

Title Here

by

Author Name

Department of Mechanical Engineering
McGill University, Montreal
December, 2019

A thesis submitted to McGill University
in partial fulfillment of the requirements of the degree of
Master of Engineering

Supervisor:

Professor James Richard Forbes

Author Name
first.last@mail.mcgill.ca

© Author Name 2019

All Rights Reserved

To my family for their steadfast support and love.

Abstract

This thesis investigates the invariant extended Kalman filter (IEKF), a recently introduced method for nonlinear state estimation on matrix Lie groups. The IEKF is well suited to a particular class of systems, namely those with group-affine process models and invariant measurement models. In fact, when these conditions are met, the IEKF is a locally asymptotically convergent observer. However, in practice, process models are often not group-affine, and measurement models are often not invariant. The effect of removing these assumptions is investigated in this thesis. In particular, a 3D example is considered, with and without bias estimation. Estimating bias renders the process model not group affine. Then, a non-invariant measurement model is considered. Two different techniques are proposed to incorporate this measurement model into an IEKF, a standard approach using the non-invariant model and a novel approach in which the measurement is preprocessed to force the preprocessed measurement to be invariant. These practical extensions of the IEKF are tested in simulation to determine the effectiveness of the IEKF for more general state estimation problems. Lastly, batch estimation in the invariant framework is formulated. The problem of interest is the simultaneous localization and mapping (SLAM) problem. A general derivation of the SLAM problem on matrix Lie groups is presented. Invariant estimation theory is then leveraged. An inertial navigation example with bias estimation is then presented, with testing done in simulation.

Résumé

Cette thèse étudie le filter de Kalman invariant (IEKF), une méthode récemment introduite pour l'estimation d'état non linéaire sur des groupes de Lie matriciels. L'IEKF est bien adapté à une classe particulière de systèmes, plus précisément ceux dotés de fonctions affines et de fonctions d'observation invariantes. En fait, lorsque ces conditions sont satisfaites, l'IEKF est un observateur localement asymptotiquement convergent. Cependant, en situation pratique, ces conditions ne sont souvent pas satisfaites. Des scénarios où ces conditions ne sont pas satisfaites sont étudiés ici. En particulier, un exemple 3D est considéré, avec et sans estimation de biais dans le gyroscope. L'estimation du biais rend la fonction non-affinée. Ensuite, une fonction d'observation non-invariante est considérée. Deux techniques différentes sont proposées pour incorporer cette fonction d'observation dans un IEKF, une approche standard utilisant la fonction non-invariante et une nouvelle approche dans laquelle la mesure est prétraitée pour la forcer d'être invariante. Ces extensions pratiques du IEKF sont testées en simulation pour déterminer son efficacité pour des problèmes d'estimation d'état plus généraux. Enfin, une technique d'estimation par lot dans le cadre invariant est formulée. Un intérêt particulier est porté au problème de la localisation et de la cartographie simultanées (SLAM). Une dérivation générale du problème SLAM sur les groupes de Lie matriciels est présentée. La théorie de l'estimation invariante est ensuite mise à profit. Un exemple de navigation inertielle avec estimation du biais est ensuite présenté, avec des tests effectués en simulation.

Acknowledgements

I would like to thank Professor James Richard Forbes for his support over the last two years. His passion and extensive knowledge for his field of research makes him an ideal supervisor, and I was extremely fortunate to work with him during my Masters degree. I would also like to take this opportunity to thank the other members of the DECAR group, current and former. In particular, I would like to thank Alex Walsh for always being available to discuss ideas and answer questions.

I would also like to thank my friends and family for their support, and for always pushing me to new heights.

Preface

The contributions of this thesis that are original to the author's knowledge are as follows.

- Chapter 5
 - Solving the batch SLAM problem using a right-invariant framework while explicitly considering bias states.

All text, plots, figures and results in this thesis are produced by Jonathan Arsenault. The IEKF was originally introduced by Silvère Bonnabel and Axel Barrau in continuous time in [1], and in discrete time in [2].

Table of Contents

Dedication	ii
Abstract	iii
Acknowledgements	v
Preface	vi
List of Figures	x
List of Tables	xi
List of Abbreviations	xii
List of Symbols	xiii
Chapter	
1. Introduction	1
1.1 Thesis Objective	2
1.2 Thesis Overview	3
2. Preliminaries	4
2.1 Matrix Lie Groups	4
2.1.1 Overview	4
2.1.2 Uncertainty Representations	5
2.1.3 The Baker-Campbell-Hausdorff Formula	6
2.1.4 Linearization	6
2.2 The Special Orthogonal Group $SO(3)$	7
2.3 The Special Euclidean Group $SE(3)$	8
2.4 The Group of Double Direct Isometries $SE_2(3)$	9
2.5 Geometry	10
2.6 Kinematics	10
2.7 Optimization	11

2.7.1	Linear Least Squares	11
2.7.2	Nonlinear Least Squares	12
2.8	State Estimation	14
2.8.1	Extended Kalman Filtering	14
2.8.2	Batch Estimation [3, pp.127-143]	16
2.9	Summary	19
3.	Invariant Extended Kalman Filtering	20
3.1	Invariant Filtering in Continuous Time	21
3.2	Invariant Filtering in Discrete Time	24
3.3	Invariant Extended Kalman Filtering	27
3.4	Left-Invariant Extended Kalman Filter	28
3.4.1	Continuous-Time Prediction	29
3.4.2	Discrete-Time Prediction	30
3.4.3	Discrete-Time Correction	30
3.5	Right-Invariant Extended Kalman Filter	32
3.5.1	Continuous-Time Prediction	32
3.5.2	Discrete-Time Prediction	33
3.5.3	Discrete-Time Correction	35
3.6	Summary	36
4.	Practical Implications of the IEKF	38
4.1	Sample Problem: $SE(3)$	39
4.1.1	System Modelling	39
4.1.2	Left-Invariant Extended Kalman Filter Derivation	40
4.1.3	Right-Invariant Extended Kalman Filter Derivation	42
4.1.4	MEKF Solutions	45
4.1.5	Simulation Results	46
4.2	Applying the IEKF to a Realistic Data Set	55
4.2.1	Right-Invariant Kalman Filter Derivation	56
4.2.2	MEKF Solution	58
4.2.3	Filtering Using Pseudo-measurements	58
4.2.4	Results	60
4.3	Estimating Bias In the Invariant Framework	63
4.3.1	MEKF Solution	68
4.3.2	Simulation Results	68
4.4	Conclusion	71
5.	Invariant Batch SLAM	72
5.1	Simultaneous Localization and Mapping on Matrix Lie Groups	72
5.2	Leveraging the Invariant Framework in Batch SLAM	77
5.2.1	Left-Invariant Error	77

5.2.2	Right-Invariant Error	77
5.3	Sample Problem 3: Inertial Navigation with Bias	78
5.3.1	Matrix Lie Group	79
5.3.2	Approach 1: Left-Invariant Error Definition with Type 1 Jacobians	83
5.3.3	Approach 2: Left-Invariant Error Defintion with Type 2 Ja- cobians	92
5.3.4	Approach 3: Right-Invariant Error Definition	93
5.3.5	Approaches 4 and 5: Left-Invariant Error with Invariant Measurement Error	102
5.3.6	Approach 6: Right-Invariant Error with Invariant Measure- ment Error	104
5.4	Simulation Results	108
6.	Closing Remarks and Future Work	113
6.1	Conclusions	113
6.2	Future Work	114

List of Figures

Figure

4.1	Problem setup.	39
4.2	Problem setup, where the blue crosses are the observed landmarks.	43
4.3	Trajectory used in simulations.	47
4.4	Results comparing the MEKF-L and LIEKF varying rate gyro noise.	50
4.5	Results comparing the MEKF-R and RIEKF varying rate gyro noise.	51
4.6	Results comparing the MEKF-L and LIEKF varying velocity sensor noise. .	51
4.7	Results comparing the MEKF-R and RIEKF varying velocity sensor noise. .	52
4.8	Results comparing the MEKF-L and LIEKF varying correcting sensor noise. .	52
4.9	Results comparing the MEKF-R and RIEKF varying correcting sensor noise. .	53
4.10	Results comparing the MEKF-L and LIEKF varying all sensor noise.	53
4.11	Results comparing the MEKF-R and RIEKF varying all sensor noise.	54
4.12	Results of different noise profiles where $\delta\phi_3$ was varied.	55
4.13	Results comparing the RIEKF-C and RIEKF-RL at different initial attitude errors.	61
4.14	Results comparing the MEKF-C, RIEKF-C, MEKF-RL, and RIEKF-RL on simulated data.	62
4.15	Results comparing the MEKF-C, RIEKF-C, MEKF-RL, and RIEKF-RL on real data.	62
4.16	Results comparing the MEKF-R and RIEKF varying rate gyro noise.	69
4.17	Results comparing the MEKF-R and RIEKF varying velocity sensor noise. .	69
4.18	Results comparing the MEKF-R and RIEKF varying landmark sensor noise. .	70
4.19	Results comparing the MEKF-R and RIEKF varying all sensor noise.	70
5.1	Trajectory used in simulations.	109
5.2	Results comparing batch SLAM approaches varying rate gyro noise.	110
5.3	Results comparing batch SLAM approaches varying zcelerometer noise. . .	111
5.4	Results comparing batch SLAM approaches vvarying rate gyro bias noise. .	111
5.5	Results comparing batch SLAM approaches vvarying accelerometer bias noise.	112
5.6	Results comparing batch SLAM approaches vvarying landmark sensor noise.	112

List of Tables

Table

3.1	Summary of the key KF/EKF, MEKF, LIEKF, RIEKF equations.	37
4.1	Results comparing different Monte Carlo initialization methods.	49
4.2	Standard deviation of the noise injected into each measurement for each trial.	49
5.1	Summary of input errors and Jacobians for the 6 different approaches. . . .	82
5.2	Summary of input errors and Jacobians for the 6 different approaches. . . .	82
5.3	Standard deviation of the noise injected into each measurement for each trial.	108

List of Abbreviations

List of Symbols

$\ \cdot\ $	the Euclidian norm of a physical vector
\mathbb{R}^n	the vector space of real n -dimensional vectors
$\mathbb{R}^{m \times n}$	the space of real $m \times n$ -dimensional matrices
$(\cdot)^T$	transpose
$(\cdot)^\times$	cross operator for $\mathfrak{so}(3)$
$(\cdot)^\wedge$	operator mapping an element of \mathbb{R}^d to \mathfrak{g}
$(\cdot)^\vee$	operator mapping an element of \mathfrak{g} to \mathbb{R}^d
$\mathbf{0}$	zero matrix
$\mathbf{1}$	identity matrix
$\text{diag}(\mathbf{M}_1, \dots, \mathbf{M}_n)$	block diagonal matrix with $\mathbf{M}_1, \dots, \mathbf{M}_n$ on diagonals, and zeros elsewhere
$\underline{\mathcal{F}}_i$	reference frame
\underline{r}	a physical vector
\underline{r}^{zw}	the position of point z relative to point w
$\dot{\underline{r}}^a$	the time derivative of \underline{r} with respect to $\underline{\mathcal{F}}_a$
$\dot{\underline{r}}^{zw}{}^a = \underline{v}^{zw/a}$	velocity of point z relative to point w with respect to $\underline{\mathcal{F}}_a$
$\underline{\mathcal{F}}_a$	a vectrix, that is a matrix of unit length physical vectors that form a basis for $\underline{\mathcal{F}}_a$, where $\underline{\mathcal{F}}_a^T = [\underline{a}^1 \ \underline{a}^2 \ \underline{a}^3]$
\mathbf{r}_a	the physical vector \underline{r} resolved in $\underline{\mathcal{F}}_a$
\mathbf{C}_{ab}	a DCM parameterizing the attitude of $\underline{\mathcal{F}}_a$ relative to $\underline{\mathcal{F}}_b$

$$\underline{\omega}^{ba} \qquad \text{angular velocity of } \underline{\mathcal{F}}_b \text{ relative to } \underline{\mathcal{F}}_a$$

Chapter 1

Introduction

The onboard computers of autonomous robots, such as unmanned aerial vehicles (UAV), mobile robots, or autonomous underwater vehicles (AUV), run navigation, guidance, and control algorithms that enable the robotic system to perform desired tasks. The navigation algorithm is responsible for estimating the states of the robot. The guidance algorithm considers planning the path the robot will take to complete its task. Lastly, the controller computes control inputs, such as forces and torques, to be applied so that the robot follows the desired trajectory. These three modules are of equal importance, and are intrinsically linked.

This thesis is focused on the navigation problem, also commonly called the state estimation problem. State estimation is the process of estimating the states of a system given noisy and biased sensor data. For example, an UAV must typically maintain a robust and accurate estimate of its position, velocity, and attitude in order to perform precision tasks, such as parcel delivery or surveillance. However, the sensors onboard UAVs are often of lower quality, to minimize the cost of the system, necessitating a state estimation algorithm that can reliably estimate the position, velocity, and attitude of the UAV given low-quality sensor data.

Several different state estimation techniques exist, each with their advantages and disadvantages. Roughly speaking, they can be separated into batch algorithms, which typically run offline, and sequential algorithms, which typically run in real time. Batch algorithms use sensor data over the entire trajectory to in turn provide an estimate of the states over the entire trajectory. Traditional batch algorithms include the (nonlinear) least-squares formulation [3, Sec. 4.3] and the forward-backward smoother [3, Sec. 3.2.2] and Rauch-Tung-Striebel smoother [3, Sec. 3.2.3]. Batch algorithms are especially useful when reconstructing scenes for metrology or photogrammetry applications, for example. In addition, simultaneous lo-

calization and mapping (SLAM) algorithms are often batch algorithms that do not run in real time.

In real-time applications, sequential state estimation methods are often preferred. The most commonly used algorithms for real-time state estimation are approximations of the Bayes filter [4], such as the Kalman filter, extended Kalman filter (EKF), or sigma-point Kalman filter. Other real-time state estimation methods that leverage concepts from the batch formulation, such as using a bundle of sensor data or iteration, include the sliding window filter [5], iterative extended Kalman filter [3, Sec. 4.2.5], and iterative sigma-point Kalman filter [3, Sec. 4.2.10].

In industry, the EKF is often the algorithm of choice, due to its relative simplicity and its track record of effectiveness. However, it does have its deficiencies. In this thesis, a variant of the EKF, the invariant extended Kalman filter (IEKF) is considered. For a review of the IEKF, see Chapter 3. The main idea behind the invariant filtering framework is that certain problems (i.e., so-called “left-invariant” problems) do not explicitly depend on a particular inertial frame, and others (i.e., so-called “right-invariant” problems) do not explicitly depend on a particular body-fixed frame. Not all estimation problems fit the invariant filtering framework, but when an estimation problem does, extremely appealing properties appear.

1.1 Thesis Objective

The objective of this thesis is to determine how the invariant filtering framework can be used to improve existing state estimation methods. In particular, the contribution of this thesis is an overview of practical considerations of the IEKF and an extension of the invariant estimation theory to the SLAM problem posed in a batch framework.

Another contribution of this thesis is to thoroughly summarize the theory behind the IEKF. This includes some proofs that are missing from the literature. A major contribution of this thesis is to compare the various error definitions that can be used in solving the SLAM problem. This includes a general formulation for performing SLAM when the state can be formulated as an element of a matrix Lie group. Modifying the error definitions leads to Jacobians that may depend less, or not at all, on the state estimate. Lastly, this thesis provides a thorough analysis of the practical implications of the IEKF. The theory behind the IEKF is sound, but the assumptions made often do not hold in practice. Of note, a novel method of using the IEKF in conjunction with a stereo camera is presented.

1.2 Thesis Overview

This thesis is structured as follows.

Chapter 2 summarizes mathematical concepts and notation that are used throughout this thesis.

Chapter 3 outlines the IEKF. The relevant theorems and proofs are presented in continuous and discrete-time. The left-invariant extended Kalman filter and right-invariant extended Kalman filter are then detailed.

In Chapter 4, several examples of the IEKF are presented to illustrate how to practically implement an IEKF and to compare its performance to that of a standard multiplicative extended Kalman filter (MEKF).

In Chapter 5, a solution to the SLAM problem in the invariant framework is presented. Simulation results are shown comparing the novel formulation to more traditional batch-based solutions to the SLAM problem.

This thesis is concluded in Chapter 6, where a summary of the findings are presented, along with recommended future work.

Chapter 2

Preliminaries

2.1 Matrix Lie Groups

In many robotics applications, the states are an element of a matrix Lie groups. Any navigation, guidance, or control algorithm should explicitly accommodate the matrix Lie group nature of the states. As such, matrix Lie group properties and theory will be introduced next.

2.1.1 Overview

This summary of matrix Lie group theory is based on Section 2 of [6]. A matrix Lie group \mathcal{G} , is composed of invertible $n \times n$ matrices that is closed under matrix multiplication. The matrix Lie algebra associated with \mathcal{G} , denoted \mathfrak{g} , is the tangent space around the identity of \mathcal{G} , denoted $T_{\mathbf{I}}\mathcal{G}$. The tangent space of \mathcal{G} at any $\mathbf{X} \in \mathcal{G}$ is denoted $T_{\mathbf{X}}\mathcal{G}$. The matrix Lie algebra is a vector space closed under the operation of the matrix Lie bracket defined $[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA}$, $\forall \mathbf{A}, \mathbf{B} \in \mathfrak{g}$. Furthermore, $\mathbf{XAX}^{-1} \in \mathfrak{g}$, $\forall \mathbf{X} \in \mathcal{G}$, $\forall \mathbf{A} \in \mathfrak{g}$. Any $\mathbf{A} \in \mathfrak{g}$ can be written as $\mathbf{A} = \boldsymbol{\xi}^\wedge = \sum_{i=1}^n \xi_i \mathbf{B}_i$, where $\{\mathbf{B}_1, \dots, \mathbf{B}_n\}$ is a basis for \mathfrak{g} , also known as the generators, and $\mathbf{A} = [\xi_1, \dots, \xi_n]^\top \in \mathbb{R}^n$ is the column matrix of coefficients associated with \mathbf{A} . Alternatively, $\mathbf{A}^\vee = \boldsymbol{\xi}$.

The exponential map takes elements in the Lie algebra and maps them to the Lie group. For matrix Lie groups, the exponential map is simply the matrix exponential. The inverse of the matrix exponential, the matrix logarithm, is also defined and maps elements of the matrix Lie group to the matrix Lie algebra. In more detail,

$$\mathbf{X} = \exp(\boldsymbol{\xi}^\wedge)$$

and

$$\boldsymbol{\xi}^\wedge = \log(\mathbf{X})$$

where $\mathbf{X} \in \mathcal{G}$ and $\boldsymbol{\xi}^\wedge \in \mathfrak{g}$.

The matrix representation of the adjoint operator is used throughout this thesis. It is not unique, as it depends on the parametrization. Denoting the adjoint representation of \mathbf{X} as $\text{Ad}(\mathbf{X})$, then $(\text{Ad}(\mathbf{X})\boldsymbol{\xi})^\wedge = \mathbf{X}\boldsymbol{\xi}^\wedge\mathbf{X}^{-1}$. This leads to the useful identity

$$\mathbf{X} \exp(\boldsymbol{\xi}^\wedge) \mathbf{X}^{-1} = \exp((\text{Ad}(\mathbf{X})\boldsymbol{\xi})^\wedge). \quad (2.1)$$

The adjoint representation of an element of the matrix Lie algebra can also be defined [1, 7]. Given $\boldsymbol{\xi}^\wedge, \boldsymbol{\zeta}^\wedge \in \mathfrak{g}$, the adjoint matrix satisfies $\text{ad}(\boldsymbol{\zeta})\boldsymbol{\xi} = -\text{ad}(\boldsymbol{\xi})\boldsymbol{\zeta}$ and

$$\boldsymbol{\xi}^\wedge\boldsymbol{\zeta}^\wedge - \boldsymbol{\zeta}^\wedge\boldsymbol{\xi}^\wedge = (-\text{ad}(\boldsymbol{\zeta})\boldsymbol{\xi})^\wedge. \quad (2.2)$$

2.1.2 Uncertainty Representations

In standard linear vector spaces, uncertainty is simply additive, such that $\mathbf{x} = \bar{\mathbf{x}} + \delta\mathbf{x}$, where $\delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. However, this is not applicable to matrix Lie groups, as they are not closed under addition. Rather, a multiplicative uncertainty must be used [8]. This leads to two distinct options, namely

$$\mathbf{X} = \bar{\mathbf{X}} \exp(\delta\boldsymbol{\xi}^\wedge), \quad (2.3)$$

$$\mathbf{X} = \exp(\delta\boldsymbol{\xi}^\wedge)\bar{\mathbf{X}}, \quad (2.4)$$

where $\delta\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Note that \mathbf{X} is not normally distributed. Two additional uncertainty definitions can also be defined. They are

$$\mathbf{X} = \bar{\mathbf{X}} \exp(-\delta\boldsymbol{\xi}^\wedge), \quad (2.5)$$

$$\mathbf{X} = \exp(-\delta\boldsymbol{\xi}^\wedge)\bar{\mathbf{X}}, \quad (2.6)$$

defined as the left-invariant and right-invariant uncertainty representations, respectively. They are named as such as they are consistent with left and right-invariant error definitions, which are introduced in Chapter 3.

2.1.3 The Baker-Campbell-Hausdorff Formula

The Baker-Campbell-Hausdorff (BCH) formula is the solution to the equation [3]

$$\mathbf{z}^\wedge = \log(\exp(\mathbf{a}^\wedge) \exp(\mathbf{b}^\wedge)).$$

The detailed solution is available in [3, pp. 230-232]. Herein only a first-order approximation is needed, that being

$$\log(\exp(\mathbf{a}^\wedge) \exp(\mathbf{b}^\wedge)) = \mathbf{a}^\wedge + \mathbf{b}^\wedge.$$

This is exact in the case that $[\mathbf{a}^\wedge, \mathbf{b}^\wedge] = \mathbf{0}$.

2.1.4 Linearization

Any element of a matrix Lie group can be expressed using the exponential map, which is in fact the matrix exponential,

$$\mathbf{X} = \exp(\boldsymbol{\xi}^\wedge).$$

The matrix exponential itself is defined by a power series,

$$\begin{aligned}\exp(\boldsymbol{\xi}^\wedge) &= \sum_{k=0}^{\infty} \frac{1}{k!} (\boldsymbol{\xi}^\wedge)^k \\ &= \mathbf{1} + \boldsymbol{\xi}^\wedge + \frac{(\boldsymbol{\xi}^\wedge)^2}{2} + \frac{(\boldsymbol{\xi}^\wedge)^3}{6} + \dots\end{aligned}$$

Now, consider the case where $\boldsymbol{\xi}$ can be considered small. This small element of \mathbb{R}^d is denoted $\delta\boldsymbol{\xi}$ and $\delta\mathbf{X} = \exp(\delta\boldsymbol{\xi}^\wedge)$. As $\delta\boldsymbol{\xi}$ is already considered small, it is common to assume that terms of order $\mathcal{O}(\|\delta\boldsymbol{\xi}\|^2)$ can be neglected, leading to the approximation

$$\delta\mathbf{X} \approx \mathbf{1} + \delta\boldsymbol{\xi}^\wedge.$$

Thus, the uncertainty representations (2.3) and (2.4) can be approximated as

$$\mathbf{X} = \bar{\mathbf{X}}(\mathbf{1} + \delta\boldsymbol{\xi}^\wedge),$$

$$\mathbf{X} = (\mathbf{1} + \delta\boldsymbol{\xi}^\wedge)\bar{\mathbf{X}},$$

respectively. Similarly for (2.5) and (2.6),

$$\mathbf{X} = \bar{\mathbf{X}}(\mathbf{1} - \delta\boldsymbol{\xi}^\wedge),$$

$$\mathbf{X} = (\mathbf{1} - \delta\boldsymbol{\xi}^\wedge)\bar{\mathbf{X}}.$$

2.2 The Special Orthogonal Group $SO(3)$

The properties of $SO(3)$ are from [3, Ch. 7]. Three dimensional rotations can be represented by the special orthogonal group $SO(3)$,

$$SO(3) = \{\mathbf{C} \in \mathbb{R}^{3 \times 3} \mid \mathbf{C}^\top \mathbf{C} = \mathbf{1}, \det \mathbf{C} = +1\}.$$

The matrix \mathbf{C} is known as a direction cosine matrix (DCM). $SO(3)$ has three degrees of freedom for rotation. The Lie algebra associated with $SO(3)$ is

$$\mathfrak{so}(3) = \{\boldsymbol{\phi}^\times \in \mathbb{R}^{3 \times 3} \mid \boldsymbol{\phi} \in \mathbb{R}^3\},$$

where $\boldsymbol{\phi}^\times$ is the skew-symmetric representation of $\boldsymbol{\phi}$,

$$\boldsymbol{\phi}^\times = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^\times = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}.$$

The adjoint representation of an element of $SO(3)$ is identically that element, $\text{Ad}(\mathbf{C}) = \mathbf{C}$. Similarly, the adjoint representation of an element of $\mathfrak{so}(3)$ is identical to that element, $\text{ad}(\boldsymbol{\phi}) = \boldsymbol{\phi}^\times$. The closed form of the exponential map from $\mathfrak{so}(3)$ to $SO(3)$ is known as the Rodrigues formula,

$$\exp(\boldsymbol{\phi}^\times) = \cos \phi \mathbf{1} + (1 - \cos \phi) \mathbf{a} \mathbf{a}^\top + \sin \phi \mathbf{a}^\times,$$

where $\phi = \|\boldsymbol{\phi}\|$ and $\mathbf{a} = \boldsymbol{\phi}/\phi$. The logarithmic map from $SO(3)$ to $\mathfrak{so}(3)$ is

$$\log(\mathbf{C}) = (\mathbf{a}\phi)^\times,$$

where the angle ϕ is given by

$$\phi = \cos^{-1} \left(\frac{\text{tr}(\mathbf{C}) - 1}{2} \right) + 2\pi m$$

and the axis \mathbf{a} is

$$\mathbf{a} = \frac{1}{2 \sin(\phi)} \begin{bmatrix} \mathbf{C}_{2,3} - \mathbf{C}_{3,2} \\ \mathbf{C}_{3,1} - \mathbf{C}_{1,3} \\ \mathbf{C}_{1,2} - \mathbf{C}_{2,1} \end{bmatrix}.$$

2.3 The Special Euclidean Group $SE(3)$

Poses can be represented by the special euclidean group $SE(3)$ [3, Ch. 7],

$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \mathbf{C} \in SO(3), \mathbf{r} \in \mathbb{R}^3 \right\}.$$

$SE(3)$ has three degrees of freedom for rotation and three for translation, for a total of six.

The inverse of \mathbf{T} is defined as

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{C}^\top & -\mathbf{C}^\top \mathbf{r} \\ \mathbf{0} & 1 \end{bmatrix}.$$

The matrix Lie algebra associated with $SE(3)$ is

$$\mathfrak{se}(3) = \left\{ \boldsymbol{\Xi} = \boldsymbol{\xi}^\wedge \in \mathbb{R}^{4 \times 4} \mid \boldsymbol{\xi} \in \mathbb{R}^6 \right\},$$

where

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\xi}^\phi \\ \boldsymbol{\xi}^r \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\xi}^{\phi \times} & \boldsymbol{\xi}^r \\ \mathbf{0} & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, \quad \boldsymbol{\xi}^\phi, \boldsymbol{\xi}^r \in \mathbb{R}^3.$$

Note that, in [3], $\boldsymbol{\xi}$ is defined in the opposite order, such that $\boldsymbol{\xi}^\phi$ is below $\boldsymbol{\xi}^r$. The convention used here is adopted from [1]. The exponential map from $\mathfrak{se}(3)$ to $SE(3)$ is

$$\exp(\boldsymbol{\xi}^\wedge) = \begin{bmatrix} \exp_{SO(3)}(\boldsymbol{\xi}^{\phi \times}) & \mathbf{J}\boldsymbol{\xi}^r \\ \mathbf{0} & 1 \end{bmatrix},$$

where

$$\mathbf{J} = \frac{\sin \phi}{\phi} \mathbf{1} + \left(1 - \frac{\sin \phi}{\phi} \right) \mathbf{a} \mathbf{a}^\top + \frac{1 - \cos \phi}{\phi} \mathbf{a}^\times, \quad (2.7)$$

where $\phi = \|\boldsymbol{\xi}^\phi\|$ and $\mathbf{a} = \boldsymbol{\xi}^\phi / \phi$. The logarithmic map from $SE(3)$ to $\mathfrak{se}(3)$ is

$$\log(\mathbf{T}) = \begin{bmatrix} \log_{SO(3)}(\mathbf{C}) & \mathbf{J}^{-1} \mathbf{r} \\ \mathbf{0} & 0 \end{bmatrix},$$

where

$$\mathbf{J}^{-1} = \frac{\phi}{2} \cot \frac{\phi}{2} \mathbf{1} + \left(1 - \frac{\phi}{2} \cot \frac{\phi}{2} \right) \mathbf{a} \mathbf{a}^\top - \frac{\phi}{2} \mathbf{a}^\times. \quad (2.8)$$

The adjoint representation of an element of $SE(3)$ is

$$\text{Ad}(\mathbf{T}) = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{r}^\times \mathbf{C} & \mathbf{C} \end{bmatrix} \in \mathbb{R}^{6 \times 6}.$$

The inverse of $\text{Ad}(\mathbf{T})$ is

$$(\text{Ad}(\mathbf{T}))^{-1} = \text{Ad}(\mathbf{T}^{-1}) = \begin{bmatrix} \mathbf{C}^\top & \mathbf{0} \\ -\mathbf{C}^\top \mathbf{r}^\times & \mathbf{C}^\top \end{bmatrix} \in \mathbb{R}^{6 \times 6}.$$

The adjoint representation of an element of $\mathfrak{se}(3)$ is

$$\text{ad}(\boldsymbol{\xi}) = \begin{bmatrix} \boldsymbol{\xi}^{\phi \times} & \mathbf{0} \\ \boldsymbol{\xi}^{\mathbf{r} \times} & \boldsymbol{\xi}^{\phi \times} \end{bmatrix}.$$

2.4 The Group of Double Direct Isometries $SE_2(3)$

Introduced in [9] and explored in detail in [10], $SE_2(3)$, the group of double direct isometries, is

$$SE_2(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{v} & \mathbf{r} \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix} \in \mathbb{R}^{5 \times 5} \mid \mathbf{C} \in SO(3), \mathbf{v}, \mathbf{r} \in \mathbb{R}^3 \right\}.$$

The matrix Lie algebra associated with $SE_2(3)$ is

$$\mathfrak{se}_2(3) = \{ \boldsymbol{\Xi} = \boldsymbol{\xi}^\wedge \in \mathbb{R}^{5 \times 5} \mid \boldsymbol{\xi} \in \mathbb{R}^9 \},$$

where

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\xi}^\phi \\ \boldsymbol{\xi}^{\mathbf{v}} \\ \boldsymbol{\xi}^{\mathbf{r}} \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\xi}^{\phi \times} & \boldsymbol{\xi}^{\mathbf{v}} & \boldsymbol{\xi}^{\mathbf{r}} \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 \end{bmatrix}.$$

The exponential map from $\mathfrak{se}_2(3)$ to $SE_2(3)$ is

$$\exp(\boldsymbol{\xi}^\wedge) = \begin{bmatrix} \exp_{SO(3)}(\boldsymbol{\xi}^{\phi \times}) & \mathbf{J}\boldsymbol{\xi}^{\mathbf{v}} & \mathbf{J}\boldsymbol{\xi}^{\mathbf{r}} \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix},$$

where \mathbf{J} is given by (2.7). The logarithmic map from $SE_2(3)$ to $\mathfrak{se}_2(3)$ is

$$\log(\mathbf{T}) = \begin{bmatrix} \log_{SO(3)}(\mathbf{C}) & \mathbf{J}^{-1}\mathbf{v} & \mathbf{J}^{-1}\mathbf{r} \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 \end{bmatrix},$$

where \mathbf{J}^{-1} is given by (2.8). The adjoint representation of an element of $SE_2(3)$ is

$$\text{Ad}(\mathbf{T}) = \begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{v}^\times \mathbf{C} & \mathbf{C} & \mathbf{0} \\ \mathbf{r}^\times \mathbf{C} & \mathbf{0} & \mathbf{C} \end{bmatrix}.$$

The adjoint representation of an element of $\mathfrak{se}_2(3)$ is

$$\text{ad}(\boldsymbol{\xi}) = \begin{bmatrix} \boldsymbol{\xi}^\phi{}^\times & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\xi}^v{}^\times & \boldsymbol{\xi}^\phi{}^\times & \mathbf{0} \\ \boldsymbol{\xi}^r{}^\times & \mathbf{0} & \boldsymbol{\xi}^\phi{}^\times \end{bmatrix}.$$

2.5 Geometry

A reference frame $\underline{\mathcal{F}}_a$ is defined by three physical basis vectors \underline{a}^1 , \underline{a}^2 , and \underline{a}^3 . In particular, the vectrix $\underline{\mathcal{F}}_a$ can be defined as [11]

$$\underline{\mathcal{F}}_a = \begin{bmatrix} \underline{a}^1 \\ \underline{a}^2 \\ \underline{a}^3 \end{bmatrix}.$$

A physical vector \underline{u} can then be written as

$$\underline{u} = \underline{\mathcal{F}}_a^\top \mathbf{u}_a,$$

where $\underline{u} \in \mathbb{P}$ and $\mathbf{u}_a \in \mathbb{R}^3$ is the physical vector \underline{u} resolved in $\underline{\mathcal{F}}_a$. The orientation of $\underline{\mathcal{F}}_b$ relative to $\underline{\mathcal{F}}_a$ is given by a DCM $\mathbf{C}_{ab} \in SO(3)$. The relationship between \underline{u} resolved in $\underline{\mathcal{F}}_a$, \mathbf{u}_a , and \underline{u} resolved in $\underline{\mathcal{F}}_b$, \mathbf{u}_b , is $\mathbf{u}_a = \mathbf{C}_{ab} \mathbf{u}_b$.

2.6 Kinematics

The position of point z relative to point w is described by the physical vector \underline{r}^{zw} . The rate of change of \underline{r}^{zw} with respect to $\underline{\mathcal{F}}_a$ is denoted $\dot{\underline{r}}^{zw}{}^a = \underline{v}^{zw/a}$. Similarly, $\dot{\underline{r}}^{zw}{}^{a\bullet a} = \underline{v}^{zw/a\bullet a} = \underline{a}^{zw/a/a}$. These physical vectors can all then be resolved in a frame, as appropriate. Poisson's equation is

$$\dot{\mathbf{C}}_{ab} = \mathbf{C}_{ab} \boldsymbol{\omega}_b^{ba}{}^\times,$$

where $\boldsymbol{\omega}_b^{ba}$ is the angular velocity of \mathcal{F}_b relative to \mathcal{F}_a resolved in \mathcal{F}_b . When discretized, Poisson's equation becomes

$$\mathbf{C}_{ab_k} = \mathbf{C}_{ab_{k-1}} \exp \left(\left(T \boldsymbol{\omega}_{b_{k-1}}^{b_{k-1}a} \right)^\times \right),$$

where $T = t_k - t_{k-1}$.

2.7 Optimization

Consider a standard optimization problem

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} J(\mathbf{x}),$$

where \mathbf{x}^* is the minimizing solution of the cost function $J(\mathbf{x})$. Two types of optimization problems are of interest here, namely linear and nonlinear least squares.

2.7.1 Linear Least Squares

Consider a linear system that can be written

$$\mathbf{Ax} = \mathbf{b},$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{b} \in \mathbb{R}^m$. Define the error to be $\boldsymbol{\rho}(\mathbf{x}) = \mathbf{Ax} - \mathbf{b}$. The objective function to be minimized is

$$J(\mathbf{x}) = \frac{1}{2} \boldsymbol{\rho}(\mathbf{x})^\top \boldsymbol{\rho}(\mathbf{x}). \quad (2.9)$$

The solution minimizing (2.9) is

$$\mathbf{x}^* = (\mathbf{A}^\top \mathbf{A})^{-1} (\mathbf{A}^\top \mathbf{b}).$$

Explicitly computing the inverse of $\mathbf{A}^\top \mathbf{A}$ can be computationally costly. To alleviate these costs, a Cholesky factorization can be used to decompose $\mathbf{A}^\top \mathbf{A}$. This leads to

$$\mathbf{A}^\top \mathbf{A} = \mathbf{L} \mathbf{L}^\top,$$

where \mathbf{L} is lower triangular. The linear least squares problem can then be rewritten as

$$\mathbf{L} \mathbf{L}^\top \mathbf{x}^* = \mathbf{A}^\top \mathbf{b}.$$

Letting $\mathbf{L}^\top \mathbf{x}^* = \mathbf{z}$, it is possible to solve

$$\mathbf{L}^\top \mathbf{z} = \mathbf{A}^\top \mathbf{b}$$

for \mathbf{z} via forwards substitution. The minimizing solution \mathbf{x}^* is then found by solving $\mathbf{L}\mathbf{x}^* = \mathbf{z}$ using backward substitution.

2.7.2 Nonlinear Least Squares

This section is based on [3]. In practice, the error function $\rho(\mathbf{x})$ is often not linear, and is instead some nonlinear function of \mathbf{x} . In this case, a Taylor series expansion of the cost function is used to linearize the problem. Consider a function $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$. A Taylor series expansion of f is

$$f(\mathbf{x}^{\text{op}} + \delta\mathbf{x}) = f(\mathbf{x}^{\text{op}}) + \left[\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}^{\text{op}}} \right] \delta\mathbf{x} + \frac{1}{2} \delta\mathbf{x}^\top \left[\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top} \Big|_{\mathbf{x}=\mathbf{x}^{\text{op}}} \right] \delta\mathbf{x} + \mathcal{O}(\|\delta\mathbf{x}^3\|),$$

where \mathbf{x}^{op} is the operating point. The Jacobian of f is

$$\nabla f(\mathbf{x}^{\text{op}}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}^{\text{op}}}$$

and the Hessian of f is

$$\nabla^2 f(\mathbf{x}^{\text{op}}) = \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top} \Big|_{\mathbf{x}=\mathbf{x}^{\text{op}}}.$$

2.7.2.1 Newton's Method [3]

Taking a second-order Taylor series expansion of the cost function $J(\mathbf{x})$ yields

$$J(\mathbf{x}^{\text{op}} + \delta\mathbf{x}) = J(\mathbf{x}^{\text{op}}) + \nabla J(\mathbf{x}^{\text{op}}) \delta\mathbf{x} + \frac{1}{2} \delta\mathbf{x}^\top \nabla^2 J(\mathbf{x}^{\text{op}}) \delta\mathbf{x}. \quad (2.10)$$

To minimize the cost function, the derivative of (2.10) with respect to $\delta\mathbf{x}$ is computed and set to zero,

$$\begin{aligned} \nabla J(\mathbf{x}^{\text{op}}) + \delta\mathbf{x}^* \nabla^2 J(\mathbf{x}^{\text{op}}) &= 0, \\ \nabla^2 J(\mathbf{x}^{\text{op}}) \delta\mathbf{x}^* &= -\nabla J(\mathbf{x}^{\text{op}}). \end{aligned} \quad (2.11)$$

Assuming the Hessian is positive definite and therefore invertible, (2.11) can be solved. The operating point is then updated,

$$\mathbf{x}^{\text{op}} \leftarrow \mathbf{x}^{\text{op}} + \delta\mathbf{x}^*.$$

To improve the performance of Newton's method, it is common to multiply the update by a step length $\alpha > 0$, leading to

$$\mathbf{x}^{\text{op}} \leftarrow \mathbf{x}^{\text{op}} + \alpha \delta \mathbf{x}^*.$$

A backtracking procedure is used to find the step length [12, p. 37]. This method is summarized in Algorithm 2.1. The parameter c is typically chosen to be small (approximately 10^{-4}) and ρ is tuned to a desired convergence rate. In Newton methods, $\bar{\alpha} = 1$.

Algorithm 2.1

- 1: Select $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$
 - 2: Set $\alpha \leftarrow \bar{\alpha}$
 - 3: **while** $J(\mathbf{x}^{\text{op}} + \alpha \delta \mathbf{x}^*) > J(\mathbf{x}^{\text{op}}) + c\alpha \nabla J(\mathbf{x}^{\text{op}})^T \delta \mathbf{x}^*$ **do**
 - 4: $\alpha \leftarrow \rho \alpha$
 - 5: **end while**
-

2.7.2.2 Gauss-Newton Method

The Gauss-Newton method [3] is applicable when the cost function is a nonlinear least squares cost function,

$$J(\mathbf{x}) = \frac{1}{2} \boldsymbol{\rho}(\mathbf{x})^T \boldsymbol{\rho}(\mathbf{x}).$$

A first-order Taylor series expansion yields

$$\begin{aligned} J(\mathbf{x}^{\text{op}} + \delta \mathbf{x}) &= \frac{1}{2} \boldsymbol{\rho}(\mathbf{x}^{\text{op}} + \delta \mathbf{x})^T \boldsymbol{\rho}(\mathbf{x}^{\text{op}} + \delta \mathbf{x}), \\ 2J(\mathbf{x}^{\text{op}} + \delta \mathbf{x}) &= (\boldsymbol{\rho}(\mathbf{x}^{\text{op}}) + \nabla J(\mathbf{x}^{\text{op}}) \delta \mathbf{x})^T (\boldsymbol{\rho}(\mathbf{x}^{\text{op}}) + \nabla J(\mathbf{x}^{\text{op}}) \delta \mathbf{x}) \\ &= \boldsymbol{\rho}(\mathbf{x}^{\text{op}})^T \boldsymbol{\rho}(\mathbf{x}^{\text{op}}) + \boldsymbol{\rho}(\mathbf{x}^{\text{op}})^T \nabla J(\mathbf{x}^{\text{op}}) \delta \mathbf{x} + (\nabla J(\mathbf{x}^{\text{op}}) \delta \mathbf{x})^T \boldsymbol{\rho}(\mathbf{x}^{\text{op}}) + \\ &\quad (\nabla J(\mathbf{x}^{\text{op}}) \delta \mathbf{x})^T \nabla J(\mathbf{x}^{\text{op}}) \delta \mathbf{x}. \end{aligned} \tag{2.12}$$

Once again, the derivative of (2.12) is taken with respect to $\delta \mathbf{x}$ and set to zero, yielding

$$2 \frac{\partial J(\mathbf{x}^{\text{op}} + \delta \mathbf{x})}{\partial \delta \mathbf{x}} = 2 \boldsymbol{\rho}(\mathbf{x}^{\text{op}})^T \nabla J(\mathbf{x}^{\text{op}}) + 2 \delta \mathbf{x}^T \nabla J(\mathbf{x}^{\text{op}})^T \nabla J(\mathbf{x}^{\text{op}}) = 0,$$

leading to

$$\nabla J(\mathbf{x}^{\text{op}})^T \nabla J(\mathbf{x}^{\text{op}}) \delta \mathbf{x} = -\nabla J(\mathbf{x}^{\text{op}})^T \boldsymbol{\rho}(\mathbf{x}^{\text{op}}).$$

This is often written

$$(\mathbf{H}\mathbf{H}^T) \delta \mathbf{x} = -\mathbf{H}^T \boldsymbol{\rho}(\mathbf{x}^{\text{op}}),$$

where

$$\mathbf{H} = \nabla J(\mathbf{x}^{\text{op}}).$$

This is iterated until convergence using the technique described in Section 2.7.2.1 .

2.7.2.3 Levenberg-Marquardt Method

A small modification to the Gauss-Newton method leads to the Levenberg-Marquardt method [13–15],

$$(\mathbf{H}\mathbf{H}^T + \lambda \text{diag}(\mathbf{H}\mathbf{H}^T)) \delta\mathbf{x} = -\mathbf{H}^T \boldsymbol{\rho}(\mathbf{x}^{\text{op}}),$$

where $\lambda \geq 0$ is a damping factor. This damping factor allows the condition of the Hessian to be improved. Multiplying by the diagonal elements of the Hessian allows for the scaling of the problem to be preserved. The parameter λ can be found using a technique such as the one shown in Algortihm 2.2. The value of λ is chosen such that the computed update will result in a decrease of the objective function. If the initial λ is too low, it is increased.

Algorithm 2.2

- 1: Set $\lambda = 0$
 - 2: Compute $\delta\mathbf{x}^*$
 - 3: Set $\lambda \geq 0$
 - 4: **while** $J(\mathbf{x}^{\text{op}} + \delta\mathbf{x}^*) > J(\mathbf{x}^{\text{op}})$ **do**
 - 5: $\lambda \leftarrow 10\lambda$
 - 6: Recompute $\delta\mathbf{x}^*$
 - 7: **end while**
-

2.8 State Estimation

A robotic system can be described by a set of states. These states often include position, attitude, and any other quantity that can help describe the motion of the body. Two methods of state estimation are considered herein, namely the Kalman filter and batch estimation.

2.8.1 Extended Kalman Filtering

The Kalman filter and its nonlinear variant, the extended Kalman filter (EKF), are two of the most common state estimation algorithms used today. Only the EKF is presented here.

Consider nonlinear process and measurement models given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}), \quad (2.13)$$

$$\mathbf{y}_k = \mathbf{g}_k(\mathbf{x}_k, \mathbf{v}_k), \quad (2.14)$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ is the state, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the interoceptive measurement, $\mathbf{w} \in \mathbb{R}^{n_w}$ is the interoceptive measurement, or process, noise, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is the exteroceptive measurement at time t_k and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the exteroceptive measurement noise. The process noise is white and band-limited, and, when discretized, is normally distributed with zero mean and covariance \mathbf{Q}_k .

To implement an EKF, the process and measurement models must be linearized about an operating point. A first-order Taylor series expansion of (2.13) yields

$$\delta \dot{\mathbf{x}} = \mathbf{A} \delta \mathbf{x} + \mathbf{L} \delta \mathbf{w},$$

where

$$\begin{aligned} \mathbf{A} &= \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}, \mathbf{u}, \bar{\mathbf{w}}}, \\ \mathbf{L} &= \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{w}} \right|_{\bar{\mathbf{x}}, \mathbf{u}, \bar{\mathbf{w}}}, \end{aligned}$$

are the process model Jacobians evaluated at the nominal solution. Similarly, a first-order Taylor series expansion of (2.14) yields

$$\delta \mathbf{y}_k = \mathbf{H}_k \delta \mathbf{x}_k + \mathbf{L}_k \delta \mathbf{v}_k,$$

where

$$\begin{aligned} \mathbf{H}_k &= \left. \frac{\partial \mathbf{g}_k(\mathbf{x}_k, \mathbf{v}_k)}{\partial \mathbf{x}_k} \right|_{\bar{\mathbf{x}}_k, \bar{\mathbf{v}}_k}, \\ \mathbf{M}_k &= \left. \frac{\partial \mathbf{g}_k(\mathbf{x}_k, \mathbf{v}_k)}{\partial \mathbf{v}_k} \right|_{\bar{\mathbf{x}}_k, \bar{\mathbf{v}}_k}, \end{aligned}$$

are the measurement model Jacobians evaluated at the nominal solution. In an EKF, the nominal noise values are assumed to be $\bar{\mathbf{w}} = \mathbf{0}$ and $\bar{\mathbf{v}}_k = \mathbf{0}$, and the nominal value of the state is the best estimate provided by the filter.

As interoceptive measurements are available, the state estimate is predicted by integrating

(2.13),

$$\check{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \int_{t_{k-1}}^{t_k} \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}, \mathbf{0}) dt,$$

where $(\hat{\cdot})$ is used to denote the corrected state and $(\check{\cdot})$ is used to denote the predicted state. The covariance is predicted using

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^T.$$

When exteroceptive measurements are available, the state estimate is corrected. The correction equations are

$$\begin{aligned} \mathbf{K}_k &= \check{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \check{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{M}_k \check{\mathbf{R}}_k \mathbf{M}_k^T)^{-1}, \\ \hat{\mathbf{x}}_k &= \check{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{y}_k - \mathbf{g}(\check{\mathbf{x}}_k, \mathbf{0})), \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \check{\mathbf{P}}_k (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^T \mathbf{K}_k^T, \end{aligned}$$

where \mathbf{K}_k is the Kalman gain.

2.8.2 Batch Estimation [3, pp.127-143]

Batch estimation is useful in situations when the estimation algorithm does not need to run in real time. Once again, only the nonlinear variant of batch estimation is presented. The state at each time step composes the trajectory

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix},$$

where $t \in [t_0, t_n]$. The discrete-time kinematics are given by

$$\mathbf{x}_k = \mathbf{f}_{k-1} (\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}),$$

where \mathbf{u}_{k-1} are the interoceptive measurements, or inputs, and $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ is zero-mean white noise. The exteroceptive measurements are modelled as

$$\mathbf{y}_k = \mathbf{g}_k(\mathbf{x}_k) + \mathbf{v}_k$$

where $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. A batch maximum *a posteriori* method is used to solve the batch estimation problem. The input errors are

$$\begin{aligned}\mathbf{e}_{u,0}(\mathbf{x}) &= \check{\mathbf{x}}_0 - \mathbf{x}_0, \\ \mathbf{e}_{u,k}(\mathbf{x}) &= \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) - \mathbf{x}_k, \quad k = 1, \dots, n,\end{aligned}$$

where $\check{\mathbf{x}}_0$ is the initial state estimate, $\check{\mathbf{x}}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0)$. The errors in the exteroceptive measurements are

$$\mathbf{e}_{y,k}(\mathbf{x}) = \mathbf{y}_k - \mathbf{g}_k(\mathbf{x}_k), \quad k = 1, \dots, n.$$

The objective function to minimize is

$$J(\mathbf{x}) = \frac{1}{2} \mathbf{e}_{u,0}(\mathbf{x})^\top \mathbf{W}_{u,0}^{-1} \mathbf{e}_{u,0}(\mathbf{x}) + \frac{1}{2} \sum_{k=1}^n \mathbf{e}_{u,k}(\mathbf{x})^\top \mathbf{W}_{u,k}^{-1} \mathbf{e}_{u,k}(\mathbf{x}) + \frac{1}{2} \sum_{k=1}^n \mathbf{e}_{y,k}(\mathbf{x})^\top \mathbf{W}_{y,k}^{-1} \mathbf{e}_{y,k}(\mathbf{x}),$$

where $\mathbf{W}_{u,0}$, $\mathbf{W}_{u,k}$, and $\mathbf{W}_{y,k}$ are weighting matrices related to the error distribution. The error is deemed Gaussian, as it is assumed that it arises only due to the presence of Gaussian noise in the measurements. Further defining

$$\mathbf{e}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_{u,0}(\mathbf{x}) \\ \vdots \\ \mathbf{e}_{u,n}(\mathbf{x}) \\ \hline \mathbf{e}_{y,1}(\mathbf{x}) \\ \vdots \\ \mathbf{e}_{y,n}(\mathbf{x}) \end{bmatrix},$$

and $\mathbf{W}_u = \text{diag}(\mathbf{W}_{u,0}, \mathbf{W}_{u,1}, \dots, \mathbf{W}_{u,n})$, $\mathbf{W}_y = \text{diag}(\mathbf{W}_{y,1}, \dots, \mathbf{W}_{y,n})$ and $\mathbf{W} = \text{diag}(\mathbf{W}_u, \mathbf{W}_y)$, the objective function is rewritten as

$$J(\mathbf{x}) = \frac{1}{2} \mathbf{e}(\mathbf{x})^\top \mathbf{W}^{-1} \mathbf{e}(\mathbf{x}).$$

To minimize this objective function, linearize the errors about an operating point \mathbf{x}^{op} . The linearized prior error has the form

$$\mathbf{e}_{u,0}(\mathbf{x}) = \mathbf{e}_{u,0}(\mathbf{x}^{\text{op}}) - \mathbf{F}_0^2 \delta \mathbf{\epsilon}_0,$$

where $\delta\boldsymbol{\epsilon}_k$ is the error between the operating and truth trajectory. The linearized input error has the form

$$\mathbf{e}_{u,k}(\mathbf{x}) = \mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) + \mathbf{F}_k^1 \delta\boldsymbol{\epsilon}_{k-1} - \mathbf{F}_k^2 \delta\boldsymbol{\epsilon}_k.$$

The linearized measurement error has the form

$$\mathbf{e}_{y,k}(\mathbf{x}) = \mathbf{e}_{y,k}(\mathbf{x}^{\text{op}}) - \mathbf{H}_k \delta\boldsymbol{\epsilon}_k.$$

By stacking the estimation errors

$$\delta\mathbf{x} = \begin{bmatrix} \delta\boldsymbol{\epsilon}_0 \\ \vdots \\ \delta\boldsymbol{\epsilon}_n \end{bmatrix},$$

the linearized system can then be written

$$\mathbf{e}(\mathbf{x}) = \mathbf{e}(\mathbf{x}^{\text{op}}) - \boldsymbol{\Gamma} \delta\mathbf{x},$$

where

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{A}^{-1} \\ \mathbf{H} \end{bmatrix},$$

where

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{F}_0^2 & & & \\ \mathbf{F}_1^1 & \mathbf{F}_1^2 & & \\ & \ddots & \ddots & \\ & & \mathbf{F}_{n-1}^1 & \mathbf{F}_{n-1}^2 \end{bmatrix},$$

and

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & & \\ & \ddots & \\ & & \mathbf{H}_n \end{bmatrix}.$$

The linearized objective function is

$$J(\mathbf{x} + \mathbf{x}^{\text{op}}) = (\mathbf{e}(\mathbf{x}^{\text{op}}) + \boldsymbol{\Gamma} \delta\mathbf{x})^\top \mathbf{W}^{-1} (\mathbf{e}(\mathbf{x}^{\text{op}}) + \boldsymbol{\Gamma} \delta\mathbf{x}) \quad (2.15)$$

Minimizing (2.15) with respect to $\delta\mathbf{x}$ yields the Gauss-Newton update,

$$(\boldsymbol{\Gamma}^\top \mathbf{W}^{-1} \boldsymbol{\Gamma}) \delta\mathbf{x} = \boldsymbol{\Gamma}^\top \mathbf{W}^{-1} \mathbf{e}(\mathbf{x}^{\text{op}}),$$

which can be iteratively solved for the minimizing solution $\delta\mathbf{x}^*$.

2.9 Summary

The tools presented in this chapter are used throughout this thesis. The invariant filtering theory applies directly to problems defined on matrix Lie groups. Matrix lie group theory, along with knowledge of geometry and kinematics, are used to build models. The optimization techniques detailed here are extensively used in solving SLAM problems and are essential to understanding the material presented in Chapter 5. Lastly, this thesis builds on several well established state estimation techniques.

Chapter 3

Invariant Extended Kalman Filtering

For many robotics applications, the extended Kalman filter (EKF) is the state estimation algorithm of choice. Unlike its linear counterpart, the EKF does not have any global convergence properties. The EKF, when treated as an observer, has been shown to have locally convergent error dynamics when the initial error in the state estimate is sufficiently small [16, 17]. This is due to the fact that, in general, the Jacobians used in an EKF depend on the states. As these states are not known, only the best estimate can be used. When this estimate is far from the true state, the Jacobians can be inaccurate, leading to poor filter performance, and, in some cases, divergence. However, in practice, the EKF is often sufficient, as demonstrated by its pervasiveness in many state estimation applications.

In recent years, Barrau and Bonnabel have introduced the invariant extended Kalman filter (IEKF) [1], building on the theory of symmetry-preserving observers on matrix Lie groups [18, 19]. The IEKF exploits the fact that, in robotics, the estimated states are not elements of a linear vector space, but are rather elements of a matrix Lie group. For a certain class of systems, known as group-affine systems, Barrau and Bonnabel show that a careful definition of the error leads to state-independent error dynamics [1]. This implies the process model Jacobian is state independent. Furthermore, for specific measurement models, the measurement model Jacobian is also state-independent. Using the fact that the Jacobians are state-independent, it can be shown that the IEKF is a locally asymptotically stable observer, no matter the trajectory. An estimate for $\mathbf{x}(t)$, denoted $\hat{\mathbf{x}}(t)$, is said to be locally asymptotically stable if $\forall \epsilon > 0, \exists \delta(\epsilon) > 0$ such that

$$\|\mathbf{x}(0) - \hat{\mathbf{x}}(t)\| < \delta(\epsilon) \implies \|\phi(t, \mathbf{x}(0), \hat{\mathbf{x}}(t))\| < \epsilon, \forall t \geq 0$$

and $\exists \eta > 0$ such that

$$\|\mathbf{x}(0) - \hat{\mathbf{x}}(t)\| < \eta \implies \lim_{t \rightarrow \infty} \|\phi(t, \mathbf{x}(0), \hat{\mathbf{x}}(t))\| = 0.$$

In this chapter, the IEKF theory is presented, along with relevant proofs. The continuous-time [1] and discrete-time [2] variants are shown. For each, both the left-invariant extended Kalman filter (LIEKF) and right-invariant extended Kalman filter (RIEKF) are presented.

3.1 Invariant Filtering in Continuous Time

This section is a summary of the results from [1]. In addition, the proofs for a right-invariant error definition are included here. Let $\mathcal{G} \subset \mathbb{R}^{n \times n}$ be a matrix Lie group. Denote its matrix Lie algebra $\mathfrak{g} \subset \mathbb{R}^{d \times d}$. Suppose the evolution of the system can be described by the differential equation

$$\dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{u}(t)) + \mathbf{W}(t), \quad (3.1)$$

where $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is an input variable, $\mathbf{X}(t) \in \mathcal{G}$ is the state, and $\mathbf{W}(t) \in \mathfrak{g}$ is band-limited white noise. The noise in \mathbb{R}^d is $\mathbf{W}(t)^\vee = \mathbf{w}(t)$. The argument of time will be suppressed throughout for brevity. When discretized, the noise at time t_k is $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$. An alternative model is $\dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{u}(t)) + \mathbf{W}(t)\mathbf{X}(t)$, however, problems found in real-world applications almost never have this form. As such, the form given in (3.1) is exclusively considered for the remainder of this thesis. The function $\mathbf{F}(\mathbf{X}, \mathbf{u})$ is said to be group affine if it satisfies

$$\mathbf{F}(\mathbf{X}_1 \mathbf{X}_2, \mathbf{u}) = \mathbf{F}(\mathbf{X}_1, \mathbf{u}) \mathbf{X}_2 + \mathbf{X}_1 \mathbf{F}(\mathbf{X}_2, \mathbf{u}) - \mathbf{X}_1 \mathbf{F}(\mathbf{1}, \mathbf{u}) \mathbf{X}_2, \quad (3.2)$$

where $\mathbf{X}_1, \mathbf{X}_2 \in \mathcal{G}$.

Consider the true state \mathbf{X} and the estimated state $\hat{\mathbf{X}}$. The left and right-invariant errors are

$$\delta \mathbf{X}^L = \mathbf{X}^{-1} \hat{\mathbf{X}}, \quad (3.3)$$

$$\delta \mathbf{X}^R = \hat{\mathbf{X}} \mathbf{X}^{-1}, \quad (3.4)$$

respectively. The errors are said to have state-independent propagation if their derivative with respect to time satisfies the differential equation

$$\dot{\delta \mathbf{X}} = \mathbf{G}(\delta \mathbf{X}, \mathbf{u}). \quad (3.5)$$

Theorem 3.1.1. For (3.1), where noise has been neglected, the following three statements are equivalent.

1. The left invariant error propagation (3.3) is state independent.
2. The right invariant error propagation (3.4) is state independent.
3. Equation (3.2) is satisfied.

Note that the case of left and right-invariant functions is captured by (3.2). The proofs shown below are based on [1, 9]. *Proof* It will be shown that 1 \implies 3. The proof will use the identity

$$\begin{aligned}
\frac{d}{dt} \mathbf{X}^{-1} &= \frac{d}{dt} (\mathbf{X}^{-1} \mathbf{X} \mathbf{X}^{-1}) \\
&= \frac{d}{dt} \mathbf{X}^{-1} + \mathbf{X}^{-1} \frac{d}{dt} (\mathbf{X} \mathbf{X}^{-1}) \\
&= \frac{d}{dt} \mathbf{X}^{-1} + \mathbf{X}^{-1} \dot{\mathbf{X}} \mathbf{X}^{-1} + \mathbf{X}^{-1} \mathbf{X} \frac{d}{dt} \mathbf{X}^{-1} \\
&= -\mathbf{X}^{-1} \dot{\mathbf{X}} \mathbf{X}^{-1}.
\end{aligned} \tag{3.6}$$

Given the left-invariant error (3.3) and using (3.6), write (3.5) as

$$\begin{aligned}
\mathbf{G}(\mathbf{X}_a^{-1} \mathbf{X}_b, \mathbf{u}) &= \frac{d}{dt} (\mathbf{X}_a^{-1} \mathbf{X}_b) \\
&= \frac{d}{dt} \mathbf{X}_a^{-1} \mathbf{X}_b + \mathbf{X}_a^{-1} \dot{\mathbf{X}}_b \\
&= -\mathbf{X}_a^{-1} \dot{\mathbf{X}}_a \mathbf{X}_a^{-1} \mathbf{X}_b + \mathbf{X}_a^{-1} \mathbf{F}(\mathbf{X}_b, \mathbf{u}) \\
&= -\mathbf{X}_a^{-1} \mathbf{F}(\mathbf{X}_a, \mathbf{u}) \mathbf{X}_a^{-1} \mathbf{X}_b + \mathbf{X}_a^{-1} \mathbf{F}(\mathbf{X}_b, \mathbf{u}),
\end{aligned} \tag{3.7}$$

where $\mathbf{X}_a, \mathbf{X}_b \in \mathcal{G}$. Letting $\mathbf{X}_b = \mathbf{X}_2$ and $\mathbf{X}_a = \mathbf{1}$, (3.7) is

$$\mathbf{G}(\mathbf{X}_2, \mathbf{u}) = -\mathbf{F}(\mathbf{1}, \mathbf{u}) \mathbf{X}_2 + \mathbf{F}(\mathbf{X}_2, \mathbf{u}). \tag{3.8}$$

Next, let $\mathbf{X}_b = \mathbf{X}_1 \mathbf{X}_2$ and $\mathbf{X}_a = \mathbf{X}_1$ in (3.7) to get

$$\begin{aligned}
\mathbf{G}(\mathbf{X}_1^{-1} \mathbf{X}_1 \mathbf{X}_2, \mathbf{u}) &= -\mathbf{X}_1^{-1} \mathbf{F}(\mathbf{X}_1, \mathbf{u}) \mathbf{X}_1^{-1} \mathbf{X}_1 \mathbf{X}_2 + \mathbf{X}_1^{-1} \mathbf{F}(\mathbf{X}_1 \mathbf{X}_2, \mathbf{u}), \\
\mathbf{G}(\mathbf{X}_2, \mathbf{u}) &= -\mathbf{X}_1^{-1} \mathbf{F}(\mathbf{X}_1, \mathbf{u}) \mathbf{X}_2 + \mathbf{X}_1^{-1} \mathbf{F}(\mathbf{X}_1 \mathbf{X}_2, \mathbf{u}).
\end{aligned} \tag{3.9}$$

Equating (3.8) and (3.9),

$$\begin{aligned}-\mathbf{X}_1^{-1}\mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{X}_2 + \mathbf{X}_1^{-1}\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) &= -\mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X}_2 + \mathbf{F}(\mathbf{X}_2, \mathbf{u}), \\ \mathbf{X}_1^{-1}\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) &= \mathbf{X}_1^{-1}\mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{X}_2 + \mathbf{F}(\mathbf{X}_2, \mathbf{u}) - \mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X}_2, \\ \mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{X}_2 + \mathbf{X}_1\mathbf{F}(\mathbf{X}_2, \mathbf{u}) - \mathbf{X}_1\mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X}_2,\end{aligned}$$

which is (3.2). Thus, 1 \implies 3. Proving 2 \implies 3 is analogous. To prove 3 \implies 1, begin with (3.2) and rearrange to yield

$$\begin{aligned}\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) &= \mathbf{X}_1\mathbf{F}(\mathbf{X}_2, \mathbf{u}) + \mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{X}_2 - \mathbf{X}_1\mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X}_2, \\ \mathbf{X}_1^{-1}\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_2, \mathbf{u}) + \mathbf{X}_1^{-1}\mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{X}_2 - \mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X}_2, \\ -\mathbf{X}_1^{-1}\mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{X}_2 + \mathbf{X}_1^{-1}\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_2, \mathbf{u}) - \mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X}_2.\end{aligned}\tag{3.10}$$

Letting $\mathbf{X}_1 = \mathbf{X}_a$ and $\mathbf{X}_2 = \mathbf{X}_a^{-1}\mathbf{X}_b$, (3.10) becomes

$$\begin{aligned}-\mathbf{X}_a^{-1}\mathbf{F}(\mathbf{X}_a, \mathbf{u})\mathbf{X}_a^{-1}\mathbf{X}_b + \mathbf{X}_a^{-1}\mathbf{F}(\mathbf{X}_a\mathbf{X}_a^{-1}\mathbf{X}_b, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_a^{-1}\mathbf{X}_b, \mathbf{u}) - \mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X}_a^{-1}\mathbf{X}_b, \\ -\mathbf{X}_a^{-1}\dot{\mathbf{X}}_a\mathbf{X}_a^{-1}\mathbf{X}_b + \mathbf{X}_a^{-1}\dot{\mathbf{X}}_b &= \mathbf{F}(\mathbf{X}_a^{-1}\mathbf{X}_b, \mathbf{u}) - \mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X}_a^{-1}\mathbf{X}_b.\end{aligned}\tag{3.11}$$

Using (3.6) and defining $\mathbf{X}_a^{-1}\mathbf{X}_b$ to be the left-invariant error $\delta\mathbf{X}^L$, (3.11) is

$$\begin{aligned}\frac{d}{dt}\mathbf{X}_a^{-1}\mathbf{X}_b + \mathbf{X}_a^{-1}\dot{\mathbf{X}}_b &= \mathbf{G}(\delta\mathbf{X}^L, \mathbf{u}), \\ \frac{d}{dt}(\mathbf{X}_a^{-1}\mathbf{X}_b) &= \mathbf{G}(\delta\mathbf{X}^L, \mathbf{u}), \\ \dot{\delta\mathbf{X}}^L &= \mathbf{G}(\delta\mathbf{X}^L, \mathbf{u}).\end{aligned}$$

which is (3.5). Thus, 3 \implies 1. The proof that 3 \implies 2 is similar.

Theorem 3.1.2. Consider the error $\delta\mathbf{X}$ between two trajectories and let \mathbf{A} be defined such that

$$\mathbf{G}(\exp(\xi^\wedge)) = \mathbf{A}\xi + \mathcal{O}(\|\delta\xi\|^2).$$

Let $\xi(0)$ be defined such that $\exp(\xi(0)^\wedge) = \delta\mathbf{X}(0)$. If ξ is defined for $t > 0$ by the linear differential equation

$$\frac{d}{dt}\xi = \mathbf{A}\xi,\tag{3.12}$$

then

$$\delta\mathbf{X} = \exp(\xi^\wedge), \forall t \geq 0.$$

Theorem 3.1.2 implies that the nonlinear error can be recovered exactly from a time-varying linear differential equation. The proof of this theorem can be found in [1].

Theorems 3.1.1 and 3.1.2 form the basis of invariant filtering. Building on these theorems, it can be shown that the IEKF is a locally asymptotically stable observer. The proof, which can be found in [1], is outside the scope of this thesis. However, it can be understood through an analogy with the Kalman filter. For a linear system, the Kalman filter is asymptotically stable, meaning the norm of the error goes to zero as time goes to infinity. The regular EKF does not have the same stability characteristics because the differential equation describing the evolution of the error is only accurate to first order. However, as seen in (3.12), the linearized error propagation in an IEKF is exact. As such, the stability properties of the IEKF is then more akin to the linear Kalman filter. Note, this is just a high-level overview of why the IEKF possess such stability properties. The proof in [1] should be consulted for a rigorous explanation of local asymptotic stability.

3.2 Invariant Filtering in Discrete Time

The discrete-time invariant extended Kalman filter was introduced in [2]. However, only the left-invariant case is presented. Here, both the left and right-invariant cases are considered. The proofs shown herein are also either incomplete or missing in the literature [20, 21]. The proofs are similar to those presented in Section 3.1. However, the definition of a group-affine function is different for a discrete-time process model, and Theorem 3.1.1 must be altered.

Consider the discrete-time system

$$\mathbf{X}_k = \mathbf{F}_{k-1}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}). \quad (3.13)$$

Equation (3.13) represents the most general case, as using (3.13) allows for the noise to enter the system in a physically meaningful way. However, an approximation is often used to ease derivations. This approximation makes use of the uncertainty representations described in Section 2.1.2. Recall the definition of the left-invariant error (3.3). Letting $\mathbf{F}_{k-1}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})$ be the true state, the perturbed state can be written as

$$\mathbf{X}_k = \mathbf{F}_{k-1}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}) \exp(\mathbf{w}_{k-1}^\wedge). \quad (3.14)$$

Similarly, for a right-invariant error,

$$\mathbf{X}_k = \exp(\mathbf{w}_{k-1}^\wedge) \mathbf{F}_{k-1}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}). \quad (3.15)$$

The discrete-time model must be group affine. As the group-affine definition neglects noise, the choice of model is irrelevant. Choosing the discrete-time model (3.13), a discrete-time group affine function satisfies

$$\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}, \mathbf{0}) = \mathbf{F}(\mathbf{X}_1, \mathbf{u}, \mathbf{0})\mathbf{F}(\mathbf{1}, \mathbf{u}, \mathbf{0})^{-1}\mathbf{F}(\mathbf{X}_2, \mathbf{u}, \mathbf{0}), \quad (3.16)$$

where the subscripts k have been dropped.

Theorem 3.2.1. For (3.14) or (3.15), where noise has been neglected, the following three statements are equivalent:

1. There exists a function \mathbf{G} such that $\mathbf{F}(\mathbf{X}_2, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_1, \mathbf{u}) = \mathbf{G}(\mathbf{X}_2^{-1}\mathbf{X}_1, \mathbf{u})$.
2. There exists a function \mathbf{G} such that $\mathbf{F}(\mathbf{X}_2, \mathbf{u})\mathbf{F}(\mathbf{X}_1, \mathbf{u})^{-1} = \mathbf{G}(\mathbf{X}_2\mathbf{X}_1^{-1}, \mathbf{u})$.
3. Equation (3.16) is satisfied.

Moreover, for each $\mathbf{u} \in \mathbb{R}^{n_u}$, there exists a matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ such that $\forall \boldsymbol{\xi} \in \mathbb{R}^d$, $\mathbf{G}(\exp(\boldsymbol{\xi}^\wedge), \mathbf{u}) = \exp((\mathbf{A}\boldsymbol{\xi})^\wedge)$.

The proofs for this theorem can be partly found in [20, 21]. A partial proof is presented here. *Proof:* To prove 1 \implies 3, begin with $\mathbf{F}(\mathbf{X}_b, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_a, \mathbf{u}) = \mathbf{G}(\mathbf{X}_b^{-1}\mathbf{X}_a, \mathbf{u})$ and let $\mathbf{X}_a = \mathbf{X}_2$ and $\mathbf{X}_b = \mathbf{1}$. Then,

$$\mathbf{F}(\mathbf{1}, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_2, \mathbf{u}) = \mathbf{G}(\mathbf{X}_2, \mathbf{u}). \quad (3.17)$$

Now, let $\mathbf{X}_b = \mathbf{X}_1$ and $\mathbf{X}_a = \mathbf{X}_1\mathbf{X}_2$ giving

$$\mathbf{F}(\mathbf{X}_1, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) = \mathbf{G}(\mathbf{X}_1^{-1}\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) = \mathbf{G}(\mathbf{X}_2, \mathbf{u}). \quad (3.18)$$

Equating (3.17) and (3.18) yields

$$\begin{aligned} \mathbf{F}(\mathbf{X}_1, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) &= \mathbf{F}(\mathbf{1}, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_2, \mathbf{u}), \\ \mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{F}(\mathbf{1}, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_2, \mathbf{u}), \end{aligned}$$

which is 3.

To show 3 \implies 1, begin with

$$\begin{aligned} \mathbf{F}(\mathbf{X}_a, \mathbf{u})\mathbf{F}(\mathbf{1}, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_b, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_a\mathbf{X}_b, \mathbf{u}), \\ \mathbf{F}(\mathbf{X}_a, \mathbf{u})\mathbf{G}(\mathbf{X}_b, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_a\mathbf{X}_b, \mathbf{u}), \\ \mathbf{G}(\mathbf{X}_b, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_a, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_a\mathbf{X}_b, \mathbf{u}). \end{aligned}$$

Letting $\mathbf{X}_a = \mathbf{X}_2$ and $\mathbf{X}_b = \mathbf{X}_2^{-1}\mathbf{X}_1$, then

$$\mathbf{G}(\mathbf{X}_2^{-1}\mathbf{X}_1) = \mathbf{F}(\mathbf{X}_2, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_1, \mathbf{u}),$$

which is 1.

To prove $2 \implies 3$, similarly begin with $\mathbf{F}(\mathbf{X}_b, \mathbf{u})\mathbf{F}(\mathbf{X}_a, \mathbf{u})^{-1} = \mathbf{G}(\mathbf{X}_b\mathbf{X}_a^{-1}, \mathbf{u})$ and let $\mathbf{X}_b = \mathbf{X}_1$ and $\mathbf{X}_a = \mathbf{1}$. Then,

$$\mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{F}(\mathbf{1}, \mathbf{u})^{-1} = \mathbf{G}(\mathbf{X}_1, \mathbf{u}). \quad (3.19)$$

Now, let $\mathbf{X}_b = \mathbf{X}_1\mathbf{X}_2$ and $\mathbf{X}_a = \mathbf{X}_2$ giving

$$\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u})\mathbf{F}(\mathbf{X}_2, \mathbf{u})^{-1} = \mathbf{G}(\mathbf{X}_1\mathbf{X}_2\mathbf{X}_2^{-1}, \mathbf{u}) = \mathbf{G}(\mathbf{X}_1, \mathbf{u}). \quad (3.20)$$

Equating (3.19) and (3.20) yields

$$\begin{aligned} \mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u})\mathbf{F}(\mathbf{X}_2, \mathbf{u})^{-1} &= \mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{F}(\mathbf{1}, \mathbf{u})^{-1}, \\ \mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{F}(\mathbf{1}, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_2, \mathbf{u}), \end{aligned}$$

which is 3.

To show $3 \implies 2$, begin with

$$\begin{aligned} \mathbf{F}(\mathbf{X}_a, \mathbf{u})\mathbf{F}(\mathbf{1}, \mathbf{u})^{-1}\mathbf{F}(\mathbf{X}_b, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_a\mathbf{X}_b, \mathbf{u}), \\ \mathbf{G}(\mathbf{X}_a, \mathbf{u})\mathbf{F}(\mathbf{X}_b, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_a\mathbf{X}_b, \mathbf{u}), \\ \mathbf{G}(\mathbf{X}_a, \mathbf{u}) &= \mathbf{F}(\mathbf{X}_a\mathbf{X}_b, \mathbf{u})\mathbf{F}(\mathbf{X}_b, \mathbf{u})^{-1}. \end{aligned}$$

Letting $\mathbf{X}_a = \mathbf{X}_1\mathbf{X}_2^{-1}$ and $\mathbf{X}_b = \mathbf{X}_2$, then

$$\mathbf{G}(\mathbf{X}_1\mathbf{X}_2^{-1}) = \mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{F}(\mathbf{X}_2, \mathbf{u})^{-1},$$

which is 1.

The proof that for each $\mathbf{u} \in \mathbb{R}^{n_u}$, there exists a matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ such that $\forall \boldsymbol{\xi} \in \mathbb{R}^d$, $\mathbf{G}(\exp(\boldsymbol{\xi}^\wedge), \mathbf{u}) = \exp((\mathbf{A}\boldsymbol{\xi})^\wedge)$ is out of the scope of this thesis, but can be found in [1]. The input \mathbf{u} is generally absorbed by \mathbf{A} , but it may also vanish in the derivation.

3.3 Invariant Extended Kalman Filtering

This section, along with Sections 3.4 and 3.5, summarizes in more detail some main results from [1] and [2], in which the authors outline how to implement the IEKF. As with any Kalman filter, the states are predicted using the input \mathbf{u} and corrected using some measurement $\mathbf{y}_k \in \mathbb{R}^{n_y}$. Throughout this thesis, $\check{\mathbf{X}}$ and $\hat{\mathbf{X}}$ are used to denote the estimated state after the prediction and correction steps, respectively.

The continuous-time prediction step is

$$\dot{\check{\mathbf{X}}} = \mathbf{F}(\hat{\mathbf{X}}, \mathbf{u}, \mathbf{0}), \quad (3.21)$$

$$\dot{\check{\mathbf{P}}} = \mathbf{A}\hat{\mathbf{P}} + \hat{\mathbf{P}}\mathbf{A}^\top + \mathbf{L}\mathbf{Q}\mathbf{L}^\top. \quad (3.22)$$

The method to find \mathbf{A} and \mathbf{L} is described in Sections 3.4.1 and 3.5.1 for left and right-invariant errors, respectively. In practice, (3.21) and (3.22) must be integrated using some numerical integration method, such as an Euler integration or a Runge-Kutta method. The discrete-time prediction is

$$\check{\mathbf{X}}_k = \mathbf{F}_{k-1}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}), \quad (3.23)$$

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^\top + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^\top. \quad (3.24)$$

The discrete-time Jacobians are derived in Sections 3.4.2 and 3.5.2 for left and right-invariant errors, respectively. The correction step is performed when a measurement \mathbf{y}_k is available. The left and right-invariant measurement models considered in this thesis are

$$\mathbf{y}_k^L = \mathbf{X}_k \mathbf{b}_k + \mathbf{v}_k, \quad (3.25)$$

$$\mathbf{y}_k^R = \mathbf{X}_k^{-1} \mathbf{b}_k + \mathbf{v}_k, \quad (3.26)$$

respectively, where \mathbf{b}_k is some known vector and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. Typically, a first-order Taylor series expansion of the measurement model leads to the measurement model and noise Jacobians \mathbf{H}_k and \mathbf{M}_k . Here an alternative method is used, where the innovation is linearized. The cases for both the left and right-invariant measurement models are covered in Sections 3.4 and 3.5, respectively. The Kalman gain at time t_k is computed using

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{H}_k^\top (\mathbf{H}_k \check{\mathbf{P}}_k \mathbf{H}_k^\top + \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^\top)^{-1}. \quad (3.27)$$

The covariance update is

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k \mathbf{H}_k) \check{\mathbf{P}}_k (\mathbf{1} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^T \mathbf{K}_k^T. \quad (3.28)$$

3.4 Left-Invariant Extended Kalman Filter

The LIEKF is used for a left-invariant measurement model of the form (3.25). Multiple vector measurements may be used, but only one is shown here. The measurement model is left-invariant because, in the absence of noise,

$$\begin{aligned}\mathbf{y}_k &= \mathbf{X}_k \mathbf{b}_k, \\ \mathbf{X}'_k \mathbf{y}_k &= \mathbf{X}'_k (\mathbf{X}_k \mathbf{b}_k), \\ \tilde{\mathbf{y}}_k &= \tilde{\mathbf{X}}_k \mathbf{b}_k,\end{aligned}$$

where \mathbf{X}'_k is arbitrary element of \mathcal{G} . The left-invariant error is given by (3.3). This error is left-invariant because the error is invariant to left multiplication by an element of \mathcal{G} ,

$$\begin{aligned}\delta \mathbf{X}^L &= (\mathbf{X}' \mathbf{X})^{-1} (\mathbf{X}' \hat{\mathbf{X}}), \\ &= \mathbf{X}^{-1} \mathbf{X}'^{-1} \mathbf{X}' \hat{\mathbf{X}} \\ &= \mathbf{X}^{-1} \hat{\mathbf{X}},\end{aligned}$$

where $\mathbf{X}' \in \mathcal{G}$.

3.4.1 Continuous-Time Prediction

The error propagation, found by computing the time-derivative of (3.3), leads to an equation of the form

$$\begin{aligned}
\delta \dot{\mathbf{X}}^L &= \frac{d}{dt} (\mathbf{X}^{-1} \hat{\mathbf{X}}) \\
&= \frac{d}{dt} \mathbf{X}^{-1} \hat{\mathbf{X}} + \mathbf{X}^{-1} \dot{\hat{\mathbf{X}}} \\
&= -\mathbf{X}^{-1} \dot{\mathbf{X}} \mathbf{X}^{-1} \hat{\mathbf{X}} + \mathbf{X}^{-1} \dot{\hat{\mathbf{X}}} \\
&= -\mathbf{X}^{-1} \dot{\mathbf{X}} \delta \mathbf{X}^L + \mathbf{X}^{-1} \dot{\hat{\mathbf{X}}} \\
&= -\mathbf{X}^{-1} (\mathbf{F}(\mathbf{X}, \mathbf{u}) + \mathbf{X} \mathbf{W}) \delta \mathbf{X}^L + \mathbf{X}^{-1} \mathbf{F}(\hat{\mathbf{X}}, \mathbf{u}) \\
&= \mathbf{X}^{-1} (\mathbf{F}(\hat{\mathbf{X}}, \mathbf{u}) - \mathbf{F}(\mathbf{X}, \mathbf{u}) \delta \mathbf{X}^L) - \mathbf{W} \delta \mathbf{X}^L \\
&= \mathbf{X}^{-1} (\mathbf{F}(\mathbf{X} \delta \mathbf{X}^L, \mathbf{u}) - \mathbf{F}(\mathbf{X}, \mathbf{u}) \delta \mathbf{X}^L) - \mathbf{W} \delta \mathbf{X}^L.
\end{aligned} \tag{3.29}$$

Rearranging part of (3.29) using (3.2) where $\mathbf{X}_1 = \mathbf{X}$ and $\mathbf{X}_2 = \delta \mathbf{X}^L$,

$$\begin{aligned}
\delta \dot{\mathbf{X}}^L &= \mathbf{X}^{-1} (\mathbf{X} \mathbf{F}(\delta \mathbf{X}^L, \mathbf{u}) - \mathbf{X} \mathbf{F}(\mathbf{1}, \mathbf{u}) \delta \mathbf{X}^L) - \mathbf{W} \delta \mathbf{X}^L \\
&= \mathbf{F}(\delta \mathbf{X}^L, \mathbf{u}) - \mathbf{F}(\mathbf{1}, \mathbf{u}) \delta \mathbf{X}^L - \mathbf{W} \delta \mathbf{X}^L.
\end{aligned} \tag{3.30}$$

Equation (3.30) is then linearized by letting $\delta \mathbf{X}^L \approx \mathbf{1} + \delta \boldsymbol{\xi}^{L^\wedge}$. The noise is linearized by letting $\mathbf{w} = \bar{\mathbf{w}} + \delta \mathbf{w}$, where $\delta \mathbf{w} = \delta \mathbf{W}^\vee$. As the noise is assumed band-limited and white, $\bar{\mathbf{w}} = \mathbf{0}$ and $\mathbf{w} = \delta \mathbf{w}$. The exact linearization depends on the function $\mathbf{F}(\mathbf{X}, \mathbf{u})$, but, neglecting second order terms, it will have the form

$$\delta \dot{\boldsymbol{\xi}}^L = \mathbf{A} \delta \boldsymbol{\xi}^L + \mathbf{L} \delta \mathbf{w}. \tag{3.31}$$

As guaranteed by Theorem 3.1.1, as (3.1) is group affine, the left-invariant error propagation (3.30), in the absence of noise, is state independent. This remains true even when linearizing. Thus, \mathbf{A} will be state-independent. However, no guarantees are made regarding \mathbf{L} .

3.4.2 Discrete-Time Prediction

Here, the discrete-time model consistent with a left-invariant error, given by (3.14), is used. The left-invariant error at time t_k is

$$\begin{aligned}\delta\check{\mathbf{X}}_k^L &= \mathbf{X}_k^{-1}\check{\mathbf{X}}_k \\ &= [\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}) \exp(\mathbf{w}_{k-1}^\wedge)]^{-1} \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) \\ &= \exp(-\mathbf{w}_{k-1}^\wedge) [\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})]^{-1} \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}).\end{aligned}\quad (3.32)$$

Assuming the system satisfies (3.16), Theorem 3.2.1 states that

$$\begin{aligned}[\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})]^{-1} \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) &= \mathbf{G}(\mathbf{X}_{k-1}^{-1}\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) \\ &= \mathbf{G}(\delta\mathbf{X}_{k-1}^L, \mathbf{u}_{k-1}) \\ &= \mathbf{G}(\exp(\delta\xi_{k-1}^L), \mathbf{u}_{k-1}) \\ &= \exp((\mathbf{A}_{k-1}\delta\xi_{k-1}^L)^\wedge).\end{aligned}\quad (3.33)$$

Substituting (3.33) into (3.32) and letting $\mathbf{w}_{k-1} = \bar{\mathbf{w}}_{k-1} + \delta\mathbf{w}_{k-1}$, where $\bar{\mathbf{w}}_{k-1} = \mathbf{0}$, a first-order approximation of the error is

$$\delta\check{\mathbf{X}}_k^L = \exp(-\delta\mathbf{w}_{k-1}^\wedge) \exp((\mathbf{A}_{k-1}\delta\xi_{k-1}^L)^\wedge), \quad (3.34)$$

$$\exp(\delta\xi_k^L) = \exp((- \delta\mathbf{w}_{k-1} + \mathbf{A}_{k-1}\delta\xi_{k-1}^L)^\wedge), \quad (3.35)$$

$$\delta\check{\xi}_k^L = \mathbf{A}_{k-1}\delta\xi_{k-1}^L - \delta\mathbf{w}_{k-1}, \quad (3.36)$$

where $\mathbf{L}_{k-1} = \mathbf{1}$. This is a first order approximation because the BCH formula was used to go from (3.34) to (3.35).

3.4.3 Discrete-Time Correction

In the seminal IEKF literature [1], the left-invariant update is

$$\hat{\mathbf{X}}_k = \check{\mathbf{X}}_k \exp((\mathbf{K}_k \mathbf{z}_k)^\wedge),$$

which is not consistent with the left-invariant error definition. The authors remedy this by negating the measurement Jacobian \mathbf{H}_k , which in turn negates the gain \mathbf{K}_k . This approach is not used here, and an update consistent with the left-invariant error definition is used.

This left-invariant correction is

$$\hat{\mathbf{X}}_k = \check{\mathbf{X}}_k \exp \left(- (\mathbf{K}_k \mathbf{z}_k)^\wedge \right). \quad (3.37)$$

This form emerges from rearranging the definition of the left-invariant error. The left-invariant error between the predicted and corrected state is $\delta \mathbf{X}_k^L = \hat{\mathbf{X}}_k^{-1} \check{\mathbf{X}}_k$. Rearranging leads to

$$\begin{aligned} \delta \mathbf{X}_k^L &= \hat{\mathbf{X}}_k^{-1} \check{\mathbf{X}}_k, \\ \hat{\mathbf{X}}_k \delta \mathbf{X}_k^L &= \check{\mathbf{X}}_k, \\ \hat{\mathbf{X}}_k &= \check{\mathbf{X}}_k \delta \mathbf{X}_k^{L^{-1}} \\ &= \check{\mathbf{X}}_k \exp \left((\mathbf{K}_k \mathbf{z}_k)^\wedge \right)^{-1} \\ &= \check{\mathbf{X}}_k \exp \left(- (\mathbf{K}_k \mathbf{z}_k)^\wedge \right), \end{aligned}$$

which is (3.37). The innovation \mathbf{z}_k can be written as

$$\mathbf{z}_k = \check{\mathbf{X}}_k^{-1} (\mathbf{y}_k - \check{\mathbf{y}}_k) \quad (3.38)$$

$$\begin{aligned} &= \check{\mathbf{X}}_k^{-1} (\mathbf{X}_k \mathbf{b}_k + \mathbf{v}_k - \check{\mathbf{X}}_k \mathbf{b}_k) \\ &= \delta \check{\mathbf{X}}_k^{L^{-1}} \mathbf{b}_k + \check{\mathbf{X}}_k^{-1} \mathbf{v}_k - \mathbf{b}_k. \end{aligned} \quad (3.39)$$

The innovation is written as (3.38) to ensure that the left-invariant error appears in (3.39). To linearize (3.39), as in the prediction step, let $\delta \check{\mathbf{X}}_k^{L^{-1}} = \mathbf{1} - \delta \check{\xi}_k^{L^\wedge}$, and $\mathbf{v}_k = \bar{\mathbf{v}}_k + \delta \mathbf{v}_k$ with $\bar{\mathbf{v}}_k = \mathbf{0}$,

$$\mathbf{z}_k = \left(\mathbf{1} - \delta \check{\xi}_k^{L^\wedge} \right) \mathbf{b}_k + \check{\mathbf{X}}_k^{-1} \delta \mathbf{v}_k - \mathbf{b}_k. \quad (3.40)$$

$$= -\delta \check{\xi}_k^{L^\wedge} \mathbf{b}_k + \check{\mathbf{X}}_k^{-1} \delta \mathbf{v}_k. \quad (3.41)$$

Lastly, (3.41) must be rearranged such that it can be written as

$$\mathbf{z}_k = \mathbf{H}_k \delta \check{\xi}_k^L + \mathbf{M}_k \delta \mathbf{v}_k, \quad (3.42)$$

where $\mathbf{M}_k = \check{\mathbf{X}}_k^{-1}$. The measurement model Jacobian \mathbf{H}_k will only depend on the known vector \mathbf{b}_k .

3.5 Right-Invariant Extended Kalman Filter

The RIEKF is to be used for a right-invariant measurement model of the form (3.26). Multiple vector measurements may be used, but only one is shown here. The measurement model is right-invariant because, in the absence of noise,

$$\begin{aligned}\mathbf{y}_k &= \mathbf{X}_k^{-1} \mathbf{b}_k, \\ \mathbf{X}'_k^{-1} \mathbf{y}_k &= \mathbf{X}'_k^{-1} (\mathbf{X}_k^{-1} \mathbf{b}_k) \\ &= (\mathbf{X}_k \mathbf{X}'_k)^{-1} \mathbf{b}_k, \\ \tilde{\mathbf{y}}_k &= \tilde{\mathbf{X}}_k^{-1} \mathbf{b}_k.\end{aligned}$$

where \mathbf{X}'_k is an arbitrary element of \mathcal{G} . The right-invariant error is given by (3.4). This error is right-invariant because it is invariant to right multiplication by an element of \mathcal{G} ,

$$\begin{aligned}\delta \mathbf{X}^R &= (\hat{\mathbf{X}} \mathbf{X}') (\mathbf{X} \mathbf{X}')^{-1} \\ &= \hat{\mathbf{X}} \mathbf{X}' \mathbf{X}'^{-1} \mathbf{X}^{-1} \\ &= \hat{\mathbf{X}} \mathbf{X}^{-1}.\end{aligned}$$

where $\mathbf{X}' \in \mathcal{G}$.

3.5.1 Continuous-Time Prediction

The error propagation, found by computing the time-derivative of (3.4), leads to an equation of the form

$$\begin{aligned}\dot{\delta \mathbf{X}}^R &= \frac{d}{dt} (\hat{\mathbf{X}} \mathbf{X}^{-1}) \\ &= \dot{\hat{\mathbf{X}}} \mathbf{X}^{-1} + \hat{\mathbf{X}} \frac{d}{dt} \mathbf{X}^{-1} \\ &= \dot{\hat{\mathbf{X}}} \mathbf{X}^{-1} - \hat{\mathbf{X}} \mathbf{X}^{-1} \dot{\hat{\mathbf{X}}} \mathbf{X}^{-1} \\ &= \mathbf{F}(\hat{\mathbf{X}}, \mathbf{u}) \mathbf{X}^{-1} - \delta \mathbf{X}^R (\mathbf{F}(\mathbf{X}, \mathbf{u}) + \mathbf{X} \mathbf{W}) \mathbf{X}^{-1} \\ &= (\mathbf{F}(\hat{\mathbf{X}}, \mathbf{u}) - \delta \mathbf{X}^R \mathbf{F}(\mathbf{X}, \mathbf{u})) \mathbf{X}^{-1} - \delta \mathbf{X}^R \mathbf{X} \mathbf{W} \mathbf{X}^{-1} \\ &= (\mathbf{F}(\hat{\mathbf{X}}, \mathbf{u}) - \delta \mathbf{X}^R \mathbf{F}(\mathbf{X}, \mathbf{u})) \mathbf{X}^{-1} - \hat{\mathbf{X}} \mathbf{W} \hat{\mathbf{X}}^{-1} \delta \mathbf{X}^R.\end{aligned}\tag{3.43}$$

Rearranging part of (3.43) using (3.2) where $\mathbf{X}_1 = \delta\mathbf{X}^R$ and $\mathbf{X}_2 = \mathbf{X}$,

$$\begin{aligned}\delta\dot{\mathbf{X}}^R &= (\mathbf{F}(\delta\mathbf{X}^R, \mathbf{u})\mathbf{X} - \delta\mathbf{X}^R\mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X})\mathbf{X}^{-1} - \hat{\mathbf{X}}\mathbf{W}\hat{\mathbf{X}}^{-1}\delta\mathbf{X}^R \\ &= \mathbf{F}(\delta\mathbf{X}^R, \mathbf{u}) - \delta\mathbf{X}^R\mathbf{F}(\mathbf{1}, \mathbf{u}) - \hat{\mathbf{X}}\mathbf{W}\hat{\mathbf{X}}^{-1}\delta\mathbf{X}^R.\end{aligned}\quad (3.44)$$

As in the left-invariant case, (3.44) is linearized by letting $\delta\mathbf{X}^R \approx \mathbf{1} + \delta\xi^{R^\wedge}$, $\mathbf{W} = \delta\mathbf{W}$, and neglecting second order terms. The exact linearization depends on the function $\mathbf{F}(\mathbf{X}, \mathbf{u})$, but it will have the form

$$\delta\dot{\xi}^R = \mathbf{A}\delta\xi^R + \mathbf{L}\delta\mathbf{w}, \quad (3.45)$$

where $\delta\mathbf{w} = \delta\mathbf{W}^\vee$.

3.5.2 Discrete-Time Prediction

Here, the discrete-time model consistent with a right-invariant error, given by (3.15), is used. The right-invariant error at time t_k is

$$\begin{aligned}\delta\check{\mathbf{X}}_k^R &= \check{\mathbf{X}}_k\mathbf{X}_k^{-1} \\ &= \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) [\exp(\mathbf{w}_{k-1}^\wedge)\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})]^{-1} \\ &= \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) [\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})]^{-1} \exp(-\mathbf{w}_{k-1}^\wedge)\end{aligned}\quad (3.46)$$

From (3.46), the result of Theorem 3.2.1 can now be used to yield

$$\begin{aligned}\delta\check{\mathbf{X}}_k^R &= \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) [\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})]^{-1} \exp(-\mathbf{w}_{k-1}^\wedge) \\ &= [\mathbf{G}(\mathbf{X}_{k-1}^{-1}\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1})]^{-1} \exp(-\mathbf{w}_{k-1}^\wedge) \\ &= [\mathbf{G}(\delta\mathbf{X}_{k-1}^{R^{-1}}, \mathbf{u}_{k-1})]^{-1} \exp(-\mathbf{w}_{k-1}^\wedge) \\ &= [\mathbf{G}(\exp(-\delta\xi_{k-1}^{R^\wedge}), \mathbf{u}_{k-1})]^{-1} \exp(-\mathbf{w}_{k-1}^\wedge) \\ &= [\exp(-(\mathbf{A}_{k-1}\delta\xi_{k-1}^{R^\wedge})^\wedge)]^{-1} \exp(-\mathbf{w}_{k-1}^\wedge) \\ &= \exp((\mathbf{A}_{k-1}\delta\xi_{k-1}^{R^\wedge})^\wedge) \exp(-\mathbf{w}_{k-1}^\wedge) \\ &= \exp((\mathbf{A}_{k-1}\delta\xi_{k-1}^{R^\wedge} - \mathbf{w}_{k-1})^\wedge).\end{aligned}\quad (3.47)$$

Letting $\delta\check{\mathbf{X}}_k^R = \exp(\delta\check{\boldsymbol{\xi}}_k^{R^\wedge})$, a first-order approximation of (3.47) is

$$\begin{aligned}\exp(\delta\check{\boldsymbol{\xi}}_k^{R^\wedge}) &= \exp((\mathbf{A}_{k-1}\delta\boldsymbol{\xi}_{k-1}^R - \mathbf{w}_{k-1})^\wedge), \\ \delta\check{\boldsymbol{\xi}}_k^R &= \mathbf{A}_{k-1}\delta\boldsymbol{\xi}_{k-1}^R + \mathbf{L}_{k-1}\mathbf{w}_{k-1}.\end{aligned}$$

where $\mathbf{L}_{k-1} = -\mathbf{1}$.

The previous derivation uses a mathematically consistent uncertainty representation when injecting noise into the system. However, this approach does not properly represent the way noise typically enters systems. Noise typically enters in a way much more consistent with a left-invariant error definition, as will be seen in Chapter 4. Therefore, an alternate derivation is presented, where the discrete-time noisy kinematics are given by (3.14). The right-invariant error at time t_k is

$$\begin{aligned}\delta\check{\mathbf{X}}_k^R &= \check{\mathbf{X}}_k\mathbf{X}_k^{-1} \\ &= \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) [\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}) \exp(\mathbf{w}_{k-1}^\wedge)]^{-1} \\ &= \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) \exp(-\mathbf{w}_{k-1}^\wedge) [\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})]^{-1} \\ &= \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) \exp(-\mathbf{w}_{k-1}^\wedge) [\mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1})]^{-1} \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) [\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})]^{-1}.\end{aligned}\tag{3.48}$$

Given that $\mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) \in \mathcal{G}$, and using the adjoint identity (2.1), (3.48) can be written as

$$\delta\check{\mathbf{X}}_k^R = \exp\left[-\left(\text{Ad}\left(\mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1})\right)\mathbf{w}_{k-1}\right)^\wedge\right]\mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}) [\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})]^{-1}.\tag{3.49}$$

For ease of notation, let $\text{Ad}(\hat{\mathbf{F}}) = \text{Ad}(\mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}))$. From (3.49), the result of Theo-

rem 3.2.1 can now be used to yield

$$\begin{aligned}
\delta \check{\mathbf{X}}_k^R &= \exp \left(- \left(\text{Ad}(\hat{\mathbf{F}}) \mathbf{w}_{k-1} \right)^\wedge \right) \mathbf{F} \left(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1} \right) [\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1})]^{-1} \\
&= \exp \left(- \left(\text{Ad}(\hat{\mathbf{F}}) \mathbf{w}_{k-1} \right)^\wedge \right) \left[\mathbf{G} \left(\mathbf{X}_{k-1}^{-1} \hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1} \right) \right]^{-1} \\
&= \exp \left(- \left(\text{Ad}(\hat{\mathbf{F}}) \mathbf{w}_{k-1} \right)^\wedge \right) \left[\mathbf{G} \left(\delta \mathbf{X}_{k-1}^R{}^{-1}, \mathbf{u}_{k-1} \right) \right]^{-1} \\
&= \exp \left(- \left(\text{Ad}(\hat{\mathbf{F}}) \mathbf{w}_{k-1} \right)^\wedge \right) \left[\mathbf{G} \left(\exp \left(- \delta \xi_{k-1}^R{}^\wedge \right), \mathbf{u}_{k-1} \right) \right]^{-1} \\
&= \exp \left(- \left(\text{Ad}(\hat{\mathbf{F}}) \mathbf{w}_{k-1} \right)^\wedge \right) [\exp \left(- (\mathbf{A}_{k-1} \delta \xi_{k-1}^R)^\wedge \right)]^{-1} \\
&= \exp \left(- \left(\text{Ad}(\hat{\mathbf{F}}) \mathbf{w}_{k-1} \right)^\wedge \right) \exp \left((\mathbf{A}_{k-1} \delta \xi_{k-1}^R)^\wedge \right). \tag{3.50}
\end{aligned}$$

Letting $\delta \check{\mathbf{X}}_k^R = \exp \left(\delta \check{\boldsymbol{\xi}}_k^R{}^\wedge \right)$ and using the BCH formula, a first-order approximation of (3.50) is

$$\begin{aligned}
\exp \left(\delta \check{\boldsymbol{\xi}}_k^R{}^\wedge \right) &= \exp \left(- \left(\text{Ad}(\hat{\mathbf{F}}) \mathbf{w}_{k-1} \right)^\wedge \right) \exp \left((\mathbf{A}_{k-1} \delta \xi_{k-1}^R)^\wedge \right) \\
&\approx \exp \left(\left(\mathbf{A}_{k-1} \delta \xi_{k-1}^R - \text{Ad}(\hat{\mathbf{F}}) \mathbf{w}_{k-1} \right)^\wedge \right), \\
\delta \check{\boldsymbol{\xi}}_k^R &= \mathbf{A}_{k-1} \delta \xi_{k-1}^R + \mathbf{L}_{k-1} \mathbf{w}_{k-1}.
\end{aligned}$$

where $\mathbf{L}_{k-1} = -\text{Ad}(\hat{\mathbf{F}})$.

3.5.3 Discrete-Time Correction

The right-invariant update in the literature is

$$\hat{\mathbf{X}}_k = \check{\mathbf{X}}_k \exp \left((\mathbf{K}_k \mathbf{z}_k)^\wedge \right),$$

which is not consistent with the right-invariant error definition. An update consistent with the right-invariant error definition is used. This right-invariant correction is

$$\hat{\mathbf{X}}_k = \exp \left(- (\mathbf{K}_k \mathbf{z}_k)^\wedge \right) \check{\mathbf{X}}_k. \tag{3.51}$$

This form emerges from rearranging the definition of the right-invariant error. The right-invariant error between the predicted and corrected state is $\delta \mathbf{X}_k^R = \check{\mathbf{X}}_k \hat{\mathbf{X}}_k^{-1}$. Rearranging

leads to

$$\begin{aligned}
\delta \mathbf{X}_k^R &= \check{\mathbf{X}}_k \hat{\mathbf{X}}_k^{-1}, \\
\delta \mathbf{X}_k^R \dot{\check{\mathbf{X}}}_k &= \check{\mathbf{X}}_k, \\
\dot{\check{\mathbf{X}}}_k &= \delta \mathbf{X}_k^{R^{-1}} \check{\mathbf{X}}_k \\
&= \exp((\mathbf{K}_k \mathbf{z}_k)^\wedge)^{-1} \check{\mathbf{X}}_k \\
&= \exp(-(\mathbf{K}_k \mathbf{z}_k)^\wedge) \check{\mathbf{X}}_k,
\end{aligned}$$

which is (3.51).

The innovation \mathbf{z}_k can be written as

$$\mathbf{z}_k = \check{\mathbf{X}}_k (\mathbf{y}_k - \check{\mathbf{y}}_k) \quad (3.52)$$

$$\begin{aligned}
&= \check{\mathbf{X}}_k (\mathbf{X}_k^{-1} \mathbf{b}_k + \mathbf{v}_k - \check{\mathbf{X}}_k^{-1} \mathbf{b}_k) \\
&= \delta \check{\mathbf{X}}_k^R \mathbf{b}_k - \mathbf{b}_k + \check{\mathbf{X}}_k \mathbf{v}_k. \tag{3.53}
\end{aligned}$$

The innovation is written as (3.52) to ensure that the right-invariant error appears in (3.53). To linearize (3.53), let $\delta \check{\mathbf{X}}_k^R \approx \mathbf{1} + \delta \check{\xi}_k^{R^\wedge}$ and $\mathbf{v}_k = \bar{\mathbf{v}}_k + \delta \mathbf{v}_k$ with $\bar{\mathbf{v}}_k = \mathbf{0}$. Neglecting second order terms, (3.53) is then

$$\begin{aligned}
\mathbf{z}_k &\approx \left(\mathbf{1} + \delta \check{\xi}_k^{R^\wedge} \right) \mathbf{b}_k - \mathbf{b}_k + \check{\mathbf{X}}_k \delta \mathbf{v}_k \\
&= \delta \check{\xi}_k^{R^\wedge} \mathbf{b}_k + \check{\mathbf{X}}_k \delta \mathbf{v}_k, \tag{3.54}
\end{aligned}$$

Lastly, (3.54) must be rearranged such that it can be written as

$$\mathbf{z}_k = \mathbf{H}_k \delta \check{\xi}_k^R + \mathbf{M}_k \delta \mathbf{v}_k, \tag{3.55}$$

where $\mathbf{M}_k = \check{\mathbf{X}}_k$. Once again, the measurement model Jacobian \mathbf{H}_k is state independent, as it only depends on the known vector \mathbf{b}_k .

3.6 Summary

Table 3.1 offers a summary of some key equations that differ between the standard (extended) Kalman filter, MEKF, LIEKF, and RIEKF.

The key driving force behind the forms of the LIEKF and RIEKF stem from the form of the measurement model. The measurement model dictates which IEKF is used. In particular,

	Error Definition	Innovation	State Correction
KF/EKF	$\delta \mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}$	$\mathbf{z}_k = \mathbf{y} - \check{\mathbf{y}}_k$	$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k \mathbf{z}_k$
MEKF	$\delta \mathbf{X} = \mathbf{X} \hat{\mathbf{X}}^{-1}$	$\mathbf{z}_k = \mathbf{y} - \check{\mathbf{y}}_k$	$\hat{\mathbf{X}}_k = \exp((\mathbf{K}_k \mathbf{z}_k)^\wedge) \check{\mathbf{X}}_k$
LIEKF	$\delta \mathbf{X}^L = \mathbf{X}^{-1} \hat{\mathbf{X}}$	$\mathbf{z}_k = \check{\mathbf{X}}_k^{-1} (\mathbf{y}_k - \check{\mathbf{y}}_k)$	$\hat{\mathbf{X}}_k = \check{\mathbf{X}}_k \exp(-(\mathbf{K}_k \mathbf{z}_k)^\wedge)$
RIEKF	$\delta \mathbf{X}^R = \hat{\mathbf{X}} \mathbf{X}^{-1}$	$\mathbf{z}_k = \check{\mathbf{X}}_k (\mathbf{y}_k - \check{\mathbf{y}}_k)$	$\hat{\mathbf{X}}_k = \exp(-(\mathbf{K}_k \mathbf{z}_k)^\wedge) \check{\mathbf{X}}_k$

Table 3.1: Summary of the key KF/EKF, MEKF, LIEKF, RIEKF equations.

if \mathbf{y}_k is left invariant, then a left-invariant innovation, error, and state correction should be used. On the other hand, if \mathbf{y}_k is right invariant, then a right-invariant innovation, error, and state correction should be used. Using the correct invariant formulation will lead to matrices \mathbf{A} , or \mathbf{A}_{k-1} , and \mathbf{H}_k being state-estimate independent.

Note, although \mathbf{A} , or \mathbf{A}_{k-1} , and \mathbf{H}_k will be state independent, the invariant framework does not assure \mathbf{L} , or \mathbf{L}_{k-1} , and \mathbf{M}_k will be state independent.

Chapter 4

Practical Implications of the IEKF

Several applied examples of the IEKF can be found in the literature. Barrau and Bonnabel provide examples of the IEKF applied to two problems in [1]. In [22], three different applications of the IEKF are presented. However, no results were shown. A case study, clearly demonstrating the geometric advantages of the IEKF is presented in [23]. An IEKF applied to scan matching-aided navigation is shown in [24]. A quaternion-based IEKF is applied to a spacecraft attitude estimation problem in [25]. An IEKF is used as a state estimator for a bipedal robot in [26], which includes bias estimation. The IEKF has also been used to improve on the classic EKF-SLAM solution [27]. The popular multi-state constrained Kalman filter [28] is formulated in an invariant framework, including bias estimation, in [29]. However, some questions remained unanswered. In this chapter, a sample problem is used to illustrate the performance of the IEKF in both simulation and using experimental data. A 3D problem, where the state is an element of $SE(3)$, is considered. The goal of this chapter is to answer four questions left at least partially unanswered in the literature, in order to better explain the practical implications of the IEKF. These questions are as follows.

1. Given a left-invariant measurement model, is it more advantageous to use a LIEKF than an MEKF, and given a right-invariant measurement model, is it more advantageous to use a RIEKF than an MEKF?
2. What is the effect of process and measurement noise? How quickly do the convergence properties of the IEKF claimed in [1, 2] break down?
3. How can the IEKF be used in situations when the measurement model isn't left or right-invariant?
4. How can bias estimation be performed in an invariant framework?

To answer the first two questions, a sample problem is presented, followed by some simulation

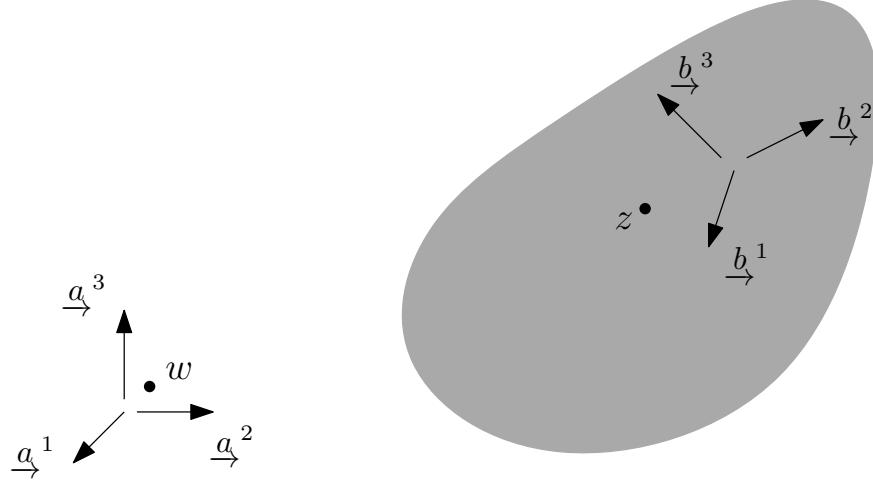


Figure 4.1: Problem setup.

results. The sample problem is then modified in Section 4.2 to fit a realistic measurement model. This new sample problem is then tested on simulated and experimental data. Lastly, the sample problem is once again modified to incorporate bias, followed by simulation results. This will build on the work presented in , [9], [26] and [29].

4.1 Sample Problem: $SE(3)$

4.1.1 System Modelling

Consider a body free to translate and rotate in 3D space. Let \mathcal{F}_a be a global frame and \mathcal{F}_b be a frame which rotates with the body. Point z moves with the body, while point w is a reference point. The problem setup is illustrated in Figure 4.1. The estimated states are \mathbf{r}_a^{zw} and \mathbf{C}_{ab} . The pose can be represented as an element of $SE(3)$ as

$$\mathbf{T}_{ab} = \begin{bmatrix} \mathbf{C}_{ab} & \mathbf{r}_a^{zw} \\ \mathbf{0} & 1 \end{bmatrix}.$$

The kinematics of this problem are

$$\begin{aligned} \dot{\mathbf{C}}_{ab} &= \mathbf{C}_{ab} \boldsymbol{\omega}_b^{ba} \times, \\ \dot{\mathbf{r}}_a^{zw} &= \mathbf{v}_a^{zw/a}. \end{aligned}$$

In matrix form, the kinematics are

$$\dot{\mathbf{T}}_{ab} = \mathbf{T}_{ab} \boldsymbol{\varpi}_b^\wedge,$$

where $\boldsymbol{\varpi}_b = \left[\boldsymbol{\omega}_b^{ba\top} \mathbf{v}_b^{zw/a\top} \right]^\top$. The body is equipped with a rate gyro which measures

$$\mathbf{u}_b^1 = \boldsymbol{\omega}_b^{ba} - \mathbf{w}_b^1,$$

where \mathbf{w}_b^1 is band-limited white noise. When discretized, $\mathbf{w}_{b_k}^1 \sim \mathcal{N}(0, \mathbf{Q}_k^1)$. The body is also equipped with a velocity sensor that measures

$$\mathbf{u}_b^2 = \mathbf{v}_b^{zw/a} - \mathbf{w}_b^2,$$

where \mathbf{w}_b^2 is band-limited white noise. When discretized, $\mathbf{w}_{b_k}^2 \sim \mathcal{N}(0, \mathbf{Q}_k^2)$. Rearranging and substituting into the kinematics leads to

$$\begin{aligned}\dot{\mathbf{C}}_{ab} &= \mathbf{C}_{ab} (\mathbf{u}_b^1 + \mathbf{w}_b^1)^\times, \\ \dot{\mathbf{r}}_a^{zw} &= \mathbf{C}_{ab} (\mathbf{u}_b^2 + \mathbf{w}_b^2).\end{aligned}$$

Incorporating the measurement model, the kinematics in matrix form are

$$\dot{\mathbf{T}}_{ab} = \mathbf{T}_{ab} (\mathbf{u}_b + \mathbf{w}_b)^\wedge,$$

where $\mathbf{u}_b = \left[\mathbf{u}_b^{1\top} \mathbf{u}_b^{2\top} \right]^\top$ and $\mathbf{w}_b = \left[\mathbf{w}_b^{1\top} \mathbf{w}_b^{2\top} \right]^\top$. Throughout this chapter, the subscripts on \mathbf{T}_{ab} are dropped to be concise.

4.1.2 Left-Invariant Extended Kalman Filter Derivation

In order to derive a LIEKF, the exteroceptive measurement model must be left invariant. In this case, appropriate measurements are position measurements, which could be from a GPS receiver, for example. The discrete-time measurement model is

$$\begin{bmatrix} \mathbf{y}_{a_k} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_a^{z_k w} + \mathbf{v}_{a_k} \\ 1 \end{bmatrix} = \mathbf{T}_k \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{a_k} \\ 0 \end{bmatrix}, \quad (4.1)$$

where $\mathbf{v}_{a_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$.

The left-invariant error $\delta \mathbf{T} = \mathbf{T}^{-1} \hat{\mathbf{T}}$ is used because the measurement model is left invariant.

The error propagation is

$$\begin{aligned}
\delta \dot{\mathbf{T}} &= \dot{\mathbf{T}}^{-1} \hat{\mathbf{T}} + \mathbf{T}^{-1} \dot{\hat{\mathbf{T}}} \\
&= -\mathbf{T}^{-1} \dot{\mathbf{T}} \mathbf{T}^{-1} \hat{\mathbf{T}} + \mathbf{T}^{-1} \dot{\mathbf{T}} \mathbf{u}_b^\wedge \\
&= -\mathbf{T}^{-1} \mathbf{T} (\mathbf{u}_b + \mathbf{w}_b)^\wedge \mathbf{T}^{-1} \hat{\mathbf{T}} + \delta \mathbf{T} \mathbf{u}_b^\wedge \\
&= -(\mathbf{u}_b + \mathbf{w}_b)^\wedge \delta \mathbf{T} + \delta \mathbf{T} \mathbf{u}_b^\wedge.
\end{aligned} \tag{4.2}$$

To linearize (4.2), let $\delta \mathbf{T} \approx \mathbf{1} + \delta \boldsymbol{\xi}^\wedge$, $\delta \mathbf{T}^{-1} \approx \mathbf{1} - \delta \boldsymbol{\xi}^\wedge$, and $\mathbf{w}_b = \delta \mathbf{w}_b$. Neglecting second order terms, (4.2) is then approximated as

$$\begin{aligned}
\frac{d}{dt} (\mathbf{1} + \delta \boldsymbol{\xi}^\wedge) &= (\mathbf{1} + \delta \boldsymbol{\xi}^\wedge) \mathbf{u}_b^\wedge - (\mathbf{u}_b + \delta \mathbf{w}_b)^\wedge (\mathbf{1} + \delta \boldsymbol{\xi}^\wedge), \\
\delta \dot{\boldsymbol{\xi}}^\wedge &= \mathbf{u}_b^\wedge + \delta \boldsymbol{\xi}^\wedge \mathbf{u}_b^\wedge - (\mathbf{u}_b + \delta \mathbf{w}_b)^\wedge - (\mathbf{u}_b + \delta \mathbf{w}_b)^\wedge \delta \boldsymbol{\xi}^\wedge \\
&= \delta \boldsymbol{\xi}^\wedge \mathbf{u}_b^\wedge - \mathbf{u}_b^\wedge \delta \boldsymbol{\xi}^\wedge - \delta \mathbf{w}_b^\wedge.
\end{aligned} \tag{4.3}$$

Using the identity (2.2), (4.3) is rewritten as

$$\delta \dot{\boldsymbol{\xi}}^\wedge = (-\text{ad}(\mathbf{u}_b) \delta \boldsymbol{\xi})^\wedge - \delta \mathbf{w}_b^\wedge,$$

which in turn can be written as

$$\delta \dot{\boldsymbol{\xi}} = -\text{ad}(\mathbf{u}_b) \delta \boldsymbol{\xi} - \delta \mathbf{w}_b.$$

Therefore, $\mathbf{A} = -\text{ad}(\mathbf{u}_b)$ and $\mathbf{L} = -\mathbf{1}$. Notice that \mathbf{A} is only a function of the measurement \mathbf{u}_b .

When a position measurement \mathbf{y}_{a_k} is available, the state estimate is corrected. This correction has the form

$$\hat{\mathbf{T}}_k = \check{\mathbf{T}}_k \exp(-(\mathbf{K}_k \mathbf{z}_k)^\wedge).$$

The innovation is

$$\begin{aligned}
\begin{bmatrix} \mathbf{z}_k \\ 0 \end{bmatrix} &= \check{\mathbf{T}}_k^{-1} \left(\begin{bmatrix} \mathbf{y}_{a_k} \\ 1 \end{bmatrix} - \begin{bmatrix} \check{\mathbf{y}}_{a_k} \\ 1 \end{bmatrix} \right) \\
&= \check{\mathbf{T}}_k^{-1} \left(\mathbf{T}_k \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{a_k} \\ 0 \end{bmatrix} - \check{\mathbf{T}}_k \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \right) \\
&= \delta \check{\mathbf{T}}_k^{-1} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} + \check{\mathbf{T}}_k^{-1} \begin{bmatrix} \mathbf{v}_{a_k} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}.
\end{aligned} \tag{4.4}$$

Linearizing (4.4) by letting $\delta\check{\mathbf{T}}_k^{-1} \approx \mathbf{1} - \delta\xi_k^\wedge$ and $\mathbf{v}_{a_k} = \delta\mathbf{v}_{a_k}$,

$$\begin{aligned}
\begin{bmatrix} \mathbf{z}_k \\ 0 \end{bmatrix} &\approx (\mathbf{1} - \delta\xi_k^\wedge) \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} + \check{\mathbf{T}}_k^{-1} \begin{bmatrix} \delta\mathbf{v}_{a_k} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \\
&= -\delta\xi_k^\wedge \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} + \check{\mathbf{T}}_k^{-1} \begin{bmatrix} \delta\mathbf{v}_{a_k} \\ 0 \end{bmatrix} \\
&= - \begin{bmatrix} \delta\xi_k^{\phi^\times} & \delta\xi_k^r \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} + \check{\mathbf{T}}_k^{-1} \begin{bmatrix} \delta\mathbf{v}_{a_k} \\ 0 \end{bmatrix} \\
&= \tilde{\mathbf{H}}\delta\xi_k + \check{\mathbf{T}}_k^{-1} \begin{bmatrix} \delta\mathbf{v}_{a_k} \\ 0 \end{bmatrix}, \tag{4.5}
\end{aligned}$$

where

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{0} & -\mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Noting that the bottom row of (4.5) is only zeros, it can be written

$$\mathbf{z}_k = \mathbf{H}\delta\xi_k + \mathbf{M}_k\delta\mathbf{v}_{a_k}$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & -\mathbf{1} \end{bmatrix}$$

and $\mathbf{M}_k = \check{\mathbf{C}}_{ab_k}^\top$. Notice that \mathbf{H} is constant.

4.1.3 Right-Invariant Extended Kalman Filter Derivation

In order to derive a RIEKF, the exteroceptive measurement model must be right invariant. In this case, appropriate measurements are landmark measurements resolved in (i.e. observed in) the body frame, which could be from a LIDAR or camera, for example. In reality, the measurement model depends on the sensor. However, for the sake of simplicity, the exteroceptive measurements are modelled as relative landmark locations. This ensures the measurement model is actually right-invariant. Let point p^i be associated with the i^{th} landmark. The position of the i^{th} landmark relative to point w resolved in $\underline{\mathcal{F}}_a$ is $\mathbf{r}_a^{p_i w}$. The landmark sensor is assumed to measure $\mathbf{r}_b^{p_i z}$. Figure 4.2 displays the geometry of the problem. The discrete-time measurement model is then

$$\begin{bmatrix} \mathbf{y}_{b_k}^i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{ab_k}^\top (\mathbf{r}_a^{p_i w} - \mathbf{r}_a^{z_k w}) + \mathbf{v}_{b_k}^i \\ 1 \end{bmatrix} = \mathbf{T}_k^{-1} \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{b_k}^i \\ 0 \end{bmatrix}, \tag{4.6}$$

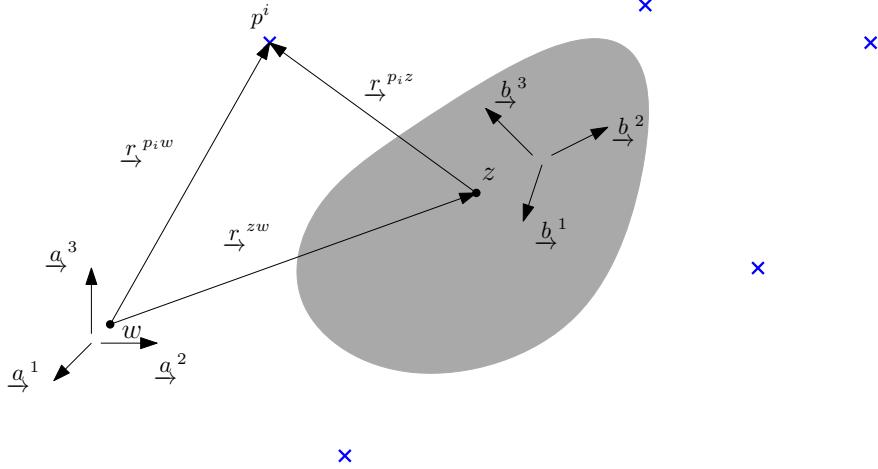


Figure 4.2: Problem setup, where the blue crosses are the observed landmarks.

where $\mathbf{v}_{b_k}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k^i)$, $i = 1, \dots, m$, where m is the number of landmarks.

The right-invariant error $\delta \mathbf{T} = \hat{\mathbf{T}} \mathbf{T}^{-1}$ is used. The error propagation is

$$\begin{aligned}
\delta \dot{\mathbf{T}} &= \dot{\hat{\mathbf{T}}} \mathbf{T}^{-1} + \hat{\mathbf{T}} \dot{\mathbf{T}}^{-1} \\
&= \hat{\mathbf{T}} \mathbf{u}_b^\wedge \mathbf{T}^{-1} - \hat{\mathbf{T}} \mathbf{T}^{-1} \dot{\hat{\mathbf{T}}} \mathbf{T}^{-1} \\
&= \hat{\mathbf{T}} \mathbf{u}_b^\wedge \mathbf{T}^{-1} - \hat{\mathbf{T}} \mathbf{T}^{-1} \mathbf{T} (\mathbf{u}_b + \mathbf{w}_b)^\wedge \mathbf{T}^{-1} \\
&= \hat{\mathbf{T}} (\mathbf{u}_b - \mathbf{u}_b - \mathbf{w}_b)^\wedge \mathbf{T}^{-1} \\
&= -\hat{\mathbf{T}} \mathbf{w}_b^\wedge \hat{\mathbf{T}}^{-1} \delta \mathbf{T} \\
&= -\left(\text{Ad}(\hat{\mathbf{T}}) \mathbf{w}_b \right)^\wedge \delta \mathbf{T}.
\end{aligned} \tag{4.7}$$

To linearize (4.7), let $\delta \mathbf{T} \approx \mathbf{1} + \delta \boldsymbol{\xi}^\wedge$ and $\mathbf{w}_b = \delta \mathbf{w}_b$. Neglecting second order terms, (4.7) is then

$$\begin{aligned}
\frac{d}{dt} (\mathbf{1} + \delta \boldsymbol{\xi}^\wedge) &= -\left(\text{Ad}(\hat{\mathbf{T}}) \delta \mathbf{w}_b \right)^\wedge (\mathbf{1} + \delta \boldsymbol{\xi}^\wedge), \\
\delta \dot{\boldsymbol{\xi}}^\wedge &= -\left(\text{Ad}(\hat{\mathbf{T}}) \delta \mathbf{w}_b \right)^\wedge, \\
\delta \dot{\boldsymbol{\xi}} &= -\text{Ad}(\hat{\mathbf{T}}) \delta \mathbf{w}_b.
\end{aligned} \tag{4.8}$$

Therefore, $\mathbf{A} = \mathbf{0}$ and $\mathbf{L} = -\text{Ad}(\hat{\mathbf{T}})$. Notice that \mathbf{A} is constant.

When a relative landmark measurement $\mathbf{y}_{b_k}^i$ is available, the state estimate is corrected. This correction has the form

$$\hat{\mathbf{T}}_k = \exp(-(\mathbf{K}_k \mathbf{z}_k)^\wedge) \check{\mathbf{T}}_k.$$

Edited

2021/2/17

The innovation associated with the i^{th} landmark is

$$\begin{aligned}
\begin{bmatrix} \mathbf{z}_k^i \\ 0 \end{bmatrix} &= \check{\mathbf{T}}_k \left(\begin{bmatrix} \mathbf{y}_{b_k}^i \\ 1 \end{bmatrix} - \begin{bmatrix} \check{\mathbf{y}}_{b_k}^i \\ 1 \end{bmatrix} \right) \\
&= \check{\mathbf{T}}_k \left(\mathbf{T}_k^{-1} \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{b_k}^i \\ 0 \end{bmatrix} - \check{\mathbf{T}}_k^{-1} \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix} \right) \\
&= \delta \check{\mathbf{T}}_k \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix} + \check{\mathbf{T}}_k \begin{bmatrix} \mathbf{v}_{b_k}^i \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix}.
\end{aligned} \tag{4.9}$$

Linearizing (4.9) by letting $\delta \check{\mathbf{T}}_k \approx \mathbf{1} + \delta \check{\boldsymbol{\xi}}_k^\wedge$ and $\mathbf{v}_{b_k} = \delta \mathbf{v}_{b_k}$,

$$\begin{aligned}
\begin{bmatrix} \mathbf{z}_k^i \\ 0 \end{bmatrix} &\approx (\mathbf{1} + \delta \check{\boldsymbol{\xi}}_k^\wedge) \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix} + \check{\mathbf{T}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix} \\
&= \delta \check{\boldsymbol{\xi}}_k^\wedge \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix} + \check{\mathbf{T}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \delta \check{\boldsymbol{\xi}}_k^\phi \times & \delta \check{\boldsymbol{\xi}}_k^r \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix} + \check{\mathbf{T}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \delta \check{\boldsymbol{\xi}}_k^\phi \times \mathbf{r}_a^{p_i w} + \delta \check{\boldsymbol{\xi}}_k^r \\ \mathbf{0} \end{bmatrix} + \check{\mathbf{T}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} -\mathbf{r}_a^{p_i w} \times \delta \check{\boldsymbol{\xi}}_k^\phi + \delta \check{\boldsymbol{\xi}}_k^r \\ \mathbf{0} \end{bmatrix} + \check{\mathbf{T}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ 0 \end{bmatrix} \\
&= \mathbf{H}^i \delta \check{\boldsymbol{\xi}}_k + \check{\mathbf{T}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ 0 \end{bmatrix},
\end{aligned} \tag{4.10}$$

where

$$\tilde{\mathbf{H}} = \begin{bmatrix} -\mathbf{r}_a^{p_i w} \times & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Noting that the bottom row of (4.10) is full of zeros, it can be written

$$\mathbf{z}_k^i = \mathbf{H}^i \delta \check{\boldsymbol{\xi}}_k + \mathbf{M}_k \delta \mathbf{v}_{b_k}^i$$

where

$$\mathbf{H}^i = \begin{bmatrix} -\mathbf{r}_a^{p_i w} \times & \mathbf{1} \end{bmatrix}$$

and $\mathbf{M}_k = \check{\mathbf{C}}_{ab_k}^T$. For the m landmarks,

$$\mathbf{H} = \underset{i=1,\dots,m}{\text{row}} \left(\begin{bmatrix} -\mathbf{r}_a^{p_i w \times} & \mathbf{1} \end{bmatrix} \right) \quad (4.11)$$

and $\mathbf{M}_k = \text{diag}(\check{\mathbf{C}}_{ab_1}^T, \dots, \check{\mathbf{C}}_{ab_m}^T)$. Once again, notice that \mathbf{H} is constant, because $\mathbf{r}_a^{p_i w}, i = 1, \dots, m$ are constant and known a priori.

4.1.4 MEKF Solutions

Throughout this section, the results are compared to that of a standard multiplicative extended Kalman filter (MEKF). A MEKF is a variant of the EKF better suited for attitude estimation. The Jacobians for the MEKF are presented here, but no derivation is given. The errors are defined as $\delta\mathbf{C} = \bar{\mathbf{C}}_{ab}^T \mathbf{C}_{ab}$ and $\delta\mathbf{r} = \mathbf{r}_a^{zw} - \bar{\mathbf{r}}_a^{zw}$. Two different MEKFs must be used for the two different measurement models. The first MEKF, which uses the left-invariant measurements (4.1), is referred to as MEKF-L. The MEKF using the right-invariant measurements (4.6) is referred to as MEKF-R. The process model Jacobians for both are

$$\mathbf{A} = \begin{bmatrix} -\mathbf{u}_b^{1 \times} & \mathbf{0} \\ -\hat{\mathbf{C}}_{ab} \mathbf{u}_b^{2 \times} & \mathbf{0} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{C}}_{ab} \end{bmatrix}.$$

Note that \mathbf{A} depends on the attitude estimate, which was not the case for the IEKFs. For the MEKF-L, the measurement model Jacobians are

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix}, \quad \mathbf{M} = \mathbf{1}.$$

For the MEKF-R, the measurement model Jacobians are

$$\mathbf{H}_k^i = \begin{bmatrix} \left(\check{\mathbf{C}}_{ab_k}^T \left(\mathbf{r}_a^{p_i w} - \check{\mathbf{r}}_a^{z_k^i w} \right) \right)^{\times} & -\check{\mathbf{C}}_{ab_k}^T \end{bmatrix}, \quad \mathbf{M} = \mathbf{1}.$$

For the m landmarks,

$$\mathbf{H}_k = \underset{i=1,\dots,m}{\text{row}} \left(\begin{bmatrix} \left(\check{\mathbf{C}}_{ab_k}^T \left(\mathbf{r}_a^{p_i w} - \check{\mathbf{r}}_a^{z_k^i w} \right) \right)^{\times} & -\check{\mathbf{C}}_{ab_k}^T \end{bmatrix} \right) \quad (4.12)$$

and $\mathbf{M} = \mathbf{1}$. The MEKF-R's measurement model Jacobian \mathbf{H}_k depends on the state estimate, which was not the case for the RIEKF.

4.1.5 Simulation Results

In this section, simulations are performed to compare the four Kalman filters shown in Sections 4.1.2, 4.1.3, and 4.1.4, namely the LIEKF, RIEKF, MEKF-L and MEKF-R. The first question raised in the introduction of this chapter asked whether it was advantageous to use an invariant filter over its standard multiplicative counterpart. The second question concerns the effect of noise on the theoretical convergence properties of the IEKF. The two questions are simultaneously addressed in this section.

4.1.5.1 Simulation Parameters

In one set of trials, the MEKF-L and LIEKF are compared. In another, the RIEKF and MEKF-R are compared. Certain parameters are kept constant throughout the simulation. The trajectory used is shown in Figure 4.3. The landmarks used in the RIEKF and MEKF-R are also displayed. It is assumed that, at any given time, all the landmarks are visible. The rate gyro and velocity sensor operate at 100 Hz, while the corrections occur at 10 Hz. The trajectory is parametrized such that $\|\mathbf{v}_b^{zw/a}\|$ is constant at 1 m/s. The resulting trajectory takes slightly over 30 s to execute, which is generally enough time for the transients associated with the convergence of the filters to dissipate. The attitude is constrained to have constant roll. To simplify the situation, the sensor noise is assumed isotropic. That is, $\mathbf{Q}_k^1 = \sigma_k^{1^2} \mathbf{1}$, $\mathbf{Q}_k^2 = \sigma_k^{2^2} \mathbf{1}$, and $\mathbf{R}_k^i = \sigma_k^{R^2} \mathbf{1}$. Doing this makes the fact \mathbf{M}_k may depend on $\check{\mathbf{C}}_{ab_k}$ irrelevant because $\mathbf{M}_k^\top \mathbf{R}_k \mathbf{M}_k = \mathbf{R}_k$ thereby making the computation of the Kalman gain independent of \mathbf{M}_k . In many applications, the covariance matrices are seen as tuning parameters. Here, no tuning is done. Tuning may greatly effect the results, but in simulation, keeping the theoretical values is more valuable because it demonstrates how the filters behave “out of the box” without extensive tuning.

4.1.5.2 Initialization

To run appropriate Monte Carlo trials, a discussion of the initial error is necessary. When testing any EKF variant using Monte Carlo trials, the initial error $\delta\mathbf{x}_0$ is drawn from a normal distribution with zero mean and covariance \mathbf{P}_0 . That is, $\delta\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0)$. This initial error, along with the truth data, is then used to initialize the filters. For the MEKF-R and MEKF-L, this means

$$\begin{aligned}\hat{\mathbf{r}}_a^{z_0 w} &= \mathbf{r}_a^{z_0 w} - \delta\mathbf{r}_0, \\ \hat{\mathbf{C}}_{ab_0} &= \mathbf{C}_{ab_0} \delta\mathbf{C}_0^\top.\end{aligned}$$

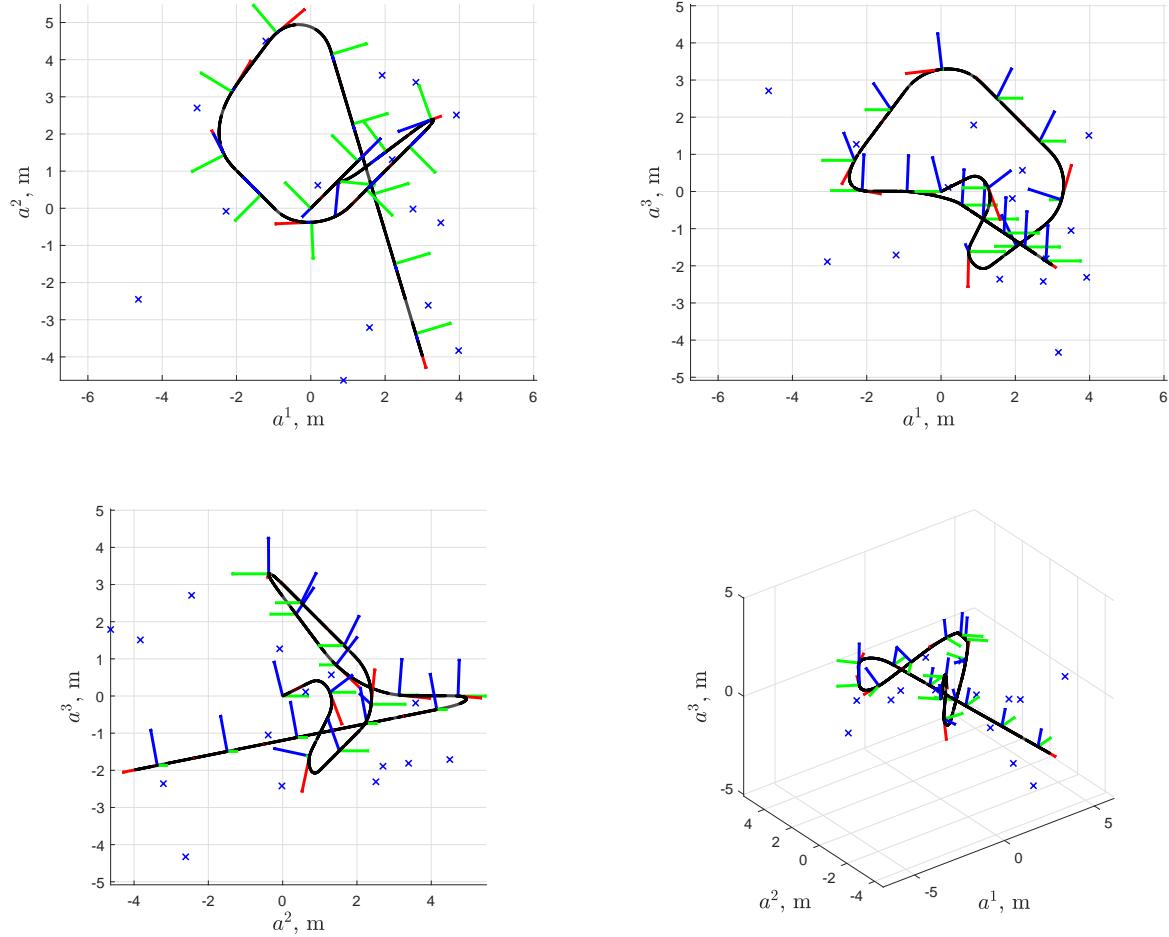


Figure 4.3: Trajectory used in simulations. The blue markers represent the landmarks, the trajectory is shown in black, and a triad is used to show the orientation at various times during the trajectory.

For the LIEKF,

$$\hat{\mathbf{T}}_0 = \mathbf{T}_0 \delta \mathbf{T}_0,$$

where $\delta \mathbf{T}_0 = \exp(\delta \xi_0^\wedge)$. Similarly, for the RIEKF,

$$\hat{\mathbf{T}}_0 = \delta \mathbf{T}_0 \mathbf{T}_0.$$

By using the appropriate error definition to initialize each filter, the same initial error leads to different actual initial conditions. This should not, in theory, change the difficulty of the estimation problem. Nonetheless, the effect of these differing initial conditions is tested.

To determine the effect of changing the initial error definition, 50 Monte Carlo trials are run using two different ways of initializing the invariant filters. First, the invariant filters are initialized in a manner consistent with their error definition. They are then initialized in the same manner as the MEKF. The sensor noise standard deviations were set to $\sigma_k^1 = 0.05 \text{ rad/s}$, $\sigma_k^2 = 0.05 \text{ m/s}$ and $\sigma_k^R = 0.05 \text{ m}$. The initial errors are drawn from a normal distribution with covariance $\mathbf{P}_0 = \text{diag}\left(0.1^2 \mathbf{1}, \left(\frac{\pi}{4}\right)^2 \mathbf{1}\right)$, with appropriate units.

The results obtained from initializing the filters in different manners are shown in Table 4.1. In practice, it is preferable to know the error in position and error in attitude separately, rather than the error in the pose. Therefore, it may be more representative to compare the filters using the MEKF error definitions. Thus, unless otherwise specified, the attitude error at each time step is given by $\delta \phi_k = \log_{SO(3)} \left(\hat{\mathbf{C}}_{ab_k}^\top \mathbf{C}_{ab_k} \right)^\vee$ and the position error is $\delta r_k = \|\mathbf{r}_a^{z_k w} - \hat{\mathbf{r}}_a^{z_k w}\|$. Using this metric, the initialization technique does not have a significant effect on the performance of the filters. For the LIEKF, changing from the correct initialization to the MEKF initialization sees a slight increase in mean root mean square error (RMSE) for both position and attitude. The difference for the RIEKF is negligible. Importantly, independent of the way the invariant filters are initialized, they still outperform the MEKFs. In this thesis, the appropriate error definition is used to initialize the filters. In fact, doing so ensures that the initial covariance is actually representative of the initial uncertainty in the state estimate, meaning the filters are always consistent to begin the simulation.

4.1.5.3 Monte Carlo Results

Having determined the appropriate procedure for comparing the filters, they can be extensively tested to determine the effect of sensor noise on their performance. To do so, the noise in each sensor is varied independently. Four different trials are run. The standard deviations of the noises injected into each measurement for each trial are summarized in Table 4.2. When the sensor noise is varied, it is incremented by 0.05. At each noise level, 50 Monte

Filter	Mean Attitude RMSE (rad)	Mean Position RMSE (m)
MEKF-L	0.2035	0.0310
LIEKF - App. Init.	0.1993	0.0288
LIEKF - MEKF Init.	0.2017	0.0297
MEKF-R	0.0739	0.0532
RIEKF - App. Init.	0.0734	0.0475
RIEKF - MEKF Init.	0.0733	0.0473

Table 4.1: Mean RMSE in the estimated states over 50 Monte Carlo simulations. The LIEKF and RIEKF were initialized both using an appropriate error definition consistent with the left and right-invariant errors and the error definition used to initialize the MEKFs.

Trial	σ_k^1 (rad/s)	σ_k^2 (m/s)	σ_k^R (m)
Rate Gyro Noise	0 to 1	0.05	0.05
Velocity Sensor Noise	0.05	0 to 1	0.05
Exteroceptive Sensor Noise	0.05	0.05	0 to 1
All Noises	0 to 1	0 to 1	0 to 1

Table 4.2: Standard deviation of the noise injected into each measurement for each trial.

Carlo trials are run, where the initial state estimate error and the noise profile is varied. As in the previous simulations, the initial errors are drawn from a normal distribution with covariance $\mathbf{P}_0 = \text{diag} \left(0.1^2 \mathbf{1}, \left(\frac{\pi}{4} \right)^2 \mathbf{1} \right)$, with appropriate units.

The results from varying the noise in the gyro, velocity sensor, correcting sensor, and all the sensors simultaneously can be found in Figures 4.4 to 4.11. Each figure presents the mean RMSE at each noise level for the IEKF and MEKF. The shaded areas encompass 80% of the data, to show the spread of the results. The difference between the mean MEKF and IEKF RMSE is also shown. A positive value indicates that the IEKF outperformed the MEKF. The shaded area once again represents 80% of the data. A convincing result would see the shaded area be entirely positive, meaning that only on rare occasions did the MEKF outperform the IEKF. On average, the invariant filters outperformed the standard MEKF in all the trials. A few important trends to note are as follows.

- The increased noise magnitude had a relatively small impact on the MEKF-R and RIEKF, meaning the difference in the filters remained somewhat constant over the all the trials.
- Increasing the noise in the velocity sensor led to erratic behaviour of the MEKF-L, leading to some outliers.
- Increasing the noise in the GPS measurements led to an increase in the difference

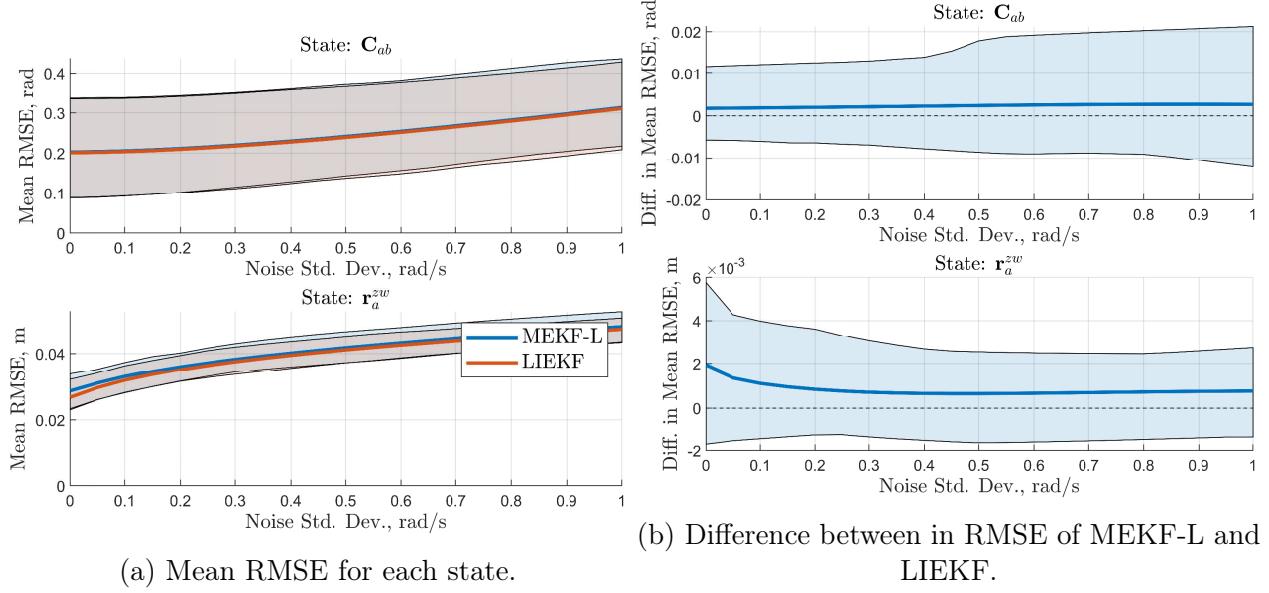


Figure 4.4: Results of 50 Monte Carlo simulations comparing the MEKF-L and LIEKF, where the amplitude of the noise in the rate gyro sensor was varied.

between the LIEKF and MEKF-L, as seen in Figure 4.8.

It is unsurprising that the IEKF would outperform the MEKF. The state-independent Jacobians mean the computed Kalman gain is accurate, as all the matrices used in its computations are known or are made up of noise-corrupted measurements, except for the error covariance \mathbf{P} . However, at steady state, the state estimate should be close enough to the true state that the state-dependence of the Jacobians is mitigated. This means that the improvement of the IEKF over the MEKF may simply be due to better performance in the transient. This is investigated in the next section.

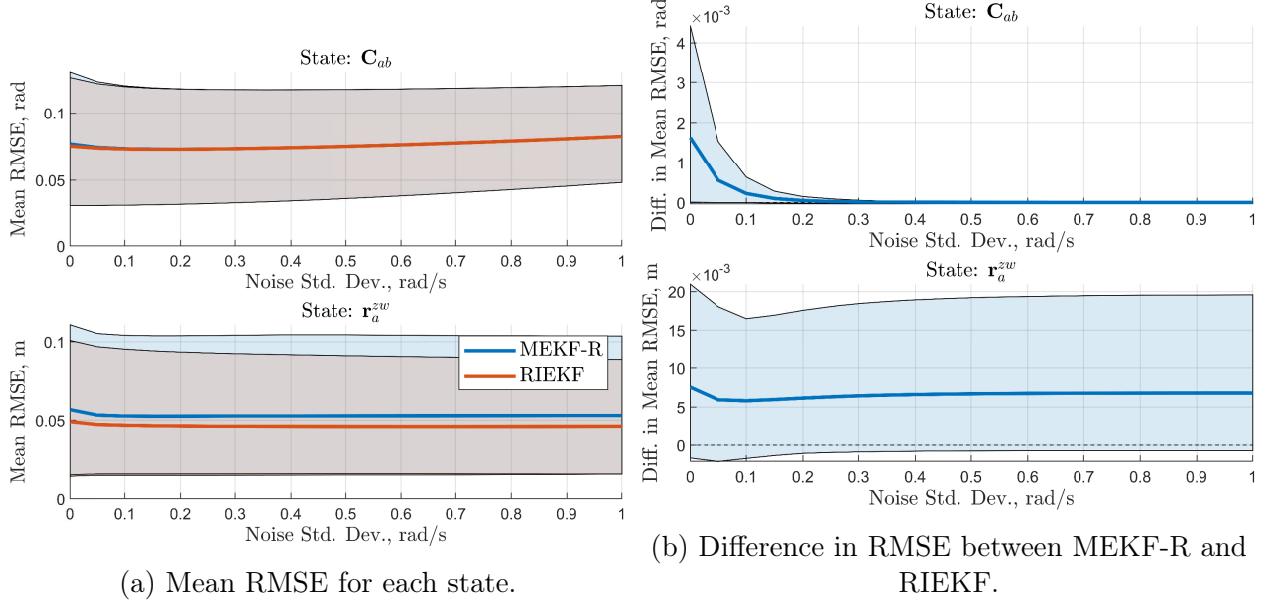


Figure 4.5: Results of 50 Monte Carlo simulations comparing the MEKF-R and RIEKF, where the amplitude of the noise in the rate gyro sensor was varied.

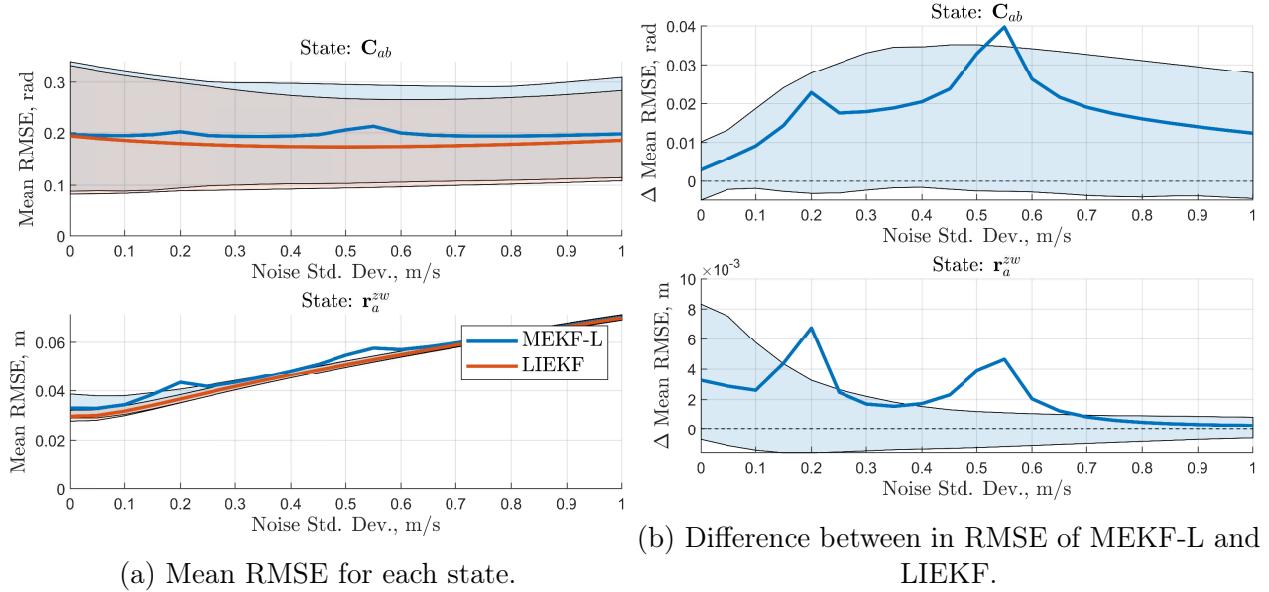


Figure 4.6: Results of 50 Monte Carlo simulations comparing the MEKF-L and LIEKF, where the amplitude of the noise in the velocity sensor was varied.

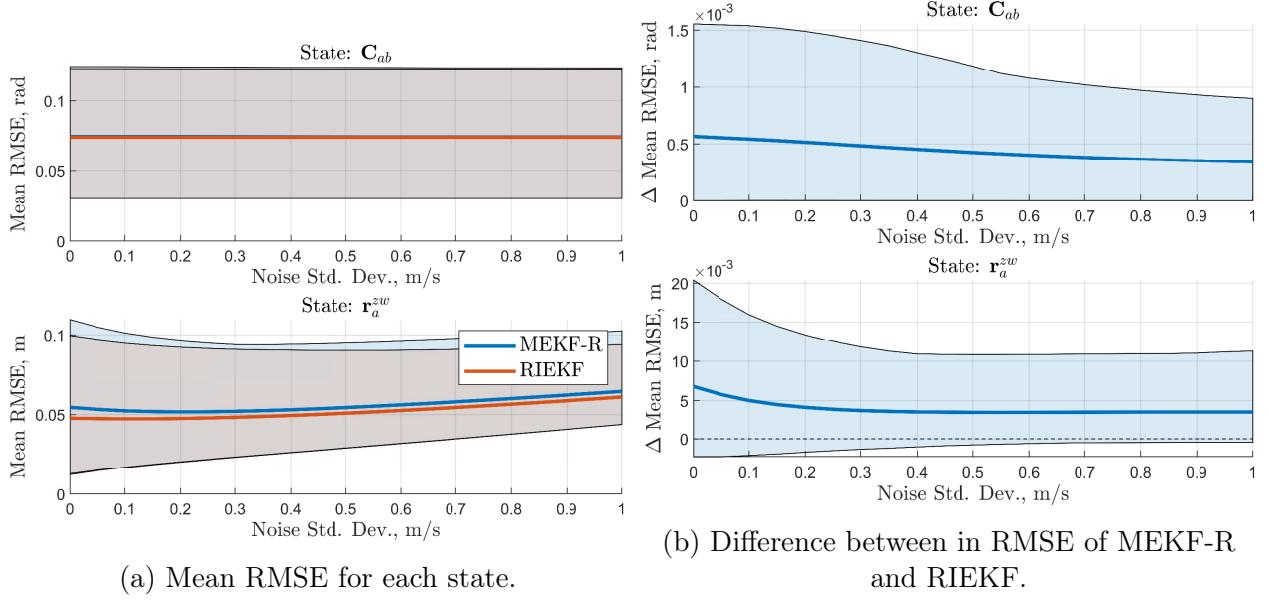


Figure 4.7: Results of 50 Monte Carlo simulations comparing the MEKF-R and RIEKF, where the amplitude of the noise in the velocity sensor was varied.

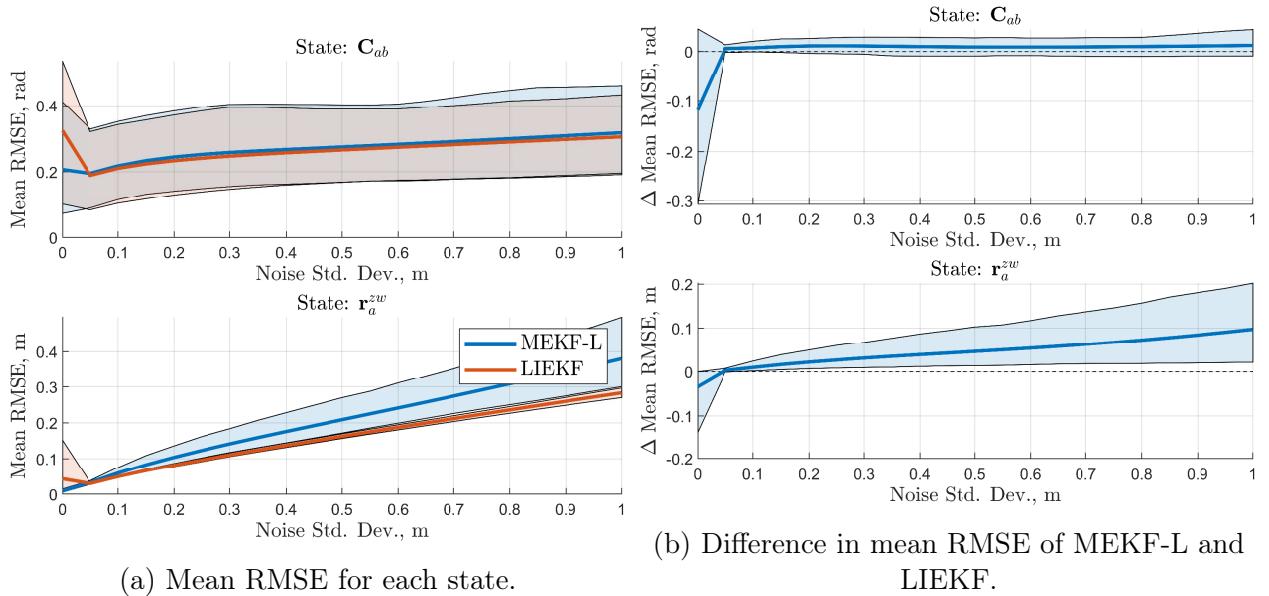


Figure 4.8: Results of 50 Monte Carlo simulations comparing the MEKF-L and LIEKF, where the amplitude of the noise in the correcting sensor was varied.

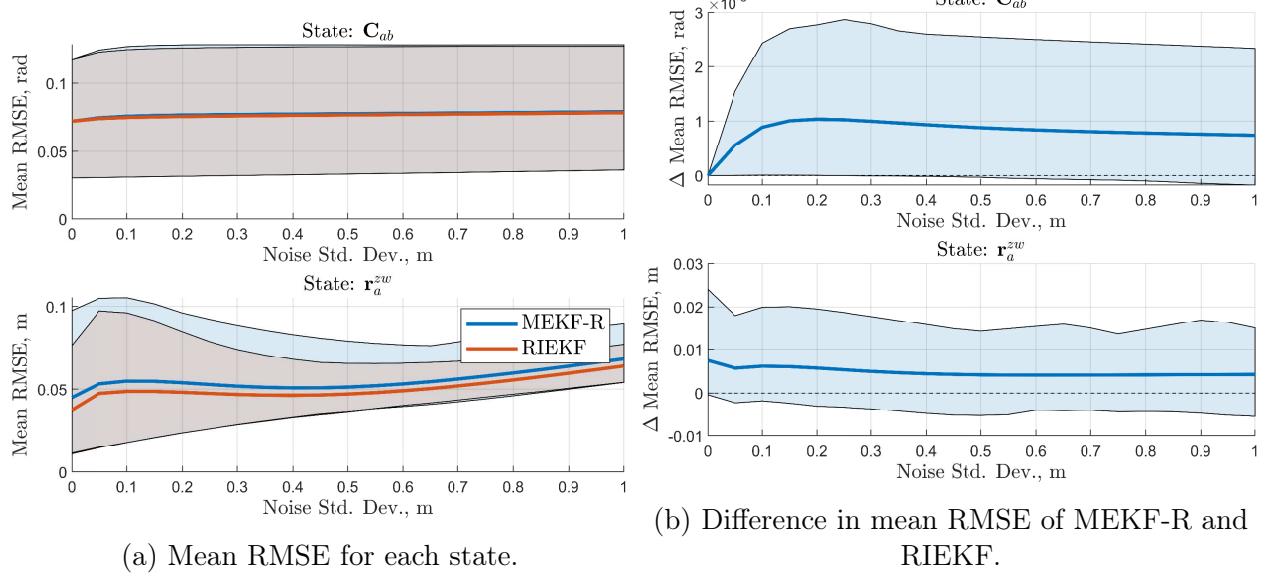


Figure 4.9: Results of 50 Monte Carlo simulations comparing the MEKF-R and RIEKF, where the amplitude of the noise in the correcting sensor was varied.

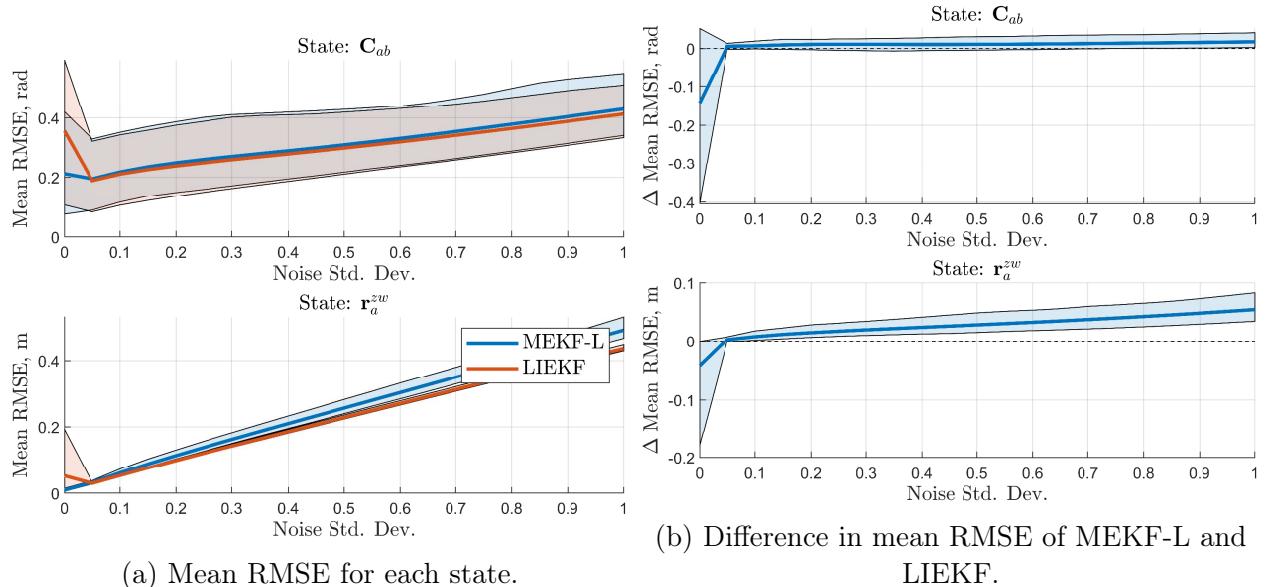


Figure 4.10: Results of 50 Monte Carlo simulations comparing the MEKF-L and LIEKF, where the amplitude of the noise in all sensors was varied.

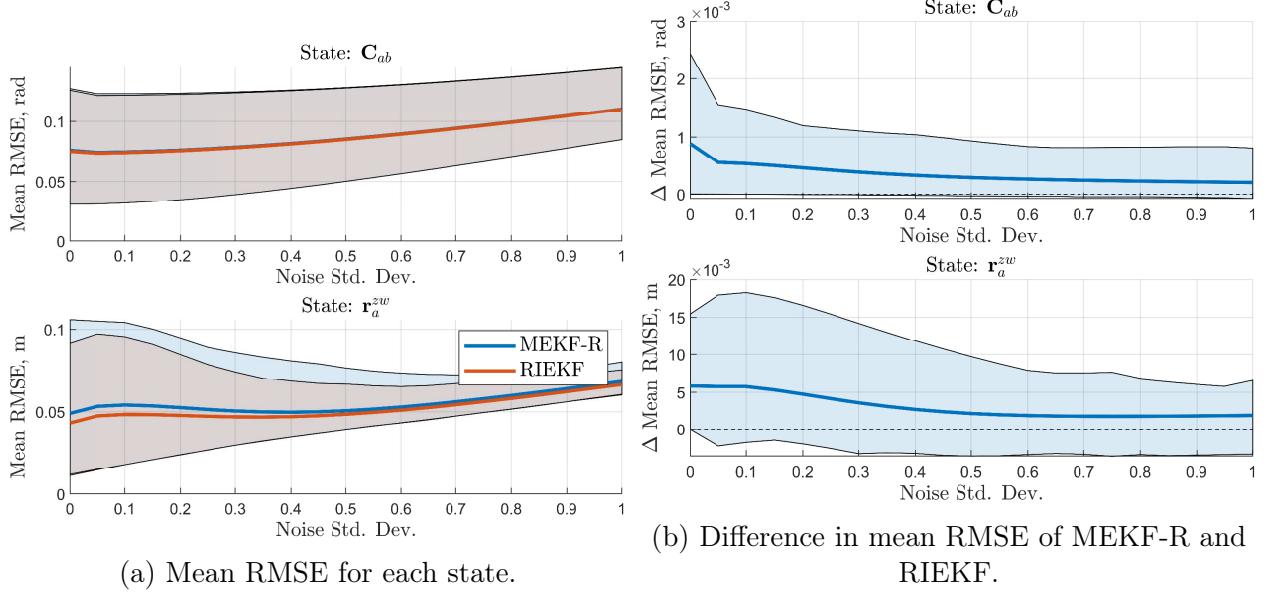


Figure 4.11: Results of 50 Monte Carlo simulations comparing the MEKF-R and RIEKF, where the amplitude of the noise in all sensors was varied.

4.1.5.4 Effect of Initial Attitude Error

As discussed in [1] and [2], the main advantage the IEKF claims over the MEKF are its convergence properties. Therefore, it may be reasonable to expect the IEKFs to perform substantially better than the MEKFs for high initial error. To test this, $\delta\phi_3$ was varied from 0 rad to $\frac{23\pi}{24}$ rad in increments of $\pi/24$ rad. In degrees, $\delta\phi_3$ is varied from 0° to 172.5° in increments of 7.5° . The initial position was initialized such that there was zero initial error.

The magnitude of the standard deviation of the noise in the sensors are set to $\sigma_k^1 = 0.05 \text{ rad/s}$, $\sigma_k^2 = 0.05 \text{ m/s}$ and $\sigma_k^R = 0.05 \text{ m}$. The initial covariance is set $\mathbf{P}_0 = \text{diag}(\delta\phi_3^2 \mathbf{1}, 1 \times 10^{-4} \mathbf{1})$, with appropriate units. This was chosen to reflect the varying uncertainty in the initial attitude and high certainty in the initial position. At each initial attitude error, 50 trials were run, where the noise profile was varied. The results of these simulations are shown in Figure 4.12. The results comparing the RIEKF and MEKF-R show that at high initial attitude error, the RIEKF consistently outperforms the MEKF-R. This is especially the case for position. The improvement in attitude error is marginal. The trend is less clear when comparing the LIEKF and MEFK-L, but the conclusion is similar. As expected, when the initial estimate of the state is poor, the invariant filters offer superior performance. When the initial state estimate is accurate, the Jacobians are close to the true Jacobians, nullifying any advantage the IEKFs had over the MEKFs, and their performance is indistinguishable.

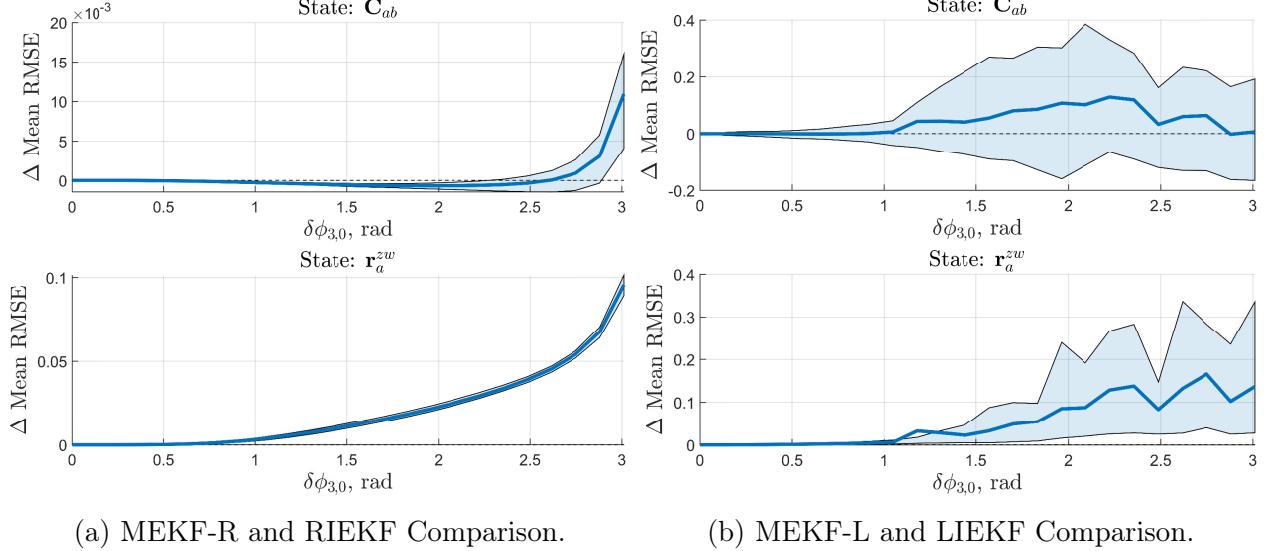


Figure 4.12: Results of 50 different noise profiles comparing invariant and multiplicative filters, where $\delta\phi_3$ was varied.

4.2 Applying the IEKF to a Realistic Data Set

The measurement model used in Section 4.1.3 is perfectly right-invariant. However, this is not a realistic measurement model, as no sensor exists that directly measures a relative landmark position. Rather, a LIDAR or camera is used. These measurements are neither left nor right-invariant. Therefore, the IEKF can not be directly applied, and must be adapted, as demonstrated in [30] with a range-and-bearing measurement model. To illustrate this, the sample problem from the previous section is replicated using a camera as an exteroceptive sensor. Let point c be fixed to the camera, and frame $\underline{\mathcal{F}}_c$ rotate with the camera. The position of the i^{th} landmark p^i relative to c resolved in the camera frame is

$$\mathbf{r}_c^{p_i c} = \mathbf{C}_{bc}^\top (\mathbf{C}_{ab}^\top (\mathbf{r}_a^{p_i w} - \mathbf{r}_a^{z w}) - \mathbf{r}_b^{c z}), \quad (4.13)$$

where \mathbf{C}_{bc} and $\mathbf{r}_b^{c z}$ are known. Letting $\mathbf{r}_c^{p_i c} = [x^i \ y^i \ z^i]^\top$, the measurements from a stereo camera at time t_k are then [3, p. 208]

$$\mathbf{y}_k^i = \mathbf{g}(\mathbf{r}_{c_k}^{p_i c_k}) = \frac{1}{z_k^i} \begin{bmatrix} f_u x_k^i \\ f_v y_k^i \\ f_u(x_k^i - b) \\ f_v y_k^i \end{bmatrix} + \begin{bmatrix} c_u \\ c_v \\ c_u \\ c_v \end{bmatrix} + \mathbf{v}_k \quad (4.14)$$

where c_u and c_v are the horizontal and vertical optical centres of the camera in pixels, f_u and f_v are the horizontal and vertical focal lengths of the camera in pixels, and b is the distance between the two centers of projection of the camera, in meters. The noise $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is zero-mean white noise with covariance \mathbf{R}_k .

4.2.1 Right-Invariant Kalman Filter Derivation

As the measurement model (4.14) is a function of a right-invariant measurement (4.13), a RIEKF is implemented. Therefore, the process model Jacobians from Section 4.1.3 are kept. Typically, the innovation $\mathbf{z}_k^i = \check{\mathbf{T}}_k(\mathbf{y}_k^i - \check{\mathbf{y}}_k^i)$ would be linearized to obtain the measurement model Jacobians. However, this is impossible to do in this case, as it is impossible to multiply the measurement from the camera by an element of $SE(3)$. To avoid this, the standard innovation from the EKF, $\mathbf{z}_k^i = \mathbf{y}_k^i - \check{\mathbf{y}}_k^i$, must be used.

To compute the new measurement model Jacobians, consider the first-order Taylor series expansion of (4.14),

$$\check{\mathbf{y}}_k^i + \delta\mathbf{y}_k^i = \mathbf{g}(\check{\mathbf{r}}_{c_k}^{p_i c_k}) + \underbrace{\frac{\partial \mathbf{g}(\mathbf{r}_{c_k}^{p_i c_k})}{\partial \mathbf{x}_k} \Big|_{\check{\mathbf{x}}_k, \check{\mathbf{v}}_k} \delta\mathbf{x}_k}_{\mathbf{H}_k} + \underbrace{\frac{\partial \mathbf{g}(\mathbf{r}_{c_k}^{p_i c_k})}{\partial \mathbf{v}_k} \Big|_{\check{\mathbf{x}}_k, \check{\mathbf{v}}_k} \delta\mathbf{v}_k.}_{\mathbf{M}_k}$$

Using the chain rule, the matrix \mathbf{H}_k is

$$\mathbf{H}_k = \frac{\partial \mathbf{g}(\mathbf{r}_{c_k}^{p_i c_k})}{\partial \mathbf{x}_k} \Big|_{\check{\mathbf{x}}_k, \check{\mathbf{v}}_k} = \frac{\partial \mathbf{g}(\mathbf{r}_{c_k}^{p_i c_k})}{\partial \mathbf{r}_{c_k}^{p_i c_k}} \frac{\partial \mathbf{r}_{c_k}^{p_i c_k}}{\partial \mathbf{x}_k} \Big|_{\check{\mathbf{x}}_k, \check{\mathbf{v}}_k}.$$

The term $\frac{\partial \mathbf{r}_{c_k}^{p_i c_k}}{\partial \mathbf{x}_k} \Big|_{\check{\mathbf{x}}_k, \check{\mathbf{v}}_k}$ is found using the perturbation method. The right-invariant error in

the pose $\delta\check{\mathbf{T}}_k$ can be written as

$$\begin{aligned}
\delta\check{\mathbf{T}}_k &= \check{\mathbf{T}}_k \mathbf{T}_k^{-1} \\
&= \begin{bmatrix} \check{\mathbf{C}}_{ab_k} & \check{\mathbf{r}}_a^{z_k w} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{C}_{ab_k}^\top & -\mathbf{C}_{ab_k}^\top \mathbf{r}_a^{z_k w} \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} \check{\mathbf{C}}_{ab_k} \mathbf{C}_{ab_k}^\top & \check{\mathbf{C}}_{ab_k} \mathbf{C}_{ab_k}^\top \mathbf{r}_a^{z_k w} - \check{\mathbf{r}}_a^{z_k w} \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} \delta\check{\mathbf{C}}_k & \delta\check{\mathbf{C}}_k \mathbf{r}_a^{z_k w} - \check{\mathbf{r}}_a^{z_k w} \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} \delta\check{\mathbf{C}}_k & \delta\check{\mathbf{r}}_k \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} \exp_{SO(3)}(\delta\check{\boldsymbol{\xi}}_k^\phi) & \mathbf{J}\delta\check{\boldsymbol{\xi}}_k^r \\ \mathbf{0} & 1 \end{bmatrix}.
\end{aligned}$$

Thus, perturbing the attitude results in $\mathbf{C}_{ab_k} = \delta\check{\mathbf{C}}_k^\top \check{\mathbf{C}}_{ab_k}$, which can also be written $\mathbf{C}_{ab_k} = \exp_{SO(3)}(-\delta\check{\boldsymbol{\xi}}_k^\phi) \check{\mathbf{C}}_{ab_k}$. The position perturbation results in $\mathbf{r}_a^{z_k w} = \delta\mathbf{C}_k^\top (\check{\mathbf{r}}_a^{z_k w} - \delta\mathbf{r}_k)$, which is equivalent to $\mathbf{r}_a^{z_k w} = \exp_{SO(3)}(-\delta\check{\boldsymbol{\xi}}_k^\phi) (\check{\mathbf{r}}_a^{z_k w} - \mathbf{J}\delta\check{\boldsymbol{\xi}}_k^r)$. Using these perturbations, along with $\mathbf{r}_{c_k}^{p_i c_k} = \check{\mathbf{r}}_{c_k}^{p_i c_k} + \delta\mathbf{r}_{c_k}^{p_i c_k}$, yields

$$\begin{aligned}
\mathbf{r}_{c_k}^{p_i c_k} &= \mathbf{C}_{bc}^\top (\mathbf{C}_{ab_k}^\top (\mathbf{r}_a^{p_i w} - \mathbf{r}_a^{z_k w}) - \mathbf{r}_b^{cz}), \\
\check{\mathbf{r}}_{c_k}^{p_i c_k} + \delta\mathbf{r}_{c_k}^{p_i c_k} &= \mathbf{C}_{bc}^\top \left(\left(\exp_{SO(3)}(-\delta\check{\boldsymbol{\xi}}_k^\phi) \check{\mathbf{C}}_{ab_k} \right)^\top \right. \\
&\quad \left. \left(\mathbf{r}_a^{p_i w} - \exp_{SO(3)}(-\delta\check{\boldsymbol{\xi}}_k^\phi) (\check{\mathbf{r}}_a^{z_k w} - \mathbf{J}\delta\check{\boldsymbol{\xi}}_k^r) \right) - \mathbf{r}_b^{cz} \right) \\
&= \mathbf{C}_{bc}^\top \left(\check{\mathbf{C}}_{ab_k}^\top \exp_{SO(3)}(\delta\check{\boldsymbol{\xi}}_k^\phi) \left(\mathbf{r}_a^{p_i w} - \exp_{SO(3)}(-\delta\check{\boldsymbol{\xi}}_k^\phi) (\check{\mathbf{r}}_a^{z_k w} - \mathbf{J}\delta\check{\boldsymbol{\xi}}_k^r) \right) - \mathbf{r}_b^{cz} \right).
\end{aligned} \tag{4.15}$$

Linearizing (4.15) by letting $\exp_{SO(3)}(-\delta\check{\boldsymbol{\xi}}_k^\phi) \approx \mathbf{1} + \delta\check{\boldsymbol{\xi}}_k^{\phi \times}$ and $\mathbf{J} \approx \mathbf{1}$,

$$\begin{aligned}
\check{\mathbf{r}}_{c_k}^{p_i c_k} + \delta\mathbf{r}_{c_k}^{p_i c_k} &= \mathbf{C}_{bc}^\top (\check{\mathbf{C}}_{ab_k}^\top (\mathbf{1} + \delta\check{\boldsymbol{\xi}}_k^{\phi \times})(\mathbf{r}_a^{p_i w} - (\mathbf{1} - \delta\check{\boldsymbol{\xi}}_k^{\phi \times})(\check{\mathbf{r}}_a^{z_k w} - \delta\check{\boldsymbol{\xi}}_k^r)) - \mathbf{r}_b^{cz}) \\
&= \mathbf{C}_{bc}^\top (\check{\mathbf{C}}_{ab_k}^\top (\mathbf{r}_a^{p_i w} - \check{\mathbf{r}}_a^{z_k w}) + \check{\mathbf{C}}_{ab_k}^\top (\delta\check{\boldsymbol{\xi}}_k^r + \delta\check{\boldsymbol{\xi}}_k^{\phi \times} \check{\mathbf{r}}_a^{z_k w}) + \check{\mathbf{C}}_{ab_k}^\top \delta\check{\boldsymbol{\xi}}_k^{\phi \times} (\mathbf{r}_a^{p_i w} - \check{\mathbf{r}}_a^{z_k w}) - \mathbf{r}_b^{cz}), \\
\delta\mathbf{r}_{c_k}^{p_i c_k} &= \mathbf{C}_{bc}^\top (\check{\mathbf{C}}_{ab_k}^\top (\delta\check{\boldsymbol{\xi}}_k^r + \delta\check{\boldsymbol{\xi}}_k^{\phi \times} \check{\mathbf{r}}_a^{z_k w}) + \check{\mathbf{C}}_{ab_k}^\top \delta\check{\boldsymbol{\xi}}_k^{\phi \times} (\mathbf{r}_a^{p_i w} - \check{\mathbf{r}}_a^{z_k w})) \\
&= \mathbf{C}_{bc}^\top (\check{\mathbf{C}}_{ab_k}^\top \delta\check{\boldsymbol{\xi}}_k^r + \check{\mathbf{C}}_{ab_k}^\top \delta\check{\boldsymbol{\xi}}_k^{\phi \times} (\mathbf{r}_a^{p_i w} - \check{\mathbf{r}}_a^{z_k w} + \check{\mathbf{r}}_a^{z_k w})) \\
&= \mathbf{C}_{bc}^\top \check{\mathbf{C}}_{ab_k}^\top \delta\check{\boldsymbol{\xi}}_k^r - \mathbf{C}_{bc}^\top \check{\mathbf{C}}_{ab_k}^\top \mathbf{r}_a^{p_i w \times} \delta\check{\boldsymbol{\xi}}_k^\phi.
\end{aligned}$$

Therefore

$$\frac{\partial \mathbf{r}_{c_k}^{p_i c_k}}{\partial \mathbf{x}_k} \Big|_{\check{\mathbf{x}}_k, \check{\mathbf{v}}_k} = \begin{bmatrix} -\mathbf{C}_{bc}^T \check{\mathbf{C}}_{ab_k}^T \mathbf{r}_a^{p_i w \times} & \mathbf{C}_{bc}^T \check{\mathbf{C}}_{ab_k}^T \end{bmatrix}.$$

The term $\frac{\partial \mathbf{g}(\mathbf{r}_{c_k}^{p_i c_k})}{\partial \mathbf{r}_{c_k}^{p_i c_k}}$ is computed analytically, yielding

$$\frac{\partial \mathbf{g}(\mathbf{r}_{c_k}^{p_i c_k})}{\partial \mathbf{r}_{c_k}^{p_i c_k}} = \frac{1}{z_k^i} \begin{bmatrix} f_u & 0 & -\frac{f_u x_k^i}{z_k^i} \\ 0 & f_v & -\frac{f_v y_k^i}{z_k^i} \\ f_u & 0 & -\frac{f_u(x_k^i - b)}{z_k^i} \\ 0 & f_v & -\frac{f_v y_k^i}{z_k^i} \end{bmatrix}.$$

Therefore, the measurement model Jacobians are

$$\mathbf{H}_k = \underset{i=1,\dots,m}{\text{row}} \left(\frac{1}{z_k^i} \begin{bmatrix} f_u & 0 & -\frac{f_u x_k^i}{z_k^i} \\ 0 & f_v & -\frac{f_v y_k^i}{z_k^i} \\ f_u & 0 & -\frac{f_u(x_k^i - b)}{z_k^i} \\ 0 & f_v & -\frac{f_v y_k^i}{z_k^i} \end{bmatrix} \begin{bmatrix} -\mathbf{C}_{bc}^T \check{\mathbf{C}}_{ab_k}^T \mathbf{r}_a^{p_i w \times} & \mathbf{C}_{bc}^T \check{\mathbf{C}}_{ab_k}^T \end{bmatrix} \right) \quad (4.16)$$

and $\mathbf{M} = \mathbf{1}$. Comparing (4.16) to (4.11), the impacts of the using the camera model become clear. The measurement model Jacobian depends on the state estimate, as it depends on $\check{\mathbf{r}}_{c_k}^{p_i c_k}$ and $\check{\mathbf{C}}_{ab_k}$.

4.2.2 MEKF Solution

The process model Jacobians are identical to those in Section 4.1.4. Using a similar technique to the RIEKF just described, the measurement Jacobians for the MEKF are

$$\mathbf{H}_k = \underset{i=1,\dots,m}{\text{row}} \left\{ \frac{1}{z_k^i} \begin{bmatrix} f_u & 0 & -\frac{f_u x_k^i}{z_k^i} \\ 0 & f_v & -\frac{f_v y_k^i}{z_k^i} \\ f_u & 0 & -\frac{f_u(x_k^i - b)}{z_k^i} \\ 0 & f_v & -\frac{f_v y_k^i}{z_k^i} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{bc}^T (\check{\mathbf{C}}_{ab_k}^T (\mathbf{r}_a^{p_i w} - \check{\mathbf{r}}_a^{z_k w}))^\times & -\mathbf{C}_{bc}^T \check{\mathbf{C}}_{ab_k}^T \end{bmatrix} \right\} \quad (4.17)$$

and $\mathbf{M} = \mathbf{1}$.

4.2.3 Filtering Using Pseudo-measurements

To avoid the state-dependent \mathbf{H}_k matrix from the previous section, the measurement model must be right-invariant. To attain this using the nonlinear camera model, the measurements

are preprocessed so that the output of the sensor is $\mathbf{r}_{b_k}^{p_i z_k}$. The issue with doing this is that the noise is no longer Gaussian. However, in practice, EKFs perform well despite the Gaussian noise assumption being routinely not met.

In a pre-processing step, the output from the camera is passed through

$$\mathbf{v}_k^i = \mathbf{h}(\mathbf{y}_k^i) = \mathbf{C}_{bc} \left(\frac{b}{y_{k_1}^i - y_{k_3}^i} \begin{bmatrix} y_{k_1}^i - c_u \\ \frac{f_u}{f_v} (y_{k_2}^i - c_v) \\ f_u \end{bmatrix} \right) + \mathbf{r}_b^{cz}$$

where the output of the camera model $\mathbf{r}_{b_k}^{p_i z_k}$ is denoted \mathbf{v}_k^i . These are the pesudo-measurements. Given the truth data, it is also possible to compute the expected pesudo-measurements, $\hat{\mathbf{v}}_k^i$, using

$$\hat{\mathbf{v}}_k^i = \mathbf{C}_{ab_k}^\top (\mathbf{r}_a^{p_i w} - \mathbf{r}_a^{z_k w}). \quad (4.18)$$

To obtain the noise, each pseudomeasurement is subtracted from its expected value, $\boldsymbol{\nu}_k^i = \mathbf{v}_k^i - \hat{\mathbf{v}}_k^i$. The noises at each pseudomeasurement are then arranged in a wide matrix,

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_1 & \cdots & \boldsymbol{\nu}_k \end{bmatrix}.$$

Each row of $\boldsymbol{\nu}$ can now be treated as an observation drawn from some probability distribution. For the Kalman filter assumptions to hold, this distribution must be a zero-mean Gaussian distribution. Computing the sample mean yields $\bar{\boldsymbol{\nu}} = \mathbb{E}[\boldsymbol{\nu}]$, which, ideally, should be $\mathbf{0}$. The sample covariance is then computed to yield \mathbf{R} . This value can be used as an approximate measurement noise covariance matrix in the filters derived below.

By preprocessing the measurement, a right-invariant measurement model (4.18) is obtained, and a standard RIEKF can be used. The right-invariant innovation is

$$\mathbf{z}_k^i = \mathbf{C}_{ab_k}(\mathbf{v}_k^i - \hat{\mathbf{v}}_k^i).$$

The derivation of the measurement model Jacobians is identical to that in Section 4.1.3. Therefore,

$$\mathbf{H} = \underset{i=1,\dots,m}{\text{row}} \left(\begin{bmatrix} -\mathbf{r}_a^{p_i w \times} & \mathbf{1} \end{bmatrix} \right)$$

and $\mathbf{M}_k = \text{diag}(\check{\mathbf{C}}_{ab_k}, \dots, \check{\mathbf{C}}_{ab_k})$. As before, the MEKF measurement Jacobians are

$$\mathbf{H}_k = \underset{i=1,\dots,m}{\text{row}} \left(\begin{bmatrix} (\check{\mathbf{C}}_{ab_k}^\top (\mathbf{r}_a^{p_i w} - \check{\mathbf{r}}_a^{z_k w}))^\times & -\check{\mathbf{C}}_{ab_k}^\top \end{bmatrix} \right)$$

and $\mathbf{M} = \mathbf{1}$.

4.2.4 Results

Using the 4 different filters derived above, namely the camera-based MEKF (MEKF-C), camera-based RIEKF (RIEKF-C), the MEKF using the relative landmark pesudo-measurements(MEKF-RL), and the RIEKF using the relative landmark pesudo-measurements(RIEKF-RL), it is possible to determine whether it is still beneficial to use the invariant framework, despite the measurement model being neither left nor right-invariant. First, the filters are tested in simulation, followed by testing done on the experimental data from the Starry Night dataset [31].

To use the pseudo-measurements for the RIEKF-RL and MEKF-RL, the covariance matrix of the new measurements must be found. Using the methodology described above, the sample mean of the pseudo-measurements from the Starry Night dataset is

$$E[\boldsymbol{\nu}] = \begin{bmatrix} 0.0047 \\ 0.0031 \\ -0.0024 \end{bmatrix} \text{ m.}$$

The distribution is close to zero-mean, but as expected, the nonlinear transformation has shifted the mean. The sample covariance is

$$\mathbf{R} = E[(\boldsymbol{\nu} - E[\boldsymbol{\nu}]) (\boldsymbol{\nu} - E[\boldsymbol{\nu}])^T] = \begin{bmatrix} 1.11 \times 10^{-3} & 1.59 \times 10^{-4} & -1.23 \times 10^{-4} \\ 1.59 \times 10^{-4} & 2.42 \times 10^{-4} & -1.07 \times 10^{-4} \\ -1.23 \times 10^{-4} & -1.07 \times 10^{-4} & 7.00 \times 10^{-4} \end{bmatrix} \text{ m}^2.$$

These values are used in both simulation and on the experimental data.

In simulation, simple Monte Carlo trials are run to evaluate the performance of the filters. The covariance matrices of the noise in the sensors are set to the values found in the Starry Night data set. Several different initial covariances \mathbf{P}_0 were considered to initialize the filters. The performance was found to be highly dependent on the magnitude of the initial attitude error. Figure 4.13 shows the effect of initial attitude error on the mean RMSE in simulation. There is therefore already a clear advantage of using the pseudo-measurements as opposed to the camera-based model, regardless of the chosen EKF variant. The filters using the pesudomeasurements converge for a much wider range of initial attitude errors.

Based on these results, the error in the initial conditions were drawn from a zero-mean normal distribution with covariance $\mathbf{P}_0 = \text{diag}\left(0.1^2 \mathbf{1}, \left(\frac{\pi}{12}\right)^2 \mathbf{1}\right)$. The four filters are initially tested in simulation. The results of 500 MC simulations are shown in Figure 4.14, where the error bars capture 80% of the data. On average, the filters using the pseudo-measurements

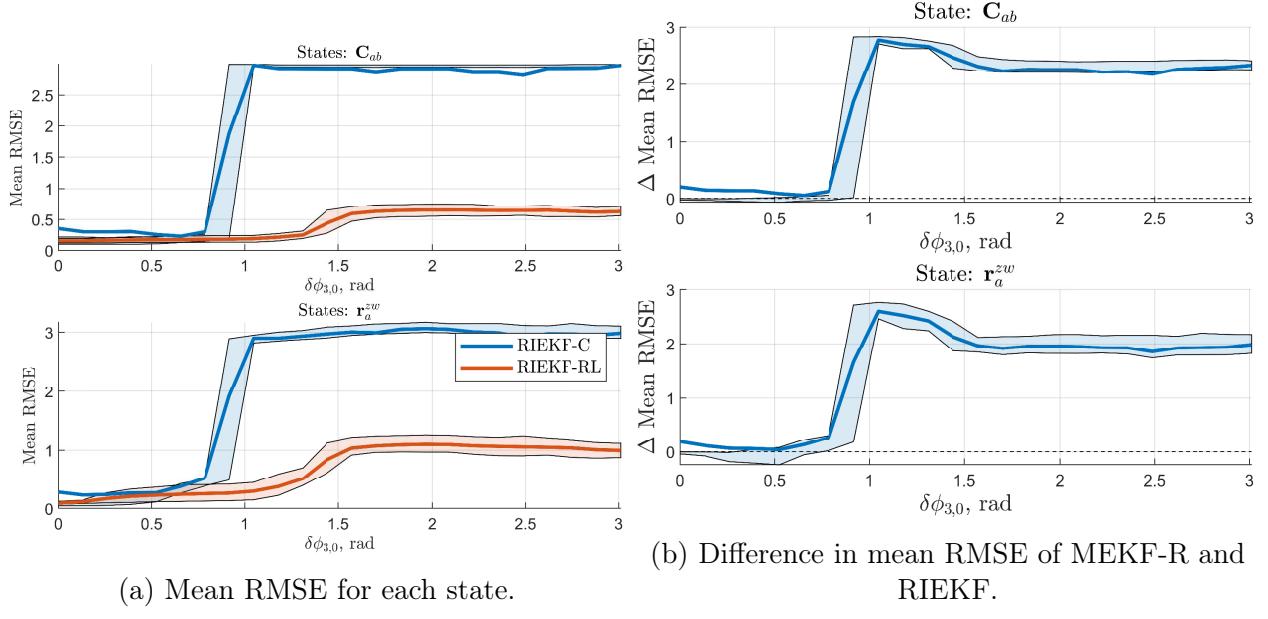


Figure 4.13: Results of 50 Monte Carlo trials comparing the RIEKF-C and RIEKF-RL at different initial attitude errors.

outperformed the sensors using the camera measurements directly. The RIEKF-C yielded a better attitude estimate than the MEKF-C, but a poorer position estimate. This trend is repeated in the filters using the pseudo-measurements. However, there is significant spread in the results, as shown by the overlapping error bars. The RIEKF-C did indeed have a lower mean attitude RMSE than the MEKF-C, but it only outperformed the MEKF-C in 51.2% of the trials. A similar trend is observed in the filters which use pseudo-measurments, where the RIEKF-RL outperformed the MEKF-RL in 53.2% of the trials. The RIEKF does provide a better attitude estimate than the MEKF on average, but the impact is marginal. Running similar trials on real data reinforced some ideas from simulation. In particular, the MEKFs outperformed the IEKFs in position estimation. Furthermore, there was again little difference in the attitude estimate. These results are shown in Figure 4.15. Despite having a clearly higher mean RMSE, the RIEKF-C outperformed the MEFK-C in 45.6% of the trials. Similarly, the RIEKF-RL outperformed the MEKF-RL 43% of the time, the camera-based filters outperform the filters that use the pseudo-measurements.

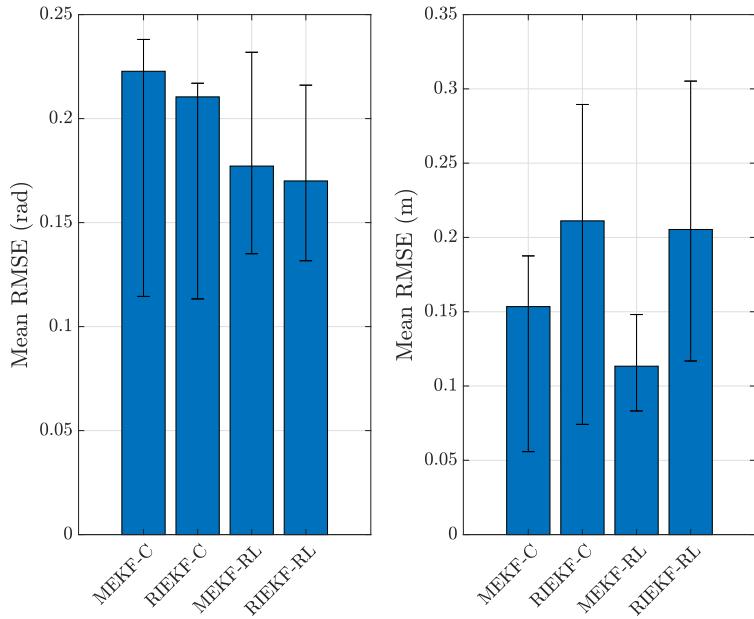


Figure 4.14: Results of 500 Monte Carlo trials comparing the MEKF-C, RIEKF-C, MEKF-RL, and RIEKF-RL on simulated data. The mean RMSEs in attitude (left) and position (right) are shown.

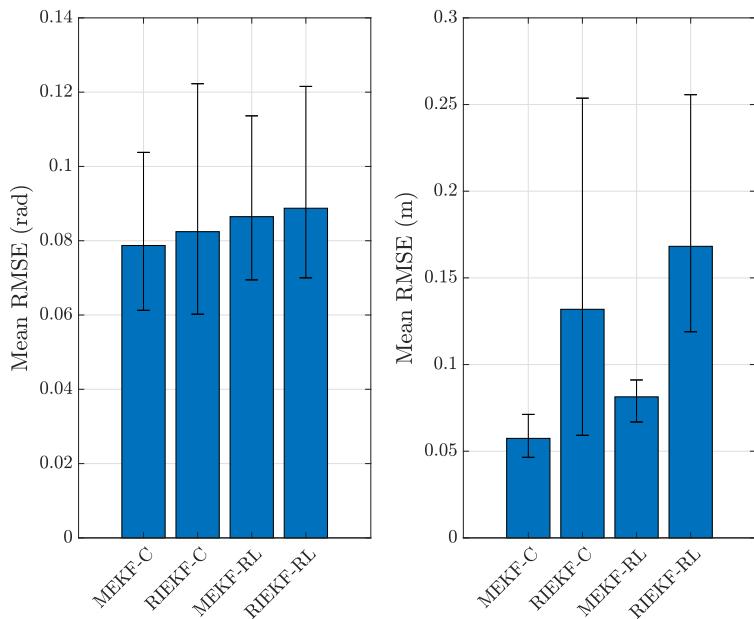


Figure 4.15: Results of 500 Monte Carlo trials comparing the MEKF-C, RIEKF-C, MEKF-RL, and RIEKF-RL on real data. The mean RMSEs in attitude (left) and position (right) are shown.

4.3 Estimating Bias In the Invariant Framework

In many estimation problems, sensor biases must be estimated. The IEKF is not particularly well suited for bias estimation, as including bias in the process model destroys the group-affine property. Despite this, there still may be advantages to estimating bias in the invariant framework. To illustrate this, the sample problem is modified to include a time-varying bias in the rate gyro. For the sake of brevity, only the right-invariant model is shown. The rate gyro measurement model is now

$$\mathbf{u}_b^1 = \boldsymbol{\omega}_b^{ba} - \boldsymbol{\beta}_b - \mathbf{w}_b^1,$$

where the bias is modelled as a random walk with $\dot{\boldsymbol{\beta}}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^3)$.

The state can now be formulated as an element of a new matrix Lie group [29], \mathcal{G}_1 . An element of \mathcal{G}_1 is

$$\mathbf{X} = \begin{bmatrix} \mathbf{C}_{ab} & \mathbf{r}_a^{zw} \\ & 1 \\ & & \mathbf{1} & \boldsymbol{\beta}_b \\ & & & 1 \end{bmatrix}.$$

The inverse is

$$\mathbf{X}^{-1} = \begin{bmatrix} \mathbf{C}_{ab} & -\mathbf{C}_{ab}\mathbf{r}_a^{zw} \\ & 1 \\ & & \mathbf{1} & -\boldsymbol{\beta}_b \\ & & & 1 \end{bmatrix}.$$

Let \mathfrak{g}_1 be the matrix Lie algebra of \mathcal{G}_1 . The column matrix $\boldsymbol{\xi} \in \mathbb{R}^9$ is mapped to \mathfrak{g}_1 using

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\xi}^\phi \\ \boldsymbol{\xi}^r \\ \boldsymbol{\xi}^\beta \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\xi}^{\phi \times} & \boldsymbol{\xi}^r \\ 0 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\xi}^\beta \\ 0 & 0 \end{bmatrix}.$$

The exponential map from \mathfrak{g}_1 to \mathcal{G}_1 is

$$\exp(\boldsymbol{\xi}^\wedge) = \begin{bmatrix} \exp_{SO(3)}(\boldsymbol{\xi}^{\phi \times}) & \mathbf{J}\boldsymbol{\xi}^r \\ & 1 \\ & & \mathbf{1} & \boldsymbol{\xi}^\beta \\ & & & 1 \end{bmatrix},$$

where \mathbf{J} is given by (2.7). The adjoint representation of an element of \mathcal{G}_1 is

$$\text{Ad}(\mathbf{X}) = \begin{bmatrix} \mathbf{C}_{ab} & \mathbf{0} & \mathbf{0} \\ \mathbf{r}_a^{zw} \times \mathbf{C}_{ab} & \mathbf{C}_{ab} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}$$

The kinematics in matrix form are

$$\dot{\mathbf{X}} = \mathbf{X}\boldsymbol{\varpi}^\wedge,$$

where

$$\boldsymbol{\varpi} = \begin{bmatrix} \boldsymbol{\omega}_b^{ba} \\ \mathbf{v}_b^{zw/a} \\ \mathbf{0} \end{bmatrix}.$$

Substituting for the measurement model,

$$\dot{\mathbf{X}} = \mathbf{X}(\mathbf{u}_b + \mathbf{w}_b)^\wedge,$$

where

$$\mathbf{u}_b = \begin{bmatrix} \mathbf{u}_b^1 \\ \mathbf{u}_b^2 \\ \mathbf{0} \end{bmatrix}$$

and

$$\mathbf{w}_b = \begin{bmatrix} \mathbf{w}_b^1 \\ \mathbf{w}_b^2 \\ \mathbf{w}_b^3 \end{bmatrix}.$$

The kinematics (4.3) are not group affine due to the bias. Therefore, the process model Jacobians will depend on the state estimate. However, using an invariant error definition, the Jacobians can still be derived in a way that causes the Jacobians to be state dependent in different ways. Whether this is advantageous is a question to be answered, at least partially, in this section, and may depend on the problem. It will be tested on the $SE(3)$ sample problem in the right-invariant case, using the measurement model (4.6).

The right-invariant error between some estimated state $\bar{\mathbf{X}}$ and the true state \mathbf{X} is

$$\begin{aligned}
\delta \mathbf{X} &= \bar{\mathbf{X}} \mathbf{X}^{-1} \\
&= \begin{bmatrix} \bar{\mathbf{C}}_{ab} & \bar{\mathbf{r}}_a^{zw} \\ 1 & \mathbf{1} & \bar{\beta}_b^1 \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{C}_{ab}^\top & -\mathbf{C}_{ab}\mathbf{r}_a^{zw} \\ & 1 \\ & \mathbf{1} & -\beta_b^1 \\ & & 1 \end{bmatrix} \\
&= \begin{bmatrix} \bar{\mathbf{C}}_{ab}\mathbf{C}_{ab}^\top & \bar{\mathbf{r}}_a^{zw} - \bar{\mathbf{C}}_{ab}\mathbf{C}_{ab}^\top\mathbf{r}_a^{zw} \\ & 1 \\ & \mathbf{1} & \bar{\beta}_b^1 - \beta_b^1 \\ & & 1 \end{bmatrix} \\
&= \begin{bmatrix} \delta \mathbf{C} & \delta \mathbf{r} \\ 1 & \mathbf{1} & \delta \boldsymbol{\beta} \\ & 1 \end{bmatrix}.
\end{aligned}$$

Defining

$$\hat{\mathbf{u}}_b = \begin{bmatrix} \mathbf{u}_b^1 + \hat{\boldsymbol{\beta}}_b \\ \mathbf{u}_b^2 \\ \mathbf{0} \end{bmatrix},$$

the right-invariant error propagation is

$$\begin{aligned}
\delta \dot{\mathbf{X}} &= \dot{\hat{\mathbf{X}}} \mathbf{X}^{-1} + \hat{\mathbf{X}} \dot{\mathbf{X}}^{-1} \\
&= \hat{\mathbf{X}} \hat{\mathbf{u}}_b^\wedge \mathbf{X}^{-1} + \hat{\mathbf{X}} \mathbf{X}^{-1} \dot{\hat{\mathbf{X}}}^{-1} \\
&= \hat{\mathbf{X}} \hat{\mathbf{u}}_b^\wedge \mathbf{X}^{-1} + \hat{\mathbf{X}} \mathbf{X}^{-1} \mathbf{X}(\mathbf{u}_b + \mathbf{w}_b)^\wedge \mathbf{X}^{-1} \\
&= \hat{\mathbf{X}} (\hat{\mathbf{u}}_b - \mathbf{u}_b - \mathbf{w}_b)^\wedge \mathbf{X}^{-1} \\
&= \hat{\mathbf{X}} (\hat{\mathbf{u}}_b - \mathbf{u}_b - \mathbf{w}_b)^\wedge \hat{\mathbf{X}}^{-1} \hat{\mathbf{X}} \mathbf{X}^{-1} \\
&= (\text{Ad}(\hat{\mathbf{X}})(\hat{\mathbf{u}}_b - \mathbf{u}_b - \mathbf{w}_b))^\wedge \delta \mathbf{X}.
\end{aligned}$$

To simplify, let

$$\begin{aligned}
\delta \mathbf{u} &= \hat{\mathbf{u}}_b - \mathbf{u}_b \\
&= \begin{bmatrix} \mathbf{u}_b^1 + \hat{\boldsymbol{\beta}}_b \\ \mathbf{u}_b^2 \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{u}_b^1 + \boldsymbol{\beta}_b \\ \mathbf{u}_b^2 \\ \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} \delta \boldsymbol{\beta} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\
&= \mathbf{B} \delta \boldsymbol{\xi},
\end{aligned}$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Linearizing by letting $\delta \mathbf{X} = \mathbf{1} + \delta \boldsymbol{\xi}^\wedge$ and neglecting second order terms,

$$\begin{aligned}
\frac{d}{dt} (\mathbf{1} + \delta \boldsymbol{\xi}^\wedge) &= \left(\text{Ad}(\hat{\mathbf{X}})(\mathbf{B} \delta \boldsymbol{\xi} - \mathbf{w}_b) \right)^\wedge (\mathbf{1} + \delta \boldsymbol{\xi}^\wedge), \\
\delta \dot{\boldsymbol{\xi}}^\wedge &= \left(\text{Ad}(\hat{\mathbf{X}})(\mathbf{B} \delta \boldsymbol{\xi} - \mathbf{w}_b) \right)^\wedge, \\
\delta \dot{\boldsymbol{\xi}} &= \text{Ad}(\hat{\mathbf{X}}) \mathbf{B} \delta \boldsymbol{\xi} - \text{Ad}(\hat{\mathbf{X}}) \mathbf{w}_b.
\end{aligned}$$

Therefore, $\mathbf{A} = \text{Ad}(\hat{\mathbf{X}}) \mathbf{B}$ and $\mathbf{L} = -\text{Ad}(\hat{\mathbf{X}})$. Explicitly,

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \hat{\mathbf{C}}_{ab} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{r}}_a^{zw} \times \hat{\mathbf{C}}_{ab} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

The matrix \mathbf{A} depends on the state estimate because the process model is not group affine.

The innovation associated with the i^{th} landmark is

$$\begin{aligned}
\begin{bmatrix} \mathbf{z}_k^i \\ \mathbf{0} \end{bmatrix} &= \check{\mathbf{X}}_k \left(\begin{bmatrix} \mathbf{y}_{b_k}^i \\ 1 \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \check{\mathbf{y}}_{b_k}^i \\ 1 \\ \mathbf{0} \end{bmatrix} \right) \\
&= \check{\mathbf{X}}_k \left(\mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{b_k} \\ \mathbf{0} \end{bmatrix} - \check{\mathbf{X}}_k^{-1} \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \\ \mathbf{0} \end{bmatrix} \right) \\
&= \delta \check{\mathbf{T}}_k \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix} + \check{\mathbf{T}}_k \begin{bmatrix} \mathbf{v}_{b_k} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \end{bmatrix}. \tag{4.19}
\end{aligned}$$

Linearizing (4.19) by letting $\delta \check{\mathbf{X}}_k \approx \mathbf{1} + \delta \check{\boldsymbol{\xi}}_k^\wedge$ and $\mathbf{v}_{b_k} = \delta \mathbf{v}_{b_k}$,

$$\begin{aligned}
\begin{bmatrix} \mathbf{z}_k^i \\ \mathbf{0} \end{bmatrix} &\approx (\mathbf{1} + \delta \check{\boldsymbol{\xi}}_k^\wedge) \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \\ \mathbf{0} \end{bmatrix} + \check{\mathbf{X}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \delta \check{\boldsymbol{\xi}}_k^\wedge \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \\ \mathbf{0} \end{bmatrix} + \check{\mathbf{X}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} \delta \check{\boldsymbol{\xi}}^{\phi \times} & \delta \check{\boldsymbol{\xi}}^r \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_a^{p_i w} \\ 1 \\ \mathbf{0} \end{bmatrix} + \check{\mathbf{X}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} \delta \check{\boldsymbol{\xi}}^{\phi \times} \mathbf{r}_a^{p_i w} + \delta \check{\boldsymbol{\xi}}^r \\ \mathbf{0} \end{bmatrix} + \check{\mathbf{X}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} -\mathbf{r}_a^{p_i w \times} \delta \check{\boldsymbol{\xi}}^{\phi \times} + \delta \check{\boldsymbol{\xi}}^r \\ \mathbf{0} \end{bmatrix} + \check{\mathbf{X}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{H}}^i \delta \check{\boldsymbol{\xi}}_k + \check{\mathbf{X}}_k \begin{bmatrix} \delta \mathbf{v}_{b_k}^i \\ \mathbf{0} \end{bmatrix}, \tag{4.20}
\end{aligned}$$

where

$$\tilde{\mathbf{H}}^i = \begin{bmatrix} -\mathbf{r}_a^{p_i w \times} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Noting that the bottom row of (4.20) is always $\mathbf{0}$, it can be written

$$\mathbf{z}_k^i = \mathbf{H}^i \delta \check{\boldsymbol{\xi}}_k + \mathbf{M}_k^i \delta \mathbf{v}_{b_k}^i$$

where

$$\mathbf{H}^i = \begin{bmatrix} -\mathbf{r}_a^{p_i w \times} & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

and $\mathbf{M}_k^i = \check{\mathbf{C}}_{ab_k}^\top$. For the m landmarks,

$$\mathbf{H} = \underset{i=1, \dots, m}{\text{row}} \left(\begin{bmatrix} -\mathbf{r}_a^{p_i w \times} & \mathbf{1} \end{bmatrix} \right) \quad (4.21)$$

and $\mathbf{M}_k = \text{diag}(\check{\mathbf{C}}_{ab_1}^\top, \dots, \check{\mathbf{C}}_{ab_m}^\top)$. In this case, \mathbf{H} is state independent.

4.3.1 MEKF Solution

Similar to Section 4.1.4, the RIEKF is compared to an MEKF. The MEKF errors are defined as $\delta \mathbf{C} = \bar{\mathbf{C}}_{ab}^\top \mathbf{C}_{ab}$, $\delta \mathbf{r} = \mathbf{r}_a^{zw} - \bar{\mathbf{r}}_a^{zw}$, and $\delta \boldsymbol{\beta}_b = \boldsymbol{\beta}_b - \bar{\boldsymbol{\beta}}_b$. The MEKF process model Jacobians are

$$\mathbf{A} = \begin{bmatrix} -\mathbf{u}_b^{1 \times} & \mathbf{0} & \mathbf{1} \\ -\hat{\mathbf{C}}_{ab} \mathbf{u}_b^{2 \times} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

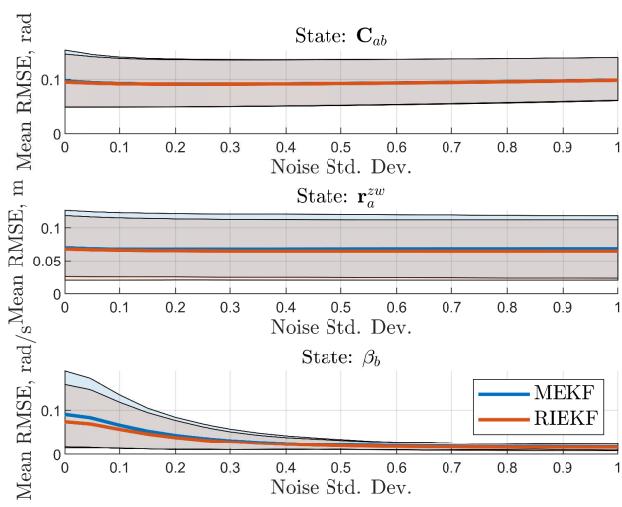
and $\mathbf{L} = \mathbf{1}$, while the measurement Jacobians are

$$\mathbf{H}_k = \underset{i=1, \dots, m}{\text{row}} \left(\begin{bmatrix} \left(\check{\mathbf{C}}_{ab_k}^\top \left(\mathbf{r}_a^{p_i w} - \check{\mathbf{r}}_a^{z_k^i w} \right) \right)^\times & -\check{\mathbf{C}}_{ab_k}^\top & \mathbf{0} \end{bmatrix} \right)$$

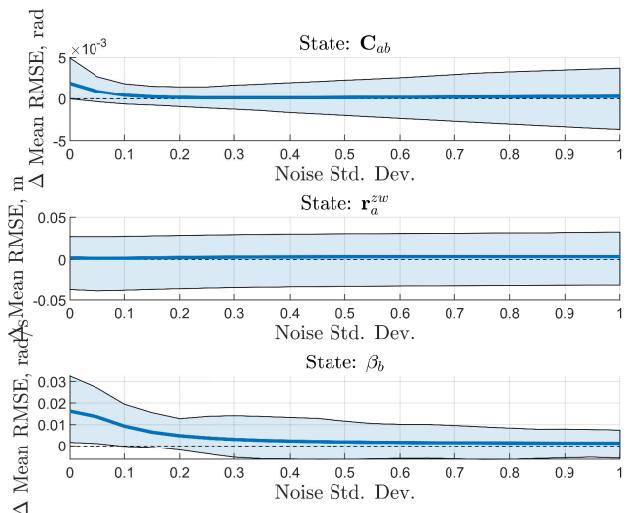
and $\mathbf{M}_k = \mathbf{1}$.

4.3.2 Simulation Results

The MEKF and RIEKF are compared in simulation. The noise standard deviations are the same as those in Table 4.2. In addition, the noise in the bias random walk is assumed isotropic, $\mathbf{Q}_k^3 = \sigma_k^3 \mathbf{1}$ and is set to $\sigma_k^3 = 0.005 \mathbf{1}$ rad/s. The error in the initial conditions were drawn from a zero-mean normal distribution with covariance $\mathbf{P}_0 = \text{diag}(0.1^2 \mathbf{1}, (\frac{\pi}{4})^2 \mathbf{1}, 0.01^2 \mathbf{1})$. The results obtained from varying the noise in the gyro, velocity sensors, landmark sensor and all the sensors simultaneously are shown in Figures 4.16 to 4.19. On average, the RIEKF outperforms the MEKF, but the improvement is not as stark as when bias wasn't being estimated. This is due to the states now appearing in the process model Jacobian. However, as the measurement model Jacobians are independent of the state estimate, there is still an advantage to using the RIEKF.

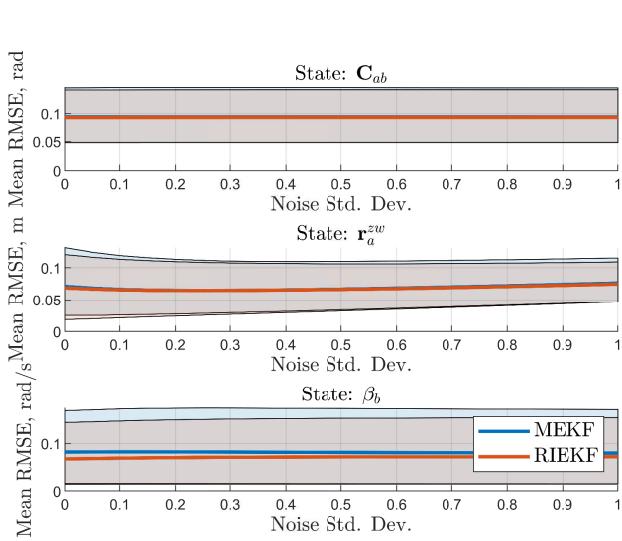


(a) Mean RMSE for each state.

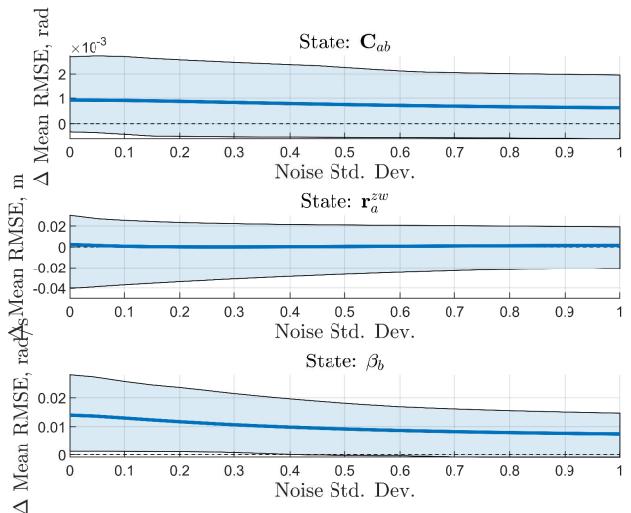


(b) Difference in mean RMSE of MEKF and RIEKF.

Figure 4.16: Results of 50 Monte Carlo trials comparing the MEKF and RIEKF, where the noise in the rate gyro was varied.

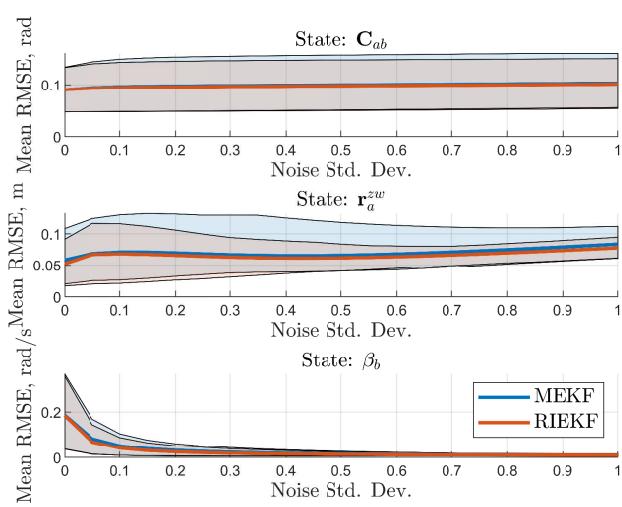


(a) Mean RMSE for each state.

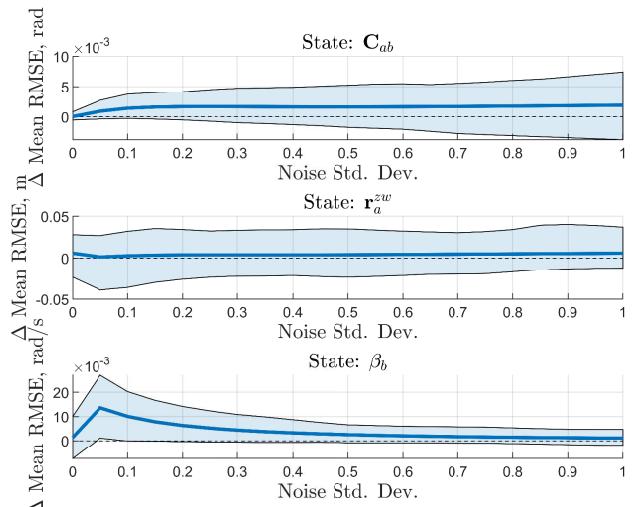


(b) Difference in mean RMSE of MEKF and RIEKF.

Figure 4.17: Results of 50 Monte Carlo trials comparing the MEKF and RIEKF, where the noise in the velocity sensor was varied.

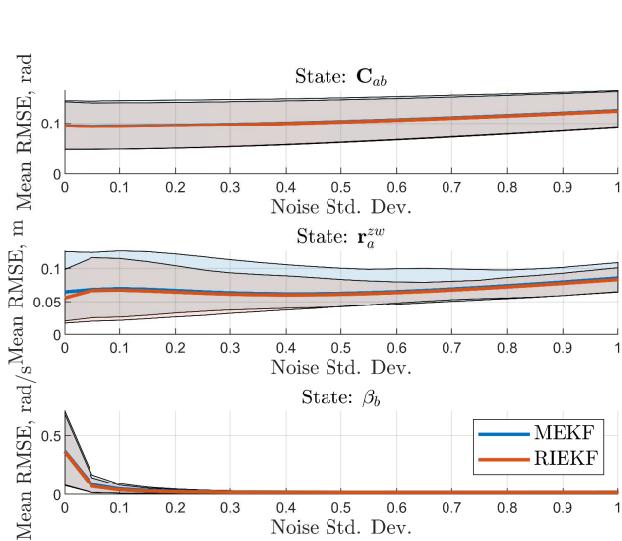


(a) Mean RMSE for each state.

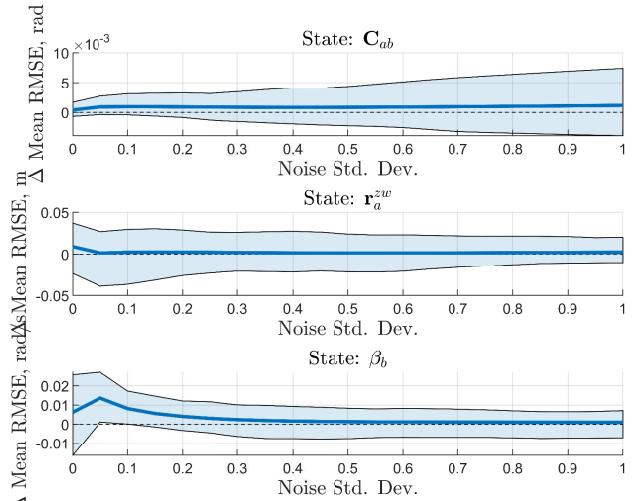


(b) Difference in mean RMSE of MEKF and RIEKF.

Figure 4.18: Results of 50 Monte Carlo trials comparing the MEKF and RIEKF, where the noise in the landmark sensor was varied.



(a) Mean RMSE for each state.



(b) Difference in mean RMSE of MEKF and RIEKF.

Figure 4.19: Results of 50 Monte Carlo trials comparing the MEKF and RIEKF, where the noise in all the sensors varied.

4.4 Conclusion

In the introduction to this chapter, a set of questions were presented to guide an in-depth discussion on the practical implications of the IEKF. It is useful to revisit these questions at the end of this chapter.

1. *Given a left-invariant measurement model, is it more advantageous to use a LIEKF than an MEKF, and given a right-invariant measurement model, is it more advantageous to use a RIEKF than an MEKF?*
2. *What is the effect of process and measurement noise? How quickly do the convergence properties of the IEKF claimed in [1, 2] break down?*

The first two questions posed regarded the performance of the IEKF relative to a standard MEKF. In general, over all the experimental trials, the IEKF outperformed the MEKF. However, a more in-depth analysis revealed that much of the improvement could be attributed to better performance in the transient, as it is during this time that the Jacobians in the IEKF are much more accurate than those of the MEKF. During these trials, the effect of sensor noise was also isolated. Despite the convergence properties holding only when noise is neglected, the IEKF is still more effective than the MEKF in the presence of noise.

3. *How can the IEKF be used in situations when the measurement model isn't left or right-invariant?*

A realistic stereo camera measurement model was used to demonstrate how the IEKF can be used when the measurement model is neither left nor right-invariant. In this scenario, the RIEKF and MEKF provided similar performance, due to the fact that the measurement model was not exactly right-invariant. This was seen both when modifying the IEKF to use the raw measurements, and preprocessing the measurements to create right-invariant pseudomeasurements.

4. *How can bias estimation be performed in an invariant framework?*

Lastly, bias estimation was performed in the invariant framework, using results from [9], [26] and [29]. The RIEKF provided better performance than the MEKF, but to a lesser degree than previously shown due to the process model no longer being group affine.

Chapter 5

Invariant Batch SLAM

In Chapters 3 and 4 of this thesis, the IEKF was presented and its practical implications were explored. The IEKF itself relies on three core concepts, namely invariant measurement models, invariant error, and group-affine process models.. When a left-invariant error definition is used in a situation with group-affine kinematics and a left-invariant measurement model, the remarkable properties resulting from the state independence of the Jacobians are recovered. The same can be said for a right-invariant error and a right-invariant measurement model. The properties, however, were only leveraged in the filtering case. In this chapter, batch estimation in an invariant framework is examined. Invariant batch estimation was first done in [20]. It was then applied to visual-inertial odometry in [32] using inertial measurement unit (IMU) preintegration. Here, the landmark-based batch SLAM problem is addressed. This problem differs from the previous work because the landmark positions are being estimated. The chapter is structured as follows. First, the problem is presented, and a standard solution for matrix Lie groups is shown. Next, the invariant framework is applied to the landmark-based batch SLAM solution. Lastly, a sample problem is presented along with simulation results.

5.1 Simultaneous Localization and Mapping on Matrix Lie Groups

The states of a robotic system are given by $\mathbf{X} \in \mathcal{G}$. Typically, only the pose and landmark locations are estimated. Therefore, the estimated states are typically elements of $SE(2)$ or $SE(3)$. However, the theory is applicable to other matrix Lie groups, as is shown here. The location of the j^{th} landmark is typically $\mathbf{r}_a^{p_j w}$, where $\underline{\mathcal{F}}_a$ is some global reference frame, w is a reference point and p_j is the point associated with the j^{th} landmark. Here, to be more concise, they are denoted $\mathbf{p}^j \in \mathbb{R}^{n_y}$, $j = 1, \dots, m$. Estimating the landmark locations and

the robots' states at each time step means estimating

$$\mathbf{x} = \{\mathbf{X}_0, \dots, \mathbf{X}_n, \mathbf{p}^1, \dots, \mathbf{p}^m\},$$

where $t_k \in [t_0, t_n]$. The discrete-time kinematics are given by

$$\mathbf{X}_k = \mathbf{F}_{k-1}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}), \quad (5.1)$$

where \mathbf{u}_{k-1} are the interoceptive measurements, or inputs, and $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ is zero-mean white noise. The exteroceptive measurements are

$$\mathbf{y} = \{\mathbf{y}_{1,1}, \dots, \mathbf{y}_{1,m}, \dots, \mathbf{y}_{n,1}, \dots, \mathbf{y}_{n,m}\},$$

where $\mathbf{y}_{k,j}$ represents a measurement of the j^{th} landmark at time t_k . The measurements are modelled as

$$\mathbf{y}_{k,j} = \mathbf{g}_k(\mathbf{X}_k, \mathbf{p}^j) + \boldsymbol{\nu}_{k,j}$$

where $\boldsymbol{\nu}_{k,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k,j})$. A batch maximum *a posteriori* method is used to solve the SLAM problem.

The SLAM solution is initialized by specifying $\check{\mathbf{X}}_0$, the best guess of the initial state. The error in the initial state is then $\mathbf{E}_{u,0}(\mathbf{x})$. In [3], a left-invariant error definition is used,

$$\exp\left(\mathbf{e}_{u,0}^L\right) = \mathbf{E}_{u,0}^L(\mathbf{x}) = \mathbf{X}_0^{-1}\check{\mathbf{X}}_0.$$

However, a right-invariant error

$$\exp\left(\mathbf{e}_{u,0}^R\right) = \mathbf{E}_{u,0}^R(\mathbf{x}) = \check{\mathbf{X}}_0\mathbf{X}_0^{-1}$$

can also be used.

Define the error due to the input to be $\mathbf{E}_{u,k}(\mathbf{x}) \in \mathcal{G}$. This error can either be left or right invariant. The left-invariant input error is

$$\exp\left(\mathbf{e}_{u,k}^L\right) = \mathbf{E}_{u,k}^L(\mathbf{x}) = \bar{\mathbf{X}}_k^{-1}\mathbf{F}_{k-1}(\bar{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}), \quad (5.2)$$

whereas the right-invariant input error is

$$\exp\left(\mathbf{e}_{u,k}^R\right) = \mathbf{E}_{u,k}^R(\mathbf{x}) = \mathbf{F}_{k-1}(\bar{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0})\bar{\mathbf{X}}_k^{-1}. \quad (5.3)$$

Here, $\bar{\mathbf{X}}_k$ is the nominal state. The measurement error $\mathbf{e}_{y,k,j}(\mathbf{x})$ is typically defined as

$$\mathbf{e}_{y,k,j}(\mathbf{x}) = \mathbf{y}_{k,j} - \mathbf{g}_k(\mathbf{X}_k, \mathbf{p}^j). \quad (5.4)$$

Follwing [3, pp. 127-143], the SLAM problem is to minimize

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{e}_{u,0}(\mathbf{x})^\top \mathbf{W}_{u,0}^{-1} \mathbf{e}_{u,0}(\mathbf{x}) + \frac{1}{2} \sum_{k=1}^n \mathbf{e}_{u,k}(\mathbf{x})^\top \mathbf{W}_{u,k}^{-1} \mathbf{e}_{u,k}(\mathbf{x}) + \frac{1}{2} \sum_{k=1}^n \sum_{j=1}^m \mathbf{e}_{y,k,j}(\mathbf{x})^\top \mathbf{W}_{y,k,j}^{-1} \mathbf{e}_{y,k,j}(\mathbf{x}),$$

where $\mathbf{W}_{u,0}$, $\mathbf{W}_{u,k}$, and $\mathbf{W}_{y,k,j}$ are weighting matrices related to the probability distribution of the error. The error is deemed Gaussian, as it is assumed that it arises only due to the presence of Gaussian noise in the measurements. Further defining

$$\mathbf{e}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_{u,0}(\mathbf{x}) \\ \vdots \\ \mathbf{e}_{u,n}(\mathbf{x}) \\ \hline \mathbf{e}_{y,1,1}(\mathbf{x}) \\ \mathbf{e}_{y,1,2}(\mathbf{x}) \\ \vdots \\ \mathbf{e}_{y,n,m}(\mathbf{x}) \end{bmatrix},$$

and $\mathbf{W}_u = \text{diag}(\mathbf{W}_{u,0}, \mathbf{W}_{u,1}, \dots, \mathbf{W}_{u,n})$, $\mathbf{W}_y = \text{diag}(\mathbf{W}_{y,1,1}, \mathbf{W}_{y,1,2}, \mathbf{W}_{y,n,m})$ and $\mathbf{W} = \text{diag}(\mathbf{W}_u, \mathbf{W}_y)$, the objective function is rewritten as

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{e}(\mathbf{x})^\top \mathbf{W}^{-1} \mathbf{e}(\mathbf{x}).$$

To minimize this objective function, the errors are linearized about an operating point \mathbf{x}^{op} . The linearization is to be done with an uncertainty representation consistent with the choice of error. If a left-invariant error is chosen, the left-invariant uncertainty representation (2.5) should be used. If a right-invariant error is chosen, the right-invariant uncertainty representation (2.6) should be used. Note that the alternative uncertainty representations given in (2.3) and (2.4) could also be used. Doing so would simply lead to a change in sign on various Jacobians that, once incorporated into the nonlinear least squares problem, are inconsequential. However, in order to be consistent with the definition of right- or left-invariant error, the uncertainty representations (2.5) or (2.6) should be used. The linearized prior error has the form

$$\mathbf{e}_{u,0}(\mathbf{x}) = \mathbf{e}_{u,0}(\mathbf{x}^{\text{op}}) - \mathbf{F}_0^2 \delta \mathbf{e}_0, \quad (5.5)$$

where $\delta\boldsymbol{\epsilon}_k$ is the error between the operating and truth trajectory. The linearized input error has the form

$$\mathbf{e}_{u,k}(\mathbf{x}) = \mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) + \mathbf{F}_k^1 \delta\boldsymbol{\epsilon}_{k-1} - \mathbf{F}_k^2 \delta\boldsymbol{\epsilon}_k. \quad (5.6)$$

The linearized measurement error has the form

$$\mathbf{e}_{y,k,j}(\mathbf{x}) = \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) - \mathbf{H}_{k,j}^1 \delta\boldsymbol{\epsilon}_k - \mathbf{H}_{k,j}^2 \delta\boldsymbol{\zeta}_j, \quad (5.7)$$

where $\delta\boldsymbol{\zeta}_j$ is the error between the operating position estimate and the true position of the j^{th} landmark. Various methods of obtaining (5.5), (5.6), and (5.7) are demonstrated in Section 5.3. By stacking the estimation errors

$$\delta\mathbf{x} = \begin{bmatrix} \delta\boldsymbol{\epsilon}_0 \\ \vdots \\ \frac{\delta\boldsymbol{\epsilon}_n}{\delta\boldsymbol{\zeta}_1} \\ \vdots \\ \delta\boldsymbol{\zeta}_m \end{bmatrix},$$

the linearized system can then be written

$$\mathbf{e}(\mathbf{x}) = \mathbf{e}(\mathbf{x}^{\text{op}}) - \boldsymbol{\Gamma} \delta\mathbf{x},$$

where

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{H}^1 & \mathbf{H}^2 \end{bmatrix},$$

where

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{F}_0^2 & & \\ \mathbf{F}_1^1 & \mathbf{F}_1^2 & \\ & \ddots & \ddots \\ & & \mathbf{F}_{n-1}^1 & \mathbf{F}_{n-1}^2 \end{bmatrix},$$

$$\mathbf{H}^1 = \begin{bmatrix} \mathbf{H}_{1,1}^1 & & & \\ & \vdots & & \\ & \mathbf{H}_{1,m}^1 & & \\ & & \mathbf{H}_{2,1}^1 & \\ & & & \vdots \\ & & \mathbf{H}_{2,m}^1 & \\ & & & \ddots & \\ & & & & \mathbf{H}_{n,1}^1 \\ & & & & & \vdots \\ & & & & & \mathbf{H}_{n,m}^1 \end{bmatrix},$$

and

$$\mathbf{H}_2 = \begin{bmatrix} \mathbf{H}_{1,1}^2 & & & \\ & \ddots & & \\ & & \mathbf{H}_{1,1}^2 & \\ & & & \vdots \\ & & & \mathbf{H}_{2,1}^2 \\ & & & & \vdots \\ & & & & \mathbf{H}_{2,m}^2 \\ & & & & & \ddots \\ & & & & & & \mathbf{H}_{n,1}^2 \\ & & & & & & & \vdots \\ & & & & & & & \mathbf{H}_{n,m}^2 \end{bmatrix}.$$

The linearized objective function is

$$J(\mathbf{x}^{\text{op}} + \delta\mathbf{x}) = (\mathbf{e}(\mathbf{x}^{\text{op}}) + \boldsymbol{\Gamma}\delta\mathbf{x})^\top \mathbf{W}^{-1}(\mathbf{e}(\mathbf{x}^{\text{op}}) + \boldsymbol{\Gamma}\delta\mathbf{x}). \quad (5.8)$$

Minimizing (5.8) with respect to $\delta\mathbf{x}$ yields the Gauss-Newton update,

$$(\boldsymbol{\Gamma}^\top \mathbf{W}^{-1} \boldsymbol{\Gamma}) \delta\mathbf{x} = \boldsymbol{\Gamma}^\top \mathbf{W}^{-1} \mathbf{e}(\mathbf{x}^{\text{op}}),$$

which can be iteratively solved for the minimizing solution $\delta\mathbf{x}^*$. The minimizing solution is

composed of the state update at every time step, $\delta\mathbf{x}_k^*$ and the landmark updates $\delta\mathbf{x}_j^*$.

The operating point is updated using the proper uncertainty representation. If a right-invariant error is used, the state at each time step is updated using

Edited
2020/5/1

$$\mathbf{X}_k = \exp(-\delta\mathbf{x}_k^{*\wedge}) \mathbf{X}_k^{\text{op}}. \quad (5.9)$$

If a left-invariant error is used

$$\mathbf{X}_k = \mathbf{X}_k^{\text{op}} \exp(-\delta\mathbf{x}_k^{*\wedge}). \quad (5.10)$$

The landmark position estimate is updated using

$$\mathbf{p}^j = \mathbf{p}^{j\text{op}} + \delta\mathbf{x}_j^*,$$

5.2 Leveraging the Invariant Framework in Batch SLAM

As was seen in Chapters 3 and 4, a left-invariant error definition, used when the kinematics are group-affine and the measurement model is left-invariant leads to state-independent Jacobians. This is also the case for a right-invariant error definition and right-invariant measurement model. In SLAM, it is typical to have measurement models that are right-invariant, as only information relative to the body of interest is available. Despite this, both the left and right-invariant cases will be considered. In both cases, if the landmark positions were known *a priori*, it would be possible to obtain state-independent Jacobians. However, as the landmark positions are unknown, the best possible result would be to obtain a Jacobian dependent on only the estimated landmark position. This is seen in Section 5.3.

5.2.1 Left-Invariant Error

Assume the discrete-time kinematics (5.1) are group-affine, meaning they satisfy (3.16). The left-invariant error is $\delta\mathbf{X} = \mathbf{X}^{-1}\bar{\mathbf{X}}$. The left-invariant input error remains defined as (5.2). The main difference to consider is the measurement error is now defined

$$\mathbf{e}_{y,k,j} = \bar{\mathbf{X}}_k^{-1} (\mathbf{y}_{k,j} - \mathbf{g}_k(\mathbf{X}_k, \mathbf{p}^j)).$$

Edited
2021/2/17

5.2.2 Right-Invariant Error

Assume the discrete-time kinematics (5.1) are group-affine, meaning they satisfy (3.16). The right-invariant error is $\delta\mathbf{X} = \bar{\mathbf{X}}\mathbf{X}^{-1}$. The right-invariant input error remains defined as (5.2).

The main difference to consider is the measurement error is now defined

$$\mathbf{e}_{y,k,j} = \bar{\mathbf{X}}_k (\mathbf{y}_{k,j} - \mathbf{g}_k(\mathbf{X}_k, \mathbf{p}^j)).$$

Edited
2021/2/17

5.3 Sample Problem 3: Inertial Navigation with Bias

Consider a body moving in 3D space, equipped with an accelerometer, a rate gyro and some sensor that provides relative landmark locations. The problem setup is illustrated in Figure 4.2. Here, $\underline{\mathcal{F}}_a$ is an inertial frame. The discrete-time kinematics are

$$\begin{aligned}\mathbf{C}_{ab_k} &= \mathbf{C}_{ab_{k-1}} \exp_{SO(3)}\left(T\boldsymbol{\omega}_{b_{k-1}}^{b_{k-1}a^\times}\right), \\ \mathbf{v}_a^{z_k w/a} &= \mathbf{v}_a^{z_{k-1} w/a} + T(\mathbf{C}_{ab_{k-1}} \mathbf{f}_{b_{k-1}} + \mathbf{g}_a), \\ \mathbf{r}_a^{z_k w} &= \mathbf{r}_a^{z_{k-1} w} + T\mathbf{v}_a^{z_{k-1} w/a},\end{aligned}$$

where \mathbf{f}_b are the specific body forces resolved in \mathcal{F}_b and \mathbf{g}_a is the gravity resolved in \mathcal{F}_a . The rate gyro model is

$$\mathbf{u}_{b_k}^1 = \boldsymbol{\omega}_{b_k}^{b_k a} - \boldsymbol{\beta}_{b_k}^1 - \mathbf{w}_{b_k}^1,$$

where $\mathbf{w}_{b_k}^1 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k^1)$. The bias in the rate gyro is modelled as a random walk, $\dot{\boldsymbol{\beta}}_{b_k}^1 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k^3)$. The accelerometer model is

$$\mathbf{u}_{b_k}^2 = \mathbf{f}_{b_k} - \boldsymbol{\beta}_{b_k}^2 - \mathbf{w}_{b_k}^2,$$

where $\mathbf{w}_{b_k}^2 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k^2)$. The bias in the accelerometer is modelled as a random walk, $\dot{\boldsymbol{\beta}}_{b_k}^2 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k^4)$. The landmark sensor model is

$$\mathbf{y}_k^j = \mathbf{C}_{ab_k}^\top (\mathbf{r}_a^{p_j a} - \mathbf{r}_a^{z_k a}) + \boldsymbol{\nu}_k^j,$$

where $\boldsymbol{\nu}_k^j \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k,j})$. This is not a realistic measurement model, but is used here to simplify the problem. There may be advantages to this, as was shown in Chapter 4. To ease the notation, let $\mathbf{C}_k = \mathbf{C}_{ab_k}$, $\mathbf{r}_k = \mathbf{r}_a^{z_k w}$, $\mathbf{v}_k = \mathbf{v}_a^{z_k w/a}$, $\mathbf{u}_k^i = \mathbf{u}_{b_k}^i$, $\boldsymbol{\beta}_k^i = \boldsymbol{\beta}_{b_k}^i$, $\mathbf{w}_k^i = \mathbf{w}_{b_k}^i$ and $\mathbf{r}_a^{p_j w} = \mathbf{p}^j$.

5.3.1 Matrix Lie Group

Consider the matrix Lie group \mathcal{G}_2 , where

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{C}_k & \mathbf{v}_k & \mathbf{r}_k \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \beta_k^1 & \beta_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}.$$

The inverse of an element of \mathcal{G}_2 is

$$\mathbf{X}_k^{-1} = \begin{bmatrix} \mathbf{C}_k^\top & -\mathbf{C}_k^\top \mathbf{v}_k & -\mathbf{C}_k \mathbf{r}_k \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & -\beta_k^1 & -\beta_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}.$$

Let \mathfrak{g}_2 be the Lie algebra of \mathcal{G}_2 . The column matrix $\xi \in \mathbb{R}^{15}$ is mapped to \mathfrak{g}_2 using

$$\xi^\wedge = \begin{bmatrix} \xi^\phi \\ \xi^v \\ \xi^r \\ \xi^1 \\ \xi^2 \end{bmatrix}^\wedge = \begin{bmatrix} \xi^{\phi \times} & \xi^v & \xi^r \\ 0 & 0 & \\ & \mathbf{0} & \xi^1 & \xi^2 \\ & & 0 & \\ & & & 0 \end{bmatrix}.$$

The exponential map from \mathfrak{g}_2 to \mathcal{G}_2 is

$$\exp(\xi^\wedge) = \begin{bmatrix} \exp_{SO(3)}(\xi^{\phi \times}) & \mathbf{J}\xi^v & \mathbf{J}\xi^r \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \xi^1 & \xi^2 \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix},$$

where \mathbf{J} is given by (2.7). The adjoint representation of an element of \mathcal{G}_2 is

$$\text{Ad}(\mathbf{X}_k) = \begin{bmatrix} \mathbf{C}_k & & & \\ \mathbf{v}_k^\times \mathbf{C}_k & \mathbf{C}_k & & \\ \mathbf{r}_k^\times \mathbf{C}_k & & \mathbf{C}_k & \\ & & & \mathbf{1} \\ & & & & \mathbf{1} \end{bmatrix}.$$

The discrete-time kinematics are

$$\mathbf{X}_k = \mathbf{F}_{k-1}(\mathbf{X}_{k-1}) \boldsymbol{\Xi}_{k-1}^{\text{noisy}},$$

where

$$\mathbf{F}_{k-1}(\mathbf{X}_{k-1}) = \begin{bmatrix} \mathbf{C}_{k-1} & \mathbf{v}_{k-1} + T\mathbf{g} & \mathbf{r}_{k-1} + T\mathbf{v}_{k-1} & & \\ & 1 & & & \\ & & 1 & & \\ & & & \mathbf{1} & \beta_k^1 \quad \beta_k^2 \\ & & & & 1 \\ & & & & & 1 \end{bmatrix}$$

and

$$\boldsymbol{\Xi}_{k-1}^{\text{noisy}} = \begin{bmatrix} \exp_{SO(3)}(T(\mathbf{u}_{k-1}^1 + \boldsymbol{\beta}_{k-1}^1 + \mathbf{w}_{k-1}^1)^\times) & T(\mathbf{u}_{k-1}^2 + \boldsymbol{\beta}_{k-1}^2 + \mathbf{w}_{k-1}^2) & \mathbf{0} & & \\ & 1 & & & \\ & & 1 & & \\ & & & \mathbf{1} & \mathbf{w}_{k-1}^3 \quad \mathbf{w}_{k-1}^4 \\ & & & & 1 \\ & & & & & 1 \end{bmatrix}.$$

This will be approximated as

$$\mathbf{X}_k = \mathbf{F}_{k-1}(\mathbf{X}_{k-1}) \boldsymbol{\Xi}_{k-1} \exp(\mathbf{w}_{k-1}^\wedge),$$

where $\mathbf{w}_{k-1} = [\mathbf{w}_{k-1}^1 \top \mathbf{w}_{k-1}^2 \top \mathbf{0} \mathbf{w}_{k-1}^3 \top \mathbf{w}_{k-1}^4]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$,

$$\boldsymbol{\Xi}_{k-1} = \begin{bmatrix} \boldsymbol{\Psi}_{k-1} & T(\mathbf{u}_{k-1}^2 + \boldsymbol{\beta}_{k-1}^2) & \mathbf{0} \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \\ & & & 1 \\ & & & & 1 \end{bmatrix},$$

and $\boldsymbol{\Psi}_{k-1} = \exp_{SO(3)}(T(\mathbf{u}_{k-1}^1 + \boldsymbol{\beta}_{k-1}^1)^\times)$. The noise covariance is $\mathbf{Q}_{k-1} = \text{diag}(\mathbf{Q}_{k-1}^1, \mathbf{Q}_{k-1}^2, \mathbf{0}, \mathbf{Q}_{k-1}^3, \mathbf{Q}_{k-1}^4)$. The measurement model is

$$\mathbf{y}_{k,j} = \mathbf{C}_k^\top(\mathbf{p}^j - \mathbf{r}_k) + \boldsymbol{\nu}_{k,j},$$

where $\boldsymbol{\nu}_{k,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k,j})$. It can be represented using an element of \mathcal{G}_2 as

$$\tilde{\mathbf{y}}_{k,j} = \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} + \tilde{\boldsymbol{\nu}}_{k,j}, \quad (5.11)$$

where $\tilde{\mathbf{y}}_{k,j} = [\mathbf{y}_{k,j}^\top \ 0 \ 1 \ \mathbf{0}]^\top$ and $\tilde{\boldsymbol{\nu}}_{k,j} = [\boldsymbol{\nu}_{k,j}^\top \ \mathbf{0}]^\top$.

Six different approaches are compared. First, a standard left-invariant error definition is used, as the one described in Section 5.1. The left-invariant error derivation is then changed in a way that leads to Jacobians dependent only on the measurement and not on the state. Next, the right-invariant error definition is used. Lastly, these three aforementioned approaches are repeated, using the appropriate invariant measurement error as shown in Section 5.2. Tables 5.1 and 5.2 summarize the various approaches.

App.	$\mathbf{E}_{u,k}$	$\mathbf{W}_{u,k}$	\mathbf{F}_k^1	\mathbf{F}_k^2
1	$\mathbf{X}_k^{-1} \mathbf{F}_{k-1} \Xi_{k-1}$	\mathbf{Q}_k	$\text{Ad}(\mathbf{X}_k^{\text{op}} \mathbf{F}_{k-1}^{\text{op}}) \mathbf{B}^L + \mathbf{D}_{k-1}^L$	$-\mathbf{1}$
2	$\mathbf{X}_k^{-1} \mathbf{F}_{k-1} \Xi_{k-1}$	\mathbf{Q}_k	$\text{Ad}(\Xi_{k-1}^{\text{op}}) \mathbf{B}^L + \mathbf{D}_{k-1}^L$	$-\mathbf{1}$
3	$\mathbf{F}_{k-1} \Xi_{k-1} \mathbf{X}_k^{-1}$	$\text{Ad}(\bar{\mathbf{F}}_{k-1} \Xi_{k-1}) \mathbf{Q}_k \text{Ad}(\bar{\mathbf{F}}_{k-1} \Xi_{k-1})^\top$	$\mathbf{B}^R + \text{Ad}(\mathbf{F}_{k-1}^{\text{op}}) \mathbf{D}_{k-1}^R$	$-\mathbf{1}$
4	$\mathbf{X}_k^{-1} \mathbf{F}_{k-1} \Xi_{k-1}$	\mathbf{Q}_k	$\text{Ad}(\mathbf{X}_k^{\text{op}} \mathbf{F}_{k-1}^{\text{op}}) \mathbf{B}^L + \mathbf{D}_{k-1}^L$	$-\mathbf{1}$
5	$\mathbf{X}_k^{-1} \mathbf{F}_{k-1} \Xi_{k-1}$	\mathbf{Q}_k	$\text{Ad}(\Xi_{k-1}^{\text{op}}) \mathbf{B}^L + \mathbf{D}_{k-1}^L$	$-\mathbf{1}$
6	$\mathbf{F}_{k-1} \Xi_{k-1} \mathbf{X}_k^{-1}$	$\text{Ad}(\bar{\mathbf{F}}_{k-1} \Xi_{k-1}) \mathbf{Q}_k \text{Ad}(\bar{\mathbf{F}}_{k-1} \Xi_{k-1})^\top$	$\mathbf{B}^R + \text{Ad}(\mathbf{F}_{k-1}^{\text{op}}) \mathbf{D}_{k-1}^R$	$-\mathbf{1}$

Table 5.1: Summary of input errors and Jacobians for the 6 different approaches. Approach 1 uses the standard left-invariant approach with the type 1 Jacobians, while Approach 2 uses the type 2 Jacobians. Approach 3 uses the right-invariant error. Approaches 4, 5, and 6 replicate Approaches 1, 2, and 3, substituting the regular measurement error for an invariant measurement error.

App.	$\tilde{\mathbf{e}}_{y,k,j}$	$\mathbf{W}_{y,k,j}$	$\mathbf{H}_{k,j}^1$	$\mathbf{H}_{k,j}^2$
1	$\tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{-1} \tilde{\mathbf{p}}^j$	$\mathbf{R}_{k,j}$	$- \left[\begin{array}{c cccc} \left(\mathbf{C}_k^{\text{op}} \mathbf{p}^{j\text{op}} - \mathbf{r}_k^{\text{op}} \right)^\times & \mathbf{0} & -\mathbf{1} & \mathbf{0} & \mathbf{0} \end{array} \right]$	$\mathbf{C}_k^{\text{op}} \mathbf{T}$
2	$\tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{-1} \tilde{\mathbf{p}}^j$	$\mathbf{R}_{k,j}$	$- \left[\begin{array}{c cccc} \left(\mathbf{C}_k^{\text{op}} \mathbf{p}^{j\text{op}} - \mathbf{r}_k^{\text{op}} \right)^\times & \mathbf{0} & -\mathbf{1} & \mathbf{0} & \mathbf{0} \end{array} \right]$	$\mathbf{C}_k^{\text{op}} \mathbf{T}$
3	$\tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{-1} \tilde{\mathbf{p}}^j$	$\mathbf{R}_{k,j}$	$- \left[\begin{array}{c cccc} \mathbf{C}_k^{\text{op}} \mathbf{p}^{j\text{op}} \times & \mathbf{0} & -\mathbf{C}_k^{\text{op}} \mathbf{T} & \mathbf{0} & \mathbf{0} \end{array} \right]$	$\mathbf{C}_k^{\text{op}} \mathbf{T}$
4	$\bar{\mathbf{X}}_k (\tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{-1} \tilde{\mathbf{p}}^j)$	$\bar{\mathbf{C}}_k \mathbf{R}_{k,j} \bar{\mathbf{C}}_k^\top$	$- \left[\begin{array}{c cccc} (\mathbf{p}^{j\text{op}} - \mathbf{r}_k^{\text{op}}) \times \mathbf{C}_k^{\text{op}} & \mathbf{0} & -\mathbf{C}_k^{\text{op}} & \mathbf{0} & \mathbf{0} \end{array} \right]$	$\mathbf{1}$
5	$\bar{\mathbf{X}}_k (\tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{-1} \tilde{\mathbf{p}}^j)$	$\bar{\mathbf{C}}_k \mathbf{R}_{k,j} \bar{\mathbf{C}}_k^\top$	$- \left[\begin{array}{c cccc} (\mathbf{p}^{j\text{op}} - \mathbf{r}_k^{\text{op}}) \times \mathbf{C}_k^{\text{op}} & \mathbf{0} & -\mathbf{C}_k^{\text{op}} & \mathbf{0} & \mathbf{0} \end{array} \right]$	$\mathbf{1}$
6	$\bar{\mathbf{X}}_k^{-1} (\tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{-1} \tilde{\mathbf{p}}^j)$	$\bar{\mathbf{C}}_k^\top \mathbf{R}_{k,j} \bar{\mathbf{C}}_k$	$- \left[\begin{array}{c cccc} \mathbf{p}^{j\text{op}} \times & \mathbf{0} & -\mathbf{1} & \mathbf{0} & \mathbf{0} \end{array} \right]$	$\mathbf{1}$

Table 5.2: Summary of input errors and Jacobians for the 6 different approaches. Approach 1 uses the standard left-invariant approach with the type 1 Jacobians, while Approach 2 uses the type 2 Jacobians. Approach 3 uses the right-invariant error. Approaches 4, 5, and 6 replicate Approaches 1, 2, and 3, substituting the regular measurement error for an invariant measurement error.

5.3.2 Approach 1: Left-Invariant Error Definition with Type 1 Jacobians

Let the error be defined as $\delta\mathbf{X} = \mathbf{X}^{-1}\bar{\mathbf{X}}$. The uncertainty representation is $\mathbf{X} = \bar{\mathbf{X}} \exp(-\delta\xi^\wedge)$.

Therefore the state is updated using (5.10). The left-invariant error in the state is

$$\begin{aligned}\mathbf{X}_k^{-1}\bar{\mathbf{X}}_k &= \begin{bmatrix} \mathbf{C}_k^\top & -\mathbf{C}_k^\top \mathbf{v}_k & -\mathbf{C}_k \mathbf{r}_k \\ & 1 & \\ & & 1 \\ & & \mathbf{1} & -\beta_k^1 & -\beta_k^2 \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{C}}_k & \bar{\mathbf{v}}_k & \bar{\mathbf{r}}_k \\ & 1 & \\ & & 1 \\ & & \mathbf{1} & \bar{\beta}_k^1 & \bar{\beta}_k^2 \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{C}_k^\top \bar{\mathbf{C}}_k & \mathbf{C}_k^\top (\bar{\mathbf{v}}_k - \mathbf{v}_k) & \mathbf{C}_k^\top (\bar{\mathbf{r}}_k - \mathbf{r}_k) \\ & 1 & \\ & & 1 \\ & & \mathbf{1} & \bar{\beta}_k^1 - \beta_k^1 & \bar{\beta}_k^2 - \beta_k^2 \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \\ &= \begin{bmatrix} \delta\mathbf{C}_k & \delta\mathbf{v}_k & \delta\mathbf{r}_k \\ & 1 & \\ & & 1 \\ & & \mathbf{1} & \delta\beta_k^1 & \delta\beta_k^2 \\ & & & 1 & \\ & & & & 1 \end{bmatrix}.\end{aligned}$$

Thus, the perturbation in \mathbb{R}^{15} is defined as

$$\delta\xi_k = \begin{bmatrix} \log_{SO(3)}(\delta\mathbf{C}_k)^\vee \\ \mathbf{J}_k^{-1}\delta\mathbf{v}_k \\ \mathbf{J}_k^{-1}\delta\mathbf{r}_k \\ \delta\beta_k^1 \\ \delta\beta_k^2 \end{bmatrix}.$$

The left-invariant error in $\mathbf{F}_k(\mathbf{X}_k)$ is

$$\begin{aligned}
\delta \mathbf{F}_k &= \mathbf{F}_k(\mathbf{X}_k)^{-1} \mathbf{F}_k(\bar{\mathbf{X}}_k) \\
&= \left[\begin{array}{ccc} \mathbf{C}_k^\top & -\mathbf{C}_k^\top(\mathbf{v}_k + T\mathbf{g}) & -\mathbf{C}_k^\top(\mathbf{r}_k + T\mathbf{v}_k) \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & -\boldsymbol{\beta}_k^1 & -\boldsymbol{\beta}_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{array} \right] \\
&\quad \left[\begin{array}{ccc} \bar{\mathbf{C}}_k & \bar{\mathbf{v}}_k + T\mathbf{g} & \bar{\mathbf{r}}_k + T\bar{\mathbf{v}}_k \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \bar{\boldsymbol{\beta}}_k^1 & \bar{\boldsymbol{\beta}}_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{array} \right] \\
&= \left[\begin{array}{ccc} \mathbf{C}_k^\top \bar{\mathbf{C}}_k & \mathbf{C}_k^\top(\bar{\mathbf{v}}_k - \mathbf{v}_k) & \mathbf{C}_k^\top(\bar{\mathbf{r}}_k + T\bar{\mathbf{v}}_k - \mathbf{r}_k - T\mathbf{v}_k) \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \bar{\boldsymbol{\beta}}_k^1 - \boldsymbol{\beta}_k^1 & \bar{\boldsymbol{\beta}}_k^2 - \boldsymbol{\beta}_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{array} \right] \\
&= \left[\begin{array}{ccc} \delta \mathbf{C}_k & \delta \mathbf{v}_k & \mathbf{C}_k^\top(\bar{\mathbf{r}}_k - \mathbf{r}_k) + T\mathbf{C}_k^\top(\bar{\mathbf{v}}_k - \mathbf{v}_k) \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \delta \boldsymbol{\beta}_k^1 & \delta \boldsymbol{\beta}_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{array} \right] \\
&= \left[\begin{array}{ccc} \delta \mathbf{C}_k & \delta \mathbf{v}_k & \delta \mathbf{r}_k + T\delta \mathbf{v}_k \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \delta \boldsymbol{\beta}_k^1 & \delta \boldsymbol{\beta}_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{array} \right]
\end{aligned}$$

As $\delta\mathbf{F}_k \in \mathcal{G}_2$, it can be written using the exponential map as $\delta\mathbf{F}_k = \exp(\delta\mathbf{f}_k^\wedge)$, where

$$\delta\mathbf{f}_k = \begin{bmatrix} \log_{SO(3)}(\delta\mathbf{C}_k)^\vee \\ \mathbf{J}_k^{-1}\delta\mathbf{v}_k \\ \mathbf{J}_k^{-1}(\delta\mathbf{r}_k + T\delta\mathbf{v}_k) \\ \delta\boldsymbol{\beta}_k^1 \\ \delta\boldsymbol{\beta}_k^2 \end{bmatrix}.$$

This can be rewritten as $\delta\mathbf{f}_k = \mathbf{B}^L\delta\xi_k$, where

$$\mathbf{B}^L = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & T\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}.$$

As $\boldsymbol{\Xi}_k \in \mathcal{G}_2$ is state-dependent, it has an associated left-invariant error, which is $\delta\boldsymbol{\Xi}_k =$

$\Xi_k^{-1}\bar{\Xi}_k$. Letting $\bar{\Psi}_k = \exp_{SO(3)}(T(\mathbf{u}_k^1 + \bar{\beta}_k^1)^\times)$, the left-invariant error is

$$\begin{aligned}
\delta\Xi_k &= \Xi_k^{-1}\bar{\Xi}_k \\
&= \begin{bmatrix} \Psi_k^\top & -T\Psi_k^\top(\mathbf{u}_k^2 + \beta_k^2) & \mathbf{0} \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & & 1 \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} \bar{\Psi}_k & T(\mathbf{u}_k^2 + \bar{\beta}_k^2) & \mathbf{0} \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & & 1 \\ & & & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} \Psi_k^\top \bar{\Psi}_k & T\Psi_k^\top(\mathbf{u}_k^2 + \bar{\beta}_k^2) - T\Psi_k^\top(\mathbf{u}_k^2 + \beta_k^2) & \mathbf{0} \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & & 1 \\ & & & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} \Psi_k^\top \bar{\Psi}_k & T\Psi_k^\top(\bar{\beta}_k^2 - \beta_k^2) & \mathbf{0} \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & & 1 \\ & & & & & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} \exp_{SO(3)}(-T(\beta_k^1 - \bar{\beta}_k^1)^\times) & T\Psi_k^\top \delta\beta_k^2 & \mathbf{0} \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & & 1 \\ & & & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} \exp_{SO(3)}(T\delta\beta_k^{1\times}) & T\Psi_k^\top \delta\beta_k^2 & \mathbf{0} \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & & 1 \\ & & & & & 1 \end{bmatrix}.
\end{aligned}$$

To simplify, let $\Psi_k^\top = \exp_{SO(3)}(T\delta\beta_k^{1\times}) \bar{\Psi}_k^\top$, and linearize by letting $\exp_{SO(3)}(T\delta\beta_k^{1\times}) \approx$

$\mathbf{1} + T\delta\beta_k^{1\times}$ and neglecting second order terms. Then,

$$\begin{aligned}\delta\Xi_k &= \begin{bmatrix} \exp_{SO(3)}\left(T\delta\beta_k^{1\times}\right) & T\exp_{SO(3)}\left(T\delta\beta_k^{1\times}\right)\bar{\Psi}_k^T\delta\beta_k^2 & \mathbf{0} \\ & 1 & \\ & & 0 \\ & & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \\ &\approx \begin{bmatrix} \exp_{SO(3)}\left(T\delta\beta_k^{1\times}\right) & T(\mathbf{1} + T\delta\beta_k^{1\times})\bar{\Psi}_k^T\delta\beta_k^2 & \mathbf{0} \\ & 1 & \\ & & 1 \\ & & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \\ &= \begin{bmatrix} \exp_{SO(3)}\left(T\delta\beta_k^{1\times}\right) & T\bar{\Psi}_k^T\delta\beta_k^2 & \mathbf{0} \\ & 1 & \\ & & 1 \\ & & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & 1 & \\ & & & & 1 \end{bmatrix}.\end{aligned}$$

As $\delta\Xi_k \in \mathcal{G}_2$, it can be expressed using the exponential map as $\delta\Xi_k = \exp(\delta\mathbf{u}_k^\wedge)$, where

$$\delta\mathbf{u}_k = \begin{bmatrix} T\delta\beta_k^1 \\ T\mathbf{J}^{-1}\bar{\Psi}_k^T\delta\beta_k^2 \\ \mathbf{0} \end{bmatrix}.$$

This can be rewritten as $\delta\mathbf{u}_k = \mathbf{D}_k^L\delta\xi_k$, where

$$\mathbf{D}_k^L = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & T\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & T\bar{\Psi}_k^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

5.3.2.1 Probability Distribution of the Errors

The prior error probability distribution is found by linearizing $\mathbf{E}_{u,0} = \mathbf{X}_0^{-1}\bar{\mathbf{X}}_0$. Letting $\mathbf{X}_0 = \bar{\mathbf{X}}_0 \exp(-\delta\xi_0^\wedge)$, the prior input error is

$$\begin{aligned}\mathbf{E}_{u,0}(\mathbf{x}) &= \mathbf{X}_0^{-1}\bar{\mathbf{X}}_0, \\ \exp(\mathbf{e}_{u,0}^\wedge) &= \exp(\delta\xi_0^\wedge)\bar{\mathbf{X}}_0^{-1}\bar{\mathbf{X}}_0 \\ &= \exp(\delta\xi_0^\wedge), \\ \mathbf{e}_{u,0}^\wedge &\approx \delta\xi_0^\wedge, \\ \mathbf{e}_{u,0} &= \delta\xi_0.\end{aligned}$$

Thus, $\mathbf{e}_{u,0} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0)$, where \mathbf{P}_0 is the covariance matrix of the initial error, which is assumed to have the form $\mathbf{P}_0 = \text{diag}(\mathbf{P}_0^\phi, \mathbf{P}_0^v, \mathbf{P}_0^r, \mathbf{P}_0^4, \mathbf{P}_0^5)$.

To find the input error probability distribution, the process model is linearized and the effect of noise is recovered. The linearized kinematics are

$$\begin{aligned}\mathbf{X}_k &= \mathbf{F}_{k-1}(\mathbf{X}_{k-1})\Xi_{k-1} \exp(\mathbf{w}_{k-1}^\wedge), \\ \bar{\mathbf{X}}_k \exp(-\delta\xi_k^\wedge) &= \mathbf{F}_{k-1}(\bar{\mathbf{X}}_{k-1}) \exp(-(\mathbf{B}^L \delta\xi_{k-1})^\wedge) \bar{\Xi}_{k-1} \exp(-(\mathbf{D}_{k-1}^L \delta\xi_{k-1})^\wedge) \exp(\mathbf{w}_{k-1}^\wedge), \\ \exp(-\delta\xi_k^\wedge) &= \bar{\mathbf{X}}_k^{-1}\mathbf{F}_{k-1}(\bar{\mathbf{X}}_{k-1}) \exp(-(\mathbf{B}^L \delta\xi_{k-1})^\wedge) \bar{\Xi}_{k-1} \exp(-(\mathbf{D}_{k-1}^L \delta\xi_{k-1})^\wedge) \exp(\mathbf{w}_{k-1}^\wedge) \\ &= \bar{\Xi}_{k-1}^{-1} \exp(-(\mathbf{B}^L \delta\xi_{k-1})^\wedge) \bar{\Xi}_{k-1} \exp(-(\mathbf{D}_{k-1}^L \delta\xi_{k-1})^\wedge) \exp(\mathbf{w}_{k-1}^\wedge).\end{aligned}\quad (5.12)$$

Using (2.2) and the BCH formula, (5.12) simplifies to

$$\begin{aligned}\exp(-\delta\xi_k^\wedge) &= \exp\left(-(\text{Ad}(\bar{\Xi}_{k-1}^{-1})\mathbf{B}^L \delta\xi_{k-1})^\wedge\right) \exp(-(\mathbf{D}_{k-1}^L \delta\xi_{k-1})^\wedge) \exp(\mathbf{w}_{k-1}^\wedge) \\ &\approx \exp\left(-(\text{Ad}(\bar{\Xi}_{k-1}^{-1})\mathbf{B}^L + \mathbf{D}_{k-1}^L) \delta\xi_{k-1} + \mathbf{w}_{k-1}\right)^\wedge, \\ -\delta\xi_k^\wedge &= (-(\text{Ad}(\bar{\Xi}_{k-1}^{-1})\mathbf{B}^L + \mathbf{D}_{k-1}^L) \delta\xi_{k-1} + \mathbf{w}_{k-1})^\wedge, \\ \delta\xi_k &= (\text{Ad}(\bar{\Xi}_{k-1}^{-1})\mathbf{B}^L + \mathbf{D}_{k-1}^L) \delta\xi_{k-1} - \mathbf{w}_{k-1}.\end{aligned}\quad (5.13)$$

Similarly, the linearized input error is

$$\begin{aligned}\mathbf{E}_{u,k} &= \mathbf{X}_k^{-1}\mathbf{F}_{k-1}(\mathbf{X}_{k-1})\Xi_{k-1} \\ &= \exp(\delta\xi_k^\wedge)\bar{\mathbf{X}}_k^{-1}\mathbf{F}_{k-1}(\bar{\mathbf{X}}_{k-1}) \exp(-(\mathbf{B}^L \delta\xi_{k-1})^\wedge) \bar{\Xi}_{k-1} \exp(-(\mathbf{D}_{k-1}^L \delta\xi_{k-1})^\wedge) \\ &= \exp(\delta\xi_k^\wedge)\bar{\Xi}_{k-1}^{-1} \exp(-(\mathbf{B}^L \delta\xi_{k-1})^\wedge) \bar{\Xi}_{k-1} \exp(-(\mathbf{D}_{k-1}^L \delta\xi_{k-1})^\wedge).\end{aligned}\quad (5.14)$$

Using (2.2) and the BCH formula, (5.14) simplifies to

$$\begin{aligned}\exp(\tilde{\mathbf{e}}_{u,k}^\wedge) &= \exp(\delta\xi_k^\wedge) \exp\left(-\left(\text{Ad}(\bar{\Xi}_{k-1}^{-1})\mathbf{B}^L\delta\xi_{k-1}\right)^\wedge\right) \exp\left(-(\mathbf{D}_{k-1}^L\delta\xi_{k-1})^\wedge\right) \\ &\approx \exp\left(-\left(\left(\text{Ad}(\bar{\Xi}_{k-1}^{-1})\mathbf{B}^L + \mathbf{D}_{k-1}^L\right)\delta\xi_{k-1} + \delta\xi_k\right)^\wedge\right) \\ \mathbf{e}_{u,k}^\wedge &= -\left(\left(\text{Ad}(\bar{\Xi}_{k-1}^{-1})\mathbf{B}^L + \mathbf{D}_{k-1}^L\right)\delta\xi_{k-1} + \delta\xi_k\right)^\wedge, \\ \mathbf{e}_{u,k} &= -\left(\text{Ad}(\bar{\Xi}_{k-1}^{-1})\mathbf{B}^L + \mathbf{D}_{k-1}^L\right)\delta\xi_{k-1} + \delta\xi_k \quad (5.15) \\ &= -\mathbf{w}_{k-1}. \quad (5.16)\end{aligned}$$

Going from (5.15) to (5.16) is done by rearranging (5.13). Thus, $\mathbf{e}_{u,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$.

The measurement error is

$$\begin{aligned}\tilde{\mathbf{e}}_{y,k,j} &= \tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\ &= \tilde{\boldsymbol{\nu}}_{k,j}.\end{aligned}$$

Eliminating the rows of zeros without consequence, $\mathbf{e}_{y,k,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k,j})$.

5.3.2.2 Linearized Error at Operating Point

The uncertainty representation used when linearizing the error about the operating trajectory \mathbf{x}^{op} is $\mathbf{X}_k = \mathbf{X}_k^{\text{op}} \exp(-\delta\epsilon_k^\wedge)$. The linearized prior error is

$$\begin{aligned}\mathbf{E}_{u,0}(\mathbf{x}) &= \mathbf{X}_0^{-1} \check{\mathbf{X}}_0 \\ &= \exp(\delta\epsilon_0^\wedge) \mathbf{X}_0^{\text{op}-1} \check{\mathbf{X}}_0 \\ &= \exp(\delta\epsilon_0^\wedge) \mathbf{E}_{u,0}(\mathbf{x}^{\text{op}}), \\ \exp(\mathbf{e}_{u,0}(\mathbf{x}^\wedge)) &\approx \exp\left((\mathbf{e}_{u,0}(\mathbf{x}^{\text{op}}) + \delta\epsilon_0)^\wedge\right), \\ \mathbf{e}_{u,0}(\mathbf{x}) &= \mathbf{e}_{u,0}(\mathbf{x}^{\text{op}}) + \delta\epsilon_0.\end{aligned}$$

This has the form of (5.5). Therefore, $\mathbf{F}_0^2 = \mathbf{1}$.

Letting $\mathbf{F}_{k-1}(\mathbf{X}_{k-1}^{\text{op}}) = \mathbf{F}_{k-1}^{\text{op}}$, the linearized input error is

$$\begin{aligned}\mathbf{E}_{u,k} &= \mathbf{X}_k^{-1} \mathbf{F}_{k-1}(\mathbf{X}_{k-1}) \Xi_{k-1} \\ &= \exp(\delta\epsilon_k^\wedge) \mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \exp(-(\mathbf{B}\delta\epsilon_{k-1})^\wedge) \Xi_{k-1}^{\text{op}} \exp(-(\mathbf{D}_{k-1}\delta\epsilon_{k-1})^\wedge)\end{aligned}\quad (5.17)$$

$$\begin{aligned}&= \exp(\delta\epsilon_k^\wedge) \mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \exp(-(\mathbf{B}\delta\epsilon_{k-1})^\wedge) \\ &\quad \left(\mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \right)^{-1} \mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \Xi_{k-1}^{\text{op}} \exp(-(\mathbf{D}_{k-1}\delta\epsilon_{k-1})^\wedge) \\ &= \exp(\delta\epsilon_k^\wedge) \mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \exp(-(\mathbf{B}\delta\epsilon_{k-1})^\wedge) \left(\mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \right)^{-1} \mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) \exp(-(\mathbf{D}_{k-1}\delta\epsilon_{k-1})^\wedge),\end{aligned}\quad (5.18)$$

where the error at the operating point is $\mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) = \mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \Xi_{k-1}$. Using the identity (2.2) and the BCH formula, (5.18) is

$$\begin{aligned}\exp(\mathbf{e}_{u,k}^\wedge) &= \exp(\delta\epsilon_k^\wedge) \exp\left(-\left(\text{Ad}(\mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}}) \mathbf{B}^L \delta\epsilon_{k-1}\right)^\wedge\right) \mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) \exp(-(\mathbf{D}_{k-1}^L \delta\epsilon_{k-1})^\wedge) \\ &= \exp(\delta\epsilon_k^\wedge) \exp\left(-\left(\text{Ad}(\mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}}) \mathbf{B}^L \delta\epsilon_{k-1}\right)^\wedge\right) \\ &\quad \exp(\mathbf{e}_{u,k}(\mathbf{x}^{\text{op}})^\wedge) \exp(-(\mathbf{D}_{k-1}^L \delta\epsilon_{k-1})^\wedge) \\ &\approx \exp\left(\left(\mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) - \left(\text{Ad}(\mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}}) \mathbf{B}^L + \mathbf{D}_{k-1}^L\right) \delta\epsilon_{k-1} + \delta\epsilon_k\right)^\wedge\right), \\ \mathbf{e}_{u,k}^\wedge &= \left(\mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) - \left(\text{Ad}(\mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}}) \mathbf{B}^L + \mathbf{D}_{k-1}^L\right) \delta\epsilon_{k-1} + \delta\epsilon_k\right)^\wedge, \\ \mathbf{e}_{u,k} &= \mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) - \left(\text{Ad}(\mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}}) \mathbf{B}^L + \mathbf{D}_{k-1}^L\right) \delta\epsilon_{k-1} + \delta\epsilon_k.\end{aligned}$$

This has the form of (5.6). Therefore, $\mathbf{F}_k^1 = \left(\text{Ad}(\mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}}) \mathbf{B}^L + \mathbf{D}_{k-1}^L\right)$ and $\mathbf{F}_k^2 = -\mathbf{1}$.

The linearized measurement error is

$$\begin{aligned}
\tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) &= \tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{y}}_k^j - (\mathbf{X}_k^{\text{op}} \exp(-\delta\epsilon_k^\wedge))^{-1} \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{y}}_k^j - \exp(\delta\epsilon_k^\wedge) \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&\approx \tilde{\mathbf{y}}_k^j - (\mathbf{1} + \delta\epsilon_k^\wedge) \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \mathbf{p}^{j\text{op}} \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} - \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} - \delta\epsilon_k^\wedge \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} - \delta\epsilon_k^\wedge \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \begin{bmatrix} \mathbf{C}_k^{\text{op}\top} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \delta\epsilon_k^\phi \mathbf{C}_k^{\text{op}\top} \mathbf{p}^{j\text{op}} - \delta\epsilon_k^\phi \mathbf{C}_k^{\text{op}\top} \mathbf{r}_k^{\text{op}} + \delta\epsilon_k^r \\ \mathbf{0} \end{bmatrix}.
\end{aligned}$$

Neglecting the bottom rows of zeros,

$$\begin{aligned}
\mathbf{e}_{y,k,j}(\mathbf{x}) &= \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) - \mathbf{C}_k^{\text{op}\top} \delta\zeta_j - \delta\epsilon_k^\phi \mathbf{C}_k^{\text{op}\top} (\mathbf{p}^{j\text{op}} - \mathbf{r}_k^{\text{op}}) - \delta\epsilon_k^r \\
&= \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) - \delta\epsilon_k^r + \left(\mathbf{C}_k^{\text{op}\top} (\mathbf{p}^{j\text{op}} - \mathbf{r}_k^{\text{op}}) \right)^\times \delta\epsilon_k^\phi - \mathbf{C}_k^{\text{op}\top} \delta\zeta_j \\
&= \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) + \left[\begin{pmatrix} \mathbf{C}_k^{\text{op}\top} (\mathbf{p}^{j\text{op}} - \mathbf{r}_k^{\text{op}}) \end{pmatrix}^\times \quad \mathbf{0} \quad -1 \quad \mathbf{0} \quad \mathbf{0} \right] \delta\epsilon_k - \mathbf{C}_k^{\text{op}\top} \delta\zeta_j,
\end{aligned}$$

which has the form of (5.7), where

$$\mathbf{H}_{k,j}^1 = - \left[\begin{array}{ccccc} \left(\mathbf{C}_k^{\text{op}\top} (\mathbf{p}^{j\text{op}} - \mathbf{r}_k^{\text{op}}) \right)^{\times} & \mathbf{0} & -\mathbf{1} & \mathbf{0} & \mathbf{0} \end{array} \right]$$

and $\mathbf{H}_{k,j}^2 = \mathbf{C}_k^{\text{op}\top}$.

5.3.3 Approach 2: Left-Invariant Error Definition with Type 2 Jacobians

This section offers an alternative way of linearizing the input error from Section 5.3.2.2, leading to Jacobians referred to here as type 2 Jacobians. Consider (5.17). In Section 5.3.2.2, the term $(\mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}})^{-1} \mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}}$ is inserted such that the nominal error $\mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) = \mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \Xi_{k-1}^{\text{op}}$ appears. Alternatively, the nominal error is rearranged to $\mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) \Xi_{k-1}^{\text{op}}^{-1} = \mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}}$. Substituting this into (5.17) yields

$$\begin{aligned} \mathbf{E}_{u,k} &= \exp(\delta\epsilon_k^\wedge) \mathbf{X}_k^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \exp(-(\mathbf{B}^L \delta\epsilon_{k-1})^\wedge) \Xi_{k-1}^{\text{op}} \exp(-(\mathbf{D}_{k-1}^L \delta\epsilon_{k-1})^\wedge) \\ &= \exp(\delta\epsilon_k^\wedge) \mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) \Xi_{k-1}^{\text{op}}^{-1} \exp(-(\mathbf{B}^L \delta\epsilon_{k-1})^\wedge) \Xi_{k-1}^{\text{op}} \exp(-(\mathbf{D}_{k-1}^L \delta\epsilon_{k-1})^\wedge). \end{aligned} \quad (5.19)$$

Using the identity (2.2) and the BCH formula, (5.19) is

$$\begin{aligned} \mathbf{E}_{u,k}(\mathbf{x}) &= \exp(\delta\epsilon_k^\wedge) \mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) \exp\left(-\left(\text{Ad}(\Xi_{k-1}^{\text{op}}{}^{-1}) \mathbf{B}^L \delta\epsilon_{k-1}\right)^\wedge\right) \exp(-(\mathbf{D}_{k-1}^L \delta\epsilon_{k-1})^\wedge), \\ \exp(\mathbf{e}_{u,k}(\mathbf{x})^\wedge) &= \exp(\delta\epsilon_k^\wedge) \exp(\mathbf{e}_{u,k}(\mathbf{x}^{\text{op}})^\wedge) \exp\left(-\left(\text{Ad}(\Xi_{k-1}^{\text{op}}{}^{-1}) \mathbf{B}^L \delta\epsilon_{k-1}\right)^\wedge\right) \\ &\quad \exp(-(\mathbf{D}_{k-1}^L \delta\epsilon_{k-1})^\wedge) \\ &\approx \exp\left(\left(\mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) - \left(\text{Ad}(\Xi_{k-1}^{\text{op}}{}^{-1}) \mathbf{B}^L + \mathbf{D}_{k-1}^L\right) \delta\epsilon_{k-1} + \delta\epsilon_k\right)^\wedge\right), \\ \mathbf{e}_{u,k}(\mathbf{x})^\wedge &= \left(\mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) - \left(\text{Ad}(\Xi_{k-1}^{\text{op}}{}^{-1}) \mathbf{B}^L + \mathbf{D}_{k-1}^L\right) \delta\epsilon_{k-1} + \delta\epsilon_k\right)^\wedge, \\ \mathbf{e}_{u,k}(\mathbf{x}) &= \mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) - \left(\text{Ad}(\Xi_{k-1}^{\text{op}}{}^{-1}) \mathbf{B}^L + \mathbf{D}_{k-1}^L\right) \delta\epsilon_{k-1} + \delta\epsilon_k. \end{aligned}$$

This has the form of (5.6). Therefore, $\mathbf{F}_k^1 = \left(\text{Ad}(\Xi_{k-1}^{\text{op}}{}^{-1}) \mathbf{B}^L + \mathbf{D}_{k-1}^L\right)$ and $\mathbf{F}_k^2 = -\mathbf{1}$. The other Jacobians from Section 5.3.2 hold, meaning $\mathbf{F}_0^2 = -\mathbf{1}$, $\mathbf{H}_{k,j}^2 = \mathbf{C}_k^{\text{op}\top}$ and

$$\mathbf{H}_{k,j}^1 = - \left[\begin{array}{ccccc} \left(\mathbf{C}_k^{\text{op}\top} (\mathbf{p}^{j\text{op}} - \mathbf{r}_k^{\text{op}}) \right)^{\times} & \mathbf{0} & -\mathbf{1} & \mathbf{0} & \mathbf{0} \end{array} \right].$$

The weighting matrices remain $\mathbf{W}_{u,0} = \mathbf{P}_0$, $\mathbf{W}_{u,k} = \mathbf{Q}_k$, and $\mathbf{W}_{y,k,j} = \mathbf{R}_{k,j}$.

5.3.4 Approach 3: Right-Invariant Error Definition

Let the error be defined as $\delta\mathbf{X} = \bar{\mathbf{X}}\mathbf{X}^{-1}$. The uncertainty representation is $\mathbf{X} = \exp(\delta\xi^\wedge)\bar{\mathbf{X}}$. Therefore the state is updated using (5.9). The right-invariant error in the state is

$$\begin{aligned}\bar{\mathbf{X}}_k \mathbf{X}_k^{-1} &= \begin{bmatrix} \bar{\mathbf{C}}_k & \bar{\mathbf{v}}_k & \bar{\mathbf{r}}_k \\ 1 & & \\ & 1 & \\ & & \mathbf{1} & \bar{\beta}_k^1 & \bar{\beta}_k^2 \\ & & 1 & & \\ & & & 1 & \end{bmatrix} \begin{bmatrix} \mathbf{C}_k^\top & -\mathbf{C}_k^\top \mathbf{v}_k & -\mathbf{C}_k \mathbf{r}_k \\ 1 & & \\ & 1 & \\ & & \mathbf{1} & -\beta_k^1 & -\beta_k^2 \\ & & 1 & & \\ & & & 1 & \end{bmatrix} \\ &= \begin{bmatrix} \bar{\mathbf{C}}_k \mathbf{C}_k^\top & \bar{\mathbf{v}}_k - \bar{\mathbf{C}}_k \mathbf{C}_k^\top \mathbf{v}_k & \bar{\mathbf{r}}_k - \bar{\mathbf{C}}_k \mathbf{C}_k^\top \mathbf{r}_k \\ 1 & & \\ & 1 & \\ & & \mathbf{1} & \bar{\beta}_k^1 - \beta_k^1 & \bar{\beta}_k^2 - \beta_k^2 \\ & & 1 & & \\ & & & 1 & \end{bmatrix} \\ &= \begin{bmatrix} \delta\mathbf{C}_k & \delta\mathbf{v}_k & \delta\mathbf{r}_k \\ 1 & & \\ & 1 & \\ & & \mathbf{1} & \delta\beta_k^1 & \delta\beta_k^2 \\ & & 1 & & \\ & & & 1 & \end{bmatrix}.\end{aligned}$$

Thus, the perturbation in \mathbb{R}^{15} is defined as

$$\delta\xi_k = \begin{bmatrix} \log_{SO(3)}(\delta\mathbf{C}_k)^\vee \\ \mathbf{J}_k^{-1} \delta\mathbf{v}_k \\ \mathbf{J}_k^{-1} \delta\mathbf{r}_k \\ \delta\beta_k^1 \\ \delta\beta_k^2 \end{bmatrix}.$$

The right-invariant error in $\mathbf{F}_k(\mathbf{X}_k)$ is

$$\begin{aligned}
\delta \mathbf{F}_k &= \mathbf{F}_k(\bar{\mathbf{X}}_k) \mathbf{F}_k(\mathbf{X}_k)^{-1} \\
&= \left[\begin{array}{ccc|c} \bar{\mathbf{C}}_k & \bar{\mathbf{v}}_k + T\mathbf{g} & \bar{\mathbf{r}}_k + T\bar{\mathbf{v}}_k & \\ & 1 & & \\ & & 1 & \\ & & \mathbf{1} & \bar{\beta}_k^1 \quad \bar{\beta}_k^2 \\ & & & 1 \\ & & & 1 \end{array} \right] \\
&\quad \left[\begin{array}{ccc|c} \mathbf{C}_k^\top & -\mathbf{C}_k^\top(\mathbf{v}_k + T\mathbf{g}) & -\mathbf{C}_k^\top(\mathbf{r}_k + T\mathbf{v}_k) & \\ & 1 & & \\ & & 1 & \\ & & \mathbf{1} & -\beta_k^1 \quad -\beta_k^2 \\ & & & 1 \\ & & & 1 \end{array} \right] \\
&= \left[\begin{array}{ccc|c} \bar{\mathbf{C}}_k \mathbf{C}_k^\top & \delta \mathbf{v}_k + T(\mathbf{1} - \delta \mathbf{C}_k)\mathbf{g} & \delta \mathbf{r}_k + T\delta \mathbf{v}_k & \\ & 1 & & \\ & & 1 & \\ & & \mathbf{1} & \bar{\beta}_k^1 - \beta_k^1 \quad \bar{\beta}_k^2 - \beta_k^2 \\ & & & 1 \\ & & & 1 \end{array} \right].
\end{aligned}$$

To simplify, let $\delta\mathbf{C}_k = \mathbf{1} + \delta\xi_k^{\phi \times}$ to yield

$$\begin{aligned}
\delta\mathbf{F}_k &\approx \begin{bmatrix} \delta\mathbf{C}_k & \delta\mathbf{v}_k + T(\mathbf{1} - \mathbf{1} - \delta\xi_k^{\phi \times})\mathbf{g} & \delta\mathbf{r}_k + T\delta\mathbf{v}_k \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \delta\beta_k^1 & \delta\beta_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} \delta\mathbf{C}_k & \delta\mathbf{v}_k - T\delta\xi_k^{\phi \times}\mathbf{g} & \delta\mathbf{r}_k + T\delta\mathbf{v}_k \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \delta\beta_k^1 & \delta\beta_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} \delta\mathbf{C}_k & \delta\mathbf{v}_k + T\mathbf{g}^\times\delta\xi_k^\phi & \delta\mathbf{r}_k + T\delta\mathbf{v}_k \\ & 1 & \\ & & 1 \\ & & & \mathbf{1} & \delta\beta_k^1 & \delta\beta_k^2 \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}.
\end{aligned}$$

As in Section 5.3.2, this can be written using the exponential map as $\delta\mathbf{F}_k = \exp((\mathbf{B}^R\delta\xi_k)^\wedge)$, where

$$\mathbf{B}^R = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ T\mathbf{g}^\times & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & T\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

As Ξ_{k-1} is state-dependent, it has an associated right-invariant error, which is $\delta\Xi_k = \bar{\Xi}_k\Xi_k^{-1}$.

Expanding, the right-invariant error is

$$\begin{aligned}
\delta \Xi_k &= \Xi_k^{-1} \bar{\Xi}_k \\
&= \begin{bmatrix} \bar{\Psi}_k & T(\mathbf{u}_k^2 + \bar{\beta}_k^2) \\ 1 & \\ & 1 \\ & \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \\ & 1 \\ & 1 \end{bmatrix} \begin{bmatrix} \Psi_k^\top & -T\Psi_k^\top(\mathbf{u}_k^2 + \beta_k^2) & \mathbf{0} \\ 1 & & \\ & 1 & \\ & \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \\ & 1 \\ & 1 \end{bmatrix} \\
&= \begin{bmatrix} \bar{\Psi}_k \Psi_k^\top & -T\bar{\Psi}_k \Psi_k^\top(\mathbf{u}_k^2 + \beta_k^2) + T(\mathbf{u}_k^2 + \bar{\beta}_k^2) \\ 1 & \\ & 0 \\ & \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \\ & 1 \\ & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} \exp_{SO(3)}(T\delta\beta_k^{1\times}) & -T\exp_{SO(3)}(T\delta\beta_k^{1\times})(\mathbf{u}_k^2 + \beta_k^2) + T(\mathbf{u}_k^2 + \bar{\beta}_k^2) \\ 1 & \\ & 0 \\ & \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \\ & 1 \\ & 1 \end{bmatrix}.
\end{aligned}$$

Linearizing by letting $\exp_{SO(3)}(T\delta\beta_k^{1\times}) \approx \mathbf{1} + T\delta\beta_k^{1\times}$ and neglecting second order terms

yields

$$\begin{aligned}
\delta \Xi_k &\approx \begin{bmatrix} \exp_{SO(3)}(T\delta\beta_k^{1\times}) & -T(\mathbf{1} + T\delta\beta_k^{1\times})(\mathbf{u}_k^2 + \beta_k^2) + T(\mathbf{u}_k^2 + \bar{\beta}_k^2) \\ & 1 \\ & & 0 \\ & & & \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \\ & & & 1 \\ & & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} \exp_{SO(3)}(T\delta\beta_k^{1\times}) & T^2\delta\beta_k^{1\times}\mathbf{u}_k^2 - T\beta_k^2 + T\bar{\beta}_k^2 & \mathbf{0} \\ & 1 & & 1 \\ & & \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \\ & & 1 \\ & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} \exp_{SO(3)}(T\delta\beta_k^{1\times}) & -T^2\mathbf{u}_k^{2\times}\delta\beta_k^1 + T\delta\beta_k^2 & \mathbf{0} \\ & 1 & & 1 \\ & & \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \\ & & 1 \\ & & & 1 \end{bmatrix}.
\end{aligned}$$

As $\delta\Xi_k \in \mathcal{G}_2$, it can be expressed using the exponential map as $\delta\Xi_k = \exp(\delta\mathbf{u}_k^\wedge)$, where

$$\delta\mathbf{u}_k = \begin{bmatrix} T\delta\beta_k^1 \\ \mathbf{J}^{-1}(-T^2\mathbf{u}_k^{2\times}\delta\beta_k^1 + T\delta\beta_k^2) \\ \mathbf{0} \end{bmatrix}.$$

This can be rewritten as $\delta\Xi_k = \exp((\mathbf{D}_k^R \delta\boldsymbol{\xi}_k)^\wedge)$, where

$$\mathbf{D}_k^R = \begin{bmatrix} \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} & T\mathbf{1} & \mathbf{0} \\ \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} & -T^2\mathbf{u}_k^{2\times} & T\mathbf{1} \\ \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

5.3.4.1 Probability Distribution of the Error

The prior error distribution is found by linearizing $\mathbf{E}_{u,0}(\mathbf{x}) = \bar{\mathbf{X}}_0 \mathbf{X}_0^{-1}$. Letting $\mathbf{X}_0 = \exp(-\delta\xi_0^\wedge) \bar{\mathbf{X}}_0$,

$$\begin{aligned}\mathbf{E}_{u,0}(\mathbf{x}) &= \bar{\mathbf{X}}_0 \mathbf{X}_0^{-1}, \\ \exp(\mathbf{e}_{u,0}(\mathbf{x})^\wedge) &= \bar{\mathbf{X}}_0 \bar{\mathbf{X}}_0^{-1} \exp(\delta\xi_0^\wedge) \\ &= \exp(\delta\xi_0^\wedge), \\ \mathbf{e}_{u,0}(\mathbf{x})^\wedge &\approx \delta\xi_0^\wedge, \\ \mathbf{e}_{u,0}(\mathbf{x}) &= \delta\xi_0.\end{aligned}$$

Thus, $\mathbf{e}_{u,0}(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0)$.

To find the input error probability distribution, the process model is linearized and the effect of noise is recovered. To ease the notation, let $\mathbf{F}_{k-1}(\mathbf{X}_{k-1}) = \mathbf{F}_{k-1}$. Letting $\mathbf{F}_{k-1} = \exp(-(\mathbf{B}^R \delta\xi_{k-1})^\wedge) \bar{\mathbf{F}}_{k-1}$ and $\Xi_{k-1} = \exp(-(\mathbf{D}_{k-1}^R \delta\xi_{k-1})^\wedge) \bar{\Xi}_{k-1}$ the linearized process model is

$$\begin{aligned}\mathbf{X}_k &= \mathbf{F}_{k-1} \Xi_{k-1} \exp(\mathbf{w}_{k-1}^\wedge), \\ \exp(-\delta\xi_k^\wedge) \bar{\mathbf{X}}_k &= \exp(-(\mathbf{B}^R \delta\xi_{k-1})^\wedge) \bar{\mathbf{F}}_{k-1} \exp(-(\mathbf{D}_{k-1}^R \delta\xi_{k-1})^\wedge) \bar{\Xi}_{k-1} \exp(\mathbf{w}_{k-1}^\wedge), \\ \exp(-\delta\xi_k^\wedge) &= \exp(-(\mathbf{B}^R \delta\xi_{k-1})^\wedge) \bar{\mathbf{F}}_{k-1} \exp(-(\mathbf{D}_{k-1}^R \delta\xi_{k-1})^\wedge) \bar{\Xi}_{k-1} \exp(\mathbf{w}_{k-1}^\wedge) \bar{\mathbf{X}}_k^{-1} \\ &= \exp(-(\mathbf{B}^R \delta\xi_{k-1})^\wedge) \bar{\mathbf{F}}_{k-1} \exp(-(\mathbf{D}_{k-1}^R \delta\xi_{k-1})^\wedge) \\ &\quad \bar{\mathbf{F}}_{k-1}^{-1} \bar{\mathbf{F}}_{k-1} \bar{\Xi}_{k-1} \exp(\mathbf{w}_{k-1}^\wedge) (\bar{\mathbf{F}}_{k-1} \bar{\Xi}_{k-1})^{-1}.\end{aligned}\tag{5.20}$$

Using (2.2) and the BCH formula, (5.20) simplifies to

$$\begin{aligned}\exp(-\delta\xi_k^\wedge) &= \exp(-(\mathbf{B}^R \delta\xi_{k-1})^\wedge) \exp(-(\text{Ad}(\bar{\mathbf{F}}_{k-1}) \mathbf{D}_{k-1}^R \delta\xi_{k-1})^\wedge) \exp((\text{Ad}(\bar{\mathbf{F}}_{k-1} \bar{\Xi}_{k-1}) \mathbf{w}_{k-1})^\wedge) \\ &\approx \exp(-((\mathbf{B} + \text{Ad}(\bar{\mathbf{F}}_{k-1}) \mathbf{D}_{k-1}^R) \delta\xi_{k-1} + \text{Ad}(\bar{\mathbf{F}}_{k-1} \bar{\Xi}_{k-1}) \mathbf{w}_{k-1})^\wedge), \\ \delta\xi_k &= (\mathbf{B}^R + \text{Ad}(\bar{\mathbf{F}}_{k-1}) \mathbf{D}_{k-1}^R) \delta\xi_{k-1} - \text{Ad}(\bar{\mathbf{F}}_{k-1} \bar{\Xi}_{k-1}) \mathbf{w}_{k-1}.\end{aligned}\tag{5.21}$$

Letting $\mathbf{F}_{k-1} = \exp(-(\mathbf{B}^R \delta\xi_{k-1})^\wedge) \bar{\mathbf{F}}_{k-1}$ and $\Xi_{k-1} = \exp((\mathbf{D}_{k-1}^R \delta\xi_{k-1})^\wedge) \bar{\Xi}_{k-1}$, the lin-

earized input error is

$$\begin{aligned}
\mathbf{E}_{u,k}(\mathbf{x}) &= \mathbf{F}_{k-1} \boldsymbol{\Xi}_{k-1} \mathbf{X}_k^{-1} \\
&= \exp\left(-(\mathbf{B}^R \delta \boldsymbol{\xi}_{k-1})^\wedge\right) \bar{\mathbf{F}}_{k-1} \exp\left(-(\mathbf{D}_{k-1}^R \delta \boldsymbol{\xi}_{k-1})^\wedge\right) \bar{\boldsymbol{\Xi}}_{k-1} \bar{\mathbf{X}}_k^{-1} \exp(\delta \boldsymbol{\xi}_k^\wedge) \\
&= \exp\left(-(\mathbf{B}^R \delta \boldsymbol{\xi}_{k-1})^\wedge\right) \bar{\mathbf{F}}_{k-1} \exp\left(-(\mathbf{D}_{k-1}^R \delta \boldsymbol{\xi}_{k-1})^\wedge\right) \bar{\mathbf{F}}_{k-1}^{-1} \bar{\mathbf{F}}_{k-1} \bar{\boldsymbol{\Xi}}_{k-1} \bar{\mathbf{X}}_k^{-1} \exp(\delta \boldsymbol{\xi}_k^\wedge) \\
&= \exp\left(-(\mathbf{B}^R \delta \boldsymbol{\xi}_{k-1})^\wedge\right) \bar{\mathbf{F}}_{k-1} \exp\left(-(\mathbf{D}_{k-1}^R \delta \boldsymbol{\xi}_{k-1})^\wedge\right) \bar{\mathbf{F}}_{k-1}^{-1} \bar{\mathbf{F}}_{k-1} \exp(\delta \boldsymbol{\xi}_k^\wedge). \quad (5.22)
\end{aligned}$$

Using (2.2) and the BCH formula, (5.22) simplifies to

$$\begin{aligned}
\exp(\mathbf{e}_{u,k}(\mathbf{x})^\wedge) &= \exp\left(-(\mathbf{B}^R \delta \boldsymbol{\xi}_{k-1})^\wedge\right) \exp\left(-(\text{Ad}(\bar{\mathbf{F}}_{k-1}) \mathbf{D}_{k-1}^R \delta \boldsymbol{\xi}_{k-1})^\wedge\right) \exp(\delta \boldsymbol{\xi}_k^\wedge) \\
&\approx \exp\left((- \mathbf{B}^R + \text{Ad}(\bar{\mathbf{F}}_{k-1}) \mathbf{D}_{k-1}^R) \delta \boldsymbol{\xi}_{k-1} + \delta \boldsymbol{\xi}_k\right), \quad (5.23)
\end{aligned}$$

$$\mathbf{e}_{u,k}(\mathbf{x})^\wedge = (- \mathbf{B}^R + \text{Ad}(\bar{\mathbf{F}}_{k-1}) \mathbf{D}_{k-1}^R) \delta \boldsymbol{\xi}_{k-1} + \delta \boldsymbol{\xi}_k,$$

$$\mathbf{e}_{u,k}(\mathbf{x}) = -(\mathbf{B}^R + \text{Ad}(\bar{\mathbf{F}}_{k-1}) \mathbf{D}_{k-1}^R) \delta \boldsymbol{\xi}_{k-1} + \delta \boldsymbol{\xi}_k \quad (5.23)$$

$$= -\text{Ad}(\bar{\mathbf{F}}_{k-1} \bar{\boldsymbol{\Xi}}_{k-1}) \mathbf{w}_{k-1}. \quad (5.24)$$

Going from (5.23) to (5.24) is done by rearranging (5.21). Thus, $\mathbf{e}_{u,k}(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \text{Ad}(\bar{\mathbf{F}}_{k-1} \bar{\boldsymbol{\Xi}}_{k-1}) \mathbf{Q}_k \text{Ad}(\bar{\mathbf{F}}_{k-1} \bar{\boldsymbol{\Xi}}_{k-1})^\top)$.

The measurement error distribution is found in a manner identical to that used in Section 5.3.2, yielding $\mathbf{e}_{y,k,j}(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k,j})$.

5.3.4.2 Linearized Error at Operating Point

The uncertainty representation used when linearizing the error about the operating trajectory \mathbf{x}^{op} is $\mathbf{X}_k = \exp(-\delta \boldsymbol{\epsilon}_k^\wedge) \mathbf{X}_k^{\text{op}}$. It follows that $\mathbf{F}_{k-1} = \exp\left(-(\mathbf{B}^R \delta \boldsymbol{\epsilon}_{k-1})^\wedge\right) \mathbf{F}_{k-1}^{\text{op}}$ and $\boldsymbol{\Xi}_{k-1} = \exp\left(-(\mathbf{D}_{k-1}^R \delta \boldsymbol{\xi}_{k-1})^\wedge\right) \boldsymbol{\Xi}_{k-1}^{\text{op}}$.

The linearized prior error is

$$\begin{aligned}
\mathbf{E}_{u,0}(\mathbf{x}) &= \check{\mathbf{X}}_0 \mathbf{X}_0^{-1} \\
&= \check{\mathbf{X}}_0 \mathbf{X}_0^{\text{op}-1} \exp(\delta \boldsymbol{\epsilon}_0^\wedge) \\
&= \mathbf{E}_{u,0}(\mathbf{x}^{\text{op}}) \exp(\delta \boldsymbol{\epsilon}_0^\wedge) \\
&\approx \exp((\mathbf{e}_{u,0}(\mathbf{x}^{\text{op}}) + \delta \boldsymbol{\epsilon}_0)^\wedge) \\
\mathbf{e}_{u,0}(\mathbf{x}) &= \mathbf{e}_{u,0}(\mathbf{x}^{\text{op}}) + \delta \boldsymbol{\epsilon}_0.
\end{aligned}$$

This has the form of (5.5). Therefore, $\mathbf{F}_0^2 = \mathbf{1}$.

The linearized input error is

$$\begin{aligned}
\mathbf{E}_{u,k} &= \mathbf{F}_{k-1} \boldsymbol{\Xi}_{k-1} \mathbf{X}_k^{-1} \\
&= \exp \left(-(\mathbf{B}^R \delta \boldsymbol{\epsilon}_{k-1})^\wedge \right) \mathbf{F}_{k-1}^{\text{op}} \exp \left(-(\mathbf{D}_{k-1}^R \delta \boldsymbol{\xi}_{k-1})^\wedge \right) \boldsymbol{\Xi}_{k-1}^{\text{op}} \mathbf{X}_k^{\text{op}-1} \exp (\delta \boldsymbol{\epsilon}_k^\wedge) \\
&= \exp \left(-(\mathbf{B}^R \delta \boldsymbol{\epsilon}_{k-1})^\wedge \right) \mathbf{F}_{k-1}^{\text{op}} \exp \left(-(\mathbf{D}_{k-1}^R \delta \boldsymbol{\xi}_{k-1})^\wedge \right) \mathbf{F}_{k-1}^{\text{op}-1} \mathbf{F}_{k-1}^{\text{op}} \boldsymbol{\Xi}_{k-1}^{\text{op}} \mathbf{X}_k^{\text{op}-1} \exp (\delta \boldsymbol{\epsilon}_k^\wedge) \\
&= \exp \left(-(\mathbf{B}^R \delta \boldsymbol{\epsilon}_{k-1})^\wedge \right) \mathbf{F}_{k-1}^{\text{op}} \exp \left(-(\mathbf{D}_{k-1}^R \delta \boldsymbol{\xi}_{k-1})^\wedge \right) \mathbf{F}_{k-1}^{\text{op}-1} \mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) \exp (\delta \boldsymbol{\epsilon}_k^\wedge), \quad (5.25)
\end{aligned}$$

where the error at the operating point is $\mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) = \mathbf{F}_{k-1}^{\text{op}} \boldsymbol{\Xi}_{k-1}^{\text{op}} \mathbf{X}_k^{\text{op}-1}$. Using (2.2) and the BCH formula, (5.25) is

$$\begin{aligned}
\exp (\mathbf{e}_{u,k}(\mathbf{x})^\wedge) &= \exp \left(-(\mathbf{B}^R \delta \boldsymbol{\epsilon}_{k-1})^\wedge \right) \exp \left(-(\text{Ad}(\mathbf{F}_{k-1}^{\text{op}}) \mathbf{D}_{k-1}^R \delta \boldsymbol{\epsilon}_{k-1})^\wedge \right) \mathbf{E}_{u,k}(\mathbf{x}^{\text{op}}) \exp (\delta \boldsymbol{\epsilon}_k^\wedge) \\
&\approx \exp \left((\mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) - (\mathbf{B}^R + \text{Ad}(\mathbf{F}_{k-1}^{\text{op}}) \mathbf{D}_{k-1}^R) \delta \boldsymbol{\epsilon}_{k-1} + \delta \boldsymbol{\epsilon}_k)^\wedge \right), \\
\mathbf{e}_{u,k}(\mathbf{x})^\wedge &= (\mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) - (\mathbf{B}^R + \text{Ad}(\mathbf{F}_{k-1}^{\text{op}}) \mathbf{D}_{k-1}^R) \delta \boldsymbol{\epsilon}_{k-1} + \delta \boldsymbol{\epsilon}_k)^\wedge, \\
\mathbf{e}_{u,k}(\mathbf{x}) &= \mathbf{e}_{u,k}(\mathbf{x}^{\text{op}}) - (\mathbf{B}^R + \text{Ad}(\mathbf{F}_{k-1}^{\text{op}}) \mathbf{D}_{k-1}^R) \delta \boldsymbol{\epsilon}_{k-1} + \delta \boldsymbol{\epsilon}_k.
\end{aligned}$$

This has the form of (5.6). Therefore, $\mathbf{F}_k^1 = \mathbf{B}^R + \text{Ad}(\mathbf{F}_{k-1}^{\text{op}}) \mathbf{D}_{k-1}^R$ and $\mathbf{F}_k^2 = -\mathbf{1}$.

Letting $\tilde{\mathbf{e}}_{y,k,j} = [\mathbf{e}_{y,k,j}(\mathbf{x})^\top \mathbf{0}]^\top$, the measurement error is

$$\begin{aligned}
\tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) &= \tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{y}}_{k,j} - (\exp (-\delta \boldsymbol{\epsilon}_k^\wedge) \mathbf{X}_k^{\text{op}}) \begin{bmatrix} \mathbf{p}^{j \text{ op}} + \delta \boldsymbol{\zeta}_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{\text{op}-1} \exp (\delta \boldsymbol{\epsilon}_k^\wedge) \begin{bmatrix} \mathbf{p}^{j \text{ op}} + \delta \boldsymbol{\zeta}_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix}. \quad (5.26)
\end{aligned}$$

Linearizing with $\exp(\delta\epsilon_k^\wedge) \approx \mathbf{1} + \delta\epsilon_k^\wedge$, and neglecting second order terms, (5.26) becomes

$$\begin{aligned}
\tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) &= \tilde{\mathbf{y}}_k^j - \mathbf{X}_k^{\text{op}-1} (\mathbf{1} + \delta\epsilon_k^\wedge) \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} - \mathbf{X}_k^{\text{op}-1} \delta\epsilon_k^\wedge \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} - \mathbf{X}_k^{\text{op}-1} \delta\epsilon_k^\wedge \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} - \mathbf{X}_k^{\text{op}-1} \delta\epsilon_k^\wedge \begin{bmatrix} \mathbf{p}^{j\text{op}} \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} - \mathbf{X}_k^{\text{op}-1} \delta\epsilon_k^\wedge \begin{bmatrix} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \left[\begin{array}{c} \mathbf{C}_k^{\text{op}\top} \delta\zeta_j \\ \mathbf{0} \end{array} \right] - \left[\begin{array}{c} \mathbf{C}_k^{\text{op}\top} \delta\epsilon_k^{\phi \times} \mathbf{p}^{j\text{op}} + \mathbf{C}_k^{\text{op}\top} \delta\epsilon_k^r \\ \mathbf{0} \end{array} \right] \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \left[\begin{array}{c} \mathbf{C}_k^{\text{op}\top} \delta\zeta_j \\ \mathbf{0} \end{array} \right] - \left[\begin{array}{c} -\mathbf{C}_k^{\text{op}\top} \mathbf{p}^{j\text{op}\times} \delta\epsilon_k^\phi + \mathbf{C}_k^{\text{op}\top} \delta\epsilon_k^r \\ \mathbf{0} \end{array} \right]. \tag{5.27}
\end{aligned}$$

Removing the bottom rows of zeros of (5.27) without consequence yields

$$\begin{aligned}
\mathbf{e}_{y,k,j}(\mathbf{x}) &= \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) - \mathbf{C}_k^{\text{op}\top} \delta\zeta_j + \mathbf{C}_k^{\text{op}\top} \mathbf{p}^{j\text{op}\times} \delta\epsilon_k^\phi - \mathbf{C}_k^{\text{op}\top} \delta\epsilon_k^r \\
&= \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) + \left[\begin{array}{ccccc} \mathbf{C}_k^{\text{op}\top} \mathbf{p}^{j\text{op}\times} & -\mathbf{C}_k^{\text{op}\top} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right] \delta\epsilon_k - \mathbf{C}_k^{\text{op}\top} \delta\zeta_j,
\end{aligned}$$

which has the form of (5.7), where

$$\mathbf{H}_{k,j}^1 = - \left[\begin{array}{ccccc} \mathbf{C}_k^{\text{op}\top} \mathbf{p}^{j\text{op}\times} & \mathbf{0} & -\mathbf{C}_k^{\text{op}\top} & \mathbf{0} & \mathbf{0} \end{array} \right]$$

and $\mathbf{H}_{k,j}^2 = \mathbf{C}_k^{\text{op}\top}$.

5.3.5 Approaches 4 and 5: Left-Invariant Error with Invariant Measurement Error

The previous three approaches defined the measurement error as (5.4). Consider instead a right-invariant measurement error,

Edited
2021/2/17

$$\tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) = \bar{\mathbf{X}}_k \left(\tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \right).$$

Using this error definition runs counter to what is done in the literature. It is only advantageous to use a left-invariant error with a left-invariant measurement model. However, this measurement model and measurement error are right-invariant. Despite this, the derivation is presented for completeness. The probability distribution of this new measurement error is

$$\begin{aligned} \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) &= \bar{\mathbf{X}}_k \left(\tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \right), \\ &= \bar{\mathbf{X}}_k \boldsymbol{\nu}_{k,j}. \end{aligned}$$

By removing the unnecessary rows of zeros, $\mathbf{e}_{y,k,j}(\mathbf{x}) = \bar{\mathbf{C}}_k \boldsymbol{\nu}_{k,j}$ and $\mathbf{e}_{y,k,j}(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \bar{\mathbf{C}}_k \mathbf{R}_{k,j} \bar{\mathbf{C}}_k^\top)$.

Letting $\bar{\mathbf{X}}_k = \mathbf{X}_k^{\text{op}}$, $\mathbf{X}_k = \mathbf{X}_k^{\text{op}} \exp(-\delta \epsilon_k^\wedge)$ and $\mathbf{p}^j = \mathbf{p}^{j \text{ op}} + \delta \zeta_j$, the measurement error at the

operating point is

$$\begin{aligned}
\tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) &= \mathbf{X}_k^{\text{op}} \left(\tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \right) \\
&= \mathbf{X}_k^{\text{op}} \left(\tilde{\mathbf{y}}_{k,j} - \exp(\delta\epsilon_k^\wedge) \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \right) \\
&= \mathbf{X}_k^{\text{op}} \tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{\text{op}} \exp(\delta\epsilon_k^\wedge) \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \mathbf{X}_k^{\text{op}} \tilde{\mathbf{y}}_{k,j} - \exp((\text{Ad}(\mathbf{X}_k^{\text{op}})\delta\epsilon_k^\wedge)) \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix}.
\end{aligned}$$

Linearizing by letting $\exp(\text{Ad}(\mathbf{X}_k^{\text{op}})\delta\epsilon_k^\wedge) \approx \mathbf{1} + \text{Ad}(\mathbf{X}_k^{\text{op}})\delta\epsilon_k$ and neglecting higher order terms

yields

$$\begin{aligned}
\tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) &\approx \mathbf{X}_k^{\text{op}} \tilde{\mathbf{y}}_{k,j} - (\mathbf{1} + (\text{Ad}(\mathbf{X}_k^{\text{op}}) \delta \boldsymbol{\epsilon}_k)^{\wedge}) \begin{bmatrix} \mathbf{p}^{j \text{op}} + \delta \boldsymbol{\zeta}_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \mathbf{X}_k^{\text{op}} \tilde{\mathbf{y}}_{k,j} - \begin{bmatrix} \mathbf{p}^{j \text{op}} \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \delta \boldsymbol{\zeta}_j \\ \mathbf{0} \end{bmatrix} - (\text{Ad}(\mathbf{X}_k^{\text{op}}) \delta \boldsymbol{\epsilon}_k)^{\wedge} \begin{bmatrix} \mathbf{p}^{j \text{op}} + \delta \boldsymbol{\zeta}_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \begin{bmatrix} \delta \boldsymbol{\zeta}_j \\ \mathbf{0} \end{bmatrix} - (\text{Ad}(\mathbf{X}_k^{\text{op}}) \delta \boldsymbol{\epsilon}_k)^{\wedge} \begin{bmatrix} \mathbf{p}^{j \text{op}} + \delta \boldsymbol{\zeta}_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \begin{bmatrix} \delta \boldsymbol{\zeta}_j \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} (\mathbf{C}_k^{\text{op}} \delta \boldsymbol{\epsilon}_k^{\phi})^{\times} \mathbf{p}^{j \text{op}} + \mathbf{r}_k^{\text{op} \times} \mathbf{C}_k^{\text{op}} \delta \boldsymbol{\epsilon}_k^{\phi} + \mathbf{C}_k^{\text{op}} \delta \boldsymbol{\epsilon}_k^{\text{r}} \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \begin{bmatrix} \delta \boldsymbol{\zeta}_j \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} -\mathbf{p}^{j \text{op} \times} \mathbf{C}_k^{\text{op}} \delta \boldsymbol{\epsilon}_k^{\phi} + \mathbf{r}_k^{\text{op} \times} \mathbf{C}_k^{\text{op}} \delta \boldsymbol{\epsilon}_k^{\phi} + \mathbf{C}_k^{\text{op}} \delta \boldsymbol{\epsilon}_k^{\text{r}} \\ \mathbf{0} \end{bmatrix}. \quad (5.28)
\end{aligned}$$

Neglecting the bottom three rows of (5.28) without consequence yields

$$\begin{aligned}
\mathbf{e}_{y,k,j}(\mathbf{x}) &= \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) - \delta \boldsymbol{\zeta}_j + \mathbf{p}^{j \text{op} \times} \mathbf{C}_k^{\text{op}} \delta \boldsymbol{\epsilon}_k^{\phi} - \mathbf{r}_k^{\text{op} \times} \mathbf{C}_k^{\text{op}} \delta \boldsymbol{\epsilon}_k^{\phi} - \mathbf{C}_k^{\text{op}} \delta \boldsymbol{\epsilon}_k^{\text{r}} \\
&= \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) + \begin{bmatrix} (\mathbf{p}^{j \text{op}} - \mathbf{r}_k^{\text{op}})^{\times} \mathbf{C}_k^{\text{op}} & \mathbf{0} & -\mathbf{C}_k^{\text{op}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \delta \boldsymbol{\epsilon}_k - \delta \boldsymbol{\zeta}_j,
\end{aligned}$$

which has the form of (5.7), where

$$\mathbf{H}_{k,j}^1 = - \begin{bmatrix} (\mathbf{p}^{j \text{op}} - \mathbf{r}_k^{\text{op}})^{\times} \mathbf{C}_k^{\text{op}} & \mathbf{0} & -\mathbf{C}_k^{\text{op}} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

and $\mathbf{H}_{k,j}^2 = \mathbf{1}$.

Approach 4 substitutes the measurement Jacobians of Approach 1 with those derived here, while Approach 5 substitutes the measurement Jacobians of Approach 2 with those derived here. In both cases, the left-invariant measurement error is used to compute the measurement error.

5.3.6 Approach 6: Right-Invariant Error with Invariant Measurement Error

Similarly to Approaches 4 and 5, an invariant measurement error is used instead of (5.4). The right-invariant measurement error used here is

$$\tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) = \bar{\mathbf{X}}_k \left(\tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \right).$$

Using this measurement error definition, coupled with the fact that the measurement model (5.11) is right invariant, leads to Jacobians dependent only on the landmark estimate. The probability distribution of this new measurement error is

$$\begin{aligned} \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) &= \bar{\mathbf{X}}_k \left(\tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \right) \\ &= \bar{\mathbf{X}}_k \boldsymbol{\nu}_{k,j} \end{aligned}$$

By removing the three unnecessary rows of zeros, $\mathbf{e}_{y,k,j}(\mathbf{x}) = \bar{\mathbf{C}}_k \boldsymbol{\nu}_{k,j}$ and $\mathbf{e}_{y,k,j}(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \bar{\mathbf{C}}_k \mathbf{R}_{k,j} \bar{\mathbf{C}}_k^\top)$.

Letting $\bar{\mathbf{X}}_k = \mathbf{X}_k^{\text{op}}$, $\mathbf{X}_k = \exp(-\delta \boldsymbol{\epsilon}_k^\wedge) \mathbf{X}_k^{\text{op}}$ and $\mathbf{p}^j = \mathbf{p}^{j,\text{op}} + \delta \boldsymbol{\zeta}_j$, the measurement error at the

operating point is

$$\begin{aligned}
\tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) &= \mathbf{X}_k^{\text{op}} \left(\tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{p}^j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \right) \\
&= \mathbf{X}_k^{\text{op}} \left(\tilde{\mathbf{y}}_{k,j} - \exp(\delta \epsilon_k^\wedge) \mathbf{X}_k^{\text{op}-1} \begin{bmatrix} \mathbf{p}^{j \text{op}} + \delta \zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \right) \\
&= \mathbf{X}_k^{\text{op}} \tilde{\mathbf{y}}_{k,j} - \mathbf{X}_k^{\text{op}} \mathbf{X}_k^{\text{op}-1} \exp(\delta \epsilon_k^\wedge) \begin{bmatrix} \mathbf{p}^{j \text{op}} + \delta \zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \mathbf{X}_k^{\text{op}} \tilde{\mathbf{y}}_{k,j} - \exp(\delta \epsilon_k^\wedge) \begin{bmatrix} \mathbf{p}^{j \text{op}} + \delta \zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix}.
\end{aligned}$$

Linearizing by letting $\exp(\delta\epsilon_k^\wedge) \approx \mathbf{1} + \delta\epsilon_k$ and neglecting higher order terms yields

$$\begin{aligned}
\tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}) &\approx \mathbf{X}_k^{\text{op}} \tilde{\mathbf{y}}_{k,j} - (\mathbf{1} + \delta\epsilon_k)^\wedge \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \mathbf{X}_k^{\text{op}} \tilde{\mathbf{y}}_{k,j} - \begin{bmatrix} \mathbf{p}^{j\text{op}} \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} - \delta\epsilon_k^\wedge \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \begin{bmatrix} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} - \delta\epsilon_k^\wedge \begin{bmatrix} \mathbf{p}^{j\text{op}} + \delta\zeta_j \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \begin{bmatrix} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \delta\epsilon_k^{\phi \times} \mathbf{p}^{j\text{op}} + \delta\epsilon_k^r \\ \mathbf{0} \end{bmatrix} \\
&= \tilde{\mathbf{e}}_{y,k,j}(\mathbf{x}^{\text{op}}) - \begin{bmatrix} \delta\zeta_j \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} -\mathbf{p}^{j\text{op} \times} \delta\epsilon_k^\phi + \delta\epsilon_k^r \\ \mathbf{0} \end{bmatrix}. \tag{5.29}
\end{aligned}$$

Neglecting the bottom rows of zeros (5.29) without consequence yields

$$\begin{aligned}
\mathbf{e}_{y,k,j}(\mathbf{x}) &= \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) - \delta\zeta_j + \mathbf{p}^{j\text{op} \times} \delta\epsilon_k^\phi - \delta\epsilon_k^r \\
&= \mathbf{e}_{y,k,j}(\mathbf{x}^{\text{op}}) + \begin{bmatrix} \mathbf{p}^{j\text{op} \times} & \mathbf{0} & -\mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \delta\epsilon_k - \delta\zeta_j,
\end{aligned}$$

which has the form of (5.7), where

$$\mathbf{H}_{k,j}^1 = - \begin{bmatrix} \mathbf{p}^{j\text{op} \times} & \mathbf{0} & -\mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

and $\mathbf{H}_{k,j}^2 = \mathbf{1}$.

To implement Approach 6, the measurement Jacobians of Approach 3 are substituted with those derived here, and the right-invariant innovation is used to compute the measurement error.

Trial	σ_k^1 rad	σ_k^2 m/s	σ_k^3 rad/s ²	σ_k^4 m/s ²	$\sigma_{k,j}$ m
Rate Gyro Noise	0.05 to 1	0.05	0.005	0.005	0.05
Accelerometer Noise	0.05	0.05 to 1	0.001	0.005	0.05
Rate Gyro Bias Noise	0.05	0.05	0 to 0.1	0.005	0.05
Accelerometer Bias Noise	0.05	0.05	0.005	0 to 0.1	0.05
Landmark Sensor Noise	0.05	0.05	0.005	0.005	0.05 to 1

Table 5.3: Standard deviation of the noise injected into each measurement for each trial.

5.4 Simulation Results

The proposed batch SLAM approaches are all compared in simulation. The effect of noise in the sensors is investigated for two reasons. The noise in the sensors directly effects the quality of the dead-reckoning solution, which is used to initialize the nonlinear least squares problem. The quality of this initialization typically has a significant effect on the speed of convergence and accuracy of the solution. The magnitude of the noise in the sensors will also effect the accuracy of the Jacobians which mainly depend on measurements, namely \mathbf{F}_k^1 of Approaches 2 and 5.

The trajectory used is similar to that used in Chapter 4. However, this trajectory, shown in Figure 5.1 is shorter, to reduce the computational load. As in Chapter 4, the noise is assumed isotropic. Therefore, $\mathbf{Q}_k^i = \sigma_k^i \mathbf{1}$ and $\mathbf{R}_{k,j} = \sigma_{k,j} \mathbf{1}$. The noise standard deviations used in the simulations are found in Table 5.3. The sensor noises are incremented by 0.05, while the bias noises are incremented by 0.005, where appropriate units are assumed. At each noise level, 25 simulations are performed, where the noise profile and initial bias estimate are varied. As the initial position, attitude and velocity are not observable, they are initialized accurately. However, the initial biases are observable. The initial gyro bias error is sampled from a zero-mean Gaussian distribution with covariance $\mathbf{P}_0^4 = 0.01^2 \mathbf{1}$ (rad/s²)² and the initial accelerometer bias error is sampled from a zero-mean Gaussian distribution with covariance $\mathbf{P}_0^5 = 0.01^2 \mathbf{1}$ (m/s²)². A simple Gauss-Newton scheme is used to solve the nonlinear least-squares problem. The line-search method and Levenberg-Marquardt method presented in Chapter 2 were tested, but in the end did not provide better results than simple Gauss-Newton.

The results are presented in Figures 5.2 to 5.6. The main takeaway is that, generally, the six different approaches provide similar results. In particular, there is no discernible difference between using the invariant formulations of the measurement error (Approaches 4, 5, and 6) as opposed to their standard counterparts (Approaches 1, 2, and 3). The results from Approaches 1, 2, 4, and 5 are similar to such an extent that they are often indistinguishable

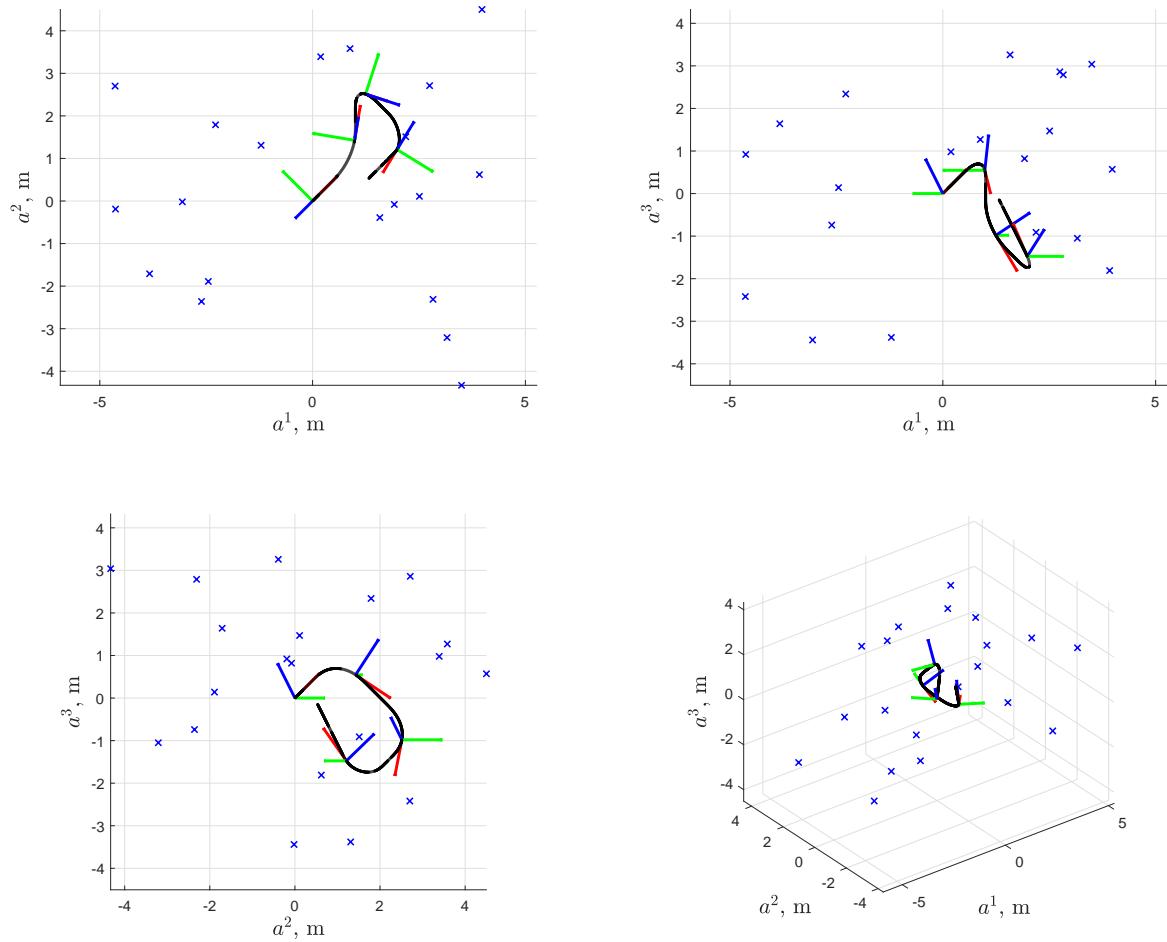


Figure 5.1: Trajectory used in simulations. The blue markers represent the landmarks, the trajectory is shown in black, and a triad is used to show the orientation at various times during the trajectory.

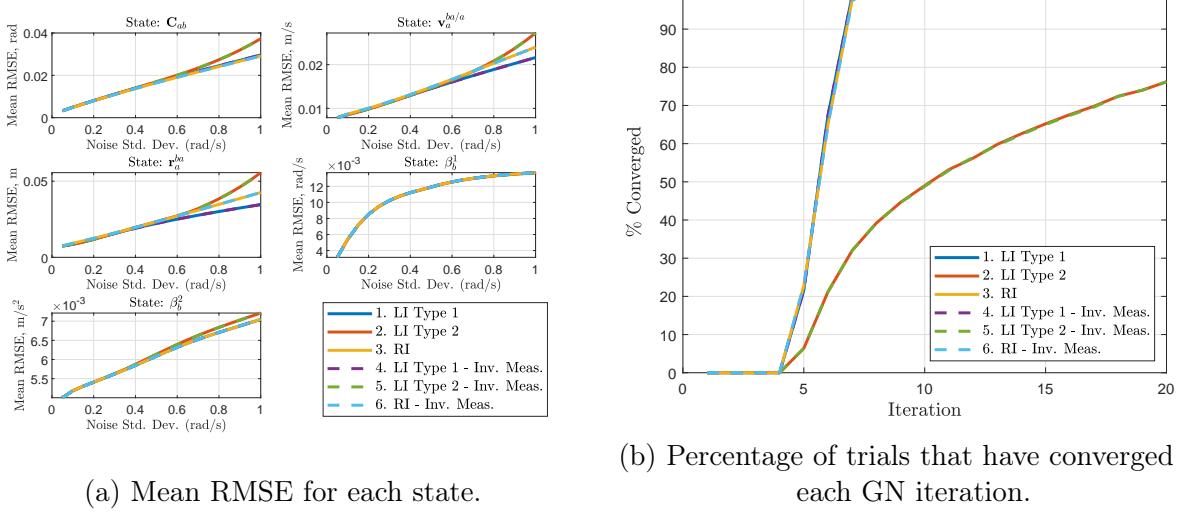
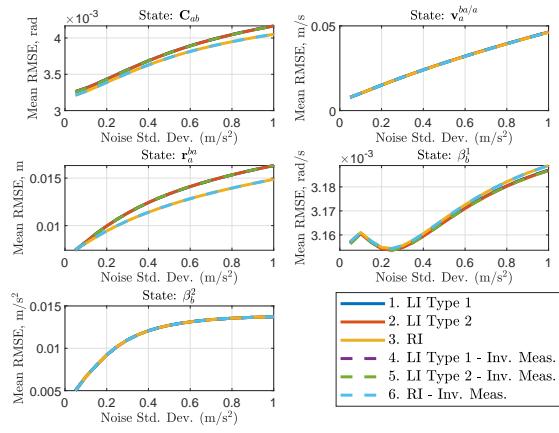


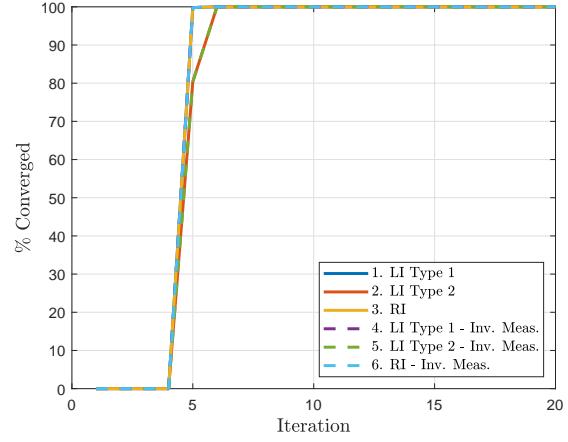
Figure 5.2: Results of 25 Monte Carlo trials comparing the various batch SLAM approaches, where the noise in the rate gyro was varied.

when they are overlaid. Furthermore, the impact of reducing the state dependence of the Jacobians is unclear. After multiple iterations of the nonlinear least squares solver, the operating trajectory begins approaching the truth trajectory. Thus, assuming the solver converges to the optimal solution, the Jacobians become more and more accurate. The advantage of state-independent Jacobians is most notable in situations when the Jacobians are normally inaccurate, which isn't the case here. In addition, the error definitions used to derive the Jacobians in Approaches 3 and 6 simply shift the state dependence from \mathbf{F}_k^1 to weighting matrix $\mathbf{W}_{u,k}$. Some additional observations are as follows.

- Increasing the magnitude of the standard deviation of the noise in the gyro had the largest impact on the results, as seen in Figure 5.2. While Approaches 2 and 5 have a lower mean RMSE over a wide range of noise magnitudes, these formulations converge much slower than the other approaches. In the end, the left-invariant error definition using the type 1 Jacobians performed best when the noise level was increased.
- In general, a measurement-dependent Jacobian should be avoided when there is a lot of noise in the measurement, as it increases greatly the amount of steps needed to converge
- The right-invariant approaches show improved performance when there is significant noise in the landmark sensor and the accelerometer.

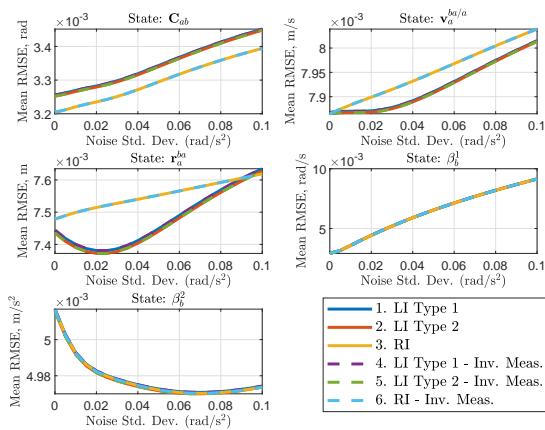


(a) Mean RMSE for each state.

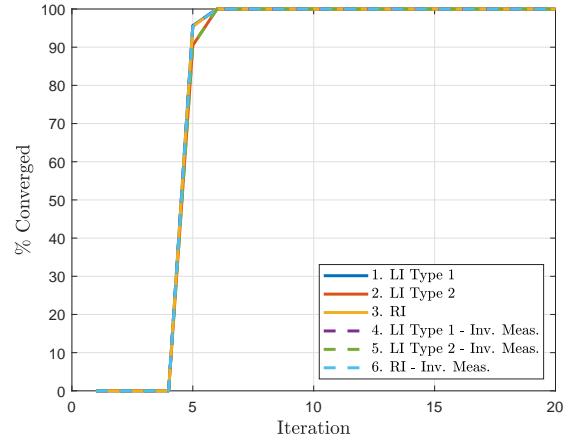


(b) Percentage of trials that have converged at each GN iteration.

Figure 5.3: Results of 25 Monte Carlo trials comparing the various batch SLAM approaches, where the noise in the accelerometer was varied.

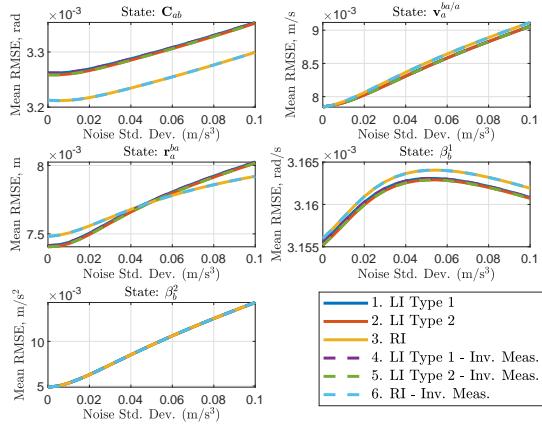


(a) Mean RMSE for each state.

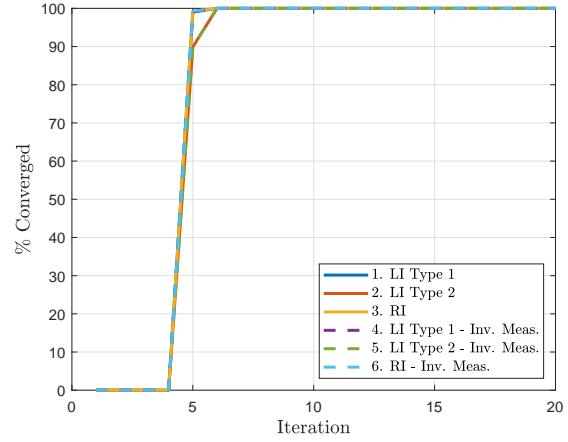


(b) Percentage of trials that have converged at each GN iteration.

Figure 5.4: Results of 25 Monte Carlo trials comparing the various batch SLAM approaches, where the noise in the gyro bias was varied.

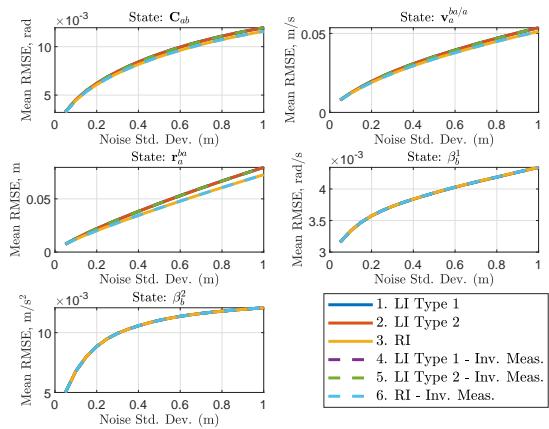


(a) Mean RMSE for each state.

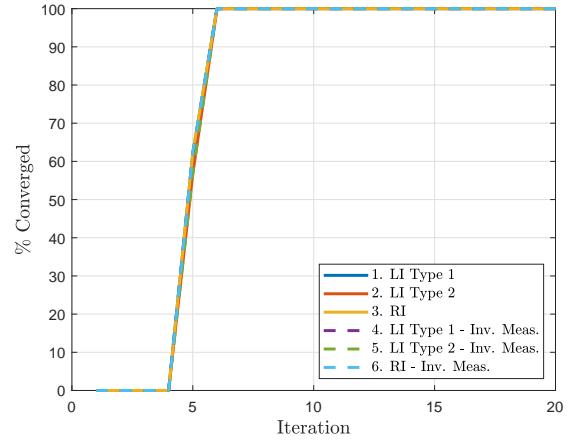


(b) Percentage of trials that have converged at each GN iteration.

Figure 5.5: Results of 25 Monte Carlo trials comparing the various batch SLAM approaches, where the noise in the accelerometer bias was varied.



(a) Mean RMSE for each state.



(b) Percentage of trials that have converged at each GN iteration.

Figure 5.6: Results of 25 Monte Carlo trials comparing the various batch SLAM approaches, where the noise in the landmark sensor was varied.

Chapter 6

Closing Remarks and Future Work

6.1 Conclusions

In this thesis, an in-depth analysis of state estimation in an invariant framework is presented. Through rigorous testing, the advantages and limitations of these different techniques are determined. Furthermore, an extension of invariant filtering theory to the problem of a batch solution to the SLAM problem is presented.

The IEKF is superior to the traditional MEKF in certain situations. It is better suited to problems where the state can be defined on matrix Lie groups, which is the case for many robotics problems. Throughout the simulations presented herein, the performance of the IEKF is on average better than that of the MEKF. However, only particular sample problems are used to illustrate this. It would therefore be irresponsible to state that the IEKF would always perform better than the MEKF. However, certain clear conclusions can be drawn.

First, state-independent Jacobians, such as those obtained in an IEKF, are advantageous in cases where the best estimate of the state is far from the true value. In most situations, this is seen when the initialization is poor. The IEKF's better performance is therefore mostly attributed to better performance in the transient period before the filter reaches steady state. Thus, the IEKF should be the state estimator of choice in applications where the initial state is unknown, and no other initialization scheme is available.

Second, leveraging the invariant framework in batch estimation only has limited advantages. In standard batch estimation, the Jacobians may initially be inaccurate if they depend on the state. However, as the solution converges, the Jacobians will be closer to the true Jacobians, as the error in the state estimate decreases. At this stage, there is minimal difference between a state-independent Jacobian and a Jacobian computed using an accurate state estimate.

6.2 Future Work

In Chapter 4, the IEKF is compared to the MEKF. The MEKF was used a baseline as it is commonly used. However, comparing the IEKF to an iterative version of the MEKF may yield different results. The iterative MEKF improves upon the MEKF by recomputing the Jacobians at each time step until convergence. Furthermore, an iterative version of the IEKF could also be developed. This iterative IEKF would only be useful in scenarios where the process model is not group affine, or the measurement model is not invariant, leading to state dependent Jacobians like the MEKF.

Another avenue to explore would involve using a realistic sensor model in invariant batch SLAM. A future study using a stereo camera model or LIDAR model would also allow the invariant batch SLAM algorithms to be tested on experimental data.

Lastly, a study analyzing the consistency of the IEKF versus other other filtering techniques should be conducted. The IEKF should theoretically be more consistent, as its more accurate Jacobians mean the covariance better captures the underlying distribution. In a similar vein, the impact of unknown disturbances should be studied. The IEKF may be better suited to handle these, once again, due to theoretically exact Jacobians.

Bibliography

- [1] A. Barrau and S. Bonnabel, “The Invariant Extended Kalman Filter as a Stable Observer”, *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2017.
- [2] A. Barrau and S. Bonnabel, “Invariant Kalman Filtering”, *Annu. Rev. Control Robot. Auton. Syst.*, vol. 1, no. 521, pp. 1–5, 2018.
- [3] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.
- [4] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- [5] G. Sibley, “A Sliding Window Filter for SLAM”, University of Southern California, Tech. Rep., 2006, pp. 1–17.
- [6] E. Eade, *Lie Groups for Computer Vision*, 2013.
- [7] B. C. Hall, *Lie Groups, Lie Algebras, and Representations*, 2nd ed. Springer, 2014.
- [8] T. D. Barfoot and P. T. Furgale, “Associating Uncertainty with Three-Dimensional Poses for Use in Estimation Problems”, *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.
- [9] A. Barrau, “Non-linear State Error Based Extended Kalman Filters with Applications to Navigation”, PhD thesis, Mines ParisTech, 2015.
- [10] A. Barrau and S. Bonnabel, “Linear Observed Systems on Groups”, *Systems and Control Letters*, vol. 129, pp. 36–42, 2019.
- [11] P. Hughes, *Spacecraft Attitude Dynamics*, ser. Dover Books on Aeronautical Engineering. Dover Publications, 2012.
- [12] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.
- [13] K. Levenberg, “A Method for the Solution of Certain Non-linear Problems in Least Squares”, *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [14] D. W. Marquardt, “A Method for the Solution of Certain Non-linear Problems in Least Squares”, *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [15] S. Roweis, *Levenberg-Marquardt Optimization*, <https://cs.nyu.edu/~roweis/notes/lm.pdf>, 1996.
- [16] Y. Song and J. W. Grizzle, “The Extended Kalman Filter as a Local Asymptotic Observer for Nonlinear Discrete-Time Systems”, in *1992 American Control Conference*, 1992, pp. 3365–3369.
- [17] A. J. Krener, “The Convergence of the Extended Kalman Filter”, in *Directions in Mathematical Systems Theory and Optimization*, Springer, 2003, pp. 173–182. arXiv: 0212255 [math].
- [18] S. Bonnabel, P. Martin, and P. Rouchon, “Symmetry-Preserving Observers”, *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2514–2526, 2008.

- [19] ——, “Non-linear Symmetry-preserving Observer on Lie Groups”, *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1709–1713, 2009.
- [20] P. Chauchat, A. Barrau, and S. Bonnabel, *Invariant Smoothing on Lie Groups*, <hal-01725847>, 2018.
- [21] S. Bonnabel, *Personal Communication with James R. Forbes*, Nov. 7-9, 2017.
- [22] A. Barrau and S. Bonnabel, “Three Examples of the Stability Properties of the Invariant Extended Kalman Filter”, *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 431–437, 2017.
- [23] ——, “Navigating With Highly Precise Odometry and Noisy GPS: A Case Study”, *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 618–623, 2016.
- [24] M. Barczyk, S. Bonnabel, J.-E. Deschaud, and F. Goulette, “Invariant EKF Design for Scan Matching-aided Localization”, *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2440–2448, 2016. arXiv: [hal-01256770](https://arxiv.org/abs/1603.01256).
- [25] H. Gui and A. H. J. de Ruiter, “Quaternion Invariant Extended Kalman Filtering for Spacecraft Attitude Estimation”, *AIAA Journal of Guidance, Control, and Dynamics*, vol. 41, no. 4, pp. 863–878, 2018.
- [26] R. Hartley, M. G. Jadidi, J. W. Grizzle, and R. M. Eustice, *Contact-aided invariant extended kalman filtering for legged robot state estimation*, arXiv:1805.10410, 2018.
- [27] M. Brossard, A. Barrau, and S. Bonnabel, “Exploiting Symmetries to Design EKFs With Consistency Properties for Navigation and SLAM”, *IEEE Sensors Journal*, vol. 19, no. 4, pp. 1572–1579, 2019.
- [28] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint Kalman filter for vision-aided inertial navigation”, *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3565–3572, 2007.
- [29] S. Heo and C. G. Park, “Consistent EKF-Based Visual-Inertial Odometry on Matrix Lie Group”, *IEEE Sensors Journal*, vol. 18, no. 9, pp. 3780–3788, 2018.
- [30] A. Barrau and S. Bonnabel, *An EKF-SLAM Algorithm with Consistency Properties*, arXiv:1510.06263, 2015.
- [31] T. D. Barfoot, *AER1513 Course Notes, Assignments and Data Sets*, University of Toronto, Institute for Aerospace Studies, 2011.
- [32] S.-C. Hsiung, “Toward Invariant Visual-Inertial State Estimation using Information Sparsification”, PhD thesis, Carnegie Mellon University, 2018.