Consistency Analysis and Improvement for Vision-aided Inertial Navigation

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Joel A. Hesch

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Stergios I. Roumeliotis, Advisor

March, 2016

Acknowledgements

I have received the help and support of a number of individuals while working on my doctorate.

First, I would like to express my gratitude and appreciation for my advisor, Professor Stergios I. Roumeliotis. Your dedication to research and your students has always been unwavering. Your contribution through countless hours of discussion and instruction, is immeasurable. I would not be the researcher that I am without you. I thank my Ph.D. committee members, Professors Tryphon Georgiou, Yousef Saad, Volkan Isler, and Junaed Sattar. I am grateful for your input and feedback while writing my dissertation, as well as your inspirational roles in my graduate studies at the University of Minnesota. To my lab mates at the Multiple Autonomous Robotics Systems Lab, you have been wonderful collaborators and friends. Remember, it's just one line of code!

I am thankful for the love and support of my family. You have taught me to work hard and strive to acheive my goals. You have stood by me and helped me to stay balanced through the ups and downs of graduate school.

Last, but not least, I am grateful for the numerous sources of financial support that funded my education and enabled me to attend academic conferences: the National Science Foundation, the National Institutes of Health, IEEE, the University of Minnesota Dept. of Computer Science and Engineering, University of Minnesota Digital Technology Center, Honeywell International, Institut National de Recherche en Informatique et en Automatique, the National Aeronautics and Space Administration, the Air Force Research Laboratory, and the Naval Research Laboratory.

Abstract

Navigation systems capable of estimating the six-degrees-of-freedom (d.o.f.) position and orientation (pose) of an object while in motion have been actively developed within the research community for several decades. Numerous potential applications include human-navigation aids for the visually impaired, first responders, and firefighters, as well as localization systems for autonomous vehicles such as submarines, ground robots, unmanned aerial vehicles, and spacecraft. The mobile industry has also recently become interested in six-dof localization for enabling interesting new applications on smart phones and tablets, such as games that are aware of motions in 3D space. The Global Positioning System (GPS) satellite network has been relied on extensively in pose-estimation applications; however, both humans and vehicles often need to operate in a wide variety of environments that preclude the use of GPS (e.g., underwater, inside buildings, in the urban canyon, and on other planets).

In order to estimate the 3D motion of person or robot in GPS-denied areas, it is requisite to employ sensors to determine the platform's displacement over time. To this end, inertial measurement units (IMUs) that sense the three-d.o.f rotational velocity as well as three-d.o.f. linear acceleration have been extensively used. IMU measurements, however, are corrupted by both sensor noise and bias, causing the resulting pose estimates to quickly become unreliable for navigation purposes. Although high-accuracy IMUs exist, they remain prohibitively expensive for widespread use. For this reason, it is common to *aid* an inertial navigation system (INS) with an alternative sensor such as a laser scanner, sonar, radar, or camera whose measurements can be employed to determine the platform's pose (or motion) with respect to the surrounding environment. Of these possible aiding sources, cameras have received significant attention due to their small size and weight, and the rich information that they supply.

State-of-the-art vision-aided inertial navigation systems (VINS) are able to provide highly-accurate pose estimates over short periods of time, however, they continue to exhibit limitations that prevent them from being used in critical applications for long-term deployment. Most notably, current approaches produce *inconsistent state estimates*, i.e., the errors are biased and the corresponding uncertainty in the estimate is unduely

small. In this thesis, we examine two key sources of estimator inconsistency for VINS, and propose solutions to mitigate these issues.

Contents

List of Tables

List of Figures

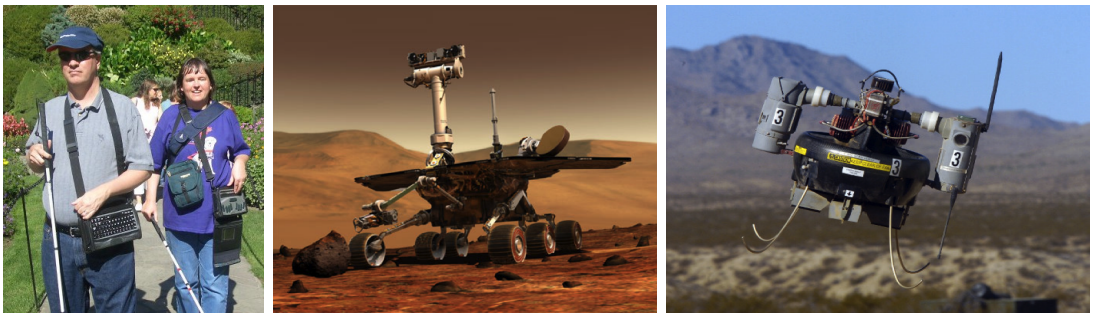
Chapter 1

Introduction

1.1 Motivation

Mobile robotics has long been heralded as a catalyst for the next major advancement in our technological revolution. By enabling greater efficiency in manufacturing, safer transportation through autonomous vehicles, time savings with house cleaning and service robots, and improved medical care with surgical robotics and caretakers for the elderly, robots promise to have a prodigious impact in the quality of our lives. Localization, i.e., the ability of a robot to determine its own position and orientation (pose) in three-dimensions (3D), is one of the fundamental requirements for enabling the development of autonomous robots. Similar to humans, mobile robots must be able to sense their motion, observe and memorize landmarks around them, and track their location with respect to important points of reference as they move. This information about the state of the world and its own pose within the world, is what will allow a robot to tackle complex high-level tasks such as planning an efficient path between two locations, while avoiding obstacles.

Figure 1.1 displays several pertinent examples in which a robot or intelligent system uses its sensor information to fulfill task-specific objectives in a mobile setting. In the first scenario [see Fig. 1.1(a)], a GPS-based navigation aid [1] provides turn-by-turn walking directions to a visually-impaired user. The system must employ information from both user input, via a braille keyboard, and its own sensor data (e.g., GPS, compass, and altimeter), in order to provide timely feedback to the person. In Fig. 1.1(b), a



(a) (b) (c)

Figure 1.1: (a) Sendero BrailleNote localization system [1], which employs the GPS network to determine the user's position, and relays navigation information to the user via a braille display. (b) NASA planetary rover [2] exploring the surface of Mars collecting visual data and geological samples. (c) Honeywell T-Hawk MAV [3] performing hovering surveillance of an area of interest.

NASA rover [2] explores the surface of Mars, collecting imagery and geological samples as it visits important sites of interest. In order to maximize the scientific benefits of the mission, the rover must maintain an accurate pose estimate so that the spatial distribution of samples can be properly determined and studied by scientists back on earth. Lastly, Fig. 1.1(c) depicts a Honeywell T-Hawk micro aerial vehicle (MAV) [3] that may be deployed in surveillance missions, such as border patrol or building security. This robot has the advantage of being able to hover and stare at a scene from a variety of vantage points, making it ideal for detecting suspicious activities and assessing potential security threats. In order, however, for law enforcement or security personell to make use of the surveillance footage, the vehicle pose must be available at all times.

In each of the above scenarios, the localization system employed should be able to determine the pose of the platform using only its onboard sensing modalities. Although GPS can provide accurate position information, it should not be relied upon as a primary aid for localization since many environments preclude its usage. For example, an earthbound robot will be GPS-denied when it is indoors, underground, or underwater, and will experience severe signal degradation in dense foliage, next to tall buildings (i.e., in the urban canyon), and on the battlefield where GPS may be jammed. Furthermore, even if GPS is available, it may not be accurate enough to enable a robot to maneuver in tight spaces, and since it does not directly provide orientation information, sole reliance

on GPS may lead a robot to become disoriented during stationary intervals.

As an alternative to GPS, a robot localization system can utilize artificial landmarks such as radio frequency identification (RFID) tags [4] or visual markers [5] placed throughout the environment. Subsequently, any robot with a map of the landmarks can safely navigate through the area, updating its pose whenever a landmark is within its detection range. This *beacon-based* approach has the key benefit that each landmark can be uniquely identified and validated via its digital signature, which virtually eliminates the possibility of misdetection. Unfortunately, beacons may be sparsely placed, and a robot may travel large distances between landmarks without precise knowledge of its location. Furthermore, since such systems must be installed and calibrated to determine the precise position of each landmark, beacon-based localization systems remain too costly for wide-spread adoption and have seen only limited testing in public buildings such as museums [6].

A more flexible approach is to develop a localization system that utilizes onboard sensors to infer the location of the robot. Sensors can broadly be classified in two categories: (i) *proprioceptive* and (ii) *exteroceptive*. Proprioceptive sensors measure some portion of the robot's state. For example, an odometer on a wheeled vehicle can measure its linear and rotational velocities as it moves on the ground plane. In applications that involve tracking 3D motions, the *de facto* proprioceptive sensor employed is an inertial measurement unit (IMU), which measures the 3D rotational velocity and linear acceleration of the sensing platform. Exteroceptive sensors, on the other hand, observe the environment from the robot's point of view. By tracking the apparent motion of distinctive objects that belong to the static scene, the robot can infer some degrees of freedom (d.o.f.) of its egomotion. For example, if a camera-equipped robot observes a chair moving from right-to-left through its field of view, it can infer that its own motion is left-to-right, since the chair is stationary.

This thesis focuses on the development and improvement of so-called *aided inertial navigation systems*, which utilize an IMU in conjunction with a camera or laser scanner to improve the accuracy and robustness of long-term localization. In both vision-aided and laser-aided inertial navigation systems (hereafter referred to as VINS and LINS, respectively), the system model governing the time evolution of the state, as well as the measurement models describing the camera and laser scanner observations are nonlinear.

Thus, even if the sensor noise terms can be well-characterized by Gaussian probability density functions (pdfs), the true pdf of the state (i.e., robot pose and environment) will be non-Gaussian and multimodal. Unfortunately, no general estimation framework exists for nonlinear systems, hence, in practice we resort to using *linearized* estimators, which typically approximate the pdf of the state as a unimodal Gaussian distribution. As we will show, the estimation errors incurred during this process can lead to *inconsistent* estimates which are both overconfident (i.e., the estimated pdf covariance is smaller than the true) and error prone [7].

1.2 Research Objectives

The principal research objective of this thesis is to analyze and mitigate two sources of VINS inconsistency, specifically: (i) improving consistency by ensuring that the number of unobservable directions of the estimator's (linearized) system matches that of the underlying true (nonlinear) system, and (ii) directly computing all solutions of the VINS pose-estimation problem to prevent errors due to tracking a single, incorrect solution.

- **Reducing estimator inconsistency due to mismatched un/observable directions:** A key source of estimation error arises due to the mismatch in the observability properties of nonlinear systems and their linearized counterparts used for estimation purposes. For instance, it is common practice to employ the extended Kalman filter (EKF) [8] to estimate the state of a nonlinear system by assuming that the nonlinear plant and measurement equations can be well approximated as locally linear, through first-order Taylor series expansion. Unfortunately, as has been shown for certain 2D localization problems [9, 10, 11], this approximation can fundamentally alter the structure of the system's observable and unobservable subspaces, allowing spurious information to be surreptitiously obtained along directions in which it should not. Our objective in this research thread is to understand the interplay between observability and consistency in VINS and propose estimator modifications that reduce or eliminate the inconsistency.

- **Reducing estimator inconsistency due to multiple pose hypotheses:** A second source of pose error arises from not properly accounting for the existence

of multiple solutions for the vision-based pose determination problem[1] in the static-sensor, single-image case [12, 13]. In particular, tracking the trajectory of the sensing package through time is a nonlinear estimation problem (requiring the determination of a multimodal pdf over the quantities to be estimated) that can be addressed recursively within a filtering framework (e.g., iterated extended Kalman filter (I-EKF) [8]), or refined in a batch form over several time-steps in a maximum a posteriori (MAP) estimator [14]. Although the true pdf is multimodal, many parametric estimators (e.g., I-EKF, unscented Kalman filter [15], and MAP) approximate the estimated pdf as unimodal, therefore they can only track one of the possible hypotheses, which, depending on the accuracy of the prior, may be an incorrect solution. In order to address this issue, we propose to directly solve the vision-based pose determination problem from a single image given observations of known landmarks. We formulate this as a nonlinear batch least-squares (BLS) optimization problem, whose minima we can determine directly by solving the corresponding Karush-Kuhn-Tucker (KKT) optimality conditions [16].

In order to accomplish these research objectives, we begin by introducing a LINS for the visually impaired that exploits measurements of structural planes indoors to track the person's pose as they move. This system is analyzed in detail, including its observability properties as well as practical issues of laser-to-IMU extrinsic calibration. Due to limitations in scanning a 3D world with a moving, arbitrarily oriented 2D laser scanner, the LINS position error drifts unless three or more orthogonal structural planes are concurrently observed. Unfortunately, small-sized, human-portable 2D laser scanners do not have sufficient range to ensure that this three-orthogonal-plane condition is always met in realistic scenarios. This motivates the use of alternative exteroceptive sensors that do not place stringent constraints on the environment composition or layout.

We extend our investigation to VINS in three stages, beginning with the general case, and gradually reducing the amount of information available until we reach the static camera scenario. First, we consider an IMU and camera package that moves along an

[1] Vision-based pose determination is the task of estimating the six-d.o.f. pose of a camera from observations of three or more known features in a single image. In the computer vision literature, this problem is commonly referred to as the perspective-n-point problem. It is typically addressed by assuming the measurements are noise free, and analytically solving for the camera pose in the nonlinear measurement equations [12].

arbitrary trajectory in a general indoor or outdoor environment. In this case, we focus on mitigating inconsistency caused by a mismatch in the observability properties of the true system and the one employed by the estimator. Second, we consider the vision-based navigation system that does not receive any inputs from an IMU. By using a local-velocity tracking model, we show that a camera alone can be used for navigation purposes, but the scale of the motion and scene will be arbitrary. We extend our consistency analysis to show how the drift of the scale estimate causes inconsistency in vision-only navigation, and propose a method to address it. Third, we consider the case of pose determination from a single, stationary camera observing known point features. This parameter-estimation problem is different compared to the two cases above, since no motion model is involved. As we will show, a key source of inconsistency in this case arises from not properly accounting for all system solutions, when multiple solutions exist.

1.3 Organization of the Manuscript

The remainder of this dissertation is organized in the following manner: Chapters 2 through 5 detail our specific accomplishments in the analysis and improvement of aided inertial navigation, focussing on the cases of LINS and VINS, while Chapter 6 presents our conclusions and future research directions. In particular,

- Chapter 2 describes a novel 3D indoor LINS for the visually impaired. An EKF fuses information from an IMU and a 2D laser scanner, to concurrently estimate the six d.o.f. pose of the sensing package and a 3D map of the environment. Rather than constraining the person to purely planar motion, the IMU measurements are integrated to estimate the pose along a general 3D trajectory. To mitigate the accumulation of inertial drift errors, the pose estimates are corrected using laser measurements, namely line-to-plane correspondences between linear segments in the laser-scan data and structural planes of the building. Utilizing orthogonal building planes as map features results in a human-interpretable layout of the environment, and ensures that the each feature can be efficiently initialized and estimated. A practical method is presented to initialize the pose and the IMU biases using observations of known planes and zero-velocity updates, respectively.

In addition to the filter design, the observability properties of the nonlinear system are studied to show under which measurement conditions the 3D pose can be accurately estimated. Lastly, an approach for utilizing the sensors' measurements to perform on-line calibration of the laser-to-IMU transformation is developed, which enables the highest possible localization accuracy. The proposed LINS is experimentally validated by a person traveling in both known and unknown 3D environments to demonstrate its reliability and accuracy for indoor localization and mapping.

- In Chapter 3, we study estimator inconsistency in VINS from a standpoint of system observability. We postulate that a leading cause of inconsistency is the gain of spurious information along unobservable directions, resulting in smaller uncertainties, larger estimation errors, and divergence. We develop an observability-constrained VINS (OC-VINS), which explicitly enforces the unobservable directions of the system, hence preventing spurious information gain and reducing inconsistency. This framework is applicable to several variants of the VINS problem such as visual simultaneous localization and mapping (V-SLAM) as well as visual-inertial odometry using the multi-state constraint Kalman filter (MSC-KF). Our analysis, along with the proposed method for reducing inconsistency, are extensively validated in simulation and experimentally.

- In Chapter 4, we study the problem of estimator inconsistency in single-camera simultaneous localization and mapping (MonoSLAM). Using a local-velocity tracking model for the camera motion, we study the system observability properties for MonoSLAM and show that scale becomes erroneously observable due to linearization errors when using linearized estimation approaches. Moreover, we introduce an observability-constrained MonoSLAM (OC-MonoSLAM) approach, following the methodology of OC-VINS, which explicitly enforces the unobservable directions of the system, hence preventing spurious information gain and reducing inconsistency. Our analysis, along with the proposed method for reducing inconsistency, are validated in simulation and through real-world experimentation.

- In Chapter 5, we present a direct least-squares (DLS) method for computing all solutions of the perspective-n-point (PnP) camera pose determination problem

for the general case when $n \geq 3$. Specifically, based on the camera measurement equations, we formulate a nonlinear least-squares (LS) cost function whose optimality conditions constitute a system of three third-order polynomial equations. Subsequently, we employ the multiplication matrix to determine all the roots of the system analytically, and hence all minima of the LS cost function, without requiring iterations or an initial guess of the parameters. A key advantage of our method is scalability, since the order of the polynomial system that we solve is independent of the number of points. We compare the performance of our algorithm with the leading PnP approaches, both in simulation and experimentally, and demonstrate that DLS consistently achieves accuracy close to the maximum-likelihood estimator (MLE).

- Chapter 6 reviews the contributions of this thesis and presents an overview of the future research directions. In particular, we would like to extend our investigation to several areas that are outside the scope of the current work. For example, it may be possible to improve consistency further by making use of known motion constraints, such as a dynamics model on a vehicle, to augment the information processed by the estimator. Moreover, accuracy can be improved by employing motion models when the device is in an environment with fewer features than necessary for enabling estimation of the system's observable modes (e.g., when all visual features are far away for the camera, it is not possible to resolve the device's linear velocity).

Chapter 2

Laser-aided Inertial Navigation in Unknown Indoor Environments

2.1 Introduction

For humans, safe and efficient navigation requires knowledge of the environmental layout, path planning, obstacle avoidance, and determining one's pose with respect to the world. For a *visually-impaired* person, these tasks can be exceedingly difficult to accomplish, and there are high risks associated with failure in any of them. To address some of these issues, guide dogs and white canes are widely used for the purposes of wayfinding and environment sensing. The former, however, has costly training requirements, while the latter can only provide cues about one's immediate surroundings. On the other hand, commercially available electronic navigation systems designed for the visually impaired (e.g., [17], [1]) rely on GPS signals and cannot be utilized indoors, under tree cover, or next to tall buildings where reception is poor.

In the academic community, numerous electronic navigation systems for GPS-denied environments have been proposed. However, the majority of the existing algorithms are designed for mobile robots that are limited to move on planar surfaces [18, 19] or require heavy sensors, such as a 3D laser scanner [20, 21], that cannot be carried by a human. Other algorithms, which have relied on visual information [22, 23], are sensitive to variable lighting conditions and require processing resources that are not typically available on portable computing devices.

To address these issues, we aim to design a personal *indoor* navigation system that fulfills the following requirements:

- The system must accurately track the *six-d.o.f. pose* of the visually impaired person, allowing them to safely navigate in a *3D environment*.
- The navigation aid should enable the person to walk through previously *unknown buildings* without getting lost. This requires constructing a map of the explored area and localizing with respect to it in *real-time*.
- The selected sensors should be *robust* to environmental changes, such as lighting conditions, reliable in the presence of clutter and moving objects, and work within the *computational and memory limits* of a portable computing device.
- The navigation aid should be *compact, unobtrusive* to the person, and *lightweight* enough to be carried without fatigue.

To meet these objectives, we focus on designing an indoor LINS using an IMU and a *2D laser scanner*, based upon our preliminary results in [24, 25]. Employing this sensor pair ensures feasibility of manufacturing a light-weight and compact sensor package that can be carried by a person, since a wide variety of small IMUs (e.g., Memsense nIMU) and compact-size 2D laser scanners (e.g., Hokuyo URG) are commercially available. Additionally, using a laser scanner instead of a camera provides greater reliability and robustness under poor lighting conditions.

The proposed algorithm tracks the six-d.o.f. pose of the person by integrating the IMU measurements (linear acceleration and rotational velocity) using an EKF. However, without corrections from an exteroceptive sensor, the IMU measurement noise and bias drift would cause the pose estimation errors to grow unbounded over time. To mitigate this issue, we propose to update the pose estimates by utilizing straight-line features extracted from the 2D laser scans. In particular, as the person moves, the laser scanner's attitude changes which causes its scanning plane to intersect a variety of structural planes of the building (i.e., the walls, floor, and ceiling). If the structural planes are known *a priori* from a building map, we can use the information from the line-to-plane measurements in order to update the person's pose estimates [24]. Unfortunately, in many cases in practice, a building map is not available in advance. To overcome this

challenge, we simultaneously construct a building map in order to utilize *previously unknown* structural planes in the localization process [25]. We exploit the fact that most indoor structural planes are *orthogonal to each other*, which allows us to fix each plane's orientation the first time it is observed, and only estimate its distance to the origin of the global reference frame.

Constructing the map based on orthogonal planar structures has the advantage of keeping the person's orientation error bounded [26] in addition to providing inherent robustness to clutter and moving objects. Furthermore, the estimated map directly provides a *human-interpretable layout* of the building that can simplify the task of wayfinding towards a destination. Moreover, due to the limited number of structural planes in each building, the computational load of the algorithm remains bounded. This, together with the low processing cost of line-segment extraction from the 2D laser scans, ensures the real-time execution of the algorithm on a hand-held computer with limited computational and memory resources.

We demonstrate the validity and reliability of the proposed approach with real-world experiments in both known and unknown environments. In the first case, we present a loop trajectory of 120 m in length that covers part of one floor of the Keller Hall at the University of Minnesota. The second test covers multiple levels of Akerman Hall at the University of Minnesota. In this 270 m trajectory, the person traverses several staircases and a disability access ramp. In addition, both test environments includes significant clutter (e.g., trashcans, storage boxes, and furniture), as well as a normal flow of pedestrian traffic. Despite these challenges, our algorithm accurately tracks the person's pose, and correctly estimates a map of the building layout.

In order to ensure that the IMU and the laser scanner measurements provide sufficient information for estimating the person's pose, we study the observability of the corresponding nonlinear system. We also address the more practical matter of how to efficiently initialize the filter. Lastly, we provide a novel on-line method for calibrating the laser-to-IMU transformation using either previously known or unknown planar features, since inaccurate calibration can lead to biased filter estimates.

The remainder of the chapter is organized as follows: In Section 2.2, we begin with an overview of the related literature. Section 2.3 presents the core of our algorithm, which is an EKF-based pose estimator. We describe how to efficiently initialize the

state of the filter in Section 2.4. In Section 2.5, we study the observability properties of the map-based localization system, and show the system is observable under mild conditions that are typically fulfilled in practice. Subsequently, we describe our approach for calibrating the laser-to-IMU transformation using line-to-plane correspondences in Section 2.6. Experimental validation of the proposed method is provided in Section 2.7. Lastly, we conclude the paper and present future research directions in Section 2.8.

2.2 Related Work

Recent work has focused primarily on developing hazard-detection aids for the visually impaired with the purpose of *detecting and avoiding obstacles* [27, 28] and describing objects' size and color [29]. These systems cannot be directly used as wayfinding aids without the development of appropriate algorithms for localization. In contrast to the above systems, navigation aids have been designed that explicitly track a person's location and heading direction. Most relevant efforts have primarily addressed GPS-based *outdoor navigation* which cannot be used inside a building [30, 31]. *Indoor navigation* is more challenging, since pose information can only be inferred from ego-motion and environmental cues. In what follows, we provide a discussion of several existing indoor navigation systems.

Navigating using ego-motion

Dead-reckoning systems track a person's pose *without any external references*. Common approaches are based on foot-mounted accelerometers [32]. As a person walks, their position is computed by double integration of the acceleration measurements. Unfortunately, the accelerometer bias and noise are integrated as well, which causes the *position error* to grow unbounded. Even if the rate of position-error increase can be reduced with static-period drift corrections [33, 34], dead-reckoning systems still remain unreliable over long time intervals.

Navigating with known references

Unlike dead-reckoning approaches that do not employ external references, map-based systems infer position and orientation information from known landmarks or beacons

in the environment. For example, in [4], a robot is attached at the end of a leash as a substitute for a guide dog, and localizes using odometry and a network of RFID tags. In [5], the authors presented another approach in which a hand-held camera identifies retro-reflective digital signs. Similar methods also exist based on ultrasound [31] and infrared [35] beacons. In [36], we presented a map-based indoor localization aid for the visually impaired comprised of a pedometer, a tri-axial gyroscope, and a 2D laser scanner. We exploited known corners at hallway intersections (computed from the building blueprints) as landmarks for localization. Unfortunately, all map-based or beacon-based localization methods suffer from common limitations which include: (i) *time and cost* associated with acquiring the map or installing the beacons, (ii) the system's *inability to adapt* to spatial layout changes, and (iii) the *restriction* of use to previously mapped areas.

Navigating in unknown environments

The most flexible navigation aids are those that can exploit environment sensing to perform SLAM. The majority of the proposed systems for SLAM consider either 2D map and sensor motion [37, 38], or restrict the sensor motion to planar surfaces and create a 3D map of the surroundings [18, 19, 26]. These algorithms are not generally suitable for use on a personal navigation system since the motion of a human is not limited to a planar surface (e.g., when climbing stairs).

There exist several approaches for estimating a 3D map and the six-d.o.f. pose of a robot (3D SLAM) that employ 3D point cloud matching techniques [e.g., Iterative Closest Point (ICP)] [20, 21, 39, 40, 41]. However, the computational requirements for matching 3D point clouds are typically prohibitive for real-time implementation. More importantly, the 3D laser scanners needed for acquiring the point clouds are too large and heavy for a person to carry, thus making these systems inappropriate for use as a personal navigation aid. Alternative methods for performing 3D SLAM employ cameras to map the environment based on visual landmarks [22, 23]. The main drawback of camera-based systems is their sensitivity to variable lighting conditions, which restricts their use as navigation aids for the visually impaired where reliability is of paramount importance. Additionally, processing images and extracting visual features are typically

computationally intensive tasks that are impractical to carry out on hand-held computing devices. Furthermore, constructing a map of the 3D locations of visual landmarks (e.g., SIFT features [42]) often used in these approaches may not be geometrically meaningful or interpretable by humans. Finally, extracting and matching visual landmarks in indoor environments can be challenging and unreliable due to insufficient texture.

To address these limitations, we propose an LINS based on a 2D laser scanner and an IMU. The key differentiating factor of our work is that we can explore and map 3D environments with a sensing package that follows arbitrary 3D trajectories, despite the fact that the exteroceptive sensor employed only senses in 2D during each laser scan. Specifically, our system tracks the six-d.o.f. pose of the person and measures both known building planes as well as new planes which it maps as the unknown portions of the environment are explored. We note that using commonly-occurring structural planes as map features ensures the applicability of the method in practice. The estimated structural planes directly represent the geometric layout of the building that can be easily interpreted by humans. Moreover, due to the limited number of structural planes in each building, the computational requirements of our algorithm do not grow unbounded over time, since the size of the estimated state vector remains bounded. Finally, our algorithm can perform *on-line calibration* of the relative pose between laser and IMU, which may not be accurately known *a priori*.

2.3 Algorithm Description

A hand-held computer collects measurements from the navigation aid consisting of an IMU and a 2D laser scanner, which are rigidly connected (see Fig. 2.1). The sensor data is fused in an EKF to concurrently estimate the six-d.o.f. pose of the sensor platform, as well as the 3D map of the building's perpendicular structural planes (i.e., the walls, floor, and ceiling). In what follows, we present the propagation and update models used by the EKF.

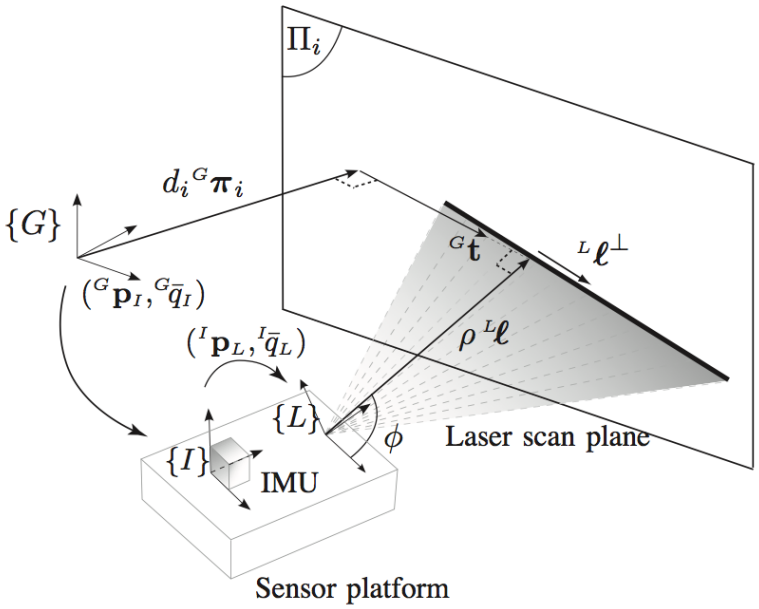


Figure 2.1: As the IMU-laser sensor platform moves, the laser scan plane intersects a structural planar surface, Π_i, described by d_i and ${}^G\boldsymbol{\pi}_i$, which are the Hessian normal form components of the plane with respect to the global frame of reference, $\{G\}$. The shortest vector in the laser scan plane from the origin of the laser frame, $\{L\}$, to Π_i has length ρ and direction ${}^L\boldsymbol{\ell}$, with respect to $\{L\}$. The line of intersection has direction, ${}^L\boldsymbol{\ell}^\perp$, with respect to $\{L\}$ and is described by the polar parameters (ρ, ϕ). The vector from the intersection of ${}^G\boldsymbol{\pi}_i$ and Π_i to the intersection of $\rho{}^L\boldsymbol{\ell}$ and Π_i, is ${}^G\mathbf{t}$. The IMU-laser transformation is denoted by $({}^I\mathbf{p}_L, {}^I\bar{q}_L)$, while the IMU pose with respect to $\{G\}$ is $({}^G\mathbf{p}_I, {}^G\bar{q}_I)$.

2.3.1 Filter Propagation

The EKF estimates the IMU pose and linear velocity together with the time-varying IMU biases and the map. The filter state is the $(16+N)\times 1$ vector:

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} {}^I\bar{q}_G^T & \mathbf{b}_g^T & {}^G\mathbf{v}_I^T & \mathbf{b}_a^T & {}^G\mathbf{p}_I^T & | & d_1 & \cdots & d_N \end{bmatrix}^T \\ &= \begin{bmatrix} \mathbf{x}_s^T & | & \mathbf{x}_d^T \end{bmatrix}^T, \end{aligned} \tag{2.1}$$

where $\mathbf{x}_s(t)$ is the 16×1 sensor platform state, and $\mathbf{x}_d(t)$ is the $N\times 1$ state of the structural plane map. The first component of the sensor platform state is, ${}^I\bar{q}_G(t)$, which

is the unit quaternion representing the orientation of the *global frame* $\{G\}$ in the IMU frame, $\{I\}$, at time t. The frame $\{I\}$ is attached to the IMU (see Fig. 2.1), while $\{G\}$ is an inertial reference frame whose origin coincides with the initial IMU position, and whose orientation is aligned with the perpendicular structural planes according to the filter initialization procedure described in Section 2.4. The sensor platform state also includes the position and velocity of $\{I\}$ in $\{G\}$, denoted by the 3×1 vectors ${}^G\mathbf{p}_I(t)$ and ${}^G\mathbf{v}_I(t)$, respectively. The remaining components are the biases, $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$, affecting the gyroscope and accelerometer measurements, which are modeled as random-walk processes driven by the zero-mean, white Gaussian noise $\mathbf{n}_{wg}(t)$ and $\mathbf{n}_{wa}(t)$, respectively.

The building map is comprised of N static planar features Π_i, $i = 1, \ldots, N$, which includes all planes (if any) that are known from the building blue prints, and grows as new planes are detected. Each plane is described by its Hessian normal form components d_i and ${}^G\boldsymbol{\pi}_i$, which are the distance from the plane to the origin of $\{G\}$, and the 3×1 normal vector of the plane expressed in $\{G\}$, respectively.[1] The map state, $\mathbf{x}_d$, consists of the scalar distances, d_i, $i = 1, \ldots, N$, which are estimated along with the state of the sensing package. We only map perpendicular structural planes, hence, we do not need to estimate each plane's normal-vector. Instead, we store them in the map parameter vector $\left[{}^G\boldsymbol{\pi}_1^T \ldots {}^G\boldsymbol{\pi}_N^T\right]^T$, where each component ${}^G\boldsymbol{\pi}_i$ is determined once during the new plane initialization step (see Section 2.3.3) or is available from the blueprint layout. With the state of the system now defined, we turn our attention to the continuous-time dynamical model which governs the state of the system.

[1] A point ${}^G\mathbf{p}$ lies on plane Π_i if ${}^G\boldsymbol{\pi}_i^T {}^G\mathbf{p} - d_i = 0$.

Continuous-time model

The system model describing the time evolution of the state is (see [43, 44]):

$$
\begin{aligned}
{}^{I}_{G}\dot{\bar{q}}(t) &= \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}(t)){}^{I}\bar{q}_{G}(t) && (2.2)\\
{}^{G}\dot{\mathbf{p}}_{I}(t) &= {}^{G}\mathbf{v}_{I}(t) && (2.3)\\
{}^{G}\dot{\mathbf{v}}_{I}(t) &= {}^{G}\mathbf{a}(t) && (2.4)\\
\dot{\mathbf{b}}_{g}(t) &= \mathbf{n}_{wg}(t) && (2.5)\\
\dot{\mathbf{b}}_{a}(t) &= \mathbf{n}_{wa}(t) && (2.6)\\
\dot{d}_{i}(t) &= 0\ ,\ \ i = 1,\ldots,N. && (2.7)
\end{aligned}
$$

In these expressions, $\boldsymbol{\omega}(t) = [\omega_1(t)\ \ \omega_2(t)\ \ \omega_3(t)]^T$ is the rotational velocity of the IMU, expressed in $\{I\}$, ${}^{G}\mathbf{a}$ is the IMU acceleration expressed in $\{G\}$, and

$$
\mathbf{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor \boldsymbol{\omega} \times \rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \quad \lfloor \boldsymbol{\omega} \times \rfloor \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.
$$

The gyroscope and accelerometer measurements, $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$, used for state propagation, are

$$
\begin{aligned}
\boldsymbol{\omega}_m(t) &= \boldsymbol{\omega}(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) && (2.8)\\
\mathbf{a}_m(t) &= \mathbf{C}({}^{I}\bar{q}_{G}(t))\,({}^{G}\mathbf{a}(t) - {}^{G}\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t), && (2.9)
\end{aligned}
$$

where $\mathbf{n}_g$ and $\mathbf{n}_a$ are zero-mean, white Gaussian noise processes, and ${}^{G}\mathbf{g}$ is the gravitational acceleration. The matrix $\mathbf{C}(\bar{q})$ is the rotation matrix corresponding to $\bar{q}$. Also note that the distances to the building planes are fixed with respect to $\{G\}$, thus their time derivatives are zero [see (2.7)].

Linearizing at the current estimates and applying the expectation operator on both

sides of (2.2)-(2.7), we obtain the state estimate propagation model

$$
\begin{aligned}
{}^{I}_{G}\dot{\hat{\bar{q}}}(t) &= \frac{1}{2}\mathbf{\Omega}(\hat{\boldsymbol{\omega}}(t)){}^{I}_{G}\hat{\bar{q}}(t) && (2.10)\\
{}^{G}\dot{\hat{\mathbf{p}}}_{I}(t) &= {}^{G}\hat{\mathbf{v}}_{I}(t) && (2.11)\\
{}^{G}\dot{\hat{\mathbf{v}}}_{I}(t) &= \mathbf{C}^{T}({}^{I}_{G}\hat{\bar{q}}(t))\,\hat{\mathbf{a}}(t) + {}^{G}\mathbf{g} && (2.12)\\
\dot{\hat{\mathbf{b}}}_{g}(t) &= \mathbf{0}_{3\times 1} && (2.13)\\
\dot{\hat{\mathbf{b}}}_{a}(t) &= \mathbf{0}_{3\times 1} && (2.14)\\
\dot{\hat{d}}_{i}\ (t) &= 0\ ,\ \ i=1,\ldots,N, && (2.15)
\end{aligned}
$$

with $\hat{\mathbf{a}}(t)=\mathbf{a}_m(t)-\hat{\mathbf{b}}_a(t)$, and $\hat{\boldsymbol{\omega}}(t)=\boldsymbol{\omega}_m(t)-\hat{\mathbf{b}}_g(t)$.
The $(15+N)\times 1$ error-state vector is defined as

$$
\begin{aligned}
\widetilde{\mathbf{x}} &= \begin{bmatrix} {}^{I}\boldsymbol{\delta\theta}_{G}^{T} & \widetilde{\mathbf{b}}_{g}^{T} & {}^{G}\widetilde{\mathbf{v}}_{I}^{T} & \widetilde{\mathbf{b}}_{a}^{T} & {}^{G}\widetilde{\mathbf{p}}_{I}^{T} & | & \widetilde{d}_{1} & \cdots & \widetilde{d}_{N} \end{bmatrix}^{T}\\
&= \begin{bmatrix} \widetilde{\mathbf{x}}_{s}^{T} & | & \widetilde{\mathbf{x}}_{d}^{T} \end{bmatrix}^{T}, && (2.16)
\end{aligned}
$$

where $\widetilde{\mathbf{x}}_s(t)$ is the 15×1 error state corresponding to the sensing platform, and $\widetilde{\mathbf{x}}_d(t)$ is the $N\times 1$ error state of the map. For the IMU position, velocity, biases, and the map, an additive error model is utilized (i.e., $\widetilde{x}=x-\hat{x}$ is the error in the estimate $\hat{x}$ of a quantity x). However, for the quaternion we employ a multiplicative error model. Specifically, the error between the quaternion $\bar{q}$ and its estimate $\hat{\bar{q}}$ is the 3×1 angle-error vector, $\boldsymbol{\delta\theta}$, implicitly defined by the *error quaternion*

$$
\delta\bar{q} = \bar{q}\otimes\hat{\bar{q}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}^{T} & 1 \end{bmatrix}^{T}, \qquad (2.17)
$$

where $\delta\bar{q}$ describes the small rotation that causes the true and estimated attitude to coincide. The main advantage of this error definition is that it allows us to represent the attitude uncertainty by the 3×3 covariance matrix $E\{\boldsymbol{\delta\theta}\boldsymbol{\delta\theta}^{T}\}$. Since the attitude corresponds to three d.o.f., this is a minimal representation.

The linearized continuous-time error-state equation is

$$
\begin{aligned}
\dot{\widetilde{\mathbf{x}}} &= \begin{bmatrix} \mathbf{F}_{s,c} & \mathbf{0}_{15\times N} \\ \mathbf{0}_{N\times 15} & \mathbf{I}_{N} \end{bmatrix}\widetilde{\mathbf{x}} + \begin{bmatrix} \mathbf{G}_{s,c} \\ \mathbf{0}_{N\times 15} \end{bmatrix}\mathbf{n}\\
&= \mathbf{F}_{c}\,\widetilde{\mathbf{x}} + \mathbf{G}_{c}\,\mathbf{n}\,, && (2.18)
\end{aligned}
$$

where $\mathbf{I}_N$ denotes the $N \times N$ identity matrix, $\mathbf{F}_{s,c}$ is the continuous-time error-state transition matrix corresponding to the sensor platform state, and $\mathbf{G}_{s,c}$ is the continuous time input noise matrix, i.e.,

$$\mathbf{F}_{s,c} = \begin{bmatrix} -\lfloor \hat{\boldsymbol{\omega}} \times \rfloor & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\mathbf{C}^T({}^I_G\hat{\bar{q}})\lfloor \hat{\mathbf{a}} \times \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^I_G\hat{\bar{q}}) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

$$\mathbf{G}_{s,c} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^I_G\hat{\bar{q}}) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}, \quad \mathbf{n} = \begin{bmatrix} \mathbf{n}_g \\ \mathbf{n}_{wg} \\ \mathbf{n}_a \\ \mathbf{n}_{wa} \end{bmatrix},$$

where $\mathbf{0}_3$ is the 3×3 matrix of zeros. The system noise covariance matrix $\mathbf{Q}_c$ depends on the IMU noise characteristics and is computed off-line [44].

Discrete-time implementation

The IMU signals $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ are sampled at a constant rate $1/T$, where $T \triangleq t_{k+1} - t_k$. Every time a new IMU measurement is received, the state estimate is propagated using 4th-order Runge-Kutta numerical integration of (2.10)–(2.15). In order to derive the discrete-time covariance propagation equation, we evaluate the discrete-time state transition matrix

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}_c(\tau) \mathrm{d}\tau\right) \tag{2.19}$$

and the discrete-time system noise covariance matrix

$$\mathbf{Q}_{d,k} = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \tau) \mathbf{G}_c \mathbf{Q}_c \mathbf{G}_c^T \boldsymbol{\Phi}^T(t_{k+1}, \tau) \mathrm{d}\tau. \tag{2.20}$$

The propagated covariance is then computed as

$$\mathbf{P}_{k+1|k} = \boldsymbol{\Phi}_k \mathbf{P}_{k|k} \boldsymbol{\Phi}_k^T + \mathbf{Q}_{d,k}. \tag{2.21}$$

After processing the IMU measurements to propagate the filter state and covariance, we process any available laser scan measurements in the filter update step (see Section 2.3.2).

2.3.2 Landmark Update

As the IMU-laser platform moves in an indoor environment, the laser-scan plane intersects the perpendicular structural planes of the building. These measurements are exploited to update the state estimate. To simplify the discussion, we consider a single line measurement, ${}^{L}\boldsymbol{\ell}^{\perp}$, corresponding to the intersection of the laser-scan plane and map plane, Π_i (see Fig. 2.1). The line is described in the laser frame, $\{L\}$, by (ρ, ϕ), where ρ is the distance from the origin of $\{L\}$ to the line, and ϕ is the angle of the vector ${}^{L}\boldsymbol{\ell}$ perpendicular to the line.[2] We will hereafter express the line direction in $\{I\}$, as ${}^{I}\boldsymbol{\ell}^{\perp} = \mathbf{C}({}^{I}\bar{q}_L)\begin{bmatrix}\sin\phi & -\cos\phi & 0\end{bmatrix}^T$, where ${}^{I}\bar{q}_L$ is the unit quaternion representing the orientation of the laser frame in the IMU frame (see Sect. 2.6). In what follows, we describe how each line is exploited to define two *measurement constraints*, which are used by the EKF to update the state estimates.

Orientation Constraint

The first constraint is on the orientation of $\{I\}$ with respect to $\{G\}$. The normal to the plane Π_i, vector ${}^{G}\boldsymbol{\pi}_i$, is perpendicular to $\mathbf{C}^T({}^{I}\bar{q}_G)\,{}^{I}\boldsymbol{\ell}^{\perp}$ (see Fig. 2.1), which yields the following *orientation measurement constraint*

$$z_1 = {}^{G}\boldsymbol{\pi}_i^T\,\mathbf{C}^T({}^{I}\bar{q}_G)\,{}^{I}\boldsymbol{\ell}^{\perp} = 0\,. \tag{2.22}$$

The expected measurement is

$$\hat{z}_1 = {}^{G}\boldsymbol{\pi}_i^T\,\mathbf{C}^T({}^{I}\hat{\bar{q}}_G)\,{}^{I}\boldsymbol{\ell}_m^{\perp}\,, \tag{2.23}$$

where ${}^{I}\boldsymbol{\ell}_m^{\perp} = \mathbf{C}({}^{I}\bar{q}_L)\begin{bmatrix}\sin\phi_m & -\cos\phi_m & 0\end{bmatrix}^T$ is the *measured* line direction with $\phi_m = \phi - \tilde{\phi}$. The measurement residual is $r_1 = z_1 - \hat{z}_1 = -\hat{z}_1$ and the corresponding linearized

[2] We utilized the Split-and-Merge algorithm [45] to segment the laser-scan data and a weighted line-fitting algorithm [46] to estimate the line parameters (ρ, ϕ) for each line.

error model is

$$\widetilde{z}_1 \simeq \begin{bmatrix} -{}^G\boldsymbol{\pi}_i^T \mathbf{C}^T({}^I\hat{\bar{q}}_G) \lfloor {}^I\boldsymbol{\ell}_m^{\perp} \times \rfloor & \mathbf{0}_{1\times 12} \end{bmatrix} \widetilde{\mathbf{x}}_s + \begin{bmatrix} \mathbf{0}_{1\times N} \end{bmatrix} \widetilde{\mathbf{x}}_d + \begin{bmatrix} {}^G\boldsymbol{\pi}_i^T \mathbf{C}^T({}^I\hat{\bar{q}}_G) {}^I\boldsymbol{\ell}_m & 0 \end{bmatrix} \begin{bmatrix} \tilde{\phi} \\ \tilde{\rho} \end{bmatrix}$$
$$= \mathbf{h}_{1,s}^T \widetilde{\mathbf{x}}_s + \mathbf{h}_{1,d}^T \widetilde{\mathbf{x}}_d + \boldsymbol{\gamma}_1^T \mathbf{n}_\ell, \tag{2.24}$$

where ${}^I\boldsymbol{\ell}_m = \mathbf{C}({}^I\bar{q}_L) \begin{bmatrix} \cos\phi_m & \sin\phi_m & 0 \end{bmatrix}^T$ is the perpendicular to the measured line direction and $\rho_m = \rho - \tilde{\rho}$ is the measured distance to the line. The vectors $\mathbf{h}_{1,s}^T$, $\mathbf{h}_{1,d}^T$, and $\boldsymbol{\gamma}_1^T$ are the Jacobians of (2.22) with respect to the state and line parameters. The 2×1 error vector $\mathbf{n}_\ell$ is assumed to be zero-mean, white Gaussian, with covariance matrix $\mathbf{R} = E\{\mathbf{n}_\ell \mathbf{n}_\ell^T\}$ computed for each line from the weighted line-fitting procedure [46].

Distance Constraint

From Fig. 2.1, the following geometric relationship holds:

$${}^G\mathbf{p}_I + \mathbf{C}^T({}^I\bar{q}_G) \left({}^I\mathbf{p}_L + \rho\, {}^I\boldsymbol{\ell}\right) = d_i\, {}^G\boldsymbol{\pi}_i + {}^G\mathbf{t}, \tag{2.25}$$

where ${}^I\boldsymbol{\ell} = \mathbf{C}({}^I\bar{q}_L) \begin{bmatrix} \cos\phi & \sin\phi & 0 \end{bmatrix}^T$ is the perpendicular to the line direction, and ${}^I\mathbf{p}_L$ is the position of the laser scanner in the IMU frame. Since the vector ${}^G\mathbf{t}$ is unknown and cannot be measured we need to eliminate it from the equation. We do so by projecting (2.25) onto ${}^G\boldsymbol{\pi}_i^T$, yielding the *distance measurement constraint*

$$z_2 = {}^G\boldsymbol{\pi}_i^T \left({}^G\mathbf{p}_I + \mathbf{C}^T({}^I\bar{q}_G) \left({}^I\mathbf{p}_L + \rho\, {}^I\boldsymbol{\ell}\right)\right) - d_i = 0. \tag{2.26}$$

The expected measurement is

$$\hat{z}_2 = {}^G\boldsymbol{\pi}_i^T \left({}^G\hat{\mathbf{p}}_I + \mathbf{C}^T({}^I\hat{\bar{q}}_G) \left({}^I\mathbf{p}_L + \rho_m {}^I\boldsymbol{\ell}_m\right)\right) - \hat{d}_i. \tag{2.27}$$

The measurement residual is $r_2 = z_2 - \hat{z}_2 = -\hat{z}_2$ and the corresponding linearized error model is

$$\widetilde{z}_2 \simeq \begin{bmatrix} -{}^G\boldsymbol{\pi}_i^T \mathbf{C}^T\left({}^I\hat{\bar{q}}_G\right) \lfloor {}^I\mathbf{p}_L + \rho_m {}^I\boldsymbol{\ell}_m \times \rfloor & \mathbf{0}_{1\times 9} & {}^G\boldsymbol{\pi}_i^T \end{bmatrix} \widetilde{\mathbf{x}}_s + \begin{bmatrix} \mathbf{0}_{1\times(i-1)} & -1 & \mathbf{0}_{1\times(N-i)} \end{bmatrix} \widetilde{\mathbf{x}}_d$$
$$+ \begin{bmatrix} -{}^G\boldsymbol{\pi}_i^T \mathbf{C}^T\left({}^I\hat{\bar{q}}_G\right) \rho_m {}^I\boldsymbol{\ell}_m^{\perp} & {}^G\boldsymbol{\pi}_i^T \mathbf{C}^T\left({}^I\hat{\bar{q}}_G\right) {}^I\boldsymbol{\ell}_m \end{bmatrix} \begin{bmatrix} \tilde{\phi} \\ \tilde{\rho} \end{bmatrix}$$
$$= \mathbf{h}_{2,s}^T \widetilde{\mathbf{x}}_s + \mathbf{h}_{2,d}^T \widetilde{\mathbf{x}}_d + \boldsymbol{\gamma}_2^T \mathbf{n}_\ell, \tag{2.28}$$

where the vectors $\mathbf{h}_{2,s}^T$, $\mathbf{h}_{2,d}^T$, and $\boldsymbol{\gamma}_2^T$ are the Jacobians of (2.26) with respect to the state and line parameters, respectively.

We process the two measurement constraints together; stacking (2.24) and (2.28), we obtain the measurement Jacobians

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_{1,s}^T & \mathbf{h}_{1,d}^T \\ \mathbf{h}_{2,s}^T & \mathbf{h}_{2,d}^T \end{bmatrix} \quad , \quad \boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\gamma}_1^T \\ \boldsymbol{\gamma}_2^T \end{bmatrix}, \tag{2.29}$$

which are used in the expression for the Kalman gain

$$\mathbf{K} = \mathbf{P}_{k+1|k}\mathbf{H}^T \left(\mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T + \boldsymbol{\Gamma}\mathbf{R}\boldsymbol{\Gamma}^T\right)^{-1}. \tag{2.30}$$

The residual vector is $\mathbf{r} = \begin{bmatrix} r_1 & r_2 \end{bmatrix}^T$, and the state and the covariance update equations are

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}\mathbf{r} \tag{2.31}$$

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_{k+1|k}(\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}\boldsymbol{\Gamma}\mathbf{R}\boldsymbol{\Gamma}^T\mathbf{K}^T. \tag{2.32}$$

After updating the state and covariance with measurements to planes currently in the map, we may have additional measurements to process corresponding to planes that have not been observed previously. In Section 2.3.3 we describe how to augment the map with an initial estimate of each new feature.

2.3.3 Landmark Initialization

There are three cases which we distinguish for plane initialization. The first is planes which are known perfectly *a priori* (e.g., from "as-built" building blueprints). The second class are planes which are approximately known (e.g., extracted from imprecise building blueprints, or "as-designed"). The third type are the unknown planes that occur in the principal building directions (i.e., the floor, ceiling, and orthogonal building walls). While we do not know the location or number of these planes, whenever we observe them, we know they exhibit one of the three known principle orientations, and only the distance to the plane must be estimated.

Perfectly known planes

Perfectly known planes are straight forward to exploit in our navigation framework since all three d.o.f. of the plane parameters are known *a priori*. We could include each plane

in the state vector with an associated zero-covariance and zero-correlation to the rest of the state. However, in practice we simply maintain an additional parameter vector of known planes, which reduces the computational cost of the filter by limiting the state size even further. When observing a perfectly known plane, we follow the procedure in Section 2.3.2 to update the state, with the caveat that the Jacobians taken with respect to the known plane parameters are set identically to zero [see (2.24) and (2.28)].

Approximately known planes

Planes which are known approximately are the most common to arise in typical implementations when a blueprint of the building is available. This occurs because for practical reasons during building construction, walls are not always placed precisely where they were designated and building tolerances permit some room for error. In these instances, we include an initial estimate of each building plane in the map, and we set the covariance for each plane according to the accuracy of the blueprints. In practice, if the quality of the blueprints is unknown, it suffices to hand-measure a small number of the building planes in order to characterize the blueprint accuracy. We assume that the errors in the initial estimates of the approximately known planes are uncorrelated with each other and the sensor platform state, and set the corresponding cross-correlation entries in $\mathbf{P}$ to zero.

Unknown planes

When measuring a new plane, Π_{N+1}, we first determine if the plane's orientation, ${}^{G}\boldsymbol{\pi}_{N+1}$, corresponds to one of the three cardinal directions, $\mathbf{e}_j$, $j = 1, 2, 3$, considered in the map. We employ a Mahalanobis distance test to measure the probability of correspondence between the plane's orientation and each of the cardinal directions in the map. Specifically, we compute the orientation residual $r_{1,j} = -\mathbf{e}_j^T \mathbf{C}^T\big({}^{I}\hat{\bar{q}}_G\big)\, {}^{I}\boldsymbol{\ell}_m^{\perp}$, $j = 1, 2, 3$, and the covariance of the residual

$$s_j = \begin{bmatrix} \mathbf{h}_{1,s}^T & \mathbf{h}_{1,d}^T \end{bmatrix} \mathbf{P}_{k+1|k} \begin{bmatrix} \mathbf{h}_{1,s} \\ \mathbf{h}_{1,d} \end{bmatrix} + \sigma_\phi^2 \boldsymbol{\gamma}_1^T \boldsymbol{\gamma}_1, \tag{2.33}$$

where $\mathbf{h}_{1,s}$ and $\boldsymbol{\gamma}_1$ are the measurement Jacobians defined in (2.24) evaluated at ${}^G\boldsymbol{\pi}_i = \mathbf{e}_j$, and σ_ϕ^2 is the $(1,1)$ element of $\mathbf{R}$. If the smallest Mahalanobis distance

$$\mu_{jmin}^2 = \min_j \frac{r_{1,j}^2}{s_j} \tag{2.34}$$

is less than a probabilistic threshold, then a new landmark is initialized with normal vector ${}^G\boldsymbol{\pi}_{N+1} = \mathbf{e}_{jmin}$. After determining the new plane's orientation, we compute the distance to the new plane by solving (2.27) for $\hat{d}_{N+1}$, i.e.,

$$\hat{d}_{N+1} = {}^G\boldsymbol{\pi}_{N+1}^T \left({}^G\hat{\mathbf{p}}_I + \mathbf{C}^T({}^I\hat{\bar{q}}_G) \left({}^I\mathbf{p}_L + \rho_m {}^I\boldsymbol{\ell}_m \right) \right) \tag{2.35}$$

and augment the state vector as $\hat{\mathbf{x}}^{aug} \triangleq \begin{bmatrix} \hat{\mathbf{x}}^T & | & \hat{d}_{N+1} \end{bmatrix}^T$. Next, we need to augment the filter's covariance, which requires first partitioning the prior covariance into

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P}_{ss} & \mathbf{P}_{sd} \\ \mathbf{P}_{ds} & \mathbf{P}_{dd} \end{bmatrix}, \tag{2.36}$$

where $\mathbf{P}_{ss}$ is the 15×15 sensor error-state covariance, $\mathbf{P}_{dd}$ is the $N \times N$ map error-state covariance, and $\mathbf{P}_{sd} = \mathbf{P}_{ds}^T$ are the $15 \times N$ cross-correlation components. We then compute the scalar variance of the new plane, $\mathbf{P}_{d'd'}$, and the correlation between the new plane and the current state, $\mathbf{P}_{d'\mathbf{x}}$, as:

$$\mathbf{P}_{d'd'} = \mathbf{h}_{2,s}^T \mathbf{P}_{ss} \mathbf{h}_{2,s} + \boldsymbol{\gamma}_2^T \mathbf{R} \boldsymbol{\gamma}_2 \tag{2.37}$$

$$\mathbf{P}_{d'\mathbf{x}} = \mathbf{P}_{\mathbf{x}d'}^T = \begin{bmatrix} \mathbf{h}_{2,s}^T \mathbf{P}_{ss} & \mathbf{h}_{2,s}^T \mathbf{P}_{sd} \end{bmatrix} \tag{2.38}$$

where $\mathbf{h}_{2,s}$ and $\boldsymbol{\gamma}_2$ are defined in (2.28). Lastly, the augmented covariance, $\mathbf{P}^{aug}$, is computed as:

$$\mathbf{P}^{aug} = \begin{bmatrix} \mathbf{P}_{k+1|k} & \mathbf{P}_{\mathbf{x}d'} \\ \mathbf{P}_{d'\mathbf{x}} & \mathbf{P}_{d'd'} \end{bmatrix}. \tag{2.39}$$

After performing state and covariance augmentation during the landmark initialization step, we return to the propagation step and process the next IMU measurement (see Section 2.3.1).

2.3.4 Zero-Velocity Update

When the laser scanner does not detect any structural planes along certain directions for an extended period of time, the pose estimates accumulate errors due to drifts in the accelerometer and gyroscope biases. In addition, build up of orientation errors can cause the filter to incorrectly integrate a portion of the gravitational acceleration. This effect is closely related to the system's observability (see Section 2.5) and can be compensated by means of drift correction during instantaneous stationary periods of the motion (e.g., when a shoe-mounted IMU is stationary during the stance phase while walking, see [33]).

This procedure, termed a *zero-velocity update*, is challenging for two reasons: (i) the stationary periods must be identified without an *external reference*, and (ii) the IMU drift error must be corrected while properly accounting for the state uncertainty and IMU noise. Existing methods typically detect stationary periods based on a threshold check for the accelerometer measurement. These require significant hand tuning, and cannot account for the uncertainty in the current state estimate.

In contrast, we formulate the zero-velocity constraint as an EKF measurement and use the Mahalanobis distance test to identify the stationary intervals. Specifically, for the zero-velocity update, we employ the following measurement constraints for the linear acceleration, and linear and rotational velocities which are (instantaneously) equal to zero

$$\mathbf{z}_\zeta = \begin{bmatrix} \mathbf{a}^T & \boldsymbol{\omega}^T & {}^G\mathbf{v}_I^T \end{bmatrix}^T = \mathbf{0}_{9\times 1}. \tag{2.40}$$

The zero-velocity measurement residual is

$$\mathbf{r}_\zeta = \mathbf{z}_\zeta - \hat{\mathbf{z}}_\zeta = \begin{bmatrix} \mathbf{a}_m - \hat{\mathbf{b}}_a + \mathbf{C}\left({}^I\hat{\bar{q}}_G\right){}^G\mathbf{g} \\ \boldsymbol{\omega}_m - \hat{\mathbf{b}}_g \\ -{}^G\hat{\mathbf{v}}_I \end{bmatrix} \tag{2.41}$$

and the corresponding linearized error model is

$$\begin{aligned} \widetilde{\mathbf{z}}_\zeta \simeq & \begin{bmatrix} -\lfloor \mathbf{C}\left({}^I\hat{\bar{q}}_G\right){}^G\mathbf{g}\times \rfloor & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix} \widetilde{\mathbf{x}} + \begin{bmatrix} \mathbf{n}_a \\ \mathbf{n}_g \\ \mathbf{n}_v \end{bmatrix} \\ = & \mathbf{H}_\zeta\, \widetilde{\mathbf{x}} + \mathbf{n}_\zeta\,, \end{aligned} \tag{2.42}$$

where $\mathbf{H}_\zeta$ is the Jacobian of the zero-velocity measurement with respect to the state, and $\mathbf{n}_\zeta$ is a zero-mean, white Gaussian process noise that acts as a regularization term for computing the inverse of the measurement residuals' covariance. Based on this update model, at time step k we compute the Mahalanobis distance $\chi^2 = \mathbf{r}_\zeta^T \mathbf{S}_k^{-1} \mathbf{r}_\zeta$, where $\mathbf{S}_k = \mathbf{H}_\zeta \mathbf{P}_{k|k} \mathbf{H}_\zeta^T + \mathbf{R}_\zeta$ is the covariance of the measurement residual and $\mathbf{R}_\zeta = E\{\mathbf{n}_\zeta \mathbf{n}_\zeta^T\}$ is the measurement noise covariance. If χ^2 is less than a chosen probabilistic threshold, a stationary interval is detected and the state vector and the covariance matrix are updated using (2.40)-(2.42). We note that once we use the inertial measurements for an update, we cannot use them for propagation. However, this is not an issue, since the IMU is static and we do not need to use the kinematic model (2.2)-(2.7) to propagate the state estimates. Instead we employ the following equations:

$$
\begin{aligned}
{}^I_G\dot{\bar{q}}(t) &= \mathbf{0}_{4\times 1} && (2.43)\\
{}^G\dot{\mathbf{p}}_I(t) &= \mathbf{0}_{3\times 1} && (2.44)\\
{}^G\dot{\mathbf{v}}_I(t) &= \mathbf{0}_{3\times 1} && (2.45)\\
\dot{\mathbf{b}}_g(t) &= \mathbf{n}_{wg}(t) && (2.46)\\
\dot{\mathbf{b}}_a(t) &= \mathbf{n}_{wa}(t). && (2.47)
\end{aligned}
$$

In essence, this static-IMU propagation model indicates that the state vector and the covariance matrix of all components are kept constant. The only exceptions are the covariances of the errors in the gyroscope and accelerometer bias estimates which increase at each time step to reflect the effect of the random walk model.

2.4 Filter State Initialization

Before using the EKF to fuse measurements from the laser scanner and the IMU, we need to initialize the state vector estimate $\widehat{\mathbf{x}}_{0|0}$ along with its covariance $\mathbf{P}_{0|0}$. This is performed in three sequential stages: (i) the gyroscopes' biases, $\mathbf{b}_g$, are initialized using the *partial* zero-velocity updates (Section 2.4.1), (ii) the IMU orientation, ${}^I\bar{q}_G$, is initialized employing the laser scans (Section 2.4.2), and (iii) the accelerometers' biases, $\mathbf{b}_a$, are initialized using zero-velocity updates (Section 2.4.3). Once these three stages are completed, we initialize the position of the sensing platform, ${}^G\mathbf{p}_I$. However, we note that if there are no structural planes in the building map initially known (i.e., if no

blue print was provided), we can arbitrarily select the origin of the global frame. Thus, for our convenience, we set the origin of the global frame to coincide with the origin of the initial IMU frame, i.e., ${}^{G}\mathbf{p}_I = \mathbf{0}_{3\times 1}$. The initial covariance for ${}^{G}\widetilde{\mathbf{p}}_I$ is set to zero accordingly.

2.4.1 Gyroscopes' Biases Initialization

The *complete* zero-velocity update described in Section 2.3.4 cannot be directly applied to initialize the gyroscope biases. This is due to the fact that an estimate of the orientation ${}^{I}\bar{q}_G$, required for evaluating $\mathbf{H}_\zeta$ [see (2.42)], cannot be obtained before estimating the gyroscope biases (Section 2.4.2). Instead, to provide an initial estimate for the gyroscope biases, $\mathbf{b}_g$, we use *partial* zero-velocity updates. In particular, we initially set $\hat{\mathbf{b}}_g$ to an arbitrary value (e.g., zero), while its covariance is set to a large value, reflecting the lack of *a priori* knowledge about the estimates. Then, we keep the IMU static (i.e., $\boldsymbol{\omega} = \mathbf{0}_{3\times 1}$) and use the second block row of (2.40)-(2.42) to perform a partial zero-velocity update. This process is equivalent to averaging the (static) gyroscope measurements to compute an initial estimate of the bias.

2.4.2 Orientation Initialization

Since the IMU and the laser scanner are rigidly connected and their relative transformation is known (see Section 2.6), the initial orientation of the IMU can be directly computed from the initial orientation of the laser scanner. We describe two methods to compute the orientation of the laser scanner using line measurements of three planes with linearly-independent normal vectors. The first method, adapted from [47], requires observation of all three planes from the same viewpoint, while the second method is capable of using laser scan measurements taken from different perspectives by exploiting the motion information from the gyroscopes.

Concurrent observation of three planes

When three non-parallel planes are scanned from the same viewpoint (i.e., the same frame of reference), the estimate of the orientation ${}^{I}\bar{q}_G$ is initialized using the method of [47]. In this case, three quadratic constraints in terms of the unit quaternion ${}^{I}\bar{q}_G$

are obtained from the laser scans [see (2.22)], each of them describing the relationship between a line measurement and the corresponding plane:

$$z_{1,i} = {}^{G}\boldsymbol{\pi}_i^T \, \mathbf{C}^T({}^{I}\bar{q}_G) \, {}^{I}\boldsymbol{\ell}_i^{\perp} = 0, \quad i = 1, \ldots, 3. \tag{2.48}$$

Chen's algorithm algebraically manipulates the rotation matrix to convert this system of equations to an eighth-order univariate polynomial in one of the d.o.f. of the unknown rotation. Eight solutions for this univariate polynomial are obtained, for example, using the Companion matrix [48]. The remaining two d.o.f. of the rotation, ${}^{I}\bar{q}_G$, are subsequently determined by back-substitution. In general, an external reference is required to identify the true solution from the eight possibilities. In our work, we employ the gravity measurement from the accelerometers and the planes' identities to find the unique solution.

Motion-aided orientation initialization

In order to use Chen's method for initializing the orientation, all three line measurements must be expressed with respect to the same frame of reference; hence three non-parallel planes must be concurrently observed by the laser scanner from the same viewpoint. However, satisfying this prerequisite is quite limiting since it requires facing a corner of a room, for example, where three structural planes intersect. In this work, we address this issue by using the gyroscope measurements to *transform* the laser scans, taken from different viewpoints at different time instants, to a *common frame of reference*. We choose as the common frame, the IMU frame when the first laser scan is recorded (i.e., at time t_1), and denote it by $\{I_1\}$. In this way, we can rewrite the inferred measurement constraints (2.22) at time t_j, $j = 2, 3$ as

$${}^{G}\boldsymbol{\pi}_j^T \mathbf{C}^T({}^{I}\bar{q}_G(t_j)) {}^{I}\boldsymbol{\ell}_j^{\perp}(t_j) = {}^{G}\boldsymbol{\pi}_j^T \mathbf{C}^T({}^{I}\bar{q}_G(t_1)) {}^{I_1}\boldsymbol{\ell}_j^{\perp}(t_j) = 0 \tag{2.49}$$

where ${}^{I_1}\boldsymbol{\ell}_j^{\perp}(t_j) = \mathbf{C}({}^{I_1}\bar{q}_{I_j}) {}^{I}\boldsymbol{\ell}_j^{\perp}(t_j)$ is the line direction corresponding to the plane Π_j, recorded at time t_j, and transformed to the frame $\{I_1\}$. Since the gyroscope biases are already initialized, the quaternions ${}^{I_1}\bar{q}_{I_j}$ can be obtained by integrating the rotational velocity measurements [see (2.8) and (2.10)] between time instants t_1 and t_j. Once all the line directions, ${}^{I}\boldsymbol{\ell}_j^{\perp}(t_j)$, are expressed with respect to $\{I_1\}$, we employ Chen's algorithm, described before, to find the initial orientation, ${}^{I}\bar{q}_G(t_1)$.

The covariance of the initial orientation estimate is obtained by computing the corresponding Jacobians [by linearizing (2.49)] and using the uncertainty (covariance) in the estimates of ${}^{I}\boldsymbol{\ell}_j^{\perp}$ and ${}^{I_1}\bar{q}_{I_j}$. However, note that the estimates of the relative transformations ${}^{I_1}\bar{q}_{I_2}$ and ${}^{I_1}\bar{q}_{I_3}$ are correlated. To account for these correlations, we employ the *stochastic cloning* technique [49] to augment the state vector and the covariance matrix of the EKF with ${}^{I_1}\bar{q}_{I_j}$ at time t_j (assuming we have started integrating from time t_1). In this way, we are able to estimate the IMU orientation by integrating the gyroscope measurements, and concurrently compute the correlations between the IMU orientation estimates at the time instants when laser scans are recorded.

2.4.3 Accelerometers' Biases Initialization

In this step, similar to the gyroscope bias initialization, we set the estimate for the accelerometer biases, $\mathbf{b}_a$, to an arbitrary value (e.g., zero), and set its covariance to a sufficiently large value, representing our uncertainty about the arbitrary initial estimate. Since the IMU is initially static, we set the velocity estimate, ${}^{G}\mathbf{v}_I$, and its covariance to zero. Then, keeping the IMU static, we utilize the *complete* zero-velocity update described in Section 2.3.4 to initialize the accelerometer biases.

2.5 Observability Analysis

A key task when designing any estimator is to study the observability properties of the underlying system, to determine if the available measurements will provide enough information to estimate the state. In this section, we prove that the presented system for IMU-laser scanner localization is observable when three known planes (i.e., available from the "as-built" or "as-designed" blueprints), whose normal vectors are linearly independent, are concurrently observed by the laser scanner. Under this condition, which is fulfilled in most practical scenarios (e.g., if the scan plane intersects two walls and the floor), we can employ the pose estimation method described in Section 2.4 to estimate $({}^{G}\mathbf{p}_I, {}^{I}\bar{q}_G)$. For the purpose of observability analysis, we introduce two new *inferred measurements* $\mathbf{h}_1^*$ and $\mathbf{h}_2^*$ that replace the laser scan measurements (2.22),

(2.26):

$$ {}^{I}\bar{q}_{G} = \mathbf{h}_1^*(\mathbf{x}) = \boldsymbol{\xi}_1({}^{I}\boldsymbol{\ell}_1, {}^{I}\boldsymbol{\ell}_2, {}^{I}\boldsymbol{\ell}_3) \tag{2.50} $$

$$ {}^{G}\mathbf{p}_{I} = \mathbf{h}_2^*(\mathbf{x}) = \boldsymbol{\xi}_2({}^{I}\boldsymbol{\ell}_1, {}^{I}\boldsymbol{\ell}_2, {}^{I}\boldsymbol{\ell}_3). \tag{2.51} $$

The two functions $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ in (2.50) and (2.51) do not need to be known explicitly; only their functional relation with the random variables, ${}^{I}\bar{q}_{G}$ and ${}^{G}\mathbf{p}_{I}$, is required for the observability analysis. Our approach uses the Lie derivatives [50] of the above inferred measurements (2.50) and (2.51) for the system in (2.2)-(2.7), to show that the corresponding observability matrix is full rank. For this purpose, we first rearrange the nonlinear kinematic equations (2.2)-(2.7) in a suitable form for computing the Lie derivatives:

$$ \begin{bmatrix} {}^{I}\dot{\bar{q}}_{G} \\ \dot{\mathbf{b}}_g \\ {}^{G}\dot{\mathbf{v}}_{I} \\ \dot{\mathbf{b}}_a \\ {}^{G}\dot{\mathbf{p}}_{I} \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{1}{2}\boldsymbol{\Xi}({}^{I}\bar{q}_{G})\mathbf{b}_g \\ \mathbf{0}_{3\times1} \\ {}^{G}\mathbf{g} - \mathbf{C}^T({}^{I}\bar{q}_{G})\mathbf{b}_a \\ \mathbf{0}_{3\times1} \\ {}^{G}\mathbf{v}_{I} \end{bmatrix}}_{\mathbf{f}_0} + \underbrace{\begin{bmatrix} \frac{1}{2}\boldsymbol{\Xi}({}^{I}\bar{q}_{G}) \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \end{bmatrix}}_{\underline{\mathbf{f}}_1} \boldsymbol{\omega}_m + \underbrace{\begin{bmatrix} \mathbf{0}_{4\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{C}^T({}^{I}\bar{q}_{G}) \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \end{bmatrix}}_{\underline{\mathbf{f}}_2} \mathbf{a}_m \,, \tag{2.52} $$

where $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ are considered the control inputs, and

$$ \boldsymbol{\Xi}(\bar{q}) \triangleq \begin{bmatrix} q_4\mathbf{I}_3 + \lfloor \mathbf{q} \times \rfloor \\ -\mathbf{q}^T \end{bmatrix} \quad \text{with} \quad \bar{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix}. \tag{2.53} $$

Note also that $\mathbf{f}_0$ is a 16×1 vector, while $\underline{\mathbf{f}}_1$ and $\underline{\mathbf{f}}_2$ are matrices of dimensions 16×3.

In order to prove that the system is locally weakly observable, it *suffices* to show that the observability matrix, whose rows comprise the gradients of the Lie derivatives of the measurements $\mathbf{h}_1^*$ and $\mathbf{h}_2^*$ with respect to $\mathbf{f}_0$, $\underline{\mathbf{f}}_1$, and $\underline{\mathbf{f}}_2$ [see (2.52)], is full rank [50]. Since the measurement and kinematic equations describing the IMU-laser scanner localization are infinitely smooth, the observability matrix has an infinite number of rows. However, to prove it is full rank, it suffices to show that a subset of its rows are linearly independent. The following matrix contains one such subset of rows whose linear

independence can be easily shown using block Gaussian elimination [51]:

$$\begin{bmatrix} \nabla \mathfrak{L}^0_{\mathbf{f}_0} \mathbf{h}^*_1 \\ \nabla \mathfrak{L}^0_{\mathbf{f}_0} \mathbf{h}^*_2 \\ \nabla \mathfrak{L}^1_{\mathbf{f}_0} \mathbf{h}^*_1 \\ \nabla \mathfrak{L}^1_{\mathbf{f}_0} \mathbf{h}^*_2 \\ \nabla \mathfrak{L}^2_{\mathbf{f}_0} \mathbf{h}^*_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_4 & \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} \\ \mathbf{0}_{3\times 4} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 \\ \mathbf{X}_1 & -\frac{1}{2}\mathbf{\Xi}({}^I\bar{q}_G) & \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} \\ \mathbf{0}_{3\times 4} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{X}_2 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{C}^T({}^I\bar{q}_G) & \mathbf{0}_{3\times 3} \end{bmatrix}.$$

In this matrix, $\mathfrak{L}^i_{\mathbf{f}_0} \mathbf{h}^*_j(\mathbf{x})$ denotes the i-th order Lie derivative of $\mathbf{h}^*_j(\mathbf{x})$ with respect to $\mathbf{f}_0$. The matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ have dimensions 4×4 and 3×4, respectively, and do not need to be computed explicitly since they will be eliminated by the block element $(1, 1)$ of the matrix, i.e., the identity matrix $\mathbf{I}_4$. Since $\mathbf{\Xi}(\bar{q})$ and $\mathbf{C}(\bar{q})$ are always full rank for any unit quaternion $\bar{q}$ [51], all the rows of the above matrix are linearly independent. Hence, we conclude the observability analysis with the following lemma:

Lemma 1 *Given line measurements corresponding to three known planes with linearly independent normal vectors, the system describing the IMU-laser scanner localization is locally weakly observable.*

Simply put, as long as the laser scanner measures the walls, as well as the floor or ceiling, the filter should be able to maintain an accurate estimate the pose of the person. As the person moves through the environment, the laser scanner measures different planes over time, leading to higher accuracy estimates. When the sensing platform stops moving, we can apply zero velocity updates (see Section 2.3.4), to reduce drift.

2.6 IMU-Laser Scanner Extrinsic Calibration

The laser scan measurements must be transformed to the IMU frame before an EKF update can be performed. In particular, in the orientation constraint (2.22), the measured line direction ${}^L\boldsymbol{\ell}^\perp$ that is registered in the laser scan frame, is expressed with respect to the IMU frame, $\{I\}$. Similarly, in the distance constraint (2.26), the perpendicular vector to the line direction, ${}^L\boldsymbol{\ell}$, is first transformed to the IMU frame. To perform these transformations, we need to know $({}^I\bar{q}_L, {}^I\mathbf{p}_L)$, i.e., the rotation and translation between the IMU frame and the laser frame. If the transformation between the IMU and the laser scanner is not precisely known, the constraints (2.22) and (2.26) will not hold,

and updating the filter based on them can result in inconsistency and divergence of the estimator.

Some methods exist in the literature for extrinsic laser scanner calibration (e.g., [52, 53]), however, these have primarily focused on recovering the relative orientation of the sensor (i.e., roll, pitch, and yaw angles), and utilize GPS as an additional aid in the calibration process. In contrast, we seek to compute the frame transformation between the laser and IMU using only the sensors' own motion and measurements to planes in the environment.

To address this issue, we have employed a method similar to our previous work for IMU-camera calibration [54] to calibrate the transformation between the IMU and the laser scanner. For this purpose, we have included $({}^I\bar{q}_L, {}^I\mathbf{p}_L)$ in the state vector of the EKF, i.e.,

$$\begin{aligned}\mathbf{x}^{aug} &= \begin{bmatrix} {}^I_G\bar{q}^T & \mathbf{b}_g^T & {}^G\mathbf{v}_I^T & \mathbf{b}_a^T & {}^G\mathbf{p}_I^T & | & {}^I_L\bar{q}^T & {}^I\mathbf{p}_L^T & | & d_1 & \cdots & d_N \end{bmatrix}^T \\ &= \begin{bmatrix} \mathbf{x}_s^T & | & \mathbf{x}_c^T & | & \mathbf{x}_d^T \end{bmatrix}^T . \end{aligned} \tag{2.54}$$

We augment the system equations (2.2)-(2.7) with

$${}^I\dot{\bar{q}}_L^T = 0 \quad , \quad {}^I\dot{\mathbf{p}}_L^T = 0 \tag{2.55}$$

which specify that the IMU-laser transformation is rigid and does not change with time. We also extend (2.24) and (2.28) to include the corresponding Jacobians with respect to the ${}^I\mathbf{p}_L$ and ${}^I\bar{q}_L$. We do so by first writing the orientation and distance constraints explicitly in terms of the laser-to-IMU transformation parameters $({}^I\bar{q}_L, {}^I\mathbf{p}_L)$, i.e.,

$$z_1 = {}^G\boldsymbol{\pi}_i^T \, \mathbf{C}^T({}^I\bar{q}_G) \, \mathbf{C}({}^I\bar{q}_L) \, {}^L\boldsymbol{\ell}^{\perp} = 0 \tag{2.56}$$

$$z_2 = {}^G\boldsymbol{\pi}_i^T \left({}^G\mathbf{p}_I + \mathbf{C}^T({}^I\bar{q}_G) \left({}^I\mathbf{p}_L + \rho \, \mathbf{C}({}^I\bar{q}_L) {}^L\boldsymbol{\ell} \right) \right) - d_i = 0. \tag{2.57}$$

The linearized error models for (2.56) and (2.57) are

$$\begin{aligned}\widetilde{z}_1 &\simeq \mathbf{h}_{1,s}^T \, \widetilde{\mathbf{x}}_s + \mathbf{h}_{1,d}^T \, \widetilde{\mathbf{x}}_d + \boldsymbol{\gamma}_1^T \, \mathbf{n}_\ell + \begin{bmatrix} {}^G\boldsymbol{\pi}_i^T \mathbf{C}^T\left({}^I\hat{\bar{q}}_G\right) \lfloor \mathbf{C}\left({}^I\hat{\bar{q}}_L\right) {}^L\boldsymbol{\ell}_m^{\perp} \times \rfloor & \mathbf{0}_{1\times 3} \end{bmatrix} \widetilde{\mathbf{x}}_c \\ &= \mathbf{h}_{1,s}^T \, \widetilde{\mathbf{x}}_s + \mathbf{h}_{1,c}^T \, \widetilde{\mathbf{x}}_c + \mathbf{h}_{1,d}^T \, \widetilde{\mathbf{x}}_d + \boldsymbol{\gamma}_1^T \, \mathbf{n}_\ell, \end{aligned} \tag{2.58}$$

$$\begin{aligned}\widetilde{z}_2 &\simeq \mathbf{h}_{2,s}^T \, \widetilde{\mathbf{x}}_s + \mathbf{h}_{2,d}^T \, \widetilde{\mathbf{x}}_d + \boldsymbol{\gamma}_2^T \, \mathbf{n}_\ell + \begin{bmatrix} {}^G\boldsymbol{\pi}_i^T \mathbf{C}^T\left({}^I\hat{\bar{q}}_G\right) \lfloor \mathbf{C}\left({}^I\hat{\bar{q}}_L\right) \rho \, {}^L\boldsymbol{\ell}_m \times \rfloor & {}^G\boldsymbol{\pi}_i^T \mathbf{C}^T\left({}^I\hat{\bar{q}}_G\right) \end{bmatrix} \widetilde{\mathbf{x}}_c \\ &= \mathbf{h}_{2,s}^T \, \widetilde{\mathbf{x}}_s + \mathbf{h}_{2,c}^T \, \widetilde{\mathbf{x}}_c + \mathbf{h}_{2,d}^T \, \widetilde{\mathbf{x}}_d + \boldsymbol{\gamma}_2^T \, \mathbf{n}_\ell, \end{aligned} \tag{2.59}$$

where the calibration error-state is $\widetilde{\mathbf{x}}_c = \begin{bmatrix} {}^I\boldsymbol{\delta\theta}_L & {}^I\widetilde{\mathbf{p}}_L^T \end{bmatrix}^T$, the Jacobians with respect to the state and line parameters, $\mathbf{h}_{i,s}^T$, $\mathbf{h}_{i,d}^T$, $\boldsymbol{\gamma}_i^T$, $i = 1, 2$, are defined in (2.24) and (2.28), and the Jacobians with respect to the calibration parameters, $\mathbf{h}_{i,c}^T$, $i = 1, 2$ are implicitly defined in (2.58) and (2.59).

The key idea for IMU-laser calibration is to estimate the augmented state $\mathbf{x}^{aug}$ while in a known or unknown environment with at least three perpendicular walls. Note that since there is not enough information to estimate the calibration from a single viewpoint, we must employ a "motion-induced" calibration strategy. In particular, based on a Lie derivative analysis of the system observability properties (see Section 2.5, as well as [51, 54]), we have shown that the IMU-laser calibration parameters are observable when at least two rotations are performed about different axes, but we omit the details here for brevity. We move the sensor package and collect data until a satisfactory level of accuracy for the calibration parameters (based on the 3σ bounds computed from the estimated covariance matrix) has been achieved. The results of our on-line calibration process, obtained while exploring an unknown area, are presented in Section 2.7.3.

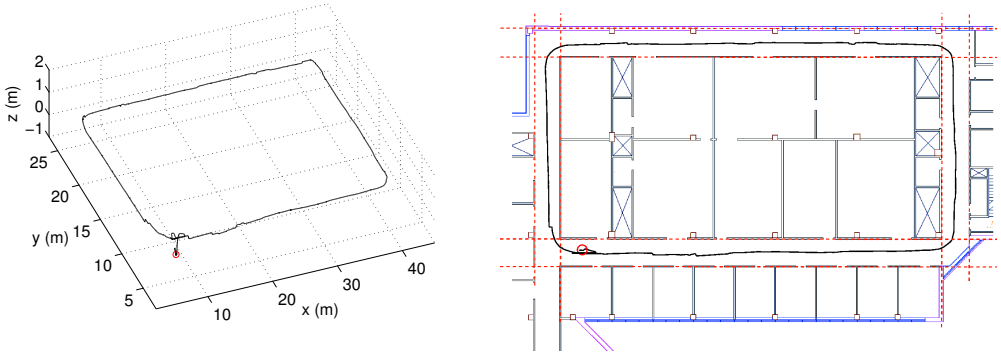
2.7 Experimental Results

Our proposed IMU-laser localization and mapping algorithm was evaluated with a sensing package comprised of a solid-state ISIS IMU operating at 100 Hz and a SICK LMS200 laser scanner operating at 10 Hz, mounted on a navigation box to log data. These sensors were interfaced to a laptop via RS-232 which recorded the time-stamped measurements. The data-logging software was implemented in C++, whereas the EKF was written in Matlab.

2.7.1 Navigation in a known environment

During the first experiment we tested the navigation algorithm in a known environment along a trajectory loop of 120 m in length.[3] The motion profile of the sensor platform contained instantaneous stationary time periods to allow for zero-velocity updates. These updates cause small reductions in the position estimate's covariance [see Fig. 2.5(a)]. Larger reductions in the covariance take place whenever the laser scanner

[3] Video available at `http://mars.cs.umn.edu/videos/IMU-Laser.m4v`



(a) (b)

Figure 2.2: (a) 3D view of the estimated trajectory. The sensing package was initially placed on the ground for the purpose of IMU-bias initialization, and subsequently picked up and carried in a clock-wise loop of 120 m in length through the building hallways. (b) Top-view of the estimated 3D trajectory during an 8.5 min experiment. The red circle indicates the starting position (on the floor), and the dashed red lines indicate the walls which were included in the building map.

detects three planes whose normal vectors are linearly independent (e.g., two perpendicular walls and the ceiling) within a short period of time; an event that typically occurs at hallway intersections (e.g., $t = 49$ sec). The *a priori* known map, available from the building blueprints, contained 9 walls and the ceiling. Employing this map, nearly $12,000$ measurement updates were performed during the 8.5 minute trial. The combination of the laser measurements and zero-velocity updates allowed the filter to maintain a precise pose estimate of the sensor platform. Specifically, the maximum uncertainty in the position estimates was 9.16 cm (1σ), while the maximum uncertainty in the attitude estimates was 0.1 deg (1σ) [see Fig. 2.5(a) and Fig. 2.5(b)]. The final position uncertainty was $\begin{bmatrix} 27.5 & 1.2 & 1.3 \end{bmatrix}$ cm (3σ). Note that the x-direction uncertainty is larger in the final corridor, since no planes are observed that provide information along the x-axis.

2.7.2 Navigation in a previously unknown environment

We conducted a second experiment in a previously unknown indoor environment, along a closed-loop path of approximately 270 m in length (see Fig. 2.4(a) and Fig. 2.4(b)). The 3D trajectory covered two floors of Akerman Hall at the University of Minnesota, which included traversing two stairways and a ramp. The environment contained a multitude of clutter (e.g., trash cans, open and closed doors, storage boxes, and furniture), as well as normal pedestrian traffic flow. Despite the large amount of non-planar objects observed by the laser scanner, our localization aid accurately captured the 3D layout of the building, which in turn enabled precise localization.

During the experiment, as in the known map case, the motion profile of the sensor platform contained instantaneous stationary time periods to allow for zero-velocity updates. These updates caused small reductions in the position estimates' covariance [see Fig. 2.5(a)]. Larger reductions in the covariance occurred whenever an estimated structural plane was re-detected (e.g., $t = 555$ sec, x-axis update). The trajectory was accurately tracked, with an average position uncertainty of 3.18 cm (1σ), and an average attitude uncertainty of 0.02 deg (1σ) [see Fig. 2.5(a) and Fig. 2.5(b)]. The final position uncertainty after loop closure was $\begin{bmatrix}2.29 & 6.84 & 0.43\end{bmatrix}$ cm (1σ). In addition to tracking the six-d.o.f. pose of the person, a map was constructed which contained 16 walls and the ceilings of both building levels (see Fig. 2.4(a) and Fig. 2.4(b)). The uncertainty of the least accurately estimated distance to a wall was 4.57 cm (1σ), whereas the average uncertainty for all planes was 1.51 cm (1σ). The quality of the map and trajectory estimates is due to more than $19,000$ measurement updates that were performed during the 13 minute trial.

2.7.3 Extrinsic laser-to-IMU calibration

We now present the results of our extrinsic laser-to-IMU calibration process. Following the procedure of Section 2.6, we augmented the state vector with the laser-to-IMU transformation $\{{}^{I}\bar{q}_{L}, {}^{I}\mathbf{p}_{L}\}$, and concurrently estimated these parameters while navigating in a previously unknown building (see Section 2.7.2). We note that calibration can be made more accurate and converge faster if completed during a separate initialization phase in an environment with perfectly known planes; however, our algorithm performs

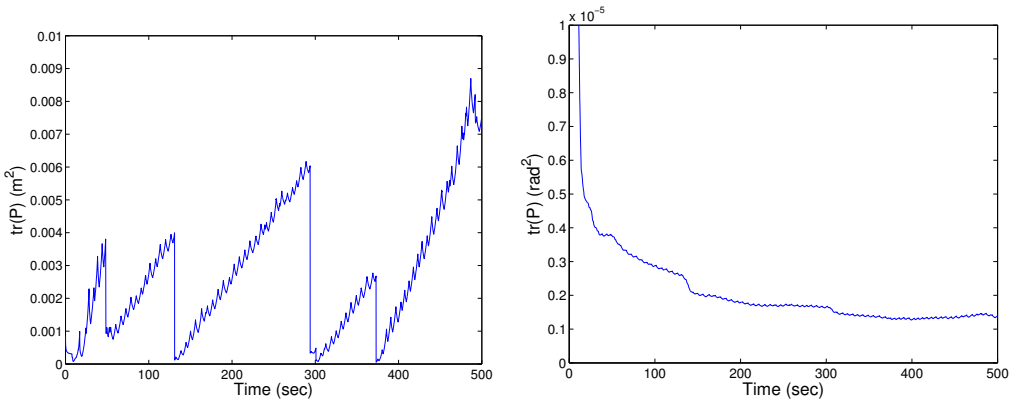
accurately in both scenarios.

Figures 2.6(a) and 2.6(b) depict the results for the position and orientation estimates, respectively. In order to demonstrate the consistency of the calibration process, we compute the error of the estimates with respect to the final estimate, along with the corresponding 3σ bounds. We note that since this is an experimental trial, it is impossible to know the true value of the rotation and translation between the laser and IMU; however, the obtained results match closely with the best estimates that we could achieve through hand-measured techniques. The estimated laser-to-camera translation vector was ${}^{I}\mathbf{p}_{L} = \begin{bmatrix} 25.91 & -3.13 & -13.42 \end{bmatrix}$ cm, and the estimated orientation was 177.44 deg in roll, 67.4 deg in pitch, and -2.29 deg in yaw, which we converted from quaternion to roll-pitch-yaw convention for ease of presentation. The most uncertain axis for position was 1.47 cm (3σ) along z, while the most uncertain axis for orientation was 0.11 deg (3σ) about y.

2.8 Summary

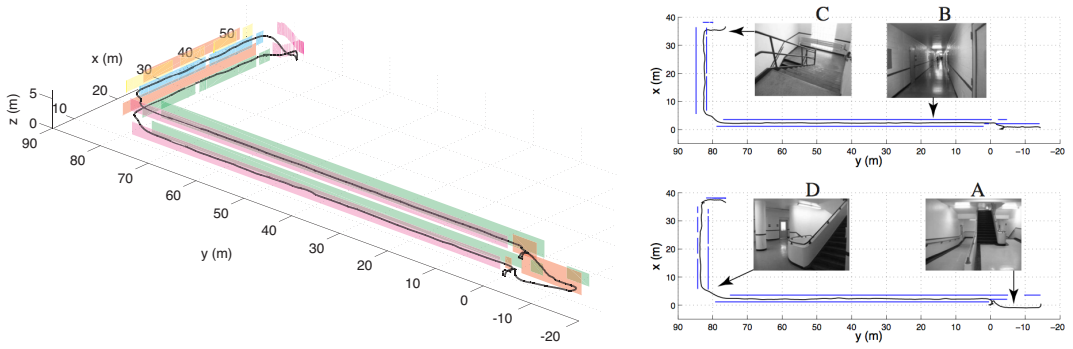
This chapter presented a novel LINS, based on a 2D laser scanner and an IMU, capable of 3D localization and mapping in indoor environments. In the proposed method, the orthogonal structural planes of the building are employed as landmarks to aid in localization. Since the building layout may be partially or completely unknown, the planes' parameters are estimated concurrently with the six-d.o.f. pose of the person. To this end, an EKF is utilized to fuse information from an IMU and a 2D laser scanner, and estimate the person's motion, and the building's structural planes. We presented a practical method for filter initialization using line-to-plane correspondences to initialize the orientation and zero-velocity updates to initialize the IMU bias estimates. Furthermore, we studied the observability properties of the system to determine a sufficient condition on the number and type of measurements so as to ensure the pose can be estimated. As a final contribution of this chapter, we proposed a laser-to-IMU calibration method which is capable of on-line estimation of the laser-to-IMU transformation. The validity of the proposed method is demonstrated in experimental trials in both previously known and unknown environments, which include challenging 3D building structures such as staircases, a disability access ramp, and long corridors. Furthermore, the environments

contained a typical amount of office clutter (e.g., chairs and desks) as well as pedestrian traffic.



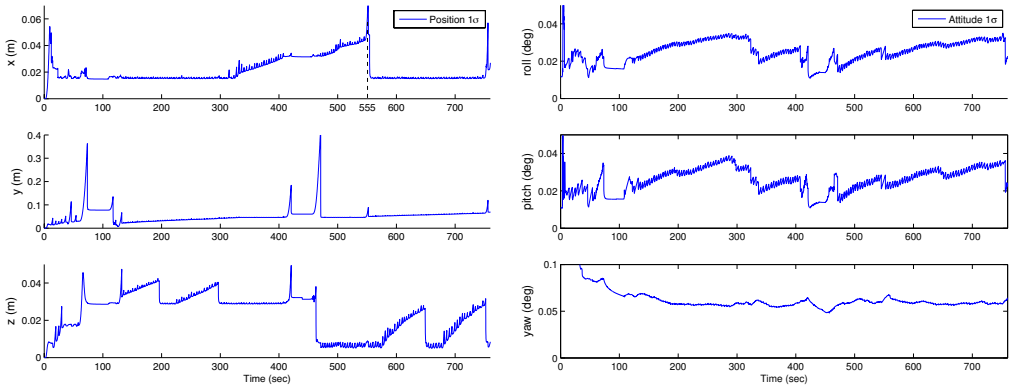
(a) (b)

Figure 2.3: (a) The trace of the position covariance. During the run, the maximum uncertainty along any axis was 9.16 cm. (1σ). (b) The trace of the attitude covariance. During the run, the maximum uncertainty about any axis was 0.1 deg. (1σ).



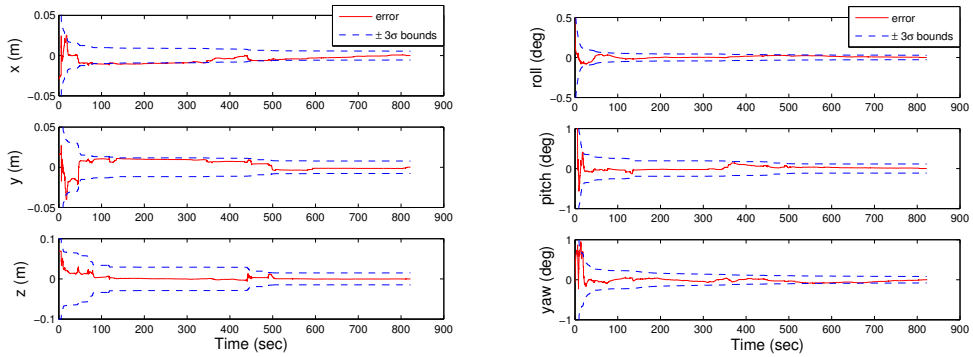
(a) (b)

Figure 2.4: (a) As the person walks with the sensing package, the filter estimates their 3D trajectory as well as a 3D representation of the unknown environment comprised of planar features. A side-view of the estimated 270 m trajectory is shown, which covers two floors of the building. The estimated walls on the first and second floors are depicted, but the estimated ceiling and floor planes have been omitted for clarity of presentation. (b) A top-view of the estimated 3D trajectory during the 13 min experiment. The total length of the trajectory is 270 m. The trajectory starts on the first floor (bottom figure), climbs up the disability ramp and the front stairs (picture A), and traverses the corridors (picture B) of the second floor clockwise (top figure). Subsequently, it descends back to the first floor on the second staircase (picture C), and traverses the first floor (bottom figure) counter clockwise, returning to the origin. Picture D shows the *curved* intersection of the two corridors where no wall was detected. The estimated walls are depicted in blue, and the ceiling and floor have been omitted for clarity of presentation.



(a) (b)

Figure 2.5: (a) The 1σ for the x, y, and z axes. During the run, the maximum uncertainty along any axis was 43.94 cm (1σ), while the average 1σ for the least accurate axis was 5.16 cm. (b) The 1σ for the roll, pitch, and yaw angles computed from the angle-error covariance. During the run, the maximum uncertainty about any axis was 0.06 deg. (1σ).



(a) (b)

Figure 2.6: (a) The relative-translation error (computed versus the final estimate) and the corresponding 3σ bounds for the laser-to-IMU translation vector. The final uncertainties were 0.54 cm along x, 0.76 cm along y, and 1.47 cm along z (3σ). The final translation estimate was ${}^{I}\mathbf{p}_{L} = [25.91 \quad -3.13 \quad -13.42]^{T}$ cm, which agrees with our best hand-measured estimates. (b) The relative-orientation error (computed versus the final estimate) and the corresponding 3σ bounds for the laser-to-IMU rotation ${}^{I}\bar{q}_{L}$. The final uncertainties were 0.02 deg in roll, 0.11 deg in pitch, and 0.08 deg in yaw (3σ). The final orientation estimate was 177.44 deg in roll, 67.4 deg in pitch, and -2.29 deg in yaw (converted from quaternion to roll-pitch-yaw convention), which agrees with our best hand-measured estimates.

Chapter 3

Observability-constrained Vision-aided Inertial Navigation

3.1 Introduction

Wide adoption of robotic technologies hinges on the ability of robots to freely navigate in our human-centric world. To do so, robots must maintain an accurate estimate of their six-degrees-of-freedom (d.o.f.) position and orientation (pose) as they navigate in 3D. Ideally, a localization algorithm should work seamlessly outdoors and indoors. This unfortunately prohibits reliance on GPS, since coverage is not available everywhere. For this reason, researchers have focused on designing localization methods that fuse onboard sensor data to estimate the robot's ego motion.

Tracking 3D motion can be accomplished by integrating the rotational velocity and linear acceleration signals provided by an Inertial Measurement Unit (IMU). However, due to integration of sensor noise and bias, pose estimates based on IMU data alone will quickly accumulate errors. To reduce the impact of these errors, so-called aided Inertial Navigation Systems (INS) have been proposed. Laser-aided INS (LINS) methods typically rely on the existence of structural planes [25] or height invariance in semi-structured spaces [55], and are not easily generalizable to cluttered or unstructured environments. On the other hand, Vision-aided INS (VINS) approaches, which fuse data from a camera and an IMU, can operate in both structured and unstructured areas. VINS methods have the additional benefit that both inertial and visual sensors are

lightweight, inexpensive (available in most mobile devices today), and passive, hence requiring a smaller power budget compared to LINS.

Existing work on VINS has employed a number of different estimators such as the Extended Kalman Filter (EKF) [22, 56, 49], the Unscented Kalman Filter (UKF) [57], and Batch-least Squares (BLS) [58]. Non-parametric estimators, such as the Particle Filter (PF), have also been used for visual-inertial odometry (e.g., [59, 60]). However, these have focused on the reduced problem of estimating a 2D robot pose, since the number of particles required is exponential in the size of the state vector. Within these works, a number of challenging issues have been addressed, such as reducing the computational cost of VINS [49, 61], dealing with delayed measurements [62], increasing the accuracy of feature initialization and estimation [63], and improving the robustness to estimator initialization errors [64]. Only limited attention, however, has been devoted to understanding how estimator inconsistency affects VINS.[1] The work described in this chapter, addresses this limitation through the following three main contributions:

- We introduce a novel methodology for identifying the unobservable modes of a nonlinear system. Contrary to previous methods [50] that require investigating an *infinite* number of Lie derivatives, our approach employs a factorization of the observability matrix, according to the observable and unobservable modes, and only requires computing a *finite* number of Lie derivatives.

- We apply our method to VINS and determine its unobservable directions, providing their analytical form as functions of the system states.

- We leverage our results to improve the consistency and accuracy of VINS, and extensively validate the proposed estimation framework both in simulations and real-world experiments.

The rest of this chapter is organized as follows: We begin with an overview of the related work (Section 3.2). In Section 3.3, we describe the system and measurement models used in VINS. Subsequently, we introduce our methodology for analyzing the observability properties of unobservable nonlinear systems (Section 3.4), which we leverage for determining the unobservable directions of the VINS model (Section 3.5). In

[1] As defined in [7], a state estimator is consistent if the estimation errors are zero-mean and have covariance equal to the one calculated by the filter.

Section 3.6, we present an overview of the analogous observability properties for the linearized system employed for estimation purposes, and show how linearization errors can lead to a violation of the observability properties, gain of spurious information, and estimator inconsistency. We propose an estimator modification to mitigate this issue in Section 3.6.3, and validate our algorithm in simulations and experimentally (Sections 3.7 and 3.8). Finally, we provide our concluding remarks and outline our future research directions in Section 3.9.

3.2 Related Work

The interplay between a nonlinear system's observability properties and the consistency of the corresponding linearized estimator has become a topic of increasing interest within the robotics community in recent years. Huang et al. [9, 10, 11] first studied this connection for 2D Simultaneous Localization and Mapping (SLAM) and extended their work to 2D cooperative multi-robot localization. In both cases, they proved that a mismatch exists between the number of unobservable directions of the true nonlinear system and the linearized system used for estimation purposes. Specifically, the estimated (linearized) system has one-fewer unobservable direction than the true system, allowing the estimator to surreptitiously gain spurious information along the direction corresponding to global orientation.

Extending this analysis to 3D VINS is a formidable task, most notably since the VINS system state has 15 d.o.f. instead of 3. Some authors have attempted to avoid this complexity by using abstract models (e.g., by assuming a 3D odometer that measures 6-d.o.f. pose displacements [65]); though these approaches cannot be easily extended to the VINS. The observability properties of VINS have been examined for a variety of scenarios. For example, Mirzaei and Roumeliotis [54] as well as Kelly and Sukhatme [66] have studied the observability properties of IMU-camera extrinsic calibration using Lie derivatives [50]. The former analysis, however, relies on known feature coordinates, while the latter employs an inferred measurement model (i.e., assuming the camera observes its pose in the world frame, up to scale), which requires a non-minimal set of visual measurements. This limitation is also shared by Weiss [67], who employs symbolic/numeric software tools, rather than providing an analytical study.

Jones and Soatto [63] investigated VINS observability by examining the indistinguishable trajectories of the system [68] under different sensor configurations (i.e., inertial only, vision only, vision and inertial). Their analysis, however, is restrictive due to: (i) the use of a stochastic tracking model (constant translational jerk and rotational acceleration), which cannot adequately describe arbitrary trajectories, and (ii) considering the IMU biases to be known, which is a limiting assumption. They conclude that the VINS state is observable, if the 6 d.o.f. of the first frame are fixed (e.g., by following the approach in [69]). As a result, their analysis does not provide an explicit form of the unobservable directions, nor does it fully characterize the observability properties of rotation (i.e., that yaw is unobservable).

Finally, Martinelli [70] used the concept of continuous symmetries to show that the IMU biases, 3D velocity, and absolute roll and pitch angles are observable for VINS. In this case, the unobservable directions are determined analytically for the special case of a single point feature located at the origin, but the unobservable directions for the case of multiple points are not provided. More importantly, however, among these VINS observability studies, no one has examined the link between observability and estimator consistency, or used their observability study to bolster estimator performance.

We presented preliminary results on VINS observability at the International Workshop on the Algorithmic Foundations of Robotics [71] and the International Symposium on Experimental Robotics [72], focussing on analytically showing the observability properties of the nonlinear system and the linearized system used for estimation purposes.

Li and Mourikis [73, 74] have also presented an investigation of estimator inconsistency utilizing linearized observability analysis of a bias-free VINS model. Based on their findings, they leveraged the First-Estimates Jacobian (FEJ) methodology of [9] to reduce the impact of inconsistency in Visual-Inertial Odometry (VIO). In contrast, our observability analysis encompasses both the nonlinear and linearized systems, using the full VINS state (i.e., including IMU biases). Furthermore, the implementation of our approach is more flexible since any linearization method can be employed (e.g., computing Jacobians analytically, numerically, or using sample points) by the estimator.

In what follows, we analytically determine the observability properties of both the

nonlinear VINS model and its linearized counterpart to prove that they have four unobservable degrees of freedom, corresponding to three-d.o.f. global translations and one-d.o.f. global rotation about the gravity vector. Then, we show that due to linearization errors, the number of unobservable directions is reduced in a standard EKF-based VINS approach, allowing the estimator to gain spurious information and leading to inconsistency. Finally, we propose a solution for reducing estimator inconsistency in VINS that is general, and can be directly applied in a variety of linearized estimation frameworks such as the EKF and UKF both for Visual SLAM (V-SLAM) and VIO.

3.3 VINS Estimator Description

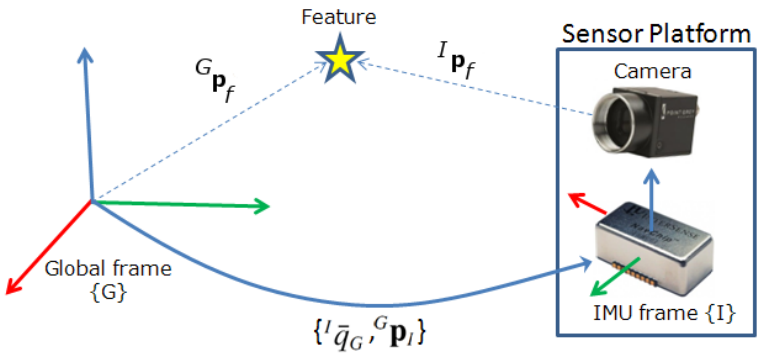


Figure 3.1: The pose of the camera-IMU frame $\{I\}$ with respect to the global frame $\{G\}$ is expressed by the position vector ${}^{G}\mathbf{p}_I$ and the quaternion of orientation ${}^{I}\bar{q}_G$. The observed feature is expressed in the global frame by its 3×1 position coordinate vector ${}^{G}\mathbf{p}_f$, and in the sensor frame by ${}^{I}\mathbf{p}_f = \mathbf{C}\left({}^{I}\bar{q}_G\right)\left({}^{G}\mathbf{p}_f - {}^{G}\mathbf{p}_I\right)$.

We begin with an overview of the VINS propagation and measurement models, and describe the EKF employed for fusing the camera and IMU measurements. In the following analysis, we consider the general case, in which the system state contains both the sensor platform state (i.e., pose, velocity, and IMU biases) and the observed features. However, it is important to note that the same analysis applies to VINS applications that do not explicitly estimate a map of the environment, such as VIO [49].

3.3.1 System State and Propagation Model

The system state comprises the IMU pose and linear velocity together with the time-varying IMU biases and a map of visual features. The $(16+3N)\times 1$ state vector is

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} {}^{I}\bar{q}_{G}^{T} & \mathbf{b}_{g}^{T} & {}^{G}\mathbf{v}_{I}^{T} & \mathbf{b}_{a}^{T} & {}^{G}\mathbf{p}_{I}^{T} & | & {}^{G}\mathbf{p}_{f_1}^{T} \cdots {}^{G}\mathbf{p}_{f_N}^{T} \end{bmatrix}^{T} \\ &= \begin{bmatrix} \mathbf{x}_{s}^{T} & | & \mathbf{x}_{m}^{T} \end{bmatrix}^{T}, \end{aligned} \tag{3.1}$$

where $\mathbf{x}_s(t)$ is the 16×1 sensor platform state, and $\mathbf{x}_m(t)$ is the $3N\times 1$ state of the map. The first component of the sensor platform state, ${}^{I}\bar{q}_{G}(t)$, is the unit quaternion representing the orientation of the *global frame* $\{G\}$ in the IMU frame, $\{I\}$, at time t (see Fig. 3.1). The frame $\{I\}$ is attached to the IMU, while $\{G\}$ is a local-vertical reference frame whose origin coincides with the initial IMU position. The sensor platform state also includes the position and velocity of $\{I\}$ in $\{G\}$, denoted by the 3×1 vectors ${}^{G}\mathbf{p}_{I}(t)$ and ${}^{G}\mathbf{v}_{I}(t)$, respectively. The remaining components are the biases, $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$, affecting the gyroscope and accelerometer measurements, which are modeled as random-walk processes driven by the zero-mean, white Gaussian noise $\mathbf{n}_{wg}(t)$ and $\mathbf{n}_{wa}(t)$, respectively. The map, $\mathbf{x}_m$, comprises N visual features ${}^{G}\mathbf{p}_{f_i}$, $i=1,\ldots,N$. Note that for the case of VIO, the features are not stored in the state vector, but can be processed and marginalized on-the-fly [49] (see Section 3.3.2). With the state of the system now defined, we turn our attention to the continuous-time kinematic model which governs the time evolution of the system state.

Continuous-time model

The system model describing the time evolution of the state is (see [75, 76]):

$$\dot{{}^{I}_{G}\bar{q}}(t) = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}(t)){}^{I}\bar{q}_{G}(t) \tag{3.2}$$

$${}^{G}\dot{\mathbf{p}}_{I}(t) = {}^{G}\mathbf{v}_{I}(t) \tag{3.3}$$

$${}^{G}\dot{\mathbf{v}}_{I}(t) = {}^{G}\mathbf{a}_{I}(t) \tag{3.4}$$

$$\dot{\mathbf{b}}_{g}(t) = \mathbf{n}_{wg}(t) \tag{3.5}$$

$$\dot{\mathbf{b}}_{a}(t) = \mathbf{n}_{wa}(t) \tag{3.6}$$

$${}^{G}\dot{\mathbf{p}}_{f_i}(t) = \mathbf{0}_{3\times 1}\ ,\ \ i=1,\ldots,N. \tag{3.7}$$

In these expressions, $\boldsymbol{\omega}(t) = [\omega_1(t) \;\; \omega_2(t) \;\; \omega_3(t)]^T$ is the rotational velocity of the IMU, expressed in $\{I\}$, ${}^G\mathbf{a}_I(t)$ is the body acceleration expressed in $\{G\}$, and

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor \boldsymbol{\omega} \times \rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \quad \lfloor \boldsymbol{\omega} \times \rfloor \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

The gyroscope and accelerometer measurements, $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$, are modeled as

$$\begin{aligned} \boldsymbol{\omega}_m(t) &= \boldsymbol{\omega}(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \\ \mathbf{a}_m(t) &= \mathbf{C}({}^I\bar{q}_G(t))\,({}^G\mathbf{a}_I(t) - {}^G\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t), \end{aligned}$$

where $\mathbf{n}_g$ and $\mathbf{n}_a$ are zero-mean, white Gaussian noise processes, and ${}^G\mathbf{g}$ is the gravitational acceleration. The matrix $\mathbf{C}(\bar{q})$ is the rotation matrix corresponding to $\bar{q}$. The observed features belong to a static scene, hence, their time derivatives are zero [see (3.7)]. Linearizing at the current estimates and applying the expectation operator on both sides of (3.2)-(3.7), we obtain the state estimate propagation model

$$\dot{\hat{\bar{q}}}(t) = \frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}(t))\,\hat{\bar{q}}(t) \tag{3.8}$$

$${}^G\dot{\hat{\mathbf{p}}}_I(t) = {}^G\hat{\mathbf{v}}_I(t) \tag{3.9}$$

$${}^G\dot{\hat{\mathbf{v}}}_I(t) = \mathbf{C}^T({}^I_G\hat{\bar{q}}(t))\,\hat{\mathbf{a}}_I(t) + {}^G\mathbf{g} \tag{3.10}$$

$$\dot{\hat{\mathbf{b}}}_g(t) = \mathbf{0}_{3\times 1} \tag{3.11}$$

$$\dot{\hat{\mathbf{b}}}_a(t) = \mathbf{0}_{3\times 1} \tag{3.12}$$

$${}^G\dot{\hat{\mathbf{p}}}_{f_i}(t) = \mathbf{0}_{3\times 1}\;, \quad i = 1, \ldots, N, \tag{3.13}$$

where $\hat{\mathbf{a}}_I(t) = \mathbf{a}_m(t) - \hat{\mathbf{b}}_a(t)$, and $\hat{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}_m(t) - \hat{\mathbf{b}}_g(t)$. The $(15+3N)\times 1$ error-state vector is defined as

$$\begin{aligned} \widetilde{\mathbf{x}} &= \begin{bmatrix} {}^I\boldsymbol{\delta\theta}_G^T & \widetilde{\mathbf{b}}_g^T & {}^G\widetilde{\mathbf{v}}_I^T & \widetilde{\mathbf{b}}_a^T & {}^G\widetilde{\mathbf{p}}_I^T & | & {}^G\widetilde{\mathbf{p}}_{f_1}^T \cdots {}^G\widetilde{\mathbf{p}}_{f_N}^T \end{bmatrix}^T \\ &= \begin{bmatrix} \widetilde{\mathbf{x}}_s^T & | & \widetilde{\mathbf{x}}_m^T \end{bmatrix}^T, \end{aligned}$$

where $\widetilde{\mathbf{x}}_s(t)$ is the 15×1 error state corresponding to the sensing platform, and $\widetilde{\mathbf{x}}_m(t)$ is the $3N\times 1$ error state of the map. For the IMU position, velocity, biases, and the map, an additive error model is employed (i.e., $\widetilde{\mathbf{y}} = \mathbf{y} - \hat{\mathbf{y}}$ is the error in the estimate $\hat{\mathbf{y}}$ of a

quantity $\mathbf{y}$). However, for the quaternion we employ a multiplicative error model [76]. Specifically, the error between the quaternion $\bar{q}$ and its estimate $\hat{\bar{q}}$ is the 3×1 angle-error vector, $\boldsymbol{\delta\theta}$, implicitly defined by the error quaternion

$$\delta\bar{q} = \bar{q} \otimes \hat{\bar{q}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}^T & 1 \end{bmatrix}^T,$$

where $\delta\bar{q}$ describes the small rotation that causes the true and estimated attitude to coincide. This allows us to represent the attitude uncertainty by the 3×3 covariance matrix $\mathbb{E}[\boldsymbol{\delta\theta}\boldsymbol{\delta\theta}^T]$, which is a minimal representation.

The linearized continuous-time error-state equation is

$$\begin{aligned} \dot{\widetilde{\mathbf{x}}} &= \begin{bmatrix} \mathbf{F}_s & \mathbf{0}_{15\times 3N} \\ \mathbf{0}_{3N\times 15} & \mathbf{0}_{3N} \end{bmatrix} \widetilde{\mathbf{x}} + \begin{bmatrix} \mathbf{G}_s \\ \mathbf{0}_{3N\times 12} \end{bmatrix} \mathbf{n} \\ &= \mathbf{F}_c\, \widetilde{\mathbf{x}} + \mathbf{G}_c\, \mathbf{n} \end{aligned} \tag{3.14}$$

where $\mathbf{0}_{3N}$ denotes the $3N \times 3N$ matrix of zeros, $\mathbf{n} = \begin{bmatrix} \mathbf{n}_g^T & \mathbf{n}_{wg}^T & \mathbf{n}_a^T & \mathbf{n}_{wa}^T \end{bmatrix}^T$ is the system noise, $\mathbf{F}_s$ is the continuous-time error-state transition matrix corresponding to the sensor platform state, and $\mathbf{G}_s$ is the continuous-time input noise matrix, i.e.,

$$\mathbf{F}_s = \begin{bmatrix} -\lfloor \hat{\boldsymbol{\omega}} \times \rfloor & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\mathbf{C}^T({}^I_G\hat{\bar{q}})\lfloor \hat{\mathbf{a}}_I \times \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^I_G\hat{\bar{q}}) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

$$\mathbf{G}_s = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^I_G\hat{\bar{q}}) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

where $\mathbf{0}_3$ is the 3×3 matrix of zeros. The system noise is modelled as a zero-mean white Gaussian process with autocorrelation $\mathbb{E}[\mathbf{n}(t)\mathbf{n}^T(\tau)] = \mathbf{Q}_c\delta(t-\tau)$ which depends on the IMU noise characteristics and is computed off-line [76].

Discrete-time implementation

The IMU signals $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ are sampled at a constant rate $1/\delta t$, where $\delta t \triangleq t_{k+1} - t_k$. Every time a new IMU measurement is received, the state estimate is propagated using numerical integration of (3.8)–(3.13). In order to derive the covariance propagation equation, we compute the discrete-time state transition matrix, $\boldsymbol{\Phi}_{k+1,k}$, from time-step t_k to t_{k+1}, as the solution to the following matrix differential equation:

$$\dot{\boldsymbol{\Phi}}_{k+1,k} = \mathbf{F}_c \boldsymbol{\Phi}_{k+1,k} \tag{3.15}$$
$$\text{initial condition} \quad \boldsymbol{\Phi}_{k,k} = \mathbf{I}_{15+3N}$$

which can be calculated analytically as we show in [77] or numerically. We also compute the discrete-time system noise covariance matrix, $\mathbf{Q}_{d,k}$,

$$\mathbf{Q}_{d,k} = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1},\tau)\mathbf{G}_c\mathbf{Q}_c\mathbf{G}_c^T\boldsymbol{\Phi}^T(t_{k+1},\tau)\mathrm{d}\tau.$$

The propagated covariance is then computed as

$$\mathbf{P}_{k+1|k} = \boldsymbol{\Phi}_{k+1,k}\mathbf{P}_{k|k}\boldsymbol{\Phi}_{k+1,k}^T + \mathbf{Q}_{d,k}. \tag{3.16}$$

3.3.2 Measurement Update Model

As the camera-IMU platform moves, the camera observes visual features which are tracked over multiple image frames. These measurements are exploited to estimate the motion of the sensing platform and (optionally) the map of the environment.

To simplify the discussion, we consider the observation of a single point $\mathbf{p}_{f_i}$. The camera measures $\mathbf{z}_i$, which is the perspective projection of the 3D point ${}^I\mathbf{p}_{f_i}$ expressed in the current IMU frame $\{I\}$, onto the image plane, i.e.,

$$\mathbf{z}_i = \frac{1}{p_z}\begin{bmatrix} p_x \\ p_y \end{bmatrix} + \eta_i, \tag{3.17}$$

$$\text{where} \quad \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = {}^I\mathbf{p}_{f_i} = \mathbf{C}\left({}^I\overline{q}_G\right)\left({}^G\mathbf{p}_{f_i} - {}^G\mathbf{p}_I\right), \tag{3.18}$$

where the measurement noise, η_i, is modeled as zero mean, white Gaussian with covariance $\mathbf{R}_i$. We note that, without loss of generality, we consider the image measurement

in normalized pixel coordinates, and define the camera frame to be coincident with the IMU. In practice, we perform both intrinsic and extrinsic camera-IMU calibration off-line [54, 78].

The linearized error model is

$$\tilde{\mathbf{z}}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i \simeq \mathbf{H}_i \, \tilde{\mathbf{x}} + \eta_i, \tag{3.19}$$

where $\hat{\mathbf{z}} = \mathbf{h}\left(\hat{\mathbf{x}}\right)$ is the expected measurement computed by evaluating (3.17)-(3.18) at the current state estimate, and the measurement Jacobian, $\mathbf{H}_i$, is

$$\mathbf{H}_i = \mathbf{H}_c \begin{bmatrix} \mathbf{H}_\theta & \mathbf{0}_{3\times 9} & \mathbf{H}_p & | & \mathbf{0}_3 \cdots \mathbf{H}_{f_i} \cdots \mathbf{0}_3 \end{bmatrix} \tag{3.20}$$

where the partial derivatives are

$$\begin{aligned}
\mathbf{H}_c &= \frac{\partial \mathbf{h}}{\partial {}^I\mathbf{p}_{f_i}} = \frac{1}{p_z^2} \begin{bmatrix} p_z & 0 & -p_x \\ 0 & p_z & -p_y \end{bmatrix} \\
\mathbf{H}_\theta &= \frac{\partial {}^I\mathbf{p}_{f_i}}{\partial \theta} = \lfloor \mathbf{C}\left({}^I\bar{q}_G\right)\left({}^G\mathbf{p}_{f_i} - {}^G\mathbf{p}_I\right) \times \rfloor \\
\mathbf{H}_p &= \frac{\partial {}^I\mathbf{p}_{f_i}}{\partial {}^G\mathbf{p}_I} = -\mathbf{C}\left({}^I\bar{q}_G\right) \\
\mathbf{H}_{f_i} &= \frac{\partial {}^I\mathbf{p}_{f_i}}{\partial {}^G\mathbf{p}_{f_i}} = \mathbf{C}\left({}^I\bar{q}_G\right)
\end{aligned}$$

i.e., $\mathbf{H}_c$, is the Jacobian of the perspective projection with respect to ${}^I\mathbf{p}_{f_i}$, while $\mathbf{H}_\theta$, $\mathbf{H}_p$, and $\mathbf{H}_{f_i}$, are the Jacobians of ${}^I\mathbf{p}_{f_i}$ with respect to ${}^I\bar{q}_G$, ${}^G\mathbf{p}_I$, and ${}^G\mathbf{p}_{f_i}$, respectively.

This measurement model is used, independently of whether the map of the environment $\mathbf{x}_m$ is part of the state vector (V-SLAM) or not (VIO). Specifically, for the case of V-SLAM, when features that are already mapped are observed, the measurement model (3.17)-(3.20) can be directly applied to update the filter. In particular, we compute the measurement residual,

$$\mathbf{r}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i$$

the covariance of the residual,

$$\mathbf{S}_i = \mathbf{H}_i \mathbf{P}_{k+1|k} {\mathbf{H}_i}^T + \mathbf{R}_i$$

and the Kalman gain,

$$\mathbf{K}_i = \mathbf{P}_{k+1|k}\mathbf{H}_i{}^T\mathbf{S}_i^{-1}.$$

Employing these quantities, we compute the EKF state and covariance update as

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_i\,\mathbf{r}_i \\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_i\,\mathbf{S}_i\,\mathbf{K}_i^T.\end{aligned}$$

When features are first observed in V-SLAM, we initialize them into the feature map. To accomplish this, we compute an initial estimate, along with covariance and cross-correlations, by solving a bundle-adjustment over a short time window [77]. Finally, for the case of VIO, the map is not estimated explicitly; instead we use the Multi-State Constraint Kalman Filter (MSC-KF) approach [49] to impose a filter update constraining all the views from which a feature was seen. To accomplish this, we employ stochastic cloning [79] over a window of M camera poses.

3.4 Nonlinear System Observability Analysis

In this section, we provide a brief overview of the method in [50] for studying the observability of nonlinear systems and then introduce a new methodology for determining its unobservable directions.

3.4.1 Observability Analysis with Lie Derivatives

Consider a nonlinear, continuous-time system:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^{\ell} \mathbf{f}_i(\mathbf{x})u_i \\ \mathbf{z} = \mathbf{h}(\mathbf{x}) \end{cases} \tag{3.21}$$

where $\mathbf{u} = \begin{bmatrix} u_1 & \dots & u_\ell \end{bmatrix}^T$ is the control input, $\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_m \end{bmatrix}^T$ is the state vector, $\mathbf{z}$ is the output, and the vector functions $\mathbf{f}_i$, $i = 0, \dots, \ell$, comprise the process model.

Our objective is to study the observability properties of the system and to determine the directions in state-space that the measurements provide information. To this end,

we compute the Lie derivatives of the system. The zeroth-order Lie derivative of the measurement function $\mathbf{h}$ is defined as the function itself [50]:

$$\mathfrak{L}^0\mathbf{h} = \mathbf{h}(\mathbf{x}).$$

Each subsequent Lie derivative is formed recursively from the definition of $\mathfrak{L}^0\mathbf{h}$. Specifically, for any i-th-order Lie derivative, $\mathfrak{L}^i\mathbf{h}$, the $(i+1)$-th-order Lie derivative $\mathfrak{L}^{i+1}_{\mathbf{f}_j}\mathbf{h}$ with respect to a process function $\mathbf{f}_j$ is computed as:

$$\mathfrak{L}^{i+1}_{\mathbf{f}_j}\mathbf{h} = \nabla\mathfrak{L}^i\mathbf{h}\cdot\mathbf{f}_j,$$

where $\nabla\mathfrak{L}^i\mathbf{h}$ denotes the span of the i-th-order Lie derivative, i.e.,

$$\nabla\mathfrak{L}^i\mathbf{h} = \begin{bmatrix}\frac{\partial\mathfrak{L}^i\mathbf{h}}{\partial x_1} & \frac{\partial\mathfrak{L}^i\mathbf{h}}{\partial x_2} & \dots & \frac{\partial\mathfrak{L}^i\mathbf{h}}{\partial x_m}\end{bmatrix}.$$

In order to determine the directions along which information can be acquired, we examine the span of the Lie derivatives. We do this by forming the observability matrix, $\mathcal{O}$, whose block-rows comprise the spans of the Lie derivatives of the system, i.e.,

$$\mathcal{O} = \begin{bmatrix}\nabla\mathfrak{L}^0\mathbf{h}\\ \nabla\mathfrak{L}^1_{\mathbf{f}_i}\mathbf{h}\\ \nabla\mathfrak{L}^2_{\mathbf{f}_i\mathbf{f}_j}\mathbf{h}\\ \nabla\mathfrak{L}^3_{\mathbf{f}_i\mathbf{f}_j\mathbf{f}_k}\mathbf{h}\\ \vdots\end{bmatrix}$$

where $i,j,k = 1,\dots,\ell$. Based on [50], to prove that a system is observable, it suffices to show that a submatrix of $\mathcal{O}$ comprising a subset of its rows is of full column rank. In contrast, to prove that a system is unobservable and find its unobservable directions, we need to: (i) Show that the infinitely many block rows of $\mathcal{O}$ can be written as a linear combination of a subset of its block rows, which form a submatrix $\mathcal{O}'$; (ii) Find the nullspace of $\mathcal{O}'$ in order to determine the system's unobservable directions. Although accomplishing (ii) can be straightforward for certain systems, achieving (i) is extremely challenging especially for high-dimensional systems, such as the one describing VINS.

To address this issue, in the following section, we present a new methodology that relies on a change of variables for proving that a system is unobservable and finding its unobservable directions.

3.4.2 Observability Analysis with Basis Functions

In order to gain intuition for the following derivations, we provide a brief overview of the motivation for this methodology. As stated in the previous section, following the approach of [50] for analyzing the observability properties of a nonlinear system is quite challenging. The main issue is that we must analytically compute the nullspace of a matrix with an infinite number of rows (since there are infinitely many Lie derivatives). However, our analysis can be significantly simplified if we can find a process for decomposing the observability matrix into a product of two matrices: (i) a full-rank matrix with infinitely many rows, and (ii) a rank-deficient matrix with only a limited number of rows. In what follows, we show how to achieve such a factorization of the observability matrix, by computing a set of basis functions of the state, which comprise its observable modes.

We start by proving the following:

Theorem 1: Assume that there exists a nonlinear transformation $\boldsymbol{\beta}(\mathbf{x}) = \begin{bmatrix}\boldsymbol{\beta}_1(\mathbf{x})^T & \dots & \boldsymbol{\beta}_t(\mathbf{x})^T\end{bmatrix}^T$. These bases are functions of the variable $\mathbf{x}$ in (3.21), and the number of basis elements, t, is defined so as to fulfill:

(C1) $\boldsymbol{\beta}_1(\mathbf{x}) = \mathbf{h}(\mathbf{x})$;

(C2) $\frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}} \cdot \mathbf{f}_i, i = 0, \dots, \ell$ is a function of $\boldsymbol{\beta}$;

(C3) The system:

$$\begin{cases} \dot{\boldsymbol{\beta}} = \mathbf{g}_0(\boldsymbol{\beta}) + \sum_{i=1}^{\ell} \mathbf{g}_i(\boldsymbol{\beta}) u_i \\ \mathbf{z} = \mathbf{h} = \boldsymbol{\beta}_1 \end{cases} \tag{3.22}$$

where $\mathbf{g}_i(\boldsymbol{\beta}) = \frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}} \mathbf{f}_i(\mathbf{x}), i = 0, \dots, \ell$, is observable.

Then:

(i) The observability matrix of (3.21) can be factorized as:

$$\mathcal{O} = \Xi \cdot \mathbf{B},$$

where Ξ is the observability matrix of system (3.22) and $\mathbf{B} \triangleq \frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}}$

(ii) $null(\mathcal{O}) = null(\mathbf{B})$

Proof:

(i) Based on the chain rule, the span of any Lie derivative $\nabla \mathfrak{L}^i \mathbf{h}$ can be written as:

$$\nabla \mathfrak{L}^i \mathbf{h} = \frac{\partial \mathfrak{L}^i \mathbf{h}}{\partial \mathbf{x}} = \frac{\partial \mathfrak{L}^i \mathbf{h}}{\partial \boldsymbol{\beta}} \frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}}$$

Thus, the observability matrix $\mathcal{O}$ of (3.21) can be factorized as:

$$\mathcal{O} = \begin{bmatrix} \nabla \mathfrak{L}^0 \mathbf{h} \\ \nabla \mathfrak{L}^1_{\mathbf{f}_i} \mathbf{h} \\ \nabla \mathfrak{L}^2_{\mathbf{f}_i \mathbf{f}_j} \mathbf{h} \\ \nabla \mathfrak{L}^3_{\mathbf{f}_i \mathbf{f}_j \mathbf{f}_k} \mathbf{h} \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathfrak{L}^0 \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^1_{\mathbf{f}_i} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^2_{\mathbf{f}_i \mathbf{f}_j} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathfrak{L}^3_{\mathbf{f}_i \mathbf{f}_j \mathbf{f}_k} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \vdots \end{bmatrix} \frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}} = \Xi \cdot \mathbf{B} \tag{3.23}$$

Next we prove that Ξ is the observability matrix of the system (3.22) by induction.

To distinguish the Lie derivatives of system (3.21), let $\mathfrak{I}$ denote the Lie derivatives of system (3.22). Then, the span of its zeroth-order Lie derivative is:

$$\nabla \mathfrak{I}^0 \mathbf{h} = \frac{\partial \mathbf{h}}{\partial \boldsymbol{\beta}} = \frac{\partial \mathfrak{L}^0 \mathbf{h}}{\partial \boldsymbol{\beta}}$$

which corresponds to the first block row of Ξ.

Assume that the span of the i-th-order Lie derivative of (3.22) along any direction can be written as $\nabla \mathfrak{I}^i \mathbf{h} = \frac{\partial \mathfrak{L}^i \mathbf{h}}{\partial \boldsymbol{\beta}}$, which corresponds to a block row of Ξ. Then the span of the $(i+1)$-th-order Lie derivative $\nabla \mathfrak{I}^{i+1}_{\mathbf{g}_j} \mathbf{h}$ along the process function $\mathbf{g}_j$ can be computed as:

$$\begin{aligned} \nabla \mathfrak{I}^{i+1}_{\mathbf{g}_j} \mathbf{h} &= \frac{\partial \mathfrak{I}^{i+1}_{\mathbf{g}_j} \mathbf{h}}{\partial \boldsymbol{\beta}} = \frac{\partial (\nabla \mathfrak{I}^i \mathbf{h} \cdot \mathbf{g}_j)}{\partial \boldsymbol{\beta}} = \frac{\partial (\frac{\partial \mathfrak{L}^i \mathbf{h}}{\partial \boldsymbol{\beta}} \cdot \frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}} \mathbf{f}_j(\mathbf{x}))}{\partial \boldsymbol{\beta}} \\ &= \frac{\partial (\frac{\partial \mathfrak{L}^i \mathbf{h}}{\partial \mathbf{x}} \cdot \mathbf{f}_j(\mathbf{x}))}{\partial \boldsymbol{\beta}} = \frac{\partial \mathfrak{L}^{i+1}_{\mathbf{f}_j} \mathbf{h}}{\partial \boldsymbol{\beta}} \end{aligned}$$

which is also a block row of Ξ. Therefore, we conclude that Ξ is a matrix whose rows are the span of all the Lie derivatives of system (3.22), and thus it is the observability matrix of system (3.22). ■

(ii) From $\mathcal{O} = \Xi \mathbf{B}$, we have $null(\mathcal{O}) = null(\mathbf{B}) + null(\Xi) \cap range(\mathbf{B})$ (see (4.5.1) in [80]). Moreover, from condition *(C3)* system (3.22) is observable, and Ξ is of full column rank. Therefore $null(\mathcal{O}) = null(\mathbf{B})$. ■

Based on *Theorem 1*, the unobservable directions can be determined with significantly less effort. Specifically, to find a system's unobservable directions, we first need to define the basis functions that satisfy conditions *(C1)* and *(C2)*, and prove that matrix Ξ is of full column rank, which is condition *(C3)*. Once all the conditions are

satisfied, the unobservable directions of (3.21) correspond to the nullspace of matrix $\mathbf{B}$, which has finite dimensions and thus is fairly easy to find.

In the following sections, we will leverage *Theorem 1* to prove that the VINS model is unobservable and find its unobservable directions. To do this, in Section 3.5.1 we first review the VINS model. In Section 3.5.2, we find the set of basis functions that satisfy conditions *(C1)* and *(C2)* of *Theorem 1*, and construct the basis functions' system as in (3.22) for this particular problem. In Section 3.5.3, we prove that the observability matrix Ξ for the basis functions' system is of full column rank, which is condition *(C3)* of *Theorem 1*. Lastly, we determine the unobservable directions of the VINS model by finding the nullspace of matrix $\mathbf{B}$.

3.5 Observability Analysis of the VINS Model

In this section, we present the observability analysis for the VINS model using basis functions.

3.5.1 Revisiting the System Model

For the purpose of simplifying the observability analysis, we express the IMU orientation using the Cayley-Gibbs-Rodriguez (CGR) parameterization [81], which is a minimal representation. Specifically, the orientation of $\{G\}$ with respect to $\{I\}$ is the 3×1 vector of CGR parameters, ${}^{I}\mathbf{s}_{G}$. Hence, we rewrite (3.1) as

$$\mathbf{x} = \begin{bmatrix} {}^{I}\mathbf{s}_{G}^{T} & \mathbf{b}_{g}^{T} & {}^{G}\mathbf{v}_{I}^{T} & \mathbf{b}_{a}^{T} & {}^{G}\mathbf{p}_{I}^{T} & {}^{G}\mathbf{p}_{f}^{T} \end{bmatrix}^{T}.$$

The time evolution of ${}^{I}\mathbf{s}_{G}$ is

$${}^{I}\dot{\mathbf{s}}_{G}(t) = \mathbf{D}\left({}^{I}\boldsymbol{\omega}(t) - \mathbf{b}_{g}(t)\right) \tag{3.24}$$

$$\text{where } \mathbf{D} \triangleq \frac{\partial \mathbf{s}}{\partial \boldsymbol{\theta}} = \frac{1}{2}\left(\mathbf{I}_{3} + \lfloor \mathbf{s} \times \rfloor + \mathbf{s}\mathbf{s}^{T}\right). \tag{3.25}$$

3.5.2 Determining the System's Basis Functions

In this section, we define the basis functions for the VINS model that satisfy conditions *(C1), (C2)* of *Theorem 1*. We achieve this by applying *(C1)* to obtain $\boldsymbol{\beta}_{1}$ and recursively employing *(C2)* to define the additional elements $\boldsymbol{\beta}_{j}$, $j = 2, \ldots, 6$. We note that at each

step of this process there may be multiple options for selecting $\boldsymbol{\beta}_j$, and we mitigate this by favoring bases that have a meaningful physical interpretation. After determining the bases, we present the model of the corresponding system (3.44), and show that it is observable in the next section.

To preserve the clarity of presentation, we retain only a few of the subscripts and superscripts in the state elements and write the system state vector as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{s}^T & \mathbf{b}_g^T & \mathbf{v}^T & \mathbf{b}_a^T & \mathbf{p}^T & \mathbf{p}_f^T \end{bmatrix}^T.$$

The VINS model [see (3.2)-(3.7), (3.17)-(3.18), and (3.24)] is expressed in input-affine form as:

$$\begin{bmatrix} \dot{\mathbf{s}} \\ \dot{\mathbf{b}}_g \\ \dot{\mathbf{v}} \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{p}} \\ \dot{\mathbf{p}}_f \end{bmatrix} = \underbrace{\begin{bmatrix} -\mathbf{D}\,\mathbf{b}_g \\ \mathbf{0}_{3\times 1} \\ \mathbf{g} - \mathbf{C}^T\mathbf{b}_a \\ \mathbf{0}_{3\times 1} \\ \mathbf{v} \\ \mathbf{0}_{3\times 1} \end{bmatrix}}_{f_0} + \underbrace{\begin{bmatrix} \mathbf{D} \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \end{bmatrix}}_{\mathbf{f}_1} \boldsymbol{\omega} + \underbrace{\begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{C}^T \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \end{bmatrix}}_{\mathbf{f}_2} \mathbf{a} \tag{3.26}$$

$$\mathbf{z} = \frac{1}{p_z}\begin{bmatrix} p_x \\ p_y \end{bmatrix}, \quad \text{where } \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = {}^I\mathbf{p}_f = \mathbf{C}\left(\mathbf{p}_f - \mathbf{p}\right) \tag{3.27}$$

and $\mathbf{C} \triangleq \mathbf{C}(\mathbf{s})$. Note that f_0 is an 18×1 vector, while $\mathbf{f}_1$ and $\mathbf{f}_2$ are both 18×3 matrices which is a compact way for representing three process functions:

$$\mathbf{f}_1\boldsymbol{\omega} = f_{11} \cdot \omega_1 + f_{12} \cdot \omega_2 + f_{13} \cdot \omega_3$$
$$\mathbf{f}_2\mathbf{a} = f_{21} \cdot a_1 + f_{22} \cdot a_2 + f_{23} \cdot a_3.$$

Using this model, we define the bases for this system by applying the conditions of *Theorem 1*. Specifically, we (i) select $\boldsymbol{\beta}_1$ as the measurement function $\mathbf{z}$, and (ii) recursively determine the remaining bases so that $\frac{\partial \boldsymbol{\beta}_j}{\partial \mathbf{x}} \cdot \mathbf{f}_i$ can be expressed in terms of $\boldsymbol{\beta}$ for all the process functions. Note also that the definition of the bases is not unique, any basis functions that satisfy the conditions of *Theorem 1* span the same space.

The first basis is defined as the measurement function:

$$\boldsymbol{\beta}_1 \triangleq \mathbf{h}(\mathbf{x}) = \frac{1}{p_z}\begin{bmatrix} p_x \\ p_y \end{bmatrix}.$$

In order to compute the remaining basis elements, we must ensure that the properties of *Theorem 1* are satisfied. We do so by applying *(C2)* to $\boldsymbol{\beta}_1$.

Satisfying Condition *(C2)* of *Theorem 1* for $\boldsymbol{\beta}_1$

We start by computing the span of $\boldsymbol{\beta}_1$ with respect to $\mathbf{x}$, i.e.,

$$\begin{aligned}\frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{x}} &= \begin{bmatrix} \frac{\partial \boldsymbol{\beta}_1}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} & \frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{b}_g} & \frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{v}} & \frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{b}_a} & \frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{p}} & \frac{\partial \boldsymbol{\beta}_1}{\partial {}^I\mathbf{p}_f} \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} \frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2} \\ 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2} \end{bmatrix}}_{\frac{\partial \mathbf{h}}{\partial {}^I\mathbf{p}_f}} \underbrace{\begin{bmatrix} \lfloor {}^I\mathbf{p}_f \times \rfloor \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C} & \mathbf{C} \end{bmatrix}}_{\frac{\partial {}^I\mathbf{p}_f}{\partial \mathbf{x}}} \end{aligned} \tag{3.28}$$

where $\frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} = \mathbf{D}^{-1}$ [see (3.25)]. Once the span of the first basis function $\boldsymbol{\beta}_1$ is obtained, we project it onto *all* the process functions, f_0, $\mathbf{f}_1$, and $\mathbf{f}_2$ [see (3.26)], in order to determine the other basis functions that satisfy condition *(C2)* of *Theorem 1*. During this procedure, our aim is to ensure that every term in the resulting product is a function of the existing basis elements. Whenever a term cannot be expressed by the previously defined basis functions, we incorporate it as a new basis function.

Specifically, beginning with the projection of $\frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{x}}$ along f_0 we obtain

$$\begin{aligned}\frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{x}} \cdot f_0 &= \begin{bmatrix} \frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2} \\ 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2} \end{bmatrix} \left(-\lfloor {}^I\mathbf{p}_f \times \rfloor \mathbf{b}_g - \mathbf{C}\,\mathbf{v}\right) \\ &= \begin{bmatrix} \mathbf{I}_2 & -\boldsymbol{\beta}_1 \end{bmatrix} \left(-\lfloor \begin{bmatrix} \boldsymbol{\beta}_1 \\ 1 \end{bmatrix} \times \rfloor \mathbf{b}_g - \frac{1}{p_z}\mathbf{C}\,\mathbf{v} \right). \end{aligned} \tag{3.29}$$

This is a function of $\boldsymbol{\beta}_1$ and of other elements of the state $\mathbf{x}$, namely $\mathbf{b}_g$ and $\mathbf{v}$, as well as functions of $\mathbf{x}$, which are $\frac{1}{p_z}$ and $\mathbf{C}$. Hence, in order to satisfy *(C2)*, we must define

new basis elements, which we select as physically interpretable quantities:

$$\begin{aligned} \boldsymbol{\beta}_2 &\triangleq \frac{1}{p_z} \\ \boldsymbol{\beta}_3 &\triangleq \mathbf{C}\,\mathbf{v} \\ \boldsymbol{\beta}_4 &\triangleq \mathbf{b}_g, \end{aligned} \tag{3.30}$$

where $\boldsymbol{\beta}_2$ is the inverse depth to the point, $\boldsymbol{\beta}_3$ is the velocity expressed in the local frame, and $\boldsymbol{\beta}_4$ is the gyroscope bias. Rewriting (3.29) using these definitions we have:

$$\frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{x}} \cdot f_0 \triangleq \begin{bmatrix} \mathbf{I}_2 & -\boldsymbol{\beta}_1 \end{bmatrix} \left(-\lfloor \begin{bmatrix} \boldsymbol{\beta}_1 \\ 1 \end{bmatrix} \times \rfloor \boldsymbol{\beta}_4 - \boldsymbol{\beta}_2\, \boldsymbol{\beta}_3 \right).$$

Note that later on we will need to ensure that the properties of *Theorem 1* are also satisfied for these new elements, $\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_3$, and $\boldsymbol{\beta}_4$, but first we examine the projections of the span of $\boldsymbol{\beta}_1$ along $\mathbf{f}_1$ and $\mathbf{f}_2$.

The projections of $\frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{x}}$ along the three directions of $\mathbf{f}_1$ (i.e., $\mathbf{f}_1\mathbf{e}_i$, $i = 1, 2, 3$, where $\begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} = \mathbf{I}_3$) are

$$\begin{aligned} \frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{x}} \cdot \mathbf{f}_1\mathbf{e}_i &= \begin{bmatrix} \frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2} \\ 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2} \end{bmatrix} \lfloor {}^I\mathbf{p}_f \times \rfloor \mathbf{e}_i \\ &= \begin{bmatrix} \mathbf{I}_2 & -\boldsymbol{\beta}_1 \end{bmatrix} \lfloor \begin{bmatrix} \boldsymbol{\beta}_1 \\ 1 \end{bmatrix} \times \rfloor \mathbf{e}_i, \quad i = 1, 2, 3. \end{aligned} \tag{3.31}$$

Note that in this case no new basis functions need to be defined since (3.31) already satisfies condition *(C2)* of *Theorem 1*. Lastly, the projections of $\frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{x}}$ along the $\mathbf{f}_2$ directions are

$$\frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{x}} \cdot \mathbf{f}_2\mathbf{e}_i = \mathbf{0}_{2\times 1}, \quad i = 1, 2, 3.$$

Hence, by adding the new basis elements $\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_3$, and $\boldsymbol{\beta}_4$, we ensure that the properties of *Theorem 1* are fulfilled for $\boldsymbol{\beta}_1$. To make the newly defined basis functions, $\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_3$, and $\boldsymbol{\beta}_4$, satisfy condition *(C2)*, we proceed by projecting their spans on the process functions.

Satisfying Condition *(C2)* of *Theorem 1* for $\boldsymbol{\beta}_2$

The derivative of $\boldsymbol{\beta}_2$ [see (3.30)] with respect to the state is:

$$\frac{\partial \boldsymbol{\beta}_2}{\partial \mathbf{x}} = -\frac{1}{p_z^2}\mathbf{e}_3^T \begin{bmatrix} \lfloor {}^I\mathbf{p}_f \times \rfloor \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C} & \mathbf{C} \end{bmatrix}. \tag{3.32}$$

Projecting (3.32) along f_0 we obtain

$$\begin{aligned} \frac{\partial \boldsymbol{\beta}_2}{\partial \mathbf{x}} \cdot f_0 &= -\frac{1}{p_z^2}\mathbf{e}_3^T \left(-\lfloor {}^G\mathbf{p}_f \times \rfloor \mathbf{b}_g - \mathbf{C}\,\mathbf{v}\right) \\ &= -\boldsymbol{\beta}_2 \mathbf{e}_3^T \left(-\lfloor \begin{bmatrix} \boldsymbol{\beta}_1 \\ 1 \end{bmatrix} \times \rfloor \boldsymbol{\beta}_4 - \boldsymbol{\beta}_2\, \boldsymbol{\beta}_3 \right), \end{aligned}$$

which is a function of only the currently enumerated basis elements.

We also project $\frac{\partial \boldsymbol{\beta}_2}{\partial \mathbf{x}}$ along the remaining input directions, i.e., $\mathbf{f}_j\,\mathbf{e}_i$, $j = 1, 2$, $i = 1, 2, 3$.

$$\begin{aligned} \frac{\partial \boldsymbol{\beta}_2}{\partial \mathbf{x}} \cdot \mathbf{f}_1 \mathbf{e}_i &= -\frac{1}{p_z^2}\mathbf{e}_3^T \lfloor {}^G\mathbf{p}_f \times \rfloor \mathbf{e}_i \\ &= -\boldsymbol{\beta}_2 \mathbf{e}_3^T \lfloor \begin{bmatrix} \boldsymbol{\beta}_1 \\ 1 \end{bmatrix} \times \rfloor \mathbf{e}_i, \quad i = 1, 2, 3 \\ \frac{\partial \boldsymbol{\beta}_2}{\partial \mathbf{x}} \cdot \mathbf{f}_2 \mathbf{e}_i &= 0, \quad i = 1, 2, 3, \end{aligned} \tag{3.33}$$

which does not admit any new basis elements. Thus, we see that $\boldsymbol{\beta}_2$ fulfills the properties of *Theorem 1* without requiring us to define any new basis elements.

Satisfying Condition *(C2)* of *Theorem 1* for $\boldsymbol{\beta}_3$

Following the same procedure again, we compute the span of $\boldsymbol{\beta}_3$ with respect to $\mathbf{x}$:

$$\frac{\partial \boldsymbol{\beta}_3}{\partial \mathbf{x}} = \begin{bmatrix} \lfloor \mathbf{C}\,\mathbf{v} \times \rfloor \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} & \mathbf{0}_3 & \mathbf{C} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{3.34}$$

and then the projection of (3.34) along the input direction f_0

$$\begin{aligned} \frac{\partial \boldsymbol{\beta}_3}{\partial \mathbf{x}} \cdot f_0 &= -\lfloor \mathbf{C}\,\mathbf{v} \times \rfloor \mathbf{b}_g + \mathbf{C}\,\mathbf{g} - \mathbf{b}_a \\ &\triangleq -\lfloor \boldsymbol{\beta}_3 \times \rfloor \boldsymbol{\beta}_4 + \boldsymbol{\beta}_5 - \boldsymbol{\beta}_6, \end{aligned}$$

where we assign two new basis elements, i.e.,

$$\boldsymbol{\beta}_5 \triangleq \mathbf{C}\,\mathbf{g}$$
$$\boldsymbol{\beta}_6 \triangleq \mathbf{b}_a.$$

Note again that we selected physically interpretable functions: (i) $\boldsymbol{\beta}_5$ is the gravity vector expressed in the local frame, and (ii) $\boldsymbol{\beta}_6$ is the accelerometer bias. The projections of (3.34) along $\mathbf{f}_j\,\mathbf{e}_i$, $j = 1, 2$, $i = 1, 2, 3$, are

$$\frac{\partial \boldsymbol{\beta}_3}{\partial \mathbf{x}} \cdot \mathbf{f}_1\mathbf{e}_i = \lfloor \mathbf{C}\,\mathbf{v} \times \rfloor \mathbf{e}_i = \lfloor \boldsymbol{\beta}_3 \times \rfloor \mathbf{e}_i, \quad i = 1, 2, 3$$
$$\frac{\partial \boldsymbol{\beta}_3}{\partial \mathbf{x}} \cdot \mathbf{f}_2\mathbf{e}_i = \mathbf{I}_3\mathbf{e}_i = \mathbf{e}_i, \quad i = 1, 2, 3$$

which do not produce additional bases.

Satisfying Condition *(C2)* of *Theorem 1* for $\boldsymbol{\beta}_4$

We proceed by examining the span of $\boldsymbol{\beta}_4$ with respect to $\mathbf{x}$, i.e.,

$$\frac{\partial \boldsymbol{\beta}_4}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{3.35}$$

with corresponding projections

$$\frac{\partial \boldsymbol{\beta}_4}{\partial \mathbf{x}} \cdot f_0 = \mathbf{0}_{3\times 1}$$
$$\frac{\partial \boldsymbol{\beta}_4}{\partial \mathbf{x}} \cdot \mathbf{f}_j\mathbf{e}_i = \mathbf{0}_{3\times 1}, \quad j = 1, 2,\ i = 1, 2, 3.$$

We note here that no additional basis elements are produced.

Satisfying Condition *(C2)* of *Theorem 1* for $\boldsymbol{\beta}_5$

The derivative of $\boldsymbol{\beta}_5$ with respect to $\mathbf{x}$ is:

$$\frac{\partial \boldsymbol{\beta}_5}{\partial \mathbf{x}} = \begin{bmatrix} \lfloor \mathbf{C}\,\mathbf{g} \times \rfloor \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}. \tag{3.36}$$

Projecting (3.36) along the input directions, we obtain

$$\frac{\partial \boldsymbol{\beta}_5}{\partial \mathbf{x}} \cdot f_0 = -\lfloor \mathbf{C}\,\mathbf{g} \times \rfloor \mathbf{b}_g = -\lfloor \boldsymbol{\beta}_5 \times \rfloor \boldsymbol{\beta}_4$$
$$\frac{\partial \boldsymbol{\beta}_5}{\partial \mathbf{x}} \cdot \mathbf{f}_1\mathbf{e}_i = \lfloor \mathbf{C}\,\mathbf{g} \times \rfloor \mathbf{e}_i = \lfloor \boldsymbol{\beta}_5 \times \rfloor \mathbf{e}_i, \quad i = 1, 2, 3$$
$$\frac{\partial \boldsymbol{\beta}_5}{\partial \mathbf{x}} \cdot \mathbf{f}_2\mathbf{e}_i = \mathbf{0}_{3\times 1}, \quad i = 1, 2, 3.$$

All of these are either a function of the existing basis elements, or are equal to zero, and thus we do not need to define any additional bases.

Satisfying Condition *(C2)* of *Theorem 1* for $\boldsymbol{\beta}_6$

Lastly, we examine the span of the remaining basis element $\boldsymbol{\beta}_6$, i.e.,

$$\frac{\partial \boldsymbol{\beta}_6}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}. \tag{3.37}$$

The projections of (3.37) along the input directions are

$$\begin{aligned} \frac{\partial \boldsymbol{\beta}_6}{\partial \mathbf{x}} \cdot f_0 &= \mathbf{0}_{3\times 1} \\ \frac{\partial \boldsymbol{\beta}_6}{\partial \mathbf{x}} \cdot \mathbf{f}_j \mathbf{e}_i &= \mathbf{0}_{3\times 1}, \quad j = 1, 2,\ i = 1, 2, 3, \end{aligned}$$

which do not produce any additional basis elements.

At this point, we have proved that the conditions *(C1)* and *(C2)* of *Theorem 1* are satisfied for all of the basis elements; hence, we have defined a complete basis set for the VINS model:

$$\boldsymbol{\beta}_1 = \mathbf{h}(\mathbf{x}) \tag{3.38}$$

$$\boldsymbol{\beta}_2 = \frac{1}{p_z} \tag{3.39}$$

$$\boldsymbol{\beta}_3 = \mathbf{C}\,\mathbf{v} \tag{3.40}$$

$$\boldsymbol{\beta}_4 = \mathbf{b}_g \tag{3.41}$$

$$\boldsymbol{\beta}_5 = \mathbf{C}\,\mathbf{g} \tag{3.42}$$

$$\boldsymbol{\beta}_6 = \mathbf{b}_a. \tag{3.43}$$

These correspond to the landmark projection on the image plane (3.38), the inverse depth to the landmark (3.39), the velocity expressed in the local frame (3.40), the gyro bias (3.41), the gravity vector expressed in the local frame (3.42), and the accelerometer bias (3.43). Based on *Theorem 1*, the resulting system in the basis functions [see (3.22)] is:

$$\begin{bmatrix}\dot{\boldsymbol{\beta}}_1\\ \dot{\boldsymbol{\beta}}_2\\ \dot{\boldsymbol{\beta}}_3\\ \dot{\boldsymbol{\beta}}_4\\ \dot{\boldsymbol{\beta}}_5\\ \dot{\boldsymbol{\beta}}_6\end{bmatrix} = \underbrace{\begin{bmatrix}\bar{\bar{\boldsymbol{\beta}}}_1\left(-\lfloor\bar{\boldsymbol{\beta}}_1\times\rfloor\boldsymbol{\beta}_4-\boldsymbol{\beta}_2\,\boldsymbol{\beta}_3\right)\\ \boldsymbol{\beta}_2\mathbf{e}_3^T\left(\lfloor\bar{\boldsymbol{\beta}}_1\times\rfloor\boldsymbol{\beta}_4+\boldsymbol{\beta}_2\,\boldsymbol{\beta}_3\right)\\ -\lfloor\boldsymbol{\beta}_3\times\rfloor\boldsymbol{\beta}_4+\boldsymbol{\beta}_5-\boldsymbol{\beta}_6\\ \mathbf{0}_{3\times1}\\ -\lfloor\boldsymbol{\beta}_5\times\rfloor\boldsymbol{\beta}_4\\ \mathbf{0}_{3\times1}\end{bmatrix}}_{g_0} + \underbrace{\begin{bmatrix}\bar{\bar{\boldsymbol{\beta}}}_1\lfloor\bar{\boldsymbol{\beta}}_1\times\rfloor\\ -\boldsymbol{\beta}_2\mathbf{e}_3^T\lfloor\bar{\boldsymbol{\beta}}_1\times\rfloor\\ \lfloor\boldsymbol{\beta}_3\times\rfloor\\ \mathbf{0}_3\\ \lfloor\boldsymbol{\beta}_5\times\rfloor\\ \mathbf{0}_3\end{bmatrix}}_{\mathbf{g}_1}\boldsymbol{\omega} + \underbrace{\begin{bmatrix}\mathbf{0}_{2\times3}\\ \mathbf{0}_{1\times3}\\ \mathbf{I}_3\\ \mathbf{0}_3\\ \mathbf{0}_3\\ \mathbf{0}_3\end{bmatrix}}_{\mathbf{g}_2}\mathbf{a}$$

$$\mathbf{y} = \boldsymbol{\beta}_1, \tag{3.44}$$

where $\bar{\boldsymbol{\beta}}_1 = \begin{bmatrix}\boldsymbol{\beta}_1^T & 1\end{bmatrix}^T$ denotes $\boldsymbol{\beta}_1$ expressed as a 3×1 homogeneous vector, and $\bar{\bar{\boldsymbol{\beta}}}_1 = \begin{bmatrix}\mathbf{I}_2 & -\boldsymbol{\beta}_1\end{bmatrix}$. In the next section, we will show that system (3.44) is observable by proving its observability matrix Ξ is of full column rank. Therefore, the basis functions $\boldsymbol{\beta}_1$ to $\boldsymbol{\beta}_6$ correspond to the observable modes of system (3.26)-(3.27), and the system model (3.44) governs the time evolution of the observable state.

3.5.3 Determining the System's Observability Matrix and its Unobservable Directions

Based on *Theorem 1*, the observability matrix $\mathcal{O}$ of the VINS model [see (3.26)] is the product of the observability matrix Ξ of system (3.44) with the matrix $\mathbf{B}$ comprising the derivatives of the basis functions. In what follows, we first prove that matrix Ξ is of full column rank. Then, we find the nullspace of matrix $\mathbf{B}$, which according to *Theorem 1* corresponds to the unobservable directions of the VINS model.

Lemma 2: System (3.44) is observable.

Proof: See Appendix B.

Since system (3.44) is observable, based on *Theorem 1*, we can find the unobservable directions of system (3.26) from the nullspace of matrix $\mathbf{B}$.

Theorem 3: The VINS model (3.26) is unobservable, and its unobservable subspace is spanned by four directions [see (3.46)] corresponding to the IMU-camera global position and its rotation around the gravity vector in the global frame.

Proof: System (3.44) satisfies the conditions of *Theorem 1*. Therefore, $null(\mathcal{O}) = null(\mathbf{B})$, which spans the unobservable subspace of the original system (3.26). Stacking the derivatives of the basis functions with respect to the variable $\mathbf{x}$, the matrix $\mathbf{B}$ can

be written as [see (3.28), (3.32), (3.34), (3.35), (3.36), and (3.37)]:

$$\mathbf{B}=\underbrace{\begin{bmatrix}\boldsymbol{\zeta} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3\end{bmatrix}}_{\mathbf{B}_1}\underbrace{\begin{bmatrix}\lfloor {}^I\mathbf{p}_f\times\rfloor\frac{\partial\boldsymbol{\theta}}{\partial\mathbf{s}} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C} & \mathbf{C}\\ \lfloor\mathbf{C}\,\mathbf{v}\times\rfloor\frac{\partial\boldsymbol{\theta}}{\partial\mathbf{s}} & \mathbf{0}_3 & \mathbf{C} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \lfloor\mathbf{C}\,\mathbf{g}\times\rfloor\frac{\partial\boldsymbol{\theta}}{\partial\mathbf{s}} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}}_{\mathbf{B}_2} \tag{3.45}$$

where we have factorized $\mathbf{B}=\mathbf{B}_1\mathbf{B}_2$ to further simplify the proof, and for conciseness, we have denoted the first subblock of $\mathbf{B}_1$ as

$$\boldsymbol{\zeta}=\begin{bmatrix}\frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2}\\ 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2}\\ 0 & 0 & -\frac{1}{p_z^2}\end{bmatrix}.$$

It is easy to verify that $\mathbf{B}_1$ is full rank, since it is comprised of block-diagonal identity matrices, as well as the 3×3 upper-triangular matrix $\boldsymbol{\zeta}$, which is itself full rank (since $\frac{1}{p_z}\neq 0$). Hence, we can study the unobservable modes of VINS by examining the right nullspace of $\mathbf{B}_2$.

The 15×18 matrix $\mathbf{B}_2$ is rank deficient by exactly four, and these four unobservable modes are spanned by the columns of the following matrix

$$\mathbf{N}=\begin{bmatrix}\mathbf{0}_3 & \frac{\partial\mathbf{s}}{\partial\boldsymbol{\theta}}\mathbf{C}\,\mathbf{g}\\ \mathbf{0}_3 & \mathbf{0}_{3\times 1}\\ \mathbf{0}_3 & -\lfloor\mathbf{v}\times\rfloor\mathbf{g}\\ \mathbf{0}_3 & \mathbf{0}_{3\times 1}\\ \mathbf{I}_3 & -\lfloor\mathbf{p}\times\rfloor\mathbf{g}\\ \mathbf{I}_3 & -\lfloor\mathbf{p}_f\times\rfloor\mathbf{g}\end{bmatrix}. \tag{3.46}$$

By multiplying $\mathbf{B}_2$ from the right with $\mathbf{N}$, it is straightforward to verify that $\mathbf{N}$ is indeed the right nullspace of $\mathbf{B}_2$ [see (3.45) and (3.46)]. We note that the first three columns of $\mathbf{N}$ correspond to globally translating the feature and the IMU-camera sensor pair together, while the fourth column corresponds to global rotations about the gravity vector.

We further prove that there are no additional right nullspace directions by showing that the 15×18 matrix $\mathbf{B}_2$ has rank 14 (note that if $\mathbf{B}_2$ had 5 or more right nullspace

directions, then it would be of rank 13 or less). To do so, we examine the left nullspace of $\mathbf{B}_2$. Specifically, we postulate that $\mathbf{B}_2$ has a left nullspace comprising the block elements $\mathbf{M}_1, \ldots, \mathbf{M}_5$, i.e.,

$$\mathbf{0} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{M}_2 & \mathbf{M}_3 & \mathbf{M}_4 & \mathbf{M}_5 \end{bmatrix} \begin{bmatrix} \lfloor {}^I\mathbf{p}_f \times \rfloor \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C} & \mathbf{C} \\ \lfloor \mathbf{C}\,\mathbf{v} \times \rfloor \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} & \mathbf{0}_3 & \mathbf{C} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \lfloor \mathbf{C}\,\mathbf{g} \times \rfloor \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$$

Based on the relationships involving the second and fourth block columns of $\mathbf{B}_2$, we see that $\mathbf{M}_3\mathbf{I}_3 = \mathbf{0}$ and $\mathbf{M}_5\mathbf{I}_3 = \mathbf{0}$, which can only hold if both $\mathbf{M}_3$ and $\mathbf{M}_5$ are zero. From the third and sixth columns of $\mathbf{B}_2$ we see that $\mathbf{M}_2\mathbf{C} = \mathbf{0}$ and $\mathbf{M}_1\mathbf{C} = \mathbf{0}$, which again can only hold if $\mathbf{M}_1$ and $\mathbf{M}_2$ are zero, since the rotation matrix $\mathbf{C}$ is full rank. Thus far, the only potentially nonzero element in the left nullspace of $\mathbf{B}_2$ is $\mathbf{M}_4$. By writing the relationship involving the first block column of $\mathbf{B}_2$, we obtain

$$\mathbf{M}_4 \lfloor \mathbf{C}\,\mathbf{g} \times \rfloor \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} = \mathbf{0}.$$

The matrix $\frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}}$ is full rank, hence, the only nonzero $\mathbf{M}_4$ which can satisfy this relationship is

$$\mathbf{M}_4 = \pm \left(\mathbf{C}\,\mathbf{g}\right)^T.$$

Therefore, we conclude that $\mathbf{B}_2$ has a one dimensional left nullspace, i.e.,

$$\mathbf{M} = \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \left(\mathbf{C}\,\mathbf{g}\right)^T & \mathbf{0}_{1\times 3} \end{bmatrix}. \tag{3.47}$$

Since $\mathbf{B}_2$ is a matrix of dimensions 15×18 with exactly one left null vector [see (3.47)], it is of rank 14. Applying this fact to determine the dimension of the right nullspace, we see that the right nullspace comprises $18 - 14 = 4$ directions, which are spanned by $\mathbf{N}$ [see (3.46)]. ■

3.6 VINS Observability Analysis

In this section, we examine the observability properties of the linearized VINS model in the general case when a single point feature is observed by a sensor platform performing arbitrary motion.[2] Specifically, we first study and analytically determine the

[2] An accompanying technical report [77] is available online: `http://www-users.cs.umn.edu/~joel/_files/Joel_Hesch_TR12.pdf`

four unobservable directions of the *ideal* linearized VINS model (i.e., the system whose Jacobians are evaluated at the true states). Subsequently, we show that the linearized VINS model used by the EKF, whose Jacobians are evaluated using the current state estimates, has only three unobservable directions (i.e., the ones corresponding to global translation), while the one corresponding to global rotation about the gravity vector becomes (erroneously) observable. The key findings of this analysis are then employed in Section 3.6.3 for improving the consistency of the EKF-based VINS.

The Observability matrix $\mathbf{M}$ is defined as a function of the linearized measurement model, $\mathbf{H}$, and the discrete-time state transition matrix, $\mathbf{\Phi}$ [8]. These, in turn, are functions of the linearization point, $\mathbf{x}^\star$, i.e.,

$$\mathbf{M}\left(\mathbf{x}^\star\right) = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2\mathbf{\Phi}_{2,1} \\ \vdots \\ \mathbf{H}_k\mathbf{\Phi}_{k,1} \end{bmatrix} \tag{3.48}$$

where $\mathbf{\Phi}_{k,1} = \mathbf{\Phi}_{k-1}\cdots\mathbf{\Phi}_1$ is the state transition matrix from time-step 1 to k, and $\mathbf{H}_k$, is the jacobian of the measurement model [see (3.17)], for the feature observation at time-step k. If $\mathbf{M}\left(\mathbf{x}^\star\right)$ was full column rank, then the linearized VINS model would be observable. However, as we will show in the following analysis, $\mathbf{M}\left(\mathbf{x}^\star\right)$ is rank deficient and hence the VINS model is unobservable. More importantly, the number of unobservable directions (right nullspace dimensions) differs depending on the selection of the linearization point (i.e., $\mathbf{x}^\star = \mathbf{x}$ in the ideal model, or $\mathbf{x}^\star = \hat{\mathbf{x}}$ for the estimated one).

3.6.1 Observability analysis of the ideal linearized VINS model

In the ideal linearized VINS model, the corresponding Jacobians are evaluated at the true system state (i.e., $\mathbf{x}^\star = \mathbf{x}$). Based on this definition, the first block-row of $\mathbf{M}\left(\mathbf{x}\right)$ is written as [see (3.20)] (for $i = 1$ feature):

$$\mathbf{H}_k = \mathbf{\Psi}_1 \begin{bmatrix} \mathbf{\Psi}_2 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix} \tag{3.49}$$

where

$$\mathbf{\Psi}_1 = \mathbf{H}_{c,k}\mathbf{C}\left({}^{I_k}\bar{q}_G\right) \tag{3.50}$$

$$\mathbf{\Psi}_2 = \lfloor {}^{G}\mathbf{f} - {}^{G}\mathbf{p}_{I_k} \times \rfloor \mathbf{C}\left({}^{I_k}\bar{q}_G\right)^T \tag{3.51}$$

and ${}^{I_k}\bar{q}_G$, denotes the rotation of $\{G\}$ with respect to frame $\{I_k\}$ at time-step $k = 1$. We note that here we focus on the observation of a single point, for the purpose of simplifying the presentation, but the analysis is extensible to the case of multiple features.

To compute the remaining block rows of the observability matrix, we require $\mathbf{\Phi}_{k,1}$, which is defined from the following matrix differential equation [8]:

$$\dot{\mathbf{\Phi}}_{k,1} = \mathbf{F}\mathbf{\Phi}_{k,1} \tag{3.52}$$

$$\text{initial condition } \mathbf{\Phi}_{1,1} = \mathbf{I}_{18} \tag{3.53}$$

where $\mathbf{F}$ is defined in (3.14). By examining the block elements of (3.52), we can obtain a solution analytically. For example, the $(2,1)$ element of $\dot{\mathbf{\Phi}}_{k,1}$ is the product of the second block row of $\mathbf{F}$ [i.e., $\mathbf{F}^{(2,:)} = \mathbf{0}_{3\times 18}$, see (3.14)] and the first block column of $\mathbf{\Phi}_{k,1}$ [i.e., $\mathbf{\Phi}_{k,1}^{(:,1)} = \begin{bmatrix}\mathbf{I}_3 & \mathbf{0}_{3\times 15}\end{bmatrix}^T$, see (3.52)].[3] Hence $\dot{\mathbf{\Phi}}_{k,1}^{(2,1)} = \mathbf{0}_3$, and recalling the initial condition, $\mathbf{\Phi}_{1,1}^{(2,1)} = \mathbf{0}_3$ [see (3.53)], we obtain

$$\mathbf{\Phi}_{k,1}^{(2,1)} = \mathbf{0}_3. \tag{3.54}$$

Following a similar approach, we can determine the other block elements of $\mathbf{\Phi}_{k,1}$ that are either $\mathbf{0}_3$ or $\mathbf{I}_3$, respectively. Specifically, $\mathbf{\Phi}_{k,1}$ has the following structure:

$$\mathbf{\Phi}_{k,1} = \begin{bmatrix} \mathbf{\Phi}_{k,1}^{(1,1)} & \mathbf{\Phi}_{k,1}^{(1,2)} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{\Phi}_{k,1}^{(3,1)} & \mathbf{\Phi}_{k,1}^{(3,2)} & \mathbf{I}_3 & \mathbf{\Phi}_{k,1}^{(3,4)} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{\Phi}_{k,1}^{(5,1)} & \mathbf{\Phi}_{k,1}^{(5,2)} & \delta t_k \mathbf{I}_3 & \mathbf{\Phi}_{k,1}^{(5,4)} & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}, \tag{3.55}$$

where $\delta t_k = \delta t(k-1)$, is the time difference between time-steps 1 and k.

[3] The superscript notations $\mathbf{E}^{(i,:)}$ and $\mathbf{E}^{(:,i)}$ refer to the i-th block row and block column of matrix $\mathbf{E}$, respectively, while $\mathbf{E}^{(i,j)}$ references the block element (i,j).

Of the remaining block elements, we only require a few in analytic form here, the others we provide explicitly in [77]. We begin by computing $\mathbf{\Phi}_{k,1}^{(1,1)}$. Proceeding from equation (3.52),

$$\begin{aligned}
\dot{\mathbf{\Phi}}_{k,1}^{(1,1)} &= \mathbf{F}^{(1,:)}\mathbf{\Phi}_{k,1}^{(:,1)} \\
&= \begin{bmatrix} -\lfloor {}^{I_k}\boldsymbol{\omega} \times \rfloor & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_{k,1}^{(1,1)} \\ \mathbf{0}_3 \\ \mathbf{\Phi}_{k,1}^{(3,1)} \\ \mathbf{0}_3 \\ \mathbf{\Phi}_{k,1}^{(5,1)} \\ \mathbf{0}_3 \end{bmatrix} \\
&= -\lfloor {}^{I_k}\boldsymbol{\omega} \times \rfloor \mathbf{\Phi}_{k,1}^{(1,1)}. \qquad (3.56)
\end{aligned}$$

Thus, the solution for $\mathbf{\Phi}_{k,1}^{(1,1)}$ is computed as

$$\begin{aligned}
\mathbf{\Phi}_{k,1}^{(1,1)} &= \mathbf{\Phi}_{1,1}^{(1,1)} \exp\left(\int_{t_1}^{t_k} -\lfloor {}^{I_\tau}\boldsymbol{\omega} \times \rfloor \mathrm{d}\tau \right) \\
&= \exp\left(-\int_{t_1}^{t_k} \lfloor {}^{I_\tau}\boldsymbol{\omega} \times \rfloor \mathrm{d}\tau \right) \\
&= \mathbf{C}({}^{I_k}\bar{q}_{I_1}), \qquad (3.57)
\end{aligned}$$

where we have employed the initial condition $\mathbf{\Phi}_{1,1}^{(1,1)} = \mathbf{I}_3$. We follow an analogous approach to compute the other elements pertinent to the observability study, i.e.,

$$\mathbf{\Phi}_{k,1}^{(1,2)} = -\int_{t_1}^{t_k} \mathbf{C}\left({}^{I_k}\bar{q}_{I_\tau}\right) \mathrm{d}\tau \qquad (3.58)$$

$$\mathbf{\Phi}_{k,1}^{(3,1)} = -\lfloor ({}^{G}\mathbf{v}_{I_k} - {}^{G}\mathbf{v}_{I_1}) + {}^{G}\mathbf{g}\,\delta t_k \times \rfloor \mathbf{C}\left({}^{G}\bar{q}_{I_1}\right) \qquad (3.59)$$

$$\mathbf{\Phi}_{k,1}^{(5,1)} = \lfloor {}^{G}\mathbf{p}_{I_1} + {}^{G}\mathbf{v}_{I_1}\delta t_k - \frac{1}{2}{}^{G}\mathbf{g}\,\delta t_k^2 - {}^{G}\mathbf{p}_{I_k} \times \rfloor \mathbf{C}({}^{G}\bar{q}_{I_1}) \qquad (3.60)$$

$$\mathbf{\Phi}_{k,1}^{(5,2)} = \int_{t_1}^{t_k} \int_{t_1}^{\theta} \mathbf{C}({}^{G}\bar{q}_{I_s}) \lfloor {}^{I_s}\mathbf{a} \times \rfloor \int_{t_1}^{s} \mathbf{C}({}^{I_s}\bar{q}_{I_\tau}) \,\mathrm{d}\tau \,\mathrm{d}s \,\mathrm{d}\theta \qquad (3.61)$$

$$\mathbf{\Phi}_{k,1}^{(5,4)} = -\int_{t_1}^{t_k} \int_{t_1}^{s} \mathbf{C}({}^{G}\bar{q}_{I_\tau}) \,\mathrm{d}\tau \,\mathrm{d}s. \qquad (3.62)$$

Using these expressions, we can obtain the k-th block row of $\mathbf{M}$, for any $k > 1$, by

multiplying out (3.49), i.e.,

$$\begin{aligned}\mathbf{M}_k &= \mathbf{H}_k \mathbf{\Phi}_{k,1} \\ &= \mathbf{\Gamma}_1 \begin{bmatrix} \mathbf{\Gamma}_2 & \mathbf{\Gamma}_3 & -\delta t_k \mathbf{I}_3 & \mathbf{\Gamma}_4 & -\mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix}\end{aligned} \tag{3.63}$$

where

$$\mathbf{\Gamma}_1 = \mathbf{H}_{c,k} \mathbf{C}\left({}^{I_k}\bar{q}_G\right) \tag{3.64}$$

$$\mathbf{\Gamma}_2 = \lfloor {}^{G}\mathbf{f} - {}^{G}\mathbf{p}_{I_1} - {}^{G}\mathbf{v}_{I_1}\delta t_k + \frac{1}{2}{}^{G}\mathbf{g}\,\delta t_k^2 \times \rfloor \mathbf{C}\left({}^{I_1}\bar{q}_G\right)^T \tag{3.65}$$

$$\mathbf{\Gamma}_3 = \lfloor {}^{G}\mathbf{f} - {}^{G}\mathbf{p}_{I_k} \times \rfloor \mathbf{C}^T\left({}^{I_k}\bar{q}_G\right)\mathbf{\Phi}_{k,1}^{(1,2)} - \mathbf{\Phi}_{k,1}^{(5,2)} \tag{3.66}$$

$$\mathbf{\Gamma}_4 = -\mathbf{\Phi}_{k,1}^{(5,4)}. \tag{3.67}$$

We note that for generic motions (i.e., $\boldsymbol{\omega} \neq \mathbf{0}_{3\times 1}$, $\mathbf{a} \neq \mathbf{0}_{3\times 1}$) both $\mathbf{\Gamma}_3$ and $\mathbf{\Gamma}_4$ are time varying matrices, whose columns are linearly independent. The structure of the remaining block elements, $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$, is employed to form a basis of the nullspace of $\mathbf{M}$ analytically.

At this point, we state the main result of our analysis:

Theorem 3.1 *The right nullspace* $\mathbf{N}_1$ *of the observability matrix* $\mathbf{M}(\mathbf{x})$ *[see* (3.48)*] of the linearized VINS model*

$$\mathbf{M}(\mathbf{x})\mathbf{N}_1 = \mathbf{0} \tag{3.68}$$

is spanned by the following four directions:

$$\mathbf{N}_1 = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}\left({}^{I_1}\bar{q}_G\right){}^{G}\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & -\lfloor {}^{G}\mathbf{v}_{I_1} \times \rfloor {}^{G}\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\lfloor {}^{G}\mathbf{p}_{I_1} \times \rfloor {}^{G}\mathbf{g} \\ \mathbf{I}_3 & -\lfloor {}^{G}\mathbf{f} \times \rfloor {}^{G}\mathbf{g} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{t,1} & | & \mathbf{N}_{r,1} \end{bmatrix}. \tag{3.69}$$

Proof 3.1 *The fact that* $\mathbf{N}_1$ *is indeed the right nullspace of* $\mathbf{M}(\mathbf{x})$ *can be verified by*

multiplying each block row of $\mathbf{M}$ *[see* (3.63)*] with* $\mathbf{N}_{t,1}$ *and* $\mathbf{N}_{r,1}$ *in* (3.69)*. Specifically,*

$$\begin{aligned}\mathbf{M}_k\mathbf{N}_{t,1} &= \mathbf{\Gamma}_1 \begin{bmatrix}\mathbf{\Gamma}_2 & \mathbf{\Gamma}_3 & -\delta t_k\mathbf{I}_3 & \mathbf{\Gamma}_4 & -\mathbf{I}_3 & \mathbf{I}_3\end{bmatrix} \begin{bmatrix}\mathbf{0}_3\\ \mathbf{0}_3\\ \mathbf{0}_3\\ \mathbf{0}_3\\ \mathbf{I}_3\\ \mathbf{I}_3\end{bmatrix} \\ &= \mathbf{\Gamma}_1(-\mathbf{I}_3+\mathbf{I}_3) = \mathbf{0}_{2\times 3} \end{aligned} \tag{3.70}$$

while,

$$\begin{aligned}\mathbf{M}_k\mathbf{N}_{r,1} &= \mathbf{\Gamma}_1 \begin{bmatrix}\mathbf{\Gamma}_2 & \mathbf{\Gamma}_3 & -\delta t_k\mathbf{I}_3 & \mathbf{\Gamma}_4 & -\mathbf{I}_3 & \mathbf{I}_3\end{bmatrix} \begin{bmatrix}\mathbf{C}\left({}^{I_1}\bar{q}_G\right){}^G\mathbf{g}\\ \mathbf{0}_{3\times 1}\\ -\lfloor {}^G\mathbf{v}_{I_1}\times\rfloor {}^G\mathbf{g}\\ \mathbf{0}_{3\times 1}\\ -\lfloor {}^G\mathbf{p}_{I_1}\times\rfloor {}^G\mathbf{g}\\ -\lfloor {}^G\mathbf{f}\times\rfloor {}^G\mathbf{g}\end{bmatrix} \\ &= \mathbf{\Gamma}_1(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{I_1} - {}^G\mathbf{v}_{I_1}\delta t_k + \frac{1}{2}{}^G\mathbf{g}\,\delta t_k^2\times\rfloor \mathbf{C}\left({}^{I_1}\bar{q}_G\right)^T\mathbf{C}\left({}^{I_1}\bar{q}_G\right){}^G\mathbf{g} \\ &\quad + \delta t_k\lfloor {}^G\mathbf{v}_{I_1}\times\rfloor {}^G\mathbf{g} + \lfloor {}^G\mathbf{p}_{I_1}\times\rfloor {}^G\mathbf{g} - \lfloor {}^G\mathbf{f}\times\rfloor {}^G\mathbf{g}) \\ &= \mathbf{\Gamma}_1\left((\lfloor {}^G\mathbf{f} - {}^G\mathbf{f}\times\rfloor {}^G\mathbf{g}) + (\lfloor -{}^G\mathbf{p}_{I_1} + {}^G\mathbf{p}_{I_1}\times\rfloor {}^G\mathbf{g})\right) \\ &\quad + \mathbf{\Gamma}_1\left(\lfloor -{}^G\mathbf{v}_{I_1} + {}^G\mathbf{v}_{I_1}\times\rfloor {}^G\mathbf{g}\delta t_k\right) + \mathbf{\Gamma}_1\left(\lfloor {}^G\mathbf{g}\times\rfloor {}^G\mathbf{g}\right)\frac{1}{2}\delta t_k^2 = \mathbf{0}_{2\times 1}\end{aligned} \tag{3.71}$$

Since $\mathbf{M}_k\mathbf{N}_{t,1} = \mathbf{0}$ *and* $\mathbf{M}_k\mathbf{N}_{r,1} = \mathbf{0}$*,* $\forall k \geq 1$ *it follows that* $\mathbf{M}\mathbf{N}_1 = \mathbf{0}$*. Hence* $\mathbf{N}_1$ *belongs to the right nullspace of* $\mathbf{M}$*. The fact that the right nullspace contains only the four directions of* $\mathbf{N}_1$ *follows from the structure of* $\mathbf{\Gamma}_3$ *and* $\mathbf{\Gamma}_4$*, which are full rank and time varying [see* (3.66) *and* (3.67)*].*

Remark 1 *The* 18×3 *block column* $\mathbf{N}_{t,1}$ *corresponds to global translations, i.e., translating both the sensing platform and the landmark by the same amount.*

Remark 2 *The* 18×1 *column* $\mathbf{N}_{r,1}$ *corresponds to global rotations of the sensing platform and the landmark about the gravity vector.*

3.6.2 Observability analysis of the EKF linearized VINS model

Ideally, any VINS estimator should employ a linearized system with an unobservable subspace that matches the true unobservable directions (3.69), both in number and

structure. However, when linearizing about the estimated state $\hat{\mathbf{x}}$, $\widehat{\mathbf{M}} = \mathbf{M}(\hat{\mathbf{x}})$ gains rank due to errors in the state estimates across time [77]. In particular, the last two block elements of $\mathbf{M}_k$ comprise identity matrices, and hence, are not a function of the linearization point. This, in effect, preserves the left nullspace directions corresponding to translation, i.e.,

$$\mathbf{M}(\hat{\mathbf{x}})\,\mathbf{N}_{t,1} = \mathbf{0}. \tag{3.72}$$

In contrast, the remaining of the block elements of (3.63), in particular $\mathbf{\Gamma}_2$, *are* functions of the linearization point. Over time, evaluating the system and measurement Jacobians at the current state estimate invalidates the structure in (3.65), due to the presence of linearization errors. This causes the direction corresponding to global rotations (which hits $\mathbf{\Gamma}_2$), $\mathbf{N}_{r,1}$, not to be in the nullspace of $\mathbf{M}(\hat{\mathbf{x}})$, and as a result the rank of the observability matrix $\widehat{\mathbf{M}}$ corresponding to the EKF linearized VINS model increases by one. This effect can also be verified by numerically evaluating the observability matrix during any experiment.

In order to address this issue, in the next section we describe our methodology for adapting existing linearized VINS estimation approaches in order to account for our knowledge of the number and structure of the unobservable directions. We pursue this strategy since it provides a simple direct extension of existing methods, which allows us to estimate the full VINS state. We note that there are potentially other ways to restrict the directions in which an estimator gains information. For example, it may be possible to decompose the system model, into observable and unobservable states (analogous to the Kalman canonical decomposition for linear systems), and construct an estimator around only the subsystem comprising the observable states. While this approach may sound appealing, it contains several limiting factors. First, although decomposing a *linear* system into observable and unobservable modes is a trivial matter, for *nonlinear* systems, particularly high dimensional ones such as VINS, this is not a straightforward task. Second, even if such a decomposition is obtained, the resulting estimates of the observable states may not be useful for navigation purposes (most notable, no estimate would be computed for the unobservable global yaw angle). Third, although work exists [70] to examine the observable modes of VINS, it has focussed only on the deterministic solution under the simplifying assumption that the gyroscopes are

bias free.

3.6.3 OC-VINS: Algorithm Description

Hereafter, we present our OC-VINS algorithm which enforces the observability constraints dictated by the VINS system structure. Rather than changing the linearization points explicitly (e.g., as in [9]), we maintain the nullspace, $\mathbf{N}_k$, at each time step, and use it to enforce the unobservable directions. We refer to the first set of block rows of $\mathbf{N}_k$ as the nullspace corresponding to the robot state, which we term $\mathbf{N}_k^R$, whereas the last block row of $\mathbf{N}_k$ is the nullspace corresponding to the feature state, i.e., $\mathbf{N}_k^f$. Specifically, the 15×4 nullspace sub-block, $\mathbf{N}_k^R$, corresponding to the robot state is analytically defined as [see (3.69) and [77]]:

$$\mathbf{N}_1^R = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}\left({}^I\hat{\bar{q}}_{G,1|1}\right){}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & -\lfloor {}^G\hat{\mathbf{v}}_{I,1|1} \times \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{I,1|1} \times \rfloor {}^G\mathbf{g} \end{bmatrix}$$

$$\mathbf{N}_k^R = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}\left({}^I\hat{\bar{q}}_{G,k|k-1}\right){}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & -\lfloor {}^G\hat{\mathbf{v}}_{I,k|k-1} \times \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{I,k|k-1} \times \rfloor {}^G\mathbf{g} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{t,k}^R & | & \mathbf{N}_{r,k}^R \end{bmatrix}. \tag{3.73}$$

The 3×4 nullspace sub-block, $\mathbf{N}_k^f$, corresponding to the feature state, is a function of the feature estimate at time t_ℓ when it was initialized, i.e.,

$$\mathbf{N}_k^f = \begin{bmatrix} \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{f_{\ell|\ell}} \times \rfloor {}^G\mathbf{g} \end{bmatrix}. \tag{3.74}$$

Modification of the State Transition Matrix Φ

During the propagation step, we must ensure that (3.75) is satisfied. We expand (3.75) by substituting the definitions of the state transition matrix (3.55) and the nullspace

for both the robot state (3.73) and the feature (3.74), i.e.,

$$\begin{aligned} & \mathbf{N}_{k+1} = \Phi_{k+1,k}\mathbf{N}_k \\ \Leftrightarrow \quad & \begin{bmatrix} \mathbf{N}_{k+1}^R \\ \mathbf{N}_{k+1}^f \end{bmatrix} = \begin{bmatrix} \Phi_{k+1,k}^R & \mathbf{0}_{15\times 3} \\ \mathbf{0}_{3\times 15} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{N}_k^R \\ \mathbf{N}_k^f \end{bmatrix} \end{aligned}$$

which, after multiplying out, provides two relationships that should be satisfied:

$$\mathbf{N}_{k+1}^R = \Phi_{k+1,k}^R \, \mathbf{N}_k^R \tag{3.75}$$

$$\mathbf{N}_{k+1}^f = \mathbf{N}_k^f. \tag{3.76}$$

From the definition of $\mathbf{N}_k^f$ [see (3.74)], it is clear that (3.76) holds automatically, and does not require any modification of $\Phi_{k+1,k}$. However, (3.75) will in general not hold, and hence it requires changing $\Phi_{k+1,k}^R$ such that $\mathbf{N}_{k+1}^R = \Phi_{k+1,k}^R \, \mathbf{N}_k^R$.

In order to determine which elements of $\Phi_{k+1,k}^R$ should be modified to satisfy (3.75), we further analyze the structure of this constraint. To do so, we partition $\mathbf{N}_k^R$ into two components: (i) the first three columns corresponding to the unobservable translation, $\mathbf{N}_{t,k}^R$, and (ii) the fourth column corresponding to the unobservable rotation about the gravity vector, $\mathbf{N}_{r,k}^R$ [see (3.69)]. We rewrite (3.75) based on this partitioning to obtain:

$$\begin{aligned} & \mathbf{N}_{k+1}^R = \Phi_{k+1,k}^R \, \mathbf{N}_k^R \\ \Leftrightarrow \quad & \begin{bmatrix} \mathbf{N}_{t,k+1}^R & \mathbf{N}_{r,k+1}^R \end{bmatrix} = \Phi_{k+1,k}^R \begin{bmatrix} \mathbf{N}_{t,k}^R & \mathbf{N}_{r,k}^R \end{bmatrix} \end{aligned}$$

which is equivalent to satisfying the following two relationships simultaneously, i.e.,

$$\mathbf{N}_{t,k+1}^R = \Phi_{k+1,k}^R \, \mathbf{N}_{t,k}^R \tag{3.77}$$

$$\mathbf{N}_{r,k+1}^R = \Phi_{k+1,k}^R \, \mathbf{N}_{r,k}^R. \tag{3.78}$$

Treating these in order, we see that (3.77) is automatically satisfied, since every block row results in $\mathbf{0}_3 = \mathbf{0}_3$ or $\mathbf{I}_3 = \mathbf{I}_3$, i.e.,

$$\begin{aligned} & \mathbf{N}_{t,k+1}^R = \Phi_{k+1,k}^R \, \mathbf{N}_{t,k}^R \\ \Leftrightarrow \quad & \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{I}_3 \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{31} & \Phi_{32} & \mathbf{I}_3 & \Phi_{34} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \Phi_{51} & \Phi_{52} & \delta t \mathbf{I}_3 & \Phi_{54} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{I}_3 \end{bmatrix}. \end{aligned}$$

We proceed by expanding the second relationship element-wise [see (3.78)] and we obtain

$$\mathbf{N}_{r,k+1}^{R} = \Phi_{k+1,k}^{R}\,\mathbf{N}_{r,k}^{R} \Leftrightarrow$$
$$\begin{bmatrix} \mathbf{C}\left({}^{I}\hat{\bar{q}}_{G,k+1|k}\right){}^{G}\mathbf{g} \\ \mathbf{0}_{3\times 1} \\ -\lfloor {}^{G}\hat{\mathbf{v}}_{I,k+1|k}\times\rfloor {}^{G}\mathbf{g} \\ \mathbf{0}_{3\times 1} \\ -\lfloor {}^{G}\hat{\mathbf{p}}_{I,k+1|k}\times\rfloor {}^{G}\mathbf{g} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{31} & \Phi_{32} & \mathbf{I}_3 & \Phi_{34} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \Phi_{51} & \Phi_{52} & \delta t\mathbf{I}_3 & \Phi_{54} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{C}\left({}^{I}\hat{\bar{q}}_{G,k|k-1}\right){}^{G}\mathbf{g} \\ \mathbf{0}_{3\times 1} \\ -\lfloor {}^{G}\hat{\mathbf{v}}_{I,k|k-1}\times\rfloor {}^{G}\mathbf{g} \\ \mathbf{0}_{3\times 1} \\ -\lfloor {}^{G}\hat{\mathbf{p}}_{I,k|k-1}\times\rfloor {}^{G}\mathbf{g} \end{bmatrix}.$$

From the first block row we have that

$$\begin{aligned} \mathbf{C}\left({}^{I}\hat{\bar{q}}_{G,k+1|k}\right){}^{G}\mathbf{g} &= \Phi_{11}\mathbf{C}\left({}^{I}\hat{\bar{q}}_{G,k|k-1}\right){}^{G}\mathbf{g} \\ \Rightarrow\ \Phi_{11} &= \mathbf{C}\left({}^{I,k+1|k}\hat{\bar{q}}_{I,k|k-1}\right). \end{aligned} \tag{3.79}$$

The requirements for the third and fifth block rows are:

$$\Phi_{31}\mathbf{C}\left({}^{I}\hat{\bar{q}}_{G,k|k-1}\right){}^{G}\mathbf{g} = \lfloor {}^{G}\hat{\mathbf{v}}_{I,k|k-1}\times\rfloor {}^{G}\mathbf{g} - \lfloor {}^{G}\hat{\mathbf{v}}_{I,k+1|k}\times\rfloor {}^{G}\mathbf{g} \tag{3.80}$$

$$\begin{aligned} \Phi_{51}\mathbf{C}\left({}^{I}\hat{\bar{q}}_{G,k|k-1}\right){}^{G}\mathbf{g} = \delta t\lfloor {}^{G}\hat{\mathbf{v}}_{I,k|k-1}\times\rfloor {}^{G}\mathbf{g} + \lfloor {}^{G}\hat{\mathbf{p}}_{I,k|k-1}\times\rfloor {}^{G}\mathbf{g} \\ - \lfloor {}^{G}\hat{\mathbf{p}}_{I,k+1|k}\times\rfloor {}^{G}\mathbf{g}. \end{aligned} \tag{3.81}$$

both of which are of the form $\mathbf{A}\mathbf{u} = \mathbf{w}$, where $\mathbf{u}$ and $\mathbf{w}$ comprise nullspace elements that are fixed [see (3.73)], and we seek to find a perturbed $\mathbf{A}^*$, for $\mathbf{A} = \Phi_{31}$ and $\mathbf{A} = \Phi_{51}$ that fulfills the constraint. To compute the minimum perturbation, $\mathbf{A}^*$, of $\mathbf{A}$, we formulate the following minimization problem

$$\min_{\mathbf{A}^*} ||\mathbf{A}^* - \mathbf{A}||_{\mathcal{F}}^2 \quad , \quad \text{s.t. } \mathbf{A}^*\mathbf{u} = \mathbf{w} \tag{3.82}$$

where $||\cdot||_{\mathcal{F}}$ denotes the Frobenius matrix norm. After employing the method of Lagrange multipliers, and solving the corresponding KKT optimality conditions [16], the optimal $\mathbf{A}^*$ that fulfills (3.82) is:

$$\mathbf{A}^* = \mathbf{A} - (\mathbf{A}\mathbf{u} - \mathbf{w})(\mathbf{u}^T\mathbf{u})^{-1}\mathbf{u}^T. \tag{3.83}$$

In summary, satisfying (3.75) only requires modifying three block elements of Φ_k during each propagation step. Specifically, we compute the modified Φ_{11} from (3.79), and Φ_{31} and Φ_{51} from (3.82)-(3.83) and construct the observability-constrained discrete-time state transition matrix. We then proceed with covariance propagation [see (3.16)].

Modification of the Measurement Jacobian H

During each update step, we seek to satisfy (3.68), i.e., $\mathbf{H}_k\mathbf{N}_k = \mathbf{0}$. Based on (3.20), (3.73), and (3.74) we can write this relationship *per feature* as

$$\mathbf{H}_c \begin{bmatrix} \mathbf{H}_\theta & \mathbf{0}_{3\times 9} & \mathbf{H}_p & | & \mathbf{H}_f \end{bmatrix} \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}\left({}^I\hat{\bar{q}}_{G,k|k-1}\right){}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & -\lfloor {}^G\hat{\mathbf{v}}_{I,k|k-1} \times \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{I,k|k-1} \times \rfloor {}^G\mathbf{g} \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{f_{\ell|\ell}} \times \rfloor {}^G\mathbf{g} \end{bmatrix} = \mathbf{0}. \tag{3.84}$$

The first block column of (3.84) requires that $\mathbf{H}_f = -\mathbf{H}_p$. Hence, we rewrite the second block column of (3.84) as

$$\mathbf{H}_c \begin{bmatrix} \mathbf{H}_\theta & \mathbf{H}_p \end{bmatrix} \begin{bmatrix} \mathbf{C}\left({}^I\hat{\bar{q}}_{G,k|k-1}\right){}^G\mathbf{g} \\ \left(\lfloor {}^G\hat{\mathbf{p}}_{f_{\ell|\ell}} \times \rfloor - \lfloor {}^G\hat{\mathbf{p}}_{I,k|k-1} \times \rfloor\right){}^G\mathbf{g} \end{bmatrix} = \mathbf{0}$$

This is a constraint of the form $\mathbf{A}\mathbf{u} = \mathbf{0}$, where $\mathbf{u}$ is a fixed quantity determined by elements in the nullspace, and $\mathbf{A}$ comprises elements of the measurement Jacobian $\mathbf{H}_k$. We compute the optimal $\mathbf{A}^*$ that satisfies this relationship using (3.82)-(3.83), which is a special case of this optimization problem when $\mathbf{w} = \mathbf{0}$. After computing the optimal $\mathbf{A}^*$, we recover the Jacobian as

$$\mathbf{H}_c\mathbf{H}_\theta = \mathbf{A}^*_{1:2,1:3} \tag{3.85}$$

$$\mathbf{H}_c\mathbf{H}_p = \mathbf{A}^*_{1:2,4:6} \tag{3.86}$$

$$\mathbf{H}_c\mathbf{H}_f = -\mathbf{A}^*_{1:2,4:6} \tag{3.87}$$

where the subscripts (i:j, m:n) denote the matrix sub-block spanning rows i to j, and columns m to n. After computing the modified measurement Jacobian, we proceed with the filter update as described in Section 3.3.2.

3.6.4 Application to the MSC-KF

The MSC-KF [49] is a VINS that performs tightly-coupled visual-inertial odometry over a sliding window of M poses, while maintaining linear complexity in the number

of observed features. The key advantage of the MSC-KF is that it exploits all the constraints for each feature observed by the camera over M poses, without requiring to build a map or estimate the features as part of the state vector. We hereafter describe how to apply our OC-VINS methodology to the MSC-KF.

Each time the camera records an image, the MSC-KF creates a stochastic clone [79] of the sensor pose. This enables the MSC-KF to use delayed image measurements; in particular, it allows all of the observations of a given feature $\mathbf{p}_{f_i}$ to be processed during a single update step (when the first pose that observed the feature is about to be marginalized). Every time the current pose is cloned, we also clone the corresponding nullspace elements to obtain an augmented nullspace, i.e.,

$$\mathbf{N}_k^{aug} = \begin{bmatrix} \mathbf{N}_k \\ \mathbf{N}_{k,clone} \end{bmatrix}$$

$$\text{where} \quad \mathbf{N}_{k,clone} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}\left({}^I\hat{\bar{q}}_{G,k|k-1}\right){}^G\mathbf{g} \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{I,k|k-1} \times \rfloor {}^G\mathbf{g} \end{bmatrix}.$$

During propagation, the current state estimate evolves forward in time by integrating (3.8)-(3.13), while the current clone poses are static. Moreover, we employ (3.79) and solve in closed form the optimization problem (3.82) for the constraints (3.80)-(3.81), using (3.83), so as to compute the observability-constrained discrete-time state transition matrix $\mathbf{\Phi}_{k+1,k}$, and propagate the covariance as

$$\mathbf{P}_{k+1|k}^{aug} = \mathbf{\Phi}_{k+1,k}^{aug}\mathbf{P}_{k|k}^{aug}\mathbf{\Phi}_{k+1,k}^{augT} + \begin{bmatrix} \mathbf{Q}_k & \mathbf{0}_{15\times 6M} \\ \mathbf{0}_{6M\times 15} & \mathbf{0}_{6M} \end{bmatrix}$$

$$\mathbf{\Phi}_{k+1,k}^{aug} = \begin{bmatrix} \mathbf{\Phi}_{k+1,k} & \mathbf{0}_{15\times 6M} \\ \mathbf{0}_{6M\times 15} & \mathbf{I}_{6M} \end{bmatrix}$$

where $\mathbf{P}_{i|j}^{aug}$ denotes the covariance of the augmented state corresponding to M cloned poses, along with the current state.

During the MSC-KF update step, we process all measurements of the features observed by the M-th clone (i.e., the one about to be marginalized from the sliding window of poses). We use (3.85)-(3.87) to compute the observability-constrained measurement Jacobian, $\widehat{\mathbf{H}}_k$, for each measurement and stack all observations of the i-th feature across

M time steps into a large measurement vector

$$\begin{bmatrix} \tilde{\mathbf{z}}_k \\ \vdots \\ \tilde{\mathbf{z}}_{k-M} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_k \\ \vdots \\ \mathbf{H}_{k-M} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{aug} \\ \tilde{\mathbf{p}}_f \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}_k \\ \vdots \\ \boldsymbol{\eta}_{k-M} \end{bmatrix} = \mathbf{H}_x \tilde{\mathbf{x}}^{aug} + \mathbf{H}_f \tilde{\mathbf{p}}_f + \boldsymbol{\eta} \tag{3.88}$$

where $\mathbf{H}_x$ and $\mathbf{H}_f$ are the Jacobians corresponding to the augmented state vector $\tilde{\mathbf{x}}^{aug}$, and to the feature, respectively. To avoid including $\mathbf{p}_f$ into the state, we marginalize it by projecting (3.88) onto the left nullspace of $\mathbf{H}_f$, which we term $\mathbf{W}$. This yields

$$\mathbf{W}^T \tilde{\mathbf{z}} = \mathbf{W}^T \mathbf{H}_x \tilde{\mathbf{x}}^{aug} + \mathbf{W}^T \boldsymbol{\eta} \quad \Rightarrow \quad \tilde{\mathbf{z}}' = \mathbf{H}'_x \tilde{\mathbf{x}}^{aug} + \boldsymbol{\eta}',$$

which we employ to update the state estimate and covariance using the standard EKF update equations [49].

3.7 Simulations

We conducted Monte-Carlo simulations to evaluate the impact of the proposed Observability-Constrained VINS (OC-VINS) method on estimator consistency. We applied the proposed methodology to two VINS systems: (i) Visual Simultaneous Localization and Mapping (V-SLAM) (see Section 3.7.1), and (ii) the Multi-state Constraint Kalman Filter (MSC-KF), which performs visual-inertial localization without constructing a map (see Section 3.7.2).

3.7.1 Simulation 1: Application of the proposed framework to V-SLAM

In this section, we present the results of applying our proposed OC-VINS to V-SLAM, which we term OC-V-SLAM. We compared its performance to the standard V-SLAM (Std-V-SLAM), as well as the ideal V-SLAM that linearizes about the true state.[4]

Specifically, we computed the Root Mean Squared Error (RMSE) and Normalized Estimation Error Squared (NEES) over 20 trials in which the camera-IMU platform traversed a circular trajectory of radius 5 m at an average velocity of 60 cm/s. The

[4] Since the ideal V-SLAM has access to the true state, it is not realizable in practice, but we include it here as a baseline comparison.

camera[5] observed visual features distributed on the interior wall of a circumscribing cylinder with radius 6 m and height 2 m [see Fig. 3.2(e)]. The effect of inconsistency during a single run is depicted in Fig. 3.7.1. The error and corresponding 3σ bounds of uncertainty are plotted for the rotation about the gravity vector. It is clear that the Std-V-SLAM gains spurious information, hence reducing its 3σ bounds of uncertainty, while the Ideal-V-SLAM and the OC-V-SLAM do not. The Std-V-SLAM becomes inconsistent on this run as the orientation errors fall outside of the uncertainty bounds, while both the Ideal-V-SLAM and the OC-V-SLAM remain consistent. Figure 3.2 also displays the RMSE and NEES plots, in which we observe that the OC-V-SLAM attains orientation accuracy and consistency levels similar to the Ideal-V-SLAM, while significantly outperforming the Std-V-SLAM. Similarly, the OC-V-SLAM obtains better positioning accuracy compared to the Std-V-SLAM.

3.7.2 Simulation 2: Application of the proposed framework to MSC-KF

We applied our OC-VINS methodology to the MSC-KF, which we term the OC-MSC-KF. In the MSC-KF framework, all the measurements to a given OF are incorporated during a single update step of the filter, after which each OF is marginalized. Hence, in the OC-MSC-KF, we do not maintain the sub-blocks of the nullspace corresponding to the features [i.e., $\mathbf{N}_{\mathbf{f}_i}$, $i = 1, \ldots, N$, see (3.74)]. Instead, we propagate forward only the portion of the nullspace corresponding to the sensor platform state, and we form the feature nullspace block for each feature, only when it is processed in an update.

We conducted Monte-Carlo simulations to evaluate the consistency of the proposed method applied to the MSC-KF [82]. Specifically, we compared the standard MSC-KF (Std-MSC-KF), with the Observability-Constrained MSC-KF (OC-MSC-KF), which is obtained by applying the methodology described in Section 3.6.3, as well as the Ideal-MSC-KF, whose Jacobians are linearized at the true states, which we use as a benchmark. We evaluated the RMSE and NEES over 30 trials (see Fig. 3.3) in which the camera-IMU platform traversed a circular trajectory of radius 5 m at an average speed of 60 cm/s, and observed 50 randomly distributed features per image. As evident, the

[5] The camera had a 45 degree field of view, with $\sigma_{px} = 1$ px, while the IMU was modeled after MEMS quality sensors.

OC-MSC-KF outperforms the Std-MSC-KF and attains performance almost indistinguishable from the Ideal-MSC-KF in terms of RMSE and NEES.

3.8 Experimental Results

The proposed OC-VINS framework has been validated experimentally and compared with standard VINS approaches. Specifically, we evaluated the performance of OC-V-SLAM (Section 3.8.2) and OC-MSC-KF (Section 3.8.3 and Section 3.8.4) on both indoor and outdoor datasets. In our experimental setup, we utilized a light-weight sensing platform comprised of an InterSense NavChip IMU and a PointGrey Chameleon camera (see Fig. 3.4). During the indoor experimental tests (see Section 3.8.2 and Section 3.8.3), the sensing platform was mounted on an Ascending Technologies Pelican quadrotor equipped with a VersaLogic Core 2 Duo single board computer. For the outdoor dataset, the sensing platform was head-mounted on a bicycle helmet (see Section 3.8.4), and interfaced to a handheld Sony Vaio. We hereafter provide an overview of the system implementation, along with a discussion of the experimental setup and results.

3.8.1 Implementation remarks

The image processing is separated into two components: one for extracting and tracking short-term OFs, and one for extracting DFs to use in V-SLAM.

OFs are extracted from images using the Shi-Tomasi corner detector [83]. After acquiring image k, it is inserted into a sliding window buffer of m images, $\{k-m+1, k-m+2, \ldots, k\}$. We then extract features from the first image in the window and track them pairwise through the window using the KLT tracking algorithm [84]. To remove outliers from the resulting tracks, we use a two-point algorithm to find the essential matrix between successive frames. Specifically, given the filter's estimated rotation (from the gyroscopes' measurements) between image i and j, ${}^{i}\hat{\bar{q}}_{j}$, we estimate the essential matrix from only two feature correspondences. This approach is more robust than the five-point algorithm [85] because it provides two solutions for the essential matrix rather than up to ten. Moreover, it requires only two data points, and thus it reaches a consensus with fewer hypotheses when used in a RANSAC framework.

The DFs are extracted using SIFT descriptors [42]. To identify global features

observed from several different images, we first utilize a vocabulary tree (VT) structure for image matching [86]. Specifically, for an image taken at time k, the VT is used to select which image(s) taken at times $1, 2, \ldots, k-1$ correspond to the same physical scene. Among those images that the VT reports as potential matches, the SIFT descriptors from each of them are compared to those from image k to create tentative feature correspondences. The epipolar constraint is then enforced using RANSAC and Nister's five-point algorithm [85] to eliminate outliers. It is important to note that the images used to construct the VT (offline) are not taken along our experimental trajectory, but rather are randomly selected from a set of representative images.

3.8.2 Experiment 1: Indoor validation of OC-V-SLAM

In the first experimental trial, we compared the performance of OC-V-SLAM to that of Std-V-SLAM on an indoor trajectory. The sensing platform traveled a total distance of 172.5 m, covering three loops over two floors in Walter Library at the University of Minnesota. The quadrotor was returned to its starting location at the end of the trajectory, to provide a quantitative characterization of the achieved accuracy.

Opportunistic features were tracked using a window of $m = 10$ images. Every m camera frames, up to 30 features from all available DFs are initialized and the state vector is augmented with their 3D coordinates. The process of initializing DFs [77] is continued until the occurrence of the first loop closure; from that point on, no new DFs are considered and the filter relies upon the re-observation of previously initialized DFs and the processing of OFs.

For both the Std-V-SLAM and the OC-V-SLAM, the final position error was approximately 34 cm, which is less than 0.2% of the total distance traveled (see Fig. 3.5). However, the estimated covariances from the Std-V-SLAM are smaller than those from the OC-V-SLAM (see Fig. 3.6). Furthermore, uncertainty estimates from the Std-V-SLAM decreased in directions that are unobservable (i.e., rotations about the gravity vector); this violates the observability properties of the system and demonstrates that spurious information is injected to the filter.

Figure 3.6(a) highlights the difference in estimated yaw uncertainty between the OC-V-SLAM and the Std-V-SLAM. In contrast to the OC-V-SLAM, the Std-V-SLAM

covariance rapidly decreases, violating the observability properties of the system. Similarly, large differences can be seen in the covariance estimates for the x-axis position estimates [see Fig. 3.6(b)]. The Std-V-SLAM estimates a much smaller uncertainty than the OC-V-SLAM, supporting the claim that the Std-V-SLAM tends to be inconsistent.

3.8.3 Experiment 2: Indoor validation of OC-MSC-KF

We validated the proposed OC-MSC-KF on real-world data. The first test comprised a trajectory 50 m in length that covered three loops in an indoor area, after which the testbed was returned to its initial position. At the end of the trajectory, the Std-MSC-KF had a position error of 18.73 cm, while the final error for the OC-MSC-KF was 16.39 cm (approx. 0.38% and 0.33% of the distance traveled, respectively). In order to assess the impact of inconsistency on the orientation estimates of both methods, we used as ground truth the rotation between the first and last images computed independently using BLS and feature point matches. The Std-MSC-KF had final orientation error $\begin{bmatrix}0.15 & -0.23 & -5.13\end{bmatrix}$ deg for roll, pitch, and yaw (rpy), while the rpy errors for the OC-MSC-KF were $\begin{bmatrix}0.19 & -0.20 & -1.32\end{bmatrix}$ deg, respectively.

In addition to achieving higher accuracy, for yaw in particular, the OC-MSC-KF is more conservative since it strictly adheres to the unobservable directions of the system. This is evident in both the position and orientation uncertainties. We plot the y-axis position and yaw angle uncertainties in Fig. 3.7, as representative results. Most notably, the yaw uncertainty of the OC-MSC-KF remains approximately 1.13 deg (3σ), while for the Std-MSC-KF it reduces to 0.87 deg (3σ). This indicates that the Std-MSC-KF gains spurious orientation information, which leads to inconsistency. Lastly, in Fig. 3.8 we show the 3D trajectory along with an overhead (x-y) view. It is evident that the Std-MSC-KF yaw error impacts the position accuracy, as the Std-MSC-KF trajectory exhibits a rotation with respect to the OC-MSC-KF.

3.8.4 Experiment 3: Outdoor validation of OC-MSC-KF

In our final experimental trial, we tested the OC-MSC-KF on a large outdoor dataset (approx. 1.5 km in length). Figure 3.9(a) depicts the OC-MSC-KF (red) and the Std-MSC-KF (blue) trajectory estimates, along with position markers from a low-grade

onboard GPS receiver (green). In order to assess the accuracy of both filters, the estimates are overlaid on an overhead image taken from Google-Earth.

Figure 3.9(b) depicts a zoomed-in plot of the starting location (center) for both filters, along with the final position estimates. In order to evaluate the accuracy of the proposed method, the sensing platform was returned to its starting location at the end of the trajectory. The OC-MSC-KF obtains a final position error of 4.38 m (approx. 0.3% of the distance travelled), while the Std-MSC-KF obtains a final position error of 10.97 m. This represents an improvement in performance of approximately 60%.

The filters' performance is also illustrated visually in Fig. 3.9(c) which shows a zoomed-in plot of the turn-around point. The OC-MSC-KF estimates remain on the light-brown portion of the ground (which is the sidewalk), which coincides with the true trajectory. In contrast, the Std-MSC-KF estimates drift over the dark triangles in the image, which are wading pools filled with water. This shifting of the trajectory represents a slight rotation around the vertical axis, indicating a violation of the rotation nullspace direction $\mathbf{N}_r$.

Figure 3.10 depicts the uncertainty in the position estimates along the x-axis (perpendicular to the direction of motion), along with the uncertainty in yaw (corresponding to rotations about the gravity vector). It is clear that the Std-MSC-KF reduces its uncertainty in its heading direction, indicating that the filter gains spurious information, while the OC-MSC-KF does not gain information for the rotation around the gravity vector.

3.9 Summary

In this chapter, we analyzed the inconsistency of VINS from the standpoint of observability. Specifically, we showed that standard EKF-based filtering approaches lead to spurious information gain since they do not adhere to the unobservable directions of the true system. Furthermore, we introduced an observability-constrained VINS approach to mitigate estimator inconsistency by enforcing the nullspace explicitly. We presented extensive simulation and experimental results to support our claims and validated the proposed estimator, by applying it to both V-SLAM and the MSC-KF.

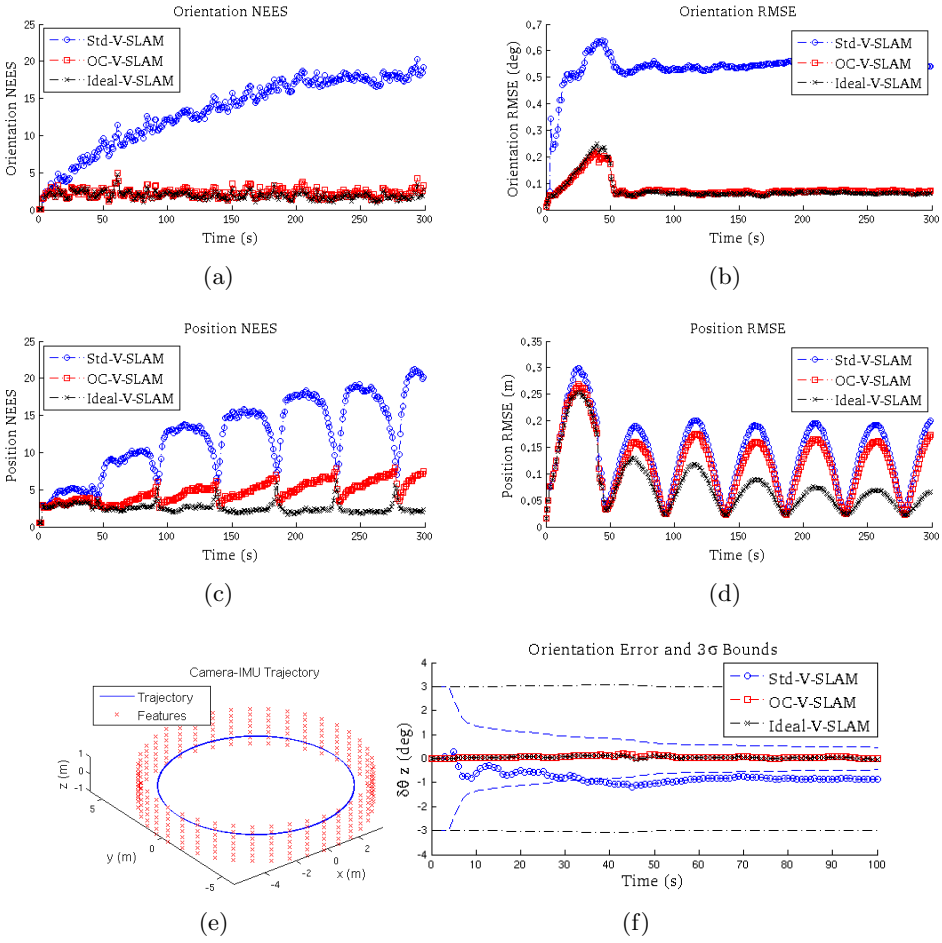


Figure 3.2: Simulation 1: The RMSE and NEES errors for orientation (a)-(b) and position (d)-(e) plotted for all three filters, averaged per time step over 20 Monte Carlo trials. (c) Camera-IMU trajectory and 3D features. (f) Error and 3σ bounds for the rotation about the gravity vector, plotted for the first 100 sec of a representative run.

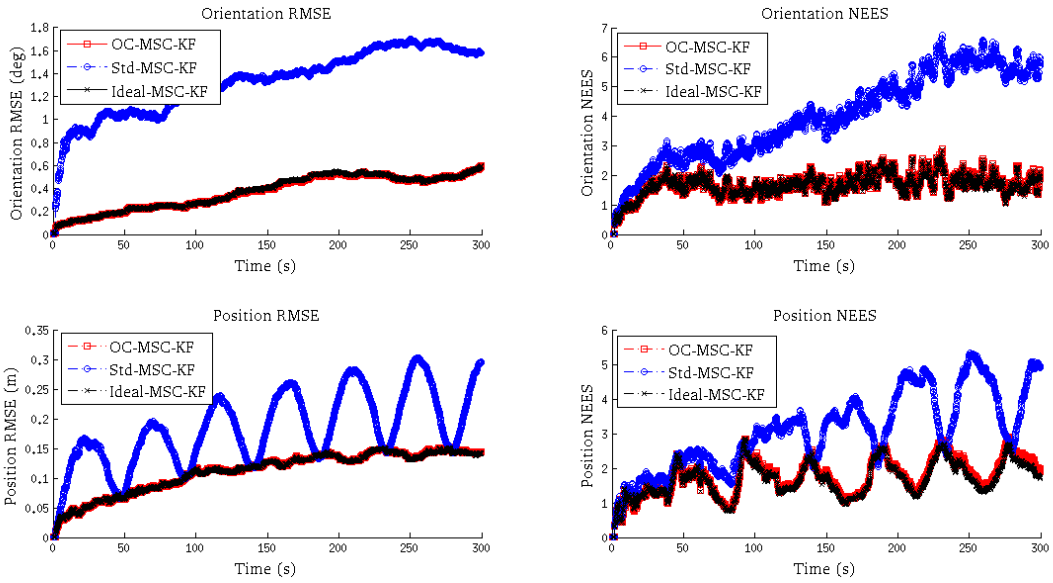
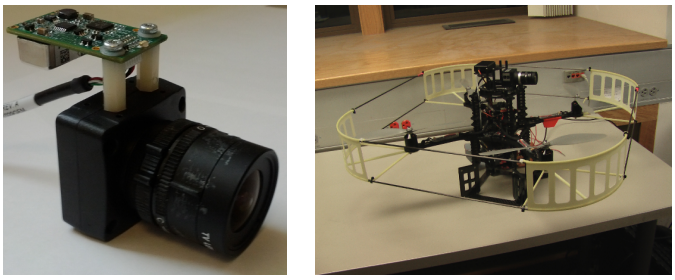


Figure 3.3: Simulation 2: The average RMSE and NEES over 30 Monte-Carlo simulation trials for orientation (above) and position (below). Note that the OC-MSC-KF attains performance almost indistinguishable to the Ideal-MSC-KF.



(a) (b)

Figure 3.4: (a) The experimental testbed comprises a light-weight InterSense NavChip IMU and a Point Grey Chameleon Camera. IMU signals are sampled at a frequency of 100 Hz while camera images are acquired at 7.5 Hz. The dimensions of the sensing package are approximately 6 cm tall, by 5 cm wide, by 8 cm deep. (b) An AscTech Pelican on which the camera-IMU package was mounted during the indoor experiments (see Section 3.8.2 and Section 3.8.3).

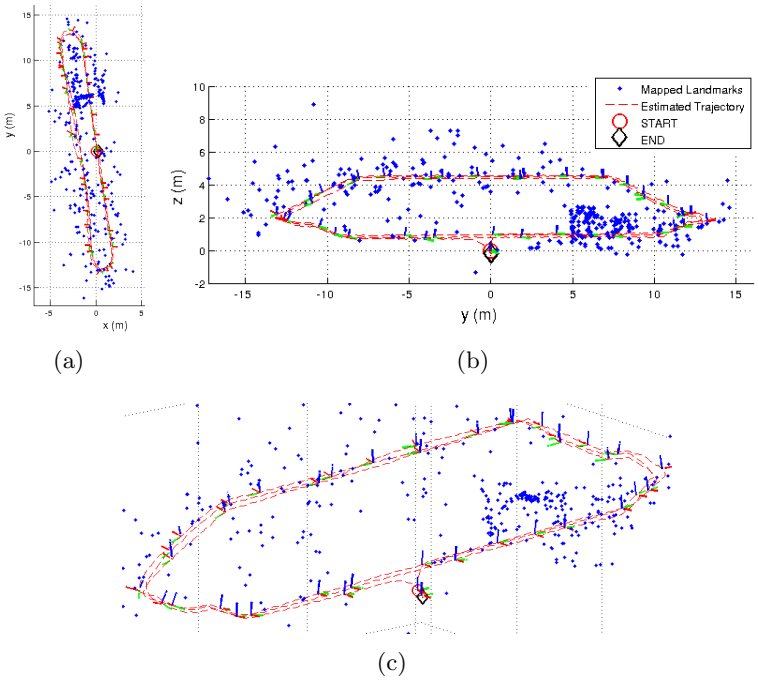


Figure 3.5: Experiment 1: The estimated 3D trajectory over the three traversals of the two floors of the building, along with the estimated positions of the persistent features. (a) projection on the x and y axis, (b) projection on the y and z axis, (c) 3D view of the overall trajectory and the estimated features.

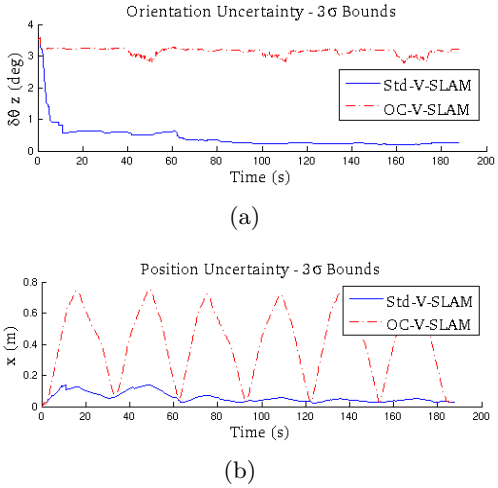
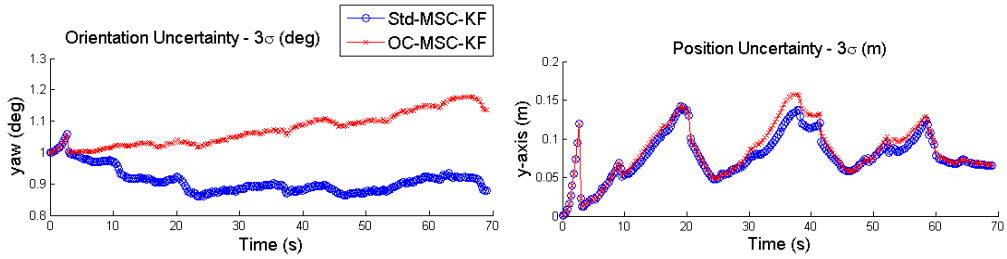
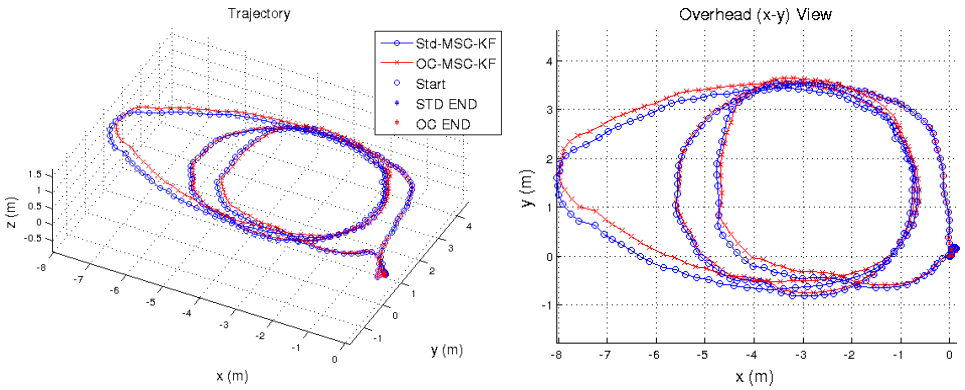


Figure 3.6: Experiment 1: Comparison of the estimated 3σ error bounds for attitude and position between Std-V-SLAM and OC-V-SLAM.



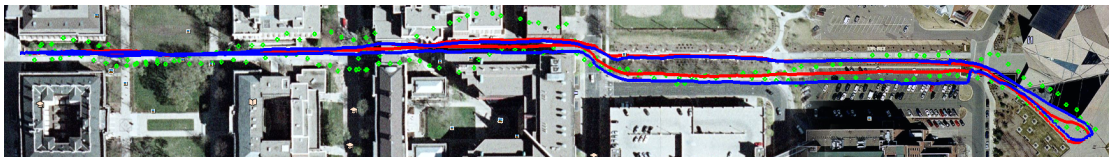
(a) (b)

Figure 3.7: Experiment 2: The position (a) and orientation (b) uncertainties (3σ bounds) for the yaw angle and the y-axis, which demonstrate that the Std-MSC-KF gains spurious information about its orientation.

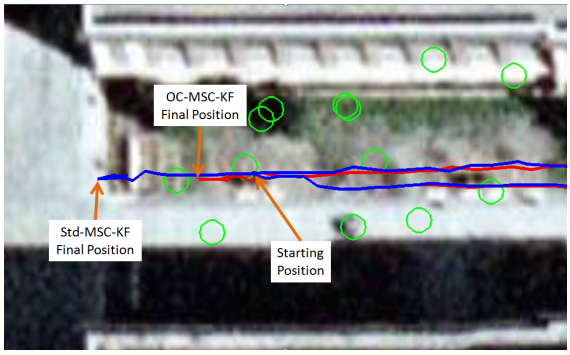


(a) (b)

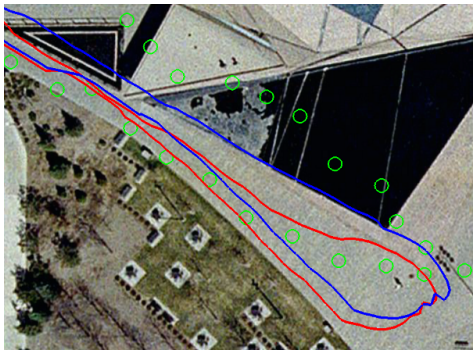
Figure 3.8: Experiment 2: The 3D trajectory (a) and corresponding overhead (x-y) view (b).



(a)

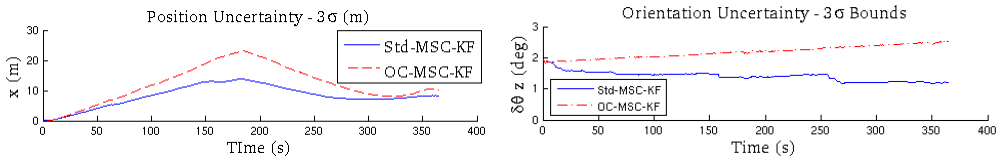


(b)



(c)

Figure 3.9: Experiment 3: (a) An outdoor experimental trajectory covering 1.5 km across the University of Minnesota campus. The red (blue) line denotes the OC-MSC-KF (Std-MSC-KF) estimated trajectory. The green circles denote a low-quality GPS-based estimate of the position across the trajectory. (b) A zoom-in view of the beginning / end of the run. Both filters start with the same initial pose estimate, however, the error for the Std-MSC-KF at the end of the run is 10.97 m, while for the OC-MSC-KF the final error is 4.38 m (an improvement of approx. 60%). Furthermore, the final error for the OC-MSC-KF is approximately 0.3% of the distance traveled. (c) A zoomed-in view of the turn-around point. The Std-MSC-KF trajectory is shifted compared to the OC-MSC-KF, which remains on the path (light-brown region).



(a) (b)

Figure 3.10: Experiment 3: (a) Position uncertainty along the x-axis (perpendicular to the direction of motion) for the Std-MSC-KF, and OC-MSC-KF respectively. The OC-MSC-KF maintains more conservative estimates for position, indicating that the Std-MSC-KF may be inconsistent. (b) Orientation uncertainty about the vertical axis (z). Since rotations about gravity are unobservable, the Std-MSC-KF should not gain any information in this direction. However, as evident from this plot, the Std-MSC-KF uncertainty reduces, indicating inconsistency. For the OC-MSC-KF, the uncertainty does not decrease, indicating that the OC-MSC-KF respects the unobservable system directions.

Chapter 4

Observability-constrained Vision-only Navigation

4.1 Introduction

In egocentric vision tasks, it is often necessary to maintain an estimate of the camera's pose over time as the person moves around. For example, a navigation aid for the visually impaired (e.g., [87]) must estimate its pose as the person walks, in order to provide them with turn-by-turn directions from point A to B. In a human-worn augmented reality system (e.g., [88]), maintaining the camera pose along with the environment structure, is necessary to annotate the scene with information.

Numerous vision-based localization approaches have been presented in the literature, including methods based on the EKF [89], UKF [90], BLS [91, 92], and PF [60]. While most existing works focus on vision navigation systems working in real-time [89] or providing dense-realistic maps [92], a key issue that has not yet been addressed in the literature is how estimator inconsistency impacts monocular localization. As defined in [7], a state estimator is consistent if the estimation errors are zero-mean and have covariance smaller than or equal to the one calculated by the filter. As we will demonstrate, a leading cause of inconsistency in monocular localization is due to spurious information gained about the scale of the scene, which is unobservable (i.e., scale cannot be determined using a monocular camera alone).

Until recently, little attention was paid to the effects that observability properties can

have on nonlinear estimator consistency. The work by Huang et al. [9, 10, 11] was the first to identify this connection for several 2D localization problems (i.e., simultaneous localization and mapping, cooperative localization). The authors showed that, for these problems, a mismatch exists between the number of unobservable directions of the true nonlinear system and the linearized system used for estimation purposes. In particular, the estimated (linearized) system has one-fewer unobservable direction than the true system, allowing the estimator to surreptitiously gain spurious information along the direction corresponding to global orientation (yaw). This increases the estimation errors while reducing the estimator uncertainty, and leads to inconsistency.

In this chapter, we analyze and improve consistency for monocular Simultaneous Localization and Mapping (MonoSLAM). The main contributions of this work are:

- We provide an overview of the MonoSLAM observability analysis using the system observability matrix, and show that seven d.o.f. are unobservable. These correspond to three-d.o.f. global translation, three-d.o.f. global rotation, and global scale.

- We report on MonoSLAM inconsistency, and demonstrate that a standard EKF-based MonoSLAM approach can gain spurious information about the scale of the system, leading to estimator inconsistency.

- We introduce an Observability-Constrained MonoSLAM (OC-MonoSLAM) algorithm which explicitly adheres to the system observability properties, and hence mitigates inconsistency. We validate our method with Monte-Carlo simulations and experimental results to show that it has increased consistency and lower errors compared to standard MonoSLAM.

The rest of this chapter is organized as follows: In Section 4.2, we describe the system and measurement models, followed by our analysis of MonoSLAM inconsistency in Section 4.3. The proposed estimator modification is presented in Section 4.3.1, and subsequently validated both in simulations and experimentally (Sects. 4.4 and 4.5). Finally, we provide our concluding remarks and outline our future research directions in Section 4.6.

4.2 Estimator Description

We begin with an overview of the propagation and measurement models which govern the MonoSLAM system. We adopt the EKF as our framework for fusing the camera measurements across time, and we employ a tracking model to predict the camera's motion between images. The sensing platform moves in a previously unknown environment, and localizes solely using DFs (e.g., SIFT keys [42]), which can be reliably tracked across images, and redetected when revisiting an area.[1]

4.2.1 System State and Propagation Model

The EKF estimates the camera pose, as well as its linear and rotational velocities, and a map corresponding to the 3D coordinates of features in the environment. The filter state is the $(13+3N)\times 1$ vector:

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} {}^{G}\mathbf{p}_S^T & {}^{S}\bar{q}_G^T & {}^{S}\mathbf{v}^T & {}^{S}\boldsymbol{\omega}^T & | & {}^{G}\mathbf{f}_1^T & \cdots & {}^{G}\mathbf{f}_N^T \end{bmatrix}^T \\ &= \begin{bmatrix} \mathbf{x}_s^T & | & \mathbf{x}_m^T \end{bmatrix}^T, \end{aligned} \tag{4.1}$$

where $\mathbf{x}_s(t)$ is the 13×1 sensor platform state, and $\mathbf{x}_m(t)$ is the $3N\times 1$ state of the map. The sensor platform state comprises ${}^{S}\bar{q}_G(t)$ which is the unit quaternion representing the orientation of the *global frame* $\{G\}$ in the sensor frame, $\{S\}$, at time t. The frame $\{S\}$ is attached to the camera, while $\{G\}$ is a reference frame whose origin coincides with the initial camera position. The linear and rotational velocities of the camera, ${}^{S}\mathbf{v}(t)$ and ${}^{S}\boldsymbol{\omega}(t)$, are expressed with respect to $\{S\}$, while the camera's position, ${}^{G}\mathbf{p}_S(t)$, is expressed in $\{G\}$.

The map, $\mathbf{x}_m$, comprises N DFs, ${}^{G}\mathbf{f}_i$, $i=1,\ldots,N$, and grows as new DFs are observed. With the state of the system now defined, we turn our attention to the continuous-time model we utilize to track the system state.

[1] While we focus on the case of the EKF, our observability analysis and proposed algorithm for improving consistency are extensible to any linearized estimation architecture (e.g., UKF and sliding window filter).

Continuous-time model

We employ a constant-velocity tracking model, in which both linear and rotational velocities are expressed in the *sensor* frame. This has the advantage of being more flexible than the "constant-global-velocity" model originally proposed for MonoSLAM [89], while at the same time enabling a simpler estimator framework than the Interacting Multiple Model (IMM) approach of Civera et al. [93].

$$ {}^{S}_{G}\dot{\bar{q}}(t) = \frac{1}{2}\mathbf{\Omega}({}^{S}\boldsymbol{\omega}(t)) {}^{S}\bar{q}_{G}(t), \quad {}^{S}\dot{\boldsymbol{\omega}}(t) = \boldsymbol{\eta}_{\omega} \tag{4.2} $$

$$ {}^{G}\dot{\mathbf{p}}_{S}(t) = {}^{G}\mathbf{v}_{S}(t) = {}^{S}_{G}\mathbf{C}^{T\,S}\mathbf{v}(t), \quad {}^{S}\dot{\mathbf{v}}(t) = \boldsymbol{\eta}_{v} \tag{4.3} $$

$$ {}^{G}\dot{\mathbf{f}}_{i}(t) = \mathbf{0}_{3\times 1}\ , \ \ i = 1, \ldots, N. \tag{4.4} $$

where ${}^{S}_{G}\mathbf{C}$ is the rotational matrix corresponding to ${}^{S}\bar{q}_{G}(t)$, and $\mathbf{\Omega}(\boldsymbol{\omega})$ is the matrix governing the quaternion time derivative, i.e.,

$$ \mathbf{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor \boldsymbol{\omega} \times \rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{T} & 0 \end{bmatrix}, \quad \lfloor \boldsymbol{\omega} \times \rfloor \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. $$

The time derivatives of the rotational and linear velocities, ${}^{S}\boldsymbol{\omega}$ and ${}^{S}\mathbf{v}$, are modeled as zero-mean white Gaussian processes, $\boldsymbol{\eta}_{\omega}$ and $\boldsymbol{\eta}_{v}$, respectively. The DFs belong to the static scene, thus, their time derivatives are zero [see (4.4)].

Linearizing at the current estimates and applying the expectation operator on both sides of (4.2)-(4.4), we obtain the state estimate propagation model

$$ {}^{S}_{G}\dot{\hat{\bar{q}}}(t) = \frac{1}{2}\mathbf{\Omega}(\hat{\boldsymbol{\omega}}(t)) {}^{S}_{G}\hat{\bar{q}}(t), \quad {}^{S}\dot{\hat{\boldsymbol{\omega}}}(t) = \mathbf{0}_{3\times 1} \tag{4.5} $$

$$ {}^{G}\dot{\hat{\mathbf{p}}}_{S}(t) = {}^{S}_{G}\hat{\mathbf{C}}^{T\,S}\hat{\mathbf{v}}_{S}(t), \quad {}^{S}\dot{\hat{\mathbf{v}}}(t) = \mathbf{0}_{3\times 1} \tag{4.6} $$

$$ {}^{G}\dot{\hat{\mathbf{f}}}_{i}\ (t) = \mathbf{0}_{3\times 1}\ , \ \ i = 1, \ldots, N. \tag{4.7} $$

The $(12 + 3N) \times 1$ error-state vector is defined as

$$ \begin{aligned} \widetilde{\mathbf{x}} &= \begin{bmatrix} {}^{G}\widetilde{\mathbf{p}}_{S}^{T} & {}^{S}\boldsymbol{\delta\theta}_{G}^{T} & {}^{S}\widetilde{\mathbf{v}}^{T} & {}^{S}\widetilde{\boldsymbol{\omega}}^{T} & | & {}^{G}\widetilde{\mathbf{f}}_{1}^{T} & \cdots & {}^{G}\widetilde{\mathbf{f}}_{N}^{T} \end{bmatrix}^{T} \\ &= \begin{bmatrix} \widetilde{\mathbf{x}}_{s}^{T} & | & \widetilde{\mathbf{x}}_{m}^{T} \end{bmatrix}^{T}, \end{aligned} \tag{4.8} $$

where $\widetilde{\mathbf{x}}_{s}(t)$ is the 12×1 error state corresponding to the sensing platform, and $\widetilde{\mathbf{x}}_{m}(t)$ is the $3N \times 1$ error state of the map. For the position, linear and rotational velocities,

and the map, an additive error model is utilized (i.e., $\widetilde{x} = x - \hat{x}$ is the error in the estimate $\hat{x}$ of a quantity x). However, for the quaternion we employ a multiplicative error model [44]. Specifically, the error between the quaternion $\bar{q}$ and its estimate $\hat{\bar{q}}$ is the 3×1 angle-error vector, $\boldsymbol{\delta\theta}$, implicitly defined by the error quaternion

$$\delta\bar{q} = \bar{q} \otimes \hat{\bar{q}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}^T & 1 \end{bmatrix}^T, \tag{4.9}$$

where $\delta\bar{q}$ describes the small rotation that causes the true and estimated attitude to coincide. This allows us to represent the attitude uncertainty by the 3×3 covariance matrix $\mathbb{E}[\boldsymbol{\delta\theta}\boldsymbol{\delta\theta}^T]$, which is a minimal representation.

The linearized continuous-time error-state equation is

$$\begin{aligned} \dot{\widetilde{\mathbf{x}}} &= \begin{bmatrix} \mathbf{F}_s & \mathbf{0}_{12\times 3N} \\ \mathbf{0}_{3N\times 12} & \mathbf{0}_{3N} \end{bmatrix} \widetilde{\mathbf{x}} + \begin{bmatrix} \mathbf{G}_s \\ \mathbf{0}_{3N\times 6} \end{bmatrix} \mathbf{n} \\ &= \mathbf{F}_c\, \widetilde{\mathbf{x}} + \mathbf{G}_c\, \mathbf{n}\,, \end{aligned} \tag{4.10}$$

where $\mathbf{0}_{3N}$ denotes the $3N \times 3N$ matrix of zeros, $\mathbf{n} = \begin{bmatrix} \boldsymbol{\eta}_\omega^T & \boldsymbol{\eta}_v^T \end{bmatrix}^T$ is the system noise, which is modeled as a zero-mean white Gaussian process with autocorrelation $\mathbb{E}[\mathbf{n}(t)\mathbf{n}^T(\tau)] = \mathbf{Q}_c\delta(t-\tau)$. The matrix $\mathbf{F}_s$ is the continuous-time error-state transition matrix corresponding to the camera state, and $\mathbf{G}_s$ is the continuous-time input noise matrix, i.e.,

$$\mathbf{F}_s = \begin{bmatrix} \mathbf{0}_3 & -{}^S_G\mathbf{C}^T \lfloor {}^S\mathbf{v} \times \rfloor & {}^S_G\mathbf{C}^T & \mathbf{0}_3 \\ \mathbf{0}_3 & -\lfloor {}^S\boldsymbol{\omega} \times \rfloor & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}, \quad \mathbf{G}_s = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}.$$

Discrete-time implementation

In order to propagate the state forward in time, we employ Euler integration of (4.5)–(4.7), with a specified step size, δt, selected to be significantly smaller than the camera frame-rate. Moreover, to derive the covariance propagation equation, we evaluate the discrete-time state transition matrix, $\boldsymbol{\Phi}_k$, and the discrete-time system noise covariance

matrix, $\mathbf{Q}_{d,k}$, as

$$\mathbf{\Phi}_k = \mathbf{\Phi}(t_{k+1}, t_k) = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}_c(\tau)\mathrm{d}\tau\right) \tag{4.11}$$
$$\mathbf{Q}_{d,k} = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1}, \tau)\mathbf{G}_c\mathbf{Q}_c\mathbf{G}_c^T\mathbf{\Phi}^T(t_{k+1}, \tau)\mathrm{d}\tau.$$

The propagated covariance is then computed as

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}_k\mathbf{P}_{k|k}\mathbf{\Phi}_k^T + \mathbf{Q}_{d,k}. \tag{4.12}$$

4.2.2 Measurement Update Model

As the camera moves it observes visual features. These measurements are exploited to concurrently estimate the motion of the sensing platform and the map of DFs. To simplify the discussion, we consider the observation of a single DF $\mathbf{f}_i$. The camera measures $\mathbf{z}_i$, which is the perspective projection of the 3D point, ${}^S\mathbf{f}_i = \begin{bmatrix} x & y & z \end{bmatrix}^T$, expressed in the current camera frame $\{S\}$, onto the image plane[2] , i.e.,

$$\mathbf{z}_i = \frac{1}{z}\begin{bmatrix} x \\ y \end{bmatrix} + \boldsymbol{\eta}_i, \quad {}^S\mathbf{f}_i = {}^S_G\mathbf{C}\left({}^G\mathbf{f}_i - {}^G\mathbf{p}_S\right). \tag{4.13}$$

The measurement noise, $\boldsymbol{\eta}_i$, is modeled as zero mean white Gaussian with covariance $\mathbf{R}_i$. The linearized error model is $\tilde{\mathbf{z}}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i \simeq \mathbf{H}_i\tilde{\mathbf{x}} + \eta_i$, where $\hat{\mathbf{z}}$ is the expected measurement computed by evaluating (4.13) at the current state estimate, and the measurement Jacobian, $\mathbf{H}_i$, is

$$\mathbf{H}_i = \mathbf{H}_{cam}\begin{bmatrix}\mathbf{H}_\mathbf{p} & \mathbf{0}_3 & \mathbf{H}_{\bar{q}} & \mathbf{0}_3 \,|\, \mathbf{0}_3 & \cdots & \mathbf{H}_{\mathbf{f}_i} & \cdots & \mathbf{0}_3\end{bmatrix} \tag{4.14}$$
$$\mathbf{H}_{cam} = \frac{1}{z^2}\begin{bmatrix} z & 0 & -x \\ 0 & z & -y \end{bmatrix}, \quad \mathbf{H}_\mathbf{p} = -{}^S_G\mathbf{C}$$
$$\mathbf{H}_{\bar{q}} = \lfloor {}^S_G\mathbf{C}\left({}^G\mathbf{f}_i - {}^G\mathbf{p}_S\right) \times \rfloor, \quad \mathbf{H}_{\mathbf{f}_i} = {}^S_G\mathbf{C}.$$

Here, $\mathbf{H}_{cam}$, is the Jacobian of the perspective projection with respect to ${}^S\mathbf{f}_i$, while $\mathbf{H}_{\bar{q}}$, $\mathbf{H}_\mathbf{p}$, and $\mathbf{H}_{\mathbf{f}_i}$, are the Jacobians of ${}^S\mathbf{f}_i$ with respect to ${}^S\bar{q}_G$, ${}^G\mathbf{p}_S$, and ${}^G\mathbf{f}_i$, respectively.

[2] Without loss of generality, we express the image measurement in normalized pixel coordinates [78].

For DFs that are already in the map, we directly apply the measurement model (4.13)-(4.14) to update the filter. We compute the measurement residual, the covariance of the residual, and the Kalman gain

$$\mathbf{r}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i \tag{4.15}$$

$$\mathbf{S}_i = \mathbf{H}_i \mathbf{P}_{k+1|k} {\mathbf{H}_i}^T + \mathbf{R}_i \tag{4.16}$$

$$\mathbf{K} = \mathbf{P}_{k+1|k} {\mathbf{H}_i}^T \mathbf{S}_i^{-1}. \tag{4.17}$$

Employing these quantities, we compute the EKF state and covariance update as

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}\mathbf{r}_i \tag{4.18}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|k} {\mathbf{H}_i}^T \mathbf{S}_i^{-1} \mathbf{H}_i \mathbf{P}_{k+1|k}. \tag{4.19}$$

For previously unseen DFs, we compute an initial estimate, along with covariance and cross-correlations by solving a bundle-adjustment over a short time window [14].

4.3 Observability-constrained MonoSLAM

Using the system and measurement models presented above, we hereafter describe how the system observability properties influence estimator consistency. In particular, we show that MonoSLAM has seven unobservable directions, corresponding to global translation, global rotation, and global scale. However, when using a linearized estimator, such as the EKF, errors in linearization while evaluating the system and measurement Jacobians change the directions in which information is acquired by the estimator. Over time, these directions can span the whole state space, including directions which should be unobservable. In particular, for MonoSLAM we observe that the estimator gains *scale* information, which can lead to scale drift over time. When spurious information is gained along unobservable directions, it leads to larger errors, smaller uncertainties, and inconsistency. In what follows, we first analyze the system observability properties and show why the standard MonoSLAM violates them. Subsequently, we present an Observability-Constrained MonoSLAM (OC-MonoSLAM) estimation algorithm that explicitly adheres to the observability properties of the system.

The observability matrix [8] is defined as a function of the linearized measurement model, $\mathbf{H}$, and the discrete-time state transition matrix, $\mathbf{\Phi}$, which are in turn functions

of the linearization point, $\mathbf{x}$, i.e.,

$$\mathbf{M}(\mathbf{x}) = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2\mathbf{\Phi}_{2,1} \\ \vdots \\ \mathbf{H}_k\mathbf{\Phi}_{k,1} \end{bmatrix} \tag{4.20}$$

where $\mathbf{\Phi}_{k,1} = \mathbf{\Phi}_{k-1}\cdots\mathbf{\Phi}_1$ is the state transition matrix from time step t_1 to t_k. We compute the discrete-time state transition matrix, $\mathbf{\Phi}_{k,1}$ as the solution to the following matrix differential equation,

$$\dot{\mathbf{\Phi}}_{t,t_1} = \mathbf{F}_c(t)\mathbf{\Phi}_{t,t_1}, \quad \text{initial condition } \mathbf{\Phi}_{t_1,t_1} = \mathbf{I}. \tag{4.21}$$

To simplify the discussion, we consider only a single landmark in the state vector. Using the initial condition and the structure of $\mathbf{F}_c$ [see (4.10)], we obtain $\mathbf{\Phi}_{t,t_1}$ as

$$\mathbf{\Phi}_{t,t_1} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{\Phi}_{[1,2]} & \mathbf{\Phi}_{[1,3]} & \mathbf{\Phi}_{[1,4]} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{\Phi}_{[2,2]} & \mathbf{0} & \mathbf{\Phi}_{[2,4]} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{4.22}$$

where

$$\mathbf{\Phi}_{[1,2]} = -\lfloor {}^G\mathbf{p}_{S(t)} - {}^G\mathbf{p}_{S(t1)} \times \rfloor \, {}^G_{S(t1)}\mathbf{C} \tag{4.23}$$

$$\mathbf{\Phi}_{[1,3]} = \int_{t1}^{t} {}^G_{S(\tau)}\mathbf{C}\, d\tau \tag{4.24}$$

$$\mathbf{\Phi}_{[1,4]} = -\int_{t1}^{t} \lfloor {}^G\mathbf{v}_{S(r)} \times \rfloor \int_{t1}^{r} {}^G_{S(\tau)}\mathbf{C}\, d\tau\, d\kappa \tag{4.25}$$

$$\mathbf{\Phi}_{[2,2]} = {}^{S(t)}_{S(t1)}\mathbf{C} \tag{4.26}$$

$$\mathbf{\Phi}_{[2,4]} = \int_{t1}^{t} {}^{S(t)}_{S(\tau)}\mathbf{C}\, d\tau \tag{4.27}$$

where $S(t)$ denotes the frame $\{S\}$ at time t. Employing (4.14) and (4.22), the k-th block row of the observability matrix [see (4.20)] is

$$\mathbf{H}_k\mathbf{\Phi}_{k,1} = \mathbf{A}_1 \begin{bmatrix} -\mathbf{I}_3 & \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 & \mathbf{I}_3 \end{bmatrix} \tag{4.28}$$

where

$$\mathbf{A}_1 = \mathbf{H}_{cam,k} \cdot {}^{S(k)}_{G}\mathbf{C} \tag{4.29}$$

$$\mathbf{A}_2 = \lfloor {}^{G}\mathbf{f} - {}^{G}\mathbf{p}_{S(k)} \times \rfloor \, {}^{G}_{S(1)}\mathbf{C} \tag{4.30}$$

$$\mathbf{A}_3 = -\int_{t1}^{tk} {}^{G}_{S(\tau)}\mathbf{C} \, d\tau \tag{4.31}$$

$$\mathbf{A}_4 = \lfloor {}^{G}\mathbf{f} - {}^{G}\mathbf{p}_{S(k)} \times \rfloor \int_{t1}^{tk} {}^{G}_{S(\tau)}\mathbf{C} \, d\tau - \mathbf{\Phi}_{[1,4]}. \tag{4.32}$$

It is straightforward to verify that the right nullspace of $\mathbf{M}(\mathbf{x})$ spans seven directions, i.e., $\mathbf{M}(\mathbf{x})\,\mathbf{N}_1 = \mathbf{0}$, where

$$\mathbf{N}_1 = \begin{bmatrix} \mathbf{I}_3 & -\lfloor {}^{G}\mathbf{p}_{S(1)} \times \rfloor & {}^{G}\mathbf{p}_{S(1)} \\ \mathbf{0}_3 & {}^{S(1)}_{G}\mathbf{C} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_3 & {}^{S(1)}\mathbf{v} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\lfloor {}^{G}\mathbf{f} \times \rfloor & {}^{G}\mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{t,1} & | & \mathbf{N}_{r,1} & | & \mathbf{N}_{s,1} \end{bmatrix} \tag{4.33}$$

where $\mathbf{N}_{t,1}$ corresponds to global translations of the camera and landmark together, $\mathbf{N}_{r,1}$ corresponds to global rotations of both together, and $\mathbf{N}_{s,1}$ is the direction corresponding to global scale (see Fig. 4.1).

Ideally, any estimator we employ should correspond to a system with an unobservable subspace that matches these directions, both in number and structure. However, when linearizing about the estimated state $\hat{\mathbf{x}}$, $\mathbf{M}(\hat{\mathbf{x}})$ gains rank due to errors in the state estimates across time. This can be easily verified by numerically evaluating (4.20) during any experiment. To address this problem and ensure that (4.33) is orthogonal to every block row of $\mathbf{M}$ when the state estimates are used for computing $\mathbf{H}_\ell$, and $\mathbf{\Phi}_{\ell,1}$, $\ell = 1, \ldots, k$, we must ensure that $\mathbf{H}_\ell \mathbf{\Phi}_{\ell,1} \mathbf{N}_1 = \mathbf{0}$, $\ell = 1, \ldots, k$.

One way to enforce this is by requiring that at each time step

$$\mathbf{N}_{\ell+1} = \mathbf{\Phi}_\ell \mathbf{N}_\ell \tag{4.34}$$

$$\mathbf{H}_\ell \mathbf{N}_\ell = \mathbf{0}, \quad \ell = 1, \ldots, k \tag{4.35}$$

both hold. This can be accomplished by propagating the nullspace in time and appropriately modifying $\mathbf{H}_\ell$ following the process described in the next section.

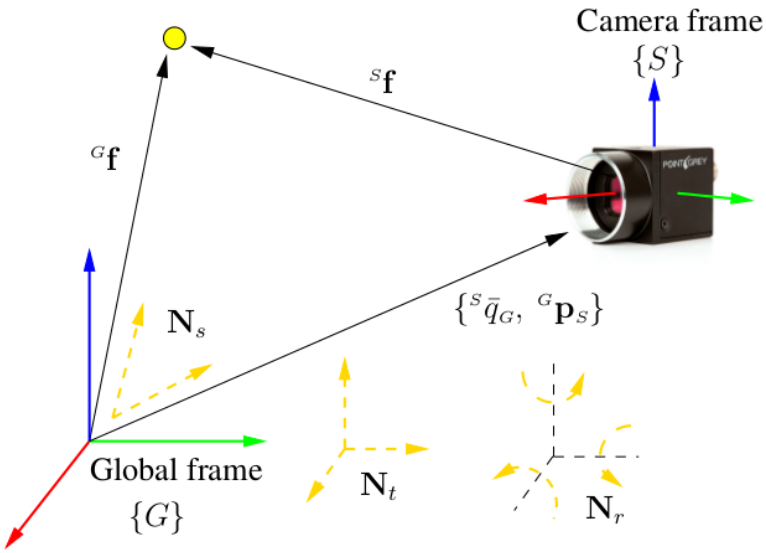


Figure 4.1: The unobservable directions are depicted in gold. $\mathbf{N}_s$ corresponds to global scale (i.e., translating the whole scene and the camera towards or away from the origin). $\mathbf{N}_t$ corresponds to global translations of the scene and camera along any of the cardinal axes. $\mathbf{N}_r$ corresponds to rotating the whole scene and the camera about the cardinal axes.

4.3.1 OC-MonoSLAM: Algorithm Description

Hereafter, we present our OC-MonoSLAM algorithm which enforces the observability constraints dictated by the MonoSLAM system structure. Rather than changing the linearization points explicitly (e.g., as in [9]), we maintain the nullspace, $\mathbf{N}_k$, at each time step, and use it to enforce the unobservable directions.

Nullspace initialization for the camera

The initial nullspace corresponding to the camera state elements is analytically defined as

$$\mathbf{N}_1 = \begin{bmatrix} \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{I,0|0} \times \rfloor & {}^G\hat{\mathbf{p}}_{I,0|0} \\ \mathbf{0}_3 & \mathbf{C}\left({}^I\hat{\bar{q}}_{G,0|0}\right) & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_3 & {}^I\hat{\mathbf{v}}_{0|0} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 1} \end{bmatrix} \tag{4.36}$$

where the $\hat{\mathbf{x}}_{i|j}$ denotes the estimate of $\mathbf{x}$ at time step i based on all measurements up to time step j. We note that in SLAM it is common to (arbitrarily) assign the global frame to coincide with the initial camera frame, while the initial velocity can be set to be unity along the estimated direction of translation between the first image pair, to set the scale. However, any other preferred method for initializing the MonoSLAM state can also be employed to initialize the nullspace.

Nullspace initialization for new landmarks

Each time a new landmark is initialized into the state vector, we must augment the nullspace, $\mathbf{N}_k$, so as to account for the new feature, and fulfill (4.34) and (4.35) at subsequent time steps. To accomplish this, we form the 3×7 block row

$$\mathbf{N}_{fi} = \begin{bmatrix} \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{f}}_{k|k} \times \rfloor & {}^G\hat{\mathbf{f}}_{k|k} \end{bmatrix} \tag{4.37}$$

which we concatenate with the current nullspace $\mathbf{N}_k$.

Nullspace propagation

During the propagation step, we need to compute the new nullspace at time $k+1$, $\mathbf{N}_{k+1}$. Based on the observability constraint (4.34), this entails propagating the nullspace from time step k to $k+1$ using the computed state transition matrix $\mathbf{\Phi}_k$.

Modification of H

During each update step, we must ensure that $\mathbf{H}_k\mathbf{N}_k = \mathbf{0}$ is satisfied. Hence, we seek a modified $\mathbf{H}_k$ that fulfills (4.35), while maintaining its structure. Based on (4.14), we can write this relationship *per feature* as

$$\mathbf{0}_{2\times 7} = \mathbf{H}_{cam} \begin{bmatrix} \mathbf{H_p} & \mathbf{H}_{\bar{q}} & \mathbf{0}_3 & \mathbf{0}_3 & | & \mathbf{H_f} \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{p}}_{I,k|k-1} \times \rfloor & {}^G\hat{\mathbf{p}}_{I,k|k-1} \\ \mathbf{0}_3 & \mathbf{C}\left({}^I\hat{\bar{q}}_{G,k|k-1}\right) & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_3 & {}^G\hat{\mathbf{v}}_{I,k|k-1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\lfloor {}^G\hat{\mathbf{f}}_{k|k-1} \times \rfloor & {}^G\hat{\mathbf{f}}_{k|k-1} \end{bmatrix} \tag{4.38}$$

The first block column of (4.38) requires that $\mathbf{H_f} = -\mathbf{H_p}$. Hence, we rewrite the second and third block columns of (4.38) as

$$\mathbf{0}_{2\times 4} = \mathbf{H}_{cam}\begin{bmatrix}\mathbf{H_p} & \mathbf{H}_{\bar{q}}\end{bmatrix}\begin{bmatrix}\lfloor {}^G\hat{\mathbf{f}}_{k|k-1} - {}^G\hat{\mathbf{p}}_{I,k|k-1}\times\rfloor & {}^G\hat{\mathbf{p}}_{I,k|k-1} - {}^G\hat{\mathbf{f}}_{k|k-1} \\ \mathbf{C}\left({}^I\hat{\bar{q}}_{G,k|k-1}\right) & \mathbf{0}_{3\times 1}\end{bmatrix} \tag{4.39}$$

This is a constraint of the form $\mathbf{0} = \mathbf{AU}$, where $\mathbf{U}$ is a fixed quantity determined by elements in the nullspace, and $\mathbf{A}$ comprises elements of the measurement Jacobian, which we seek to modify. To compute the minimum perturbation, $\mathbf{A}^*$, of $\mathbf{A}$, we formulate the following minimization problem

$$\begin{aligned}&\min_{\mathbf{A}^*} ||\mathbf{A}^* - \mathbf{A}||_{\mathcal{F}}^2 \\ &\text{subject to } \mathbf{A}^*\mathbf{U} = \mathbf{0}\end{aligned} \tag{4.40}$$

where $||\cdot||_{\mathcal{F}}$ denotes the Frobenius matrix norm. After employing the method of Lagrange multipliers, and solving the corresponding KKT optimality conditions, the optimal $\mathbf{A}^*$ that fulfills (4.40) is $\mathbf{A}^* = \mathbf{A} - \mathbf{AU}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T$. Finally, the elements of the measurement Jacobian are computed as

$$\mathbf{H}_{cam}\mathbf{H_p} = \mathbf{A}^*_{1:2,1:3} \tag{4.41}$$

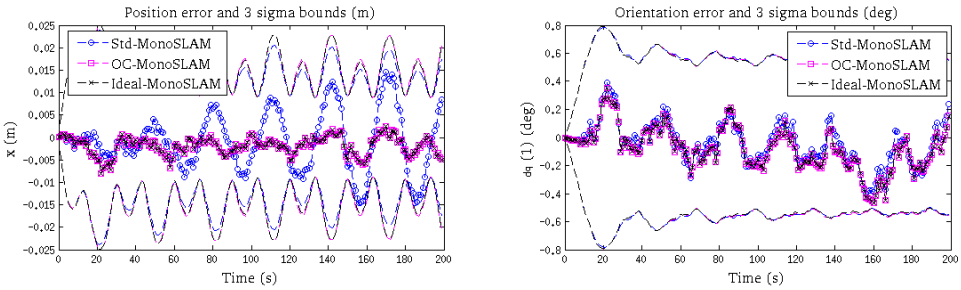
$$\mathbf{H}_{cam}\mathbf{H_f} = -\mathbf{A}^*_{1:2,1:3} \tag{4.42}$$

$$\mathbf{H}_{cam}\mathbf{H}_{\bar{q}} = \mathbf{A}^*_{1:2,4:6} \tag{4.43}$$

where the subscripts (i:j, m:n) denote the submatrix spanning rows i to j, and columns m to n. After computing the modified measurement Jacobian, we proceed with the filter update as described in Section 4.2.2.

4.4 Simulations

We conducted Monte-Carlo simulations to evaluate the impact of the proposed Observability-Constrained MonoSLAM (OC-MonoSLAM) method on estimator consistency. We compared its performance to standard MonoSLAM (Std-MonoSLAM), as well as an ideal MonoSLAM method that linearizes the Jacobians at the true state. We note that the ideal MonoSLAM is not realizable in practice, but is utilized as a benchmark for performance comparison.



(a) (b)

Figure 4.2: Errors and 3σ bounds plotted for the x-axis position (left) and $\delta\theta_1$ orientation (right) for the first 200 seconds of a representative run.

To evaluate the accuracy and consistency of the proposed approach, we computed the RMSE and NEES [7] over 50 trials in which a simulated camera traversed a circular trajectory for 500 sec at an average speed of 11 cm/s.[3] The environment contained 72 visual features distributed in a planar grid pattern, which the camera observed while moving.

The effect of inconsistency during a single run is demonstrated in Fig. 4.2 where we depict the error and corresponding 3σ bounds for the x-axis position and $\delta\theta_1$ orientation. All three filters attain comparable accuracy and uncertainty for orientation, which is not surprising since there are sufficient points in the scene to precisely track the camera's rotations. However, from the position error plot, it is clear that the 3σ bounds for the Std-MonoSLAM are smaller than for either the OC-MonoSLAM, or the Ideal-MonoSLAM. This indicates that the Std-MonoSLAM gains spurious information. Furthermore, the x-axis position error for Std-MonoSLAM starts to increase over time, eventually causing inconsistency.

Figure 4.3 displays the RMSE and NEES, in which we observe that all three filters obtain similar accuracy and consistency performance for orientation. However, the OC-MonoSLAM attains significantly better positioning accuracy and consistency compared to Std-MonoSLAM, and is almost indistinguishable from the Ideal-MonoSLAM. Based on our analysis and these results, we postulate that the key source of position error and inconsistency in the Std-MonoSLAM is violation of the unobservable scale direction [i.e.,

[3] The camera was simulated with a 45x45 deg fov, with $\sigma_{px} = 1$ px.

$\mathbf{N}_s$, see (4.33)].

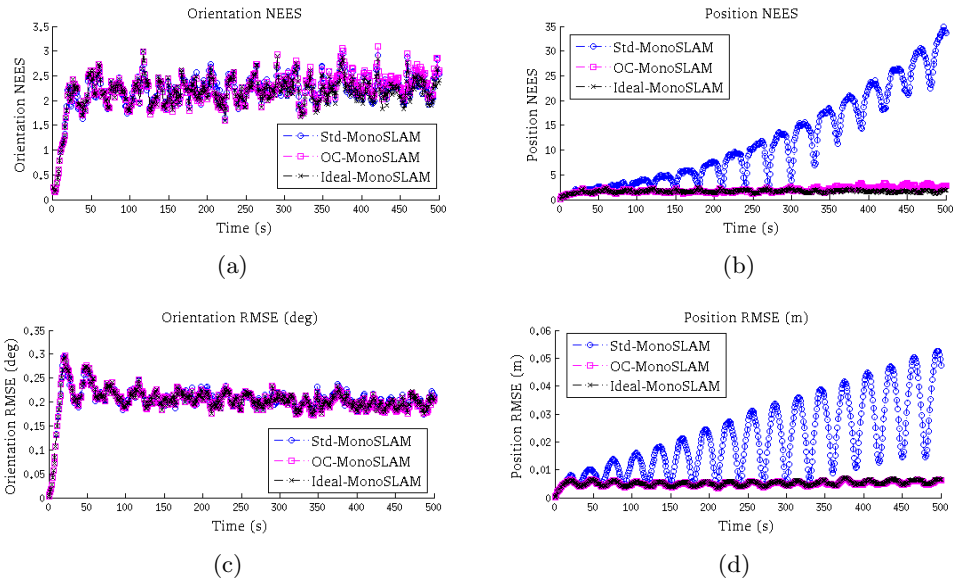


Figure 4.3: The NEES and RMSE for orientation (left) and position (right) plotted for all three filters, averaged per time step over 50 Monte-Carlo trials.

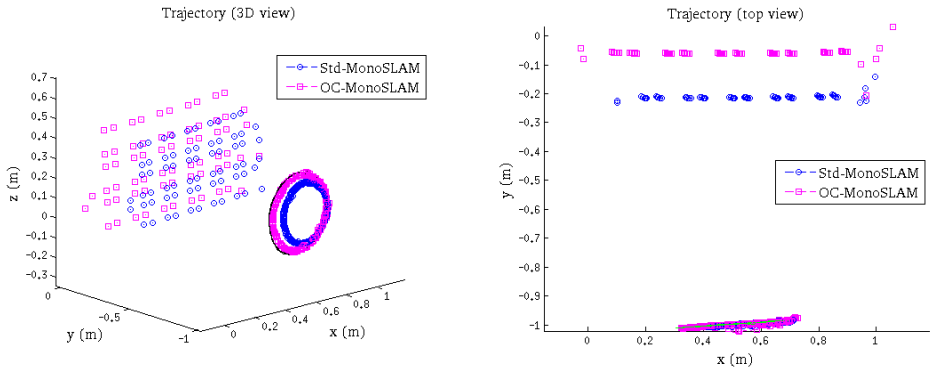
4.5 Experimental Validation

Our experimental set-up comprised a monochrome Point Grey Chameleon camera which recorded images at 7.5 Hz. We moved the camera on a circular trajectory in front of a calibration board comprising 72 corner features, whose positions are accurately known Fig. 4.4.

Using the observations of the visual features over 25 seconds (approx. 4.5 rotations), we estimated the camera trajectory and corresponding map using both the Std-MonoSLAM and the OC-MonoSLAM methods. The filters were initialized using the PnP estimate of the camera pose at the first image, along with the linear and rotation velocities computed between the first two images. In order to obtain an "approximate" ground truth trajectory, we utilized DLS-PnP [94] to compute the camera pose independently for each image, given the known landmark locations.

The estimated 3D trajectories and maps are depicted in Fig. 4.4. The PnP trajectory

is plotted in black and closely coincides with the one computed by OC-MonoSLAM, while the Std-MonoSLAM position estimates follow an estimated circular trajectory with a smaller radius (indicating inconsistent scale). The scale inconsistency is also visually apparent in Fig. 4.4 (right), which depicts a top view of the landmarks and trajectories. The true landmarks lie in the $y = 0$ cm plane, hence, the Std-MonoSLAM underestimates the depth to the scene.



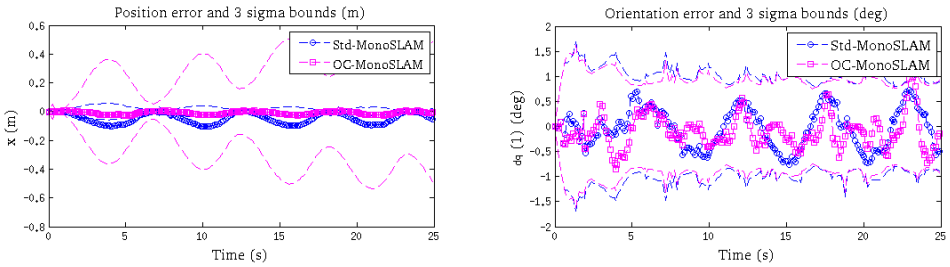
(a) (b)

Figure 4.4: (left) The estimated 3D trajectory for the Std-MonoSLAM and the OC-MonoSLAM, along with the estimated map. The PnP estimated trajectory is plotted in black, and is overlapped by the OC-MonoSLAM estimate. (right) A top view of the trajectories and landmarks. The true landmarks lie on the y = 0 plane, hence the Std-MonoSLAM underestimates the depth to the scene, demonstrating scale drift.

In Fig. 4.5, we plot the estimated 3σ bounds and corresponding errors with respect to the PnP trajectory for two representative axes (i.e., x-axis position and $\delta\theta_1$ orientation). It is evident that the orientation performance of both filters is comparable, while the OC-MonoSLAM outperforms the Std-MonoSLAM in position accuracy. In addition, the OC-MonoSLAM is more conservative that the Std-MonoSLAM in terms of position uncertainty.

4.6 Summary

In this chapter, we analyzed the inconsistency of MonoSLAM from the standpoint of observability. Specifically, we showed that using a standard EKF-based approach leads



(a) (b)

Figure 4.5: (left) The position error and corresponding 3σ bounds for the x-axis computed with respect to the PnP pose estimates. (right) The orientation error and 3σ bounds for $\delta\theta_1$.

to spurious information gain, in particular for scale, since it does not adhere to the unobservable directions of the true system. Moreover, we introduced an observability-constrained MonoSLAM method to mitigate estimator inconsistency by enforcing the nullspace explicitly. Finally, we presented simulation and experimental results to support our claims and validate the proposed estimator.

Chapter 5

Direct Least-squares PnP

5.1 Introduction

The task of determining the six-d.o.f. camera pose from observations of known points in the scene has numerous applications in computer vision and robotics. Examples include robot localization [95], spacecraft pose estimation during descent and landing [82], pose determination for model-based vision [96], as well as hand-eye calibration [97].

PnP has been studied for various numbers of points (from the minimum of 3, to the general case of n), and several different solution approaches exist, such as: (i) directly solving the nonlinear geometric constraint equations in the minimal case [98], (ii) formulating an overdetermined linear system of equations in the non-minimal case [99], and (iii) iteratively minimizing a nonlinear least-squares cost function, which accounts for the measurement noise [100].

Currently, no approach exists that directly provides all solutions for PnP ($n \geq 3$), in a Maximum-Likelihood sense, without the need for initialization or approximations in the problem treatment. Some authors have proposed methods which reach close to the global optimum, e.g., based on successive Linear Matrix Inequality (LMI) relaxations [101], transformation to a Semi-Definite Program (SDP) [102], or a geometric transformation of the problem [103]. However, these approaches are only applicable when PnP admits a unique solution, which can only be guaranteed when $n \geq 6$, and some approaches require special treatment (e.g., when all points are co-planar).

The proposed DLS method seeks to overcome the limitations of the current approaches:

- It computes all pose solutions, as the minima of a nonlinear least-squares cost function, in the general case of $n \geq 3$ points.
- No initialization is required, and the performance is consistently better than competing methods and close to that of MLE.
- The method is scalable, since the size of the nonlinear least-squares cost function which is minimized is not dependent on the number of points.

The rest of this chapter is organized as follows: Section 5.2 provides an overview of the related work on PnP. We describe our proposed approach in Section 5.3, while we present simulation and experimental comparisons to alternative approaches in Section 5.4. Lastly, we provide our concluding remarks in Section 5.5.

5.2 Related Work

The minimal PnP problem (i.e., P3P) has typically been addressed by treating the geometric constraint equations as noise-free, and solving for the camera pose [98, 104]. Haralick et al. [12] provided a comparison of the classical P3P methods and an analysis of singular configurations. Direct solutions have also been proposed for the overdetermined case (i.e., PnP, $n \geq 4$). For instance, Horaud *et al.* [105] addressed the P4P problem by connecting the four known points to form three known lines, and exploiting the nonlinear line projection equations to compute the camera pose. Linear methods (e.g., based on lifting) also exist for both P4P and PnP [99, 103, 106, 107]. Significant work has also focused on characterizing the number of solutions for P3P [108, 109, 110], and PnP [110, 111].

A key drawback of the approaches which consider noise-free measurements is that they may return inaccurate or even erroneous solutions in the presence of noise. Hence, these analytic methods are most often employed as an initialization step for an MLE of the camera pose [14].

Several authors have addressed the PnP problem from a least-squares perspective, by iteratively minimizing a cost function which is the sum of the squared errors (either

reprojection or geometric) for each point [14, 100]. These methods are more accurate, since they explicitly account for the measurement noise, and under certain noise assumptions, return the maximum-likelihood estimate of the camera pose. However, they can only compute one solution (out of possibly many), and require a good initial guess of the camera pose to converge.

Other approaches exist that seek to directly compute a global optimum without initialization. For instance, Kahl and Henrion [101] proposed a method based on a series of LMI relaxations, while Schweighofer and Pinz [102] presented an approach which first transforms the PnP problem into an SDP before optimizing for the camera pose. Unfortunately, these approaches do not provide a method for computing multiple solutions when they exist, and may require special treatment if the known points are co-planar.

In contrast to the above methods, we present a DLS approach for PnP which accounts for the measurement noise, and admits all solutions to the problem without requiring iterations or an initial guess of the camera pose. Specifically, we reparametrize the constraint equations to obtain a polynomial cost function that only depends on the unknown orientation. We then solve the corresponding optimality conditions analytically, and recover all minima (pose hypotheses) of the LS problem directly.

5.3 Problem Formulation

5.3.1 Measurement Model

The camera observation of known points in the scene projected onto the image plane can be described by the spherical camera model:

$$\mathbf{z}_i = {}^{S}\bar{\mathbf{r}}_i + \boldsymbol{\eta}_i \tag{5.1}$$

$${}^{S}\mathbf{r}_i = {}^{S}_{G}\mathbf{C}\,{}^{G}\mathbf{r}_i + {}^{S}\mathbf{p}_G \tag{5.2}$$

where $\mathbf{z}_i$ is the measurement of the unit-vector direction, ${}^{S}\bar{\mathbf{r}}_i = \frac{{}^{S}\mathbf{r}_i}{||{}^{S}\mathbf{r}_i||}$, from the sensor frame $\{S\}$ towards point i, which is corrupted by noise $\boldsymbol{\eta}_i$. The point's coordinates in the sensing frame $\{S\}$ are a function of the known coordinates, ${}^{G}\mathbf{r}_i$, in the global frame $\{G\}$, as well as the unknown global-to-sensor transformation described by the rotation matrix ${}^{S}_{G}\mathbf{C}$ and translation vector ${}^{S}\mathbf{p}_G$. Figure 5.1 depicts the observation of

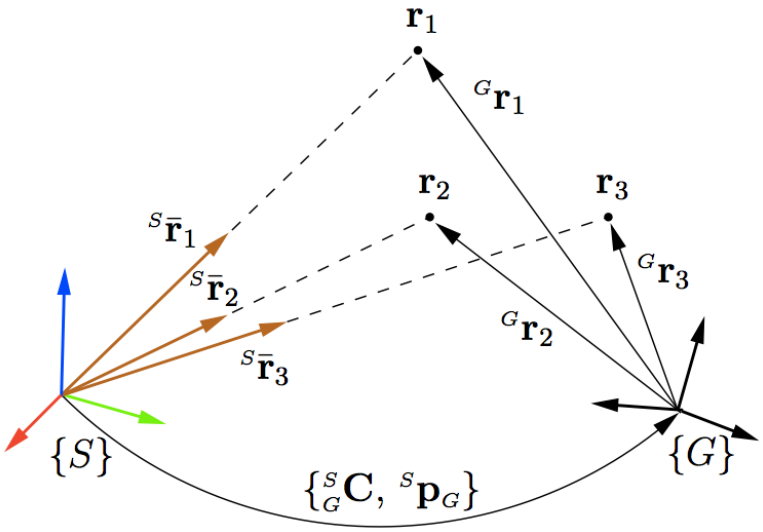


Figure 5.1: This figure depicts the observations of points $\mathbf{r}_i$, $i = 1, 2, 3$ via the unit-vector directions ${}^S\bar{\mathbf{r}}_i$ from the origin of the camera frame $\{S\}$ towards each point. The distance from $\{S\}$ to each point is $\alpha_i = ||{}^S\mathbf{r}_i||$. The vector ${}^S\mathbf{p}_G$ is the origin of $\{G\}$ with respect to $\{S\}$, the rotation matrix from $\{G\}$ to $\{S\}$ is ${}^S_G\mathbf{C}$, and ${}^G\mathbf{r}_i$ is the position of each point in $\{G\}$.

three non-collinear points, which is the minimal case required in order to be able to solve the measurement equations and recover the camera pose.

5.3.2 Cost function

PnP can be formulated as the following constrained nonlinear least-squares minimization problem:

$$\begin{aligned} \{\alpha_i^*,\ {}^S_G\mathbf{C}^*,\ {}^S\mathbf{p}_G^*\} &= \arg\min J \\ \text{subject to} \quad {}^S_G\mathbf{C}\,{}^S_G\mathbf{C}^T &= \mathbf{I}_3, \quad \det({}^S_G\mathbf{C}) = 1 \\ \alpha_i &= ||{}^S_G\mathbf{C}\,{}^G\mathbf{r}_i + {}^S\mathbf{p}_G|| \end{aligned} \tag{5.3}$$

where the cost function J is the sum of the squared measurement errors, i.e.,

$$J = \sum_{i=1}^{n} ||\mathbf{z}_i - {}^{S}\bar{\mathbf{r}}_i||^2$$
$$= \sum_{i=1}^{n} ||\mathbf{z}_i - \frac{1}{\alpha_i} ({}^{S}_{G}\mathbf{C}\, {}^{G}\mathbf{r}_i + {}^{S}\mathbf{p}_G)\,||^2. \quad (5.4)$$

Unfortunately, J is nonlinear in the unknown quantities, and computing all of its local minima is quite challenging. One approach is to select an initial guess for the parameter vector and employ an iterative minimization technique, such as Gauss-Newton, to numerically compute a single local minimum of J. A clear limitation of this approach is that it can only converge to one of the minima of the cost function, and even with multiple restarts, we are not guaranteed to obtain all minima of J [112]. An alternative approach is to attempt to analytically solve the system of equations provided by the KKT optimality conditions of (5.3) for the unknown quantities. However, this method is also challenging since the KKT conditions form a nonlinear system of equations in $6 + n$ unknowns (3 from ${}^{S}_{G}\mathbf{C}$, 3 from ${}^{G}\mathbf{p}_S$, and n from α_i, $i = 1, \ldots, n$).[1] A third strategy is to relax the original optimization problem [see (5.3)] and manipulate the measurement equations to reduce the number of unknowns. This leads to a modified LS problem for the reduced set of parameters, which can be solved analytically.

In this chapter, we follow the third approach, which is described in Sects. 5.3.3-5.3.5. Before discussing our method in detail, we first provide a brief overview. We satisfy the constraints in the following way: (i) We employ the Cayley-Gibbs-Rodriguez (CGR) parametrization of the rotation matrix ${}^{S}_{G}\mathbf{C}$ and utilize the three CGR parameters as unconstrained optimization variables. In this way we satisfy the rotation matrix constraints, ${}^{S}_{G}\mathbf{C}^{T}\,{}^{S}_{G}\mathbf{C} = \mathbf{I}$ and $\det({}^{S}_{G}\mathbf{C}) = 1$, exactly. (ii) We relax the scale constraint $\alpha_i = ||{}^{S}_{G}\mathbf{C}\, {}^{G}\mathbf{r}_i + {}^{S}\mathbf{p}_G||$, treating each α_i as a free parameter. Note that this relaxation is reasonable since solving the optimality conditions results in $\alpha_i^* = \mathbf{z}_i^T\, ({}^{S}_{G}\mathbf{C}\, {}^{G}\mathbf{r}_i + {}^{S}\mathbf{p}_G)$, which exactly satisfies the constraint when the measurements are noise free (see Appendix F). Subsequently, in order to reduce the number of unknown parameters in the LS cost function, we manipulate the measurement equations, and express ${}^{S}\mathbf{p}_G$ and α_i as

[1] Note that in this case, the KKT conditions can be written as a system of polynomial equations whose degree and number of variables depend linearly on the number of measurements. Given the doubly exponential (in the degree and number of variables) complexity of current methods for solving polynomial systems, this approach is only practical for small-scale problems.

functions of the unknown rotation ${}^{S}_{G}\mathbf{C}$. We then directly solve a modified LS problem to obtain all rotation hypotheses (local minima), from which we recover the scale α_i and translation ${}^{S}\mathbf{p}_G$.

5.3.3 Modified measurement equations

We first consider the noise-free geometric constraints which appear in the measurement model (5.1),

$$\alpha_i {}^{S}\bar{\mathbf{r}}_i = {}^{S}_{G}\mathbf{C}\, {}^{G}\mathbf{r}_i + {}^{S}\mathbf{p}_G,\ i = 1, \ldots, n. \tag{5.5}$$

This system of equations contains unknown quantities (α_i, ${}^{S}_{G}\mathbf{C}$, ${}^{S}\mathbf{p}_G$), and quantities which are either known perfectly (${}^{G}\mathbf{r}_i$), or are measured by the camera (${}^{S}\bar{\mathbf{r}}_i$). We would like to reparametrize this system of equations in terms of fewer unknowns. Since both the scale and translation parameters appear linearly, they are good candidates for reduction. We can rewrite (5.5) in matrix-vector form as

$$\underbrace{\begin{bmatrix} {}^{S}\bar{\mathbf{r}}_1 & & -\mathbf{I} \\ & \ddots & \vdots \\ & {}^{S}\bar{\mathbf{r}}_n & -\mathbf{I} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \\ {}^{S}\mathbf{p}_G \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} {}^{S}_{G}\mathbf{C} & & \\ & \ddots & \\ & & {}^{S}_{G}\mathbf{C} \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} {}^{G}\mathbf{r}_1 \\ \vdots \\ {}^{G}\mathbf{r}_n \end{bmatrix}}_{\mathbf{b}}$$

$$\Leftrightarrow \mathbf{A}\mathbf{x} = \mathbf{W}\mathbf{b} \tag{5.6}$$

where $\mathbf{A}$ and $\mathbf{b}$ comprise quantities that are known or measured, $\mathbf{x}$ is the vector of unknowns which we wish to eliminate from the system of equations, and $\mathbf{W}$ is a block diagonal matrix of the unknown rotational matrix. From (5.6), we can express ${}^{S}\mathbf{p}_G$ and α_i, $i = 1, \ldots, n$ in terms of the other system quantities as

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{W}\mathbf{b} = \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}\mathbf{W}\mathbf{b} \tag{5.7}$$

where we have partitioned $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$ into $\mathbf{U}$ and $\mathbf{V}$ such that the scale parameters are a function of $\mathbf{U}$ and the translation is a function of $\mathbf{V}$. Exploiting the sparse structure of $\mathbf{A}$, $\mathbf{U}$ and $\mathbf{V}$ in (5.7) are computed in closed form (see Appendix F).

We note that both ${}^{S}\mathbf{p}_{G}$ and α_i are *linear* functions of the unknown rotation matrix ${}^{S}_{G}\mathbf{C}$, i.e.,

$$\alpha_i = \mathbf{u}_i^T \mathbf{W}\mathbf{b}, \quad i = 1, \ldots, n \tag{5.8}$$

$${}^{S}\mathbf{p}_{G} = \mathbf{V}\mathbf{W}\mathbf{b}, \tag{5.9}$$

where $\mathbf{u}_i^T$ corresponds to the i-th row of matrix $\mathbf{U}$ [see (5.7)]. Hence, we can rewrite the constraint equations (5.5) as

$$\underbrace{\mathbf{u}_i^T \mathbf{W}\mathbf{b}}_{\alpha_i} {}^{S}\bar{\mathbf{r}}_i = {}^{S}_{G}\mathbf{C}\, {}^{G}\mathbf{r}_i + \underbrace{\mathbf{V}\mathbf{W}\mathbf{b}}_{{}^{S}\mathbf{p}_{G}}, \quad i = 1, \ldots, n. \tag{5.10}$$

At this point, we have reduced the number of unknown parameters from $6 + n$ down to 3. Furthermore, we express the rotation matrix in terms of the CGR parameters $\mathbf{s} = \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix}^T$, where

$${}^{S}_{G}\mathbf{C} = \frac{\bar{\mathbf{C}}}{1 + \mathbf{s}^T\mathbf{s}} \tag{5.11}$$

$$\bar{\mathbf{C}} \triangleq \left((1 - \mathbf{s}^T\mathbf{s})\, \mathbf{I}_3 + 2\lfloor \mathbf{s} \times \rfloor + 2\mathbf{s}\mathbf{s}^T \right), \tag{5.12}$$

where $\mathbf{I}_3$ denotes the 3×3 identity matrix, and $\lfloor \mathbf{s} \times \rfloor$ is the skew-symmetric matrix parametrized by $\mathbf{s}$. Using the CGR parameters will allow us to formulate a LS minimization problem in $\mathbf{s}$ that automatically satisfies the rotation matrix constraints, i.e., ${}^{S}_{G}\mathbf{C}^T {}^{S}_{G}\mathbf{C} = \mathbf{I}$, $\det({}^{S}_{G}\mathbf{C}) = 1$. We can explicitly show the dependence of (5.10) on $\mathbf{s}$, i.e.,

$$\mathbf{u}_i^T \mathbf{W}({}^{S}_{G}\mathbf{C}(\mathbf{s}))\, \mathbf{b}\, {}^{S}\bar{\mathbf{r}}_i = {}^{S}_{G}\mathbf{C}(\mathbf{s})\, {}^{G}\mathbf{r}_i + \mathbf{V}\mathbf{W}({}^{S}_{G}\mathbf{C}(\mathbf{s}))\, \mathbf{b}. \tag{5.13}$$

Note that ${}^{S}_{G}\mathbf{C}(\mathbf{s})$ appears *linearly* in this equation. This allows one further simplification, specifically, we can cancel the denominator $1 + \mathbf{s}^T\mathbf{s}$ from the constraint equation (5.13) [see (5.11)], i.e.,

$$\mathbf{u}_i^T \mathbf{W}\big(\bar{\mathbf{C}}(\mathbf{s})\big)\, \mathbf{b}\, {}^{S}\bar{\mathbf{r}}_i = \bar{\mathbf{C}}(\mathbf{s})\, {}^{G}\mathbf{r}_i + \mathbf{V}\mathbf{W}\big(\bar{\mathbf{C}}(\mathbf{s})\big)\, \mathbf{b}, \tag{5.14}$$

which renders constraints that are quadratic in $\mathbf{s}$.

To summarize, we began with the original geometric constraint relationship between a known point coordinate ${}^{G}\mathbf{r}_i$ and its noise-free observation ${}^{S}\bar{\mathbf{r}}_i$, and reparametrized the

geometric constraint to be only a function of the unknown rotation matrix ${}^{S}_{G}\mathbf{C}$. To do so, we treated the unknown scales $\alpha_i, i = 1, \ldots, n$, as independent variables, relaxing the original problem formulation (5.3). Subsequently, we employed the CGR parameters to express orientation, and as a final step, we canceled the denominator from the CGR rotation matrix. Hence, this approach results in constraints which are quadratic in the elements of $\mathbf{s}$.

5.3.4 Modified cost function

We employ the modified measurement constraint (5.14) to formulate a LS minimization problem for computing the optimal CGR rotation parameters $\mathbf{s}$. Recalling that the measured unit-vector direction towards each point is $\mathbf{z}_i = \bar{\mathbf{r}}_i + \boldsymbol{\eta}_i$, we rewrite the measurement constraints as

$$\mathbf{u}_i^T \mathbf{W}\big(\bar{\mathbf{C}}(\mathbf{s})\big)\,\mathbf{b}\,(\mathbf{z}_i - \boldsymbol{\eta}_i) = \bar{\mathbf{C}}\,(\mathbf{s})\,{}^{G}\mathbf{r}_i + \mathbf{V}\mathbf{W}\big(\bar{\mathbf{C}}\,(\mathbf{s})\big)\,\mathbf{b} \tag{5.15}$$

$$\Rightarrow\ \mathbf{u}_i^T \mathbf{W}\big(\bar{\mathbf{C}}\,(\mathbf{s})\big)\,\mathbf{b}\,\mathbf{z}_i - \bar{\mathbf{C}}\,(\mathbf{s})\ {}^{G}\mathbf{r}_i - \mathbf{V}\mathbf{W}\big(\bar{\mathbf{C}}\,(\mathbf{s})\big)\,\mathbf{b} = \boldsymbol{\eta}_i' \tag{5.16}$$

where $\boldsymbol{\eta}_i'$ is a zero-mean noise term that is a function of $\boldsymbol{\eta}_i$, but whose covariance depends on the system parameters, and both $\mathbf{u}_i$ and $\mathbf{V}$ are evaluated at ${}^{S}\bar{\mathbf{r}}_i = \mathbf{z}_i$.

Based on (5.16), the pose-determination problem can be reformulated as the following *unconstrained* least-squares minimization problem

$$\{s_1^*, s_2^*, s_3^*\} = \arg\min J' \tag{5.17}$$

where the cost function J' is the sum of the squared constraint errors from (5.16), i.e.,

$$\begin{aligned} J' &= \sum_{i=1}^{n} ||\mathbf{u}_i^T \mathbf{W}\big(\bar{\mathbf{C}}\,(\mathbf{s})\big)\,\mathbf{b}\,\mathbf{z}_i - \bar{\mathbf{C}}(\mathbf{s})\,{}^{G}\mathbf{r}_i - \mathbf{V}\mathbf{W}\big(\bar{\mathbf{C}}\,(\mathbf{s})\big)\,\mathbf{b}||^2 \\ &= \sum_{i=1}^{n} \boldsymbol{\eta}_i'^T \boldsymbol{\eta}_i'. \end{aligned} \tag{5.18}$$

Note that each summand in J' is quartic in the elements of $\mathbf{s}$, and J' contains all monomials up to degree four, i.e., $\{1, s_1, s_2, s_3, s_1 s_2, s_1 s_3, s_2 s_3, \ldots, s_1^4, s_2^4, s_3^4\}$.

Since J' is a fourth-order polynomial, the corresponding optimality conditions form a system of three third-order polynomials. What we show next, is how to employ the Macaulay matrix to directly compute all of the critical points of J' by finding the roots of the polynomial system. A key benefit of our proposed approach is that the polynomial system we solve is of a constant degree, independent of the number of points in the PnP problem. Changing the number of points only affects the coefficients appearing in the system. Thus, we need only compute the Macaulay matrix symbolically once. Subsequently, we simply form the elements of the Macaulay matrix from the data (an operation which is linear in the number of points), and directly find the roots via the eigen decomposition of the Schur complement of the Macaulay matrix (see Sect. 5.3.5).[2]

5.3.5 Directly computing the local minima

What follows next is a brief overview of how we employ the Macaulay matrix [113, 114] to directly determine the roots of a system of polynomial equations. We refer the interested reader to "Using Algebraic Geometry" by Cox *et al.* [48] for a more complete perspective.

Since J' is a fourth-order polynomial function in three unknowns, the corresponding optimality conditions form a system of polynomial equations, i.e.,

$$\nabla_{s_i} J' = F_i = 0, \quad i = 1, 2, 3. \tag{5.19}$$

Each F_i is a polynomial of degree three in the variables s_1, s_2, s_3. The Bézout bound (i.e., the maximum number of possible solutions) for this system of equations is 27. Under mild conditions [48], which are met for general PnP instantiations, the Bézout bound is reached.

Our goal is to compute the *multiplication matrix* from which we can directly obtain all the solutions to our system via eigen decomposition [115]. We obtain the multiplication matrix by first constructing the Macaulay resultant matrix. To do so, we augment our polynomial system with an additional linear equation, which is generally non-zero

[2] We compute the Schur complement of a sparse 120×120 matrix, followed by the eigen decomposition of a non-sparse 27×27 matrix. The total time to complete both operations in Matlab is approximately 15 ms.

at the roots of our system, i.e., $F_0 = u_0 + u_1 s_1 + u_2 s_2 + u_3 s_3$, where each u_j, $j = 0, \ldots, 3$ is randomly generated. We denote the set of all monomials up to degree 7 as

$$S = \{\mathbf{s}^{\boldsymbol{\gamma}} : \textstyle\sum_j \gamma_j \leq 7\} \tag{5.20}$$

where we use the notation $\mathbf{s}^{\boldsymbol{\gamma}} \triangleq s_1^{\gamma_1} s_2^{\gamma_2} s_3^{\gamma_3}$, $\gamma_i \in \mathbb{Z}_{\geq 0}$, to denote a specific monomial. The set S is important, since, using S we can expand our original system of polynomials to obtain a square system that has the same number of equations as monomials. To do so, we first partition S into four subsets, such that S_3 contains all monomials that can be divided by s_3^3, S_2 contains all monomials that can be divided by s_2^3 but not s_3^3, S_1 contains all monomials that can be divided by s_1^3 but not by s_2^3 or s_3^3, and S_0 contains the remaining monomials, i.e.,

$$\begin{aligned} S_0 = \{&1, s_1, s_1^2, s_2, s_1 s_2, s_1^2 s_2, s_2^2, s_1 s_2^2, s_1^2 s_2^2, s_3, s_1 s_3, s_1^2 s_3, \\ &s_2 s_3, s_1 s_2 s_3, s_1^2 s_2 s_3, s_2^2 s_3, s_1 s_2^2 s_3, s_1^2 s_2^2 s_3, s_3^2, s_1 s_3^2, \\ &s_1^2 s_3^2, s_2 s_3^2, s_1 s_2 s_3^2, s_1^2 s_2 s_3^2, s_2^2 s_3^2, s_1 s_2^2 s_3^2, s_1^2 s_2^2 s_3^2\}. \end{aligned}$$

Note that the second, fourth, and tenth elements of S_0 are the three CGR rotation parameters $\{s_1, s_2, s_3\}$; a fact that we will exploit later.

We next form an extended system of equations by multiplying F_0 with each of the monomials in S_0, and multiplying F_i with each of the monomials in S_i divided by s_i^3, $i = 1, 2, 3$. We denote polynomials obtained from extending F_i as $G_{i,j}$, $j = 1, \ldots, |S_i|$. Thus, the extended set of polynomial equations is

$$\begin{bmatrix} G_{0,1} \\ G_{0,2} \\ \vdots \\ G_{1,1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{c}_{0,1}^T \\ \mathbf{c}_{0,2}^T \\ \vdots \\ \mathbf{c}_{1,1}^T \\ \vdots \end{bmatrix} \underline{\mathrm{s}}^{\boldsymbol{\gamma}} = \mathbf{M}\underline{\mathrm{s}}^{\boldsymbol{\gamma}} = \mathbf{M} \begin{bmatrix} \underline{\mathrm{s}}^{\boldsymbol{\alpha}} \\ \underline{\mathrm{s}}^{\boldsymbol{\beta}} \end{bmatrix} \tag{5.21}$$

where each polynomial $G_{i,j}$ is expressed as an inner product between the coefficient vector, $\mathbf{c}_{i,j}^T$, and the vector of all monomials $\underline{\mathrm{s}}^{\boldsymbol{\gamma}}$, i.e., $G_{i,j} = \mathbf{c}_{i,j}^T \underline{\mathrm{s}}^{\boldsymbol{\gamma}}$. The Macaulay matrix

$\mathbf{M}$ is formed by stacking the coefficient vectors. Finally, we partition $\underline{\mathbf{s}}^{\boldsymbol{\gamma}}$ such that $\underline{\mathbf{s}}^{\boldsymbol{\alpha}}$ comprises monomials in S_0, and $\underline{\mathbf{s}}^{\boldsymbol{\beta}}$ contains the remaining monomials.

If we evaluate (5.21) at a root, $\mathbf{p} = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^T$, of the original system (5.19), then all polynomials $G_{i,j}$ extended from F_i, $i = 1, 2, 3$ will be zero, since $F_i(\mathbf{p}) = 0$ by definition. However, F_0 and hence $G_{0,j}$, $j = 1, \ldots, |S_0|$ will not generally be zero, i.e.,

$$\begin{bmatrix} G_{0,1}(\mathbf{p}) \\ \vdots \\ G_{0,|S_0|}(\mathbf{p}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \Leftrightarrow \begin{bmatrix} F_0(\mathbf{p})\underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \mathbf{0} \end{bmatrix} = \mathbf{M} \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \tag{5.22}$$

where $\underline{\mathbf{p}}^{\boldsymbol{\alpha}}$ and $\underline{\mathbf{p}}^{\boldsymbol{\beta}}$ denote the monomial vectors evaluated at $\mathbf{p}$, i.e., $\underline{\mathbf{s}}^{\boldsymbol{\alpha}}(\mathbf{p}) = \underline{\mathbf{p}}^{\boldsymbol{\alpha}}$ and $\underline{\mathbf{s}}^{\boldsymbol{\beta}}(\mathbf{p}) = \underline{\mathbf{p}}^{\boldsymbol{\beta}}$. Based on this observation, we partition $\mathbf{M}$ into four blocks where $\mathbf{M}_{00}$ is of dimension $|S_0| \times |S_0|$, and rewrite (5.22) as

$$\begin{bmatrix} F_0(\mathbf{p})\underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{00} & \mathbf{M}_{01} \\ \mathbf{M}_{10} & \mathbf{M}_{11} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix}. \tag{5.23}$$

Finally, exploiting the Schur complement, we obtain

$$F_0(\mathbf{p})\underline{\mathbf{p}}^{\boldsymbol{\alpha}} = \mathcal{M}_{F_0}\underline{\mathbf{p}}^{\boldsymbol{\alpha}} \tag{5.24}$$

where $\mathcal{M}_{F_0} = \mathbf{M}_{00} - \mathbf{M}_{01}\mathbf{M}_{11}^{-1}\mathbf{M}_{10}$ is the multiplication matrix corresponding to F_0. From (5.24) we see that $F_0(\mathbf{p})$ is an eigenvalue of $\mathcal{M}_{F_0}$ with corresponding eigenvector $\underline{\mathbf{p}}^{\boldsymbol{\alpha}}$. We can directly obtain all 27 solutions to our system of equations (5.19) via eigen decomposition, since the eigenvectors of $\mathcal{M}_{F_0}$ are the monomials of S_0 evaluated at each of the 27 roots. Since the first element in S_0 is 1, we normalize each eigenvector by its first element, and read off the solution for s_i, $i = 1, 2, 3$, from the second, fourth, and tenth elements of the eigenvector.

Through this procedure we obtain 27 critical points, which include real and imaginary minima, maxima, and saddle points of the cost function (5.18). In practice, we have only observed up to 4 real local minima that place the points in front of the center

of perspectivity. In almost all cases, when $n \geq 6$ we obtain a single real minimum of the function. After obtaining the minima, we evaluate the cost function to find the optimal orientation, and compute the corresponding translation from (5.9). Additional details about the DLS PnP algorithm implementation are available as supplemental material [116].

5.4 Simulation and Experimental Results

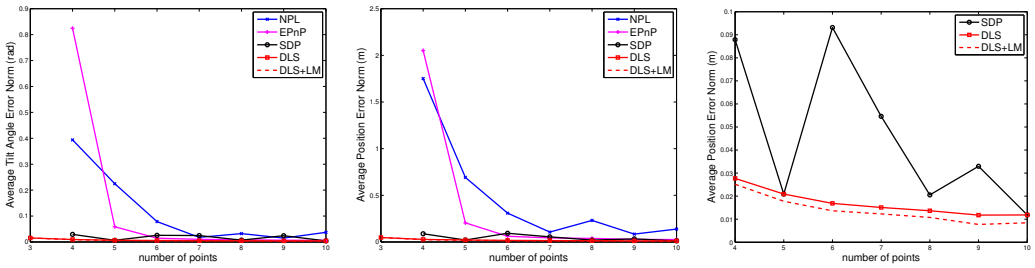
5.4.1 Simulations

We hereafter present simulation results which compare the accuracy of our method to the leading PnP approaches:

- **NPL:** The N-Point Linear (NPL) method of Ansar and Daniilidis [99].
- **EPnP:** The approach of Lepitit et al. [103].
- **SDP:** The Semi Definite Program (SDP) approach of Schweighofer and Pinz [102].
- **DLS:** The Direct Least-Squares (DLS) solution presented in this paper. An open source implementation of DLS is available at `www.umn.edu/~joel`
- **DLS-LM:** Maximum-likelihood estimate, computed using iterative Levenberg-Marquardt (LM) minimization of the sum of the squared reprojection errors, initialized with DLS.

To test the NPL, EPnP, and SDP methods, we obtained the authors' own Matlab implementations, which were either provided via e-mail request or publicly available on the web.

We first examine the performance of the above algorithms versus number of points. We randomly distribute points within the field of view (45×45 deg) of an internally calibrated camera (focal length 600 px), at distances between 0.5 and 5.5 m. We perturb each image measurement (point projection on the image plane) by independent zero-mean Gaussian noise ($\sigma = 1.5$ px along both u and v axes). We vary the number of points from 3 to 10, noting that for the methods which require a unique solution to



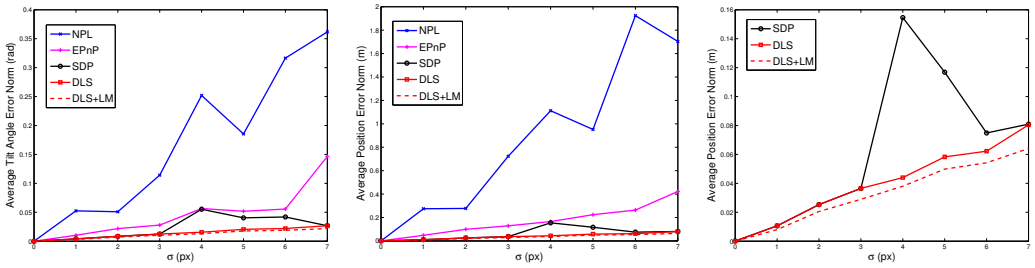
(a) (b) (c)

Figure 5.2: Accuracy comparison depicted as the average error norm, over 100 trials for each number of points, for orientation 5.2(a) and position 5.2(b). The results for just SDP, DLS, and DLS+LM are depicted in 5.2(c).

work (i.e., NPL, EPnP, and SDP), we only show results for 4 or more points (when a unique solution is probable).

Figure 5.2 shows the results comparing the five approaches based on their average error norm computed over 100 trials. We compute the position error norm as $||{}^{S}\mathbf{p}_{G,true} - {}^{S}\mathbf{p}_{G,est}||$, while we compute the tilt-angle (orientation) error norm as $||\boldsymbol{\delta\theta}|| = 2||\tilde{\mathbf{s}}||$, where $\tilde{\mathbf{s}}$ is the CGR parameter obtained from ${}^{S}_{G}\tilde{\mathbf{C}} = {}^{S}_{G}\mathbf{C}^{T}_{true}\,{}^{S}_{G}\mathbf{C}_{est}$. We see that DLS performs consistently better than other approaches, and obtains results close to the MLE estimate (DLS-LM). The SDP method treats strictly planar scenes differently than non-planar scenes [102], by using two different SDP relaxations. However in some cases, when the points are close to a coplanar configuration, neither SDP approach provides accurate results [e.g., $n = 6$ in Fig. 5.2(c), the average error is larger due to a few nearly coplanar cases out of the 100 trials]. We also note that NPL is least accurate since it sometimes returns imaginary solutions (due to recovery of the original parameters after lifting). In these instances, we compute a real solution by projecting the imaginary solution back onto the real axis.

We also examine the performance of the five approaches as a function of the pixel noise. We vary the pixel noise standard deviation between $\sigma = 0$ px and $\sigma = 7$ px, noting that we only permit noise between $\pm 3\sigma$ (to prevent outliers). Figure 5.3 displays the results of the average error norm over 100 trials for position and orientation. We note that DLS again outperforms the existing methods and is very close to the MLE



(a) (b) (c)

Figure 5.3: Accuracy comparison depicted as the average error norm, over 100 trials for each value of σ, for orientation 5.4(a) and position 5.4(b). The results for just SDP, DLS, and DLS+LM are depicted in 5.3(c).

estimate (DLS-LM).

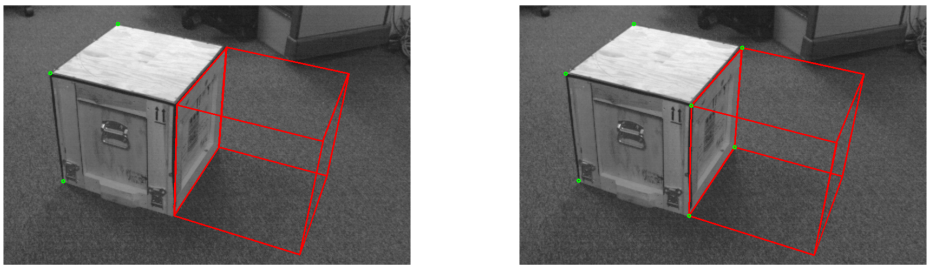
5.4.2 Experiments

We evaluated our method experimentally with observations of 7 known points at the corners of a cube. We computed the camera pose with each method (using 3, 4, and 7 known points), and compared the resulting pose value to the MLE estimate obtained using all 7 points. Table 5.1 lists the errors for orientation and position for each method.

Figure 5.4 depicts the visual results of the experiment. We show the back-projection of the known global points on the image as green circles, for DLS3 [Fig. 5.4(a)], and DLS7 [Fig. 5.4(b)]. In order to further validate the results visually, we also back-project a virtual box (of identical dimensions as the real box) next to the real box. Additional trials are included in the supplemental material [116].

5.4.3 Processing time comparison

The speed of the four direct methods was evaluated in Matlab 7.8 running on a Linux (kernel 2.6.32) computer with a 2.4 GHz Intel Core 2 Duo processor. NPL and EPnP were the fastest algorithms, requiring approximately 10 ms and 5 ms, respectively, to solve a four-point problem. Our algorithm required approximately 15 ms to compute all local minima of the LS cost function using the Macaulay resultants method. The



(a) (b)

Figure 5.4: The solution computed using DLS with 3 known points is depicted in 5.4(a), where the green circles represent the 3 known points back-projected onto the image using the computed transformation. 5.4(b) is the result obtained using DLS with 7 known points. In both cases, we also back-project a virtual cube, placed next to the real one, to aid visual verification of the result.

slowest approach was SDP which required approximately 200 ms to solve the semi-definite program (using SeDuMi). Since the implementations are Matlab-based and not optimized for speed, we provide these only as "ball-park" figures for performance. Part of our ongoing work is to compare the run-time of these methods using efficient C/C++ implementations.

5.5 Summary

In this chapter, we have presented a DLS method for PnP which has several advantages compared to existing approaches. First, it is flexible in that it can handle any number of points from the minimal case of 3, to the general case of $n \geq 4$. It computes all pose solutions analytically, as the minima of a nonlinear least-squares cost function, without the need for initialization. Instead, using a reformulation of the geometric constraints, we obtain LS optimality conditions that form a system of three third-order polynomials, which are solved efficiently using the multiplication matrix.

We have validated the proposed method alongside three leading PnP algorithms as

well as the MLE, both in simulation and experimentally. Compared to existing approaches, DLS is consistently more accurate, attaining performance close to the MLE. DLS is also efficient, since the order of the polynomial system that it solves is independent of the number of measurements. Lastly, in contrast to other techniques which seek to obtain a single global optimum (e.g., SDP and EPnP) DLS has the unique characteristic that it analytically computes all minima of the LS cost function.

n-points	Ori. Error Norm (rad)	Pos. Error Norm (m)
NPL4	2.87×10^{-3}	8.67×10^{-3}
NPL7	2.12×10^{-3}	2.42×10^{-3}
EPnP4	2.49×10^{-2}	2.33×10^{-2}
EPnP7	1.24×10^{-2}	3.41×10^{-3}
SDP4	4.26×10^{-3}	9.82×10^{-3}
SDP7	3.86×10^{-4}	3.49×10^{-4}
DLS3	5.41×10^{-3}	1.02×10^{-2}
DLS4	4.28×10^{-3}	9.83×10^{-3}
DLS7	4.29×10^{-4}	3.35×10^{-4}

Table 5.1: The orientation and position errors for different numbers of points. Errors are computed with respect to the MLE estimate of the camera pose computed using all 7 points.

Chapter 6

Concluding Remarks

6.1 Summary of contributions

This thesis focuses on analyzing and improving the performance of six d.o.f. localization using onboard sensors. In particular, we evaluate the interplay of system observability properties and estimator consistency for laser-aided and vision-aided inertial navigation systems (LINS and VINS, respectively). The following is a summary of the key contributions:

- **An approach for Laser-aided Inertial Navigation**
  In Chapter 2, we presented a novel LINS, based on a 2D laser scanner and an IMU, capable of 3D localization and mapping in indoor environments. In the proposed method, the orthogonal structural planes of the building are employed as landmarks to aid in localization. Since the building's layout may be partially or completely unknown, the planes' parameters are estimated concurrently with the six d.o.f. pose of the person. To this end, an EKF is utilized to fuse information from an IMU and a 2D laser scanner, and estimate the person's motion and the building's structural planes. We presented a practical method for filter initialization that employs line-to-plane correspondences to initialize the orientation and zero-velocity updates to initialize the IMU bias estimates. Furthermore, we studied the observability properties of the system to determine a sufficient condition on the number and type of measurements so as to ensure that the pose can be estimated. As a final contribution of this chapter, we introduced an on-line extrinsic

calibration approach for estimating the laser-to-IMU transformation. The validity of the proposed method is demonstrated experimentally in both previously known and unknown environments, which include challenging 3D building structures such as staircases, a disability access ramp, and long corridors. Furthermore, the environments contained a typical amount of office clutter (e.g., chairs and desks) as well as pedestrian traffic.

- **Consistency analysis and improvement for VINS**
  In Chapter 3, we analyzed the inconsistency of VINS from the standpoint of observability. Specifically, we showed that standard EKF-based filtering approaches lead to spurious information gain since they do not adhere to the unobservable directions of the nonlinear system. Furthermore, we introduced an observability-constrained VINS approach to mitigate estimator inconsistency by enforcing the nullspace explicitly. We presented extensive simulation and experimental results to support our claims and validated the proposed estimator, by applying it to both V-SLAM and the MSC-KF.

- **Towards consistent single-camera localization**
  In Chapter 4, we analyzed the inconsistency of MonoSLAM from the standpoint of observability. Specifically, we showed that using a standard EKF-based approach leads to spurious information gain, in particular for scale, since it does not adhere to the unobservable directions of the true system. Moreover, we introduced an observability-constrained MonoSLAM method to mitigate estimator inconsistency by enforcing the nullspace explicitly. Finally, we presented simulation and experimental results to support our claims and validate the proposed estimator.

- **A direct-least squares method for PnP**
  In Chapter 5, we presented a direct least-squares (DLS) method for PnP which has several advantages compared to existing approaches. Firstly, it is flexible in that it can handle any number of points from the minimal case of 3, to the general case of $n \geq 4$. Additionally, it computes all pose solutions analytically, as the minima of a nonlinear least-squares cost function, without the need for initialization. Specifically, using a reformulation of the geometric constraints, we obtain LS optimality conditions that form a system of three third-order polynomial

equations, which are solved efficiently using the multiplication matrix.

We have validated the proposed method alongside three leading PnP algorithms as well as the MLE, both in simulation and experimentally. Compared to existing approaches, DLS is consistently more accurate, attaining performance close to the MLE. DLS is also efficient, since the order of the polynomial system that it solves is independent of the number of measurements. Lastly, in contrast to other techniques which seek to obtain a single global optimum (e.g., SDP and EPnP) DLS has the unique characteristic that it analytically computes all minima of the LS cost function.

6.2 Future Work

The accuracy of VINS is significantly reduced in areas containing a small number of visual cues, such as when traversing long corridors with mono-color walls or sparse, clutter-free office environments. In these instances, an estimator may become increasingly reliant on inertial sensing which can lead to large estimation errors, hence invalidating the small linearization-error assumption. To mitigate this issue, we believe it is pertinent to investigate methods for exploiting information provided by stochastic and deterministic motion constraints to reduce the drift effects. For example, under normal walking conditions, a person's heading direction lies along their dorsoventral (back-to-front) axis, while their velocity is constrained by their limb length and step frequency. This information (specific to each person), can be expressed as a state constraint and used to improve estimation accuracy (e.g., by restricting the direction of travel as well as the magnitude of linear and rotational velocities).

In many applications, we have prior knowledge about how the vehicle or person moves in space. Examples include the maximum velocity or turning radius of a car specified by engineering design, or, in the case of a human, average walking speed. Through careful system modeling, a more complex understanding of the vehicle dynamics can also be developed, including masses, moments of inertia, contact forces, and material interaction properties. Exploiting this additional information has the potential to improve the accuracy and consistency of the estimation process, particularly in areas in which little visual information is available for constraining the IMU drift.

To put this idea in a historical context, one of the first Kalman filters for estimating 3D attitude, particularly for early satellites such as Nimbus I (NASA circa 1964), were designed around complex dynamical models of motion (used for propagation), while all sensor measurements (including IMU data) were used for state updates [8]. Despite the significant time and effort that was invested in generating an accurate dynamical model, the resulting pose accuracy was insufficient. Additionally, the complexity of the state-space in terms of number of parameters became prohibitively large for the computational resources available at that time. For these reasons, most modern inertial navigation systems rely on kinematics-based motion models that integrate the IMU signals in order to propagate the state estimates. This paradigm shift was ideal in many ways; first and foremost it provided higher accuracy 3D pose estimation, but more importantly, it permitted porting the navigation system from vehicle to vehicle without the need to redesign the system model.

We believe, however, that performance can be improved if we reintroduce the model-based motion constraints in the context of VINS, as an additional information source that can improve estimation accuracy, particularly when few visual features are available. The difference of what we are proposing compared to past approaches, is that instead of employing motion models at the core of the estimation process (e.g., in the EKF propagation phase), we should exploit this information as additional "measurements" which can be applied flexibly, whenever the corresponding model or constraint is valid. This represents a key novelty of our future work and a necessary caution, since during certain periods a motion constraint may be valid, while not at other times (e.g., a human may exhibit nominal walking behavior while on level ground inside a building, but when struggling to walk up a sand dune, their stride length and frequency, as well as foot-to-ground interaction and slippage will differ significantly).

6.2.1 Example sources of additional motion information

Tracking models: The first and most general type of motion information that we can exploit are statistical tracking models which characterize how a vehicle or person moves under nominal conditions [7, 117]. Common models include zero-velocity and zero-acceleration, which assume that velocity or acceleration change only as a random walk. For example, a simple discrete-time zero-acceleration tracking model can be expressed

as:

$$\begin{bmatrix} \mathbf{p}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \delta t \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{w}_k \end{bmatrix} \tag{6.1}$$

where $\mathbf{p}$ and $\mathbf{v}$ denote position and velocity, respectively, and the noise driving the system, $\mathbf{w}_k$, is distributed as a zero-mean Gaussian random vector with covariance $\mathbf{Q}_k$. The statistics of $\mathbf{w}_k$ can be inferred based on prior knowledge of vehicle capabilities and mission, or can be learned from training data.

Gait models: A second type of motion information, particularly focused on human navigation, are gait models which describe the distance traveled during each gait cycle (e.g., walking step). For instance, during one walking step a human might move d meters, which can be expressed as the following measurement constraint

$$||\mathbf{p}_{k+\ell} - \mathbf{p}_k|| = d + \eta \tag{6.2}$$

where $t_{k+\ell} - t_k$ is the duration of a single walking step, and η is a noise term capturing the uncertainty in the model. A naïve approach would be to consider both ℓ and d constant for everyone (i.e., all humans walk at the same pace with the same step size). This, however, has obvious limitations given the difference in body geometry (e.g., height and limb length) and walking styles among people. Furthermore, a single person may walk at different gait frequencies depending on if they are on a leisurely walk in the park, or if they are running late for a meeting. For these reasons, recent work (e.g., [118, 119]) has focused on answering the following questions: (i) What is the person's current gait (e.g., walking, running, or crawling)? (ii) How fast is the person executing the gait? (iii) What is the functional relationship between the distance traveled in one step, the frequency of motion, and the person's body geometry?

Other motion information: Additional motion information can also be inferred from the mechanical design of a vehicle (e.g., Ackerman or skid steering model, wheelbase length), as well as knowledge about the environment. For instance, in most practical circumstances, ground vehicles and people both traverse support planes while they move; hence motion in the vertical direction is kept to a minimum. These additional sources of information can also provide constraints on the system's motion which can be exploited to improve VINS performance.

6.2.2 Exploiting motion information as state constraints

We propose to incorporate additional sources of motion information within the MAP (or sliding-window filtering) estimation framework. Specifically, constraints in the form of (6.1) and (6.2) can be written as

$$||\mathbf{g}\left(\mathbf{x}_k,\, \ldots,\, \mathbf{x}_{k+\ell},\, \boldsymbol{\zeta}\right)||^2_{\mathbf{D}} \tag{6.3}$$

where the, in general nonlinear, constraint function $\mathbf{g}$ involves the state of the system over time-steps k through $k+\ell$, and the vector $\boldsymbol{\zeta}$ comprises parameters involved in the constraint relationship (e.g., vehicle wheel base, or the person's biometric information). The weighting matrix $\mathbf{D}$ corresponds to the uncertainty of the motion information and can be learned from training data or selected based on knowledge of the system or motion. The motion-model constraints are independent of the other sensor measurements, and will appear as additional cost terms in the estimator's cost function.

For each state constraint that we investigate, we will answer the following questions: (i) Which states does it directly impact? (ii) How does it affect the system observability properties? (iii) What conditions must be satisfied in order to employ this constraint? (iv) What is a realistic uncertainty model, and how to perform uncertainty characterization? (v) What are its practical benefits and limitations (i.e., how does it work in real-world conditions)?

References

[1] Sendero Group. BrailleNote: Global positioning system. [Online]. Available: http://www.humanware.com/. [Accessed: January 5, 2011].

[2] National Aeronautics United States of America and Space Administration. Mars exploration rover. [Online]. Available: http://marsrovers.jpl.nasa.gov/. [Accessed: July 18, 2012].

[3] Honeywell Aerospace. T-hawk: Micro aerial vehicle. [Online]. Available: http://www.thawkmav.com/. [Accessed: July 18, 2012].

[4] V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran. RFID in robot-assisted indoor navigation for the visually impaired. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1979–1984, Sendai, Japan, September 28–October 2, 2004.

[5] B.S. Tjan, P.J. Beckmann, N. Giudice, and G.E. Legge. Digital sign system for indoor wayfinding for the visually impaired. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, page 30, San Diego, CA, June 20–25, 2005.

[6] Scott Porter. Case study 25: RFID pilot solution for the miami museum of science & planetarium. [Online]. Available: http://museummedia.nl/case-studies/case-study-25-rfid-pilot-solution-for-the-miami-museum-of-science-planetarium/. [Accessed: July 18, 2012].

[7] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, NY, 2001.

[8] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume I. Academic Press, New York, NY, 1979.

[9] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. A first-estimates Jacobian EKF for improving SLAM consistency. In *Proc. of the Int. Symposium on Experimental Robotics*, pages 373–382, Athens, Greece, July 14–17, 2008.

[10] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Observability-based rules for designing consistent EKF SLAM estimators. *Int. Journal of Robotics Research*, 29(5):502–528, April 2010.

[11] Guoquan P. Huang, Nikolas Trawny, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Observability-based consistent EKF estimators for multi-robot cooperative localization. *Autonomous Robots*, 30(1):99–122, January 2011.

[12] Robert M. Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *Int. Journal of Computer Vision*, 13(3):331–356, December 1994.

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[14] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883, pages 298–372. Springer-Verlag, 2000.

[15] Eric A. Wan and Ronell Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proc. of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, Lake Louise, Alberta, Canada, October 1–4, 2000.

[16] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[17] Humanware. Trekker talking GPS, 2010. [Online]. Available: www.humanware.com. [Accessed: February 2, 2010].

[18] S. Thrun, D. Fox, and W. Burgard. Monte Carlo localization with mixture proposal distribution. In *Proc. of the AAAI National Conf. on Artificial Intelligence*, pages 859–865, Austin, TX, July 30–August 3, 2000.

[19] L. Iocchi and S. Pellegrini. Building 3D maps with semantic elements integrating 2D laser, stereo vision and IMU on a mobile robot. In *Proc. of the ISPRS Int. Workshop on Virtual Reconstruction and Visualization of Complex Architectures*, Zurich, Switzerland, July 12–13, 2007.

[20] Dirk Hähnel, Wolfram Burgard, and Sebastian Thrun. Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 44(1):15–27, July 2003.

[21] Dorit Borrmann, Jan Elseberg, Kai Lingemann, Andreas Nüchter, and Joachim Hertzberg. Globally consistent 3D mapping with scan matching. *Robotics and Autonomous Systems*, 56(2):130–142, January 2008.

[22] Jonghyuk Kim and Salah Sukkarieh. Real-time implementation of airborne inertial-SLAM. *Robotics and Autonomous Systems*, 55(1):62–71, January 2007.

[23] A. I. Mourikis and S. I. Roumeliotis. A dual-layer estimator architecture for long-term localization. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 1–8, Anchorage, AK, June 2008.

[24] Joel A. Hesch, Faraz M. Mirzaei, Gian Luca Mariottini, and Stergios I. Roumeliotis. A 3D pose estimator for the visually impaired. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2716–2723, St. Louis, MO, October 11–15, 2009.

[25] Joel A. Hesch, Faraz M. Mirzaei, Gian Luca Mariottini, and Stergios I. Roumeliotis. A Laser-aided Inertial Navigation System (L-INS) for human localization in

unknown indoor environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 5376–5382, Anchorage, AK, May 3–8, 2010.

[26] Viet Nguyen, Ahad Harati, Agostino Martinelli, Roland Siegwart, and Nicola Tomatis. Orthogonal SLAM: a step toward lightweight indoor autonomous navigation. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 5007–5012, Beijing, China, October 9–15, 2006.

[27] Iwan Ulrich and Johann Borenstein. The GuideCane - applying mobile robot technologies to assist the visually impaired. *IEEE Trans. on Systems, Man, and Cybernetics, -Part A: Systems and Humans*, 31(2):131–136, March 2001.

[28] Dan Yuan and Roberto Manduchi. Dynamic environment exploration using a virtual white cane. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 243–249, San Diego, CA, June 20–25, 2005.

[29] Andreas Hub, Joachim Diepstraten, and Thomas Ertl. Design and development of an indoor navigation and object identification system for the blind. In *Proc. of the Int. ACM SIGACCESS Conf. on Computers and Accessibility*, pages 147–152, Atlanta, GA, October 18–20, 2004.

[30] H. Makino, I. Ishii, and M. Nakashizuka. Development of navigation system for the blind using GPS and mobile phone combination. In *Proc. of the Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, pages 506–507, Amsterdam, Netherlands, October 31–November 3, 1996.

[31] L. Ran, S. Helal, and S. Moore. Drishti: an integrated indoor/outdoor blind navigation system and service. In *Proc. of the IEEE Conf. on Pervasive Computing and Communications*, pages 23–30, Orlando, FL, March 14–17, 2004.

[32] Filippo Cavallo, Angelo Sabatini, and Vincenzo Genovese. A step toward GPS/INS personal navigation systems: real-time assessment of gait by foot inertial sensing. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1187–1191, Edmonton, Canada, August 2–6, 2005.

[33] Koichi Sagawa, Hikaru Inooka, and Yutaka Satoh. Non-restricted measurement of walking distance. In *Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 1847–1852, Nashville, TN, October 8–11, 2000.

[34] Johann Borenstein, Lauro Ojeda, and Surat Kwanmuang. Heuristic reduction of gyro drift for personnel tracking systems. *Journal of Navigation*, 62(1):41–58, January 2009.

[35] S. Ertan, C. Lee, A. Willets, H. Tan, and A. Pentland. A wearable haptic navigation guidance system. In *Proc. of the Int. Sym. on Wearable Computers*, pages 164–165, Pittsburgh, PA, October 19–20, 1998.

[36] Joel A. Hesch and Stergios I. Roumeliotis. Design and analysis of a portable indoor localization aid for the visually impaired. *Int. Journal of Robotics Research*, 29(11):1400–1415, September 2010.

[37] R.C. Smith, M. Self, and P. Cheeseman. *Autonomous Robot Vehicles*, chapter Estimating Uncertain Spatial Relationships in Robotics, pages 167–193. Springer-Verlag, New York, NY, 1990.

[38] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the Simultaneous Localization and Map building (SLAM) problem. *IEEE Trans. on Robotics and Automation*, 17(3):229–241, June 2001.

[39] Peter Kohlhepp, Paola Pozzo, Marcus Walther, and Rüdiger Dillman. Sequential 3D-SLAM for mobile action planning. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 722–729, Sendai, Japan, September 28–October 2, 2004.

[40] Andreas Nüchter, Harmut Surmann, Kai Lingemann, Joachim Hertzberg, and Sebastian Thrun. 6D SLAM with an application in autonomous mine mapping. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1998–2003, New Orleans, LA, April 18–May 1, 2004.

[41] David M. Cole and Paul M. Newman. Using laser range data for 3D SLAM in outdoor environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1556–1563, Orlando, FL, May 15–19, 2006.

[42] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, November 2004.

[43] E. J. Lefferts, F. L. Markley, and M. D. Shuster. Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429, September – October 1982.

[44] Nikolas Trawny and Stergios I. Roumeliotis. Indirect Kalman filter for 3D attitude estimation. Technical Report 2005-002, University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, March 2005.

[45] Viet Nguyen, Stefan Gächter, Agostino Martinelli, Nicola Tomatis, and Roland Siegwart. A Comparison of Line Extraction Algorithms using 2D Range Data for Indoor Mobile Robotics. *Autonomous Robots*, 23(2):97–111, August 2007.

[46] Sam T. Pfister, Stergios I. Roumeliotis, and Joel W. Burdick. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1304–1311, Taipei, Taiwan, September 14–19, 2003.

[47] Homer H. Chen. Pose determination from line-to-plane correspondences: existence condition and closed-form solutions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6):530–541, June 1991.

[48] David A. Cox, John B. Little, and Don O'Shea. *Using Algebraic Geometry*. Springer-Verlag, New York, NY, 2nd edition, 2005.

[49] A. I. Mourikis, S. I. Roumeliotis, and J. W. Burdick. SC-KF mobile robot localization: A Stochastic Cloning-Kalman Filter for processing relative-state measurements. *IEEE Trans. on Robotics*, 23(4):717–730, August 2007.

[50] Robert Hermann and Arthur Krener. Nonlinear controlability and observability. *IEEE Trans. on Automatic Control*, 22(5):728–740, October 1977.

[51] Faraz M. Mirzaei and Stergios I. Roumeliotis. IMU-laser scanner localization: Observability analysis. Technical report, Dept. of Computer Science & Engineering, University of Minnesota, MARS Lab, Minneapolis, MN, January 2009.

[52] Jan Skaloud and Derek Lichti. Rigorous approach to bore-sight self-calibration in airborne laser scanning. *ISPRS Journal of Photogrammetry & Remote Sensing*, 61(1):47–59, October 2006.

[53] P. Rieger, N. Studnicka, M. Pfennigbauer, and G. Zach. Boresight alignment method for mobile laser scanning systems. *Journal of Applied Geodesy*, 4(1):13–21, 2010.

[54] Faraz M. Mirzaei and Stergois I. Roumeliotis. A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Trans. on Robotics*, 24(5):1143–1156, October 2008.

[55] Shaojie Shen, Nathan Michael, and Vijay Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 20–25, Shanghai, China, May 9–13, 2011.

[56] Mitch Bryson and Salah Sukkarieh. Observability analysis and active control for airborne SLAM. *IEEE Trans. on Aerospace and Electronic Systems*, 44(1):261–280, January 2008.

[57] Sedat Ebcin and Mike Veth. Tightly-coupled image-aided inertial navigation using the unscented Kalman filter. Technical report, Air Force Institute of Technology, Dayton, OH, 2007.

[58] Dennis W. Strelow. *Motion estimation from image and inertial measurements*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, November 2004.

[59] Jason Durrie, Tristan Gerritsen, Eric W. Frew, and Stephen Pledgie. Vision-aided inertial navigation on an uncertain map using a particle filter. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 4189–4194, Kobe, Japan, May 12–17, 2009.

[60] Jr. Teddy Yap, Mingyang Li, Anastasios I. Mourikis, and Christian R. Shelton. A particle filter for monocular vision-aided odometry. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 5663–5669, Shanghai, China, May 9–13, 2011.

[61] Brian Williams, Nicolas Hudson, Brent Tweddle, Roland Brockers, and Larry Matthies. Feature and pose constrained visual aided inertial navigation for computationally constrained aerial vehicles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 431–438, Shanghai, China, May 9–13, 2011.

[62] Stephan Weiss, Markus W. Achtelik, Simon Lynen, Margarita Chli, and Roland Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environment. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 957–964, St. Paul, MN, May 14–18, 2012.

[63] Eagle S. Jones and Stefano Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Int. Journal of Robotics Research*, 30(4):407–430, April 2011.

[64] Todd Lupton and Salah Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Trans. on Robotics*, 28(1):61–76, February 2012.

[65] Luca Carlone, Vito Macchia, Federico Tibaldi, and Basilio Bona. Robot localization and 3D mapping: Observability analysis and applications. In *Proc. of the Int. Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 7–14, Turin, Italy, September 4–6, 2012.

[66] Jonathan Kelly and Gurav S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *Int. Journal of Robotics Research*, 30(1):56–79, January 2011.

[67] Stephan Weiss. *Vision based navigation for micro helicopters.* PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, August 2012.

[68] Alberto Isidori. *Nonlinear Control Systems.* Springer, London, 3 edition, 1995.

[69] Alessandro Chiuso, Paolo Favaro, Hailin Jin, and Stefano Soatto. Structure from motion causally integrated over time. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):523–535, April 2002.

[70] Agostino Martinelli. Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Trans. on Robotics*, 28(1):44–60, February 2012.

[71] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Observability-constrained vision-aided inertial navigation. Technical Report 2012-001, University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, February 2012.

[72] Dimitios G. Kottas, Joel A. Hesch, Sean L. Bowman, and Stergios I. Roumeliotis. On the consistency of vision-aided inertial navigation. In *Proc. of the Int. Symposium on Experimental Robotics*, pages 303–317, Quebec City, Canada, June 17–21, 2012.

[73] Mingyang Li and Anastasios I. Mourikis. Improving the accuracy of EKF-based visual-inertial odometry. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 828–835, Minneapolis, MN, May 14–18, 2012.

[74] Mingyang Li and Anastasios I. Mourikis. High-precision, consistent EKF-based visualinertial odometry. *Int. Journal of Robotics Research*, 32(6):690–711, May 2013.

[75] Averil B. Chatfield. *Fundamentals of High Accuracy Inertial Navigation*. American Institute of Aeronautics and Astronautics, Inc., Reston, VA, 1997.

[76] Nikolas Trawny. Sun sensor model. Technical Report 2005-001, University of Minnesota, Dept. of Comp. Sci. & Eng., January 2005.

[77] Joel A. Hesch, Dimitios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Towards consistent vision-aided inertial navigation. In *Workshop on the Algorithmic Foundations of Robotics*, pages 559–574, Cambridge, MA, June 13–15, 2012.

[78] Jean-Yves Bouguet. Camera calibration toolbox for matlab, 2006. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calibdoc/. [Accessed: January 1, 2012].

[79] S. I. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, October 2002.

[80] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Siam, 2000.

[81] Malcolm D. Shuster. A survey of attitude representations. *Journal of the Astronautical Sciences*, 41(4):439–517, October – December 1993.

[82] Anastasios I. Mourikis, Nikolas Trawny, Stergios I. Roumeliotis, Andrew E. Johnson, Adnan Ansar, and Larry Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Trans. on Robotics*, 25(2):264–280, April 2009. [Best Journal Paper Award].

[83] J. Shi and C. Tomasi. Good features to track. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 593–600, Washington, DC, June 27–July 2, 1994.

[84] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*, pages 674–679, Vancouver, B.C., Canada, August 24–28, 1981.

[85] David Nistér. An efficient solution to the five-point relative pose problem. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 195–202, Madison, WI, June 16–22, 2003.

[86] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2161–2168, New York, NY, June 17–22, 2006.

[87] N. Molton, S. Se, J. M. Brady, D. Lee, and P. Probert. A stereo vision-based aid for the visually impaired. *Image and Vision Computing*, 16(4):251–263, March 1998.

[88] R.O. Castle, G. Klein, and D.W. Murray. Combining MonoSLAM with object recognition for scene augmentation using a wearable camera. *Image and Vision Computing*, 28(11):1548–1556, November 2010.

[89] Andrew J. Davison, Ian Reid, Nicholas Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.

[90] Niko Sünderhauf, Sven Lange, and Peter Protzel. Using the unscented Kalman filter in mono-SLAM with inverse depth parametrization for autonomous airship control. In *Proc. of the IEEE International Workshop on Safety, Security, and Rescue Robotics*, pages 1–6, Rome, Italy, September 27–29, 2007.

[91] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, Nara, Japan, November 13–16, 2007.

[92] Richard A. Newcombe, Steven Lovegrove, and Andrew J. Davison. DTAM: Dense tracking and mapping in real-time. In *International Conf. on Computer Vision*, pages 2320–2327, Barcelona, Spain, November 6–13, 2011.

[93] Javier Civera, Andrew J. Davison, and J. M. M. Montiel. Interacting multiple model monocular SLAM . In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3704–3709, Pasadena, CA, May 19–23, 2008.

[94] Joel A. Hesch and Stergios I. Roumeliotis. A direct least-squares (DLS) method for PnP. In *Proc. of the Int. Conf. on Computer Vision*, pages 383–390, Barcelona, Spain, November 6–13, 2011.

[95] Shunsuke Hijikata, Kenji Terabayashi, and Kazunori Umeda. A simple indoor self-localization system using infrared LEDs. In *Proc. of the Int. Conf. on Networked Sensing Systems*, pages 1–7, Pittsburgh, PA, June 17–19, 2009.

[96] David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[97] Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *Int. Journal of Robotics Research*, 18(3):286–298, March 1999.

[98] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applicatlons to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[99] Adnan Ansar and Kostas Daniilidis. Linear pose estimation from points or lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):578–589, May 2003.

[100] Robert M. Haralick, Hyonam Joo, Chung nan Lee, Xinhua Zhuang, Vinay G. Vaidya, and Man Bae Kim. Pose estimation from corresponding point data. *IEEE Trans. on Systems, Man, and Cybernetics*, 19(6):1426–1446, November/December 1989.

[101] Fredrik Kahl and Didier Henrion. Globally optimal estimates for geometric reconstruction problems. *Int. Journal of Computer Vision*, 74(1):3–15, August 2007.

[102] Gerald Schweighofer and Axel Pinz. Globally optimal $O(n)$ solution to the PnP problem for general comera models. In *Proc. of the Brittish Machine Vision Conf.*, pages 1–10, Leeds, United Kindom, September 1–4, 2008.

[103] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate $O(n)$ solution to the PnP problem. *Int. Journal of Computer Vision*, 81(2):155–166, June 2008.

[104] J. A. Grunert. Das pothenotische problem in erweiterter gestalt nebst bber seine anwendugen in der geodäsie. *Grunerts Archiv für Mathematik und Physik*, pages 238–248, 1841. Band 1.

[105] Radu Horaud, Bernard Conio, Olivier Leboulleux, and Bernard Lacolle. An analytic solution for the perspective 4-point problem. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 500–507, San Diego, CA, June 4–8, 1989.

[106] Long Quan and Zhongdan Lan. Linear N-point camera pose determination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):774–780, August 1999.

[107] Greg Reid, Jianliang Tang, and Lihong Zhi. A complete symbolic-numeric linear method for camera pose determination. In *Proc. of the Int. symposium on Symbolic and Algebraic Computation*, pages 215–223, Philadelphia, PA, August 3–6, 2003.

[108] Jean-Charles Faugére, Guillaume Moroz, Fabrice Rouillier, and Mohab Safey El Din. Classification of the perspective-three-point problem, discriminant variety and real solving polynomial systems of inequalities. In *Proc. of the Int. symposium on Symbolic and Algebraic Computation*, pages 79–86, Hagenberg, Austria, July 20–23, 2008.

[109] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(8):930–943, August 2003.

[110] Yihong Wu and Zhanyi Hu. P*n*P problem revisited. *Journal of Mathematical Imaging and Vision*, 24(1):131–141, January 2006.

[111] Z. Y. Hu and F. C. Wu. A note on the number of solutions of the noncoplanar P4P problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):550–555, April 2002.

[112] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Nashua, NH, 2nd edition, 1999.

[113] F. S. Macaulay. On some formulæ in elimination. *London Mathematics Society*, 35:3–27, May 1902.

[114] Faraz M. Mirzaei and Stergios I. Roumeliotis. Globally optimal pose estimation from line-to-line correspondences. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 5581–5588, Shanghai, China, May 9–13, 2011.

[115] W. Auzinger and H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *Proc. of the Int. Conf. on Numerical Mathematics*, pages 11–30, Singapore, 1988.

[116] Joel A. Hesch and Stergios I. Roumeliotis. A practical guide to DLS PnP. Technical Report 2011-001, University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, August 2011.

[117] K. V. Ramachandra. *Kalman Filtering Techniques for Radar Tracking.* Marcel Decker, Inc., New York, NY, 2000.

[118] Kamiar Aminian, Philippe Robert, Eric Jéquier, and Yves Schutz. Estimation of speed and incline of walking using neural network. *IEEE Trans. on Instrumentation and Measurement*, 44(3):743–746, June 1995.

[119] Wayne A. Soehren, Charles T. Bye, and Charles L. Keyes. Navigation system, method and software for foot travel, U.S. Patent 6,522,266 B1, filed 2000, and issued 2003.

Appendix A

Nomenclature and Abbreviations

$\mathbf{C}(\bar{q})$ rotation matrix parametrized by the quaternion $\bar{q}$

${}^{W}\mathbf{r}$ vector $\mathbf{r}$ expressed with respect to frame $\{W\}$

${}^{W}\mathbf{p}_{Z}$ origin of frame $\{Z\}$ expressed with respect to frame $\{W\}$

$\mathbf{I}_{n}$ the $n \times n$ identity matrix

$\mathbf{0}_{m\times n}$ the $m \times n$ matrix of zeros

3D three dimensions

BLS Batch Least Squares

CGR Cayley-Gibbs-Rodriguez

d.o.f. degrees of freedom

DR Dead Reckoning

DLS Direct Least Squares

EKF Extended Kalman Filter

FEJ First-Estimates Jacobian

f.o.v. field of view

GPS Global Positioning System

ICP Iterative Closest Point

IMM Interacting Multiple Model

IMU Inertial Measurement Unit

INS Inertial Navigation System

KKT Karush-Kuhn-Tucker

KF Kalman Filter

LINS Laser-aided Inertial Navigation System

LM Levenberg-Marquardt

LMI Linear Matrix Inequality

MAP Maximum A Posteriori

MAV Micro Aerial Vehicle

MonoSLAM Monocular Simultaneous Localization and Mapping

NEES Normalized Estimation Error Squared

OC Observability Constrained

PF Particle Filter

P3P Perspective 3-point Pose Determination Problem

pdf probability density function

PnP Perspective n-point Pose Determination Problem

PNA Personal Navigation System

pose position and orientation

RANSAC Random Sample Consensus

RMSE Root Mean Squared Error

rpy roll, pitch, and yaw

SDP Semi-Definite Program

SIFT Scale-Invariant Feature Transform

SLAM Simultaneous Localization And Mapping

UKF Unscented Kalman Filter

VINS Vision-aided Inertial Navigation System

VIO Visual-Inertial Odometry

VT Vocabulary Tree

Appendix B

VINS: Lie derivative observability matrix

In this section, we study the observability properties of system (3.44), by showing that its observability matrix, $\boldsymbol{\Xi}$ [see (3.23)], is of full column rank; thus system (3.44) is observable and the basis functions $\boldsymbol{\beta}$ are the observable modes of the original system (3.26).

Although the observability matrix comprising the spans of all the Lie derivatives of the system (3.26)-(3.27) will, in general, have infinite number of rows, we will only use a subset of its Lie derivatives to prove that it is observable. In particular, since we aim to prove that $\boldsymbol{\Xi}$ is of full column rank, we need only to select a subset of its rows that are linearly independent. Specifically, we select the set

$$\mathcal{L} = \{\mathcal{L}^0\mathbf{h}, \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_{13}\mathbf{g}_{21}}\mathbf{h}, \mathcal{L}^1_{\mathbf{g}_0}\mathbf{h}, \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_{13}\mathbf{g}_{13}}\mathbf{h}, \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_0\mathbf{g}_{21}}\mathbf{h}, \mathcal{L}^2_{\mathbf{g}_0\mathbf{g}_0}\mathbf{h}, \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_0\mathbf{g}_{13}}\mathbf{h}, \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_0\mathbf{g}_0}\mathbf{h}\} \quad \text{(B.1)}$$

where we have used the abbreviated notation $\mathbf{g}_{ij}$ to denote the j-th component of the i-th input, i.e., $\mathbf{g}_{ij} = \mathbf{g}_i\mathbf{e}_j$. The ordering of $\mathcal{L}$ has been selected so as to admit an advantageous structure for analyzing the observability matrix.

The observability sub-matrix corresponding to these Lie derivatives is

$$\boldsymbol{\Xi}' = \begin{bmatrix} \frac{\partial \mathcal{L}^0 \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_{13}\mathbf{g}_{21}} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}^1_{\mathbf{g}_0} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_{13}\mathbf{g}_{13}} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_0\mathbf{g}_{21}} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}^2_{\mathbf{g}_0\mathbf{g}_0} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_0\mathbf{g}_{13}} \mathbf{h}}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}^3_{\mathbf{g}_0\mathbf{g}_0\mathbf{g}_0} \mathbf{h}}{\partial \boldsymbol{\beta}} \end{bmatrix} \tag{B.2}$$

which, after expanding all of the spans of the Lie derivatives in (B.2), has the following structure:

$$\boldsymbol{\Xi}' = \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 6} & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 6} & \mathbf{0}_{1\times 6} \\ \mathbf{X}_{6\times 3} & \boldsymbol{\Psi}_{6\times 6} & \mathbf{0}_{6\times 6} \\ \mathbf{Y}_{6\times 3} & \mathbf{Z}_{6\times 6} & \boldsymbol{\Theta}_{6\times 6} \end{bmatrix}. \tag{B.3}$$

Since the second block row of $\boldsymbol{\Xi}'$ is all zero, we drop it, defining a new matrix $\boldsymbol{\Xi}''$ whose rank is the same, i.e.,

$$\boldsymbol{\Xi}'' = \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 6} & \mathbf{0}_{3\times 6} \\ \mathbf{X}_{6\times 3} & \boldsymbol{\Psi}_{6\times 6} & \mathbf{0}_{6\times 6} \\ \mathbf{Y}_{6\times 3} & \mathbf{Z}_{6\times 6} & \boldsymbol{\Theta}_{6\times 6} \end{bmatrix}. \tag{B.4}$$

Hence, we can prove that system (3.44) is observable, by showing that the matrix $\boldsymbol{\Xi}''$ is of full column rank.

The first step in our proof consists of showing that both $\boldsymbol{\Psi}_{6\times 6}$ and $\boldsymbol{\Theta}_{6\times 6}$ are full-rank matrices. Specifically,

$$\boldsymbol{\Psi} = \begin{bmatrix} -\boldsymbol{\beta}_{21} & 0 & \boldsymbol{\beta}_{11}\boldsymbol{\beta}_{21} & -\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{12} & \boldsymbol{\beta}_{11}^2 + 1 & -\boldsymbol{\beta}_{12} \\ 0 & -\boldsymbol{\beta}_{21} & \boldsymbol{\beta}_{12}\boldsymbol{\beta}_{21} & -\boldsymbol{\beta}_{12}^2 - 1 & \boldsymbol{\beta}_{11}\boldsymbol{\beta}_{12} & \boldsymbol{\beta}_{11} \\ \boldsymbol{\beta}_{21} & 0 & -\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{21} & 4\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{12} & 2\boldsymbol{\beta}_{12}^2 - 2\boldsymbol{\beta}_{11}^2 & \boldsymbol{\beta}_{12} \\ 0 & \boldsymbol{\beta}_{21} & -\boldsymbol{\beta}_{12}\boldsymbol{\beta}_{21} & 2\boldsymbol{\beta}_{12}^2 - 2\boldsymbol{\beta}_{11}^2 & -4\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{12} & -\boldsymbol{\beta}_{11} \\ 0 & 0 & -2\boldsymbol{\beta}_{21}^2 & 2\boldsymbol{\beta}_{12}\boldsymbol{\beta}_{21} & -4\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{21} & 0 \\ 0 & 0 & 0 & 0 & -2\boldsymbol{\beta}_{12}\boldsymbol{\beta}_{21} & -2\boldsymbol{\beta}_{21} \end{bmatrix} \tag{B.5}$$

where $\boldsymbol{\beta}_{ij}$ denotes the j-th component of basis element $\boldsymbol{\beta}_i$. Examining the determinant of $\boldsymbol{\Psi}$, we see that

$$\begin{aligned}\det(\boldsymbol{\Psi}) &= -4\boldsymbol{\beta}_{21}^5\left(\boldsymbol{\beta}_{11}^2+\boldsymbol{\beta}_{12}^2-1\right)\left(2\boldsymbol{\beta}_{11}^2+2\boldsymbol{\beta}_{12}^2+1\right)\\ &= -4\frac{1}{p_z^5}\left(\frac{p_x^2}{p_z^2}+\frac{p_y^2}{p_z^2}-1\right)\left(2\frac{p_x^2}{p_z^2}+2\frac{p_y^2}{p_z^2}+1\right),\end{aligned} \tag{B.6}$$

where for the purpose of analyzing the determinant, we have substituted the basis element definitions [see (3.38) and (3.39)]. First, we note that since the observed point cannot be coincident with the camera center (due to the physical size of the lens and optics), $p_z \neq 0$. Moreover, since we only process features whose positions can be triangulated from multiple views (i.e., features that are not at infinite distance from the camera) $\frac{1}{p_z} \neq 0$. Second, we note that all quantities in the last term are nonnegative, hence,

$$\left(2\frac{p_x^2}{p_z^2}+2\frac{p_y^2}{p_z^2}+1\right) \geq 1. \tag{B.7}$$

This means that $\boldsymbol{\Psi}$ is only rank deficient when the relationship

$$\left(\frac{p_x^2}{p_z^2}+\frac{p_y^2}{p_z^2}-1\right) = 0 \tag{B.8}$$

holds. This equation is satisfied when the observed point lies on a circle with radius 1 on the normalized image plane (i.e., at focal length 1 from the optical center). The corresponding bearing angle to a point on this circle is 45 deg. This corresponds to a zero-probability event, since the control inputs of the system take arbitrary values across time. Thus, we conclude that $\boldsymbol{\Psi}$ is generically full rank.

We now turn our attention to the 6×6 submatrix $\boldsymbol{\Theta}$:

$$\boldsymbol{\Theta} = \begin{bmatrix} -\boldsymbol{\beta}_{21} & 0 & \boldsymbol{\beta}_{11}\boldsymbol{\beta}_{21} & \boldsymbol{\beta}_{21} & 0 & -\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{21} \\ 0 & -\boldsymbol{\beta}_{21} & \boldsymbol{\beta}_{12}\boldsymbol{\beta}_{21} & 0 & \boldsymbol{\beta}_{21} & -\boldsymbol{\beta}_{12}\boldsymbol{\beta}_{21} \\ 0 & -\boldsymbol{\beta}_{21} & \boldsymbol{\beta}_{12}\boldsymbol{\beta}_{21} & 0 & 0 & -\boldsymbol{\beta}_{12}\boldsymbol{\beta}_{21} \\ \boldsymbol{\beta}_{21} & 0 & -\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{21} & 0 & 0 & \boldsymbol{\beta}_{11}\boldsymbol{\beta}_{21} \\ \boldsymbol{\Theta}_{5,1} & \boldsymbol{\Theta}_{5,2} & \boldsymbol{\Theta}_{5,3} & \boldsymbol{\Theta}_{5,4} & \boldsymbol{\Theta}_{5,5} & \boldsymbol{\Theta}_{5,6} \\ \boldsymbol{\Theta}_{6,1} & \boldsymbol{\Theta}_{6,2} & \boldsymbol{\Theta}_{6,3} & \boldsymbol{\Theta}_{6,4} & \boldsymbol{\Theta}_{6,5} & \boldsymbol{\Theta}_{6,6} \end{bmatrix} \tag{B.9}$$

where $\boldsymbol{\Theta}_{i,j}$ denotes the element in the i-th row and j-th column of the matrix $\boldsymbol{\Theta}$, with

$$\begin{aligned}\boldsymbol{\Theta}_{5,1} = &-2\boldsymbol{\beta}_{21}\left(\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\boldsymbol{\beta}_{33}\right)\\ &- \boldsymbol{\beta}_{21}\left(2\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\boldsymbol{\beta}_{33}\right) - 2\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{21}\boldsymbol{\beta}_{42}\end{aligned} \tag{B.10}$$

$$\boldsymbol{\Theta}_{5,2} = 2\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{41}\right) + 2\,\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{41} \tag{B.11}$$

$$\begin{aligned}\boldsymbol{\Theta}_{5,3} = &\,2\,\boldsymbol{\beta}_{21}\left(\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{21}\boldsymbol{\beta}_{31} - \boldsymbol{\beta}_{12}\boldsymbol{\beta}_{43}\right.\\ &\left.+\boldsymbol{\beta}_{11}\left(\boldsymbol{\beta}_{11}\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\boldsymbol{\beta}_{33}\right)\right) - 2\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{42}\\ &- {\boldsymbol{\beta}_{21}}^{2}\,\left(\boldsymbol{\beta}_{31} - \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{33}\right) - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{41}\right)\\ &+ 2\,\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\\ &+ \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{21}\,\left(2\,\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\end{aligned} \tag{B.12}$$

$$\begin{aligned}\boldsymbol{\Theta}_{5,4} = &\,2\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\\ &+ \boldsymbol{\beta}_{21}\,\left(2\,\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right) + \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{42}\end{aligned} \tag{B.13}$$

$$\boldsymbol{\Theta}_{5,5} = -\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{43} - \boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{41}\right) - \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{41} \tag{B.14}$$

$$\begin{aligned}\boldsymbol{\Theta}_{5,6} = &\,{\boldsymbol{\beta}_{21}}^{2}\,\left(\boldsymbol{\beta}_{31} - \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{33}\right) + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{42} - 2\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{31}\right.\\ &\left.-\boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{11}\,\left(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\right)\\ &+ \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{41}\right)\\ &- 2\,\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\\ &- \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{21}\,\left(2\,\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\end{aligned} \tag{B.15}$$

$$\boldsymbol{\Theta}_{6,1} = -2\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{43} - \boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{42}\right) - 2\,\boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{42} \tag{B.16}$$

$$\begin{aligned}\boldsymbol{\Theta}_{6,2} = &\,2\,\boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{41} - \boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - 2\,\boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\\ &- 2\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\end{aligned} \tag{B.17}$$

$$\begin{aligned}\boldsymbol{\Theta}_{6,3} = &\,2\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{41} - {\boldsymbol{\beta}_{21}}^{2}\,\left(\boldsymbol{\beta}_{32} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{33}\right)\\ &- 2\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{32} - \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{43}\right.\\ &\left.-\boldsymbol{\beta}_{12}\,\left(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\right)\\ &+ \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{42}\right)\\ &+ 2\,\boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\\ &+ \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,\left(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - 2\,\boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}\right)\end{aligned} \tag{B.18}$$

$$\boldsymbol{\Theta}_{6,4} = \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{21}\,(\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{42}) + \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{42} \tag{B.19}$$

$$\begin{aligned}\boldsymbol{\Theta}_{6,5} = {} & 2\,\boldsymbol{\beta}_{21}\,(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}) \\ & + \boldsymbol{\beta}_{21}\,(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - 2\,\boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}) - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{41}\end{aligned} \tag{B.20}$$

$$\begin{aligned}\boldsymbol{\Theta}_{6,6} = {} & \boldsymbol{\beta}_{21}{}^{2}\,(\boldsymbol{\beta}_{32} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{33}) - \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{41} \\ & + 2\,\boldsymbol{\beta}_{21}\,(\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{32} - \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{43} \\ & -\boldsymbol{\beta}_{12}\,(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33})) \\ & - \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{21}\,(\boldsymbol{\beta}_{43} + \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{42}) \\ & - 2\,\boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33}) \\ & - \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{21}\,(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{42} - 2\,\boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{21}\,\boldsymbol{\beta}_{33})\,.\end{aligned} \tag{B.21}$$

Again, by examining the matrix determinant, we can show that $\boldsymbol{\Theta}$ is generically full rank. Specifically,

$$\begin{aligned}\det(\boldsymbol{\Theta}) &= 3\boldsymbol{\beta}_{21}^{7}\,(\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{33}\,\boldsymbol{\beta}_{41} - \boldsymbol{\beta}_{32}\,\boldsymbol{\beta}_{42} - \boldsymbol{\beta}_{31}\,\boldsymbol{\beta}_{41} + \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{33}\,\boldsymbol{\beta}_{42}) \\ &= 3\boldsymbol{\beta}_{21}^{7}\begin{bmatrix}\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{33} - \boldsymbol{\beta}_{31} & \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{33} - \boldsymbol{\beta}_{32}\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_{41} \\ \boldsymbol{\beta}_{42}\end{bmatrix}.\end{aligned} \tag{B.22}$$

We hereafter employ the definitions of the basis elements [see (3.38)-(3.41)] in order to analyze $\det(\boldsymbol{\Theta})$. As before, the first term $\boldsymbol{\beta}_{21} = \frac{1}{p_z}$ is strictly positive and finite. For the remaining two terms, it suffices to show that they and their product are generically non-zero.

Starting from the last term, we note that this is zero only when $\mathbf{b}_g = \boldsymbol{\beta}_4 = \mathbf{0}_{3\times 1}$. However, this corresponds to a different system whose system equations would need to be modified to reflect that its gyro is bias free.[1]

The second term is a function of the feature observation, $\boldsymbol{\beta}_1 = \mathbf{h}$, and the velocity expressed in the local frame, $\boldsymbol{\beta}_3 = \mathbf{C}\,\mathbf{v}$, which can be written in a matrix vector form as

$$\begin{bmatrix}\boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{33} - \boldsymbol{\beta}_{31} & \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{33} - \boldsymbol{\beta}_{32}\end{bmatrix}^{T} = \mathbf{A}\boldsymbol{\beta}_3 \tag{B.23}$$

where $\mathbf{A} = \begin{bmatrix}-\mathbf{I}_2 & \boldsymbol{\beta}_1\end{bmatrix}$. Since, generically, $\boldsymbol{\beta}_3 \neq \mathbf{0}_{3\times 1}$ (the camera is moving), and $\mathbf{A}$ is full column rank, their product cannot be zero. Thus, it suffices to examine the case

[1] The observability analysis of such an ideal system is outside the scope of this work.

for which

$$\begin{bmatrix} \boldsymbol{\beta}_{11}\,\boldsymbol{\beta}_{33} - \boldsymbol{\beta}_{31} & \boldsymbol{\beta}_{12}\,\boldsymbol{\beta}_{33} - \boldsymbol{\beta}_{32} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_{41} \\ \boldsymbol{\beta}_{42} \end{bmatrix} = 0 \tag{B.24}$$

$$\Leftrightarrow \begin{bmatrix} \boldsymbol{\beta}_{41} & \boldsymbol{\beta}_{42} \end{bmatrix} \mathbf{A}\boldsymbol{\beta}_3 = 0 \tag{B.25}$$

$$\Leftrightarrow \mathbf{A}\boldsymbol{\beta}_3 = \lambda \begin{bmatrix} \boldsymbol{\beta}_{42} \\ -\boldsymbol{\beta}_{41} \end{bmatrix}, \quad \lambda \in \mathbb{R}. \tag{B.26}$$

This condition, for particular values of $\boldsymbol{\beta}_{41}$ and $\boldsymbol{\beta}_{42}$ (constant), and for time-varying values of $\boldsymbol{\beta}_1$ and hence $\mathbf{A}$, restricts $\boldsymbol{\beta}_3 = \mathbf{C}\,\mathbf{v}$ to always reside in a manifold. This condition, however, cannot hold given that arbitrary control inputs (linear acceleration and rotational velocity) are applied to the system.

We have shown that the diagonal elements of $\boldsymbol{\Xi}''$, i.e., $\boldsymbol{\Psi}$ and $\boldsymbol{\Theta}$ are both full rank [see (B.6) and (B.22)]. We can now apply block-Gaussian elimination in order to show that $\boldsymbol{\Xi}''$ itself is full rank. Specifically, we begin by eliminating both $\mathbf{X}_{6\times3}$ and $\mathbf{Y}_{6\times3}$ using the identity matrix in the upper-left 3×3 sub block. Subsequently, we can eliminate $\mathbf{Z}_{6\times6}$ using $\boldsymbol{\Psi}_{6\times6}$ to obtain the following matrix whose columns span the same space:

$$\boldsymbol{\Xi}''' = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times6} & \mathbf{0}_{3\times6} \\ \mathbf{0}_{6\times3} & \boldsymbol{\Psi}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times6} & \boldsymbol{\Theta}_{6\times6} \end{bmatrix}. \tag{B.27}$$

Since the block-diagonal elements of $\boldsymbol{\Xi}'''$ are all full-rank, and all its off-diagonal block elements are zero, we conclude that $\boldsymbol{\Xi}'''$ is full column rank. This implies that the matrices $\boldsymbol{\Xi}''$, $\boldsymbol{\Xi}'$, and $\boldsymbol{\Xi}$ are also full column rank. Therefore system (3.44) is observable, and our defined basis functions comprise the observable modes of the original system (3.26).

Appendix C

VINS: state transition matrix

We wish to compute the null space of $\mathbf{M}$. To simplify the discussion, we consider a state vector containing a single landmark. We wish to show that our conjectured initial null space $\mathbf{N}_1$,

$$\mathbf{N}_1 = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}\left({}^{I,1}\bar{q}_G\right){}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\lfloor {}^G\mathbf{v}_{I,1} \times \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{p}_{I,1} \times \rfloor {}^G\mathbf{g} \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{f} \times \rfloor {}^G\mathbf{g} \end{bmatrix} \tag{C.1}$$

spans the nullspace of $\mathbf{M}$. We can show this in terms of block rows of $\mathbf{M}$, i.e. it is equivalent to say

$$\mathbf{M}_k\mathbf{N}_1 = \mathbf{0}_{2\times 4} \quad \forall\ k \tag{C.2}$$

with the block rows of $\mathbf{M}$, $\mathbf{M}_k$, given as

$$\mathbf{M}_k = \mathbf{H}_k\Phi(t_k, t_1) \tag{C.3}$$

where $\Phi(t_k, t_1)$ denotes the state transition matrix from t_1 to t_k, i.e. from time step 1 to time step k.

We can further break up the problem by showing (C.2) in terms of the columns of

$\mathbf{N}_k$:

$$\mathbf{M}_k\mathbf{N}_{1(:,1:3)} = \mathbf{0}_{2\times 3} \tag{C.4}$$

$$\mathbf{M}_k\mathbf{N}_{1(:,4)} = \mathbf{0}_{2\times 1} \tag{C.5}$$

where $\mathbf{N}_{1(:,1:3)}$ denotes the submatrix comprising all rows of the first three columns of $\mathbf{N}_1$ and $\mathbf{N}_{1(:,4)}$ denotes all rows of the fourth column of $\mathbf{N}_1$.

To show that equations (C.4) and (C.5) hold, we must compute the rows $\mathbf{M}_k$ and hence the matrix $\Phi(t, t_0)$.

C.1 The State transition matrix $\Phi(t, t_0)$

We can analytically calculate the elements of Φ from the matrix differential equation

$$\dot{\Phi}(t, t_0) = \mathbf{F}_c(t)\Phi(t, t_0) \tag{C.6}$$

where $\mathbf{F}_c(t)$ is the continuous time error-state transition matrix, given by

$$\mathbf{F}_c(t) = \begin{bmatrix} -\lfloor {}^{L(t)}\hat{\boldsymbol{\omega}} \times \rfloor & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\mathbf{C}^T({}^{L(t)}\bar{q}_G)\lfloor {}^{L(t)}\hat{\mathbf{a}} \times \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^{L(t)}\bar{q}_G) & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{C.7}$$

where $L(t)$ denotes the local IMU frame at time t. Here the vectors ${}^{L(t)}\hat{\boldsymbol{\omega}}$ and ${}^{L(t)}\hat{\mathbf{a}}$ do not refer to estimates, but rather, to the true (i.e., noise and bias free) rotational velocity and linear acceleration (see (2.8) and (2.9)).

The structure of $\Phi(t, t_0)$

To simplify our derivation, we will first show that $\Phi(t, t_0)$ has the structure (omitting the time parameters here for clarity)

$$\Phi(t,t_0) = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} & \Phi_{14} & \Phi_{15} & \Phi_{16} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} & \Phi_{24} & \Phi_{25} & \Phi_{26} \\ \Phi_{31} & \Phi_{32} & \Phi_{33} & \Phi_{34} & \Phi_{35} & \Phi_{36} \\ \Phi_{41} & \Phi_{42} & \Phi_{43} & \Phi_{44} & \Phi_{45} & \Phi_{46} \\ \Phi_{51} & \Phi_{52} & \Phi_{53} & \Phi_{54} & \Phi_{55} & \Phi_{56} \\ \Phi_{61} & \Phi_{62} & \Phi_{63} & \Phi_{64} & \Phi_{65} & \Phi_{66} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{31} & \Phi_{32} & \mathbf{I}_3 & \Phi_{34} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{51} & \Phi_{52} & \Phi_{53} & \Phi_{54} & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{C.8}$$

To show this structure, we take advantage of the $\mathbf{F}_c$ matrix structure. Specifically, notice that its second and fourth rows are zero; from (C.6), we then have

$$\dot{\Phi}_{2,:}(t,t_0) = \mathbf{0} \tag{C.9}$$

So we see that the block row $\Phi_{2,:}$ is constant for all t. Then, because the block matrix $\Phi_{2,j}(t_0, t_0)$ is $\mathbf{I}_3$ if $j = 2$ or $\mathbf{0}_3$ if $j \neq 2$, we then have

$$\Phi_{2,j} = \mathbf{0}_3 \quad \forall j \neq 2 \tag{C.10}$$

$$\Phi_{2,2} = \mathbf{I}_3 \tag{C.11}$$

Using the same observation, we can reveal the elements of $\Phi_{4,:}$

$$\dot{\Phi}_{4,:}(t,t_0) = \mathbf{0} \implies \Phi_{4j}(t,t_0) = \mathbf{0} \tag{C.12}$$

for $j \neq 4$, while

$$\Phi_{44}(t,t_0) = \mathbf{I}_3 \tag{C.13}$$

since $\Phi_{44}(t_0, t_0) = \mathbf{I}_3$. The same applies to the elements of Φ that correspond to the landmark.

$$\dot{\Phi}_{6,:}(t,t_0) = \mathbf{0} \implies \Phi_{6j}(t,t_0) = \mathbf{0} \tag{C.14}$$

$$\dot{\Phi}_{:,6}(t,t_0) = \mathbf{0} \implies \Phi_{j6}(t,t_0) = \mathbf{0} \tag{C.15}$$

for $j \neq 6$, while

$$\Phi_{66}(t, t_0) = \mathbf{I}_3 \tag{C.16}$$

We can now take advantage of the structure of $\Phi_{2,:}$, so as to derive the corresponding elements of $\Phi_{1,:}$. In detail,

$$\dot{\Phi}_{13}(t, t_0) = -\lfloor {}^{L(t)}\hat{\boldsymbol{\omega}} \times \rfloor \Phi_{13} \tag{C.17}$$

C.2 Proof of Φ matrix

The solution to this differential equation can be found as

$$\Phi_{13}(t, t_0) = \Phi_{13}(t_0, t_0) \exp\left(\int_{t_0}^{t} -\lfloor {}^{L(\tau)}\hat{\omega} \times \rfloor \mathrm{d}\tau \right) = \mathbf{0}_3 \tag{C.18}$$

because $\Phi_{13}(t_0, t_0) = \mathbf{0}_3$.

The same differential equation appears for Φ_{14}, Φ_{15}. More precisely,

$$\dot{\Phi}_{14}(t, t_0) = -\lfloor {}^{L(\tau)}\hat{\boldsymbol{\omega}} \times \rfloor \Phi_{14} \implies \Phi_{14}(t, t_0) = \mathbf{0}_3 \tag{C.19}$$

$$\dot{\Phi}_{15}(t, t_0) = -\lfloor {}^{L(\tau)}\hat{\boldsymbol{\omega}} \times \rfloor \Phi_{15} \implies \Phi_{15}(t, t_0) = \mathbf{0}_3 \tag{C.20}$$

Turning our attention to, Φ_{35}, which is governed by:

$$\begin{aligned}
\dot{\Phi}_{35} &= \begin{bmatrix} -\mathbf{C}^T({}^{L(t)}q_G)\lfloor {}^{L(t)}\hat{\mathbf{a}} \times \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^{L(t)}q_G) & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \Phi_{35} \\ \mathbf{0} \\ \Phi_{55} \end{bmatrix} \\
&= -\mathbf{C}^T({}^{L(t)}q_G)\lfloor {}^{L(t)}\hat{\mathbf{a}} \times \rfloor \Phi_{15} - \mathbf{C}^T({}^{L(t)}q_G)\Phi_{45} = \mathbf{0} \implies \\
\Phi_{35}(t, t_0) &= \Phi_{35}(t_0, t_0) = \mathbf{0}
\end{aligned} \tag{C.21}$$

Similarly for, Φ_{33},

$$\dot{\Phi}_{33} = \mathbf{F}_{3,:}\Phi_{:,3} = \begin{bmatrix} -\mathbf{C}^T({}^{L(t)}q_G)\lfloor {}^{L(t)}\hat{\mathbf{a}} \times \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^{L(t)}q_G) & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \Phi_{33} \\ \mathbf{0} \\ \Phi_{53} \end{bmatrix} \implies$$

$$\dot{\Phi}_{33} = \mathbf{0} \implies \Phi_{33}(t, t_0) = \Phi_{33}(t_0, t_0) = \mathbf{I}_3 \tag{C.22}$$

And Φ_{55}, which is described by:

$$\dot{\Phi}_{55} = \mathbf{F}_{5,:}\Phi_{:,5} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \Phi_{55} \end{bmatrix} = \mathbf{0} \implies$$

$$\Phi_{55}(t, t_0) = \Phi_{55}(t_0, t_0) = \mathbf{I}_3. \tag{C.23}$$

Analytic expressions for the elements of $\Phi(t, t_0)$

We begin by computing $\Phi_{11}(t, t_0)$. Proceeding from equation (C.6),

$$\begin{aligned} \dot{\Phi}_{11}(t, t_0) &= \mathbf{F}_{1,:}\Phi_{:,1} \\ &= \begin{bmatrix} -\lfloor {}^{L(t)}\hat{\boldsymbol{\omega}} \times \rfloor & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} \Phi_{11} \\ \mathbf{0}_3 \\ \Phi_{31} \\ \mathbf{0}_3 \\ \Phi_{51} \\ \mathbf{0}_3 \end{bmatrix} \\ &= -\lfloor {}^{L(t)}\hat{\omega} \times \rfloor \Phi_{11} \end{aligned} \tag{C.24}$$

Thus, Φ_{11} is given as

$$\begin{aligned}\Phi_{11}(t,t_0) &= \Phi_{11}(t_0,t_0)\exp\left(\int_{t_0}^{t} -\lfloor {}^{L(\tau)}\hat{\omega}\times\rfloor \mathrm{d}\tau\right)\\ &= \exp\left(-\int_{t_0}^{t}\lfloor {}^{L(\tau)}\hat{\omega}\times\rfloor \mathrm{d}\tau\right) \qquad \text{(C.25)}\\ &= \mathbf{C}({}^{L(t)}q_{L(t_0)}) \qquad \text{(C.26)}\end{aligned}$$

where (C.25) holds because $\Phi_{ii}(t,t) = \mathbf{I}_3$ for all i, t, and (C.26) holds from [44].

We now turn our attention to $\Phi_{31}(t,t_0)$. Again, from (C.6),

$$\begin{aligned}\dot{\Phi}_{31}(t,t_0) &= \mathbf{F}_{3,:}\Phi_{:,1}\\ &= \begin{bmatrix}-\mathbf{C}^T({}^{L(t)}q_G)\lfloor {}^{L(t)}\hat{\mathbf{a}}\times\rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^{L(t)}q_G) & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}\begin{bmatrix}\Phi_{11}\\ \mathbf{0}_3\\ \Phi_{31}\\ \mathbf{0}_3\\ \Phi_{51}\\ \mathbf{0}_3\end{bmatrix}\\ &= -\mathbf{C}^T({}^{L(t)}q_G)\lfloor {}^{L(t)}\hat{\mathbf{a}}\times\rfloor\Phi_{11}\\ &= -\lfloor \mathbf{C}^T({}^{L(t)}q_G){}^{L(t)}\hat{\mathbf{a}}\times\rfloor \mathbf{C}^T({}^{L(t)}q_G)\Phi_{11}\\ &= -\lfloor \mathbf{C}^T({}^{L(t)}q_G){}^{L(t)}\hat{\mathbf{a}}\times\rfloor \mathbf{C}^T({}^{L(t)}q_G)\mathbf{C}({}^{L(t)}q_{L(t_0)})\\ &= -\lfloor \mathbf{C}^T({}^{L(t)}q_G){}^{L(t)}\hat{\mathbf{a}}\times\rfloor \mathbf{C}({}^{G}q_{L(t_0)})\\ &= -\lfloor {}^{G}\hat{\mathbf{a}}\times\rfloor \mathbf{C}({}^{G}q_{L(t)}) \qquad \text{(C.27)}\end{aligned}$$

Thus,

$$\begin{aligned}\Phi_{31}(t,t_0) &= -\int_{t_0}^{t}\lfloor {}^{G}\hat{\mathbf{a}}(\tau)\times\rfloor \mathbf{C}({}^{G}q_{L(t_0)})\mathrm{d}\tau\\ &= -\left(\int_{t_0}^{t}\lfloor {}^{G}\hat{\mathbf{a}}(\tau)\times\rfloor \mathrm{d}\tau\right)\mathbf{C}({}^{G}q_{L(t_0)})\\ &= -\left(\int_{t_0}^{t}\lfloor {}^{G}\mathbf{a}(\tau) + {}^{G}\mathbf{g}\times\rfloor \mathrm{d}\tau\right)\mathbf{C}({}^{G}q_{L(t_0)})\\ &= -\lfloor \left({}^{G}\mathbf{v}_{L(t)} - {}^{G}\mathbf{v}_{L(t_0)}\right) + {}^{G}\mathbf{g}\,(t-t_0)\,\times\rfloor \mathbf{C}({}^{G}q_{L(t_0)}) \qquad \text{(C.28)}\end{aligned}$$

Now, we can compute Φ_{51}

$$\begin{aligned}
\dot{\Phi}_{51}(t,t_0) &= \mathbf{F}_{5,:}\Phi_{:,1} \\
&= \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix} \begin{bmatrix}\Phi_{11}\\ \mathbf{0}_3\\ \Phi_{31}\\ \mathbf{0}_3\\ \Phi_{51}\\ \mathbf{0}_3\end{bmatrix} \\
&= \Phi_{31}(t,t_0) \\
&= -\lfloor \left({}^{G}\mathbf{v}_{L(t)} - {}^{G}\mathbf{v}_{L(t_0)}\right) + {}^{G}\mathbf{g}\,(t-t_0)\,\times \rfloor \mathbf{C}({}^{G}q_{L(t_0)}) \qquad \text{(C.29)}
\end{aligned}$$

Thus,

$$\begin{aligned}
\Phi_{51}(t,t_0) &= -\int_{t_0}^{t} \lfloor \left({}^{G}\mathbf{v}_{L(\tau)} - {}^{G}\mathbf{v}_{L(t_0)}\right) + {}^{G}\mathbf{g}\,(\tau-t_0)\,\times \rfloor \mathbf{C}({}^{G}q_{L(t_0)})\mathrm{d}\tau \\
&= -\left(\int_{t_0}^{t} \lfloor \left({}^{G}\mathbf{v}_{L(\tau)} - {}^{G}\mathbf{v}_{L(t_0)}\right) + {}^{G}\mathbf{g}\,(\tau-t_0)\,\times \rfloor \mathrm{d}\tau\right) \mathbf{C}({}^{G}q_{L(t_0)}) \\
&= -\left(\lfloor {}^{G}\mathbf{p}_{L(t)} - {}^{G}\mathbf{p}_{L(t_0)} - {}^{G}\mathbf{v}_{L(t_0)}(t-t_0) + \frac{1}{2}{}^{G}\mathbf{g}(t-t_0)^2 \times \rfloor\right) \mathbf{C}({}^{G}q_{L(t_0)}) \\
&= \lfloor {}^{G}\mathbf{p}_{L(t_0)} + {}^{G}\mathbf{v}_{L(t_0)}(t-t_0) - \frac{1}{2}{}^{G}\mathbf{g}(t-t_0)^2 - {}^{G}\mathbf{p}_{L(t)} \times \rfloor \mathbf{C}({}^{G}q_{L(t_0)}) \qquad \text{(C.30)}
\end{aligned}$$

Now, we will compute $\Phi_{12}(t,t_0)$.

$$\begin{aligned}
\dot{\Phi}_{12}(t,t_0) &= \mathbf{F}_{1,:}\Phi_{:,2} \\
&= \begin{bmatrix}-\lfloor {}^{L(t)}\hat{\boldsymbol{\omega}} \times \rfloor & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix} \begin{bmatrix}\Phi_{12}\\ \mathbf{I}_3\\ \Phi_{32}\\ \mathbf{0}_3\\ \Phi_{52}\\ \mathbf{0}_3\end{bmatrix} \\
&= -\lfloor {}^{L(t)}\hat{\omega} \times \rfloor \Phi_{12} - \mathbf{I}_3 \qquad \text{(C.31)}
\end{aligned}$$

Or,

$$\dot{\Phi}_{12} + \lfloor {}^{L(t)}\hat{\omega} \times \rfloor \Phi_{12} = -\mathbf{I}_3 \qquad \text{(C.32)}$$

To solve (C.32), multiply by an integrating factor $\exp\left(\int_{t_0}^{t} \lfloor {}^{L(\tau)}\hat{\boldsymbol{\omega}} \times \rfloor \mathrm{d}\tau\right)$

$$\begin{aligned}
\dot{\Phi}_{12} + \lfloor {}^{L(t)}\hat{\omega} \times \rfloor \Phi_{12} &= -\mathbf{I}_3 \\
\exp\left(\int_{t_0}^{t} \lfloor {}^{L(\tau)}\hat{\boldsymbol{\omega}} \times \rfloor \mathrm{d}\tau\right)\left(\dot{\Phi}_{12} + \lfloor {}^{L(t)}\hat{\omega} \times \rfloor \Phi_{12}\right) &= -\exp\left(\int_{t_0}^{t} \lfloor {}^{L(\tau)}\hat{\boldsymbol{\omega}} \times \rfloor \mathrm{d}\tau\right) \\
\frac{\mathrm{d}}{\mathrm{d}t}\left[\exp\left(\int_{t_0}^{t} \lfloor {}^{L(\tau)}\hat{\boldsymbol{\omega}} \times \rfloor \mathrm{d}\tau\right)\Phi_{12}\right] &= -\exp\left(\int_{t_0}^{t} \lfloor {}^{L(\tau)}\hat{\boldsymbol{\omega}} \times \rfloor \mathrm{d}\tau\right) \\
\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{C}^T({}^{L(t)}q_{L(t_0)})\Phi_{12}\right] &= -\mathbf{C}^T({}^{L(t)}q_{L(t_0)}) \qquad \text{(C.33)}
\end{aligned}$$

Then,

$$\begin{aligned}
\Phi_{12} &= -\mathbf{C}({}^{L(t)}q_{L(t_0)}) \int_{t_0}^{t} \mathbf{C}^T({}^{L(\tau)}q_{L(t_0)})\mathrm{d}\tau \\
&= -\int_{t_0}^{t} \mathbf{C}({}^{L(t)}q_{L(t_0)})\mathbf{C}^T({}^{L(\tau)}q_{L(t_0)})\mathrm{d}\tau \\
&= -\int_{t_0}^{t} \mathbf{C}^T({}^{L(\tau)}q_{L(t)})\mathrm{d}\tau \qquad \text{(C.34)}
\end{aligned}$$

Now Φ_{32}

$$\begin{aligned}
\dot{\Phi}_{32}(t,t_0) &= \mathbf{F}_{3,:}\Phi_{:,2} \\
&= \begin{bmatrix} -\mathbf{C}^T({}^{L(t)}q_G)\lfloor {}^{L(t)}\hat{\mathbf{a}} \times \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^{L(t)}q_G) & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} \Phi_{12} \\ \mathbf{I}_3 \\ \Phi_{32} \\ \mathbf{0}_3 \\ \Phi_{52} \\ \mathbf{0}_3 \end{bmatrix} \\
&= -\mathbf{C}^T({}^{L(t)}q_G)\lfloor {}^{L(t)}\hat{\mathbf{a}} \times \rfloor \Phi_{12} \\
&= \mathbf{C}^T({}^{L(t)}q_G)\lfloor {}^{L(t)}\hat{\mathbf{a}} \times \rfloor \int_{t_0}^{t} \mathbf{C}^T({}^{L(\tau)}q_{L(t)})\mathrm{d}\tau \qquad \text{(C.35)}
\end{aligned}$$

Then, we have

$$\Phi_{32}(t,t_0) = \int_{t_0}^{t} \mathbf{C}^T({}^{L(s)}q_G)\lfloor {}^{L(s)}\hat{\mathbf{a}} \times \rfloor \int_{t_0}^{s} \mathbf{C}^T({}^{L(\tau)}q_{L(s)})\mathrm{d}\tau\mathrm{d}s \qquad \text{(C.36)}$$

Now, Φ_{52}:

$$\begin{aligned}\dot{\Phi}_{52}(t,t_0) &= \mathbf{F}_{5,:}\Phi_{:,2} \\ &= \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}\begin{bmatrix}\Phi_{12}\\ \mathbf{I}_3\\ \Phi_{32}\\ \mathbf{0}_3\\ \Phi_{52}\\ \mathbf{0}_3\end{bmatrix} \\ &= \Phi_{32} \\ &= \int_{t_0}^{t}\mathbf{C}^T({}^{L(s)}q_G)\lfloor {}^{L(s)}\hat{\mathbf{a}}\times\rfloor\int_{t_0}^{s}\mathbf{C}^T({}^{L(\tau)}q_{L(s)})\mathrm{d}\tau\mathrm{d}s \end{aligned} \tag{C.37}$$

Thus,

$$\Phi_{52} = \int_{t_0}^{t}\int_{t_0}^{\theta}\mathbf{C}^T({}^{L(s)}q_G)\lfloor {}^{L(s)}\hat{\mathbf{a}}\times\rfloor\int_{t_0}^{s}\mathbf{C}^T({}^{L(\tau)}q_{L(s)})\mathrm{d}\tau\mathrm{d}s\mathrm{d}\theta \tag{C.38}$$

Now, Φ_{53}:

$$\dot{\Phi}_{53}(t,t_0) = \mathbf{F}_{5,:}\Phi_{:,3} = \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}\begin{bmatrix}\mathbf{0}_3\\ \mathbf{0}_3\\ \mathbf{I}_3\\ \mathbf{0}_3\\ \Phi_{53}\\ \mathbf{0}_3\end{bmatrix} = \mathbf{I}_3 \tag{C.39}$$

Thus, we simply have

$$\Phi_{53}(t,t_0) = \int_{t_0}^{t}\mathbf{I}_3\mathrm{d}t = \mathbf{I}_3(t-t_0) \tag{C.40}$$

Now we'll compute Φ_{34}:

$$\begin{aligned}
\dot{\Phi}_{34}(t,t_0) &= \mathbf{F}_{3,:}\Phi_{:,4} \\
&= \begin{bmatrix} -\mathbf{C}^T({}^{L(t)}q_G)\lfloor {}^{L(t)}\hat{\mathbf{a}} \times \rfloor & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^{L(t)}q_G) & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \Phi_{34} \\ \mathbf{I}_3 \\ \Phi_{54} \\ \mathbf{0}_3 \end{bmatrix} \\
&= -\mathbf{C}^T({}^{L(t)}q_G) \qquad \text{(C.41)}
\end{aligned}$$

Then,

$$\Phi_{34} = -\int_{t_0}^{t} \mathbf{C}^T({}^{L(\tau)}q_G)\mathrm{d}\tau \qquad \text{(C.42)}$$

Now Φ_{54}:

$$\begin{aligned}
\dot{\Phi}_{54}(t,t_0) &= \mathbf{F}_{5,:}\Phi_{:,4} \\
&= \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \Phi_{34} \\ \mathbf{I}_3 \\ \Phi_{54} \\ \mathbf{0}_3 \end{bmatrix} \\
&= \Phi_{34} \\
&= -\int_{t_0}^{t} \mathbf{C}^T({}^{L(\tau)}q_G)\mathrm{d}\tau \qquad \text{(C.43)}
\end{aligned}$$

Thus

$$\Phi_{54} = -\int_{t_0}^{t}\int_{t_0}^{s} \mathbf{C}^T({}^{L(\tau)}q_G)\mathrm{d}\tau\mathrm{d}s \qquad \text{(C.44)}$$

So we now have the entire matrix $\Phi(t,t_0)$.

The inverse state transition matrix $\Phi(t_0,t) = \Phi(t,t_0)^{-1}$

Let us partition, $\Phi(t,t_0)$ in matrix blocks and apply the block matrix inversion lemma (BMIL):

$$\Phi(t,t_0) = \left[\begin{array}{cc|ccc|c} \Phi_{11} & \Phi_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \hline \Phi_{31} & \Phi_{32} & \mathbf{I}_3 & \Phi_{34} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{51} & \Phi_{52} & \Phi_{53} & \Phi_{54} & \mathbf{I}_3 & \mathbf{0}_3 \\ \hline \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{array}\right] = \left[\begin{array}{ccc} \Gamma_{6\times6} & \mathbf{0}_{6\times9} & \mathbf{0}_{6\times3} \\ \Delta_{9\times6} & E_{9\times9} & \mathbf{0}_{9\times3} \\ \mathbf{0}_{3\times6} & \mathbf{0}_{3\times9} & \mathbf{I}_{3\times3} \end{array}\right] \Longrightarrow$$

$$\Phi(t,t_0)^{-1} = \Phi(t_0,t) = \left[\begin{array}{ccc} \Gamma^{-1} & \mathbf{0}_{6,9} & \mathbf{0}_{6\times3} \\ -E^{-1}\Delta\Gamma^{-1} & E^{-1} & \mathbf{0}_{9\times3} \\ \mathbf{0}_{3\times6} & \mathbf{0}_{3\times9} & \mathbf{I}_{3\times3} \end{array}\right] \tag{C.45}$$

Where, Γ^{-1}, E^{-1} and Δ can be written in elements of $\Phi = \Phi(t,t_0)$. Specifically, by using BMIL on Γ and E,

$$\Gamma = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \Longrightarrow \Gamma^{-1} = \begin{bmatrix} \Phi_{11}^T & -\Phi_{11}^T\Phi_{12} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{C.46}$$

where we used the fact that Φ_{11} is a rotation matrix, hence $\Phi_{11}^{-1} = \Phi_{11}^T$. In the same way, we approach E, which we partition in blocks and apply BMIL:

$$E = \left[\begin{array}{cc|c} \mathbf{I}_{3\times3} & \Phi_{34} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \Phi_{53} & \Phi_{54} & \mathbf{I}_{3\times3} \end{array}\right] = \begin{bmatrix} Z_{6\times6} & \mathbf{0}_{6\times3} \\ H_{3\times6} & \mathbf{I}_{3\times3} \end{bmatrix} \Longrightarrow$$

$$E^{-1} = \begin{bmatrix} Z^{-1} & \mathbf{0} \\ -HZ^{-1} & \mathbf{I} \end{bmatrix} \tag{C.47}$$

where Z^{-1} can be computed using the BMIL:

$$Z = \begin{bmatrix} \mathbf{I} & \Phi_{34} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \Longrightarrow Z^{-1} = \begin{bmatrix} \mathbf{I} & -\Phi_{34} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{C.48}$$

Hence,

$$E^{-1} = \begin{bmatrix} \mathbf{I} & -\Phi_{34} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\Phi_{53} & \Phi_{53}\Phi_{34} - \Phi_{54} & \mathbf{I} \end{bmatrix} \tag{C.49}$$

In conclusion, we can retieve $\Phi(t, t_0)^{-1}$, by substituting our results into (C.45).

$$\Phi(t,\,t_0)^{-1} = \begin{bmatrix} \Phi_{11}^T & -\Phi_{11}^T\Phi_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\Phi_{31}\Phi_{11}^T & \Phi_{31}\Phi_{11}^T\Phi_{12} - \Phi_{32} & \mathbf{I}_3 & -\Phi_{34} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{53}\Phi_{31}\Phi_{11}^T - \Phi_{51}\Phi_{11}^T & -\Phi_{53}\Phi_{31}\Phi_{11}^T\Phi_{12} + \Phi_{51}\Phi_{11}^T\Phi_{12} + \Phi_{53}\Phi_{32} - \Phi_{52} & -\Phi_{53} & \Phi_{53}\Phi_{34} - \Phi_{54} & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{C.50}$$

where $\mathbf{0}_3$ denotes the 3×3 matrix of zeros, and $\mathbf{I}_3$ is the 3×3 identity matrix.

The state transition matrix $\Phi(t_2, t_1)$

Using the results above, we can also derive the analytic expression for $\Phi(t_2, t_1) = \Phi(t_2, t_0)\Phi(t_1, t_0)^{-1}$, that is the transition between a past and a future state. Let us define

$$\Phi(t)_{ij} \triangleq [\Phi(t, t_0)]_{i,j} \tag{C.51}$$

Substituting (C.8) and (C.50) in $\Phi(t_2, t_0)$ and $\Phi(t_1, t_0)^{-1}$ respectivelly and multiplying, we get:

$$\Phi(t_2, t_1) = \begin{bmatrix} \Phi(t_2)_{11}\Phi(t_1)_{11}^T & K & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Lambda & \Xi & \mathbf{I}_3 & -\Phi(t_1)_{34} + \Phi(t_2)_{34} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Pi & P & \Phi(t_2)_{53} - \Phi(t_1)_{53} & T & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{C.52}$$

where

$$K = -\Phi(t_2)_{11}\Phi(t_1)_{11}^T\Phi(t_1)_{12} + \Phi(t_2)_{12} \tag{C.53}$$

$$\Lambda = \Phi(t_2)_{31}\Phi(t_1)_{11}^T - \Phi(t_1)_{31}\Phi(t_1)_{11}^T = \left(\Phi(t_2)_{31} - \Phi(t_1)_{31}\right)\Phi(t_1)_{11}^T \tag{C.54}$$

$$\Xi = -\Phi(t_2)_{31}\Phi(t_1)_{11}^T\Phi(t_1)_{12} + \Phi(t_2)_{32} + \Phi(t_1)_{31}\Phi(t_1)_{11}^T\Phi(t_1)_{12} - \Phi(t_1)_{32} \tag{C.55}$$

$$\Pi = \Phi(t_2)_{51}\Phi(t_1)_{11}^T - \Phi(t_2)_{53}\Phi(t_1)_{31}\Phi(t_1)_{11}^T + \Phi(t_1)_{53}\Phi(t_1)_{31}\Phi(t_1)_{11}^T - \Phi(t_1)_{51}\Phi(t_1)_{11}^T \tag{C.56}$$

$$T = -\Phi(t_2)_{53}\Phi(t_1)_{34} + \Phi(t_2)_{54} + \Phi(t_1)_{53}\Phi(t_1)_{34} - \Phi(t_1)_{54} \tag{C.57}$$

$$\begin{aligned} P = &-\Phi(t_2)_{51}\Phi(t_1)_{11}^T\Phi(t_1)_{12} + \Phi(t_2)_{52} + \Phi(t_2)_{53}\Phi(t_1)_{31}\Phi(t_1)_{11}^T\Phi(t_1)_{12} \\ &- \Phi(t_2)_{53}\Phi(t_1)_{32} - \Phi(t_1)_{53}\Phi(t_1)_{31}\Phi(t_1)_{11}^T\Phi(t_1)_{12} + \\ &\Phi(t_1)_{51}\Phi(t_1)_{11}^T\Phi(t_1)_{12} + \Phi(t_1)_{53}\Phi(t_1)_{32} - \Phi(t_1)_{52} \end{aligned} \tag{C.58}$$

The rows of the observability matrix $\mathbf{M}_k$

We can now compute the block rows $\mathbf{M}_k$ as

$$\begin{aligned} \mathbf{M}_k &= \mathbf{H}_k\Phi(t_k, t_0) \\ &= \mathbf{H}_{cam,1}\mathbf{C}(^{L(k)}q_G)\begin{bmatrix}\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)}\times\rfloor\mathbf{C}^T(^{L(k)}q_G) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 - \mathbf{I}_3 & \mathbf{I}_3\end{bmatrix}\Phi(k,1) \end{aligned} \tag{C.59}$$

Performing this multiplication, we see

$$\begin{aligned} \mathbf{M}_k = &\mathbf{H}_{cam,k}\mathbf{C}(^{L(k)}q_G) \\ &\cdot\begin{bmatrix}\left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)}\times\rfloor\mathbf{C}^T(^{L(k)}q_G)\Phi_{11} - \Phi_{51}\right) & \left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)}\times\rfloor\mathbf{C}^T(^{L(k)}q_G)\Phi_{12} - \Phi_{52}\right) & -\Phi_{53} & -\Phi_{54} & -\mathbf{I} & \mathbf{I}\end{bmatrix} \end{aligned} \tag{C.60}$$

where we let $\Phi_{ij} \triangleq \Phi_{ij}(k,1)$ for clarity. Let us examine the first block column $\mathbf{M}_{k,1}$ in this matrix $\mathbf{M}_k$. Substituting in the terms Φ_{ij} and letting $\mathbf{B}_k$ denote the constant factor $\mathbf{H}_{cam,k}\mathbf{C}(^{L(k)}q_G)$, we have

$$\begin{aligned} \mathbf{M}_{k,1} &= \mathbf{B}_k\left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)}\times\rfloor\mathbf{C}^T(^{L(k)}q_G)\Phi_{11} - \Phi_{51}\right) \\ &= \mathbf{B}_k\left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)}\times\rfloor\mathbf{C}^T(^{L(k)}q_G)\mathbf{C}(^{L(k)}q_{L(1)}) - \Phi_{51}\right) \\ &= \mathbf{B}_k\left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)}\times\rfloor\mathbf{C}^T(^{L(1)}q_G) - \Phi_{51}\right) \\ &= \mathbf{B}_k\left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)}\times\rfloor\mathbf{C}^T(^{L(1)}q_G) - \lfloor {}^G\mathbf{p}_{L(1)} + {}^G\mathbf{v}_{L(1)}(k-1)\delta t - \frac{1}{2}{}^G\mathbf{g}((k-1)\delta t)^2 - {}^G\mathbf{p}_{L(k)}\times\rfloor\mathbf{C}(^Gq_{L(1)})\right) \\ &= \mathbf{B}_k\left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)}\times\rfloor - \lfloor {}^G\mathbf{p}_{L(1)} + {}^G\mathbf{v}_{L(1)}(k-1)\delta t - \frac{1}{2}{}^G\mathbf{g}((k-1)\delta t)^2 - {}^G\mathbf{p}_{L(k)}\times\rfloor\right)\mathbf{C}^T(^{L(1)}q_G) \\ &= \mathbf{B}_k\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)} - {}^G\mathbf{p}_{L(1)} - {}^G\mathbf{v}_{L(1)}(k-1)\delta t + \frac{1}{2}{}^G\mathbf{g}((k-1)\delta t)^2 + {}^G\mathbf{p}_{L(k)}\times\rfloor\mathbf{C}^T(^{L(1)}q_G) \\ &= \mathbf{B}_k\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(1)} - {}^G\mathbf{v}_{L(1)}(k-1)\delta t + \frac{1}{2}{}^G\mathbf{g}((k-1)\delta t)^2\times\rfloor\mathbf{C}^T(^{L(1)}q_G) \end{aligned} \tag{C.61}$$

Thus, we have

$$\mathbf{M}_k = \mathbf{B}_k \begin{bmatrix} \mathbf{T} & \mathbf{V} & -\mathbf{I}_3(k-1)\delta t & \int_{t_0}^{t}\int_{t_0}^{s} \mathbf{C}^T({}^{L(\tau)}q_G)\mathrm{d}\tau\mathrm{d}s & -\mathbf{I} & \mathbf{I} \end{bmatrix} \tag{C.62}$$

where

$$\mathbf{T} = \left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(1)} - {}^G\mathbf{v}_{L(1)}(k-1)\delta t + \frac{1}{2}{}^G\mathbf{g}((k-1)\delta t)^2 \times \rfloor \mathbf{C}^T({}^{L(1)}q_G) \right) \tag{C.63}$$

$$\mathbf{V} = \left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(k)} \times \rfloor \mathbf{C}^T({}^{L(k)}q_G)\Phi_{12} - \Phi_{52} \right) \tag{C.64}$$

We can easily see now that equation (C.4) holds. From equations (C.62) and (C.1), we have

$$\begin{aligned} \mathbf{M}_k\mathbf{N}_{1,1:3} &= \mathbf{B}_k\left(-\mathbf{I}_3 + \mathbf{I}_3\right) \\ &= \mathbf{B}_k\mathbf{0}_{3\times 3} \\ &= \mathbf{0}_{2\times 3} \end{aligned} \tag{C.65}$$

To show that equation (C.5) holds, we proceed similarly. As the terms in the product $\mathbf{M}_k\mathbf{N}_{1,4}$ are so large, we will examine them individually. First, we have the product of the first block column of $\mathbf{M}_k$ and the first block row of $\mathbf{N}_{1,4}$:

$$\begin{aligned} \mathbf{M}_{k,1}\mathbf{N}_{1,4,1} &= \mathbf{B}_k\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(t_1)} - {}^G\mathbf{v}_{L(t_1)}(k-1)\delta t + \frac{1}{2}{}^G\mathbf{g}((k-1)\delta t)^2 \times \rfloor \mathbf{C}^T({}^{L(t_1)}q_G)\mathbf{C}({}^{L(t_1)}q_G){}^G\mathbf{g} \\ &= \mathbf{B}_k\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(t_1)} - {}^G\mathbf{v}_{L(t_1)}(k-1)\delta t + \frac{1}{2}{}^G\mathbf{g}((k-1)\delta t)^2 \times \rfloor {}^G\mathbf{g} \\ &= \mathbf{B}_k\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(t_1)} - {}^G\mathbf{v}_{L(t_1)}(k-1)\delta t \times \rfloor {}^G\mathbf{g} \end{aligned} \tag{C.66}$$

The second product $\mathbf{M}_{k,2}\mathbf{N}_{1,4,2}$ is zero because $\mathbf{N}_{1,4,2} = \mathbf{0}_{3\times 1}$. The third product is

$$\mathbf{M}_{k,3}\mathbf{N}_{1,4,3} = \mathbf{B}_k\mathbf{I}_3(k-1)\delta t\lfloor {}^G\mathbf{v}_{L(t_1)} \times \rfloor {}^G\mathbf{g} = (k-1)\delta t\mathbf{B}_k\lfloor {}^G\mathbf{v}_{L(t_1)} \times \rfloor {}^G\mathbf{g} \tag{C.67}$$

The fourth product is zero again. The fifth is

$$\mathbf{M}_{k,5}\mathbf{N}_{1,4,5} = \mathbf{B}_k\lfloor {}^G\mathbf{p}_{L(t_1)} \times \rfloor {}^G\mathbf{g} \tag{C.68}$$

and the sixth is

$$\mathbf{M}_{k,6}\mathbf{N}_{1,4,6} = -\mathbf{B}_k\lfloor {}^G\mathbf{f} \times \rfloor {}^G\mathbf{g} \tag{C.69}$$

The final product $\mathbf{M}_k\mathbf{N}_{1,4}$ is then the sum of the above six products, or

$$\begin{aligned}
\mathbf{M}_k\mathbf{N}_{1,4} &= \mathbf{B}_k\left(\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(t_1)} - {}^G\mathbf{v}_{L(t_1)}(k-1)\delta t\times\rfloor {}^G\mathbf{g} + (k-1)\delta t\lfloor {}^G\mathbf{v}_{L(t_1)}\times\rfloor {}^G\mathbf{g} + \lfloor {}^G\mathbf{p}_{L(t_1)}\times\rfloor {}^G\mathbf{g} - \lfloor {}^G\mathbf{f}\times\rfloor {}^G\mathbf{g}\right) \\
&= \mathbf{B}_k\lfloor {}^G\mathbf{f} - {}^G\mathbf{p}_{L(t_1)} - {}^G\mathbf{v}_{L(t_1)}(k-1)\delta t + (k-1)\delta t{}^G\mathbf{v}_{L(t_1)} + {}^G\mathbf{p}_{L(t_1)} - {}^G\mathbf{f}\times\rfloor {}^G\mathbf{g} \\
&= \mathbf{B}_k\lfloor {}^G\mathbf{f} - {}^G\mathbf{f} - {}^G\mathbf{p}_{L(t_1)} + {}^G\mathbf{p}_{L(t_1)} - {}^G\mathbf{v}_{L(t_1)}(k-1)\delta t + (k-1)\delta t{}^G\mathbf{v}_{L(t_1)}\times\rfloor {}^G\mathbf{g} \\
&= \mathbf{B}_k\mathbf{0}_{3\times 3}{}^G\mathbf{g} \\
&= \mathbf{0}_{2\times 1}
\end{aligned} \tag{C.70}$$

Appendix D

VINS: nullspace propagation

As we have shown the unobservable subspace, at time t_1 is spanned by

$$\mathbf{N}_{t_1} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}\left({}^{I,t_1}q_G\right){}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\lfloor {}^G\mathbf{v}_{I,t_1} \times \rfloor {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{p}_{I,t_1} \times \rfloor {}^G\mathbf{g} \\ \mathbf{I}_3 & -\lfloor {}^G\mathbf{f} \times \rfloor {}^G\mathbf{g} \end{bmatrix} \tag{D.1}$$

We will show that $\mathbf{N}_{t_2}$ can be estimated with a propagation of $\mathbf{N}_{t_1}$ in time, described with the state transition matrix $\Phi(t_2, t_1)$. Let

$$\mathbf{N'}_{t_2} = \Phi(t_2, t_1)\mathbf{N}_{t_1} \tag{D.2}$$

In what follows, we will prove that $\mathbf{N'}_{t_2} = \mathbf{N}_{t_2}$ Multiplying $\Phi(t_2, t_1)$ (C.52) with the first block column of $\mathbf{N}_{t_1}$,we get:

$$\begin{aligned} \mathbf{N'}_{t_2\,[:,1]} = \Phi(t_2, t_1)_{[:,5]} + \Phi(t_2, t_1)_{[:,6]} &= \begin{bmatrix} \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix}^T \\ \implies \mathbf{N'}_{t_2\,[:,1]} &= \mathbf{N}_{t_2\,[:,1]} \end{aligned} \tag{D.3}$$

Next, we proceed with the second column of $\mathbf{N'}_{t_2}$

$$\begin{aligned}\mathbf{N'}_{t_2[1,2]} &= \Phi(t_2)_{11}\Phi(t_1)_{11}^T\mathbf{C}\left({}^{I,t_1}q_G\right){}^G\mathbf{g} \\ &= \mathbf{C}\left({}^{I,t_2}q_{I,t_0}\right)\mathbf{C}\left({}^{I,t_0}q_{I,t_1}\right)\mathbf{C}\left({}^{I,t_1}q_G\right){}^G\mathbf{g} \\ &= \mathbf{C}\left({}^{I,t_2}q_G\right){}^G\mathbf{g} \\ &= \mathbf{N}_{t_2[1,2]}\end{aligned} \tag{D.4}$$

$$\mathbf{N'}_{t_2[2,2]} = \mathbf{0}_3 = \mathbf{N}_{t_2[2,2]} \tag{D.5}$$

$$\begin{aligned}\mathbf{N'}_{t_2[3,2]} &= \left(\Phi(t_2)_{31} - \Phi(t_1)_{31}\right)\Phi(t_1)_{11}^T\mathbf{C}\left({}^{I,t_1}q_G\right){}^G\mathbf{g} - \lfloor {}^G\mathbf{v}_{I,t_1}\times\rfloor{}^G\mathbf{g} \\ &= \left(-\lfloor\left({}^G\mathbf{v}_{L(t_2)} - {}^G\mathbf{v}_{I,t_0}\right) + {}^G\mathbf{g}\left(t_2 - t_0\right)\times\rfloor + \lfloor({}^G\mathbf{v}_{I,t_1} + {}^G\mathbf{v}_{I,t_0}) - {}^G\mathbf{g}\left(t_1 - t_0\right)\times\rfloor\right) \\ &\quad \mathbf{C}({}^Gq_{I,t_0})\mathbf{C}\left({}^{I,t_0}q_{I,t_1}\right)\mathbf{C}\left({}^{I,t_1}q_G\right){}^G\mathbf{g} - \lfloor {}^G\mathbf{v}_{I,t_1}\times\rfloor{}^G\mathbf{g} \\ &= -\lfloor {}^G\mathbf{v}_{I,t_2}\times\rfloor{}^G\mathbf{g} = \mathbf{N}_{t_2[2,2]}\end{aligned} \tag{D.6}$$

$$\mathbf{N'}_{t_2[4,2]} = \mathbf{0}_3 = \mathbf{N}_{t_2[4,2]} \tag{D.7}$$

$$\begin{aligned}\mathbf{N'}_{t_2[5,2]} &= \left(\Phi(t_2)_{51} - \Phi(t_2)_{53}\Phi(t_1)_{31} + \Phi(t_1)_{53}\Phi(t_1)_{31} - \Phi(t_1)_{51}\right)\Phi(t_1)_{11}^T\mathbf{C}\left({}^{I,t_1}q_G\right){}^G\mathbf{g} \\ &\quad - \left(\Phi(t_2)_{53} - \Phi(t_1)_{53}\right)\lfloor {}^G\mathbf{v}_{I,t_1}\times\rfloor{}^G\mathbf{g} \\ &\quad -\lfloor {}^G\mathbf{p}_{I,t_1}\times\rfloor{}^G\mathbf{g} \implies \\ \mathbf{N'}_{t_2[5,2]} &= \left(\left[\Phi(t_2)_{51} - \Phi(t_1)_{51}\right] + \left[\Phi(t_1)_{53} - \Phi(t_2)_{53}\right]\Phi(t_1)_{31}\right)\mathbf{g} \\ &\quad - \left(\Phi(t_2)_{53} - \Phi(t_1)_{53}\right)\lfloor {}^G\mathbf{v}_{I,t_1}\times\rfloor{}^G\mathbf{g} \\ &\quad -\lfloor {}^G\mathbf{p}_{I,t_1}\times\rfloor{}^G\mathbf{g} \implies \\ \mathbf{N'}_{t_2[5,2]} &= \left(\left[\lfloor {}^G\mathbf{p}_{I,t_0} + {}^G\mathbf{v}_{I,t_0}(t_2 - t_0) - \frac{1}{2}{}^G\mathbf{g}(t_2 - t_0)^2 - {}^G\mathbf{p}_{I,t_2}\times\rfloor\right]\right)\mathbf{g} \\ &\quad - \left(\left[\lfloor {}^G\mathbf{p}_{I,t_0} + {}^G\mathbf{v}_{I,t_0}(t_1 - t_0) - \frac{1}{2}{}^G\mathbf{g}(t_1 - t_0)^2 - {}^G\mathbf{p}_{I,t_1}\times\rfloor\right]\right)\mathbf{g} \\ &\quad + \left(\left[(t_1 - t_0) - (t_2 - t_0)\right]\lfloor({}^G\mathbf{v}_{I,t_1} - {}^G\mathbf{v}_{I,t_0}) + {}^G\mathbf{g}\left(t - t_0\right)\times\rfloor\right)\mathbf{g} \\ &\quad - \left((t_2 - t_0) - (t_1 - t_0)\right)\lfloor {}^G\mathbf{v}_{I,t_1}\times\rfloor{}^G\mathbf{g} \\ &\quad -\lfloor {}^G\mathbf{p}_{I,t_1}\times\rfloor{}^G\mathbf{g} \implies \\ \mathbf{N'}_{t_2[5,2]} &= -\lfloor {}^G\mathbf{p}_{I,t_2}\times\rfloor{}^G\mathbf{g} = \mathbf{N}_{t_2[5,2]}\end{aligned} \tag{D.8}$$

$$\mathbf{N'}_{t_2\,[6,2]} = -\lfloor {}^{G}\mathbf{f} \times \rfloor {}^{G}\mathbf{g} = \mathbf{N}_{t_2\,[6,2]} \tag{D.9}$$

Which completes our proof that the basis of the unobservable subspace at time t_1 evolves to time t_2, through the state transition matrix $\Phi(t_2, t_1)$. That is,

$$\mathbf{N}_{t_2} = \Phi(t_2, t_1)\mathbf{N}_{t_1} \tag{D.10}$$

Appendix E

VINS: feature initialization

As the camera-IMU platform moves into new environments, new features must be added into the map. This entails intersecting the bearing measurements from multiple camera observations to obtain an initial estimate of each new feature's 3D location, as well as computing the initial covariance and cross-correlation between the new landmark estimate and the state. We solve this as a minimization problem over a parameter vector $\mathbf{x} = \begin{bmatrix} \mathbf{x}_{s,1}^T & \cdots & \mathbf{x}_{s,m}^T & | & \mathbf{f}^T \end{bmatrix}^T$, where $\mathbf{x}_{s,i}$, $i = 1 \dots m$, are the m camera poses which the new landmark, $\mathbf{f}$, was observed from. Specifically, we minimize

$$C(\mathbf{x}) = \frac{1}{2}\{(\mathbf{x} - \hat{\mathbf{x}})^T \begin{bmatrix} \mathbf{P}_{ss}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} (\mathbf{x} - \hat{\mathbf{x}}) + \sum_i (\mathbf{z}_i - h(\mathbf{x}))^T \mathbf{R}_i^{-1} (\mathbf{z}_i - h(\mathbf{x}))\} \quad \text{(E.1)}$$

where $\mathbf{P}_{ss}^{-1}$ is the information matrix (prior) of the state estimates across all poses obtained from the filter[1] , and we have no initial information about the feature location (denoted by the block (2,2) element of the prior information being equal to zero). The m measurements $\mathbf{z}_i$, $i = 1 \dots m$ are the perspective projection observations of the point [see (4.13)].

We obtain an initial guess for the landmark location using any intersection method, and then we iteratively minimize (E.1). At each iteration, we need to solve the following

[1] We employ stochastic cloning over m time steps to ensure that the cross-correlation between the camera poses are properly accounted for [79].

linear system of equations

$$\begin{bmatrix} \mathbf{P}_{ss}^{-1} + \mathbf{H}_s^T\mathbf{R}^{-1}\mathbf{H}_s & \mathbf{H}_s^T\mathbf{R}^{-1}\mathbf{H}_f \\ \mathbf{H}_f^T\mathbf{R}^{-1}\mathbf{H}_s & \mathbf{H}_f^T\mathbf{R}^{-1}\mathbf{H}_f \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_s \\ \tilde{\mathbf{x}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{H}_s^T\mathbf{R}^{-1} \\ \mathbf{H}_f^T\mathbf{R}^{-1} \end{bmatrix} \tilde{\mathbf{z}}$$

$$\leftrightarrow \quad \begin{bmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{V} & \mathbf{C} \end{bmatrix} \tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{P} \\ \mathbf{Q} \end{bmatrix} \tilde{\mathbf{z}} \tag{E.2}$$

Applying the Sherman-Morrison-Woodbury matrix identity, we solve the system by inverting the matrix on the left-hand side as

$$\begin{bmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{V} & \mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{\Upsilon}_1 & \boldsymbol{\Upsilon}_2 \\ \boldsymbol{\Upsilon}_3 & \boldsymbol{\Upsilon}_4 \end{bmatrix} \tag{E.3}$$

where

$$\begin{aligned} \boldsymbol{\Upsilon}_1 &= \left(\mathbf{A} - \mathbf{U}\mathbf{C}^{-1}\mathbf{V}\right)^{-1} \\ &= \mathbf{P}_{ss} - \mathbf{P}_{ss}\mathbf{H}_s^T \\ &\quad \cdot \{\mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{H}_f \left(\mathbf{H}_f^T\mathbf{M}^{-1}\mathbf{H}_f\right)^{-1} \mathbf{H}_f^T\mathbf{M}^{-1}\}\mathbf{H}_s\mathbf{P}_{ss} \qquad \text{(E.4)} \\ \boldsymbol{\Upsilon}_2 &= \boldsymbol{\Upsilon}_3^T = -\left(\mathbf{A} - \mathbf{U}\mathbf{C}^{-1}\mathbf{V}\right)^{-1} \mathbf{U}\mathbf{C}^{-1} \\ &= -\mathbf{P}_{ss}\mathbf{H}_s^T\mathbf{M}^{-1}\mathbf{H}_f \left(\mathbf{H}_f^T\mathbf{M}^{-1}\mathbf{H}_f\right)^{-1} \qquad \text{(E.5)} \\ \boldsymbol{\Upsilon}_4 &= \mathbf{C}^{-1}\mathbf{V}\left(\mathbf{A} - \mathbf{U}\mathbf{C}^{-1}\mathbf{V}\right)^{-1} \mathbf{U}\mathbf{C}^{-1} + \mathbf{C}^{-1} \\ &= \left(\mathbf{H}_f^T\mathbf{M}^{-1}\mathbf{H}_f\right)^{-1}. \qquad \text{(E.6)} \end{aligned}$$

Here, $\mathbf{M} = \mathbf{H}_s\mathbf{P}_{ss}\mathbf{H}_s^T + \mathbf{R}$. During each iteration, the parameter vector is updated as

$$\mathbf{x}^{\oplus} = \mathbf{x}^{\ominus} + \begin{bmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{V} & \mathbf{C} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{P} \\ \mathbf{Q} \end{bmatrix} \tilde{\mathbf{z}}. \tag{E.7}$$

After the minimization process converges, we compute the posterior covariance of the new state (including the initialized feature) as

$$\mathbf{P}^{\oplus} = \begin{bmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{V} & \mathbf{C} \end{bmatrix}^{-1} \tag{E.8}$$

where each element is defined from (E.4)-(E.5).

Appendix F

Analytic Substitution of scale and translation in PnP

Employing the expression for $\mathbf{A}$ from (5.6) we have:

$$\mathbf{A}^T\mathbf{A} = \left[\begin{array}{ccc|c} 1 & & & -{}^S\bar{\mathbf{r}}_1^T \\ & \ddots & & \vdots \\ & & 1 & -{}^S\bar{\mathbf{r}}_n^T \\ \hline -{}^S\bar{\mathbf{r}}_1 & \dots & -{}^S\bar{\mathbf{r}}_n & n\mathbf{I} \end{array}\right] \tag{F.1}$$

where we have exploited the fact that ${}^S\bar{\mathbf{r}}_i^T{}^S\bar{\mathbf{r}}_i = 1$. Using block-matrix inversion yields

$$(\mathbf{A}^T\mathbf{A})^{-1} = \left[\begin{array}{c|c} \mathcal{E} & \mathcal{F} \\ \hline \mathcal{G} & \mathcal{H} \end{array}\right] \tag{F.2}$$

$$\mathcal{E} = \mathbf{I} + \begin{bmatrix} {}^S\bar{\mathbf{r}}_1^T \\ \vdots \\ {}^S\bar{\mathbf{r}}_n^T \end{bmatrix} \mathcal{H} \begin{bmatrix} {}^S\bar{\mathbf{r}}_1 \dots {}^S\bar{\mathbf{r}}_n \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} {}^S\bar{\mathbf{r}}_1^T \\ \vdots \\ {}^S\bar{\mathbf{r}}_n^T \end{bmatrix} \mathcal{H}$$

$$\mathcal{G} = \mathcal{H} \begin{bmatrix} {}^S\bar{\mathbf{r}}_1 \dots {}^S\bar{\mathbf{r}}_n \end{bmatrix}, \quad \mathcal{H} = \left(n\mathbf{I} - \sum_{i=1}^{n} {}^S\bar{\mathbf{r}}_i{}^S\bar{\mathbf{r}}_i^T \right)^{-1}.$$

Next, we compute the block matrices $\mathbf{U}$ and $\mathbf{V}$ in (5.7) by post-multiplying the above expression with $\mathbf{A}^T$, i.e.,

$$\left[\begin{array}{c}\mathbf{U}\\\hline\mathbf{V}\end{array}\right] = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T \tag{F.3}$$

$$\mathbf{U} = \begin{bmatrix} {}^S\bar{\mathbf{r}}_1^T & & \\ & \ddots & \\ & & {}^S\bar{\mathbf{r}}_n^T \end{bmatrix} + \begin{bmatrix} {}^S\bar{\mathbf{r}}_1^T \\ \vdots \\ {}^S\bar{\mathbf{r}}_n^T \end{bmatrix} \mathbf{V} \tag{F.4}$$

$$\mathbf{V} = \mathcal{H}\left[{}^S\bar{\mathbf{r}}_1\,{}^S\bar{\mathbf{r}}_1^T - \mathbf{I} \quad \dots \quad {}^S\bar{\mathbf{r}}_n\,{}^S\bar{\mathbf{r}}_n^T - \mathbf{I}\right] \tag{F.5}$$

where $\mathbf{U}$ is $n \times 3n$ and $\mathbf{V}$ is $3 \times 3n$. Based on (5.6), (5.7), and (F.3), we compute both the scale and the translation as a function of the unknown rotation matrix:

$${}^S\mathbf{p}_G = \mathcal{H}\sum_{i=1}^{n}\left({}^S\bar{\mathbf{r}}_i\,{}^S\bar{\mathbf{r}}_i^T - \mathbf{I}\right){}^S_G\mathbf{C}\,{}^G\mathbf{r}_i \tag{F.6}$$

$$\alpha_i = {}^S\bar{\mathbf{r}}_i^T\left({}^S_G\mathbf{C}\,{}^G\mathbf{r}_i + {}^S\mathbf{p}_G\right). \tag{F.7}$$

Appendix G

Transformed measurement noise for modified PnP constraint

In the following appendix, we show the form of the noise term $\boldsymbol{\eta}'$ from the modified geometric constraint equation in (5.16). Our objective is to substitute for ${}^{S}\bar{\mathbf{r}}_i$ in (5.15), and obtain a set of constraint equations that are a function of the measurements and a modified noise term $\boldsymbol{\eta}'_i$. To do so, we note that ${}^{S}\bar{\mathbf{r}}_i$ appears as a linear term (multiplying $\mathbf{b}$) on the left-hand side of (5.14), however, it also enters nonlinearly through $\mathbf{u}_i^T$ and $\mathbf{V}$. We linearize $\mathbf{u}_i$ and $\mathbf{V}$ in the measurement constraint (i.e., apply first-order Taylor series expansion) at the point ${}^{S}\bar{\mathbf{r}}_i = \mathbf{z}_i$ to obtain

$$\begin{aligned}&\left(\mathbf{u}_i\left(\mathbf{z}\right) + \nabla_{\bar{\mathbf{r}}}\mathbf{u}_i\boldsymbol{\eta}\right)^T \mathbf{W}\big(\bar{\mathbf{C}}\left(\mathbf{s}\right)\big)\,\mathbf{b}\,\left(\mathbf{z}_i - \boldsymbol{\eta}_i\right)\\ &\quad\simeq \bar{\mathbf{C}}\left(\mathbf{s}\right)\,{}^{G}\mathbf{r}_i + \left(\mathbf{V}\left(\mathbf{z}\right) + \nabla_{\bar{\mathbf{r}}}\mathbf{V}\boldsymbol{\eta}\right)\mathbf{W}\big(\bar{\mathbf{C}}\left(\mathbf{s}\right)\big)\,\mathbf{b}\end{aligned} \tag{G.1}$$

where $\mathbf{z}$ denotes a stacked vector containing all the measurements, $\boldsymbol{\eta}$ is a stacked vector containing all the noise terms, $\bar{\mathbf{r}}$ is a stacked vector of all $\bar{\mathbf{r}}_i$ unit-vectors, $\nabla_{\bar{\mathbf{r}}}\mathbf{u}_i$ is the Jacobian of $\mathbf{u}_i$ with respect to $\bar{\mathbf{r}}$, and $\nabla_{\bar{\mathbf{r}}}\mathbf{V}$ is the Jacobian of $\mathbf{V}$ with respect to $\bar{\mathbf{r}}$.[1]

We expand (G.1) and bring all quantities involving a noise term on the right-hand side, while all quantities involving non-noise terms we bring to the left, i.e.,

$$\mathbf{u}_i(\mathbf{z})^T\,\mathbf{W}\big(\bar{\mathbf{C}}\left(\mathbf{s}\right)\big)\,\mathbf{b}\,\mathbf{z}_i - \bar{\mathbf{C}}(\mathbf{s})\,{}^{G}\mathbf{r}_i - \mathbf{V}(\mathbf{z})\,\mathbf{W}\big(\bar{\mathbf{C}}\left(\mathbf{s}\right)\big)\mathbf{b} = \boldsymbol{\eta}'_i \tag{G.2}$$

[1] Since both $\mathbf{u}_i$ and $\mathbf{V}$ are available in closed form, their corresponding Jacobians with respect to $\bar{\mathbf{r}}$ can be computed element by element. However, due to the structure of the $\mathbf{u}_i$ and $\mathbf{V}$, we have not yet found a simplified form in which to present them.

which is the modified geometric constraint equation (5.16). The new noise term $\boldsymbol{\eta}_i'$ is

$$\begin{aligned}\boldsymbol{\eta}_i' = &\nabla_{\bar{\mathbf{r}}}\mathbf{V}\boldsymbol{\eta}\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b} - \nabla_{\bar{\mathbf{r}}}\mathbf{u}_i\boldsymbol{\eta}\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b}\mathbf{z}_i \\ &+ \mathbf{u}_i\left(\mathbf{z}\right)^T\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b}\boldsymbol{\eta}_i \end{aligned} \tag{G.3}$$

where we have kept all terms which are linear in original noise $\boldsymbol{\eta}$, however, we have omitted the quadratic noise term, since its impact is negligible. The expected value of $\boldsymbol{\eta}_i' = \mathbf{0}$, since it depends linearly on terms that contain $\boldsymbol{\eta}$, which is zero-mean, i.e.,

$$\begin{aligned} E\left[\boldsymbol{\eta}_i'\right] &= E\left[\nabla_{\bar{\mathbf{r}}}\mathbf{V}\boldsymbol{\eta}\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b} - \nabla_{\bar{\mathbf{r}}}\mathbf{u}_i\boldsymbol{\eta}\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b}\mathbf{z}_i\right. \\ &\qquad \left. + \mathbf{u}_i\left(\mathbf{z}\right)^T\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b}\boldsymbol{\eta}_i\right] \\ &= E\left[\nabla_{\bar{\mathbf{r}}}\mathbf{V}\boldsymbol{\eta}\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b}\right] - E\left[\nabla_{\bar{\mathbf{r}}}\mathbf{u}_i\boldsymbol{\eta}\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b}\mathbf{z}_i\right] \\ &\qquad + E\left[\mathbf{u}_i\left(\mathbf{z}\right)^T\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b}\boldsymbol{\eta}_i\right] \\ &= \nabla_{\bar{\mathbf{r}}}\mathbf{V}E[\boldsymbol{\eta}]\,\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b} - \nabla_{\bar{\mathbf{r}}}\mathbf{u}_i E[\boldsymbol{\eta}]\,\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b}\mathbf{z}_i \\ &\qquad + \mathbf{u}_i\left(\mathbf{z}\right)^T\mathbf{W}\left(\bar{\mathbf{C}}\left(\mathbf{s}\right)\right)\mathbf{b}E\left[\boldsymbol{\eta}_i\right] \\ &= \mathbf{0} \end{aligned} \tag{G.4}$$

The covariance of $\boldsymbol{\eta}_i'$ is $E\left[\left(\boldsymbol{\eta}_i' - E\left[\boldsymbol{\eta}_i'\right]\right)\left(\boldsymbol{\eta}_i' - E\left[\boldsymbol{\eta}_i'\right]\right)^T\right] = E\left[\boldsymbol{\eta}_i'\boldsymbol{\eta}_i'^T\right]$. However, we note that in the LS minimization problem we formulate in (5.17), we do not rely on the noise covariance to compute a solution (i.e., we perform unweighted least-squares), thus we omit the full expression of $E\left[\boldsymbol{\eta}_i'\boldsymbol{\eta}_i'^T\right]$ here for brevity.