UTS
UNIVERSITY OF TECHNOLOGY, SYDNEY

The Invariance of State Estimation for Robot Navigation

by

Teng Zhang

A thesis submitted in partial fulfilment for the
degree of Doctor of Philosophy

at the
Centre for Autonomous Systems
Faculty of Engineering and Information Technology
University of Technology Sydney

February 2018

Declaration of Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signed: Production Note: Signature removed prior to publication.

Date: Feb 22 2018

Abstract

UNIVERSITY OF TECHNOLOGY SYDNEY

Faculty of Engineering and Information Technology
Centre for Autonomous Systems

Doctor of Philosophy

by Teng Zhang

We are living in an era that are being changed by mobile robots such as unmanned aerial vehicles and self-driving cars. State estimation for navigation is one of the fundamental problems in mobile robot's applications. This work entirely focuses on two problems of state estimation for robot navigation, i.e., simultaneous localization and mapping (SLAM), and visual-inertial navigation systems (VINS).

The SLAM problem asks whether it is possible for a robot to build a map of an unknown environment and simultaneously work out its own location within the map. The VINS problem aims at the estimates of a robot's pose and velocity by the on-board sensor funsion of a camera and an inertial measurement unit (IMU). After data association, both SLAM and VINS need a back-end solver to estimate the state of a robot and the environment, which mainly includes two methods: extended Kalman filter and optimization.

The main contribution of this thesis is the invariance theory, which proposes the basic principles for state estimation, i.e., the actual estimates should be invariant under unobservable (deterministic or stochastic) transformations. The invariance theory does not only provide compact, elegant, profound explanations and insights of extended Kalman filter and optimization in SLAM and VINS, but also help to design new algorithms to improve the existing methods.

Acknowledgements

This thesis would not have been possible without the help and support of many people. I would like to thank my supervisors A/Prof. Shoudong Huang, Prof. Dikai Liu and Prof. Gamini Dissanayake for their guidance and support over the last four years. Thanks also go to Kanzhi Wu and Liyang Liu for their constant help in programming instructions. Thanks to Leo, Mahdi and Dinuka for help in the early days when I was still trying to get my head around inertial SLAM. And also to Daobilige Su for insightful discussions on inertial implementations and help in talking through my ideas in the later stages. I would also like to thank everyone at the Centre for Autonomous Systems, University of Technology Sydney.

Contents

List of Figures

List of Tables

Acronyms & Abbreviations

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
CAS	Centre for Autonomous Systems
UTS	University of Technology Sydney
SLAM	Simultaneous localization and mapping
VINS	Visual-Inertial navigation system
EKF	Extended Kalman filter
GN	Gauss-Newton
LM	Levenberg-Marquart
Dogleg	Powell's Dogleg
IMU	Inertial measurement unit
MAV	Micro aerial vehicle
UAV	Unmanned aerial vehicle

Nomenclature

	General Notations
$\mathbb{GL}(n)$	the general linear group of degree n
$\mathbb{R}^3$	The 3-dimensional Euclidean space
$\mathbb{R}^n$	The n-dimensional Euclidean space
$\mathbb{SO}(3)$	The special orthogonal group
$\mathbb{SE}(3)$	The special Euclidean group
$\mathbf{R} \in \mathbb{SO}(3)$	the orientation or the rotation
$\mathbf{p} \in \mathbb{R}^3$	the position

Chapter 1

Introduction

During the last decade, tremendous progress has been made on the development and adoption of mobile robots, which involves a wide range of applications, e.g., service robots, environment reconstruction, urban search and rescue, underground mining, underwater surveillance and planetary exploration. These mobile robots including self-driving cars, micro aerial vehicles and Mars rovers, are usually required to navigate and perform the specific task in a partially known or unknown environment. When the external information such as a global positioning system (GPS) is unavailable or not enough, a mobile robot needs to constantly estimate its state such as pose and velocity.

1.1 Background

Obviously, robust and accurate state estimation is a prerequisite for robot navigation in the unknown or GPS-denied environment. The typical examples are autonomous ground vehicles and unmanned aerial vehicles (UAV) (Fig. 1.1). To estimate the state using measurements from on-board sensors, a mobile robot needs to constantly estimate the robot pose and simultaneously build a sparse or dense map of the environment. This fundamental problem in robotics is known as simultaneous localization and mapping (SLAM).

Moreover, cameras and IMUs are widely used as the on-board sensors of mobile robots due to their small size, light weight and low power consumption. Besides the advantages

mentioned above, the sensors' complementarity is another benefit. As well-known, monocular visual-SLAM allows a robot to estimate its trajectory and the locations of landmarks in the map, up to an unknown metric scale. An IMU, as a complementary component to the camera, can provide the metric information when fused with the camera measurements. This sensor fusion problem is known as visual-inertial navigation system (VINS). In addition, VINS is also used in the virtual reality application (e.g., Fig. 1.2).

Fig. 1.3 provides the flowcharts of SLAM and VINS. In this work, we mainly focus on the back-end of SLAM and VINS from the viewpoint of two nonlinear estimation approaches: the extended Kalman filter (EKF) based approach and the optimization-based approach.

FIGURE 1.1: Examples of robots that need solutions of SLAM and VINS: the automated guided vehicle and the unmanned aerial vehicle

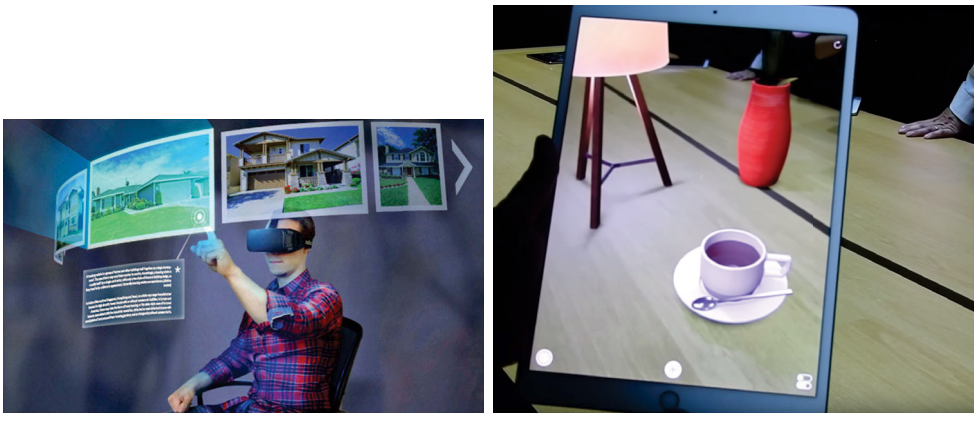

FIGURE 1.2: Applications related to SLAM and VINS beyond robot navigation: virtual reality and augmented reality

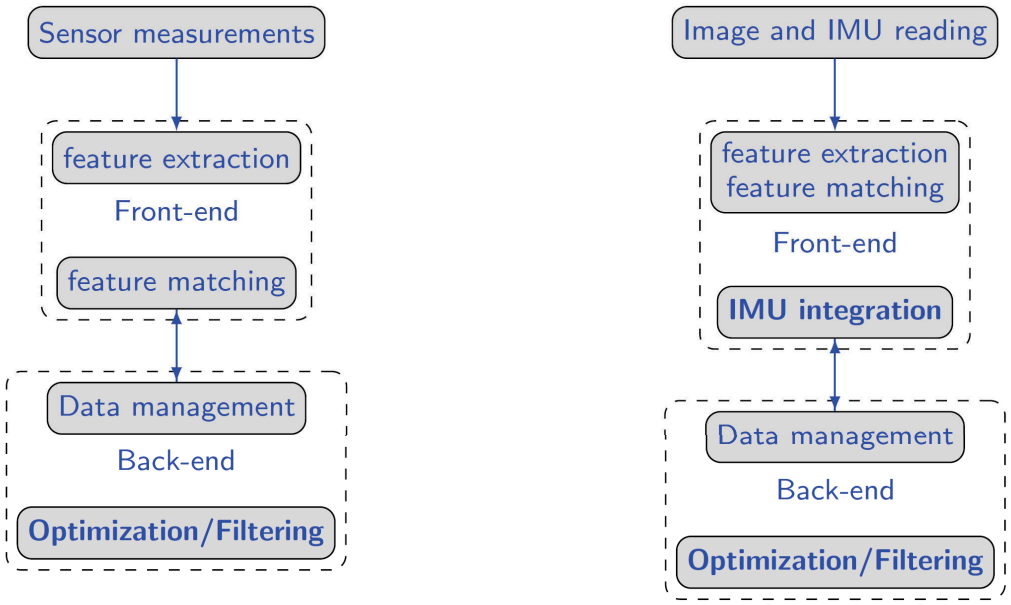

FIGURE 1.3: Left: the flow chart of SLAM. Right: the flow chart of VINS.

1.2 Motivation

The linear-model Gaussian-noise case allows exact inference by using simple algebraic operation, such as Kalman filter of linear least squares. However, this can not be directly applied to SLAM and VINS as they are nonlinear estimation problems. For SLAM and VINS, two fundamental approaches are widely used.

EKF

EKF was one of the earliest solutions to the SLAM problem and played a historical role. Due to its ease of implementation and relatively low processing requirements, EKF is still a popular and primarily solution to VINS. One of the primary limitations of the EKF based SLAM is the possible inconsistency, namely, the covariance matrix calculated by the EKF estimator underestimates the actual uncertainty. Thus the mean estimate is unreliable, which in turn affects the accuracy of the EKF estimator. Conventional view holds that EKF marginalizes out the previous robot pose at the propagation stage and thus the linearization errors involving previous states cannot be corrected. Thus the linearization errors are thought as the main reason for the inconsistency of EKF. The inconsistency of

EKF SLAM and EKF VINS algorithms has been analyzed for a decade. However, to the best knowledge of the author, few has explained the inconsistency issue in the sense of both probabilistic framework and control theory. Therefore, we are trying to find out the fundamental reason of the inconsistency of EKF from the viewpoints of both probabilistic framework and control theory.

Optimization

In contrast to EKF, the optimization-based methods for SLAM and VINS have attracted more attention in the recent decade. The optimization-based methods estimate the historical states using all measurements. Relinearization is allowed in the optimization process and it can reduce the linearization errors in general. In general, the optimization-based methods can achieve more accurate estimates, compare to the EKF-based methods. However, nonlinear optimization always needs multiple iterations to converge. One of the contributions in this thesis is the new algorithm with improved convergence in the SLAM problem. For the VINS problem, we derive a more reasonable formulation that is able to consider more correlation between the involved variables (pose, velocity and bias).

1.3 Contributions

The main contributions of the thesis are the proposed invariance concept and its applications in SLAM and VINS.

- **Invariant EKF-SLAM**

  The conventional EKF-SLAM algorithm is known as an inconsistent estimator. The linearization errors are often considered to be related to the change of the system's observability and thus it is regarded as the root reason for the inconsistency. In Chapter 3, we clarify that the uncertainty representation (together with the retraction $\oplus$) used in the conventional EKF algorithm makes the filter *not* invariant under the *stochastic* rigid body transformation, which is the root reason for the inconsistency. Then we prove that the invariance property can be satisfied when using a

novel uncertainty representation with the new manifold $\mathcal{G}(N)$. The proposed filter is shown to outperform the first estimate Jacobian filter [1] and the robocentric filter [2].

- **Invariant EKF-VINS**

  The conventional EKF-VINS algorithm is also inconsistent because it is *not* invariant under the *stochastic* unobservable transformation, associated with a translation and a rotation about the gravitational direction. In Chapter 4, we provide the necessary and sufficient conditions of the invariance under *stochastic* unobservable transformation for the general system. We also analytically derive a new filter RIEKF-VINS that preserves the invariance property and does *not* need to *enforce the Jacobians*. Considering the expensive cost of maintaining the covariance matrix for a number of landmarks, we then integrate RIEKF-VINS into the well-known visual odometry MSCKF framework [3] such that the modified algorithm has much better consistency and linear complexity w.r.t. the number of landmarks. Both Monte Carlo simulations and real-world experiments are used to validate the proposed method. A video can be viewed at $https://www.youtube.com/watch?v=w8Zp\_hbAAto$.

- **Invariant IMU Factor**

  Motivated by the linear error-state propagation of RIEKF-VINS, in Chapter 6 we analytically derive a new IMU factor under the Factor Graph framework, which is different from the existing ones [4][5][6][7]. Firstly, the new IMU factor is born to be a pre-integration method on manifold such that all Jacobians (even including the Jacobians w.r.t. IMU biases) can be pre-integrated by any high-order numerical integration method. Secondly, the IMU factor does *not* need the assumption that the IMU biases are constant between two consecutive key-frames. In the new IMU factor, the error (between actual and predicted IMU states) and the error (between actual and predicted IMU biases) are fully coupled together, which is different from the work in [4][6][7]. Therefore, this new IMU factor may be a better description for the physical system. From the results of our tests, the advantage of the new IMU factor is more obvious under the condition of the low-frequency IMU measurements and the fast change of IMU biases, compared to [4][6][7]. Hence, we argue that the

new IMU factor has potential to be applied to low-cost IMUs. We have also written a VINS code package based on ORB-SLAM2, validated using the Euroc dataset. The novelty of our code is the new IMU factor and the invariant vision factor. Under this framework, our IMU factor slightly outperforms the factor [6]. A video can be viewed at $https://www.youtube.com/watch?v=r_hhzKwMfyQ$.

1.4 Publications

Journal

- *Convergence and Consistency Analysis for a 3D Invariant-EKF SLAM*;
  Teng Zhang, Kanzhi Wu, Jingwei Song, Shoudong Huang and Gamini Dissanayake; IEEE Robotics and Automation Letters (Volume: 2, Issue: 2, April 2017); also presented in International Conference on Robotics and Automation (ICRA 2017);

Conference

- *An Invariant-EKF VINS Algorithm for Improving Consistency*;
  Kanzhi Wu*, **Teng Zhang***, Daobilige Su, Shoudong Huang and Gamini Dissanayake; IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017);
  The notation * refers to the equal contribution.

- *Constrained Sampling of 2.5D Probabilistic Maps for Augmented Inference*;
  Lei Shi, Jaime Valls Miro, **Teng Zhang**, Teresa Vidal Calleja, Liye Sun and Gamini Dissanayake; IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016);

- *A Method of State Estimation for Underwater Vehicle Navigation Around A Cylindrical Structure*;
  Teng Zhang, Shoudong Huang, Dikai Liu, Chunlin Zhou and Rong Xiong; IEEE Conference on Industrial Electronics and Applications (ICIEA 2016)

- *Active Object Detection and Pose Estimation in General Belief Space*;
  Kanzhi Wu, **Teng Zhang**, Ravindra Ranasinghe, Shoudong Huang and Gamini

Dissanayake; Robot-Environment Interaction for Perception and Manipulation: Interactive Perception Meets Reinforcement Learning and Optimal Control at Robotic Science and Systems (RSS-Workshop 2016)

- *Comparison of Two Strategies of Path Planning for Underwater Robot Navigation Under Uncertainty*;
  Teng Zhang, Shoudong Huang and Dikai Liu; International Conference on Control, Automation, Robotics and Vision (ICARCV 2014)

1.5 Thesis Outline

- Chapter 2: We introduce some preliminary knowledge on Lie groups and their applications in optimization and filter, which are useful in the analysis in the following chapters.
- Chapter 3: We investigate the inconsistency of EKF-SLAM and analyze the convergence and consistency of the RIEKF-SLAM algorithm.
- Chapter 4: We investigate the inconsistency of EKF-VINS and propose the novel RIEKF-VINS algorithm that can improve consistency. We also integrate RIEKF-VINS into the MSCKF framework.
- Chapter 5: We propose the invariance concept in the optimization framework. Then we propose a method satisfying this property and prove that the convergence can be improved.
- Chapter 6: We derive a novel invariant IMU factor based on the idea of the RIEKF-VINS algorithm, which can achieve high-accuracy.
- Chapter 7: We present conclusions and propose some future work.
- Appendix: We provide extended proofs for some theorems in this thesis.

Chapter 2

Preliminaries

Many problems in state estimation require operations in Lie groups (e.g., $\mathbb{SO}(3)$ (3-dimension) and $\mathbb{SE}(3)$ (6-dimension)) [6] [8]. A traditional method first parameterizes the Lie group (e.g., using Euler angles to represent $\mathbb{SO}(3)$) as a column vector and then perform operation in Euclidean space. However, a parameterization in a non-Euclidean space may lead to representation singularity. Furthermore, parameterization makes the gradient vector or Jacobian matrix very complicated. This chapter provides a simple tutorial of Lie groups and their applications in optimization and filter. More details of Lie group can be found in [9] [10].

2.1 $\mathbb{GL}(n)$ and the skew symmetric operator $S(\cdot)$

$\mathbb{GL}(n)$ consists of all $n \times n$ real matrices whose determinant are non-zero, defined as

Definition 2.1.

$$\mathbb{GL}(n) = \{\mathbf{A} \in \mathbb{R}^{n\times n} | \det(\mathbf{A}) \neq 0\}. \tag{2.1}$$

The skew symmetric operator $S(\cdot)$ is a linear transformation from $\mathbb{R}^3$ to $\mathbb{R}^{3\times 3}$, defined as

$$S(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{2.2}$$

for $\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \in \mathbb{R}^3$. Note that for $\mathbf{a}$ and $\mathbf{b} \in \mathbb{R}^3$:

$$S(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b} = -\mathbf{b} \times -\mathbf{a} = S(\mathbf{b})\mathbf{a}. \tag{2.3}$$

In addition,

$$S(\mathbf{Ax}) = \det(\mathbf{A})(\mathbf{A}^\intercal)^{-1} S(\mathbf{x})\mathbf{A}^{-1} \tag{2.4}$$

for $\mathbf{x} \in \mathbb{R}^3$ and $\mathbf{A} \in \mathbb{GL}(3)$. The derivation of (2.4) is given in Theorem 2.3.3 of [11].

2.2 Group and $\mathbb{GL}(n)$

In this section, we first introduce the fundamental concepts of group.

A group $\mathbf{G}$ is a set together with a group operation $\cdot$ that satisfy the following requirements

- **Closure**: For all $\mathbf{a}$ and $\mathbf{b} \in \mathbf{G}$, $\mathbf{a} \cdot \mathbf{b} \in \mathbf{G}$.
- **Associativity**: For all $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c} \in \mathbf{G}$, $(\mathbf{a} \cdot \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot (\mathbf{b} \cdot \mathbf{c})$.
- **Identity element**: There exists a unique element $\mathbf{I} \in \mathbf{G}$ such that $\mathbf{I} \cdot \mathbf{a} = \mathbf{a} \cdot \mathbf{I} = \mathbf{a}$ for all $\mathbf{a} \in \mathbf{G}$.
- **Inverse element**: For each $\mathbf{a} \in \mathbf{G}$, there exists an element $\mathbf{a}^{-1}$ such that $\mathbf{a}^{-1} \cdot \mathbf{a} = \mathbf{a} \cdot \mathbf{a}^{-1} = \mathbf{I}$.

Example 2.1. *Obviously, $\mathbb{GL}(n)$ is a group together with the matrix multiplication as the group operation. The identity element of $\mathbb{GL}(n)$ is $\mathbf{I}_n$ (the identity matrix).*

In fact, an element $\mathbf{a}$ in group is usually represented as a

- **state** contained in $\mathbf{G}$;
- **transformation** from $\mathbf{G}$ to $\mathbf{G}$, if we define $\mathbf{a}(\mathbf{b}) := \mathbf{a} \cdot \mathbf{b}$ or $\mathbf{a}(\mathbf{b}) := \mathbf{b} \cdot \mathbf{a}$, for each $\mathbf{b} \in \mathbf{G}$.

Therefore, a group is a set of states and also a set of transformations.

2.3 An abstract matrix Lie group $\mathbb{G}$

Definition 2.2. A matrix Lie group $\mathbb{G}$ is a closed subgroup of $\mathbb{GL}(n)$ with the matrix multiplication as group operation, which is also a $\dim(\mathbb{G})$ dimensional manifold. The Lie algebra $\mathfrak{g}$ of $\mathbb{G}$ is the tangent space at the identity element $\mathbf{I}$.

The Lie algebra $\mathfrak{g}$ is the tangent space of the manifold $\mathbb{G}$ at the identity $\mathbf{I}$. There is an open set $\mathbf{U} \subseteq \mathbb{G}$ containing $\mathbf{I}$ and a one to one mapping f: $\mathfrak{g} \to \mathbf{U}$. In this thesis, we do not provide all details of Lie algebra knowledge but point out there is an open set of $\mathbf{U}_c \subseteq \mathbb{R}^{\dim(\mathbb{G})}$ containing $\mathbf{0}$ and a one to one mapping f_c: $\mathfrak{g} \to \mathbf{U}_c$, namely

$$\mathfrak{g} \cong \mathbb{R}^{\dim(\mathbb{G})}. \tag{2.5}$$

Therefore, we do not distinguish between $\mathfrak{g}$ and $\mathbb{R}^{\dim(\mathbb{G})}$. For more details, see [10].

Some Associated Operations

Lie group $\mathbb{G}$ and Lie algebra $\mathfrak{g}$ are accompanied by some other operations. For simplicity, the existences of the following operations are not discussed.

- The exponential mapping exp: $\mathfrak{g} \to \mathbb{G}$. Note that exp is subjective,

$$\begin{aligned} \exp(\mathbf{0}) &= \mathbf{I} \\ (\exp(\mathbf{x}))^{-1} &= \exp(-\mathbf{x}) \end{aligned} \tag{2.6}$$

  for all $\mathbf{x} \in \mathfrak{g}$. Also, there exists an open set $\mathbf{U} \subseteq \mathbb{G}$ containing $\mathbf{0}$ such that $\exp|_{\mathbf{U}}$ is a one to one mapping.

- The logarithm mapping $\log : \mathbb{G} \to \mathfrak{g}$ is the inverse transformation of exp: for all $\mathbf{x} \in \mathbf{U}$

$$\log(\exp(\mathbf{x})) = \mathbf{x}. \tag{2.7}$$

- The operator $\mathrm{Ad}_{\mathbf{g}_1}$ for each $\mathbf{g}_1 \in \mathbb{G}$ is a mapping from $\mathbb{G}$ to $\mathbb{G}$: for all $\mathbf{g}_2 \in \mathbb{G}$

$$\mathrm{Ad}_{\mathbf{g}_1}\mathbf{g}_2 = \mathbf{g}_1\mathbf{g}_2\mathbf{g}_1^{-1} \in \mathbb{G} \tag{2.8}$$

- The adjoint operator $\mathrm{ad}_{\mathbf{g}}$ ($\mathbf{g} \in \mathbb{G}$) is a linear transformation from $\mathfrak{g}$ to $\mathfrak{g}$ such that for all $\mathbf{x} \in \mathfrak{g}$,

$$\exp(\mathrm{ad}_{\mathbf{g}}\mathbf{x}) = \mathrm{Ad}_{\mathbf{g}} \exp(\mathbf{x}). \tag{2.9}$$

- The linear transformation $\mathrm{ad}_{\mathbf{g}}$ ($\mathbf{g} \in \mathbb{G}$) can be also regarded as a square matrix. Based on this concept, the right Jacobian $\mathrm{J}_r(\mathbf{x})$ and the left Jacobian $\mathrm{J}_l(\mathbf{x})$ ($\mathbf{x} \in \mathfrak{g}$) are two linear transformations from $\mathfrak{g}$ to $\mathfrak{g}$, defined as

$$\begin{aligned} \mathrm{J}_r(\mathbf{x}) &= \int_0^1 \mathrm{ad}_{\exp(-\mathbf{x}s)} ds \\ \mathrm{J}_l(\mathbf{x}) &= \int_0^1 \mathrm{ad}_{\exp(\mathbf{x}s)} ds \end{aligned} \tag{2.10}$$

  for $\mathbf{x} \in \mathfrak{g}$. Furthermore, J_r and J_l have the properties:

$$\begin{aligned} \exp(\mathrm{J}_r(\mathbf{x})) &= \exp(\mathrm{J}_l(-\mathbf{x})) \\ \exp(\mathbf{x}+\mathbf{y}) &= \exp(\mathbf{x}) \exp(\mathrm{J}_r(\mathbf{x})\mathbf{y} + o(\|\mathbf{y}\|)) \\ \exp(\mathbf{x}+\mathbf{y}) &= \exp(\mathrm{J}_l(\mathbf{x})\mathbf{y} + o(\|\mathbf{y}\|)) \exp(\mathbf{x}) \end{aligned} \tag{2.11}$$

  for all $\mathbf{x}$ and $\mathbf{y} \in \mathfrak{g}$.

Remark 2.3. Note that exp and log are bijection when the field of definitions are restricted to $\mathbf{U}$ and $\{\exp(\mathbf{x}) | \mathbf{x} \in \mathbf{U}\} \subseteq \mathbb{G}$, respectively. Therefore, exp and log are the opposite of each other.

Remark 2.4. A linear transformation from $\mathbb{R}^n$ to $\mathbb{R}^m$ can be regarded as a $m \times n$ matrix. Hence, $\mathrm{ad}_{\mathbf{g}}$ and $\mathrm{J}_r(\mathbf{x})$ can be regarded as $\dim(\mathbb{G}) \times \dim(\mathbb{G})$ matrices.

Uncertainty: The Gaussian Case

Different from the uncertainty representation in Euclidean space, the uncertainty for the element $\mathbf{g} \in \mathbb{G}$ can be defined in the tangent space as the following:

$$\mathbf{g} = \hat{\mathbf{g}} \exp(\epsilon) \tag{2.12}$$

or

$$\mathbf{g} = \exp(\epsilon)\hat{\mathbf{g}} \tag{2.13}$$

where $\mathbf{g} \in \mathbb{G}$ denotes the element (ground truth) to be estimated, $\hat{\mathbf{g}} \in \mathbb{G}$ is the nominal estimate, and $\epsilon \in \mathfrak{g} = \mathbb{R}^{\dim(G)}$ represents the uncertainty/error.

For a Gaussian case,

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}). \tag{2.14}$$

Note that (2.12) and (2.13) are two different uncertainty descriptions, which imply that for $\mathbf{g} \in \mathbb{G}$,

$$p(\mathbf{g}|(\hat{\mathbf{g}}, \mathbf{\Sigma})) \propto e^{-\mathbf{e}_1^\intercal \mathbf{\Sigma}^{-1} \mathbf{e}_1} \tag{2.15}$$

and

$$p(\mathbf{g}|(\hat{\mathbf{g}}, \mathbf{\Sigma})) \propto e^{-\mathbf{e}_2^\intercal \mathbf{\Sigma}^{-1} \mathbf{e}_2}, \tag{2.16}$$

respectively, where $\mathbf{e}_1 = \log(\hat{\mathbf{g}}^{-1}\mathbf{g})$, $\mathbf{e}_2 = \log(\mathbf{g}\hat{\mathbf{g}}^{-1})$ and $(\hat{\mathbf{g}}, \mathbf{\Sigma})$ represents the probabilistic knowledge.

2.4 Lie Group: some instances

Based on the fundamental knowledge of an abstract Lie group provided in last section, this section introduce some instances of matrix Lie group that will be used in the later chapters.

The Special Orthogonal Group $\mathbb{SO}(3)$

The special orthogonal group $\mathbb{SO}(3)$ as a 3-dimensional manifold is also called rotation matrix group, defined as

Definition 2.5.

$$\mathbb{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} | \mathbf{R}^\intercal \mathbf{R} = \mathbf{I}_3, \ \det(\mathbf{R}) > 0\}. \tag{2.17}$$

In robotics, an element in $\mathbb{SO}(3)$ is usually used to represent the

- **orientation** of a camera/robot/IMU/coordinate system, etc;
- **transformation of rotation**.

Traditionally, orientation is represented by the minimal parameterization such as ZYX-Euler angles (yaw-pitch-roll) or axis-angle. However, it is a fundamental topological fact that singularities can never be eliminated in any minimal parameterization for orientation. Similarly, it is impossible to find a global two-dimensional coordinate chart on a sphere [12]. The possibility of singularity may affect the process of state estimation, resulting in numerical instability even failure. Therefore, a switch strategy using several minimal parameterizations with different singularities is required. Obviously, this switch method makes problem more complicated. The other commonly used parameterization for orientation is unit quaternion. Although unit quaternion does not have a singularity issue, a normalization constraint must be maintained, which is not intrinsically and naturally satisfied by the standard addition.

The Lie algebra of $\mathbb{SO}(3)$ is $\mathbf{so}(3) \cong \mathbb{R}^3$. In the following, we directly provide the functions $\exp(\cdot)$, $\log(\cdot)$, Ad, J_r related to $\mathbb{SO}(3)$ and $\mathbf{so}(3)$.

- The exponential mapping: for $\mathbf{x}(\neq \mathbf{0}) \in \mathbb{R}^3$

$$\begin{aligned} \exp(\mathbf{x}) &= \mathbf{I}_3 + \frac{\sin(\|\mathbf{x}\|)}{\|\mathbf{x}\|} S(\mathbf{x}) + \frac{1-\cos(\|\mathbf{x}\|)}{\|\mathbf{x}\|^2} S^2(\mathbf{x}), \\ \exp(\mathbf{0}) &= \mathbf{I}_3 \end{aligned} \tag{2.18}$$

  The first-order approximation of $\exp(\mathbf{x})$ is

$$\exp(\mathbf{x}) \approx \mathbf{I}_3 + S(\mathbf{x}) + o(\|\mathbf{x}\|). \tag{2.19}$$

- The logarithm mapping: for $\mathbf{R} \in \mathbb{SO}(3)$

$$\log(\mathbf{R}) = S^{-1}\left(\frac{\theta(\mathbf{R}-\mathbf{R}^\intercal)}{2\sin\theta}\right) \tag{2.20}$$

  where $\theta = \arccos(\frac{\mathrm{tr}(\mathbf{R})-1}{2})$. Note that if $\theta = \pi$, the logarithm mapping is not well-defined.

- The right Jaocbian: for $\mathbf{x}(\neq \mathbf{0}) \in \mathbb{R}^3$

$$\begin{aligned} \mathrm{J}_r(\mathbf{x}) &= \mathbf{I}_3 - \frac{1-\cos(\|\mathbf{x}\|)}{\|\mathbf{x}\|^2} S(\mathbf{x}) + \frac{\|\mathbf{x}\| - \sin(\|\mathbf{x}\|)}{\|\mathbf{x}\|^3} S^2(\mathbf{x}), \\ \mathrm{J}_r(\mathbf{0}) &= \mathbf{I}_3 \end{aligned} \tag{2.21}$$

  and the inverse of right Jacobian is

$$\begin{aligned} \mathrm{J}_r^{-1}(\mathbf{x}) &= \mathbf{I}_3 + \frac{1}{2} S(\mathbf{x}) + (\frac{1}{\|\mathbf{x}\|^2} - \frac{1+\cos(\|\mathbf{x}\|)}{2\|\mathbf{x}\|\sin(\|\mathbf{x}\|)}) S^2(\mathbf{x}), \\ \mathrm{J}_r^{-1}(\mathbf{0}) &= \mathbf{I}_3, \end{aligned} \tag{2.22}$$

- The adjoint ad: for $\mathbf{R} \in \mathbb{SO}(3)$

$$\mathrm{ad}_{\mathbf{R}} = \mathbf{R} \tag{2.23}$$

In addition, the right Jacobian has the following properties: for $\mathbf{x} \in \mathbb{R}^3$,

$$S(\mathbf{x}) = \mathrm{J}_r^{-1}(\mathbf{x})(\mathbf{I}_3 - \exp(-\mathbf{x})) \tag{2.24}$$

and

$$\det(\mathrm{J}_r(\mathbf{x})) = \frac{2(1-\cos(\|\mathbf{x}\|))}{\|\mathbf{x}\|^2}. \tag{2.25}$$

The Special Euclidean Group $\mathbb{SE}(3)$

The special Euclidean group $\mathbb{SE}(3)$ is a 6-dimensional manifold, defined as

Definition 2.6.

$$\mathbb{SE}(3) := \{(\mathbf{R}, \mathbf{p}) | \mathbf{R} \in \mathbb{SO}(3), \mathbf{p} \in \mathbb{R}^3\}. \tag{2.26}$$

The associated group operation is

$$(\mathbf{R}_1, \mathbf{p}_1) \cdot (\mathbf{R}_2, \mathbf{p}_2) = (\mathbf{R}_1\mathbf{R}_2, \mathbf{R}_1\mathbf{p}_2 + \mathbf{p}_1) \tag{2.27}$$

for $(\mathbf{R}_1, \mathbf{p}_1)$ and $(\mathbf{R}_2, \mathbf{p}_2) \in \mathbb{SE}(3)$.

In robotics, an element in $\mathbb{SE}(3)$ is usually used to represent the

- **pose** of a camera/robot/IMU or a coordinate system, etc;
- **rigid body transformation**.

The Lie algebra of $\mathbb{SE}(3)$ is $\mathbf{se}(3) \cong \mathbb{R}^6$. In the following, we directly provide the functions $\exp(\cdot)$, $\log(\cdot)$, ad, J_r of $\mathbb{SE}(3)$ and $\mathbf{se}(3)$, based on the overloading functions from $\mathbb{SO}(3)$ and $\mathbf{so}(3)$.

- The exponential mapping: for $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \in \mathbb{R}^6$ ($\mathbf{x}_1$ and $\mathbf{x}_2 \in \mathbb{R}^3$)

$$\exp(\mathbf{x}) = (\exp(\mathbf{x}_1), \mathrm{J}_l(\mathbf{x}_1)\mathbf{x}_2). \tag{2.28}$$

  The first-order approximation of $\exp(\mathbf{x})$ is

$$\exp(\mathbf{x}) = (\mathbf{I}_3 + S(\mathbf{x}_1) + o(\|\mathbf{x}\|), \mathbf{x}_2 + o(\|\mathbf{x}\|)). \tag{2.29}$$

- The logarithm mapping: for $\mathbf{g} = (\mathbf{R}, \mathbf{p}) \in \mathbb{SE}(3)$

$$\log(\mathbf{g}) = \begin{bmatrix} \log(\mathbf{R}) \\ \mathrm{J}_l^{-1}(\log(\mathbf{R}))\mathbf{p} \end{bmatrix}. \tag{2.30}$$

- The right Jaocbian: for $\mathbf{x} \in \mathbb{R}^6$

$$\mathrm{J}_r(\mathbf{x}) = \begin{bmatrix} \mathrm{J}_r(\mathbf{x}_1) & \mathbf{0} \\ \mathrm{K}_r(\mathbf{x}_1, \mathbf{x}_2) & \mathrm{J}_r(\mathbf{x}_1) \end{bmatrix} \tag{2.31}$$

  and

$$\mathrm{J}_r^{-1}(\mathbf{x}) = \begin{bmatrix} \mathrm{J}_r^{-1}(\mathbf{x}_1) & \mathbf{0} \\ -\mathrm{J}_r^{-1}(\mathbf{x}_1)\mathrm{K}_r(\mathbf{x}_1, \mathbf{x}_2)\mathrm{J}_r^{-1}(\mathbf{x}_1) & \mathrm{J}_r^{-1}(\mathbf{x}_1) \end{bmatrix}, \tag{2.32}$$

where $\mathrm{K}_r(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{K}_l(-\mathbf{x}_1, -\mathbf{x}_2)$ and

$$\begin{aligned}\mathrm{K}_l(\mathbf{x}_1, \mathbf{x}_2) &= \frac{1}{2}S(\mathbf{x}_2) + \frac{\theta - \sin(\theta)}{\theta^3}(S(\mathbf{x}_1)S(\mathbf{x}_2) + S(\mathbf{x}_2)S(\mathbf{x}_1) + S(\mathbf{x}_1)S(\mathbf{x}_2)S(\mathbf{x}_1)) \\ &- \frac{1 - \theta^2/2 - \cos(\theta)}{\theta^4}(S(\mathbf{x}_1)S(\mathbf{x}_1)S(\mathbf{x}_2) + S(\mathbf{x}_2)S(\mathbf{x}_1)S(\mathbf{x}_1) - 3S(\mathbf{x}_1)S(\mathbf{x}_2)S(\mathbf{x}_1)) \\ &- \frac{1}{2}(\frac{1 - \theta^2/2 - \cos(\theta)}{\theta^4} - 3\frac{\theta - \sin(\theta) - \theta^3/6}{\theta^5})(S(\mathbf{x}_1)S(\mathbf{x}_2)S(\mathbf{x}_1)S(\mathbf{x}_1) \\ &+ S(\mathbf{x}_1)S(\mathbf{x}_1)S(\mathbf{x}_2)S(\mathbf{x}_1))\end{aligned} \tag{2.33}$$

and $\theta = \|\mathbf{x}_1\|$.

- The adjoint ad: for $\mathbf{g} = (\mathbf{R}, \mathbf{p}) \in \mathbb{SE}(3)$

$$\mathrm{ad}_{\mathbf{g}} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ S(\mathbf{p})\mathbf{R} & \mathbf{R} \end{bmatrix} \tag{2.34}$$

The Lie Group $\mathcal{G}(N)$

In the later chapters, the element in $\mathcal{G}(N)$ is used to represent the pose and the location of landmarks (expressed in the global frame). Specifically, the Lie group $\mathcal{G}(N)$ is a $6 + 3N$-dimensional manifold, defined as

Definition 2.7.

$$\mathcal{G}(N) = \{\left(\mathbf{R}, \mathbf{p}, \mathbf{f}^1, \cdots, \mathbf{f}^N\right) | \mathbf{R} \in \mathbb{SO}(3), \mathbf{p} \text{ and } \mathbf{f}^i \in \mathbb{R}^3\} \tag{2.35}$$

with the group operation:

$$\mathbf{X}_1 \cdot \mathbf{X}_2 = \left(\mathbf{R}_1\mathbf{R}_2, \mathbf{R}_1\mathbf{p}_2 + \mathbf{p}_1, \mathbf{R}_1\mathbf{f}_2^1 + \mathbf{f}_1^1, \cdots, \mathbf{R}_1\mathbf{f}_2^N + \mathbf{f}_1^N\right), \tag{2.36}$$

where $\mathbf{X}_i = \left(\mathbf{R}_i, \mathbf{p}_i, \mathbf{f}_i^1, \cdots, \mathbf{f}_i^N\right) \in \mathcal{G}(N)$ for $i = 1, 2$.

The associated Lie algebra of $\mathcal{G}(N)$ is homomorphic to $\mathbb{R}^{3N+6}$. In the following, we also provide some functions of $\mathcal{G}(N)$ based on the overloading functions from $\mathbb{SO}(3)$.

- The exponential mapping:

$$\exp(\mathbf{x}) = \big(\exp(\mathbf{x}_\theta), \mathrm{J}_l(\mathbf{x}_\theta)\mathbf{x}_p, \mathrm{J}_l(\mathbf{x}_\theta)\mathbf{x}^1, \cdots, \mathrm{J}_l(\mathbf{x}_\theta)\mathbf{x}^N\big) \tag{2.37}$$

for $\mathbf{x} = \begin{bmatrix} \mathbf{x}_\theta & \mathbf{x}_p & \mathbf{x}^1 & \cdots & \mathbf{x}^N \end{bmatrix} \in \mathbb{R}^{3N+6}$, where $\mathbf{x}_\theta$, $\mathbf{x}_p$ and $\mathbf{x}^i \in \mathbb{R}^3$ ($i = 1, \cdots, N$). The first-order approximation of $\exp(\mathbf{x})$ is

$$\exp(\mathbf{x}) = (\mathbf{I}_3 + S(\mathbf{x}_\theta) + o(\|\mathbf{x}\|), \mathbf{x}_p + o(\|\mathbf{x}\|), \mathbf{x}^1 + o(\|\mathbf{x}\|), \cdots, \mathbf{x}^N + o(\|\mathbf{x}\|)). \tag{2.38}$$

- The logarithm mapping: for $\mathbf{X} = \big(\mathbf{R}, \mathbf{p}, \mathbf{f}^1, \cdots, \mathbf{f}^N\big) \in \mathcal{G}(N)$

$$\log(\mathbf{X}) = \begin{bmatrix} \log(\mathbf{R}) \\ \mathrm{J}_l^{-1}(\log(\mathbf{R}))\mathbf{p} \\ \mathrm{J}_l^{-1}(\log(\mathbf{R}))\mathbf{f}^1 \\ \vdots \\ \mathrm{J}_l^{-1}(\log(\mathbf{R}))\mathbf{f}^N \end{bmatrix}. \tag{2.39}$$

- The right Jacobian: for $\mathbf{x} = \begin{bmatrix} \mathbf{x}_\theta & \mathbf{x}_p & \mathbf{x}^1 & \cdots & \mathbf{x}^N \end{bmatrix} \in \mathbb{R}^{3+6N}$

$$\mathrm{J}_r(\mathbf{x}) = \begin{bmatrix} \mathrm{J}_r(\mathbf{x}_\theta) & & & & \\ \mathrm{K}_r(\mathbf{x}_\theta, \mathbf{x}_p) & \mathrm{J}_r(\mathbf{x}_\theta) & & & \\ \mathrm{K}_r(\mathbf{x}_\theta, \mathbf{x}^1) & & \mathrm{J}_r(\mathbf{x}_\theta) & & \\ \vdots & & & \ddots & \\ \mathrm{K}_r(\mathbf{x}_\theta, \mathbf{x}^N) & & & & \mathrm{J}_r(\mathbf{x}_\theta) \end{bmatrix} \tag{2.40}$$

and

$$\mathrm{J}_r^{-1}(\mathbf{x}) = \begin{bmatrix} \mathrm{J}_r^{-1}(\mathbf{x}_\theta) & & & & \\ \mathrm{J}_r^{-1}(\mathbf{x}_\theta)\mathrm{K}_r(\mathbf{x}_r, \mathbf{x}_p)\mathrm{J}_r^{-1}(\mathbf{x}_\theta) & \mathrm{J}_r^{-1}(\mathbf{x}_\theta) & & & \\ \mathrm{J}_r^{-1}(\mathbf{x}_\theta)\mathrm{K}_r(\mathbf{x}_\theta, \mathbf{x}^1)\mathrm{J}_r^{-1}(\mathbf{x}_\theta) & & \mathrm{J}_r^{-1}(\mathbf{x}_\theta) & & \\ \vdots & & & \ddots & \\ \mathrm{J}_r^{-1}(\mathbf{x}_\theta)\mathrm{K}_r(\mathbf{x}_\theta, \mathbf{x}^N)\mathrm{J}_r^{-1}(\mathbf{x}_\theta) & & & & \mathrm{J}_r^{-1}(\mathbf{x}_\theta) \end{bmatrix} \tag{2.41}$$

- The adjoint operation: for $\mathbf{X} = \left(\mathbf{R}, \mathbf{p}, \mathbf{f}^1, \cdots, \mathbf{f}^N\right) \in \mathcal{G}(N)$,

$$\mathrm{ad}_{\mathbf{X}} = \begin{bmatrix} \mathbf{R} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ S(\mathbf{p})\mathbf{R} & \mathbf{R} & \ddots & & \vdots \\ S(\mathbf{f}^1)\mathbf{R} & \mathbf{0} & \mathbf{R} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ S(\mathbf{f}^N)\mathbf{R} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{R} \end{bmatrix}. \tag{2.42}$$

2.5 Optimization on Lie group

This section firstly recalls the optimization on Euclidean space and then introduces the optimization on Lie group.

Review: Optimization on Euclidean Space $\mathbb{R}^n$

Consider the optimization problem:

$$\min_{\mathbf{X} \in \mathbb{R}^n} f(\mathbf{X}), \tag{2.43}$$

where $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is a function on Euclidean space. If a new function $h_{\mathbf{X}^*}(\cdot)$ on $\mathbb{R}^n$ is defined as

$$h_{\mathbf{X}^*}(\mathbf{x}) := f(\mathbf{X}^* + \mathbf{x}) \tag{2.44}$$

for arbitrary $\mathbf{X}^* \in \mathbb{R}^n$, the problem above can be rewritten

$$\min_{\mathbf{x} \in \mathbb{R}^n} h_{\mathbf{X}^*}(\mathbf{x}) = \min_{x \in \mathbb{R}^n} f(\mathbf{X}^* + \mathbf{x}). \tag{2.45}$$

Based on the rewritten equation (2.45), Alg. 1 presents a method for this optimization problem. Note that different algorithms, e.g., Gauss-Newton, Levenberg-Marquardt and Powell's Dogleg, can be used in Alg. 1 to compute the incremental vector $\Delta\mathbf{x} \in \mathbb{R}^n$ in which $\partial h_{\mathbf{X}^*}(\mathbf{0}) = \frac{\partial h_{\mathbf{X}^*}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{0}}$ and $h_{\mathbf{X}^*}(\mathbf{0})$ are needed.

Remark 2.8. In Alg. 1, the function $h_{\mathbf{X}^*}$ changes in every loop due to the change of $\mathbf{X}^*$.

Algorithm 1: Optimization on Euclidean Space

Input: the objective function $f(\cdot)$ and the initial guess $\mathbf{X}_0 \in \mathbb{R}^n$
Output: the local minimum $\mathbf{X}^*$ for $f(\cdot)$
Process:
$\mathbf{X}^* \leftarrow \mathbf{X}_0$;
while $\mathbf{X}^*$ *does not converge* **do**
  determine the incremental vector $\Delta\mathbf{x} \in \mathbb{R}^n$ for $h_{\mathbf{X}^*}(\mathbf{x})$ at $\mathbf{x} = 0$ by using gradient descent method (GN, LM,..., etc) ;
  $\mathbf{X}^* \leftarrow \mathbf{X}^* + \Delta\mathbf{x}$;
end

Optimization on Lie Group $\mathbb{G}$

Consider the optimization problem on Lie group $\mathbb{G}$. The optimization process in Alg. 1 has to be adjusted for Lie group structure. If there exists $\oplus$: for each $\mathbf{X} \in \mathbb{G}$, $\mathbf{X} \oplus (\cdot) : \mathbb{R}^n \to \mathbb{G}$ is bijective in a neighborhood of $\mathbf{0} \in \mathbb{R}^n$. For Lie group $\mathbb{G}$, we usually choose $g(\cdot)$ as

$$\mathbf{X} \oplus \mathbf{x} = \mathbf{X} \exp(\mathbf{x}) \tag{2.46}$$

or

$$\mathbf{X} \oplus \mathbf{x} = \exp(\mathbf{x})\mathbf{X} \tag{2.47}$$

The optimization problem can be rewritten via re-parametrization (called **lifting**)

$$\min_{\mathbf{X}\in\mathbb{G}} f(\mathbf{x}) \Leftrightarrow \min_{\mathbf{x}\in\mathbb{R}^{\dim(\mathbb{G})}} h_{\mathbf{X}^*}(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^{\dim(\mathbb{G})}} f(\mathbf{X}^* \oplus \mathbf{x}) \tag{2.48}$$

where $\mathbf{X}^* \in \mathbb{G}$ is user-selected and $h_{\mathbf{X}^*}(\cdot)$ is a transformation from $\mathbb{R}^{\dim(\mathbb{G})}$ to $\mathbb{G}$, defined as

$$h_{\mathbf{X}^*}(\mathbf{x}) := f(\mathbf{X}^* \oplus \mathbf{x}). \tag{2.49}$$

Note that the re-parametrization makes optimization back to Euclidean space. Based on the reformulation (2.48), the optimization process is presented in Alg. 2.

The main idea in Alg. 2 is that the state $\mathbf{X}^*$ is represented by the element in the Lie group while the incremental vector (small change) $\Delta\mathbf{x}$ is represented by the element in the Lie algebra (homeomorphic to Euclidean space) of the Lie group.

Algorithm 2: Optimization on Lie group

Input: the objective function $f(\cdot)$ and the initial guess $\mathbf{X}_0 \in \mathbb{G}$
Output: the local minimum $\mathbf{X}^* \in \mathbb{G}$ for $f(\cdot)$
Process:
$\mathbf{X}^* \leftarrow \mathbf{X}_0$;
while $\mathbf{X}^*$ *does not converge* **do**
  determine the incremental vector $\Delta\mathbf{x} \in \mathbb{R}^{\dim(\mathbb{G})}$ for $h_{\mathbf{X}^*}(\mathbf{x})$ at $\mathbf{x} = 0$ by using gradient descent method (GN, LM,..., etc) ;
  $\mathbf{X}^* \leftarrow \mathbf{X}^* \oplus \Delta\mathbf{x}$;
end

Example: Optimization on $\mathbb{SO}(3)$

We use the optimization problem in [13] to illustrate the idea in this section, which is

$$\min_{\mathbf{R}\in\mathbb{SO}(3)} f(\mathbf{R}) = \min_{\mathbf{R}\in\mathbb{SO}(3)} \sum_{i=1}^{N} \|\mathbf{R}\mathbf{p}_i - \mathbf{q}_i\|^2, \tag{2.50}$$

where $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{q}_i \in \mathbb{R}^3$ $(i = 1, 2, 3..., N)$ are known.

Gradient descent method

According to the previous discussion, we choose $\mathbf{g}(\mathbf{x}) = \exp(\mathbf{x}) \in \mathbb{SO}(3)$ such that

$$h_{\mathbf{R}^*}(\mathbf{x}) = \sum_{i=1}^{N} \|\mathbf{R}^* \exp(\mathbf{x})\mathbf{p}_i - \mathbf{q}_i\|^2 \tag{2.51}$$

and

$$\begin{aligned}
\partial h_{\mathbf{R}^*}(\mathbf{0}) &= (\partial_{\mathbf{x}} \sum_{i=1}^{N} \|\mathbf{R}^*(\mathbf{I}_3 + S(\mathbf{x}) + o(\|\mathbf{x}\|))\mathbf{p}_i - \mathbf{q}_i\|^2)|_{\mathbf{x}=\mathbf{0}} \\
&= (-2\partial_{\mathbf{x}} \sum_{i=1}^{N} \mathbf{q}_i^T \mathbf{R}^*(S(\mathbf{x}) + o(\|\mathbf{x}\|))\mathbf{p}_i)|_{x=0} \\
&= (-2\partial_{\mathbf{x}} \sum_{i=1}^{N} \mathbf{q}_i^T \mathbf{R}^* S(\mathbf{x})\mathbf{p}_i)|_{\mathbf{x}=0} \\
&= 2(\partial_{\mathbf{x}} \sum_{i=1}^{N} \mathbf{q}_i^T \mathbf{R}^* S(\mathbf{p}_i)\mathbf{x})|_{\mathbf{x}=0} \\
&= 2\sum_{i=1}^{N} \mathbf{q}_i^T \mathbf{R}^* S(\mathbf{p}_i)
\end{aligned} \tag{2.52}$$

Alg. 3 provides the gradient descent method for the optimization problem (Eq. 2.50).

Algorithm 3: Solving Eq. 2.50 by using gradient descent method on $\mathbb{SO}(3)$

Input: the objective function $f(\cdot)$ in Eq. 2.50, the initial guess $\mathbf{R}_0 \in \mathbb{SO}(3)$ and the step size $d \in \mathbb{R}^+$

Output: the local minimum $\mathbf{R}^* \in \mathbb{SO}(3)$ for $f(\cdot)$

Process:

$\mathbf{R}^* \leftarrow \mathbf{R}_0$;

while $\mathbf{R}^*$ *does not converge* **do**

$\mathbf{n} \leftarrow \partial h^{\intercal}_{\mathbf{R}^*}(\mathbf{0})$ via (2.52) ;

$\mathbf{n} \leftarrow \frac{\mathbf{n}}{\|\mathbf{n}\|}$;

determine the incremental vector $\Delta\mathbf{x} \in \mathbb{R}^3$: $\Delta\mathbf{x} \leftarrow -d\mathbf{n}$;

$\mathbf{R}^* \leftarrow \mathbf{R}^* \exp(\Delta\mathbf{x})$;

end

2.6 Factor graph and optimization

Definition 2.9. A factor graph $G = (\mathcal{F}, \mathcal{X})$ is a bipartite graph, consisting of factor nodes in $\mathcal{F} = \{f_i\}$ and variable nodes $\mathcal{X} = \{\mathbf{x}_i\}$. The explanations of notations of factor graph are given in the following:

- $\mathcal{X}$ denotes the set of variables to be estimated;
- $\mathcal{X}_i \subseteq \mathcal{X}$ denotes the set of variable nodes connected by $f_i \in \mathcal{F}$;
- $\mathcal{F}$ denotes the set of all functions $f_i(\cdot)$;

The inference in the factor graph G refers to the optimization problem below

$$\mathcal{X}^* = \arg\max_{\mathcal{X}} \prod_i f_i(\mathcal{X}_i). \tag{2.53}$$

MAP and NLS

When

$$f_i(\mathcal{X}_i) := p(Z_i|\mathcal{X}_i) \tag{2.54}$$

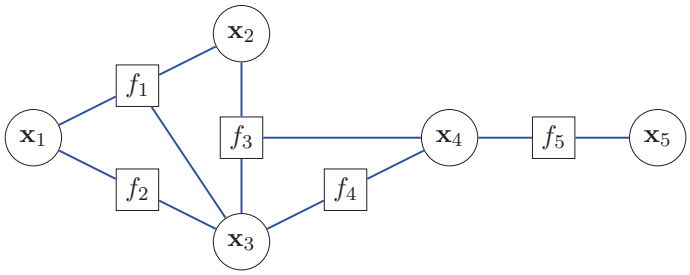

FIGURE 2.1: An example of factor graph

represents the probability density function of the measurement Z_i given $\mathcal{X}_i$, $\mathcal{X}^*$ becomes the maximum a posteriori (MAP) estimate, i.e.,

$$\mathcal{X}^* = \arg\max_{\mathcal{X}} p(\mathcal{X}|Z), \tag{2.55}$$

where $Z = \{Z_i\}$ denotes the set of all measurements.

For the Gaussian case,

$$f_i(\mathcal{X}_i) \propto \exp(-\frac{1}{2}\|h_i(\mathcal{X}_i, Z_i)\|^2_{\mathbf{\Sigma}_i^{-1}}) \tag{2.56}$$

where

$$h_i(\mathcal{X}_i, Z_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i). \tag{2.57}$$

is called the error function of the factor f_i. Under this condition, the solution of (2.53) can be computed from the nonlinear least squares optimization problem

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \sum_i \|h_i(\mathcal{X}_i)\|^2_{\mathbf{\Sigma}_i^{-1}}. \tag{2.58}$$

Therefore, the MAP estimate is the global minimum of the nonlinear least squares (NLS) optimization problem (2.58).

Variable Node and Factor Node

Variable Node

A variable node $\mathbf{x}_i \in \mathcal{X}$ is regarded as a *structure*, not required to be a column vector. Hence, we specify $\mathbf{x}_i$ a **VariableType** that is associated with a retraction $\oplus$:

$$\mathbf{x} \leftarrow \mathbf{x} \oplus \mathbf{dx} \tag{2.59}$$

where $\mathbf{dx} \in \mathbf{R}^m$ and $m = \dim(\mathbf{VariableType})$. The retraction $\oplus$ will be used in updating and computing the Jacobians.

Factor Node

As presentation in Section 2.6, for the Gaussian case, a factor node $f_i \in \mathcal{F}$ corresponds to an *error function* $h_i(\cdot, \cdot)$. Here we stress that $h_i(\mathcal{X}_i, Z_i) \in \mathbf{R}^{\dim(h_i)}$ is a column vector but the associated measurement Z_i is regarded as a *structure*. For performing optimization for (2.58), we need to calculate the following Jacobians:

$$\begin{aligned} \mathbf{H}_{i,i_1} &= \frac{\partial h_i(\mathbf{x}_{i_1} \oplus \mathbf{e}_1, \mathbf{x}_{i_2}, \cdots, \mathbf{x}_{i_K}, Z_i)}{\partial \mathbf{e_1}}|_{\mathbf{e}_1=\mathbf{0}} \\ &\vdots \\ \mathbf{H}_{i,i_K} &= \frac{\partial h_i(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \cdots, \mathbf{x}_{i_K} \oplus \mathbf{e}_K, Z_i)}{\partial \mathbf{e_K}}|_{\mathbf{e}_K=\mathbf{0}} \end{aligned} \tag{2.60}$$

where $\mathcal{X}_i = (\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \cdots, \mathbf{x}_{i_K})$. Note that the retraction $\oplus$ depends on the VariableType of $\mathbf{x}_i$.

2.7 The optimization algorithms

Gauss-Newton

Alg. 4 summarizes the Gauss-Newton algorithm for solving the nonlinear least squares problem

$$\begin{aligned}\mathcal{X}^* &= \arg\min_{\mathcal{X}} \sum_i \|h_i(\mathcal{X}_i)\|^2_{\mathbf{\Sigma}_i^{-1}} \\ &= \arg\min_{\mathcal{X}} \|f(\mathcal{X})\|^2\end{aligned} \tag{2.61}$$

Algorithm 4: Solving Eq. 2.61 by using the Gauss-Newton algorithm.

Input: the initial guess $\mathcal{X}^{(0)}$ and the retraction $\oplus$
Output: the local minimum $\mathcal{X}^*$
Process:
$\mathcal{X}^* \leftarrow \mathcal{X}^{(0)}$;
while $\mathcal{X}^*$ *does not converge* **do**
  solving the normal equation

$$\mathbf{H}\Delta\mathbf{x}_{gn} = -\mathbf{F}^\intercal f(\mathcal{X}^*) \tag{2.62}$$

  where $\mathbf{F} := \frac{\partial f(\mathcal{X}^* \oplus \mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{0}}$ and $\mathbf{H} = \mathbf{F}^\intercal\mathbf{F}$.;
  Update: $\mathcal{X}^* \leftarrow \mathcal{X}^* \oplus \Delta\mathbf{x}_{gn}$;
end

Levenberg-Marquardt and Powell's Dogleg

The Gauss-Newton algorithm can not guarantee the convengence to the local minimum even when the initial guess is close to the global minimum [14]. Therefore, Levenberg-Marquardt (LM) and Powell's Dogleg (Dogleg) are usually employed in SLAM and VINS.

In LM and Dogleg algorithms, a damping factor and backup actions to Gauss-Newton are used to control the convergence.

LM solved a damped normal equation

$$(\mathbf{H} + \lambda\mathbf{I})\Delta\mathbf{x} = -\mathbf{F}^\intercal f(\mathcal{X}^*) \tag{2.63}$$

In Dogleg, the incremental vector $\Delta\mathbf{x}$ is the linear combination of that from Gaussian-Newton and that of the deepest descend method:

$$\Delta\mathbf{x} = (1-\lambda)\Delta\mathbf{x}_{gn} + \lambda\Delta\mathbf{x}_{dp} \tag{2.64}$$

In LM and Dogleg, λ is a dynamic damping factor. At each iteration, λ would be changed: if the new error is lower than the previous one, λ would be decreased for the next iteration. Otherwise, the solution would be reverted and λ would be increased. For more details of LM and Dogleg, see [15].

2.8 Invariant system and invariant filter

This section introduce the concepts of invariant system and invariant filter, which will be used in the next Chapter.

Consider a nonlinear system

$$\begin{aligned} \mathbf{x}_{n+1} &= f_n(\mathbf{x}_n) \\ \mathbf{z}_{n+1} &= h(\mathbf{x}_{n+1}) \end{aligned} \tag{2.65}$$

where $\mathbf{x}_n \in \mathbf{S}$ represents the system state at the time step n, $\mathbf{z}_{n+1}$ represents the measurement at the time step $n+1$ and $\mathbf{S}$ is a non-Euclidean space in general. The function $f_n(\cdot)$ is the state transformation function at the time step n, i.e., a mapping from $\mathbf{S}$ to $\mathbf{S}$. The function $h(\cdot)$ is the observation function, i.e., a mapping from $\mathbf{S}$ to $\mathbf{S}'$ where $\mathbf{S}'$ is another non-Euclidean space in general.

Definition 2.10. The system (2.65) is called invariant under the state transformation $\mathcal{T} : \mathbf{S} \to \mathbf{S}$ if $\mathbf{z}_{\mathbf{x}k} = h(\mathbf{x}_k)$ is equal to $\mathbf{z}_{\mathbf{y}k} = h(\mathbf{y}_k)$ for any $k > 0$, where $\mathbf{y}_0 = \mathcal{T}(\mathbf{x}_0)$ and both $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$ follows the dynamics of the system (2.65) i.e., $\mathbf{x}_{i+1} = f_i(\mathbf{x}_i)$, $\mathbf{y}_{i+1} = f_i(\mathbf{y}_i)$ for $i = 1, 2, 3, \cdots$. Meanwhile, $\mathcal{T}$ is called an unobservable transformation to the system (2.65).

An invariant system is always born with a serious of unobservable transformation $\{\mathcal{T}\}$, which implies that some information about system state cannot be inferred by the dynamics rule and the historical measurements.

Definition 2.11. For an invariant system (2.65) with the unobservable transformations $\{\mathcal{T}\}$, a filter is called invariant under the unobservable transformations $\{\mathcal{T}\}$ if $h(\hat{\mathbf{x}}_n) = h(\hat{\mathbf{y}}_n)$ where $\hat{\mathbf{x}}_n$ is the filter's estimate at the time step n from the initial estimate $\hat{\mathbf{x}}_0$, $\hat{\mathbf{y}}_n$ is the filter's estimate at the time step n from the initial estimate $\hat{\mathbf{y}}_0$ for any $\hat{\mathbf{y}}_0$ and $\hat{\mathbf{x}}_0$: $\hat{\mathbf{y}}_0 = \mathcal{T}(\hat{\mathbf{x}}_0)$.

The invariance property of a filter can be understood as that the filter can smartly prevent spurious information. Therefore, a filter with invariance property is expected to have better performance than a filter without this property. The basic concepts above are based on deterministic systems. In the next chapter, we will generalize the concepts to a stochastic system, i.e., the SLAM system.

Chapter 3

Extended Kalman Filter in SLAM

The conventional EKF-SLAM algorithm suffers from inconsistency. Traditionally, the linearization errors are regarded as the main reason for this phenomenon. In this chapter, we deeply analyze the inconsistency and find that the absence of invariance to stochastic rigid body transformation for the conventional algorithm is the fundamental reason. We also prove that the RI-EKF algorithm can deal with the problem.

3.1 Introduction

The SLAM problem asks whether it is possible for a robot to build a map of an unknown environment and simultaneously work out its own location within the map, using information gathered from sensors mounted on the robot. Reliable solutions to SLAM underpin successful robot deployment in many application domains especially when an external location reference such as a global positioning system (GPS) is not available. In EKF-SLAM, the map is represented by a sparse set of point features, and the solution to SLAM is an estimate of the observed point feature positions and the latest robot pose, together with the associated uncertainty. EKF has been used extensively in solving the SLAM problem in the past. The state in EKF-SLAM consists of the current robot pose and all landmark locations, in contrast to optimization based algorithms where all the previous robot poses are included in the state vector. However, a limitation of the traditional

EKF based point feature SLAM is the possible estimator inconsistency. Inconsistency here refers to the fact that the algorithm underestimates the uncertainty of the estimate leading to an overconfident result. This issue was recognized as early as in 2001 [16] and then discussed in detail later in [17][18]. This is clearly a limitation of EKF SLAM and the optimization based algorithms in recent implementations have been widely used due to their good performance and the efficiency of the modern sparse solvers [19][8]. Some research to enhance the consistency of EKF SLAM is reported in the literature. Robocentric EKF SLAM [2] estimates the location of landmarks in the robot local coordinate frame. As a result landmark positions to be estimated keep changing although landmarks are stationary in a fixed global coordinate frame. It has been shown that this robot-centric formulations lead to better performance in terms of estimator consistency. Guerreiro et al. [20] also reported a Kalman filter for the SLAM problem formulated in a robocentric coordinate frame. Besides, it was shown in [1] that the inconsistency in EKF SLAM is closely related to the partial observability of SLAM problem [21][22]. This insight resulted in a number of EKF SLAM algorithms which significantly improve consistency, such as the "First Estimates Jacobian" EKF SLAM [1] or the observability-constrained EKF SLAM [23][24].

A number of authors have addressed the behaviour of EKF SLAM to examine the convergence properties and derive bounds for the uncertainty of the estimate. In 2001, Dissanayake et al. [25] proved three essential convergence properties of the algorithm under the assumption of linear motion and observation models, with theoretical achievable lower bounds on the resulting covariance matrix. In 2006, Mourikis and Roumeliotis [26] provided an analytical upper bound of the map uncertainty based on the observation noise level, the process noise level, and the size of the map. In 2007, Huang and Dissanayake [17] extended the proof of the convergence properties and the achievable lower bounds on covariance matrix in [25] to the nonlinear case, but under a restrictive assumption that the Jacobians are evaluated at the ground truth.

Recently, Lie group representation for three-dimensional orientation/pose has become popular in SLAM (e.g., [6][9]), which can achieve better convergence and accuracy for both filter based algorithms (e.g., [27][28]) and optimization based algorithms (e.g., [29][30]).

Moreover, the use of symmetry and Lie groups for observer design has gradually been recognized (e.g., [31]). The combination of the symmetry-preserving theory and EKF gives birth to the Invariant-EKF (I-EKF), which makes the traditional EKF possess the same invariance as the original system by using a geometrically adapted correction term. In [32], the I-EKF methodology is firstly applied to EKF-SLAM. And then the Right Invariant Error EKF (called "RI-EKF" in this thesis) for 2D SLAM is proposed in [33], which also intrinsically uses the Lie group representation, and the improved consistency is proven based on the linearized error-state model.

Remark 3.1. The extension of RI-EKF to 3D cases is also given in [33]. In this thesis, the adopted motion model is a bit different from that in [33].

In this chapter, we analyze the convergence and consistency properties of RI-EKF for 3D case. A convergence analysis for RI-EKF is presented without the unrealistic assumption "Jacobians evaluated at the ground truth". Furthermore, it is proven that the output of the filter is invariant under any *stochastic rigid body transformation* in contrast to $\mathbb{SO}(3)$ based EKF SLAM algorithm ($\mathbb{SO}(3)$-EKF) that is only invariant under *deterministic rigid body transformation*. Implications of these invariance properties on the consistency of the estimator are also discussed. We also discuss the relationship between these invariance properties and consistency and show that these properties have significant effect on the performance of the estimator via theoretical analysis and Monte Carlo simulations.

3.2 Problem statement

The EKF SLAM algorithms focus on estimating the current robot pose and the positions of all the observed landmarks with the given motion model and the observation model. In this work, the SLAM problem in 3D scenarios is investigated and the state to be estimated is denoted by

$$\mathbf{X} = \left(\mathbf{R}, \mathbf{p}, \mathbf{f}^1, \cdots, \mathbf{f}^N\right), \tag{3.1}$$

where $\mathbf{R} \in \mathbb{SO}(3)$ and $\mathbf{p} \in \mathbb{R}^3$ are the robot orientation and robot position, $\mathbf{f}^i \in \mathbb{R}^3$ ($i = 1, \cdots, N$) is the coordinate of the landmark i, all described in the fixed world coordinate frame.

A general propagation model for a moving robot and static landmarks in 3D scenarios can be represented by

$$\begin{aligned}\mathbf{X}_{n+1} &= f(\mathbf{X}_n, \mathbf{u}_n, \boldsymbol{\epsilon}_n) \\ &= \left(\mathbf{R}_n \exp(\mathbf{w}_n + \boldsymbol{\epsilon}_{\mathbf{n}}^w), \mathbf{p}_n + \mathbf{R}_n(\mathbf{v}_n + \boldsymbol{\epsilon}_{\mathbf{n}}^v), \mathbf{f}_n^1, \cdots, \mathbf{f}_n^N\right),\end{aligned} \tag{3.2}$$

where $\mathbf{u}_n = \begin{bmatrix} \mathbf{w}_n^\intercal & \mathbf{v}_n^\intercal \end{bmatrix}^\intercal \in \mathbb{R}^6$ is the odometry, being $\mathbf{w}_n \in \mathbb{R}^3$ and $\mathbf{v}_n \in \mathbb{R}^3$ the angular increment and linear translation from time n to time $n+1$, $\exp(\cdot)$ is the exponential mapping of $\mathbb{SO}(3)$ and $\boldsymbol{\epsilon}_n = \begin{bmatrix} (\boldsymbol{\epsilon}_{\mathbf{n}}^w)^\intercal & (\boldsymbol{\epsilon}_{\mathbf{n}}^v)^\intercal \end{bmatrix}^\intercal \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Phi}_n)$ is the odometry noise at time n.

As the robot is likely to observe different sets of landmarks in each time step, the notation O_{n+1} is used to represent the set that indicates the landmarks observed at time $n+1$. Also by assuming a 3D sensor which provides the coordinate of landmark i in the $n+1$-th robot frame, the observation model is given as follows

$$\mathbf{z}_{n+1} = h_{n+1}(\mathbf{X}_{n+1}, \boldsymbol{\xi}_{n+1}), \tag{3.3}$$

where $h_{n+1}(\mathbf{X}_{n+1}, \boldsymbol{\xi}_{n+1})$ is a column vector obtained by stacking all entries $h^i(\mathbf{X}_{n+1}, \boldsymbol{\xi}_{n+1}^i) = \mathbf{R}_{n+1}^\intercal(\mathbf{f}_{n+1}^i - \mathbf{p}_{n+1}) + \boldsymbol{\xi}_{n+1}^i \in \mathbb{R}^3$ for all $i \in \mathrm{O}_{n+1}$, $\boldsymbol{\xi}_{n+1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_{n+1})$ is the observation noise vector obtained by stacking all entries $\boldsymbol{\xi}_{n+1}^i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_{n+1}^i)$ $(i \in \mathrm{O}_{n+1})$. The covariance matrix $\boldsymbol{\Psi}_{n+1}$ of observation noise is a block diagonal matrix consisting of all $\boldsymbol{\Psi}_{n+1}^i$ $(i \in \mathrm{O}_{n+1})$.

3.3 The invariant EKF SLAM algorithm

In this section, RI-EKF based on the general EKF framework is briefly introduced. In the general EKF framework, the uncertainty of $\mathbf{X}$ is described by $\mathbf{X} = \hat{\mathbf{X}} \oplus \mathbf{e}$, where $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$ is a white Gaussian noise vector and $\hat{\mathbf{X}}$ is the mean estimate of $\mathbf{X}$. The notation $\oplus$ is commonly called retraction in differentiable geometry [34] and it is designed as a smooth mapping such that $\mathbf{X} = \mathbf{X} \oplus \mathbf{0}$ and there exists the inverse mapping $\ominus$ of $\oplus$: $\mathbf{e} = \mathbf{X} \ominus \hat{\mathbf{X}}$. The process of propagation and update based on the general EKF framework

Algorithm 5: The general EKF framework (RI-EKF)

Input: $\hat{\mathbf{X}}_n$, $\mathbf{P}_n$, $\mathbf{u}_n$, $\mathbf{z}_{n+1}$;
Output: $\hat{\mathbf{X}}_{n+1}$, $\mathbf{P}_{n+1}$;
Propagation:
$\hat{\mathbf{X}}_{n+1|n} \leftarrow f(\mathbf{X}_n, \mathbf{u}_n, \mathbf{0})$, $\mathbf{P}_{n+1|n} \leftarrow \mathbf{F}_n\mathbf{P}_n\mathbf{F}_n^\intercal + \mathbf{G}_n\Phi_n\mathbf{G}_n^\intercal$;
Update:
$\mathbf{S} \leftarrow \mathbf{H}_{n+1}\mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\intercal + \mathbf{\Psi}_{n+1}$, $\mathbf{K} \leftarrow \mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\intercal\mathbf{S}^{-1}$;
$\mathbf{y} \leftarrow h_{n+1}(\hat{\mathbf{X}}_{n+1|n}, \mathbf{0}) - \mathbf{z}_{n+1}$;
$\hat{\mathbf{X}}_{n+1} \leftarrow \hat{\mathbf{X}}_{n+1|n} \oplus \mathbf{K}\mathbf{y}$, $\mathbf{P}_{n+1} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H}_{n+1})\mathbf{P}_{n+1|n}$;

has been summarized in Alg. 5, which is very similar to the standard EKF. Due to different uncertainty representation (compared to the standard EKF), the Jacobians of the general EKF framework in Alg. 5 are obtained by: $\mathbf{F}_n = \frac{\partial f(\hat{\mathbf{X}}_n \oplus \mathbf{e}, \mathbf{u}_n, \mathbf{0}) \ominus f(\hat{\mathbf{X}}_n, \mathbf{u}_n, \mathbf{0})}{\partial \mathbf{e}}|_{\mathbf{e}=\mathbf{0}}$, $\mathbf{G}_n = \frac{\partial f(\hat{\mathbf{X}}_n, \mathbf{u}_n, \epsilon) \ominus f(\hat{\mathbf{X}}_n, \mathbf{u}_n, \mathbf{0})}{\partial \epsilon}|_{\epsilon=\mathbf{0}}$, $\mathbf{H}_{n+1} = \frac{\partial h_{n+1}(\hat{\mathbf{X}}_{n+1|n} \oplus \mathbf{e}, \mathbf{0})}{\partial \mathbf{e}}|_{\mathbf{e}=\mathbf{0}}$.

Technically, we have $\mathbf{X}_{n+1} \approx \hat{\mathbf{X}}_{n+1|n} \oplus \mathbf{e}$, where $\mathbf{e} \sim \mathcal{N}(\mathbf{K}\mathbf{y}, \mathbf{P}_{n+1})$ after the update stage, which results in $\mathbf{X}_{n+1} \approx (\hat{\mathbf{X}}_{n+1|n} \oplus \mathbf{K}\mathbf{y}) \oplus \mathbf{e}_r$ and $\mathbf{e}_r \sim \mathcal{N}(\mathbf{0}, \mathbf{J}_{n+1}\mathbf{P}_{n+1}\mathbf{J}_{n+1}^\intercal)$, where $\mathbf{J}_{n+1} = \frac{\partial((\hat{\mathbf{X}}_{n+1|n} \oplus \mathbf{K}\mathbf{y}) \ominus (\hat{\mathbf{X}}_{n+1|n} \oplus \mathbf{e}))}{\partial \mathbf{e}}|_{\mathbf{e}=\mathbf{K}\mathbf{y}}$. Hence, we should have one step $\mathbf{P}_{n+1} \leftarrow \mathbf{J}_{n+1}\mathbf{P}_{n+1}\mathbf{J}_{n+1}^\intercal$. However, we find out that the use of the step $\mathbf{P}_{n+1} \leftarrow \mathbf{J}_{n+1}\mathbf{P}_{n+1}\mathbf{J}_{n+1}^\intercal$ affects the stochastic invariance property (Section 3.5) and have negative contributions to the performance in terms of both accuracy and consistency. Thus this step is turned off.

Remark 3.2. In the conventional EKF, $\mathbf{J}$ is identity matrix due to the adopted error (in the Euclidean space).

RI-EKF

RI-EKF follows the general EKF framework summarized in Alg. 5. The state space of RI-EKF is modeled as a Lie group $\mathcal{G}(N)$. The background knowledge about Lie group $\mathcal{G}(N)$ is provided in Chapter 2.

The choice of $\oplus$

The retraction $\oplus$ of RI-EKF [35] is chosen such that

$$\begin{aligned}\mathbf{X} = \hat{\mathbf{X}} \oplus \mathbf{e} &:= \exp(\mathbf{e})\hat{\mathbf{X}} \\ &= (\exp(\mathbf{e}_\theta)\mathbf{R}, \exp(\mathbf{e}_\theta)\mathbf{p} + J_l(\mathbf{e}_\theta)\mathbf{e}_p, \\ &\exp(\mathbf{e}_\theta)\mathbf{f}_1 + J_l(\mathbf{e}_\theta)\mathbf{e}^1, \cdots, \exp(\mathbf{e}_\theta)\mathbf{f}_N + J_l(\mathbf{e}_\theta)\mathbf{e}^N)\end{aligned} \tag{3.4}$$

where exp is the exponential mapping on the Lie group $\mathcal{G}(N)$. More details and the Matlab code of the algorithms are available at *https* : *//github.com/RomaTeng/EKF* − *SLAM* − *on* − *Manifold*. $\mathbf{X} \in \mathcal{G}(N)$ is the actual pose and landmarks, $\hat{\mathbf{X}} \in \mathcal{G}(N)$ is the *mean* estimate and the uncertainty vector $\mathbf{e} = \begin{bmatrix} \mathbf{e}_\theta^\intercal & \mathbf{e}_p^\intercal & (\mathbf{e}^1)^\intercal & \cdots & (\mathbf{e}^N)^\intercal \end{bmatrix}^\intercal \in \mathbf{R}^{3N+6}$ follows the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{P})$.

Remark 3.3. In contrast to RIEKF-SLAM in [33], we use the uncertainty representation instead of the error-form to present the new filter' differences from the conventional EKF. In this way, the retraction $\oplus$ will be naturally employed in the update stage.

Jacobian matrices

The Jacobians of the propagation step of RI-EKF are

$$\mathbf{F}_n = \mathbf{I}_{3N+6}, \ \mathbf{G}_n = ad_{\hat{\mathbf{X}}_n}\mathbf{B}_n, \tag{3.5}$$

where $\mathbf{B}_n = \begin{bmatrix} -J_r(-\mathbf{w}_n) & \mathbf{0}_{3,3} \\ -S(\mathbf{v}_n)J_r(-\mathbf{w}_n) & \mathbf{I}_3 \\ \mathbf{0}_{3N,3} & \mathbf{0}_{3N,3} \end{bmatrix}$. The adjoint operation ad and the right Jacobian $J_r(\cdot)$ are given in Chapter 1. The Jacobian matrix $\mathbf{H}_{n+1}$ of the update step is obtained by stacking all matrices $\mathbf{H}_{n+1}^i$ for all $i \in \mathrm{O}_{n+1}$, where

$$\mathbf{H}_{n+1}^i = \begin{bmatrix} \mathbf{0}_{3,3} & \hat{\mathbf{R}}_{n+1|n}^\intercal & \cdots & -\hat{\mathbf{R}}_{n+1|n}^\intercal & \mathbf{0}_{3,3(N-i)} \end{bmatrix}. \tag{3.6}$$

For a general observation model that is a function of the relative position of the landmark, the Jacobian matrix $\mathbf{H}_{n+1}$ can be calculated by the chain rule.

Landmark initialization

Here we provide the method to augment the state $\mathbf{X} \in \mathcal{G}(N)$ and adjust the covariance matrix $\mathbf{P}$ when the robot observes a new landmark with the observation $\mathbf{z} \in \mathbf{R}^3$. For brevity, the mathematical derivation is ignored here and the process to augment the state is summarized in Alg. 6, where $\mathbf{M}_N := \begin{bmatrix} \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3N} \end{bmatrix}^\intercal$ and Ψ is the covariance matrix representing the noise level in the new landmark observation.

Algorithm 6: Landmark Initialization of RI-EKF

Input: $\hat{\mathbf{X}}$, $\mathbf{P}$, $\mathbf{z}$;
Output: $\hat{\mathbf{X}}_{new}$, $\mathbf{P}_{new}$;
Process:
$\hat{\mathbf{f}}^{N+1} = \hat{\mathbf{p}} + \hat{\mathbf{R}}\mathbf{z} \in \mathbb{R}^3$
$\hat{\mathbf{X}}_{new} \leftarrow (\hat{\mathbf{X}}, \hat{\mathbf{f}}^{N+1}) \in \mathcal{G}(N+1)$;
$\mathbf{P}_{new} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P}\mathbf{M}_N \\ \mathbf{M}_N^\intercal \mathbf{P} & \hat{\mathbf{R}}\Psi\hat{\mathbf{R}}^\intercal + \mathbf{M}_N^\intercal \mathbf{P}\mathbf{M}_N \end{bmatrix}$.

Discussion

The general EKF framework proposed in [33] allows more flexible uncertainty representation, compared to the standard EKF. From Alg. 6, one can see that **a general EKF framework based filter** can be designed via **a choice of retraction** $\oplus$. For example, the retraction $\oplus$ used in the 2D traditional EKF SLAM algorithm is the standard addition. A natural extension of the 2D traditional EKF SLAM algorithm is $\mathbb{SO}(3)$-EKF, in which the state space is modeled as $\mathbb{SO}(3) \times \mathbb{R}^{3+3N}$ and the retraction $\oplus$ is $\mathbf{X} = \hat{\mathbf{X}} \oplus \mathbf{e} = (\exp(\mathbf{e}_\theta)\hat{\mathbf{R}}, \mathbf{e}_p + \hat{\mathbf{p}}, \mathbf{e}^1 + \hat{\mathbf{f}}^1, \cdots, \mathbf{e}^N + \hat{\mathbf{f}}^N)$. Similarly, $\mathbb{SE}(3)$-EKF can be obtained by modeling the state space as $\mathbb{SE}(3) \times \mathbb{R}^{3N}$.

Another noticeable point is that two general EKF framework based filters with different choice of $\oplus$ may have the same Jacobians $(\mathbf{F}_n, \mathbf{G}_n, \mathbf{H}_n)$. For example, if the retraction $\oplus$ of RI-EKF is changed such that $\hat{\mathbf{X}} \oplus \mathbf{e} = (\exp(\mathbf{e}_\theta)\hat{\mathbf{R}}, \exp(\mathbf{e}_\theta)\hat{\mathbf{p}} + \mathbf{e}_p, \cdots, \exp(\mathbf{e}_\theta)\hat{\mathbf{f}}^N + \mathbf{e}^N)$, the resulting filter (Pseudo-RI-EKF) has the same Jacobians as that of RI-EKF but their performances are significantly different, showing that the choice of retraction $\oplus$ has significant effect on the behavior of the general EKF framework based filter. In the next

section, we will compare the behavior of RI-EKF with $\mathbb{SO}(3)$-EKF via the theoretical proofs for the convergence property of RI-EKF and two simple examples.

3.4 Convergence analysis of RI-EKF SLAM algorithm

The general expression for the covariance matrices evolution of RI-EKF cannot be easily obtained. Therefore, two representative scenarios are considered: (i) the robot is stationary, and (ii) the robot then moves one step. The convergence results of RI-EKF SLAM algorithm in the two scenarios are presented and proven, under the condition that Jacobians are evaluated at the latest estimate. Hence the results are significant extension to similar theorems in [17] where Jacobians evaluated at the true state are assumed to be available.

The general setting analyzed in the following subsections is as follows. The robot starts at point A with the initial condition $(\hat{\mathbf{X}}_0, \mathbf{P})$, where $\mathbf{P}$ is covariance matrix and $\hat{\mathbf{X}}_0 = (\hat{\mathbf{R}}, \hat{\mathbf{p}}, \hat{\mathbf{f}}^1, \cdots, \hat{\mathbf{f}}^N)$ (N landmarks have been observed). The covariance matrix of odometry noise is $\mathbf{\Phi}$ and the covariance matrix of one landmark observation noise is Ψ. In the following subsections, $\mathbf{M}_N := \begin{bmatrix} \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3N} \end{bmatrix}^\intercal$, $\mathbf{L} := \mathbf{P}\mathbf{M}_N$ and $\mathbf{W} := \mathbf{M}_N^\intercal \mathbf{P}\mathbf{M}_N$. The odometry and the covariance of odometry noise are $\mathbf{0}_{6,1}$ and $\mathbf{0}_{6,6}$, respectively when robot remains stationary.

Scenario A: Robot remains stationary

Theorem 3.4. *If the robot remains stationary at point A and does not observe any of the previously seen landmarks but observes a new landmark for k times, the mean estimate of robot pose and the N landmarks and their related uncertainty remain unchanged (via k times process of propagation and update of RI-EKF). The covariance matrix of the state when the new landmark is integrated becomes* $\mathbf{P}_k = \begin{bmatrix} \mathbf{P} & \mathbf{L} \\ \mathbf{L}^\intercal & \frac{\hat{\mathbf{R}}\mathbf{\Psi}\hat{\mathbf{R}}^\intercal}{k} + \mathbf{W} \end{bmatrix}$. *When* $k \to \infty$, *the covariance matrix becomes*

$$\mathbf{P}_\infty^A = \begin{bmatrix} \mathbf{P} & \mathbf{L} \\ \mathbf{L}^\intercal & \mathbf{W} \end{bmatrix}. \tag{3.7}$$

Proof. See Appendix A.1. □

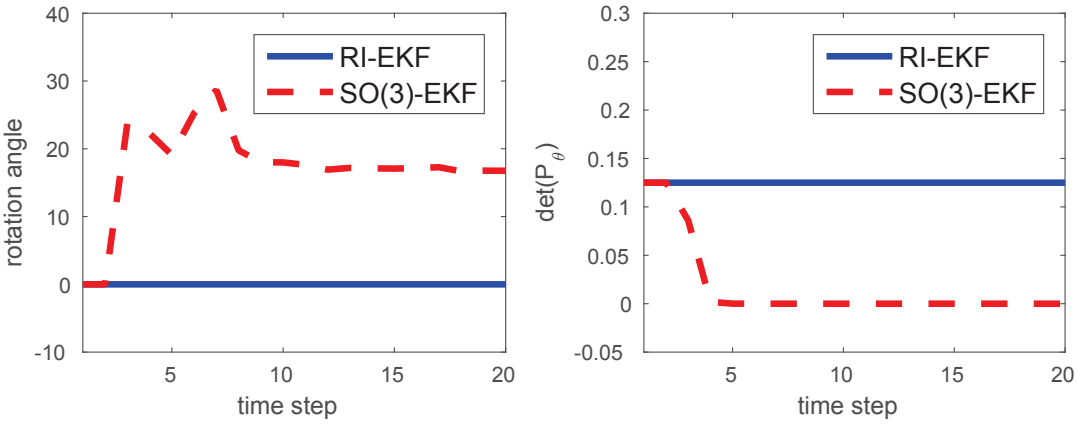

FIGURE 3.1: **Illustration of Theorem 3.4**. In this case, robot is stationary and always only observes the "new" landmark. Left: The error (unit: degree) in robot orientation w.r.t. the ground truth as estimated by RI-EKF and $\mathbb{SO}(3)$-EKF. Right: $\det(\mathbf{P}_\theta)$ estimated by RI-EKF and $\mathbb{SO}(3)$-EKF.

Theorem 3.1 can be interpreted as that the covariance matrix (w.r.t. robot pose) in RI-EKF will not be reduced by observing the "new" landmark when robot keeps stationary, which corresponds to a fact that the observations to previously unseen landmark do not convey any new information on the location of the robot. However, $\mathbb{SO}(3)$-EKF does not have this good convergence property.

We illustrate the results of Theorem 3.1 using the following scenario. The simulated robot remains stationary and always observes the "new" landmark (the covariance of observation noise is not zero). The "new" landmark is observed multiple times (a small loop closure) and the standard deviation of observation noise is set as 5% of robot-to-landmark distance along each axis. The initial covariance matrix $\mathbf{P}_\theta \in \mathbb{R}^{3\times 3}$ of robot orientation is set as $\frac{1}{2}\mathbf{I}_3$. Fig. 3.1 presents results of a simulation of this scenario. The rotation angle relative to the initial orientation and $\det(\mathbf{P}_\theta)$ from RI-EKF correctly infers that the robot remains stationary and the orientation uncertainty remains unchanged. In contrast, $\mathbb{SO}(3)$-EKF updates the robot orientation and furthermore predicts that the orientation uncertainty decreases as observations continue to be made, both of which are clearly erroneous and therefore leads to estimator inconsistency.

Theorem 3.1 can be easily extended to a multiple landmarks scenario.

Corollary 3.5. *If the robot is stationary at point A and only observes m new landmark k times, the estimate of pose from RI-EKF does not change while the covariance matrix of the estimate becomes* $\mathbf{P}_k = \begin{bmatrix} \mathbf{P} & \mathbf{L} & \mathbf{L} & \cdots & \mathbf{L} \\ \mathbf{L}^\intercal & \mathbf{Q}_k & \mathbf{W} & \cdots & \mathbf{W} \\ \mathbf{L}^\intercal & \mathbf{W} & \mathbf{Q}_k & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{W} \\ \mathbf{L}^\intercal & \mathbf{W} & \cdots & \mathbf{W} & \mathbf{Q}_k \end{bmatrix}$, *where* $\mathbf{Q}_k = \frac{\hat{\mathbf{R}}\boldsymbol{\Psi}\hat{\mathbf{R}}^\intercal}{k} + \mathbf{W}$. *When* $k \to \infty$, *the covariance matrix becomes*

$$\mathbf{P}_\infty^A = \begin{bmatrix} \mathbf{P} & \mathbf{L} & \mathbf{L} & \cdots & \mathbf{L} \\ \mathbf{L}^\intercal & \mathbf{W} & \mathbf{W} & \cdots & \mathbf{W} \\ \mathbf{L}^\intercal & \mathbf{W} & \mathbf{W} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{W} \\ \mathbf{L}^\intercal & \mathbf{W} & \cdots & \mathbf{W} & \mathbf{W} \end{bmatrix}. \tag{3.8}$$

Scenario B: Robot takes a step after a stationary period

Consider the condition that the robot moves one step after being stationary for a long period of time while observing new landmarks.

Theorem 3.6. *Assume* $\Psi = \phi\mathbf{I}_3$ *(*$\phi \in \mathbb{R}^+$*). If the robot remains stationary at point A, does not observe any of the previously seen landmarks but observes m new landmarks for* $k = \infty$ *times and then takes a step to B using control action* $\mathbf{u} = \begin{bmatrix} \mathbf{w}^\intercal & \mathbf{v}^\intercal \end{bmatrix}^\intercal$ *and observes the same set of landmarks l times, then the covariance matrix from RI-EKF becomes* $\mathbf{P}_l^B = \mathbf{P}_\infty^A + \bar{\mathbf{P}}_l^B$, *where* $\mathbf{P}_\infty^A$ *is given in (3.8),* $\bar{\boldsymbol{\Psi}} = \phi\mathbf{I}_{3m}$ *and*

$$\bar{\mathbf{P}}_l^B = ad_{\hat{\mathbf{X}}_A}\mathbf{E}(\tilde{\boldsymbol{\Phi}}^{-1} + l\tilde{\mathbf{H}}^\intercal\bar{\boldsymbol{\Psi}}^{-1}\tilde{\mathbf{H}})^{-1}\mathbf{E}^\intercal ad_{\hat{\mathbf{X}}_A}^\intercal, \tag{3.9}$$

where $\bar{\mathbf{\Psi}} = \phi \mathbf{I}_{3m}$ *and the covariance matrix of the odometry noise is* $\mathbf{\Phi}$*. In (3.9),* $\hat{\mathbf{X}}_A$ *is the estimated state at the point A,* $\tilde{\mathbf{\Phi}} = \mathbf{B}\mathbf{\Phi}\mathbf{B}^\intercal$ *is a positive definite matrix and*

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} -J_r(-\mathbf{w}) & \mathbf{0}_{3,3} \\ -S(\mathbf{v})J_r(-\mathbf{w}) & \mathbf{I}_3 \end{bmatrix}, \\ \mathbf{E} &= \begin{bmatrix} \mathbf{I}_6 \\ \mathbf{0}_{3(N+m),6} \end{bmatrix}, \ \tilde{\mathbf{H}} = \mathbf{H} ad_{\hat{\mathbf{X}}_A} \mathbf{E}, \end{aligned} \tag{3.10}$$

where $\mathbf{H}$ *is obtained by stacking all matrices* $\mathbf{H}^i = \begin{bmatrix} \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3(N+i-1)} & -\mathbf{I}_3 & \mathbf{0}_{3,3(m-i)} \end{bmatrix}$*. When* l *tends to infinity, the covariance matrix becomes* $\mathbf{P}^B_\infty = \mathbf{P}^A_\infty$ *under the condition that there are three landmarks non-coplanar with the robot position.*

Proof. See Appendix A.2. □

From Theorem 3.3, one can see that the estimate of RI-EKF follows the expectation that "the only effect of the observations made at point B is to reduce the additional robot uncertainty generated from the odometry noise. The observations made at point B cannot reduce the uncertainty of the landmark further if the robot had already observed the landmark many times at point A." [17]

We illustrate the results of Theorem 3.3 using the following scenario. Initially the robot is stationary at point A and continually observes ten previously unseen landmarks. It moves one step to point B after 200 such observations and then remains stationary for 200 more time steps while observing the same set of landmarks. The initial covariance matrix of robot pose is set as non-zero. In Fig. 3.2, we adopt $\log(\det(\mathbf{P}_r))$ as the extent of the uncertainty w.r.t. robot pose, where $\mathbf{P}_r \in \mathbb{R}^{6\times 6}$ is the covariance matrix of the robot pose. In Fig. 3.2, the pose uncertainty from RI-EKF remains unchanged and increases at time 200 when robot moves one step due to odometry noise as expected. Further landmark observations at point B while remaining stationary gradually reduce the pose uncertainty. In contrast, the pose uncertainty from $\mathbb{SO}(3)$-EKF falls below the initial value indicating incorrect injection of information, leading to an overconfident estimate of uncertainty.

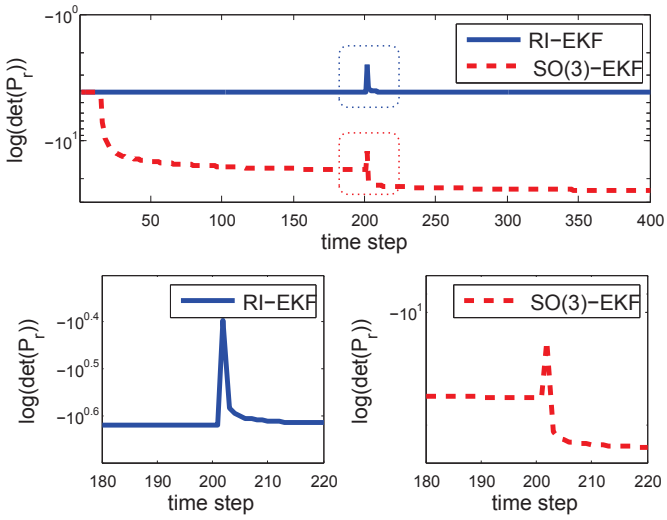

FIGURE 3.2: **Illustration of Theorem 3.6**. The y-axis is $\log(\det(\mathbf{P}_r))$ that represents the pose uncertainty. $\mathbf{P}_r$ is the covariance matrix of robot pose. Robot remains stationary from time 1 to time 200, moves one step at time 200 and then remains stationary.

3.5 Consistency analysis

As seen in the previous section, RI-EKF SLAM algorithm meets the expectation that observing new landmarks should not help in reducing the robot pose uncertainty [18][20], while $\mathbb{SO}(3)$-EKF contradicts this. This section further investigates the reason for the phenomenon above.

Unobservability and invariance property

This subsection first reviews the unobservability of SLAM formulation (3.1)–(3.3), which is strongly related to the consistency issues of various SLAM estimation algorithms. The earliest concept of observability for nonlinear systems is proposed in [36]. From the viewpoint of nonlinear systems, the SLAM formulation (as a system for the actual state $\mathbf{X}$) is not locally observable [36], as understood in [23][37]. In the following, we will mathematically describe the unobservability of SLAM formulation (3.1)–(3.3) in terms of stochastic rigid body transformation.

Definition 3.7. For SLAM problem formulation (3.1)–(3.3), a stochastic rigid body transformation $\mathcal{T}_{\mathbf{g}}$ is

$$\begin{aligned}\mathcal{T}_{\mathbf{g}}(\mathbf{X}) = (&\exp(\Theta_1)\bar{\mathbf{R}}\mathbf{R}, \exp(\Theta_1)\bar{\mathbf{R}}\mathbf{p} + \bar{\mathbf{T}} + \Theta_2, \\ &\exp(\Theta_1)\bar{\mathbf{R}}\mathbf{f}^1 + \bar{\mathbf{T}} + \Theta_2, \cdots, \exp(\Theta_1)\bar{\mathbf{R}}\mathbf{f}^N + \bar{\mathbf{T}} + \Theta_2),\end{aligned} \tag{3.11}$$

where $\mathbf{X}$ is given in (3.1), $\mathbf{g} = (\bar{\mathbf{R}}, \bar{\mathbf{T}}, \Theta)$, $\bar{\mathbf{R}} \in \mathbb{SO}(3), \bar{\mathbf{T}} \in \mathbb{R}^3$ and $\Theta = \begin{bmatrix} \Theta_1^\intercal & \Theta_2^\intercal \end{bmatrix}^\intercal \in \mathbb{R}^6$ is white Gaussian noise with covariance $\bar{\mathbf{\Sigma}}$. When the covariance $\bar{\mathbf{\Sigma}} = \mathbf{0}_{6,6}$, this transformation degenerates into a **deterministic rigid body transformation**. When $\mathbf{g} = (\mathbf{I}_3, \mathbf{0}_{3,1}, \Theta)$, this transformation degenerates into a **stochastic identity transformation**.

It can be easily verified that **the output (observations) of the system** (3.1)–(3.3) is invariant to any stochastic rigid body transformation $\mathcal{T}_{\mathbf{g}}$: for any two initial conditions, $\mathbf{X}_0$ and $\mathbf{Y}_0 := \mathcal{T}_{\mathbf{g}}(\mathbf{X}_0)$, we have $h_n(\mathbf{X}_n, \boldsymbol{\xi}_n) = h_n(\mathbf{Y}_n, \boldsymbol{\xi}_n)$ for all $n \geq 0$, where $\mathbf{X}_k = f(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\epsilon}_{k-1})$ and $\mathbf{Y}_k = f(\mathbf{Y}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\epsilon}_{k-1})$ $(k = 1, \cdots, n-1)$. Therefore, SLAM formulation (3.1)–(3.3) is *unobservable* in terms of stochastic rigid body transformation. In the following, the invariance to stochastic rigid body transformation for the EKF framework based filter of the SLAM formulation will be mathematically described.

Definition 3.8. The output (**estimated observations**) of a general EKF framework based filter is invariant under any stochastic rigid body transformation $\mathcal{T}_{\mathbf{g}}$ if for any two initial estimates $(\hat{\mathbf{X}}_0, \mathbf{P}_0)$ and $(\hat{\mathbf{Y}}_0, \mathbf{P}y_0)$, where $\hat{\mathbf{Y}}_0 = \mathcal{T}_{\mathbf{g}}(\hat{\mathbf{X}}_0)$ and $\mathbf{P}y_0 = \bar{\mathbf{Q}}_1\mathbf{P}_0\bar{\mathbf{Q}}_1^\intercal + \bar{\mathbf{Q}}_2\bar{\mathbf{\Sigma}}\bar{\mathbf{Q}}_2^\intercal$ in which

$$\begin{aligned}\bar{\mathbf{Q}}_1 &= \left.\frac{\partial \mathcal{T}_{\hat{\mathbf{g}}}(\hat{\mathbf{X}}_0 \oplus \mathbf{e}) \ominus \mathcal{T}_{\hat{\mathbf{g}}}(\hat{\mathbf{X}}_0)}{\partial \mathbf{e}}\right|_{\mathbf{e}=\mathbf{0}}, \\ \bar{\mathbf{Q}}_2 &= \left.\frac{\partial \mathcal{T}_{\mathbf{g}}(\hat{\mathbf{X}}_0) \ominus \mathcal{T}_{\hat{\mathbf{g}}}(\hat{\mathbf{X}}_0)}{\partial \Theta}\right|_{\Theta=\mathbf{0}},\end{aligned} \tag{3.12}$$

and $\hat{\mathbf{g}} = (\bar{\mathbf{R}}, \bar{\mathbf{T}}, \mathbf{0})$, we have $h_n(\hat{\mathbf{X}}_n, \mathbf{0}) = h_n(\hat{\mathbf{Y}}_n, \mathbf{0})$ for all $n > 0$. The notations $\hat{\mathbf{X}}_n$ and $\hat{\mathbf{Y}}_n$ above represent the *mean* estimate of this filter at time n by using the same input (odometry and observations) from time 0 to n, from the initial conditions $(\hat{\mathbf{X}}_0, \mathbf{P}_0)$ and $(\hat{\mathbf{Y}}_0, \mathbf{P}y_0)$, respectively.

As shown in Def. 3.4 and Def. 3.5, the invariance to stochastic rigid body transformation can be divided into two properties: 1) **the invariance to deterministic rigid body**

transformation and 2) **the invariance to stochastic identity transformation**. The results about the invariance of some general EKF framework based filters are summarized in Theorem 3.6 and TABLE 3.1.

Theorem 3.9. *The output of RI-EKF is invariant under stochastic rigid body transformation. The output of* $\mathbb{SO}(3)$*-EKF is only invariant under deterministic rigid body transformation. The output of Pseudo-RI-EKF is only invariant under stochastic identity transformation.*

Proof. See Appendix A.3. □

Remark 3.10. From the proof of Theorem 3.6, one can see that the uncertainty representation of RI-EKF can be linearly (seamlessly) transformed under a deterministic rigid body transformation, which makes RI-EKF invariant under deterministic rigid body transformation. In addition, we also show that the invariance property to stochastic identity transformation directly depends on the Jacobians $\mathbf{F}_n$ and $\mathbf{H}_n$.

TABLE 3.1: **The invariance property of the general EKF framework based filters.** DRBT/SRBT is short for "deterministic/stochastic rigid body transformation" and SIT is short for "stochastic identity transformation".

Filters	DRBT	SIT	SRBT
RI-EKF	Yes	Yes	Yes
Pseudo-RI-EKF	No	Yes	No
$\mathbb{SO}(3)$-EKF	Yes	No	No

Consistency and invariance

The unobservability in terms of stochastic rigid body transformation is a fundamental property of SLAM formulation. Therefore a consistent filter (as a system for the estimated state $\hat{\mathbf{X}}$) should maintain this unobservability, i.e., **the (estimated) output of the estimator is invariant under any stochastic rigid body transformation.** Essentially speaking, the invariance to deterministic rigid body transformation can be interpreted that the estimate does *not* depend on the selection of the global frame and the invariance to stochastic identity transformation can be understood that the uncertainty w.r.t the global frame does not affect the estimate. Hence, **consistency for the general EKF framework based filter is tightly coupled with the invariance to**

stochastic rigid body transformation. If a filter does not have this property, then unexpected information will be generated by the selection of the global frame or the uncertainty w.r.t. the global frame, which results in inconsistency (overconfidence). One can see that $\mathbb{SO}(3)$-EKF, not invariant to stochastic identity transformation, produces clearly illogical estimate (the pose uncertainty is reduced by the new landmarks) in the two cases of Section 3.4 while RI-EKF, invariant to stochastic rigid body transformation, produces the expected estimate.

Remark 3.11. In [1] [23], a framework for designing an observability constrained filter is proposed. The keypoint of the observability constrained filter is evaluating the Jacobians $\mathbf{F}_i$ and $\mathbf{H}_i$ ($i \geq 0$) at some selected points (instead of the latest estimate). such that the right nullspace of the local observability matrix $\mathbf{M}_{0:\infty} := [\mathbf{H}_0^\intercal \ \ (\mathbf{H}_1\mathbf{F}_0)^\intercal \ \ (\mathbf{H}_2\mathbf{F}_1)^\intercal \ \ \dots]^\intercal$ is of rank 3, corresponding to 3 DOFs of the rigid body transformation in 2D. In this way, the output of the filter would be invariant under the stochastic identity transformation (interested readers can see a similar proof in Theorem 4.8). On the other hand, this filter models the state space as $\mathbb{SO}(2) \times \mathbb{R}^{2+2N}$ and hence the output is invariant under deterministic rigid body transformation (see the property of $\mathbb{SO}(3)$-EKF shown in Theorem 3.6). Finally, the resulting filter indeed has the invariance property to stochastic rigid body transformation. However, FEJ-EKF and OC-EKF does not evaluate the Jacobians at the latest estimates and thus their convergence property may be not comparable to that of RI-EKF.

Remark 3.12. In [33] the observability analysis is performed on the linearized error-state model from the viewpoint of information matrix. Our insight is in a different viewpoint that an estimator should mimic the unobservability (to stochastic rigid transformation) of the original system, which makes our analysis more intuitive and general. In addition, our analysis also links probabilistic framework and control theory, which is missed in [35] and [23].

3.6 Simulation results

In order to validate the theoretical results, we perform Monte Carlo simulations and compare RI-EKF to $\mathbb{SO}(3)$-EKF, Robotcentric-EKF, the First Estimates Jacobian EKF SLAM

algorithm (FEJ-EKF), Pseudo-RI-EKF and $\mathbb{SE}(3)$-EKF under conditions of different noise levels. The original Robocentric-EKF and FEJ-EKF are proposed in 2D SLAM. For comparison, we extend these into 3D.

- $\mathbb{SO}(3)$-EKF uses the retraction

$$\mathbf{X} = \hat{\mathbf{X}} \oplus \mathbf{e} = (\exp(\mathbf{e}_\theta)\hat{\mathbf{R}}, \mathbf{e}_p + \hat{\mathbf{p}}, \mathbf{e}^1 + \hat{\mathbf{f}}^1, \cdots, \mathbf{e}^N + \hat{\mathbf{f}}^N) \tag{3.13}$$

- Presenting Robotcentric-EKF in the world frame, the used retraction in Robotcentric-EKF is equivalent to

$$\mathbf{X} = \hat{\mathbf{X}} \oplus \mathbf{e} = (\exp(\mathbf{e}_\theta)\hat{\mathbf{R}}, \mathbf{e}_p + \hat{\mathbf{p}}, \hat{\mathbf{R}}\mathbf{e}^1 + \hat{\mathbf{f}}^1, \cdots, \hat{\mathbf{R}}\mathbf{e}^N + \hat{\mathbf{f}}^N) \tag{3.14}$$

- Pseudo-RI-EKF uses the retraction

$$\mathbf{X} = \hat{\mathbf{X}} \oplus \mathbf{e} = (\exp(\mathbf{e}_\theta)\mathbf{R}, \exp(\mathbf{e}_\theta)\mathbf{p} + \mathbf{e}_p, \exp(\mathbf{e}_\theta)\mathbf{f}_1 + \mathbf{e}^1, \cdots, \exp(\mathbf{e}_\theta)\mathbf{f}_N + \mathbf{e}^N) \tag{3.15}$$

  Note that Pseudo-RI-EKF has the same Jacobians of RI-EKF when their Jacobians are evaluated at the same point.

- $\mathbb{SE}(3)$-EKF uses the retraction

$$\mathbf{X} = \hat{\mathbf{X}} \oplus \mathbf{e} = (\exp(\mathbf{e}_\theta)\hat{\mathbf{R}}, \mathbf{e}_p + J_l(\mathbf{e}_\theta)\hat{\mathbf{p}}, \mathbf{e}^1 + \hat{\mathbf{f}}^1, \cdots, \mathbf{e}^N + \hat{\mathbf{f}}^N) \tag{3.16}$$

Settings

Consider that a robot moves in a trajectory (contained in a $50m \times 40m \times 20m$ cubic) which allows sufficient 6-DOFs motion. In this environment, 300 landmarks are randomly generated around the specified robot trajectory. The observations and odometry with noises are randomly generated by this specific trajectory and the simulated robot always observes the landmarks in the sensor range (less than 20m and 120°FoV). In every simulation, the number of steps is 500 (about 8 loops), the landmarks are incrementally added into the state vector and the initial covariance of robot is set as zero matrix. For each

condition (different noise level), 100 Monte Carlo simulations are performed. The simulation results are summarized in Table 3.2, where σ_{od} is the odometry noise level and σ_{ob} is the observation noise level such that the covariance matrices of odometry and observation and is $\mathbf{\Phi}_n = \sigma_{od}^2 \text{diag}(|\mathbf{u}_{n,1}|^2, \cdots, |\mathbf{u}_{n,6}|^2)$ and $\mathbf{\Psi}_n^i = \sigma_{ob}^2 \text{diag}(|\mathbf{Z}_{n,1}^i|^2, \cdots, |\mathbf{Z}_{n,3}^i|^2)$, where $\mathbf{Z}_n^i = [\mathbf{Z}_{n,1}^i, \mathbf{Z}_{n,2}^i, \mathbf{Z}_{n,3}^i]^\intercal = \mathbf{R}_n^\intercal(\mathbf{f}^i - \mathbf{p}_n)$ is the ground truth of the coordinates of landmark i relative to the robot pose n. The root mean square (RMS) error and the average normalized estimation error squared (NEES) are used to evaluate accuracy and consistency, respectively.

Results and analysis

As shown in Table 3.2, the performance of Pseudo-RI-EKF is also poor performing. These results can be understood that Pseudo-RI-EKF is not invariant under deterministic rigid body transformation, which are proven in Theorem 3.6. $\mathbb{SO}(3)$-EKF, not invariant to stochastic identity transformation, is also not good performing in terms of consistency.

An interesting point in Table 3.2 is the performance of Robocentric-EKF. The uncertainty representation w.r.t landmarks in Robocentric-EKF refers to the uncertainty of the coordinates of landmarks relative to robot frame. In this way, Robocentric-EKF has the invariance property to stochastic rigid body transformation. However, Robocentric-EKF does not perform well under the condition of high noise ($\sigma_{od} = 5\%$, $\sigma_{ob} = 5\%$) because it incurs greater linearization errors in the propagation step due to the coordinate transformation applied to the landmarks, as compared to $\mathbb{SO}(3)$-EKF, FEJ-EKF and RI-EKF.

RI-EKF, invariant to stochastic rigid body transformation, is the best performing filter as shown in Table 3.2. Similar results for 2D cases have been reported in [33]. On the other hand, it is discussed in *Remark* 3.11 that FEJ-EKF also has the invariance property to stochastic rigid body transformation but it performs less well than RI-EKF. It can be explained that FEJ-EKF uses a less accurate estimate as linearization point for evaluating the Jacobians while RI-EKF can always safely employ the latest estimate in Jacobians.

TABLE 3.2: **Performance evaluation**

$\sigma_{od} = 1\%$, $\sigma_{ob} = 1\%$	RI-EKF	FEJ-EKF	$\mathbb{SO}(3)$-EKF	Robocentric-EKF	Pseudo-RI-EKF
RMS of position(m)	**0.25**	0.29	0.32	0.31	0.65
RMS of orientation(rad)	**0.0058**	0.0071	0.0065	0.0060	0.0081
NEES of orientation	**1.02**	1.12	1.34	1.04	2.91
NEES of pose	**1.01**	1.14	1.35	1.15	10
$\sigma_{od} = 5\%$, $\sigma_{ob} = 5\%$	RI-EKF	FEJ-EKF	$\mathbb{SO}(3)$-EKF	Robocentric-EKF	Pseudo-RI-EKF
RMS of position(m)	**1.16**	1.24	2.0	2.4	3.90
RMS of orientation(rad)	**0.027**	0.029	0.043	0.041	0.041
NEES of orientation	**1.0**	1.05	3.7	3.0	1.77
NEES of pose	**1.01**	1.13	3.1	7.5	92

Discussion: RI-EKF vs. FEJ-EKF

From TABLE 3.2, we have known that RI-EKF and FEJ-EKF outperforms other filters because both RI-EKF and FEJ-EKF are invariant to stochastic rigid body transformation and have the accurate Jacobian w.r.t feature uncertainty. In order to invest the superiority of RI-EKF relative to FEJ-EKF, we conduct another simulations in which unexpected consequences are included.

In order to invest the superiority of RI-EKF over FEJ-EKF, we conduct another Monte-Carlo simulations with the same settings in Section 3.6. Then we enforcedly set the mean estimate of robot position (for RI-EKF and FEJ-EKF) as $\begin{bmatrix} 1 & -4 & 5 \end{bmatrix}$ and keep the covariance unchanged at the 50-th time step such that the estimates of two filters are erroneous at the 50-th time step. Note that the robot has observed more than 50% landmarks at the 50-th time step.

It is observed in Fig. 3.3 that FEJ-EKF cannot work well from the 50-th time step in which the covariance does not reflect the actual uncertainty. In contrast, RI-EKF can quickly recover estimates and shows much better convergence. The fact can be understood that the invariance of FEJ-EKF will be affected by some unexpected consequences. Once the invariance of FEJ-EKF has been polluted, the filter suffers from the convergence issue because the Jacobians are evaluated at the artificial points instead of the latest points. Note that RI-EKF can always keep invariance property without the need of using historical records to compute the Jacobian, which makes the filter have better convergence and consistency properties.

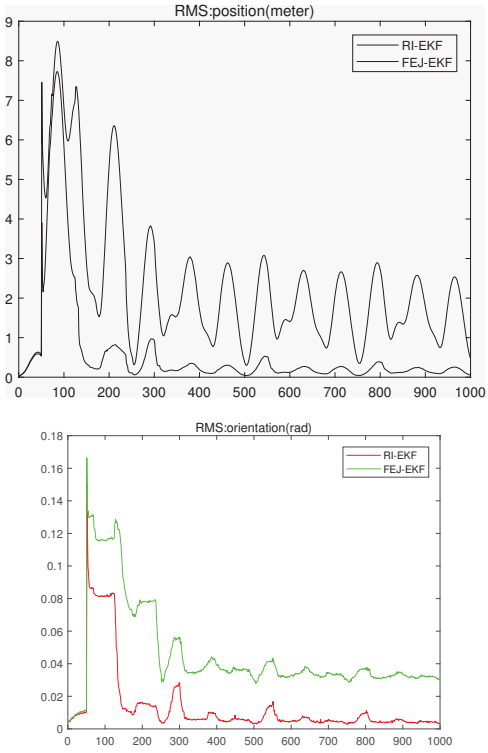

FIGURE 3.3: **Average RMS of robot position and orientation by RI-EKF and FEJ-EKF from 50 Monte Carlo results.** The mean estimates of the two filters are modified at the 50-th step. Noise level: $\sigma_{od} = 1\%$, $\sigma_{ob} = 1\%$.

Remark 3.13. Without the availability of 3D OC-EKF, we have also tested the 2D OC-EKF under the condition that the covariance matrix is artificially modified. It is observed that the 2D OC-EKF still suffers from the similar issue as FEJ-EKF in these cases. In all, both OC-EKF and FEJ-EKF requires a good initial estimate and their invariance is fragile to unexpected cases while the invariance RI-EKF automatically keeps unchanged and the filter is more robust under the condition.

3.7 Summary

In this chapter, the convergence properties and consistency of a Lie group based invariant-EKF SLAM algorithm (RI-EKF) were analyzed. For convergence, several theorems with proofs were provided for two fundamental cases. For consistency, we proposed that consistency of a filter is tightly coupled with the invariance property. We also prove that the

output of RI-EKF is invariant under stochastic rigid body transformation. Monte Carlo simulation results demonstrate that the invariance property has an important impact on the consistency and accuracy of the estimator. In the next chapter, we will focus on the consistency issue of the EKF-VINS algorithm from the invariance viewpoint.

Chapter 4

Extended Kalman Filter in VINS

Like the conventional EKF-SLAM algorithm, the conventional EKF-VINS algorithm is also inconsistent. In this chapter, we first investigate the inconsistency of the conventional EKF-VINS algorithm and then we propose the RIEKF-VINS algorithm to alleviate the inconsistency.

4.1 Introduction

VINS has been of significant interest to the robotics community in the past decade, as the fusion of information from a camera and an IMU provides an effective and affordable solution for navigation in GPS-denied environment. VINS algorithms can be classified into two categories, namely, filter based and optimization based. Although there has been recent progress in the development of optimization based algorithms [6][5], the EKF based solutions are still extensively used (e.g., [3][38][39][40]) mainly as a result of their efficiency and simplicity.

It is well known that conventional EKF based SLAM algorithms [16][17] suffer from inconsistency. Similarly it has been shown that the conventional EKF VINS algorithm (ConEKF-VINS) using point features in the environment is also inconsistent resulting in the under-estimation of the state uncertainty. This is closely related to the partial observability of these systems because conventional EKF algorithms do not necessarily maintain

the partial observability [17][41] due to the linearization errors, which is the main reason for the overconfident estimates. This insight has been a catalyst for a number of *observability-constrained* algorithms (e.g., [42][37][43]), that explicitly enforces the unobservability of the system along specific directions via the modifications to the Jacobian matrices. Although the *observability-constraint* algorithms improve the consistency and accuracy of the estimator to some extent [44], extra computations in the update stage are required. Bloesch et al. in [39] propose a robot-centric formulation to alleviate the inconsistency. Under the robot-centric formulation, the filter estimates the locations of landmarks in the local frame instead of that in the global frame. As a result, the system becomes fully observable so that this issue is inherently avoided. However, this formulation can result in larger uncertainty and extra computations in the propagation stage, as discussed in [23][45].

Recently, the manifold and Lie group representations for three-dimensional orientation/-pose have been utilized for solving SLAM and VINS. Both filter based algorithms (e.g., [45][27][28]) and optimization based algorithms (e.g., [6][29]) can benefit from the manifold representation and better accuracy can be achieved. The use of manifold does not only allow much easier algebraic computations (e.g., the computation of the Jacobian matrices) and avoid the representation singularity [12], but also have inspired a number of researchers to rethink the difference between the state representation and the state error/uncertainty representation, which is highlighted in [6][45]. In fact, this insight is also intrinsically understood in the well-known preintegration visual-inertial algorithm [4] although the algorithm does not use the manifold representation. From the viewpoint of control theory, Aghannan and Rouchon in [31] propose a framework for designing symmetry-preserving observers on manifolds by using a subtle geometrically adapted correction term. The fusion of the symmetry-preserving theory and EKF has resulted in the invariant-EKF (I-EKF), which possesses the theoretical local convergence property [46] and preserves the same invariance property of the original system. I-EKF based observers have been used in the inertial navigation [47] and the 2D EKF-SLAM [32][33]. Our recent work [45] presented in Chapter 3 also proves convergence and improved consistency of a 3D I-EKF SLAM algorithm.

In this chapter, we argue that the absence of the invariance affects the consistency of ConEKF-VINS estimates. There is a correspondence between this and the observability analysis reported in the previous literatures (e.g., [40][37]). The invariance here refers to "the output of the filter is invariant under any *stochastic unobservable transformation*". For the VINS system, the unobservable transformation is translation and rotation about the gravitational direction. Adopting the I-EKF framework, we propose the Right Invariant error EKF VINS algorithm (RIEKF-VINS) and prove that it has this invariance property. We then integrate RIEKF-VINS into the well-known visual-inertial odometry framework, i.e., the multi-state constraint Kalman filter (MSCKF) and remedy the inconsistency of the MSCKF algorithm. We show using extensive Monte Carlo simulations the proposed method outperforms the original MSCKF, especially in terms of the consistency. A preliminary real-world experiment also demonstrates the improved accuracy of the proposed method.

This chapter is organized as follows. Section 4.2 recalls the VINS system and gives an introduction of the ConEKF-VINS under the general continuous-discrete EKF. Section 4.3 introduces the concept invariance, analyzes the consistency of the general EKF algorithm from the invariance theory and proves the absence of the invariance of ConEKF-VINS. Section 4.4 proposes RIEKF-VINS, proves its invariance and extends it to the MSCKF framework. Section 4.5 reports both the simulation and experiment results. Finally, Section 4.6 summaries this chapter. Appendix B provides some necessary formulas used in the proposed algorithms and the proofs of the theorems.

Notations: To simplify the presentation, the vector transpose operators are omitted for the case $\mathbf{A} = \left[\mathbf{a}^\intercal, \mathbf{b}^\intercal, \cdots, \mathbf{c}^\intercal\right]^\intercal$.

4.2 Background knowledge

In this section, we first provide an overview of the VINS system and then describe the ConEKF-VINS algorithm based on the framework of the general continuous-discrete EKF.

The system state

The VINS system is used to estimate the state denoted as the tuple below

$$\mathbf{X} = (\mathbf{R}, \mathbf{v}, \mathbf{p}, \mathbf{b}_g, \mathbf{b}_a, \mathbf{f}) \tag{4.1}$$

where $\mathbf{R} \in \mathbb{SO}(3)$ and $\mathbf{p} \in \mathbb{R}^3$ are the orientation and position of the IMU sensor, respectively, $\mathbf{v} \in \mathbb{R}^3$ is the IMU velocity expressed in the global frame, $\mathbf{b}_g \in \mathbb{R}^3$ is the gyroscope bias, $\mathbf{b}_a \in \mathbb{R}^3$ is the accelerometer bias and $\mathbf{f} \in \mathbb{R}^3$ is the coordinates of the landmark in the global frame. Note that only one landmark is included in the system state (4.1) for a more concise notation.

The continuous-time motion model

The IMU measurements are usually used for state evolution due to its high frequency. The continuous-time motion model of the VINS system is given by the following ordinary differential equations (ODEs):

$$\begin{aligned} \dot{\mathbf{X}} &= f(\mathbf{X}, \mathbf{u}, \mathbf{n}) \\ &= (\mathbf{R}S(\mathbf{w} - \mathbf{b}_g - \mathbf{n}_g), \mathbf{R}(\mathbf{a} - \mathbf{b}_a - \mathbf{n}_a) + \mathbf{g}, \mathbf{v}, \mathbf{n}_{bg}, \mathbf{n}_{ba}, \mathbf{0}) \end{aligned} \tag{4.2}$$

where $\mathbf{w} \in \mathbb{R}^3$ is the gyroscope reading, $\mathbf{a} \in \mathbb{R}^3$ is the accelerometer reading, $\mathbf{g} \in \mathbf{R}^3$ is the global gravity vector (constant), and $\mathbf{n} = \left[\mathbf{n}_g, \mathbf{n}_{bg}, \mathbf{n}_a, \mathbf{n}_{ba}\right]$ is the system noise modeled as a white Gaussian noise with the covariance matrix $\mathbf{Q}$: $\mathbb{E}(\mathbf{n}(t)\mathbf{n}(\tau)^\intercal) = \mathbf{Q}\delta(t - \tau)$. Note that $\mathbf{u} = (\mathbf{w}, \mathbf{a}, \mathbf{g})$ is the time-varying system *input* and the IMU noise covariance $\mathbf{Q}$ is a constant matrix as prior knowledge.

The discrete-time measurement model

The visual measurement as the system *output* is discrete due to the low frequency of camera. After data association and rectification, the visual measurement of the landmark

at time-step $k \in \mathbb{N}$ is available and given by

$$\mathbf{z}_k = h(\mathbf{X}_k, \mathbf{n}_z) = \mathfrak{h}(\mathbf{R}_k^\intercal(\mathbf{f} - \mathbf{p}_k)) + \mathbf{n}_z \tag{4.3}$$

where $\mathbf{n}_z \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_k)$ is the measurement noise. Note that $\mathfrak{h}(\cdot) := \pi \circ \mathbf{T}_{CI}$, where π denotes the projection function and $\mathbf{T}_{CI}$ is the transformation from the IMU frame to the camera frame.

The general continuous-discrete EKF

Being a natural extension of the standard continuous-discrete EKF, the general continuous-discrete EKF allows more flexible uncertainty representation by the following:

$$\mathbf{X} = \hat{\mathbf{X}} \oplus \mathbf{e} \text{ and } \mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}) \tag{4.4}$$

where $(\hat{\mathbf{X}}, \mathbf{P})$ can be regarded as the mean estimate and the covariance matrix, $\mathbf{e}$ is a white Gaussian noise vector and the notation $\oplus$ is the retraction.

Once determining the retraction $\oplus$, the process of the general continuous-discrete EKF is similar to conventional continuous-discrete EKF, as summarized in Alg. 7. For propagation, we first calculate the *time-varying* Jacobians matrices $\mathbf{F}(t)$ and $\mathbf{G}(t)$ from the linearized error-state propagation model:

$$\dot{\mathbf{e}} = \mathbf{F}(t)\mathbf{e} + \mathbf{G}(t)\mathbf{n} + o(\|\mathbf{e}\|\|\mathbf{n}\|). \tag{4.5}$$

We then compute the *state transition matrix* $\mathbf{\Phi}_n := \mathbf{\Phi}(t_{n+1}, t_n)$ that is the solution at time t_{n+1} of the following ODE:

$$\frac{d}{dt}\mathbf{\Phi}(t, t_n) = \mathbf{F}(t)\mathbf{\Phi}(t, t_n) \tag{4.6}$$

with the condition $\mathbf{\Phi}(t_n, t_n) = \mathbf{I}$ at time t_n. The matrix $\mathbf{Q}_{d,n}$ can be computed as

$$\mathbf{Q}_{d,n} = \int_{t_n}^{t_{n+1}} \mathbf{\Phi}(t_{n+1}, \tau)\mathbf{G}(\tau)\mathbf{Q}\mathbf{G}^\intercal(\tau)\mathbf{\Phi}^\intercal(t_{n+1}, \tau)d\tau. \tag{4.7}$$

Algorithm 7: The general continuous-discrete EKF

Input: $\hat{\mathbf{X}}_n$, $\mathbf{P}_n$, $\mathbf{u}_{t_n:t_{n+1}}$, $\mathbf{z}_{n+1}$;
Output: $\hat{\mathbf{X}}_{n+1}$, $\mathbf{P}_{n+1}$;
Propagation:
Turn off the system noise and compute $\hat{\mathbf{X}}_{n+1|n}$ with $\hat{\mathbf{X}}_n$ and the ODEs (4.2);
$\mathbf{P}_{n+1|n} \leftarrow \mathbf{\Phi}_n \mathbf{P}_n \mathbf{\Phi}_n^\intercal + \mathbf{Q}_{d,n}$;
Update:
$\mathbf{H}_{n+1} = \frac{\partial h(\hat{\mathbf{X}}_{n+1|n} \oplus \mathbf{e}, \mathbf{0})}{\partial \mathbf{e}}|_{\mathbf{e}=\mathbf{0}}$;
$\mathbf{S} \leftarrow \mathbf{H}_{n+1}\mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\intercal + \mathbf{V}_{n+1}$, $\mathbf{K} \leftarrow \mathbf{P}_{n+1|n}\mathbf{H}_{n+1}^\intercal \mathbf{S}^{-1}$;
$\tilde{\mathbf{z}} \leftarrow h(\hat{\mathbf{X}}_{n+1|n}, \mathbf{0}) - \mathbf{z}_{n+1}$;
$\hat{\mathbf{X}}_{n+1} \leftarrow \hat{\mathbf{X}}_{n+1|n} \oplus \mathbf{K}\tilde{\mathbf{z}}$, $\mathbf{P}_{n+1} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H}_{n+1})\mathbf{P}_{n+1|n}$;

ConEKF-VINS

ConEKF-VINS [41] can be regarded as an instance of the general EKF algorithm (Alg. 7). In ConEKF-VINS, the uncertainty representation is defined as

$$\begin{aligned} \mathbf{X} &= \hat{\mathbf{X}} \oplus \mathbf{e} \\ &= \left(\hat{\mathbf{R}} \exp(\mathbf{e}_\theta), \hat{\mathbf{v}} + \mathbf{e}_v, \hat{\mathbf{p}} + \mathbf{e}_p, \hat{\mathbf{b}}_g + \mathbf{e}_{bg}, \hat{\mathbf{b}}_a + \mathbf{e}_{ba}, \hat{\mathbf{f}} + \mathbf{e}_f\right) \end{aligned} \tag{4.8}$$

where $\mathbf{e} = \left[\mathbf{e}_\theta, \mathbf{e}_v, \mathbf{e}_p, \mathbf{e}_{bg}, \mathbf{e}_{ba}, \mathbf{e}_f\right] \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$. The calculation of matrices $\mathbf{\Phi}_n$, $\mathbf{Q}_{d,n}$ and $\mathbf{H}_{n+1}$ are omitted here, which can be straightforwardly calculated in the sense of the uncertainty representation (4.8). Please refer to [41] for more details.

4.3 Invariance and consistency

In this section, we first introduce the concepts of observability, unobservable transformation and invariance. We then perform the consistency analysis for the general EKF filter and prove that ConEKF-VINS does *not* have the expected invariance property. Moreover, we also discuss the relationship between invariance and consistency.

Invariance of the VINS system

The concept observability of nonlinear systems can be traced to the early literature [36]. As discussed in the literatures [40][48][49], the state of the VINS system is not locally observable. To make it more intuitive, we introduce the unobservability of the VINS system based on the *unobservable transformation* rather than the *observability rank criterion* reported in [36].

Recall the invariance concept introduced in Section 2.8. A state transformation $\mathcal{T}$ is called an **unobservable transformation** for a system if the output of the system is **invariant** under $\mathcal{T}$. The system is called to be **unobservable** if there exists a non-trivial *unobservable transformation*.

Definition 4.1. For the VINS system (4.1)-(4.3), a state transformation $\mathcal{T}$ is an *unobservable transformation* if for arbitrary t_i, we have $h(\mathbf{X}(t_n), \mathbf{0}) = h(\mathbf{Y}(t_n), \mathbf{0})$ for all $t_n \geq t_i$, where $\mathbf{X}(\cdot)$ and $\mathbf{Y}(\cdot)$ denote the two state trajectories that follow the ODEs (4.2) (fed by the same system input) with the conditions $\mathbf{X}(t_i)$ and $\mathbf{Y}(t_i) = \mathcal{T}(\mathbf{X}(t_i))$ at time t_i, respectively.

Remark 4.2. One can see that an unobservable system is always accompanied by an *unobservable transformation*. And the invariance to the *unobservable transformation* is another description of the system unobservability.

Definition 4.3. For the system state (4.1), a **stochastic transformation of translation and rotation (about the gravitational direction)** $\mathcal{T}_{\mathbf{S}}$ is a mapping:

$$\begin{aligned} &\mathcal{T}_{\mathbf{S}}(\mathbf{X}) = (\exp(\mathbf{g}(\epsilon_1 + \theta_1))\mathbf{R}, \exp(\mathbf{g}(\epsilon_1 + \theta_1))\mathbf{v}, \\ &\exp(\mathbf{g}(\epsilon_1 + \theta_1))\mathbf{p} + \boldsymbol{\theta}_2 + \boldsymbol{\epsilon}_2, \\ &\mathbf{b}_g, \mathbf{b}_a, \exp(\mathbf{g}(\epsilon_1 + \theta_1))\mathbf{f} + \boldsymbol{\theta}_2 + \boldsymbol{\epsilon}_2) \end{aligned} \tag{4.9}$$

where $\mathbf{S} = (\boldsymbol{\theta}, \boldsymbol{\epsilon})$, $\theta_1 \in \mathbb{R}$, $\boldsymbol{\theta}_2 \in \mathbb{R}^3$, $\boldsymbol{\theta} = \begin{bmatrix} \theta_1, \boldsymbol{\theta}_2 \end{bmatrix} \in \mathbb{R}^4$, $\epsilon_1 \in \mathbb{R}$, $\boldsymbol{\epsilon}_2 \in \mathbb{R}^3$ and $\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1, \boldsymbol{\epsilon}_2 \end{bmatrix} \in \mathbb{R}^4$ is a white Gaussian noise with the covariance $\boldsymbol{\Sigma}$. $\mathcal{T}_{\mathbf{S}}$ degenerates into the **deterministic transformation** $\mathcal{T}_{\mathbf{D}}$ ($\mathbf{D} = (\boldsymbol{\theta}, \mathbf{0})$) under the condition $\boldsymbol{\Sigma} = \mathbf{0}$. $\mathcal{T}_{\mathbf{S}}$ degenerates into a **stochastic identity transformation** under the condition $\boldsymbol{\theta} = \mathbf{0}$, .

Theorem 4.4. *The stochastic transformation $\mathcal{T}_{\mathbf{S}}$ is an unobservable transformation to the VINS system (4.1)–(4.3).*

Proof. Denote $\mathbf{x}_n$ as the system state at the time step n, propagated from the initial state $\mathbf{x}_0$. Denote $\mathbf{y}_n$ as the system state at the time step n, propagated from the initial state $\mathbf{y}_0$. It is easily to verify that $\mathbf{y}_n = \mathcal{T}_{\mathbf{S}}(\mathbf{x}_n)$ if $\mathbf{y}_0 = \mathcal{T}_{\mathbf{S}}(\mathbf{x}_0)$. Obviously, $h(\mathbf{y}_n, \mathbf{0}) = h(\mathbf{x}_n, \mathbf{0})$ if $\mathbf{y}_0 = \mathcal{T}_{\mathbf{S}}(\mathbf{x}_0)$. □

Remark 4.5. Theorem 4.4 corresponds to the conclusion in [40][50] that the IMU yaw angle and the IMU position are (locally) unobservable.

The invariance of the general EKF based filter

The general EKF based filter is **not** a linear system for the estimated state $\hat{\mathbf{X}}$ and hence we argue that the observability analysis for linear systems cannot be directly applied. However, the invariance of the filter can be naturally described as the following.

Definition 4.6. The output of a general EKF framework based filter (Alg. 7) for the VINS system is invariant under a **stochastic unobservable transformation $\mathcal{T}_{\mathbf{S}}$** if the following condition is satisfied: for any two estimates $(\hat{\mathbf{X}}_i, \mathbf{P}_i)$ and $(\hat{\mathbf{Y}}_i, \mathbf{P}y_i) = (\mathcal{T}_{\mathbf{S}}(\hat{\mathbf{X}}_i), \mathbf{M}_i\mathbf{P}_i\mathbf{M}_i^{\intercal} + \mathbf{N}_i\mathbf{\Sigma}\mathbf{N}_i^{\intercal})$ at time-step i, we have $h(\hat{\mathbf{X}}_n, \mathbf{0}) = h(\hat{\mathbf{Y}}_n, \mathbf{0})$ for all $n \geq i$, where $\hat{\mathbf{X}}_n$ and $\hat{\mathbf{Y}}_n$ represent the *mean* estimate of this filter at time-step n by using the same input $\mathbf{u}$ from time t_i to t_n, from the conditions $(\hat{\mathbf{X}}_i, \mathbf{P}_i)$ and $(\hat{\mathbf{Y}}_i, \mathbf{P}y_i)$ at time-step i, respectively. Note that here

$$\mathbf{M}_i := \left.\frac{\partial \mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_i \oplus \mathbf{e}) \ominus \mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_i)}{\partial \mathbf{e}}\right|_{\mathbf{e}=\mathbf{0}} \tag{4.10}$$

and

$$\mathbf{N}_i := \left.\frac{\partial \mathcal{T}_{\mathbf{S}}(\hat{\mathbf{X}}_i) \ominus \mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_i)}{\partial \boldsymbol{\epsilon}}\right|_{\boldsymbol{\epsilon}=\mathbf{0}}. \tag{4.11}$$

As shown in Def. 4.3 and Def. 4.6, the invariance to any stochastic transformation $\mathcal{T}_{\mathbf{S}}$ can be divided into two properties: 1) **the invariance to any deterministic transformation $\mathcal{T}_{\mathbf{D}}$** and 2) **the invariance to any stochastic identity transformation**. The following two theorems analytically provide the methods to determine whether a general EKF based filter has the two invariances properties above.

Theorem 4.7. *The output of the general EKF based filter for the VINS system is invariant under any deterministic unobservable transformation if and only if for each deterministic unobservable transformation $\mathcal{T}_{\mathbf{D}}$, there exists an invertible matrix $\mathbf{W}_{\mathbf{D}}$ (unrelated to $\mathbf{X}$) such that*

$$\mathcal{T}_{\mathbf{D}}(\mathbf{X} \oplus \mathbf{e}) = \mathcal{T}_{\mathbf{D}}(\mathbf{X}) \oplus \mathbf{W}_{\mathbf{D}}\mathbf{e}. \tag{4.12}$$

Proof. See Appendix B.1. □

Theorem 4.8. *The output of the general EKF based filter for the VINS system is invariant under any stochastic identity transformation if and only if*

$$\mathbf{H}_{n+i+1}\mathbf{\Phi}_{n+i}\mathbf{\Phi}_{n+i-1}\cdots\mathbf{\Phi}_{i}\mathbf{N}_{i} = \mathbf{0},\ \forall\ n, i \geq 0. \tag{4.13}$$

Proof. See Appendix B.2. □

By using the theorems above, we can easily determine the invariance properties of ConEKF-VINS.

Theorem 4.9. *ConEKF-VINS satisfies (4.12) but does not satisfies (4.13). Hence, ConEKF-VINS has the invariance to any deterministic unobservable transformation $\mathcal{T}_{\mathbf{D}}$ but not the invariance to stochastic identity transformations. In all, the output of ConEKF-VINS is not invariant under stochastic unobservable transformation $\mathcal{T}_{\mathbf{S}}$.*

Proof. For ConEKF-VINS, the invariance to the deterministic unobservable transformation $\mathcal{T}_{\mathbf{D}}$ can be verified by using Theorem 4.7. The absence of invariance of ConEKF-VINS to stochastic identity transformations can be verified by using Theorem 4.8. □

Remark 4.10. The previous literatures [41][42][37][43] directly perform the observability analysis of the filter on the *linearized error-state model*. Theorem 4.7 and Theorem 4.8 elegantly clarify the relationship between the filter invariance and the *linearized error-state model* and prove that the observability analysis in previous literatures refers to the invariance to stochastic identity transformation in stead of the invarance to stochastic unobservable transformation.

Consistency and invariance

The unobservability in terms of stochastic unobservable transformation $\mathcal{T}_{\mathbf{S}}$ is a fundamental property of the VINS system. Therefore a consistent filter (as a system for the estimated state $\hat{\mathbf{X}}$) is expected to mimic this property, i.e., **the output of a consistent estimator should be invariant under any stochastic unobservable transformation.** The invariance to any deterministic transformation $\mathcal{T}_{\mathbf{D}}$ essentially implies that the estimates from the filter do *not* depend on the selection of the (initial) mean estimate of the unobservable variables, i.e., the IMU yaw angle and the IMU position. Similarly, the invariance to any stochastic identity transformation essentially implies that the uncertainty w.r.t these unobservable variables does not affect the subsequent *mean* estimates. We can conclude that **the consistency of a filter is tightly coupled with the invariance to stochastic unobservable transformations**. A filter lack of the invariance property will gain spurious information and produce inconsistent (overconfident) estimates. Note that ConEKF-VINS is a typical example due to the absence of the invariance property.

4.4 The proposed method: RIEKF-VINS

In this section, we propose RIEKF-VINS by using a new uncertainty representation and prove that it has the expected invariance properties. We then apply RIEKF-VINS to the MSCKF framework.

The uncertainty representation and Jacobians

RIEKF-VINS also follows the framework presented in Alg. 7. The uncertainty representation of RIEKF-VINS is defined as below

$$\begin{aligned}\mathbf{X} =&\hat{\mathbf{X}} \oplus \mathbf{e} \\ =&(\exp(\mathbf{e}_\theta)\hat{\mathbf{R}}, \exp(\mathbf{e}_\theta)\hat{\mathbf{v}} + \mathrm{J}_r(-\mathbf{e}_\theta)\mathbf{e}_v, \\ &\exp(\mathbf{e}_\theta)\hat{\mathbf{p}} + \mathrm{J}_r(-\mathbf{e}_\theta)\mathbf{e}_p, \hat{\mathbf{b}}_g + \mathbf{e}_{bg}, \hat{\mathbf{b}}_a + \mathbf{e}_{ba}, \\ &\exp(\mathbf{e}_\theta)\hat{\mathbf{f}} + \mathrm{J}_r(-\mathbf{e}_\theta)\mathbf{e}_f)\end{aligned} \tag{4.14}$$

where $\mathbf{e} = \begin{bmatrix}\mathbf{e}_\theta, \mathbf{e}_v, \mathbf{e}_p, \mathbf{e}_{bg}, \mathbf{e}_{ba}, \mathbf{e}_f\end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$ and the right Jacobian operator $\mathrm{J}_r(\cdot)$ is given in (2.21).

Remark 4.11. The design of the retraction (4.14) is motivated by RIEKF-SLAM in which the retraction of the landmarks location is coupled with the incremental vector of the orientation.

Note that this uncertainty representation intrinsically employs the Lie group so that the recent result (Theorem 2 of [46]) can be used to easily compute the Jacobians $\mathbf{F}$ and $\mathbf{G}$ of the propagation

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -\hat{\mathbf{R}} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ S(\mathbf{g}) & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & -S(\hat{\mathbf{v}})\hat{\mathbf{R}} & -\hat{\mathbf{R}} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3} & -S(\hat{\mathbf{p}})\hat{\mathbf{R}} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \end{bmatrix} \tag{4.15}$$

and

$$\mathbf{G} = \begin{bmatrix} \hat{\mathbf{R}} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ S(\hat{\mathbf{v}})\hat{\mathbf{R}} & \mathbf{0}_{3,3} & \hat{\mathbf{R}} & \mathbf{0}_{3,3} \\ S(\hat{\mathbf{p}})\hat{\mathbf{R}} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 \\ S(\hat{\mathbf{f}})\hat{\mathbf{R}} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \end{bmatrix}. \tag{4.16}$$

The measurement Jacobian is

$$\mathbf{H}_{n+1} = \partial\mathfrak{h}(\hat{\mathbf{f}}_{n+1,I}) \begin{bmatrix} \mathbf{0}_{3,6} & -\hat{\mathbf{R}}_{n+1|n}^\intercal & \mathbf{0}_{3,6} & \hat{\mathbf{R}}_{n+1|n}^\intercal \end{bmatrix} \tag{4.17}$$

where $\hat{\mathbf{f}}_{n+1,I} = \hat{\mathbf{R}}_{n+1|n}^\intercal(\hat{\mathbf{f}}_{n+1|n} - \hat{\mathbf{p}}_{n+1|n}) \in \mathbb{R}^3$.

Invariance property

Theorem 4.12. *The output of RIEKF-VINS is invariant under any stochastic unobservable transformation $\mathcal{T}_\mathbf{S}$ defined in (4.9).*

Proof. For the retraction defined in (4.14), we have $\mathcal{T}_{\mathbf{D}}(\mathbf{X} \oplus \mathbf{e}) = \mathcal{T}_{\mathbf{D}}(\mathbf{X}) \oplus \mathbf{W}_{\mathbf{D}}\mathbf{e}$ for all $\mathbf{X}$ and $\mathbf{e}$, where

$$\mathbf{W}_{\mathbf{D}} = \begin{bmatrix} \delta\mathbf{R} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \delta\mathbf{R} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ S(\boldsymbol{\theta}_2)\delta\mathbf{R} & \mathbf{0}_{3,3} & \delta\mathbf{R} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3} \\ S(\boldsymbol{\theta}_2)\delta\mathbf{R} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \delta\mathbf{R} \end{bmatrix} \tag{4.18}$$

and $\delta\mathbf{R} := \exp(\mathbf{g}\theta_1)$. According to Theorem 4.7, the output of RIEKF-VINS is invariant under any deterministic transformation $\mathcal{T}_{\mathbf{D}}$. On the other hand, for all i, we have

$$\boldsymbol{\Phi}_i = \begin{bmatrix} \mathbf{I}_3 & * & \mathbf{0}_{3,3} & * & * & \mathbf{0}_{3,3} \\ \Delta t_i S(\mathbf{g}) & * & \mathbf{0}_{3,3} & * & * & \mathbf{0}_{3,3} \\ \frac{\Delta t_i^2}{2} S(\mathbf{g}) & * & \mathbf{I}_3 & * & * & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & * & \mathbf{0}_{3,3} & * & * & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & * & \mathbf{0}_{3,3} & * & * & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & * & \mathbf{0}_{3,3} & * & * & \mathbf{I}_3 \end{bmatrix} \tag{4.19}$$

and

$$\mathbf{N}_i = \left.\frac{\partial \mathcal{T}_{\mathbf{S}}(\hat{\mathbf{X}}_i) \ominus \mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_i)}{\partial \boldsymbol{\epsilon}}\right|_{\boldsymbol{\epsilon}=\mathbf{0}} = \begin{bmatrix} \mathbf{g} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,1} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,1} & \mathbf{I}_3 \\ \mathbf{0}_{3,1} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,1} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,1} & \mathbf{I}_3 \end{bmatrix} \tag{4.20}$$

where $\Delta t_i := t_{i+1} - t_i$ and the elements denoted by the notation $*$ are omitted here because these do not have any contribution to the computation of $\boldsymbol{\Phi}_i\mathbf{N}_i$. Note that $\boldsymbol{\Phi}_i\mathbf{N}_i = \mathbf{N}_{i+1}$ and $\mathbf{H}_{i+1}\mathbf{N}_{i+1} = \mathbf{0}$ for all i and then we can easily verify that RIEKF-VINS satisfies (4.13). According to Theorem 4.8, the output of RIEKF-VINS is invariant under any stochastic identity transformation. □

Remark 4.13. The *observability-constrained* filters proposed in [41][42][37][43] artificially

modify the transition matrix $\mathbf{\Phi}_n$ and the measurement Jacobian $\mathbf{H}_{n+1}$ to meet the condition (4.13) such that they have the invariance to stochastic identity transformation. As a comparison, our proposed RIEKF-VINS employs the novel uncertainty representation (4.14) such that the resulting matrices $\mathbf{\Phi}_n$ and $\mathbf{H}_{n+1}$ can naturally and elegantly meet the condition (4.13).

Application to MSCKF

A drawback of ConEKF-VINS and RIEKF-VINS is the expensive cost of maintaining the covariance matrix for a number of landmarks. Especially, RIEKF-VINS suffers from the complexity quadratic to the number of landmarks in the propagation stage. On the other hand, the well known MSCKF [3] that has the complexity linear to the number of landmarks inherits the inconsistency of ConEKF-VINS. One can see that the uncertainty w.r.t the global yaw has effects on the mean estimates in the MSCKF algorithm, which is a drawback. Due to the reasons above, we integrate RIEKF-VINS into the MSCKF framework such that the modified algorithm has the linear complexity and better consistency. For convenience, we call the modified filter as RI-MSCKF. In this subsection, we do not state all details of RI-MSCKF but point out the main modifications on top of MSCKF.

System state and retraction

The system state $\mathcal{X}_n$ at time-step n in RI-MSCKF is

$$\mathcal{X}_n = (\bar{\mathbf{X}}_n, \mathbf{C}_{t_1}, \cdots, \mathbf{C}_{t_j}, \cdots, \mathbf{C}_{t_k}, \cdots, \mathbf{C}_{t_m}) \tag{4.21}$$

where $\bar{\mathbf{X}}_n = (\mathbf{R}_n, \mathbf{v}_n, \mathbf{p}_n, \mathbf{b}_{g,n}, \mathbf{b}_{a,n})$ denotes the IMU state at time-step n, $\mathbf{C}_{t_i} = (\mathbf{R}^c_{t_i}, \mathbf{p}^c_{t_i}) \in \mathbb{SE}(3)$ denotes the camera pose at the time t_i ($t_i < t_n$). According to the IMU state uncertainty in RIEKF-VINS, the uncertainty representation of $\mathcal{X}_n$ are defined as below

$$\begin{aligned} \mathcal{X}_n &= \hat{\mathcal{X}}_n \oplus \mathbf{e} \\ &= (\hat{\bar{\mathbf{X}}}_n \oplus_{imu} \mathbf{e}_I, \hat{\mathbf{C}}_{t_1} \oplus_{pose} \mathbf{e}^1_c, \cdots, \hat{\mathbf{C}}_{t_m} \oplus_{pose} \mathbf{e}^m_c) \end{aligned} \tag{4.22}$$

where $\mathbf{e} = \left[\mathbf{e}_I, \mathbf{e}_c\right] \in \mathbb{R}^{15+6m} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_n)$, $\mathbf{e}_I \in \mathbb{R}^{15}$ and $\mathbf{e}_c = \left[\mathbf{e}_c^1, \cdots, \mathbf{e}_c^m\right] \in \mathbb{R}^{6m}$. Note that $\oplus_{imu}$ and $\oplus_{pose}$ are given in Appendix B.

Propagation

The mean propagation $\mathcal{X}_{n+1|n}$ of RI-MSCKF also follows that of MSCKF while the covariance $\mathbf{P}_{n+1|n}$ is calculated by

$$\mathbf{P}_{n+1|n} = \bar{\mathbf{\Phi}}_n^\intercal \mathbf{P}_n \bar{\mathbf{\Phi}}_n + \bar{\mathbf{Q}}_{d,n} \tag{4.23}$$

where $\bar{\mathbf{\Phi}}_n = \mathbf{blkdiag}(\mathbf{\Phi}_n^I, \mathbf{I}_{6m})$, $\bar{\mathbf{Q}}_{d,n} = \mathbf{blkdiag}(\mathbf{Q}_{d,n}^I, \mathbf{0}_{6m,6m})$. Note that $\mathbf{\Phi}_n^I$ and $\mathbf{Q}_{d,n}^I$ are the matrices from the first 15 rows and 15 columns of $\mathbf{\Phi}_n$ and $\mathbf{Q}_{d,n}$, respectively, where $\mathbf{\Phi}_n$ and $\mathbf{Q}_{d,n}$ are the matrices of RIEKF-VINS.

State augment

Once a new image is captured at time-step $n+1$, we augment the system state and the covariance matrix as the following:

$$\hat{\mathcal{X}}_{n+1|n} \leftarrow (\hat{\mathcal{X}}_{n+1|n}, \hat{\mathbf{C}}_{t_{n+1}}) \tag{4.24}$$

$$\mathbf{P}_{n+1|n} \leftarrow \begin{bmatrix} \mathbf{I}_{15+6m} \\ \mathbf{J} \end{bmatrix} \mathbf{P}_{n+1|n} \begin{bmatrix} \mathbf{I}_{15+6m} \\ \mathbf{J} \end{bmatrix}^\intercal \tag{4.25}$$

where $\hat{\mathbf{C}}_{t_{n+1}} = (\hat{\mathbf{R}}_{n+1|n}\Delta\mathbf{R}, \hat{\mathbf{R}}_{n+1|n}\Delta\mathbf{p} + \hat{\mathbf{p}}_{n+1|n}) \in \mathbb{SE}(3)$ is the *mean* estimate of camera pose at the time t_{n+1}, $(\Delta\mathbf{R}, \Delta\mathbf{p}) \in \mathbb{SE}(3)$ denotes the transformation from the camera to the IMU. Due to the new uncertainty representation (4.22), the Jacobian $\mathbf{J}$ needs to be changed as below

$$\mathbf{J} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,6} & \mathbf{0}_{3,6m} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,6} & \mathbf{0}_{3,6m} \end{bmatrix}. \tag{4.26}$$

Update

Note that the landmark uncertainty is coupled with the IMU pose in RIEKF-VINS. In RI-MSCKF, we describe the landmark uncertainty coupled with the camera pose $\mathbf{C}_{t_j}$ that first captures the landmark within the current system state $\mathcal{X}_n$ as below

$$(\hat{\mathbf{C}}_{t_j}, \hat{\mathbf{f}}) \oplus \bar{\mathbf{e}}_c^j = (\hat{\mathbf{C}}_{t_j} \oplus_{pose} \mathbf{e}_c^j, \mathbf{e}_\theta^j \hat{\mathbf{f}} + J_r(-\mathbf{e}_\theta^j)\mathbf{e}_f) \tag{4.27}$$

where $\bar{\mathbf{e}}_c^j = [\mathbf{e}_c^j, \mathbf{e}_f] = [\mathbf{e}_\theta^j, \mathbf{e}_p^j, \mathbf{e}_f] \in \mathbf{R}^9$. From the uncertainty representations (4.22) and (4.27), we can compute the linearized measurement model for the visual measurement at time-step k ($t_1 \leq t_k \leq t_n$). With a slight abuse of notations, the linearized measurement model can be represented as below

$$\begin{aligned} \pi(\hat{\mathbf{R}}_{t_k}^{c\intercal}(\hat{\mathbf{f}} - \hat{\mathbf{p}}_{t_k}^c)) - \mathbf{z}_k &\approx \partial\pi_k \mathbf{H}_{xk}^* \mathbf{e}_{n+1|n} + \partial\pi_k \mathbf{H}_{fk}^* \mathbf{e}_f + \mathbf{V}_k \\ \tilde{\mathbf{z}}_k &\approx \partial\pi_k \mathbf{H}_{xk}^* \mathbf{e}_{n+1|n} + \partial\pi_k \mathbf{H}_{fk}^* \mathbf{e}_f + \mathbf{V}_k \\ \tilde{\mathbf{z}}_k &\approx \mathbf{H}_{xk} \mathbf{e}_{n+1|n} + \mathbf{H}_{fk} \mathbf{e}_f + \mathbf{V}_k \end{aligned} \tag{4.28}$$

where $\partial\pi_k := \partial\pi(\hat{\mathbf{R}}_{t_k}^{c\intercal}(\hat{f} - \mathbf{p}_{t_k}^c))$, $\mathbf{z}_k$ is the measurement captured at the time t_k. Here the matrices $\mathbf{H}_{fk}^*$ and $\mathbf{H}_{xk}^*$ are given by

$$\mathbf{H}_{fk}^* = \hat{\mathbf{R}}_{t_k}^{c\intercal} \tag{4.29}$$

and

$$\mathbf{H}_{xk}^* = \begin{bmatrix} \dots & \dots & \dots & \mathbf{A} & \dots & \mathbf{B} & \dots & \dots \end{bmatrix} \tag{4.30}$$

where $\mathbf{A} = \begin{bmatrix} -\hat{\mathbf{R}}_{t_k}^{c\intercal} S(\hat{\mathbf{f}}), \mathbf{0}_{3,3} \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} \hat{\mathbf{R}}_{t_k}^{c\intercal} S(\hat{\mathbf{f}}), -\hat{\mathbf{R}}_{t_k}^{c\intercal} \end{bmatrix}$. Due to the absence of the covariance of landmark, RI-MSCKF also uses the null-space trick on (4.28) and the resulting residual equation

$$\begin{aligned} \mathbf{H}_{fk}^\perp \tilde{\mathbf{z}}_k &\approx \mathbf{H}_{fk}^\perp \mathbf{H}_{xk} \mathbf{e}_{n+1|n} + \mathbf{H}_{fk}^\perp \mathbf{V}_k \\ \tilde{\mathbf{z}}_k' &\approx \mathbf{H}_{xk}' \mathbf{e}_{n+1|n} + \mathbf{V}_k' \end{aligned} \tag{4.31}$$

is employed for update.

Remark 4.14. RI-MSCKF does not need any extra computation to maintain the expected

invariance while the *observability-constraint* algorithms need to explicitly project the measurement Jacobians onto the *observable subspace*.

4.5 Simulation and experiment

Simulation Result

In order to validate the theoretical contributions in this chapter, we perform 50 Monte Carlo simulations and compare RI-MSCKF with MSCKF for a Visual-Inertial Odometry (VIO) scenario without loop closure.

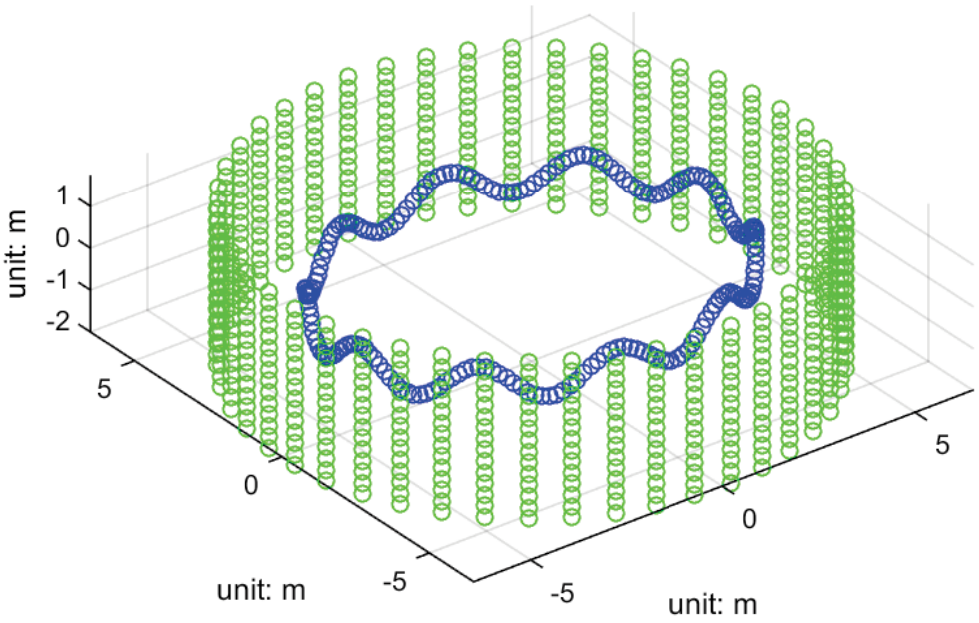

FIGURE 4.1: The simulated trajectory (blue circles) and landmarks (green circles).

Consider that a robot equipped with an IMU and a camera moves in a specific trajectory (average speed is 3m/s) with the sufficient 6-DOFs motion, shown as the blue circles in Fig. 4.1. In this environment, 675 landmarks are distributed on the surface of a cylinder with radius 6.5m and height 4m shown as the green circles in Fig. 4.1. Under the simulated environment, the camera is able to observe sufficiently overlapped landmarks

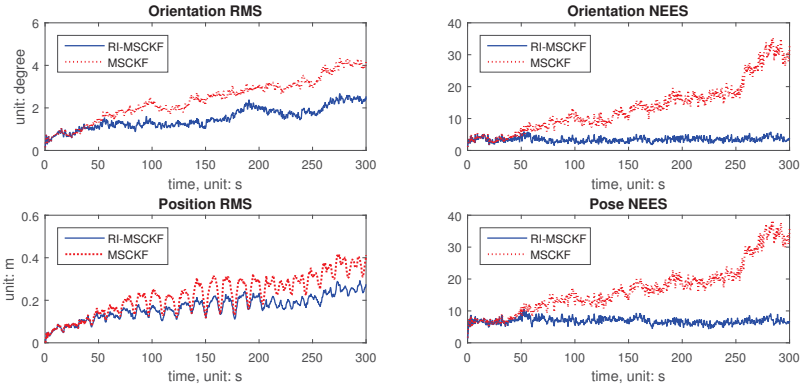

FIGURE 4.2: 50 Monte Carlo simulation results. The proposed RI-MSCKF outperforms the original MSCKF, both in terms of accuracy (RMS) and consistency (NEES).

between consecutive frames. The standard deviation of camera measurement is set as 1.5 pixels. The IMU noise covariance $\mathbf{Q}$ is set as $\mathbf{diag}(0.008^2\mathbf{I}_3, 0.0004^2\mathbf{I}_3, 0.019^2\mathbf{I}_3, 0.05^2\mathbf{I}_3)$ (the International System of Units). In each round of Monte Carlo simulation, the initial estimate is set as the ground truth. And the measurements from IMU and camera are generated from the same trajectory with random noises. The maximal number of camera poses in the system state of RI-MSCKF and MSCKF is set as 10. For robust estimation, we use the landmarks for the update step only when the landmarks are captured more than 5 times by the cameras within the current system state.

The results of 50 Monte Carlo simulations are plotted in Fig. 4.2. We use the root mean square error (RMS) and the average normalized estimation error squared (NEES) to evaluate both accuracy and consistency, respectively. Note that the *ideal* NEES of orientation is 3 and that of pose is 6. As shown in Fig. 4.2, RI-MSCKF clearly outperforms MSCKF especially for the consistency. This phenomenon can be explained as RI-MSCKF has the invariance property to stochastic translation and rotation about the gravitational direction and thus it can prevent the unexpected information gain when compared to MSCKF. In addition, the RMS of orientation and position of both filters increase with the time because the loop closure in this simulation is turned off.

Preliminary Experiment

In order to validate the performance of the proposed RI-MSCKF algorithm under practical environments, we evaluate the algorithm on Euroc dataset [51] which is collected on-board a macro aerial vehicle in the indoor environments. Without a delicately designed front-end which handles the feature extraction and tracking perfectly, we selected sequence *V2_01_easy* in this section to demonstrate the performance of the RI-MSCKF algorithm where the features can be tracked relatively easily and thus making it useful to compare our algorithm against the MSCKF algorithm.

In this preliminary experiment, we designed a front-end based on ORB-SLAM [52] while only keeping the feature tracking sub-module. Without knowing the map points, new keyframe is inserted once $n_{\text{frames}} = 5$ frames have passed since the insertion of the last keyframe. One sample image with the tracked landmarks is shown in Fig. 4.4. The uncertainty of the IMU sensor is set as instructed in the dataset. The maximal number of the camera poses in the system state is set as 10.

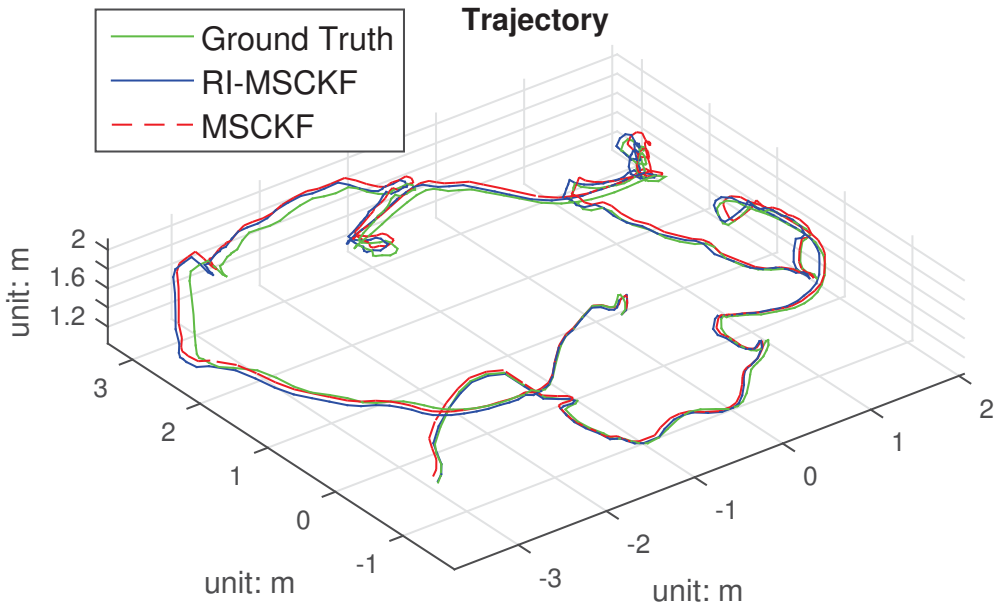

FIGURE 4.3: The estimated trajectories from MSCKF and RI-MSCKF using the Euroc dataset sequence *V2_01_easy*.

Fig. 4.3 shows the estimated trajectories using MSCKF and RI-MSCKF. As shown in Fig. 4.3 and indicated in Fig. 4.5, RI-MSCKF shows the similar accuracy of position

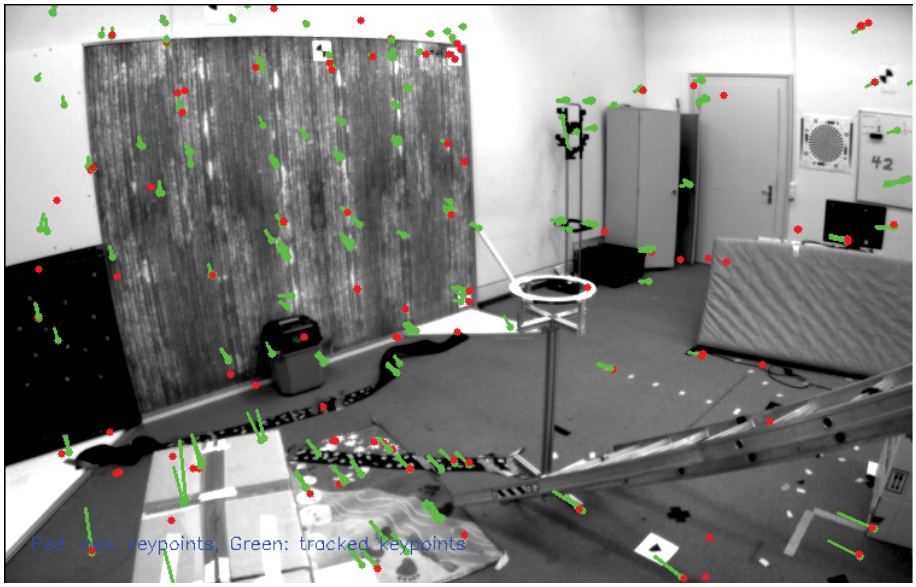

FIGURE 4.4: A sample image with landmarks in the experiment. The green dots represent the tracked key points and the red dots represents the new key points.

compared with MSCKF but also avoids the drift in the last few frames of the sequence, however, RI-MSCKF shows significant better results in terms of orientation estimation accuracy compared with the original MSCKF algorithm. Even without a robust front-end to handle feature tracking perfectly, this preliminary experiment is able to demonstrate the superiority of RI-MSCKF compared with MSCKF algorithm in terms of the estimation accuracy.

4.6 Summary

In this chapter, we proposed the RIEKF-VINS algorithm and stressed that the consistency of a filter is tightly coupled with the invariance property. We proved that RIEKF-VINS has the expected invariance property while ConEKF-VINS does not satisfy this property. We also provided the methods to check whether a general EKF based filter for the unobservable system has the invariance properties. After theoretical analysis, we integrated RIEKF-VINS into the MSCKF framework such that the resulting RI-MSCKF algorithm can achieve better consistency relative to the original MSCKF. Monte Carlo simulations illustrated the significantly improved performance of RI-MSCKF, especially for the consistency. The real-world experiments also validated its improved accuracy.

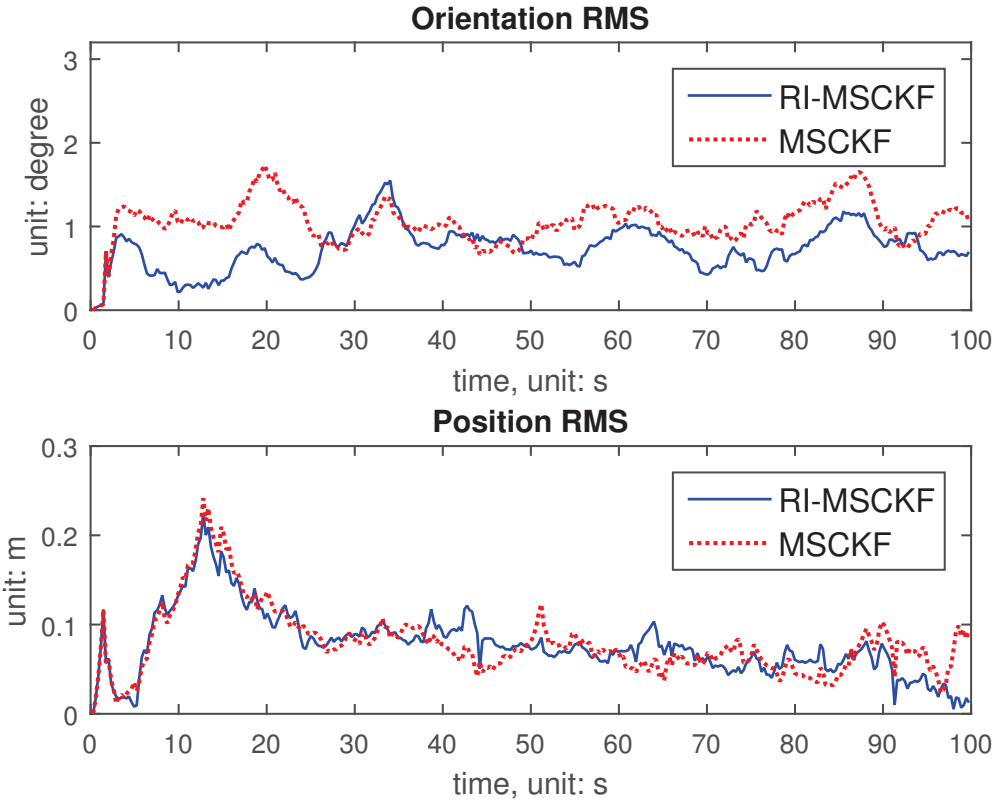

FIGURE 4.5: The RMS of orientation and position estimate from MSCKF and RI-MSCKF using the Euroc dataset sequence *V2_01_easy*.

Chapter 5

Optimization in SLAM

Our goal in this chapter is to investigate this overlooked invariance property of the optimization based SLAM algorithm. The invariance property refers to the value of the cost function of the SLAM formulation is invariant under the rigid body transformation. Exploiting this invariance property can help to speed up the optimization process by reducing the number of iterations needed for converging to a locally optimal solution. Unfortunately, the conventional method using the naive retraction does not use the property. In this chapter, we propose a method that the estimate after each iteration is invariant under the rigid body transformation.

5.1 Introduction

The conventional EKF-SLAM algorithm has two major drawbacks. The first one is the possible inconsistency estimates due to the absence of invariance to stochastic rigid body transformation, which can be alleviated by using the novel retraction (e.g., RIEKF-SLAM) or enforcing the Jacobians (e.g., first estimates Jacobian EKF). However, the second drawback cannot be easily avoided–the covariance matrix or information matrix is increasingly dense due to marginalization, which leads to high computational cost.

The current *de-facto* approach formulates the back-end of SLAM as a nonlinear least squares form, also being a MAP estimation problem, in which factor graph is used to

inference the interdependence among node variables (robot poses and features). Furthermore, the appearance of the modern sparse solvers, such as g2o, Ceres and GTSAM is also a catalyst for the popularity of the nonlinear least squares form.

In this chapter, we point out an overlooked property of SLAM that distinguishes it from a generic nonlinear least squares problem. The cost function value of SLAM is invariant under stochastic rigid body transformation, which corresponds to the invariance of the SLAM system in Chapter 3. On the other hand, the Newton-based method (e.g., Gauss-Newton, Levenberg-Marquart and Powell's Dogleg) for solving the nonlinear least squares always require multiple iterations to obtain the global/local minimum. Therefore, we propose a simple algorithm that exploits the invariance property of SLAM by designing a novel retraction while maintaining the same computational complexity compared to the standard method. Our algorithm can be used together with two Newton-based iterative solver, Gauss-Newton and Powell's Dogleg. Our results suggest that using the novel retraction leads to a fast and reliable convergence as compared to the standard method.

5.2 SLAM Formulation

The EKF-SLAM sequentially estimate the system state by using the motion model to predict and the observation model to update. As comparison, Graph SLAM processes all measurements taken during a trajectory simultaneously to obtain an estimate.

State

Technically, the state $\mathbf{X}$ in Graph SLAM refers to all robot poses and the locations of all landmarks, expressed in the global frame:

$$\begin{aligned} \mathbf{X} &= (\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{f}_1, \cdots, \mathbf{f}_L) \\ &= (\mathbf{x}_i, \mathbf{f}_l) \end{aligned} \tag{5.1}$$

where $\mathbf{x}_i = (\mathbf{R}, \mathbf{p}) \in \mathbf{SE}(3)$ denotes the robot pose at time-step i and $\mathbf{f}_j$ denotes the location of the landmark l.

Factors

Graph SLAM usually includes two types of factors: the odometry factor and the general landmark factor.

- **Odometry factor** The odometry factor describes the relative pose between two consequent robot poses, which corresponds to the motion model in EKF-SLAM. The error function of odometry factor for robot pose i and robot pose $i+1$ is

$$g(\mathbf{x}_i, \mathbf{x}_{i+1}) = \log(\mathbf{x}_i \mathbf{u}_i \mathbf{x}_{i+1}^{-1}) \tag{5.2}$$

  with the information matrix $\mathbf{IM}_i$, where $\mathbf{u}_i \in \mathbb{SE}(3)$ is the measured relative pose.

- **General landmark factor** The general landmark factor provides the relationship between robot pose and landmark, which corresponds to the observation model in EKF-SLAM. The error function of visual factor for robot pose i and landmark l is

$$h(\mathbf{x}_i, \mathbf{f}_l) = \tilde{h}(\mathbf{R}_i^\intercal(\mathbf{f}_l - \mathbf{p}_i), \mathbf{z}_{il}) \tag{5.3}$$

  with the information matrix $\mathbf{IM}_{il}$. Note that the function $h(\cdot,\cdot)$ represents a general case. For example, if we use the camera, $\tilde{h}(\mathbf{R}_i^\intercal(\mathbf{f}_l - \mathbf{p}_i), \mathbf{z}_{il}) = \pi(\mathbf{R}_i^\intercal(\mathbf{f}_l - \mathbf{p}_i)) - \mathbf{z}_{il}$.

Nonlinear least squares form

With odometry factors and general landmark factors, the MAP estimate of SLAM can be given by

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}} \|f(\mathbf{X})\|^2 = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i \|g(\mathbf{x}_i, \mathbf{x}_{i+1})\|^2_{\mathbf{IM}_i} + \sum_{i,l} \|h(\mathbf{x}_i, \mathbf{f}_l)\|^2_{\mathbf{IM}_{il}} \tag{5.4}$$

The Newton-based methods are used to solve the nonlinear least squares problem (Eq. 5.4), which need to linearize $f(\mathbf{X}) \approx f(\mathbf{X}^{(n)}) + \mathbf{F}\mathbf{e}^{(n)}$ where $\mathbf{X}^{(n)}$ denotes the estimate after n iterations. Note that the Jacobian matrix $\mathbf{F}$ depends on the *user-defined* retraction $\oplus$,

namely,

$$\mathbf{F} = \frac{\partial f(\mathbf{X}^{(n)} \oplus \mathbf{e})}{\partial \mathbf{e}}|_{\mathbf{e}=\mathbf{0}} \tag{5.5}$$

The Gauss-Newton method for solving the optimization problem (5.4) is summarized in Alg. 8.

Algorithm 8: Solving Eq. 5.4 by using the Gauss-Newton Method

Input: the initial guess $\mathbf{X}^{(0)}$
Output: the local minimum $\mathbf{X}^*$
Process:
$n \leftarrow 0$;
while $\mathbf{X}^{(n)}$ *does not converge* **do**
  $\mathbf{H} \leftarrow \mathbf{F}^\intercal\mathbf{F}$;
  solving the normal equation $\mathbf{H}\mathbf{e}^{(n)} = -\mathbf{F}^\intercal f(\mathbf{X}^{(n)})$;
  retract: $\mathbf{X}^{(n+1)} \leftarrow \mathbf{X}^{(n)} \oplus \mathbf{e}^{(n)}$;
  $n \leftarrow n + 1$;
end
$\mathbf{X}^* \leftarrow \mathbf{X}^{(n)}$;

Definition 5.1. For SLAM problem (5.4), a rigid body transformation $\mathcal{T}_\mathbf{g}$, associated with $\mathbf{g} \in \mathbb{SE}(3)$, is a mapping

$$\mathcal{T}_\mathbf{g}(\mathbf{X}) = (\mathbf{g}\mathbf{x}_i, \mathbf{g}\mathbf{f}_l). \tag{5.6}$$

5.3 Discussion for invariance

For the SLAM problem, all measurement from sensors can only provide relative information instead of global information. This property encodes the invariance to rigid body transformation.

Theorem 5.2. *The cost function of the SLAM problem (5.4) is invariant under any rigid body transformation $\mathcal{T}_\mathbf{g}$. And we have*

$$f(\mathbf{X}) = f(\mathcal{T}_\mathbf{g}(\mathbf{X})). \tag{5.7}$$

Proof.

$$\begin{aligned} f(\mathcal{T}_{\mathbf{g}}(\mathbf{X})) &= \|g(\mathcal{T}_{\mathbf{g}}(\mathbf{x}_i), \mathcal{T}_{\mathbf{g}}(\mathbf{x}_{i+1}))\|^2_{\mathbf{IM}_i} + \sum_{i,l} \|h(\mathcal{T}_{\mathbf{g}}(\mathbf{x}_i), \mathcal{T}_{\mathbf{g}}(\mathbf{f}_l))\|^2_{\mathbf{IM}_{il}} \\ &= \sum_{i} \|g(\mathbf{x}_i, \mathbf{x}_{i+1})\|^2_{\mathbf{IM}_i} + \sum_{i,l} \|h(\mathbf{x}_i, \mathbf{f}_l)\|^2_{\mathbf{IM}_{il}} \\ &= f(\mathbf{X}) \end{aligned} \tag{5.8}$$

□

The theorem above mathematically describes a fact that the SLAM problem (5.4) is up to a rigid body transformation, i.e., there are infinite global minima for the nonlinear least squares (5.4): if $\mathbf{X}^*$ is one global minima, $\mathcal{T}_{\mathbf{g}}(\mathbf{X}^*)$ would also be a global minimum for any arbitrary $\mathbf{g} \in \mathbb{SE}(3)$. Therefore, we usually choose a pose as an anchor and then perform optimization by using Gauss-Newton (Levenberg-Marquardt/Powell's Dogleg).

Naturally there is a question: is the estimate in each iteration also invariant under rigid body transformation when using Newton-based method to optimize? Let us strictly describe this question. Given two initial guesses $\mathbf{X}^{(0)}$ and $\mathbf{Y}^{(0)} = \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(0)})$, the estimates after n iterations of a Newton-based method will be $\mathbf{X}^{(n)}$ and $\mathbf{Y}^{(n)}$. Can we verify $\mathbf{Y}^{(n)} = \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(n)})$ or $\mathbf{Y}^{(n)} \neq \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(n)})$? Before providing the answer for this question, let us highlight the significance of this invariance property. An intuition is that solving a nonlinear least squares for SLAM problem is like finding a path from the foot of the mountain (initial guess) to the top of the mountain (global minimum). Obviously, we hope that a method for determining the optimal path is geometric and physical and does not necessarily depend on the parameterization. In the SLAM problem, the measurements do not contain any global information that is only passed from the fixed value of $\mathbf{x}_1$. Therefore we also expect that the choice of the fixed value of $\mathbf{x}_1$ also should have no any actual impact on the inference in each iteration when using Newton-based method. Otherwise the meaningless and spurious information will be introduced such that optimization needs more iterations or is easier to get stuck in a local minimum.

We know the degree of freedom of SLAM problem is 6, corresponding to an unobservable transformation–rigid body transformation. Another question is: can we fully decouple

the unobservable transformation $\mathcal{T}_g$ when performing retraction step? If not, the convergence of the estimation process would be affected by this spurious information when using Newton-based methods. In fact, the unobservable transformation can be decoupled when the null space of the Jacobian matrix $\mathbf{F} := \frac{\partial f(\mathbf{X}\oplus\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{0}}$ is invariant.

One can see that the Newton-based methods always need to calculate the Jacobian $\mathbf{F}$ that is related to the *user-defined* retraction $\oplus$. Thus the chosen $\oplus$ is indeed related to the invariance property. The following is the related theorem.

Theorem 5.3. *For the SLAM problem (5.4), given two initial guess* $\mathbf{X}^{(0)}$ *and* $\mathbf{Y}^{(0)} = \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(0)})$*, the estimates* $\mathbf{X}^{(n)}$ *and* $\mathbf{Y}^{(n)}$ *from Gauss-Newton are invariant under rigid body transformation, namely,* $\mathbf{Y}^{(n)} = \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(n)})$ *for arbitrary* $\mathbf{g} \in \mathbb{SE}(3)$ *if and only if there is a matrix* $\mathbf{Q}_{\mathbf{g}}$ *such that*

$$\mathcal{T}_{\mathbf{g}}(\mathbf{X} \oplus \mathbf{e}) = \mathcal{T}_{\mathbf{g}}(\mathbf{X}) \oplus \mathbf{Q}_{\mathbf{g}}\mathbf{e}. \tag{5.9}$$

If the estimates are from Levenberg-Marquart, we have $\mathbf{Y}^{(n)} \neq \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(n)})$ *in general. In addition, the estimates from Dogleg and the null space of the Jacobian matrix* $\mathbf{F}_{\mathbf{X}} := \frac{\partial f(\mathbf{X}\oplus\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{0}}$ *are invariant if and only if*

$$\mathbf{N} := \left.\frac{\partial \mathcal{T}_{\boldsymbol{\epsilon}}(\mathbf{X}) \ominus \mathbf{X}}{\partial \boldsymbol{\epsilon}}\right|_{\boldsymbol{\epsilon}=\mathbf{0}} \tag{5.10}$$

does not depend on $\mathbf{X}$*, where* $\mathcal{T}_{\boldsymbol{\epsilon}} = (\exp(\boldsymbol{\epsilon}_1), \boldsymbol{\epsilon}_2) \in \mathbb{SE}(3)$ *represents a small rigid body transformation.*

Proof. Here we just need to check $\mathbf{X}^{(1)}$ and $\mathbf{Y}^{(1)}$. Consider the case (5.9) is satisfied. When using Gauss-Newton,

$$\begin{aligned}
\mathbf{Y}^{(1)} &= \mathbf{Y}^{(0)} \oplus \mathbf{e}_y^{(0)} \\
&= \mathbf{Y}^{(0)} \oplus (-(\mathbf{F}_{\mathbf{Y}^{(0)}}^{\intercal}\mathbf{F}_{\mathbf{Y}^{(0)}})^{-1}\mathbf{F}_{\mathbf{Y}^{(0)}}^{\intercal} f(\mathbf{Y}^{(0)})) \\
&= \mathbf{Y}^{(0)} \oplus (-(\mathbf{Q}_{\mathbf{g}}^{-\intercal}\mathbf{F}_{\mathbf{X}^{(0)}}^{\intercal}\mathbf{F}_{\mathbf{X}^{(0)}}\mathbf{Q}_{\mathbf{g}}^{-1})^{-1}\mathbf{Q}_{\mathbf{g}}^{-\intercal}\mathbf{F}_{\mathbf{X}^{(0)}}^{\intercal} f(\mathbf{X}^{(0)})) \\
&= \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(0)}) \oplus (\mathbf{Q}_{\mathbf{g}}\mathbf{e}_x^{(0)}) \\
&= \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(0)} \oplus \mathbf{e}_x^{(0)}) \\
&= \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(1)})
\end{aligned} \tag{5.11}$$

Similarly, the invariance under Dogleg can be proved. However, the invariance when using LM cannot be guaranteed in general. When using LM,

$$\begin{aligned}
\mathbf{Y}^{(1)} &= \mathbf{Y}^{(0)} \oplus \mathbf{e}_y^{(0)} \\
&= \mathbf{Y}^{(0)} \oplus (\lambda \mathbf{I} + \mathbf{H}_{\mathbf{Y}^{(0)}})^{-1} \mathbf{F}_{\mathbf{Y}^{(0)}}^{\intercal} f(\mathbf{Y}^{(0)}) \\
&= \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(0)}) \oplus (\lambda \mathbf{I} + \mathbf{Q}_{\mathbf{g}}^{-\intercal} \mathbf{H}_{\mathbf{X}^{(0)}} \mathbf{Q}_{\mathbf{g}}^{-1})^{-1} \mathbf{Q}_{\mathbf{g}}^{-\intercal} \mathbf{F}_{\mathbf{X}^{(0)}}^{\intercal} f(\mathbf{X}^{(0)}) \\
&\neq \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(0)}) \oplus (\mathbf{Q}_{\mathbf{g}} \mathbf{e}_x^{(0)}) \\
&\neq \mathcal{T}_{\mathbf{g}}(\mathbf{X}^{(1)})
\end{aligned} \tag{5.12}$$

□

From the theorem above, one can see the invariance property can be held only when using Gauss-Newton or Powell's Dogleg and the retraction $\oplus$ satisfying (5.9).

Let us recall the conventional retraction $\oplus_c$ as below

$$\mathbf{X} \oplus_c \mathbf{e} = (\exp(\mathbf{e}_i)\mathbf{x}_i, \mathbf{f}_l + \mathbf{e}_l), \tag{5.13}$$

Note that the update of the landmark is independent of that of robot pose. Based on Theorem 5.9, it is easy to verify that for the SLAM problem (5.4), the whole process of estimate is not invariant under any rigid body transformation when using the retraction $\oplus_c$ and Levenberg-Marquart/Dogleg.

5.4 An invariant retraction

To achieve the invariance property, we propose "**Gauss-Newton/Dogleg + the retraction $\oplus_p$** ". In the proposed retraction, each landmark is coupled with a robot pose. If landmark l is first seen by the robot i, we choose the robot pose i as the "anchor" of the landmark l. our proposed retraction is defined as below

$$\mathbf{X} \oplus_p \mathbf{e} = (\exp(\mathbf{e}_i)\mathbf{x}_i, \exp(\mathbf{e}_i^r)\mathbf{f}_l + \mathrm{J}_r(-\mathbf{e}_i^r)\mathbf{e}_l), \tag{5.14}$$

where $\mathbf{e}_i = \begin{bmatrix} \mathbf{e}_i^r \\ \mathbf{e}_i^p \end{bmatrix} \in \mathbb{R}^6$.

Theorem 5.4. *For the SLAM problem (5.4), the whole process of estimate is invariant under rigid body transformation when using the retraction $\oplus_p$ and Gauss-Newton/Dogleg.*

5.5 Simulation results

In this section we report the performance of the proposed method on a synthetic case under different noise levels. To show the importance of the invariance property, we perform some simulations and compare our proposed invariant solver " GN/DL + $\oplus_p$ " to the solver "LM+$\oplus_p$ " and the solver "LM/DL+$\oplus_c$ ".

100 Monte Carlo simulations are performed using the configuration as in Section 3.6. All solvers are initialized using odometry data.

Table 5.1 summarizes the results obtained under different noise levels. As expected, all solvers tend to converge to local minima when the noise increases; a good initial guess is still important for the convergence. Nevertheless, according to Table 5.1 one can see

- The proposed invariant solver "DL+$\oplus_p$" significantly outperforms the solver "LM+$\oplus_p$" that is not invariant under rigid body transformation.
- The proposed invariant solver "GN/DL+$\oplus_p$" significantly outperforms the solver "LM/DL+$\oplus_c$" that is not invariant under rigid body transformation.

The proposed back-end, namely, "GN/DL+$\oplus_p$" leads to a faster and more reliable convergence than others.

5.6 Summary

In this chapter, we extended the concept of invariance into the optimization framework for the SLAM problem. We also investigated on the lack of invariance of the algorithm using

Noise Level	Solver	$iter_{global}$	n_{local}
$\sigma = 5\%$	LM+$\oplus_c$	19	18
	DL+$\oplus_c$	15	10
	GN+$\oplus_p$	7	12
	DL+$\oplus_p$	8	0
	LM+$\oplus_p$	20	18
$\sigma = 10\%$	LM+$\oplus_c$	43	47
	DL+$\oplus_c$	35	37
	GN+$\oplus_p$	15	6
	DL+$\oplus_p$	10	1
	LM+$\oplus_p$	25	24

TABLE 5.1: **Results from solvers under different noise levels**. $iter_{global}$ denotes the average iterations to converge to the global minimum; n_{local} is the number of diverge or converging to a local minima from a total of 100 simulation runs.

the commonly used retraction and proved that the invariance property can be held when using a novel retraction and Gauss-Newton/Dogleg. Monte Carlo simulations showed the convergence can be improved when using this method. In the next chapter, we will focus on the formulation of the VINS problem under the optimization framework.

Chapter 6

Optimization in VINS

While RIEKF-VINS can improve the performance of the conventional EKF-VINS, the (Factor Graph) optimization based algorithms for SLAM and VINS in recent implementations have been widely used due to their good performance and the efficiency of the modern sparse solvers. In this chapter, we analytically derive a novel invariant IMU factor motivated by the linearity of the error-state propagation of RI-EKF.

6.1 Introduction

In this chapter, we derive an invariant IMU factor for the VINS problem. Firstly, we formulate the vehicle state (orientation, position, linear velocity) in Lie group $\mathcal{G}(1)$ (introduced in Section 2.4) and describe the uncertainty in the tangent space. Then we derive that given IMU biases, the dynamics of the error vector between actual states and estimated states, in a short time interval can be safely approximated by a time-varing linear system with white Gaussian noise. In the optimization process, the time-varing IMU biases are also considered and we provide a unifying method to analytically calculate the Jacobians with respect to the vehicle state and IMU biases.

The main contribution is that we elegantly transform the VINS estimation problem from a highly nonlinear system into a time-varying linear system in the viewpoint of optimization while we also carefully deal with the IMU noises and IMU biases. In this way, our method

does not require a specific integration method (e.g., Euler integration), which allows more flexible operations. On the other hand, the linear dynamics avoids repeating integration after updating the state, which improves the computational efficiency.

6.2 Problem Formulation

IMU Measurement Model

Graph optimization allows repeated linearization while the EKF based filter methods only allows once linearization in the update stage. Thus, graph optimization typically outputs more accurate estimates than the EKF based filter method by iteratively considering all measurements at each step.

The IMU state can be represented by a tuple, i.e.,

$$\mathbf{X} = (\mathbf{R}, \mathbf{p}, \mathbf{v}, \mathbf{b}_g, \mathbf{b}_a) \tag{6.1}$$

where $(\mathbf{R}, \mathbf{p}) \in \mathbb{SE}(3)$ denotes the IMU pose,

$$\mathbf{v} := \mathbf{R}^\intercal \dot{\mathbf{p}} \in \mathbb{R}^3 \tag{6.2}$$

denotes the IMU velocity expressed in the local frame, $\mathbf{b}_g(t) \in \mathbb{R}^3$ and $\mathbf{b}_g(t) \in \mathbb{R}^3$ are the time-varying IMU biases following the random walk model:

$$\begin{aligned} \dot{\mathbf{b}_g} &= \mathbf{n}_{bg} \\ \dot{\mathbf{b}_a} &= \mathbf{n}_{ba} \end{aligned} \tag{6.3}$$

where $\mathbf{n}_{bg}$ and $\mathbf{n}_{ba}$ denote white Gaussian noises.

The IMU measurements at time t are

$$\begin{aligned} \mathbf{w}(t) &= S^{-1}(\mathbf{R}^\intercal(t)\dot{\mathbf{R}}(t)) + \mathbf{b}_g(t) + \mathbf{n}_g, \\ \mathbf{a}(t) &= \mathbf{R}^\intercal(t)(\ddot{\mathbf{p}}(t) - \mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a \end{aligned} \tag{6.4}$$

where $\mathbf{w}(t) \in \mathbb{R}^3$ and $\mathbf{a}(t) \in \mathbb{R}^3$ are the gyroscpoe reading and acclerometer reading, respectively, $\mathbf{n}_g \in \mathbb{R}^3$ and $\mathbf{n}_a \in \mathbb{R}^3$ are white Gaussian noises and $\mathbf{g} \in \mathbb{R}^3$ denotes the gravity vector in the global frame.

According to (6.1)–(6.4) , the dynamics of the IMU state $\mathbf{X}$ has a more concise form:

$$\begin{aligned}\dot{\mathbf{X}} =& F(\mathbf{X}, \mathbf{u}, \mathbf{n}) \\ =& (\mathbf{R}S(\mathbf{w} - \mathbf{b}_g - \mathbf{n}_g), \mathbf{R}(\mathbf{a} - \mathbf{b}_a - \mathbf{n}_a) + \mathbf{g}, \mathbf{v}, \mathbf{n}_{bg}, \mathbf{n}_{ba})\end{aligned} \tag{6.5}$$

where $\mathbf{u} := \begin{bmatrix}\mathbf{w} \\ \mathbf{a}\end{bmatrix}$ are the IMU measurements and $\mathbf{n} := \begin{bmatrix}\mathbf{n}_g \\ \mathbf{n}_a \\ \mathbf{n}_{bg} \\ \mathbf{n}_{ba}\end{bmatrix}$ are white Gaussian noise with the known covariance $\mathbf{\Sigma} \in \mathbb{R}^{12\times 12}$.

Monocular Camera Measurement Model

In this section, we introduce a new monocular camera measurement model instead of the conventional one (4.3), which is

$$\pi_{new}(\mathbf{z}_{k,l}, \mathbf{R}_k, \mathbf{p}_k, \mathbf{f}_l) = N(\mathbf{R}_k^\intercal(\mathbf{f}_l - \mathbf{p}_k)) - N(\mathbf{K}^{-1}\mathbf{z}_{k,l}) \sim \mathcal{N}(\mathbf{0}, \sigma_{k,l}\mathbf{I}_3) \tag{6.6}$$

where $\mathbf{f}_l$ denotes the location of the landmark l in the global frame and $\mathbf{z}_{k,l}$ denotes the uv measurement of the landmark l at the time step k.

Later, we will analyze the advantages and differences between the new measurement model and the conventional one.

VINS MAP estimate

Unlike the EKF based methods, the graph optimization method computes the MAP estimate of the states (the IMU states and the locations of landmarks) given all measurements over a period of time. More specifically, the factor graph of VINS typically includes three sorts of factors: prior factor, IMU factor and monucular camera measurement factor.

The MAP estimate of the VINS problem is

$$\begin{aligned}\mathcal{X}^* &= \arg\max_{\mathcal{X}} p(\mathcal{X}|\mathbf{z},\mathbf{u}) \\ &= \arg\max_{\mathcal{X}} p(\mathbf{X}_0)(\prod_k p(\mathbf{X}_k|\mathbf{X}_{k-1},\mathbf{u}_{k-1:k}))(\prod_{k,l} p(\mathbf{X}_k,\mathbf{f}_l|\mathbf{z}_{k,l}))\end{aligned} \tag{6.7}$$

where $\mathcal{X} = \{\mathbf{X}_k, \mathbf{f}_l\}$ includes the observed landmarks and the IMU states from time step 0 to N, $\mathbf{u}$ and $\mathbf{z}$ denotes all IMU measurements and camera measurements, respectively.

The concept of prior factor and monucular camera measurement factor is very clear and their probabilistic density functions follow Gaussian distributions. If we can derive that the IMU factor's probabilistic density function also follows a Gaussian distribution, as discussed in Section 2.6, the MAP estimate inference can be transformed into a corresponding nonlinear least sqaures problem:

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmin}} \|\mathbf{r}_0\|^2_{\mathbf{P}_0^{-1}} + \sum_k \|\mathbf{r}_k\|^2_{\mathbf{P}_k^{-1}} + \sum_{k,l} \|\mathbf{r}_{k,l}\|^2_{\sigma^{-1}\mathbf{I}_3}, \tag{6.8}$$

In the next section, we will show that **the IMU factor's probabilistic density function indeed can be approximated by a Gaussian distribution**.

6.3 Invariant IMU factor

In this section, we will prove that the probabilistic density function $p(\mathbf{X}_k|\mathbf{X}_{k-1},\mathbf{u}_{k-1:k})$ of the IMU factor can be approximated by a Gaussian distribution. More specifically, we will derive a novel invariant IMU factor and its covariance. The Jacobians of the error function will also be given.

Error-state definition and the error-state propagation

Recall the dynamic equation (6.5) of the IMU state:

$$\dot{\mathbf{X}} = F(\mathbf{X},\mathbf{u},\mathbf{n}). \tag{6.9}$$

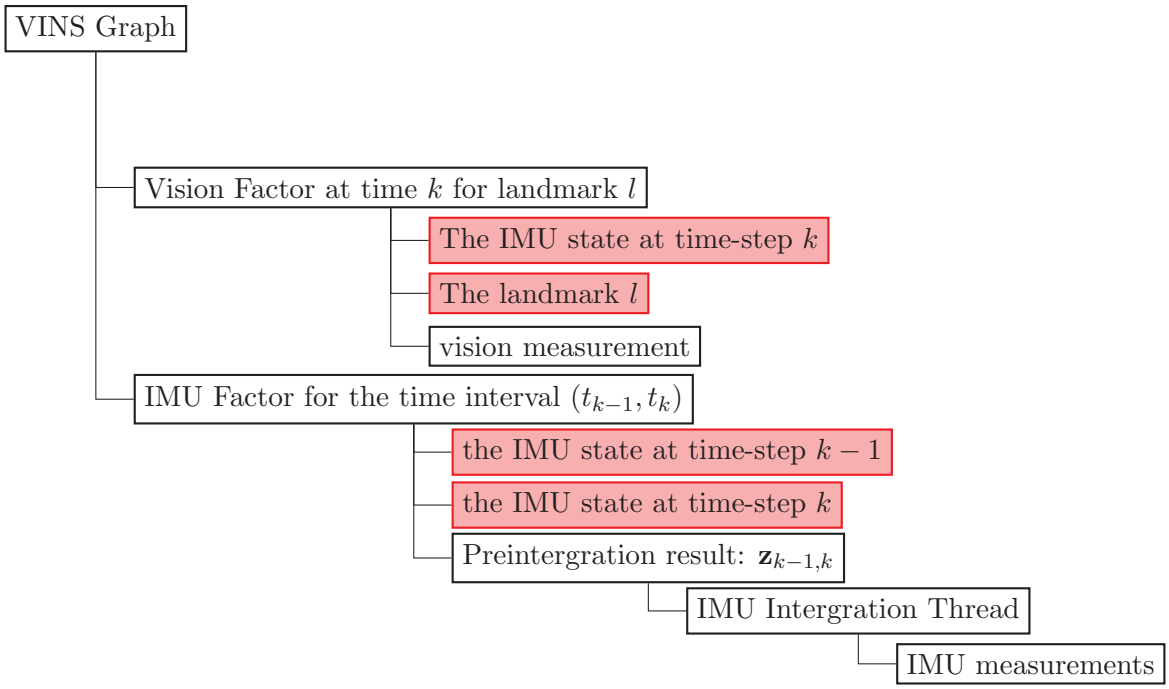

FIGURE 6.1: The VINS optimization framework. The red blocks represent the variables to be optimized.

Ignoring the IMU noises, we have the nominal dynamic model as below

$$\dot{\hat{\mathbf{X}}} = F(\hat{\mathbf{X}}, \mathbf{u}, \mathbf{0}). \tag{6.10}$$

Note that the IMU state $\mathbf{X} = (\mathbf{R}, \mathbf{V}, \mathbf{p}, \mathbf{b}_g, \mathbf{b}_a)$ can be regarded as an element in $\mathcal{G}(1) \times \mathbb{R}^6$, i.e.,

$$\begin{aligned} \mathbf{x} &:= (\mathbf{R}, \mathbf{V}, \mathbf{p}) \in \mathcal{G}(1) \\ \mathbf{b} &:= (\mathbf{b}_g, \mathbf{b}_a) \in \mathbb{R}^6 \end{aligned} \tag{6.11}$$

Therefore, we can define a difference term $\mathbf{E} \in \mathcal{G}(1) \times \mathbb{R}^6$ between the actual state $\mathbf{X}$ and the nominal state $\hat{\mathbf{X}}$:

$$\mathbf{E} = (\mathbf{E}_1, \Delta\mathbf{b}) = (\mathbf{x}^{-1}\hat{\mathbf{x}}, \mathbf{b} - \hat{\mathbf{b}}) \tag{6.12}$$

From (6.9) and (6.11) and the first-order approximation, we can obtain the dynamics of the error term $\mathbf{E}$:

$$\begin{aligned} \dot{\mathbf{E}}_1 &\approx g(\dot{\mathbf{E}}, \mathbf{n}_g, \mathbf{n}_a) \\ \dot{\Delta\mathbf{b}} &= (\mathbf{n}_{bg}, \mathbf{n}_{ba}) \end{aligned} \tag{6.13}$$

where

$$g(\dot{\mathbf{E}}, \mathbf{n}_g, \mathbf{n}_a) = \begin{pmatrix} \mathbf{R}^\intercal \hat{\mathbf{R}} S(\hat{\mathbf{w}}) - S(\hat{\mathbf{w}}) \mathbf{R}^\intercal \hat{\mathbf{R}} - S(\mathbf{n}_g + \Delta \mathbf{b}_g) \mathbf{R}^\intercal \hat{\mathbf{R}} \\ \mathbf{R}^\intercal \hat{\mathbf{R}} \hat{\mathbf{a}} - \hat{\mathbf{a}} + S(\hat{\mathbf{w}}) \mathbf{R}^\intercal (\hat{\mathbf{V}} - \mathbf{V}) - S(\mathbf{n}_g + \Delta \mathbf{b}_g) \mathbf{R}^\intercal (\hat{\mathbf{V}} - \mathbf{V}) - (\mathbf{n}_a + \Delta \mathbf{b}_a) \\ \mathbf{R}^\intercal (\hat{\mathbf{V}} - \mathbf{V}) - S(\hat{\mathbf{w}}) \mathbf{R}^\intercal (\hat{\mathbf{P}} - \mathbf{P}) - S(\mathbf{n}_g) \mathbf{R}^\intercal (\hat{\mathbf{P}} - \mathbf{P}) \end{pmatrix} \tag{6.14}$$

and

$$\begin{aligned} \hat{\mathbf{w}} &:= \mathbf{w} - \hat{\mathbf{b}}_g \\ \hat{\mathbf{a}} &:= \mathbf{a} - \hat{\mathbf{b}}_a \end{aligned} \tag{6.15}$$

Note that without the IMU biases, the dynamics (6.13) of the error term $\mathbf{E}$ is **autonomous**, i.e., the dynamics equation only depends on the current inputs, neither the estimated states nor the history of inputs. Let us define an error-state $\mathbf{e}$ in Euclidean space between the actual state $\mathbf{X}$ and the nominal state $\hat{\mathbf{X}}$:

$$\mathbf{e} = \begin{bmatrix} \log(\mathbf{E}_1) \\ \Delta \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 \\ \Delta \mathbf{b} \end{bmatrix} \in \mathbb{R}^{15} \tag{6.16}$$

Therefore, we can employ the retraction of the IMU state as

$$\mathbf{X} = \hat{\mathbf{X}} \oplus \mathbf{e} = (\exp(\mathbf{e}_1) \hat{\mathbf{x}}, \mathbf{b} + \Delta \mathbf{b}) \tag{6.17}$$

According to (6.13), (6.14) and the theorem from Lie algebra, we can easily compute the dynamics of the error-state $\mathbf{e} \in \mathbb{R}^{15}$:

$$\dot{\mathbf{e}} \approx \mathbf{A}\mathbf{e} + \mathbf{B}\mathbf{n} \tag{6.18}$$

where

$$\mathbf{A} = \begin{bmatrix} -S(\hat{\mathbf{w}}) & \mathbf{0} & \mathbf{0} & -\mathbf{I}_3 & \mathbf{0} \\ -S(\hat{\mathbf{a}}) & -S(\hat{\mathbf{w}}) & \mathbf{0} & \mathbf{0} & -\mathbf{I}_3 \\ \mathbf{0} & \mathbf{I}_3 & -S(\hat{\mathbf{w}}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \tag{6.19}$$

and

$$\mathbf{B} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \tag{6.20}$$

The covariance $\mathbf{P}$ of $\mathbf{e}$ can be computed as

$$\dot{\mathbf{P}} = \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^\intercal + \mathbf{B}cov(\mathbf{n})\mathbf{B}^\intercal \tag{6.21}$$

From the derivation above, we can see the dynamics of the error-state $\mathbf{e} \in \mathbb{R}^{15}$ is a time-varying linear system in the sense of the first-order approximation. The error-state propagation (6.18) implies that the probabilistic density function of the IMU factor follows a Gaussian distribution, i.e.,

$$-\log(p(\mathbf{X}_k|\mathbf{X}_{k-1}, \mathbf{u}_{k-1:k})) \propto \|\mathbf{e}_k\|^2_{\mathbf{P}_k^{-1}} \tag{6.22}$$

where

$$\begin{aligned} \mathbf{e}_k &= \mathbf{X}_k \ominus \hat{\mathbf{X}}_k \\ &= \begin{bmatrix} \log(\mathbf{x}_k^{-1}\hat{\mathbf{x}}_k) \\ \mathbf{b}_k - \hat{\mathbf{b}_k} \end{bmatrix} \in \mathbb{R}^{15} \end{aligned} \tag{6.23}$$

refers to the error between the actual $\mathbf{X}_k$ and the nominal propagation state $\hat{\mathbf{X}}_k$. Note that the nominal propagation state $\hat{\mathbf{X}}_k$ can be computed by using the nominal state propagation (6.10) and the start point $\hat{\mathbf{X}}_{k-1}$. Similarly, the covariance $\mathbf{P}_k \in \mathbb{R}^{15\times15}$ can be computed by using the covariance propagation (6.21) and the start point $\mathbf{P} = \mathbf{0}$.

Pre-integration method

This subsection introduces a pre-integration method to analytically derive the error function $\mathbf{e}_k$ of the IMU factor and the corresponding Jacobians.

Considering $\mathbf{e}_k = \mathbf{X}_k \ominus \hat{\mathbf{X}}_k$, we need to quickly compute the nominal propagation state $\hat{\mathbf{x}}_k$. Given the measurement $\mathbf{u}_{k-1:k}$, we denote the propagation as a mapping $\mathbf{f}(\cdot)$ and thus we have

$$\hat{\mathbf{X}}_k = \mathbf{f}(\hat{\mathbf{X}}_{k-1}) \tag{6.24}$$

In the following, we will gradually provide the expression of $\mathbf{f}(\cdot)$.

- On the one hand, we have

$$\mathbf{f}(\mathbf{x}, \hat{\mathbf{b}}) = (\mathbf{R}\mathbf{J}_1, \mathbf{V} + \mathbf{R}\mathbf{J}_2 + \mathbf{g}\Delta\mathbf{t}_k, \mathbf{p} + \frac{\Delta\mathbf{t}_k^2}{2}\mathbf{g} + \Delta\mathbf{t}_k\mathbf{V} + \mathbf{R}\mathbf{J}_3, \hat{\mathbf{b}}) \tag{6.25}$$

  where $\Delta\mathbf{t}_k$ denotes the time interval between time-step $k-1$ and k, $\hat{\mathbf{b}}$ denotes the nominal IMU biases used in propagation, $(\mathbf{J}_1, \mathbf{J}_2, \mathbf{J}_3) \in \mathcal{G}(1)$ can be pre-integrated by using the following differential equations

$$\begin{aligned} \dot{\mathbf{J}}_1 &= \mathbf{J}_1 S(\mathbf{w} - \hat{\mathbf{b}}_g) \\ \dot{\mathbf{J}}_2 &= \mathbf{J}_1 S(\mathbf{a} - \hat{\mathbf{b}}_a) \\ \dot{\mathbf{J}}_3 &= \mathbf{J}_2 \end{aligned} \tag{6.26}$$

  with the start point $(\mathbf{J}_1, \mathbf{J}_2, \mathbf{J}_3)|_{t_k} = (\mathbf{I}_3, \mathbf{0}, \mathbf{0})$. For more accuracy, a Runge-Kutta 4-th order method on the Lie group can be used to perform this pre-integration.

- Note that the error-state propagation (6.18) is a time-varying linear system. The linear property allows some simple calculations to compute $\mathbf{f}(\mathbf{X})$:

$$\begin{aligned} \mathbf{f}(\mathbf{X}) &= \mathbf{f}(\mathbf{x}, \mathbf{b}) \\ &= \mathbf{f}(\mathbf{x}, \hat{\mathbf{b}}) \exp(\mathbb{A} \begin{bmatrix} \mathbf{0} \\ \mathbf{b} - \hat{\mathbf{b}} \end{bmatrix}) \end{aligned} \tag{6.27}$$

  where $\mathbb{A}$ can be computed from the following differential equation

$$\dot{\mathbb{A}} = \mathbf{A}\mathbb{A} \tag{6.28}$$

with the start point $\mathbb{A}|_{t_{k-1}} = \mathbf{I}_{15}$. More specifically,

$$\mathbb{A} = \begin{bmatrix} \mathbf{J}_1^\intercal & \mathbf{0} & \mathbf{0} & \mathbf{J}_4 & \mathbf{0} \\ -\mathbf{J}_1^\intercal S(\mathbf{J}_2) & \mathbf{J}_1^\intercal & \mathbf{0} & \mathbf{J}_5 & \mathbf{J}_4 \\ -\mathbf{J}_1^\intercal S(\mathbf{J}_3) & -\Delta t_k \mathbf{J}_1^\intercal & \mathbf{J}_1^\intercal & \mathbf{J}_6 & \mathbf{J}_7 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{bmatrix} = \begin{bmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} \\ \mathbb{A}_{21} & \mathbb{A}_{22} \end{bmatrix} \tag{6.29}$$

where $(\mathbf{J}_4, \cdots, \mathbf{J}_7)$ can be pre-integrated using the following differential equation

$$\begin{aligned} \dot{\mathbf{J}}_4 &= -S(\hat{\mathbf{w}}_t)\mathbf{J}_4 - \mathbf{I}_3 \\ \dot{\mathbf{J}}_5 &= -S(\hat{\mathbf{a}})\mathbf{J}_4 - S(\hat{\mathbf{w}})\mathbf{J}_5 \\ \dot{\mathbf{J}}_6 &= \mathbf{J}_5 - S(\hat{\mathbf{w}})\mathbf{J}_6 \\ \dot{\mathbf{J}}_7 &= \mathbf{J}_4 - S(\hat{\mathbf{w}})\mathbf{J}_7 \end{aligned} \tag{6.30}$$

with the start point $(\mathbf{J}_4, \cdots, \mathbf{J}_7)|_{t_{k-1}} = (\mathbf{0}, \cdots, \mathbf{0})$.

Now we have analytically derived the propagation function $\mathbf{f}(\cdot)$ and the continuous IMU measurements $\mathbf{u}_{k-1:k}$ have been embedded into the pre-integration $(\mathbf{J}_1, \cdots, \mathbf{J}_7)$ and covariance matrix $\mathbf{P}_k$. Therefore, the IMU factor does not need to keep the huge $\mathbf{u}_{k-1:k}$ but only keep the concise pre-integration $(\mathbf{J}_1, \cdots, \mathbf{J}_7)$.

According to (6.23), the error function of the IMU factor between time-step $k-1$ and k can be given by

$$h(\mathbf{X}_{k-1}, \mathbf{X}_k, \mathbf{z}_{k-1,k}) = \begin{bmatrix} \mathbf{e}_{rvp} \\ \mathbf{e}_b \end{bmatrix} = \begin{bmatrix} \log(\mathbf{x}_k^{-1}\mathbf{f}_1(\mathbf{x}_{k-1})) \\ \mathbf{b}_k - \mathbf{b}_{k-1} \end{bmatrix} \tag{6.31}$$

where $\mathbf{f}_1$ denotes the propagation mapping of IMU pose and velocity. Due to the linear error-state propagation, the Jacobians of the error function can be elegantly computed.

- $$\begin{aligned}\mathbf{H}_{\mathbf{X}_{k-1}} &= \frac{\partial \mathbf{X}_k \ominus \mathbf{f}(\mathbf{X}_{k-1} \oplus \mathbf{dx})}{\partial \mathbf{dx}}|_{\mathbf{dx}=\mathbf{0}} \\ &= \begin{bmatrix} \mathbf{H}_{\mathbf{x}_{k-1}} \\ \mathbf{H}_{\mathbf{b}_{k-1}} \end{bmatrix} \in \mathbb{R}^{15\times 15}\end{aligned} \tag{6.32}$$

  where $\mathbf{H}_{\mathbf{x}_{k-1}} = \begin{bmatrix} J_r^{-1}(-\mathbf{e}_{rvp})\mathbb{A}_{11} ad(\mathbf{x}_{k-1}^{-1}) & J_r^{-1}(-\mathbf{e}_{rvp})\mathbb{A}_{12} \end{bmatrix}$ and $\mathbf{H}_{\mathbf{b}_{k-1}} = \begin{bmatrix} \mathbf{0} & -\mathbf{I}_6 \end{bmatrix}$.

- $$\begin{aligned}\mathbf{H}_{\mathbf{X}_k} &= \frac{\partial (\mathbf{X}_k \oplus \mathbf{dx}) \ominus \mathbf{f}(\mathbf{X}_{k-1})}{\partial \mathbf{dx}}|_{\mathbf{dx}=\mathbf{0}} \\ &= \begin{bmatrix} \mathbf{H}_{\mathbf{x}_k} \\ \mathbf{I}_6 \end{bmatrix} \in \mathbb{R}^{15\times 15}\end{aligned} \tag{6.33}$$

  where $\mathbf{H}_{\mathbf{x}_k} = \begin{bmatrix} J_r^{-1}(-\mathbf{e}_{rvp}) ad(\mathbf{f}_1^{-1}(\mathbf{x}_{k-1})) & \mathbf{0} \end{bmatrix}$ and $\mathbf{H}_{\mathbf{b}_k} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_6 \end{bmatrix}$.

Remark 6.1. $\mathbb{A}_{11}$ referes to the first 9 rows of $\mathbb{A}$ and $\mathbb{A}_{12}$ referes to the last 6 rows of $\mathbb{A}$.

6.4 A concise form of the IMU factor

This section summarizes the IMU factor derived in last section.

Pre-processing

The pre-processing is to mainly integrate the high frequent IMU measurements between time-steps $k-1$ and k into a concise form $\mathbf{z}_{k-1,k}$, which is summarized in Alg. 9.

Algorithm 9: The pre-processing

Input: the IMU measurements $\mathbf{u}_{k-1:k}$, the nominal IMU biases $\hat{\mathbf{b}}_k$ at time step k, the covariance matrix $cov(\mathbf{n})$ of the IMU noise

Output: The general measurements $\mathbf{z}_{k-1,k} = (\mathbf{J}_1, \cdots, \mathbf{J}_7, \hat{\mathbf{b}}_k)$ for the IMU factor and the covariance matrix $\mathbf{P}$ for the IMU factor

In Alg. 9, $(\mathbf{J}_1, \mathbf{J}_2, \cdots, \mathbf{J}_7)$ and $\mathbf{P}$ are pre-integrated by using the differential equations (6.26), (6.30) and (6.21) with the start point $(\mathbf{J}_1, \mathbf{J}_2, \cdots, \mathbf{J}_7)|_{t_{k-1}} = (\mathbf{I}_3, \mathbf{0}, \cdots, \mathbf{0})$ and $\mathbf{P}|_{t_{k-1}} = \mathbf{0}$.

IMU factor in factor graph

Here we present the IMU factor in a factor graph form. The information of the IMU factor for the time interval (t_{k-1}, t_k) is provided below

- Dimension of the error function: 15.
- The involved variable nodes: $\mathbf{X}_{k-1}$ and $\mathbf{X}_k \in \mathcal{G}(1) \times \mathbb{R}^6$ (the retraction $\oplus$ is defined in 6.17).
- The involved measurement: $\mathbf{z}_{k-1,k}$ from Alg. 9.
- The covariance matrix: $\mathbf{P}_{k-1,k}$ from Alg. 9.
- The error function: $h(\mathbf{X}_{k-1}, \mathbf{X}_k, \mathbf{z}_{k-1,k})$ as defined in (6.31).
- The Jacobians: $\mathbf{H}_{\mathbf{X}_{k-1}}$ and $\mathbf{H}_{\mathbf{X}_k}$ are given in (6.32) and (6.33).

Invariance of the proposed IMU factor

Here we highlight that the proposed IMU factor is invariant under the translation and the rotation about the gravitational direction. Therefore, the proposed IMU factor successfully captures the observability of the VINS system such that the unobservable transformation (i.e., the translation and the rotation about the gravitational direction) does not have any unexpected effect on the estimation.

Definition 6.2. The translation and the rotation about the gravitational direction can be expressed as a mapping $\mathcal{T}_{gt}$

$$\mathcal{T}_{gt}(\mathbf{X}) = (\mathbf{R}_g\mathbf{R}, \mathbf{R}_g\mathbf{v}, \mathbf{R}_g\mathbf{p} + \mathbf{t}, \mathbf{b}) \tag{6.34}$$

where $\mathbf{R}_g = \exp(\mathbf{g}\theta) \in \mathbb{SO}(3)$, $\theta \in \mathbb{R}^3$ and $\mathbf{t} \in \mathbb{R}^3$.

Theorem 6.3. *The null space of the Jacobian of the error function in the proposed IMU factor is invariant under $\mathcal{T}_{gt}$. The error function value of the proposed IMU factor is*

invariant under $\mathcal{T}_{gt}$:

$$h(\mathcal{T}_{gt}(\mathbf{X}_{k-1}), \mathcal{T}_{gt}(\mathbf{X}_k), \mathbf{z}_{k-1,k}) = h(\mathbf{X}_{k-1}, \mathbf{X}_k, \mathbf{z}_{k-1,k}) \tag{6.35}$$

where $h(\cdot,\cdot,\cdot)$ is the error function of the proposed IMU factor, given by (6.31).

Proof.

$$\begin{aligned} h(\mathcal{T}_{gt}(\mathbf{X}_{k-1}), \mathcal{T}_{gt}(\mathbf{X}_k), \mathbf{z}_{k-1,k}) &= \begin{bmatrix} \log((\mathcal{T}_{gt}\mathbf{x}_k^{-1})\mathbf{f}_1((\mathcal{T}_{gt}(\mathbf{x}_{k-1}))) \\ \mathbf{b}_k - \mathbf{b}_{k-1} \end{bmatrix} \\ &= \begin{bmatrix} \log(\mathbf{x}_k^{-1}\mathcal{T}_{gt}^{-1}\mathcal{T}_{gt}\mathbf{f}_1((\mathbf{x}_{k-1})) \\ \mathbf{b}_k - \mathbf{b}_{k-1} \end{bmatrix} \\ &= \begin{bmatrix} \log(\mathbf{x}_k^{-1}\mathbf{f}_1(\mathbf{x}_{k-1}) \\ \mathbf{b}_k - \mathbf{b}_{k-1} \end{bmatrix} \\ &= h(\mathbf{X}_{k-1}, \mathbf{X}_k, \mathbf{z}_{k-1,k}) \end{aligned} \tag{6.36}$$

□

6.5 Vision factor

Pure inertial measurements are not enough to infer the state without the initial prior knowledge and also results in drift even when the initial prior knowledge is provided, which is caused by the numerical issue on the integration and the IMU noises. In the EKF-based method, the visual measurements are used in the update stage. For the optimization-based method, all visual measurement are integrated into the estimation as the visual factors. When the sensor platform goes around well-structured and well-textured environments, images can be taken from the mounted camera and features are extracted and tracked as landmarks. This front-end (the image process including data association) traditionally employs a descriptor based method (such as **SIFT** [53], **SURF** [54], **BRIEF** [55] or **ORB**) [56], an optical flow based method (e.g., **KLT** [57]) or the combination of them.

After data association, the visual measurements are put into the optimization process as the vision factor. A visual factor includes:

- Dimension of the error function: 2.
- The involved variable nodes: $\mathbf{X}_k \in \mathcal{G}(1)$ and $\mathbf{f}_l \in \mathbf{R}^3 \times \mathbb{R}^3$.
- The involved measurement: $\mathbf{z}_{k,l} = \begin{bmatrix} u \\ v \end{bmatrix}$ from the front-end.
- The covariance matrix: typically using $\mathbf{I}_2$.
- The error function:

$$h(\mathbf{X}_k, \mathbf{f}_l) = \mathfrak{h}(\mathbf{R}_k^\intercal(\mathbf{f}_l - \mathbf{p}_k)) \tag{6.37}$$

  where $\mathfrak{h} = \pi \circ \mathbf{T}_{CI}$, π denotes the projection function and $\mathbf{T}_{CI}$ is the transformation from the IMU frame to the camera frame.

6.6 Compared to other IMU factors

To see the qualitative difference between our proposed IMU factor and other IMU factors, in this section we provide a simple numerical example to investigate the advantages of our proposed method.

We compare the following three pre-integration IMU factos:

- our proposed invariant IMU factor;
- the Euler angle based IMU factor (Euler) [4];
- the $\mathbb{SO}(3)$ based IMU factor (SO3) [6];
- the quaternion based IMU factor (Quaternion) [7];

Let us consider the case of

- The time-varing IMU state (ground truth) is

$$\mathbf{X}(t) = (\exp(\mathbf{w}(t)), \mathbf{p}(t), \mathbf{b}(t)) \tag{6.38}$$

where $\mathbf{w}(t) = \begin{bmatrix} 0.2\cos(t) \\ 0.3\sin(t) \\ mt \end{bmatrix}$, $\mathbf{p}(t) = \begin{bmatrix} 5\cos(mt) \\ 4\sin(mt) \\ 0.4\sin(mt) \end{bmatrix}$, $m = \frac{3}{20}\pi$ and $\mathbf{b}(0) = \mathbf{0} \in \mathbb{R}^6$ follows the dynamic equation $\frac{d}{dt}\mathbf{b} = \begin{bmatrix} \mathbf{n}_{bg} \\ \mathbf{n}_{ba} \end{bmatrix}$.

- The IMU noise covariance $\mathbf{Q}$ is set as $\alpha\mathbf{diag}(0.008^2\mathbf{I}_3, 0.0004^2\mathbf{I}_3, 0.019^2\mathbf{I}_3, 0.05^2\mathbf{I}_3)$.
- The IMU measurements are generated by using the ground truth of the trajectory corrupted with the noises. The IMU noises are randomly generated according to the IMU noise covariance $\mathbf{Q}$.

For independently proving the superiority of our proposed IMU factor, we do pre-integration by using all IMU measurements and get four pairs of $(\mathbf{J}, \mathbf{P})$ where $\mathbf{J}$ represents the pure propagated part about the state and $\mathbf{P}$ denotes the pre-integrated covariance matrix. Note that the values $(\mathbf{J}, \mathbf{P})$ of the different IMU factors are different because their preintegrations are different. To show the consistency and verify if the error function follows the pre-integrated covariance, we perform 50 Monte Carlo simulation and use *NEES* as the performance indicator where

$$NEES = \sum \frac{1}{15n} \|h(\mathbf{X}(0), \mathbf{X}(3), \mathbf{J})\|^2_{\mathbf{P}^{-1}} \tag{6.39}$$

where n is the number of simulation runs. Note that the corresponding IMU factor is better when *NEES* is closer to 1. The 50 Monte Carlo simulation results for 4 different IMU factors (under the condition of different noise levels) are summarized in Fig. 6.2.

From Fig. 6.2, we can see the proposed IMU factor is the best performing factor in terms of the consistency, especially when the IMU noise is relatively high. This result reflects that our IMU factor with the associated preintegation is a better description about the physical system. To explain this reason, we state the advantages of the proposed IMU factor as below:

- Non-singularity. Compared to the earliest preintegration (Euler angle based [4]) IMU factor, the proposed IMU factor does not have the singularity and does have the invariance property.

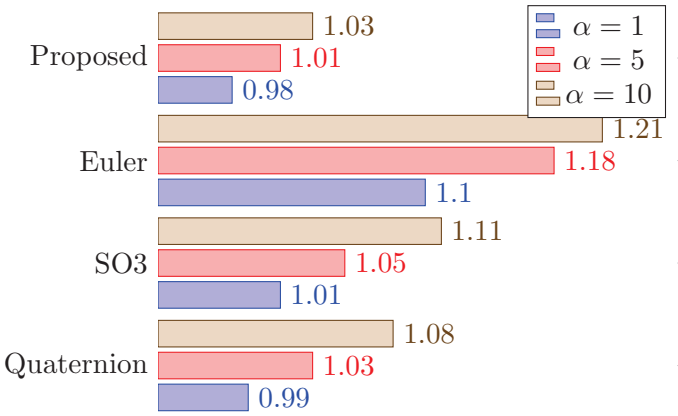

FIGURE 6.2: The *NEES* of IMU factors under conditions of different noise levels.

- Autonomous. The linearized error-state propagation equation (6.18) of the proposed IMU factor is autonomous; the matrices $\mathbf{A}(t)$ and $\mathbf{B}(t)$ only need to load the IMU measurements at the time t without the estimated states.

- Continuous. The linearized error-state propagation equation of the proposed IMU factor is a continuous form instead of the discrete form ([6]). Therefore, all preintegation terms can be processed via the high-order method (such as the RungeKutta 4-th order method). A low-cost IMU may get benefits from this feature.

- No need the assumption that the IMU biases are constant between two frames. Therefore, the proposed IMU factor has more accurate covariance matrix, compared to [6] [7].

6.7 Experiment

Monte Carlo simulations

Consider that a platform, equipped with an IMU (frequency: 200 Hz) and a camera (frequency: 10 Hz), moves along a specific trajectory (average speed is 2.5m/s) with the sufficient 6-DOFs motion, shown as in Fig. 6.5. In this environment, 675 landmarks are distributed on the surface of a cylinder with radius $6.5m$ and height $4m$ shown as the green circles in Fig. 6.5. To simulate a realistic feature tracker, the landmark measurements are

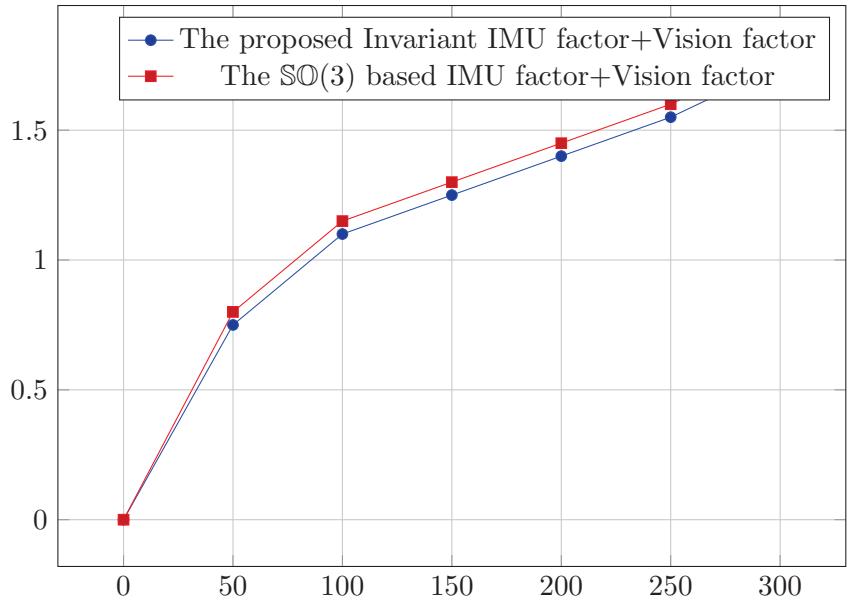

FIGURE 6.3: The RMS of orientation. The X axis is the time (unit: seconds) and the y axis is the RMS of orientation (unit: degree).

corrupted with isotropic Gaussian noise with standard deviation 1 pixel and the camera is able to observe sufficiently overlapped landmarks between consecutive frames. The loop closure is not included in this simulation. The IMU noise covariance $\mathbf{Q}$ is set as $\mathbf{diag}(0.008^2\mathbf{I}_3, 0.0004^2\mathbf{I}_3, 0.019^2\mathbf{I}_3, 0.05^2\mathbf{I}_3)$ (the International System of Units).

In each round of Monte Carlo simulation, the initial estimate is set as the ground truth. And the measurements from IMU and camera are generated from the same trajectory with random noises. We compute the MAP estimate using the proposed IMU factor and the vision factor presented in this chapter. The optimization is implemented in Matlab using the Graph-Optimization package $https://github.com/UTS-CAS/Matlab-Graph-Optimization$ that we developed.

The 50 Monte Carlo simulation results from our proposed method (the invariant IMU factor + Vision Factor) and the method ($\mathbb{SO}(3)$ based IMU factor [6]+ Vision Factor) are summarized in Fig. 6.3 and Fig. 6.4. From Fig. 6.3 and Fig. 6.4, one can see that our proposed method achieves better performance.

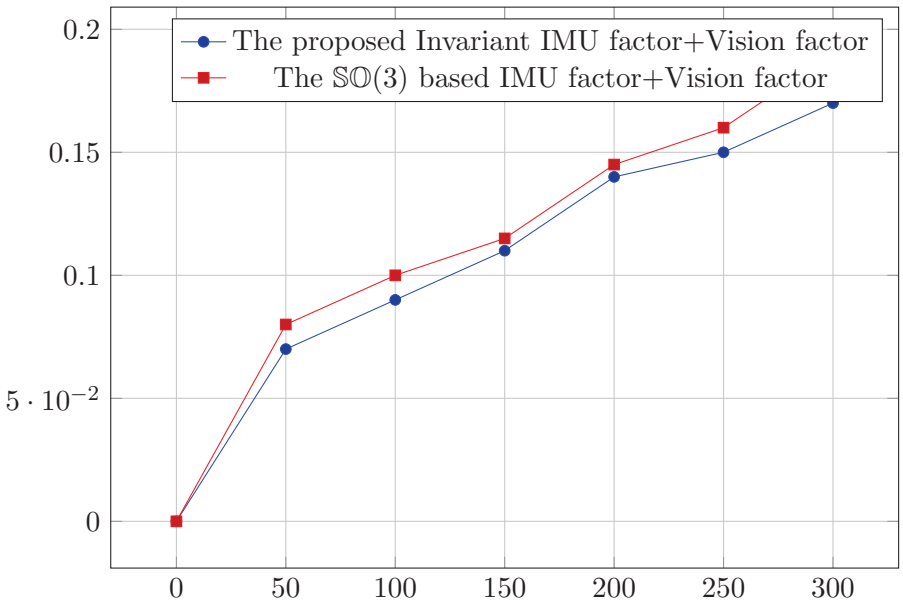

FIGURE 6.4: The RMS of position. The X axis is the time (unit: seconds) and the y axis is the RMS of position (unit: m).

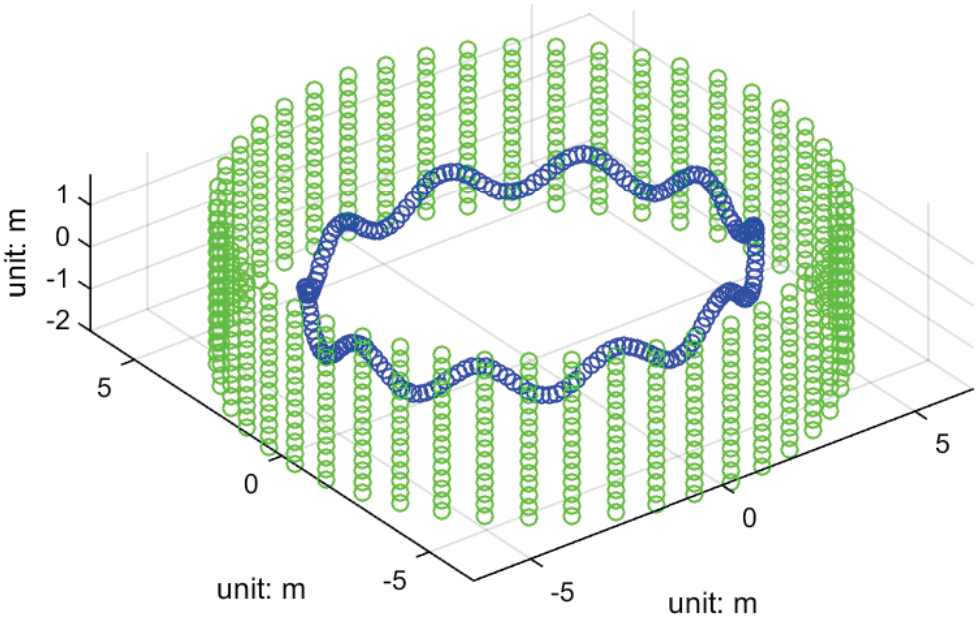

FIGURE 6.5: The simulated trajectory (blue circles) and landmarks (green circles).

Real-World Experiments

To further validate our proposed invariant IMU factor in the real system, we modify the ORB-VINS system [58] in 11 sequences of the EuroC dataset.

The Euroc dataset

There are 11 sequences in the EuroC dataset that are collected from a micro aerial vehicle (MAV), equipped with a global shutter WVGA stereo camera at 20 Hz and an IMU at 200 Hz. The environment of the V series is a laboratory of the area $30m^2$ and that of the MH series is a industrial environment of the area $30m^2$. Each series has the easy, medium and difficult sequences. Note that some challenging conditions such as strong illumination changes, poorly textured areas, motion-blur and aggressive motions are included in the difficult/medium sequences, as shown in Fig. 6.6. Besides, the trajectory ground-truth is also included. Therefore, the Euroc dataset is ideal to preliminarily test the VINS system and compare the algorithms.

ORB-VINS

The visual-inertial ORB-SLAM (ORB-VINS) [58] is a VINS system that simultaneously maintains the two data structures **Keyframe** and **Map Point** using the multi-threads **Tracking**, **Local Mapping** and **Loop Closing**.

- **Keyframe** and **Map Points** store the estimated IMU states and the estimated landmarks, respectively.
- The **Tracking** thread is in charge of data association, initialization, the insertion of keyframes, pre-integration and the motion-only optimization. The motion-only optimization estimates the IMU state at the current frame by fixing the connected variables.
- The **Local Mapping** thread is in charge of the creation/culling of **Map Point**, the culling of **Keyframe** and the local bundle adjustment. Local Bundle Adjustment

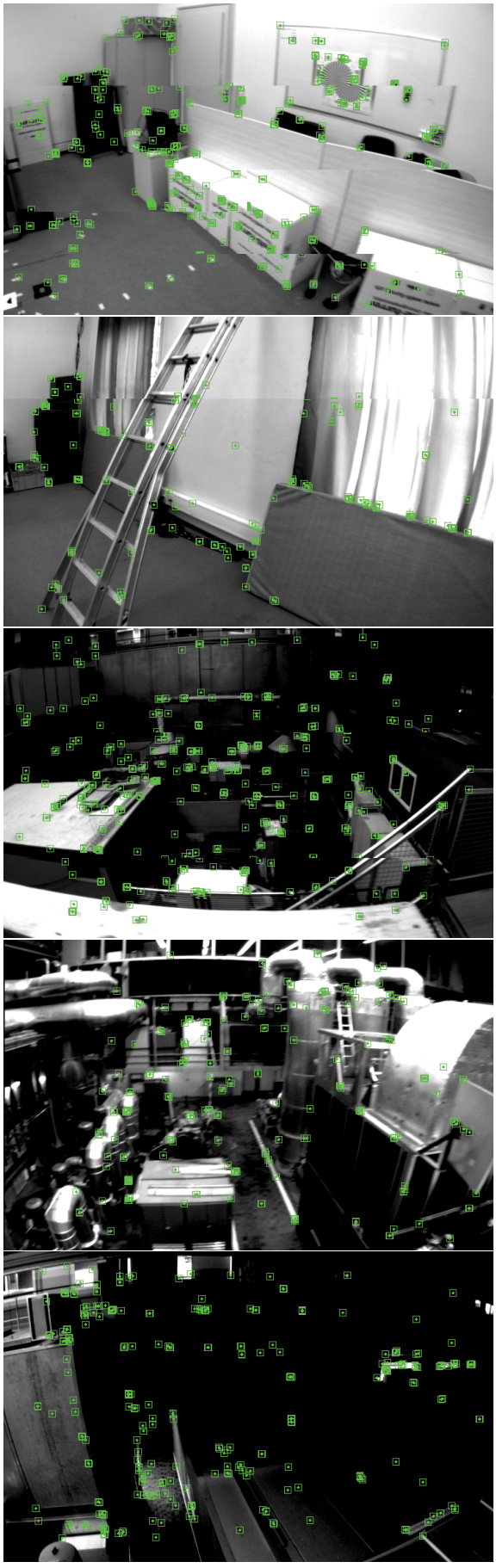

FIGURE 6.6: Some sample images with keypoints in the experiment. The green dots represent the tracked key points.

(BA) optimizes the IMU states at a local window and all connected landmarks, including the fixed IMU states at an outer window.

- The **Loop Closing** thread is in charge of detecting loop closure and performing global bundle adjustment to reduce the drift caused by the unobservable transformation.

The optimization processes mentioned above consists of the IMU factor and the vision factor.

Results

To validate the proposed IMU factor and the proposed vision factor in this chapter, we have tested three methods:

- The original method: the ORB-VINS using the $\mathbb{SO}(3)$ based IMU factor + standard Vision Factor.
- The proposed 1: the ORB-VINS using the proposed invariant IMU factor + standard Vision Factor.
- The proposed 2: the ORB-VINS using the proposed invariant IMU factor + the proposed Vision Factor.

Note that only left images are used. All experiments are processed in a computer with Intel Core i7 cpu and 16 GB RAM. The number of the local window is set as 10. The initialization time is set as 15s.

The estimated maps of $V1\_02$, $V2\_02$, MH_02 and MH_04 using the proposed method 2 are shown in Fig. 6.7 and Fig. 6.8. The estimated trajectories of MH_02 and MH_05 using the original method and the proposed method 2 are also presented in Fig. 6.9 and Fig. 6.10. From Fig. 6.9 and Fig. 6.10, we can see that the estimated trajectories by our proposed method 2 is more accurate than the original method and the method 1.

The experimental results of position accuracy are summarized in Table 6.1. From Table 6.1, one can see that the proposed method 2 behaves better than others in general. The experiments also show the advantages of our proposed factors in the real system. Note that the VINS system does not work well in the sequence in $V1\_03$ due to the texture-less environment and the too fast motion.

TABLE 6.1: **Accuracy (RMS, unit: m) in Euro dataset.**

	Original	Proposed 1	Proposed 2
$V1\_01\_easy$	0.028	0.024	0.022
$V1\_02\_medium$	0.032	0.027	0.024
$V1\_03\_difficult$	X	X	X
$V2\_01\_easy$	0.052	0.048	0.042
$V2\_02\_medium$	0.061	0.053	0.047
$V2\_03\_difficult$	0.065	0.057	0.048
MH_01_easy	0.071	0.062	0.057
MH_02_easy	0.074	0.064	0.059
MH_03_medium	0.078	0.067	0.052
$MH\_04\_difficult$	0.081	0.072	0.060
$MH\_05\_difficult$	0.086	0.072	0.054

6.8 Summary

In this chapter, we derived an invariant IMU factor with the corresponding pre-integration method. We also pointed out that the corresponding pre-integration method is related to the linearity of the propagation equation. Simulations and experiments show that the invariant IMU factor results in better estimate accuracy.

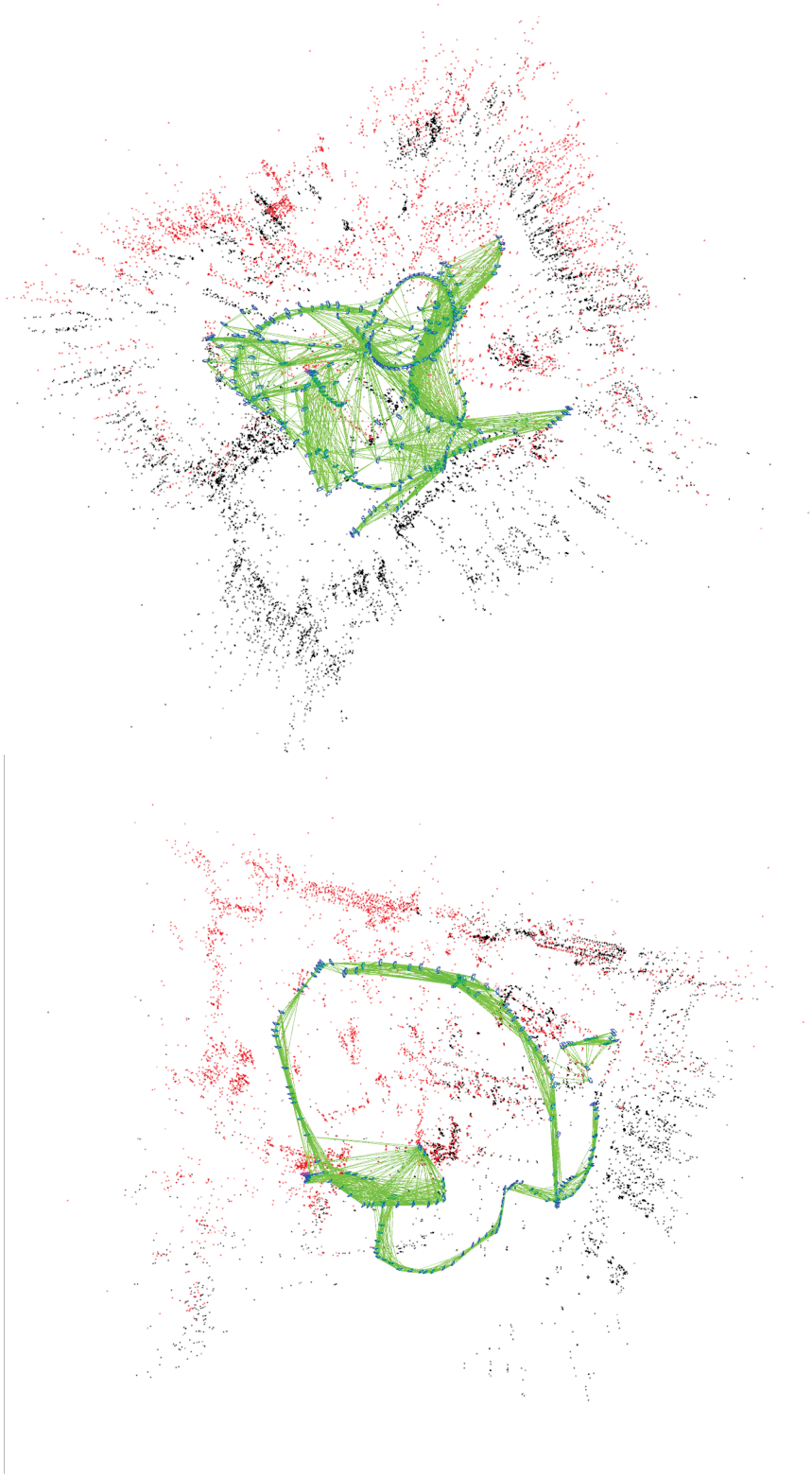

FIGURE 6.7: *V*1_02_*medium* *V*2_02_*medium*

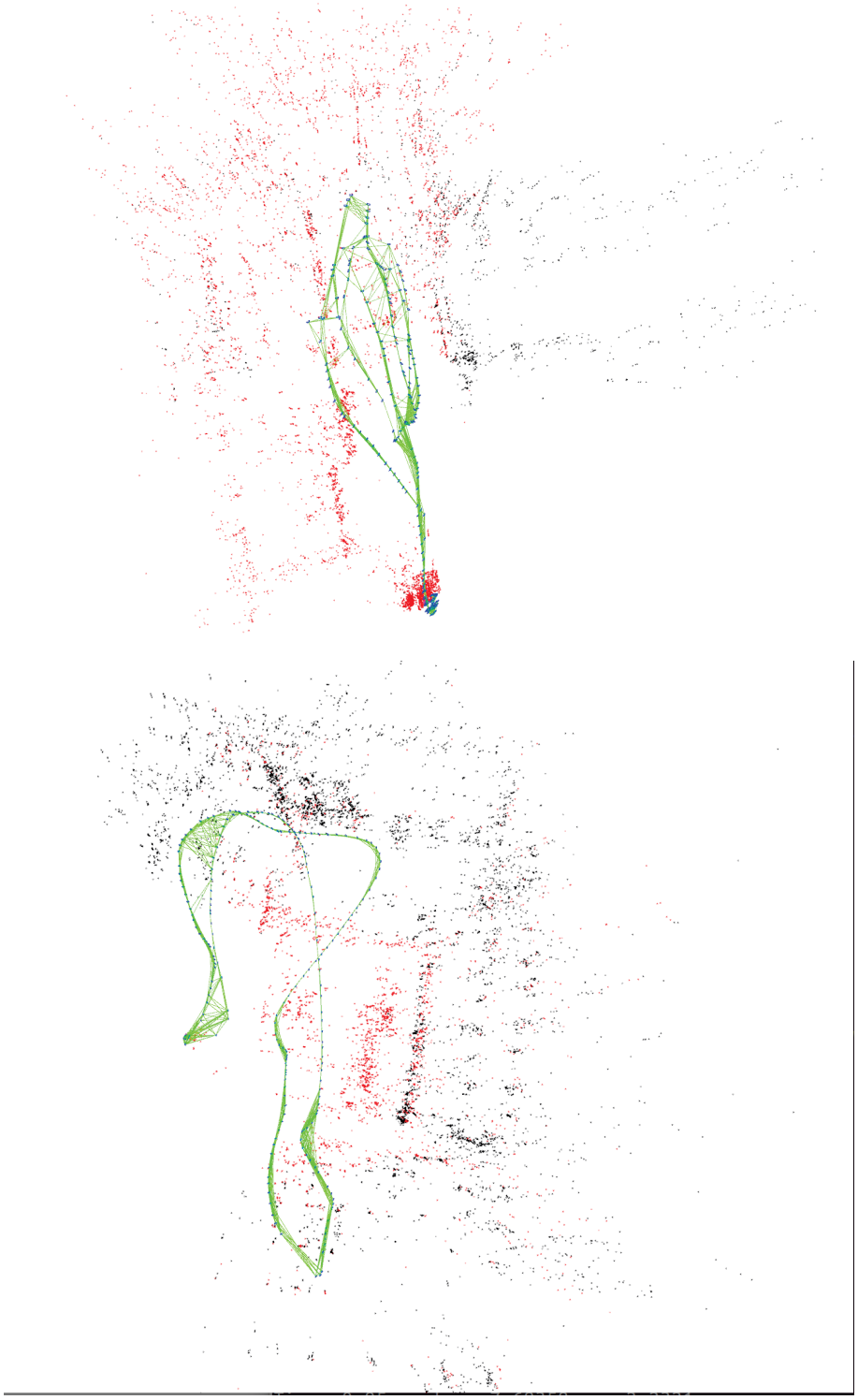

FIGURE 6.8: The constructed maps of $MH02\_easy$ (top) and $MH04\_difficult$ (bottom) using the proposed method 2

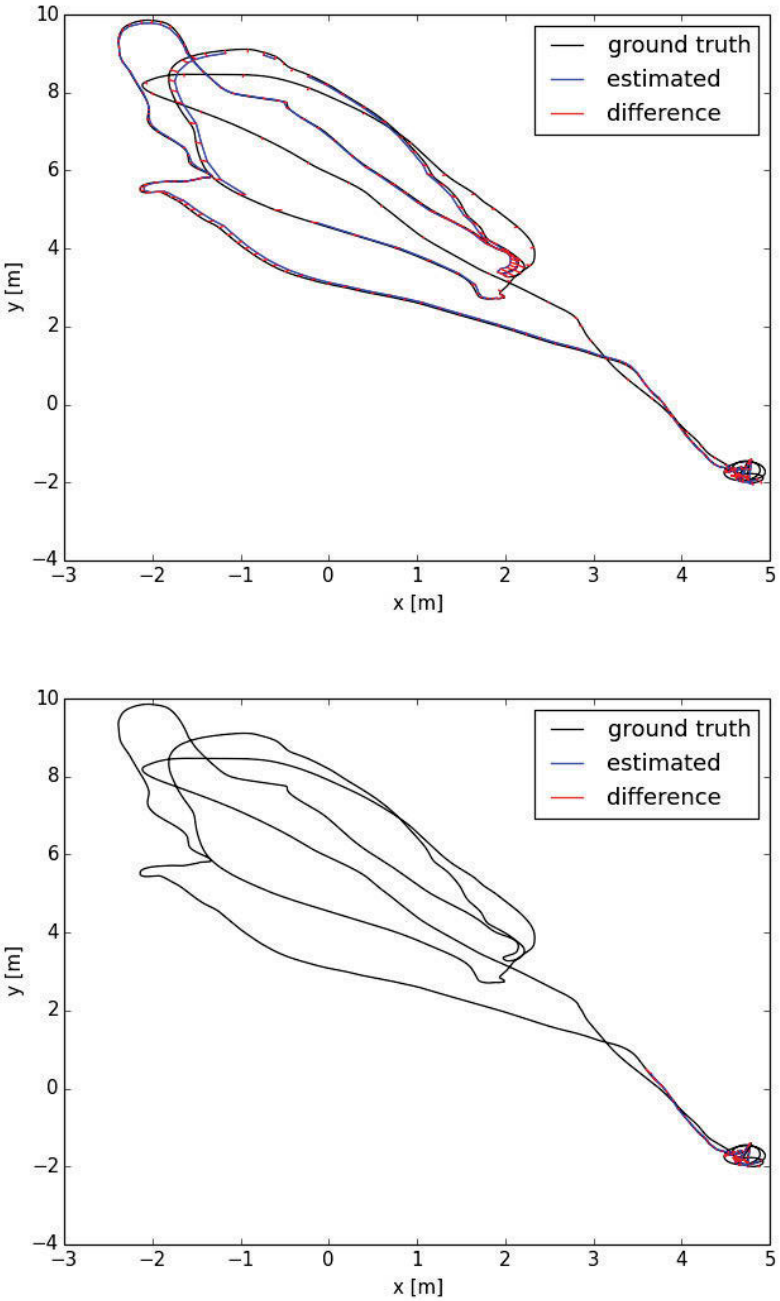

FIGURE 6.9: The estimated trajectories of $MH02\_easy$ using the original method (top) and the proposed method 2 (bottom)

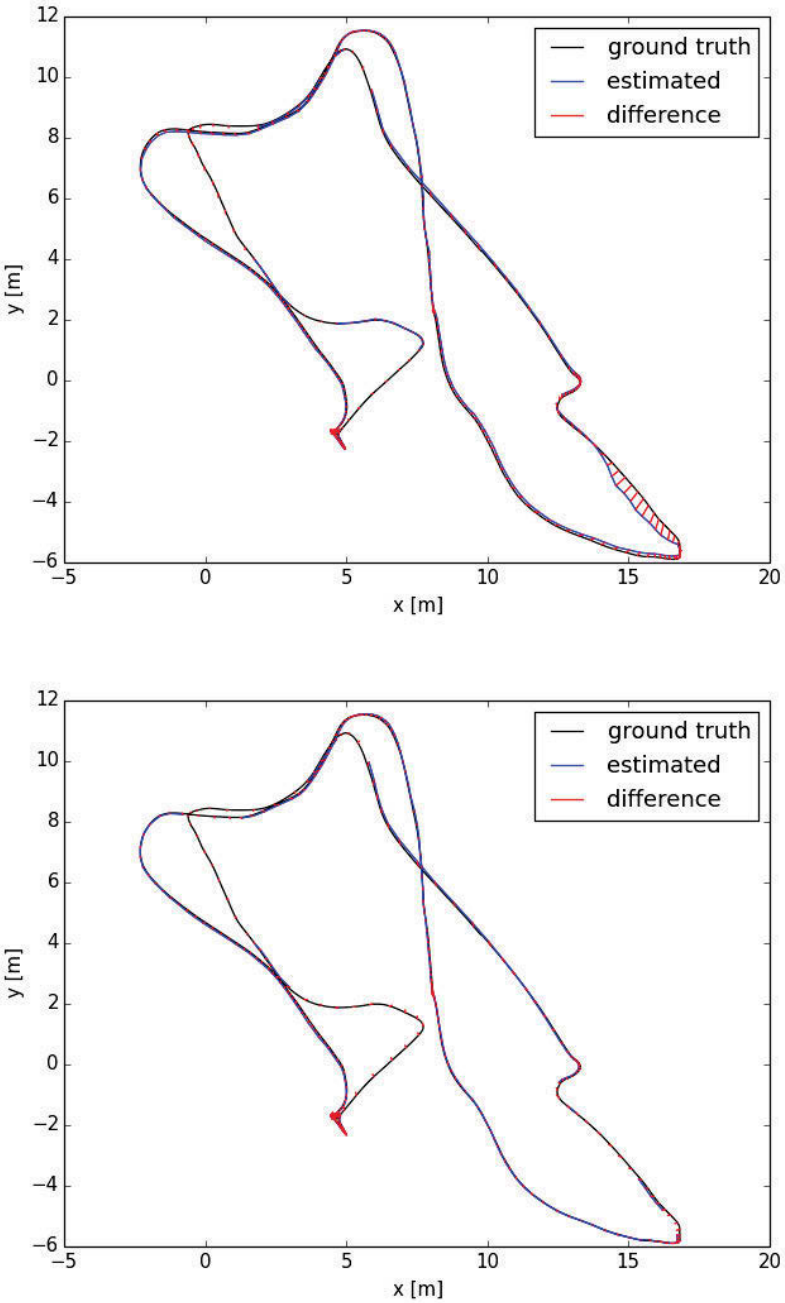

FIGURE 6.10: The estimated trajectories of $MH05\_difficult$ using the original method (top) and the proposed method 2 (bottom)

Chapter 7

Conclusion and Future Work

In this thesis, we investigated overlooked aspect of SLAM and VINS, namely the invariance property and proposed the corresponding methods to improve the existing solvers for both the EKF based approach and the optimization based approach. In this chapter, we briefly review our achievements and discuss future work.

7.1 The invariance in the EKF based approach

In Chapter 3, we investigate the convergence and consistency properties of an Invariant-Extended Kalman Filter (RI-EKF) based SLAM algorithm. Basic convergence properties of this algorithm are proven. These proofs do not require the restrictive assumption that the Jacobians of the motion and observation models need to be evaluated at the ground truth. It is also shown that the output of RI-EKF is invariant under any stochastic rigid body transformation in contrast to $\mathbb{SO}(3)$ based EKF SLAM algorithm ($\mathbb{SO}(3)$-EKF) that is only invariant under deterministic rigid body transformation. Implications of these invariance properties on the consistency of the estimator are also discussed. Monte Carlo simulation results demonstrate that RI-EKF outperforms $\mathbb{SO}(3)$-EKF, Robocentric-EKF and the "First Estimates Jacobian" EKF, for 3D point feature based SLAM.

The main contribution of Chapter 4 is an invariant EKF for VINS. It is demonstrated that the conventional EKF based VINS is not invariant under the stochastic unobservable transformation, associated with translations and a rotation about the gravitational direction. This unavoidably leads to inconsistent state estimates as the estimator does not obey a fundamental property of the physical system. To address this issue, we use a novel uncertainty representation to derive a Right Invariant error extended Kalman filter (RIEKF-VINS) that preserves this invariance property. RIEKF-VINS is then adapted to the multistate constraint Kalman filter framework to obtain a consistent state estimator. Both Monte Carlo simulations and real-world experiments are used to validate the proposed method.

Concluding Remarks

- We find out that the root reason of the inconsistency for the conventional EKF based approach is the absence of the invariance to the stochastic unobservable transformation.
- We also propose a practical and analytical method to check if the general EKF based filter satisfies the invariance property or not.
- The proposed method can be applied into SLAM and VINS, which only require some small modifications on the conventional methods and can achieve better performance in terms of both accuracy and consistency.

Future Work

We plan to further improve the RI-MSCKF algorithm on the two aspects.

- **Efficiency and robustness.** The complexity of RI-MSCKF is cubic to the number of camera poses and the efficiency is expected to be improved. The root square form is a possible option. The advantage of the implementation using the root square form has been investigated in [59].

- **Strategy.** The original MSCKF does not have the invariance property to stochastic yaw angle transformation and thus its covariance matrix cannot really reflect the actual uncertainty, especially when hovering. RI-MSCKF has potential to deal with this case without a tricky mechanism like the "two-ways marginalization" method because its invariance prevent spurious information gain.

  Remark 7.1. Another reason for MSCKF failing when hovering is the unstable triangulation, which may be dealt with using a parallax parameterization.

7.2 The invariance in the optimization based approach

In Chapter 5, we successfully translate the concept of the invariance property from EKF to optimization, which is usually ignored in the conventional viewpoint. The conventional algorithm of the optimization-based SLAM uses the naive retraction such that the update of landmarks is independent of that of the poses, which unexpectedly makes the estimation related to the selection of the global frame. To solve this problem, we propose the novel retraction without changing the cost function to obtain the invariance such that the estimate by using the novel retraction and Gauss-Newton/Leverberg-Marquart after each iteration is invariant under the rigid body transformation.

The main content of Chapter 6 is an invariant IMU factor for VINS and the corresponding pre-integration method. The existing optimization based VINS algorithms are compromised by the lack of rigorous IMU preintegration. Some methods are restricted to the derivation such that it is difficult to apply the high-order integration. Some other methods ignore the correlation between the change of the IMU biases and the IMU poses/velocity. To address these issues, we analytically derive a novel continuous pre-integration method and propose a novel IMU factor on the novel manifold.

Concluding Remarks

- We prove the invariance is also significant to the optimization based approach via numerical simulations and real experiments. Exploiting the invariance is very helpful such that the convergence can be improved.

- Establishing a link between the EKF-VINS and the optimization based VINS is another contribution. We point out that the keypoint of the so-called pre-integration method is finding an error such that the dynamic equation of this error vector is autonomous.

Future Work

We also plan to improve our current VINS in two aspects.

- **Front-end.** A reliable front-end is fundamental for the VINS system. Recently, more efficient and robust feature matching algorithms have been proposed. For example, the algorithm [60] is very promising to replace the standard feature matching algorithm while it does not require more computational time.
- **Novel Bundle Adjustment** The conventional bundle adjustment is easy to fall into the local minimum when the initial guess is poor. On the other hand, the Hessian matrix is also easy to get singular such that LM is usually used instead of Powell's Dogleg, which may significantly affect the numerical calculation. A more robust and elegant method with theoretical analysis is needed. The singularity are mainly from two aspects:
  - The zero error of re-projection only implies that the predicted feature is on the measured line instead of the measured array because the projection function $\pi(\mathbf{x}) = [\frac{x_1}{x_3}, \frac{x_2}{x_3}]'$ ($\mathbf{x} = [x_1, x_2, x_3]'$) is discontinuous when $x_3 = 0$. For this issue, we will derive a new projection function $\pi_{new}(\mathbf{x})$, globally continuous for $\mathbf{x}$. The use of new projection function $\pi_{new}(\mathbf{x})$ can reduce a number of local minimums. Even when the initial guess of a point is at the back of the camera, the correct estimate can be recovered by using $\pi_{new}(\mathbf{x})$, which is very difficult or impossible for $\pi(\mathbf{x})$.
  - The commonly used parameterization XYZ and the inverse depth have singularity. *This singularity happens when the information w.r.t. the used parameterization is not enough.* For example, we cannot accurately estimate the XYZ coordinates of the feature when the parallax angle is very small. On the other

hand, the ill-conditioned Hessian matrix also affects the estimates of the other "normal" variables. For this issue, we will extend our method on manifold for both pose parameterization and feature parameterization based on our previous work in [61].

We will prove that the Hessian matrix is non-singular when using the new projection function $\pi(\mathbf{x})$ and the new parameterization. Intuitively, the information w.r.t. the new parameterization is always enough. The non-singular Hessian matrix implies that we can safely choose the efficient Powell's Dogleg method. From our recent tests, the combination of "$\pi_{new}(\mathbf{x})$+ new parameterization+ Dogleg" is very promising: robust (converging even when initial guess is very poor) and efficient (much less iterations). We expect that the VINS system can also benefit from this novel bundle adjustment method.

Appendix A

EKF-SLAM

A.1 Proof of Theorem 3.4

In the following, we use mathematical induction to prove this theorem. Note that at the beginning, the estimate is $(\hat{\mathbf{X}}, \mathbf{P})$ where $\hat{\mathbf{X}} = \left(\hat{\mathbf{R}}, \hat{\mathbf{p}}, \hat{\mathbf{f}}^1, \cdots, \hat{\mathbf{f}}^N\right)$. After the first observation, the mean estimate of state and covariance matrix are augmented as below via the method shown in Alg. 6: $\hat{\mathbf{X}}_1 = \left(\hat{\mathbf{R}}, \hat{\mathbf{p}}, \hat{\mathbf{f}}^1, \cdots, \hat{\mathbf{f}}^N, \hat{\mathbf{f}}^{N+1}\right)$ and $\mathbf{P}_1 = \begin{bmatrix} \mathbf{P} & \mathbf{L} \\ \mathbf{L}^\intercal & \hat{\mathbf{R}}\mathbf{\Psi}\hat{\mathbf{R}}^\intercal + \mathbf{W} \end{bmatrix}$. Obviously, after one observation, the mean estimate of robot pose and the previous "landmarks" does not change and the covariance matrix follows the proposed form. We now assume that after k times observations, the estimate becomes $\hat{\mathbf{X}}_k = \left(\hat{\mathbf{R}}, \hat{\mathbf{p}}, \hat{\mathbf{f}}^1, \cdots, \hat{\mathbf{f}}^N, \hat{\mathbf{f}}_k^{N+1}\right)$ and $\mathbf{P}_k = \begin{bmatrix} \mathbf{P} & \mathbf{L} \\ \mathbf{L}^\intercal & \frac{\hat{\mathbf{R}}\mathbf{\Psi}\hat{\mathbf{R}}^\intercal}{k} + \mathbf{W} \end{bmatrix}$. Now we discuss the case after k times observations of next propagation and update. Because the robot is always perfectly stationary, after propagation at time k, the mean estimate is $\hat{\mathbf{X}}_{k+1|k} = \hat{\mathbf{X}}_k$ and covariance matrix becomes $\mathbf{P}_{k+1|k} = \mathbf{P}_k$. According to Alg. 5, we have $\mathbf{S} = \mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^\intercal + \mathbf{\Psi} = \frac{k+1}{k}\mathbf{\Psi}$ and $\mathbf{K} = \mathbf{P}_{k+1|k}\mathbf{H}^\intercal\mathbf{S}^{-1} = \begin{bmatrix} \mathbf{0}_{3,(3N+6)} & -\frac{1}{k+1}\hat{\mathbf{R}}^\intercal \end{bmatrix}^\intercal$, where $\mathbf{H} = \begin{bmatrix} \mathbf{0}_{3,3} & \hat{\mathbf{R}}^\intercal & \mathbf{0}_{3,3N} & -\hat{\mathbf{R}}^\intercal \end{bmatrix}$. Then it is easy to see that all elements from the vector $\mathbf{K}\mathbf{y}$ are zero except the last 3 elements, and hence the estimate of robot pose and the old landmarks after $k+1$ times observations are **the same** as that in the time step k. The covariance matrix at time

$k+1$ is $\mathbf{P}_{k+1} = (\mathbf{I}-\mathbf{KH})\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P} & \mathbf{L} \\ \mathbf{L}^\intercal & \frac{\hat{\mathbf{R}}\boldsymbol{\Psi}\hat{\mathbf{R}}^\intercal}{k+1} + \mathbf{W} \end{bmatrix}$. When k converges to infinity, we have (3.7).

A.2 Proof of Theorem 3.6

By using result in Theorem 1 and the Jacobian matrices in (3.5), we have

$$\mathbf{P}_B^0 = \mathbf{P}_A^\infty + \Delta\mathbf{P}, \tag{A.1}$$

where $\mathbf{P}_A^\infty$ (given in (3.8)) is the covariance matrix before moving to the point B, $\Delta\mathbf{P} = ad_{\hat{\mathbf{X}}_A}\mathbf{E}\tilde{\boldsymbol{\Phi}}\mathbf{E}^\intercal ad_{\hat{\mathbf{X}}_A}^\intercal$ can be regarded as the incremental uncertainty caused by the odometry noise, and $\tilde{\boldsymbol{\Phi}} = \mathbf{B}\boldsymbol{\Phi}\mathbf{B}^\intercal$ is a positive definite matrix.

After l observations at point B, the information matrix $\boldsymbol{\Omega}_l^B$ (the inverse of $\mathbf{P}_l^B$) becomes $\boldsymbol{\Omega}_l^B = \boldsymbol{\Omega}_0^B + \sum_{j=1}^l \mathbf{H}_j^\intercal \bar{\boldsymbol{\Psi}}^{-1}\mathbf{H}_j$, where $\mathbf{H}_j$ is obtained by stacking all matrices $\mathbf{H}_j^i = \hat{\mathbf{R}}_j^\intercal \mathbf{H}^i$ $(i = 1, \cdots, m)$, and $\hat{\mathbf{R}}_j$ is the estimated orientation after j times observations at point B. Note that $\bar{\boldsymbol{\Psi}}$ is isotropic, we have $\mathbf{H}_j^\intercal \bar{\boldsymbol{\Psi}}^{-1}\mathbf{H}_j = \mathbf{H}^\intercal \bar{\boldsymbol{\Psi}}^{-1}\mathbf{H}$ $(j = 1, \cdots, l)$. Therefore, the information matrix is $\boldsymbol{\Omega}_l^B = \boldsymbol{\Omega}_0^B + l\mathbf{H}^\intercal \bar{\boldsymbol{\Psi}}^{-1}\mathbf{H}$. Via the matrix inversion lemma in [17], the covariance matrix after l observations at point B is

$$\mathbf{P}_B^l = (\boldsymbol{\Omega}_l^B)^{-1} = \mathbf{P}_B^0 - \mathbf{P}_B^0\mathbf{H}^\intercal(\frac{\bar{\boldsymbol{\Psi}}}{l} + \mathbf{H}\mathbf{P}_B^0\mathbf{H}^\intercal)^{-1}\mathbf{H}\mathbf{P}_B^0. \tag{A.2}$$

Note that $\mathbf{H}\mathbf{P}_A^\infty = \mathbf{0}$, we substitute (A.1) into (A.2):

$$\begin{aligned} \mathbf{P}_B^l &= \mathbf{P}_A^\infty + \Delta\mathbf{P} - \Delta\mathbf{P}\mathbf{H}^\intercal(\frac{\bar{\boldsymbol{\Psi}}}{l} + \mathbf{H}\Delta\mathbf{P}\mathbf{H}^\intercal)^{-1}\mathbf{H}\Delta\mathbf{P} \\ &= \mathbf{P}_A^\infty + ad_{\hat{\mathbf{X}}_A}\mathbf{E}(\tilde{\boldsymbol{\Phi}}^{-1} + l\tilde{\mathbf{H}}^\intercal\bar{\boldsymbol{\Psi}}^{-1}\tilde{\mathbf{H}})^{-1}\mathbf{E}^\intercal ad_{\hat{\mathbf{X}}_A}^\intercal \\ &= \mathbf{P}_A^\infty + \bar{\mathbf{P}}_B^l. \end{aligned} \tag{A.3}$$

Furthermore, $\tilde{\mathbf{H}}^\intercal\bar{\boldsymbol{\Psi}}^{-1}\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{S}_2^\intercal & m\boldsymbol{\Psi}^{-1} \end{bmatrix}$ where $\mathbf{S}_1 = \sum_{i=1}^m S^\intercal(\tilde{\mathbf{f}}_i)\boldsymbol{\Psi}^{-1}S(\tilde{\mathbf{f}}_i)$, $\mathbf{S}_2 = (\sum_{i=1}^m S(\tilde{\mathbf{f}}_i))^\intercal\boldsymbol{\Psi}^{-1}$ and $\tilde{\mathbf{f}}_i = \hat{\mathbf{R}}^\intercal(\hat{\mathbf{p}} - \hat{\mathbf{f}}_i)$ $(i = 1, \cdots, m)$. Generally speaking, $\tilde{\mathbf{H}}^\intercal\boldsymbol{\Psi}^{-1}\tilde{\mathbf{H}}$ is full rank when $m > 3$

and there are three landmarks that are non-coplanar with the robot position. Under this condition, it is easy to see that $\mathbf{P}_B^l \to \mathbf{P}_A^\infty$ when $l \to \infty$.

A.3 Proof of Theorem 3.9

Here, we only prove that the invariance property of RI-EKF and $\mathbb{SO}(3)$-EKF. The invariance properties of the other algorithms can be easily proven in a similar way or through a counter example.

First, we prove that the outputs of $\mathbb{SO}(3)$-EKF and RI-EKF is invariant to deterministic rigid body transformation. Assume the estimate at time 0 is $(\hat{\mathbf{X}}_0, \mathbf{P}_0)$ in terms of the general EKF framework. After one step propagation via the odometry $\mathbf{u}_0$, the estimate becomes $(\hat{\mathbf{X}}_{1|0}, \mathbf{P}_{1|0})$. Then after obtaining observations $\mathbf{z}_1$, the estimate becomes $(\mathbf{X}_1, \mathbf{P}_1)$. On the other hand, in $\mathbb{SO}(3)$-EKF and RI-EKF, there exists a matrix $\mathbf{Q}_\mathcal{T}$ for any rigid body transformation $\mathcal{T}$ such that

$$\mathcal{T}(\mathbf{X} \oplus \mathbf{Q}_\mathcal{T}^{-1}\mathbf{e}) = \mathcal{T}(\mathbf{X}) \oplus \mathbf{e} \quad \forall\ \mathbf{X}. \tag{A.4}$$

Therefore, if a deterministic rigid body transformation $\mathcal{T}$ is applied at time 0, the estimate becomes $(\hat{\mathbf{Y}}_0, \mathbf{P}y_0)$, where $\hat{\mathbf{Y}}_0 = \mathcal{T}(\hat{\mathbf{X}}_0)$ and $\mathbf{P}y_0 = \mathbf{Q}_\mathcal{T}\mathbf{P}_0\mathbf{Q}_\mathcal{T}^\intercal$. Now we calculate the new Jacobians $\mathbf{F}y_0$ and $\mathbf{G}y_0$ in propagation

$$\begin{aligned}
\mathbf{F}y_0 &= \left.\frac{\partial f(\hat{\mathbf{Y}}_0 \oplus \mathbf{e}, \mathbf{u}_0, \mathbf{0}) \ominus f(\hat{\mathbf{Y}}_0, \mathbf{u}_0, \mathbf{0})}{\partial \mathbf{e}}\right|_{\mathbf{0}} \\
&= \left.\frac{\partial f(\mathcal{T}(\hat{\mathbf{X}}_0) \oplus \mathbf{e}, \mathbf{u}_0, \mathbf{0}) \ominus f(\mathcal{T}(\hat{\mathbf{X}}_0), \mathbf{u}_0, \mathbf{0})}{\partial \mathbf{e}}\right|_{\mathbf{0}} \\
&\overset{(A.4)}{=} \left.\frac{\partial f(\mathcal{T}(\hat{\mathbf{X}}_0 \oplus \mathbf{Q}_\mathcal{T}^{-1}\mathbf{e}), \mathbf{u}_0, \mathbf{0}) \ominus f(\mathcal{T}(\hat{\mathbf{X}}_0), \mathbf{u}_0, \mathbf{0})}{\partial \mathbf{e}}\right|_{\mathbf{0}} \\
&= \left.\frac{\partial \mathcal{T}(f(\hat{\mathbf{X}}_0 \oplus \mathbf{Q}_\mathcal{T}^{-1}\mathbf{e}, \mathbf{u}_0, \mathbf{0})) \ominus \mathcal{T}(f(\hat{\mathbf{X}}_0, \mathbf{u}_0, \mathbf{0}))}{\partial \mathbf{e}}\right|_{\mathbf{0}} \\
&= \left.\frac{\partial \mathcal{T}(f(\hat{\mathbf{X}}_0, \mathbf{u}_0, \mathbf{0}) \oplus \mathbf{F}_0\mathbf{Q}_\mathcal{T}^{-1}\mathbf{e}) \ominus \mathcal{T}(f(\hat{\mathbf{X}}_0, \mathbf{u}_0, \mathbf{0}))}{\partial \mathbf{e}}\right|_{\mathbf{0}} \\
&\overset{(A.4)}{=} \mathbf{Q}_\mathcal{T}\mathbf{F}_0\mathbf{Q}_\mathcal{T}^{-1}.
\end{aligned} \tag{A.5}$$

Similarly, we have $\mathbf{G}y_0 = \mathbf{Q}_\mathcal{T}\mathbf{G}_0$. Hence, after one step propagation the estimate becomes $(\hat{\mathbf{Y}}_{1|0}, \mathbf{P}y_{1|0})$, where $\hat{\mathbf{Y}}_{1|0} = f(\hat{\mathbf{Y}}_0, \mathbf{u}_0, \mathbf{0}) = \mathcal{T}(\hat{\mathbf{X}}_{1|0})$ and $\mathbf{P}y_{1|0} = \mathbf{F}y_0\mathbf{P}y_0\mathbf{F}y_0^\intercal + \mathbf{G}y_0\mathbf{\Phi}_0\mathbf{G}y_0^\intercal = \mathbf{Q}_\mathcal{T}\mathbf{P}_{1|0}\mathbf{Q}_\mathcal{T}^\intercal$. The new Jacobians in update becomes $\mathbf{H}y_1 = \mathbf{H}_1\mathbf{Q}_\mathcal{T}^{-1}$. Then it is easy to obtain $\mathbf{K}_y = \mathbf{Q}_\mathcal{T}\mathbf{K}$, resulting in $\hat{\mathbf{Y}}_1 = \hat{\mathbf{Y}}_{1|0} \oplus \mathbf{K}_y\mathbf{y} = \mathcal{T}(\hat{\mathbf{X}}_{1|0}) \oplus \mathbf{Q}_\mathcal{T}\mathbf{K}\mathbf{y} = \mathcal{T}(\hat{\mathbf{X}}_{1|0} \oplus \mathbf{K}\mathbf{y}) = \mathcal{T}(\hat{\mathbf{X}}_1)$. The covariance matrix after update becomes $\mathbf{P}y_1 = (\mathbf{I} - \mathbf{K}_y\mathbf{H}y_1)\mathbf{P}y_{1|0} = \mathbf{Q}_\mathcal{T}\mathbf{P}_1\mathbf{Q}_\mathcal{T}^\intercal$. In all, $\hat{\mathbf{Y}}_1 = \mathcal{T}(\hat{\mathbf{X}}_1)$ and $\mathbf{P}y_1 = \mathbf{Q}_\mathcal{T}\mathbf{P}_1\mathbf{Q}_\mathcal{T}^\intercal$. By mathematical induction, we can see the outputs of $\mathbb{SO}(3)$-EKF (and RI-EKF) are invariant under deterministic rigid body transformation.

Secondly, we prove the invariance property of RI-EKF under stochastic identity body transformation $\mathcal{T}_\mathbf{g}$ ($\mathbf{g} = (\mathbf{I}_3, \mathbf{0}, \Theta)$) for all $\bar{\mathbf{\Sigma}}$ where $\bar{\mathbf{\Sigma}}$ is the covariance matrix of noise Θ. Consider the estimate at time 0 is $(\hat{\mathbf{X}}_0, \mathbf{P}_0)$ in RI-EKF. If the stochastic rigid body transformation $\mathcal{T}_\mathbf{g}$ is applied, the estimate becomes $(\hat{\mathbf{X}}_0, \mathbf{P}_0 + \Delta\mathbf{P})$ where $\Delta\mathbf{P} = \mathbf{C}\bar{\Sigma}\mathbf{C}^\intercal$ and

$$\mathbf{C} = \frac{\partial \mathcal{T}_\mathbf{g}(\hat{\mathbf{X}}_0) \ominus \hat{\mathbf{X}}_0}{\partial \Theta}|_\mathbf{0} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{I}_3 \\ \vdots & \vdots \\ \mathbf{0}_{3,3} & \mathbf{I}_3 \end{bmatrix}. \tag{A.6}$$

After propagation, the estimate becomes $(\hat{\mathbf{X}}_{1|0}, \mathbf{P}_{1|0} + \Delta\mathbf{P})$ due to $\mathbf{F}_n = \mathbf{I}$ given in (3.5). Note that $\mathbf{H}_1\Delta\mathbf{P} = \mathbf{0}$, it is easy to get the posterior estimate $(\hat{\mathbf{X}}_1, \mathbf{P}_1 + \Delta\mathbf{P})$. By mathematical induction, we can conclude that the output of RI-EKF is invariant under stochastic identity transformation.

Appendix B

EKF-VINS

The notation $\oplus_{imu}$ is defined as

$$\begin{aligned}\bar{\mathbf{X}} \oplus_{imu} \mathbf{e}_I = (&\exp(\mathbf{e}_\theta)\mathbf{R}, \exp(\mathbf{e}_\theta)\mathbf{v} + J_r(-\mathbf{e}_\theta)\mathbf{e}_v, \\ &\exp(\mathbf{e}_\theta)\mathbf{p} + J_r(-\mathbf{e}_\theta)\mathbf{e}_p, \mathbf{b}_g + \mathbf{e}_{bg}, \mathbf{b}_a + \mathbf{e}_{ba})\end{aligned} \tag{B.1}$$

where $\bar{\mathbf{X}} = (\mathbf{R}, \mathbf{v}, \mathbf{p}, \mathbf{b}_g, \mathbf{b}_a)$ and $\mathbf{e}_I = \left[\mathbf{e}_\theta, \mathbf{e}_v, \mathbf{e}_p, \mathbf{e}_{bg}, \mathbf{e}_{ba}\right] \in \mathbb{R}^{15}$

The notation $\oplus_{pose}$ is defined as

$$\mathbf{C} \oplus_{pose} \mathbf{e}_c^i = (\exp(\mathbf{e}_\theta^i)\mathbf{R}, \exp(\mathbf{e}_\theta^i)\mathbf{p} + J_r(-\mathbf{e}_\theta^i)\mathbf{e}_p^i) \tag{B.2}$$

where $\mathbf{C} = (\mathbf{R}, \mathbf{p}) \in \mathbb{SE}(3)$ and $\mathbf{e}_c^i = \left[\mathbf{e}_\theta^i, \mathbf{e}_p^i\right] \in \mathbb{R}^6$.

B.1 Proof of Theorem 4.7

Here we first prove the sufficiency. It is assumed that this filter satisfies: for each deterministic unobservable transformation $\mathcal{T}_\mathbf{D}$ there exists $\mathbf{W}_\mathbf{D}$ such that $\mathcal{T}_\mathbf{D}(\mathbf{X} \oplus \mathbf{e}) = \mathcal{T}_\mathbf{D}(\mathbf{X}) \oplus \mathbf{W}_\mathbf{D}\mathbf{e}$. For any estimate $(\hat{\mathbf{X}}_i, \mathbf{P}_i)$ at time-step i, we have another estimate $(\hat{\mathbf{Y}}_i, \mathbf{P}y_i) = (\mathcal{T}_\mathbf{D}(\hat{\mathbf{X}}_i), \mathbf{W}_\mathbf{D}\mathbf{P}_i\mathbf{W}_\mathbf{D}^\intercal)$ after applying the deterministic transformation $\mathcal{T}_\mathbf{D}$. After one step propagation, we have $(\hat{\mathbf{X}}_{i+1|i}, \mathbf{P}_{i+1|i})$ and $(\hat{\mathbf{Y}}_{i+1|i}, \mathbf{P}y_{i+1|i})$ where $\hat{\mathbf{Y}}_{i+1|i} =$

$\mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_{i+1|i})$ and $\mathbf{P}\mathbf{y}_{i+1|i} = \mathbf{W}_{\mathbf{D}}\mathbf{P}_{i+1|i}\mathbf{W}_{\mathbf{D}}^{\intercal}$. Note that $\mathbf{H}y_{i+1} = \mathbf{H}_{i+1}\mathbf{W}_{\mathbf{D}}^{-1}$ and then it is easy to obtain $\mathbf{K}y = \mathbf{W}_{\mathbf{D}}\mathbf{K}$, resulting in the mean estimate $\hat{\mathbf{Y}}_{i+1}$ as below

$$\begin{aligned}\hat{\mathbf{Y}}_{i+1} &= \hat{\mathbf{Y}}_{i+1|i} \oplus \mathbf{K}_y\tilde{\mathbf{z}} \\ &= \mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_{i+1|i}) \oplus \mathbf{W}_{\mathbf{D}}\mathbf{K}\tilde{\mathbf{z}} \\ &= \mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_{i+1|i} \oplus \mathbf{K}\tilde{\mathbf{z}}) \\ &= \mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_{i+1})\end{aligned} \tag{B.3}$$

The covariance matrix after update becomes $\mathbf{P}y_{i+1} = (\mathbf{I}-\mathbf{K}_y\mathbf{H}y_{i+1})\mathbf{P}y_{i+1|i} = \mathbf{W}_{\mathbf{D}}\mathbf{P}_{i+1}\mathbf{W}_{\mathbf{D}}^{\intercal}$. In all, $\hat{\mathbf{Y}}_{i+1} = \mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_{i+1})$ and $\mathbf{P}y_{i+1} = \mathbf{W}_{\mathbf{D}}\mathbf{P}_{i+1}\mathbf{W}_{\mathbf{D}}^{\intercal}$. By mathematical induction, we can see $\hat{\mathbf{Y}}_n = \mathcal{T}_{\mathbf{D}}(\hat{\mathbf{X}}_n)$ for $n \geq i$ and hence the output of this filter is invariant under any deterministic transformation $\mathcal{T}_{\mathbf{D}}$. The proof of the necessity is similar.

B.2 Proof of Theorem 4.8

Here we first prove the sufficiency. It is assumed that this filter satisfies:

$$\mathbf{H}_{n+i+1}\mathbf{\Phi}_{n+i}\mathbf{\Phi}_{n+i-1}\cdots\mathbf{\Phi}_i\mathbf{N}_i = \mathbf{0} \ \forall \ n \text{ and } i \geq 0 \tag{B.4}$$

For any estimate $(\hat{\mathbf{X}}_i, \mathbf{P}_i)$ at time-step i, we have another estimate $(\hat{\mathbf{Y}}_i, \mathbf{P}y_i) = (\hat{\mathbf{X}}_i, \mathbf{P}_i + \mathbf{N}_i\mathbf{\Sigma}\mathbf{N}_i^{\intercal})$ after applying the stochastic identify transformation $\mathcal{T}_{\mathbf{S}}$ where $\mathbf{S} = (\mathbf{0}, \boldsymbol{\epsilon})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. After one step propagation, we have $(\hat{\mathbf{X}}_{i+1|i}, \mathbf{P}_{i+1|i})$ and $(\hat{\mathbf{Y}}_{i+1|i}, \mathbf{P}y_{i+1|i}) = (\hat{\mathbf{X}}_{i+1|i}, \mathbf{P}_{i+1|i} + \mathbf{\Phi}_i\mathbf{N}_i\mathbf{\Sigma}\mathbf{N}_i^{\intercal}\mathbf{\Phi}_i^{\intercal})$. Note that $\mathbf{H}_{i+1}\mathbf{\Phi}_i\mathbf{N}_i = \mathbf{0}$, we can easily get $(\hat{\mathbf{Y}}_{i+1}, \mathbf{P}y_{i+1}) = (\hat{\mathbf{X}}_{i+1}, \mathbf{P}_{i+1} + \mathbf{\Phi}_i\mathbf{N}_i\mathbf{\Sigma}\mathbf{N}_i^{\intercal}\mathbf{\Phi}_i^{\intercal})$. By mathematical induction, we have $(\hat{\mathbf{Y}}_n, \mathbf{P}y_n) = (\hat{\mathbf{X}}_n, \mathbf{P}_n + \mathbf{\Phi}_n\cdots\mathbf{\Phi}_i\mathbf{N}_i\mathbf{\Sigma}\mathbf{N}_i^{\intercal}\mathbf{\Phi}_i^{\intercal}\cdots\mathbf{\Phi}_n^{\intercal})$. Therefore, the output of this filter is invariant under any stochastic identify transformation. The proof of the necessity is similar and omitted here.

Bibliography

[1] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. Analysis and improvement of the consistency of extended kalman filter based slam. In *Robotics and Automation, 2008 IEEE International Conference on (ICRA)*, pages 473–479, May 2008.

[2] José A Castellanos, José Neira, and Juan D Tardós. Limits to the consistency of ekf-based slam. In *5th IFAC Symp, Intell. Autonom. Veh. IAV'04*, 2004.

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

[4] T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28 (1):61–76, 2012.

[5] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[6] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 2016.

[7] Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang. High-accuracy preintegration for visual-inertial navigation. In *Workshop on Algorithmic Foundations of Robotics*, 2016.

[8] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613, May 2011.

[9] T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

[10] Veeravalli Seshadri Varadarajan. *Lie groups, Lie algebras, and their representations*, volume 102. Springer Science & Business Media, 2013.

[11] Christopher James McFarland. *Adaptive Identification and Control for Underwater Vehicles: Theory and Comparative Experimental Evaluations.* PhD thesis, 2013.

[12] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC press, 1994.

[13] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987.

[14] D. M. Rosen, M. Kaess, and J. J. Leonard. Rise: An incremental trust-region method for robust online sparse least-squares estimation. *IEEE Transactions on Robotics*, 30 (5):1091–1108, 2014.

[15] Åke Björck. *Numerical methods for least squares problems.* SIAM, 1996.

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

[17] S. Huang and G. Dissanayake. Convergence and consistency analysis for extended kalman filter based slam. *IEEE Transactions on Robotics*, 23(5):1036–1049, 2007.

[18] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the ekf-slam algorithm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3562–3568, Oct 2006.

[19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3281–3288, May 2011.

[20] Pedro Lourenço, Bruno J Guerreiro, Pedro Batista, Paulo Oliveira, and Carlos Silvestre. Simultaneous localization and mapping for aerial vehicles: a 3-d sensor-based gas filter. *Autonomous Robots*, 40(5):881–902, 2016.

[21] J. Andrade-Cetto and A. Sanfeliu. The effects of partial observability in slam. In *Robotics and Automation, 2004 IEEE International Conference on. (ICRA)*, volume 1, pages 397–402 Vol.1, April 2004. doi: 10.1109/ROBOT.2004.1307182.

[22] K. W. Lee, W. S. Wijesoma, and J. I. Guzman. On the observability and observability analysis of slam. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3569–3574, 2006.

[23] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Observability-based rules for designing consistent ekf slam estimators. 29(5):502–528, 2010.

[24] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, 30(1):158–176, 2014.

[25] 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 Transactions on Robotics and Automation*, 17(3):229–241, Jun 2001. ISSN 1042-296X.

[26] Anastasios I Mourikis and Stergios I Roumeliotis. Analytical characterization of the accuracy of slam without absolute orientation measurements. In *Robotics: Science and systems*, pages 215–222, 2006.

[27] R. Mahony, T. Hamel, and J. M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, 2008.

[28] Luca Carlone, Vito Macchia, Federico Tibaldi, and Basilio Bona. Quaternion-based ekf-slam from relative pose measurements: observability analysis and applications. *Robotica*, 33(06):1250–1280, 2015.

[29] G. Dubbelman and B. Browning. Cop-slam: Closed-form online pose-chain optimization for visual slam. *IEEE Transactions on Robotics*, 31(5):1194–1213, Oct 2015. ISSN 1552-3098.

[30] Christoph Hertzberg, Ren Wagner, Udo Frese, and Lutz Schrder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013. ISSN 1566-2535.

[31] N. Aghannan and P. Rouchon. On invariant asymptotic observers. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 2, pages 1479–1484, 2002.

[32] Silvère Bonnabel. Symmetries in observer design: Review of some recent results and applications to ekf-based slam. In *Robot Motion and Control 2011*, pages 3–15. Springer London, 2012. ISBN 978-1-4471-2343-9.

[33] Axel Barrau and Silvere Bonnabel. An EKF-SLAM algorithm with consistency properties. *CoRR*, abs/1510.06263, 2015.

[34] P. A. Absil, Christopher G Baker, and Kyle A Gallivan. Trust-region methods on riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

[35] Axel Barrau and Silvere Bonnabel. An ekf-slam algorithm with consistency properties. *arXiv preprint arXiv:1510.06263v3*, 2016.

[36] R. Hermann and A. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, Oct 1977. ISSN 0018-9286.

[37] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Camera-imu-based localization: Observability and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014.

[38] Jonghyuk Kim and Salah Sukkarieh. Real-time implementation of airborne inertial-slam. *Robotics and Autonomous Systems*, 55(1):62 – 71, 2007. ISSN 0921-8890.

[39] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 298–304, 2015.

[40] Eagle S. Jones and Stefano Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011.

[41] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Observability-constrained vision-aided inertial navigation. *University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep*, 1, 2012.

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

[43] G. Huang, M. Kaess, and J. J. Leonard. Towards consistent visual-inertial navigation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4926–4933, 2014.

[44] C. N. Taylor. An analysis of observability-constrained kalman filtering for vision-aided navigation. In *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pages 1240–1246, 2012.

[45] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake. Convergence and consistency analysis for a 3-d invariant-ekf slam. *IEEE Robotics and Automation Letters*, 2(2):733–740, 2017.

[46] A. Barrau and S. Bonnabel. The invariant extended kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, PP(99):1–1, 2016.

[47] M. Barczyk and A. F. Lynch. Invariant observer design for a helicopter uav aided inertial navigation system. *IEEE Transactions on Control Systems Technology*, 21 (3):791–806, 2013.

[48] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014.

[49] Agostino Martinelli et al. Observability properties and deterministic algorithms in visual-inertial structure from motion. *Foundations and Trends® in Robotics*, 3(3): 139–209, 2013.

[50] Agostino Martinelli. Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1): 44–60, 2012.

[51] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[52] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31 (5):1147–1163, 2015.

[53] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. ISSN 1573-1405.

[54] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I*, pages 404–417. Springer Berlin Heidelberg, 2006.

[55] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. *BRIEF: Binary Robust Independent Elementary Features*, pages 778–792. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-15561-1.

[56] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[57] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[58] R. Mur-Artal and J. D. Tards. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.

[59] Vaclav Smidl and Zdeněk Peroutka. Advantages of square-root extended kalman filter for sensorless control of ac drives. *IEEE Transactions on Industrial Electronics*, 59 (11):4189–4196, 2012.

[60] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan Dat Nguyen, and Ming-Ming Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[61] Liang Zhao, Shoudong Huang, Yanbiao Sun, Lei Yan, and Gamini Dissanayake. Parallaxba: bundle adjustment using parallax angle feature parametrization. *The International Journal of Robotics Research*, 34(4-5):493–516, 2015.