mathbf{R}_0 \delta \mathbf{x}_{i+1}^- \tag{B.12}$$