Bayesian Filtering for Homography Estimation

Arturo Del Castillo Bernal

Department of Mechanical Engineering
McGill University, Montreal

July 2023

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

Acknowledgements

First, I would like to thank Prof. Forbes for being my mentor during my Master's degree. His support went beyond research mentoring, which is already elite in itself, but also his passion and contagious optimism to achieve great things were also elite. Coming to a new country is not an easy task, but Prof. Forbes provided me everything to feel welcome and offer my best version. The DECAR research group led by Prof. Forbes was indispensable to achieve my goals. Special mention to Charles Cossette, Mohammed Ayman Shalaby, Mitchell Cohen and Jonathan Arsenault, who spent a lot of their time answering any question I could have. I feel very thankful to them and the rest of the DECAR team.

I would like to thank Philippe Decoste, who was my main collaborator during my masters. Many valuable technical conversations were had that let this thesis come to fruition, as well as good humor, adventures, and football.

This thesis would not have been possible without the funding of my sponsors in Mexico, Fundacion Esposos Rodriguez and Consejo Nacional de Ciencia y Tecnologia (CONACYT). This work was supported also by the Innovation for Defence Excellence and Security (IDEaS) program, the NSERC Discovery Grant program, and the Canadian Innovation Fund (CFI) program.

Finally, I would like to thank my family and friends. Without my mother's unconditional support, this incredible experience would have never seen the light.

Table of Contents

List of Figures

Figure

List of Tables

Table

Acronyms

EKF	Extended Kalman Filter
IMU	Inertial Measurement Unit
IMM	Interacting Multiple Model
SPKF	Sigma-Point Kalman Filter
GPS	Global Positioning System
PSD	Power Spectral Density
UAV	Unmanned Aerial Vehicle
SLAM	Simultaneous Localization and Mapping
DLT	Direct Linear Transform
SVD	Singular Value Decomposition
RANSAC	Random Sample Consensus
IRLS	Iteratively Reweighted Least Squares
DCS	Dynamic Covariance Scaling
SC	Switchable Constraints
NEES	Normalized Estimation Error Squared

Notation

$\mathbb{R}^n$	The vector space of real n-dimensional vectors
$\mathbb{R}^{m \times n}$	The vector space of real $m \times n$-dimensional matrices
$\lvert a \rvert$	The absolute value of $a \in \mathbb{R}$
$\det \mathbf{A}$	The determinant of $\mathbf{A} \in \mathbb{R}^{n \times n}$
$\lVert \cdot \rVert$	The Euclidean norm of a physical vector
$\mathbf{0}$	Zero matrix
$\mathbf{1}$	Identity matrix
$\text{diag}(\cdot)$	Produces a block-diagonal matrix, such that $\text{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_n)$ contains $\mathbf{A}_1, \ldots, \mathbf{A}_n$ on the main diagonal and $\mathbf{0}$ elsewhere
$(\cdot)^\mathsf{T}$	Transpose
$(\cdot)^{-1}$	Inverse
$(\cdot)^\times$	Skew-symmetric cross operator
$\check{(\cdot)}$	A prior estimate
$\hat{(\cdot)}$	A posterior estimate
$\bar{(\cdot)}$	A mean estimate
$\tilde{(\cdot)}$	The mean estimate from a sensor or motion model, or the mean and covariance associated with a prior estimate
$\mathbf{u}_k$	The k^{th} interoceptive sensor measurement, $k = 0, \ldots, K-1$
$\mathbf{y}_\ell$	The ℓ^{th} exteroceptive sensor measurement, $\ell = 1, \ldots, L$
$\boldsymbol{\mathcal{Q}}$	A power spectral density (PSD) matrix

$\mathbf{Q}_k$	The discrete-time covariance on interoceptive sensor measurement $\mathbf{u}_k$
$\mathbf{R}_\ell$	The covariance on exteroceptive sensor measurement $\mathbf{y}_\ell$
$\mathcal{N}(\mathbf{0}, \mathbf{R})$	Zero-mean, multivariate normal distribution with covariance $\mathbf{R}$
χ^2_k	Chi-squared distribution with k degrees of freedom
$\underrightarrow{r}^{zw}$	The physical vector expressing the position of point z relative to point w
$\mathcal{F}_a$	Reference frame a
$\mathbf{C}_{ab_k}$	A DCM parameterizing the attitude of $\mathcal{F}_a$ relative to $\mathcal{F}_b$
	$\boldsymbol{\omega}_b^{ba}$ The angular velocity of $\mathcal{F}_b$ w.r.t. $\mathcal{F}_a$ resolved in $\mathcal{F}_b$
$\mathbf{r}_a^{zw}$	The physical vector $\underrightarrow{r}^{zw}$ resolved in reference frame $\mathcal{F}_a$.
$\dot{\mathbf{r}}_a^{zw}$	The time rate of change of $\mathbf{r}_a^{zw}$
$\mathbf{n}_a$	A plane normal vector resolved in frame $\mathcal{F}_a$
$\mathbf{d}_a$	Normal distance to a plane resolved in frame $\mathcal{F}_a$
$\mathbf{H}_{ab}$	A homography matrix of $\mathcal{F}_a$ relative to $\mathcal{F}_b$
$\boldsymbol{\Gamma}_{ab}$	Partial group velocity of $\mathbf{H}_{ab}$
$\mathbf{s}_{ab}$	Variable representing $\frac{\dot{\mathbf{r}}_a^{ba}}{d_b}$
G	A matrix Lie group
$\mathfrak{g}$	A matrix Lie algebra, $\mathfrak{g} = T_{\mathbf{1}}G$
$\exp(\cdot)$	The exponential map for matrix Lie groups, $\exp(\cdot) : \mathfrak{g} \to G$
$\log(\cdot)$	The logarithmic map for matrix Lie groups, $\log(\cdot) : G \to \mathfrak{g}$
$(\cdot)^\vee$	The contraction operator for matrix Lie groups, $(\cdot)^\vee : \mathfrak{g} \to \mathbb{R}^d$
$(\cdot)^\wedge$	The expansion operator for matrix Lie groups, $(\cdot)^\wedge : \mathbb{R}^d \to \mathfrak{g}$
$\mathbf{T}_{ab}^{zw}$	An element of matrix Lie group $SE(3)$ describing the vehicle pose. Often abbreviated as $\mathbf{T}_{ab}$.
$\mathbf{X}$	A general Lie group element $\mathbf{X} \in G$
$\boldsymbol{\xi}$	An element of $\mathbb{R}^d$ used to conveniently represent elements of a general matrix

	Lie algebra, $\boldsymbol{\xi}^\wedge \in \mathfrak{g}$
$\boldsymbol{\Sigma}$	The covariance on matrix Lie group element $\mathbf{X}$, $\mathbf{X} = \bar{\mathbf{X}} \exp(\delta\boldsymbol{\xi}^\wedge)$, $\boldsymbol{\Sigma} = \mathrm{E}[\, \delta\boldsymbol{\xi}\, \delta\boldsymbol{\xi}^\mathsf{T}\,]$
$\mathrm{Ad}(\cdot)$	The adjoint operator
$\mathbf{Ad}$	The adjoint matrix
$\mathrm{ad}(\cdot)$	The little adjoint operator
$\mathbf{ad}$	The little adjoint matrix
$\mathbf{J}^\ell$	The left Jacobian of matrix Lie group G
$\mathbf{J}^\mathbf{r}$	The right Jacobian of matrix Lie group G

Abstract

This thesis considers homography estimation in a Bayesian filtering framework using rate-gyro and camera measurements. Homography estimation is a fundamental problem in computer vision. A homography describes the transformation between two images of a planar scene, and is typically estimated using a set of corresponding points in the two images. Environments with planar features are common in many robotics applications, and homography estimation enables the navigation of robots in these environments.

The use of rate-gyro measurements facilitates a more reliable estimate of homography in the presence of occlusions, while a Bayesian filtering approach generates both a homography estimate along with an uncertainty. Uncertainty information opens the door to adaptive filtering approaches, post-processing procedures, and safety protocols. In particular, herein an iterative extended Kalman filter, a sigma-point Kalman Filter and an interacting multiple model (IMM) filter are tested using both simulated and experimental datasets. The IMM is shown to have good consistency properties and better performance when compared to the state-of-the-art homography estimator in simulation, while the IMM has comparable performance when evaluated on experimental data.

Résumé

Cette thèse porte sur l'estimation de l'homographie dans un cadre de filtrage bayésien en utilisant des mesures de gyroscope et de caméra. L'estimation de l'homographie est un problème fondamental en vision par ordinateur. Une homographie décrit la transformation entre deux images d'une scène plane et est généralement estimée à l'aide d'un ensemble de points correspondants dans les deux images. Les environnements avec des caractéristiques planes sont courants dans de nombreuses applications de robotique, et l'estimation de l'homographie permet la navigation des robots dans ces environnements.

L'utilisation des mesures de gyroscope facilite une estimation plus fiable de l'homographie en présence d'occlusions, tandis qu'une approche de filtrage bayésien génère à la fois une estimation d'homographie et une incertitude. Les informations sur l'incertitude ouvrent la voie à des approches de filtrage adaptatif, des procédures de post-traitement et des protocoles de sécurité. En particulier, dans cette thèse, un filtre de Kalman étendu itératif, un filtre de Kalman à points sigma et un modèle multiple interactif (IMM) sont testés à l'aide de jeux de données simulés et expérimentaux. L'IMM présente de bonnes propriétés de cohérence et de meilleures performances par rapport à l'estimateur d'homographie de pointe en simulation, tandis que l'IMM a des performances comparables lorsqu'il est évalué sur des données expérimentales.

Contributions

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

- Chapter 4
  - Employing a Bayesian filtering approach to homography estimation using camera and rate-gyro measurements.

All texts, plots, figures and results in this thesis are original to the author's knowledge unless explicitly stated otherwise.

Chapter 1

Introduction

Autonomous robots, such as unmanned aerial vehicles (UAVs) and autonomous ground vehicles (AGVs), present engineers with challenging navigation, guidance, and control problems. The navigation problem consists of estimating the state of the robot given the sensor readings. The guidance problem consists of determining a trajectory to follow given the knowledge about the robot's state and its environment. The control problem consists of activating the robot's actuators to follow the plan as close as possible. Robust and reliable solutions to these navigation, guidance, and control problems are needed for safe and trustworthy operation of autonomous robots.

The focus of this thesis is navigation. In many robotics applications, the sensors available are low quality due to cost limitations, requiring robust navigation algorithms that can extract information from these low quality sensors. The state of the robot encodes how the robot evolves in a given physical space. The position and the orientation of the robot are the most common ways to represent the state, but the state can also be represented with a homography.

A homography is a mapping that relates two views of the same planar scene. It is exploited in robotics applications when the structure of the environment is sufficiently planar, such as indoor hallways, man made experimental settings, and aerial coverage. Robotics applications that have successfully used homography include visual-servoing [1], image stabilization [2], ego-motion [3, 4], and monocular SLAM initialization [5]. Homographies can better explain the structure of planar scenes and low parallax compared to using the epipolar constraint [6], which is commonly employed for unstructured scenes.

Homography is usually estimated using feature correspondences between a pair of images, such as points, lines, conics or a combination thereof [7, 8]. Direct methods [9] and learning-based approaches [10, 11] are alternative means to estimate homography. All of these methods consider camera measurements independently and ignore any temporal correlations,

thus making them susceptible to failures in the presence of occlusions or lack of feature correspondences.

In [12], temporal information is exploited to estimate homography by designing a nonlinear deterministic observer. Angular velocity measurements are used to provide a more robust estimate of the homography, but at the cost of assuming the camera's velocity is constant parallel to the plane or exponentially converging towards to the plane. Homography is parameterized as an element of the special linear group $SL(3)$ and Lie group properties are exploited in the observer structure to prove local asymptotic stability under various assumptions.

Nonlinear deterministic observers aim to show *a priori* stability properties, and they do not take into account stochastic processes, such as noise in sensor measurements, which is present in all real systems. This thesis leverages the tools of Bayesian filtering that takes into account noise statistics to produce an accurate and consistent estimate, with a covariance describing the error distribution. A Bayesian filtering approach opens the door for procedures such as smoothing [13, Ch. 8], loop-closure detection [14], adaptive approaches [15, Ch. 11], or simply monitor the filter's quality [15, Sec. 5.4].

1.1 Objectives

This thesis presents a Bayesian filtering approach to homography estimation using rate-gyro and camera measurements. The same assumption on the camera's inertial-frame velocity from [12] is made to enable the use of a simple process model. Because this assumption is violated from time to time in practice, focus is placed on a specific type of Bayes filter, the interacting multiple model (IMM) filter. The IMM filter adapts the noise level of the process model when the velocity assumption of the process model is violated. This approach is similar to the IMM application found in [16]. The IMM used here is composed of two iterated extended Kalman filters (EKFs). This thesis clearly demonstrates improved performance, along with consistency, in simulation, as well as comparable performance in experiments, of the IMM relative to a nonlinear deterministic observer. As such, the contribution of this thesis is the combination of

(1) a Bayesian filtering framework,

(2) utilizing the camera velocity assumption of [12] thus providing a simple process model,

(3) considering the IMM structure to account for violations of the camera velocity assumption, and

(4) simulating and experimentally testing the proposed IMM filter.

Deterministic and Bayesian approaches to homography estimation are not mutually exclusive nor competitors. The choice to use one or the other, or perhaps both in “primary" and “back-up" roles, or even in synergy, will be application dependent. In situations where a computationally simple observer is needed, but covariance information is not needed, a deterministic observer is a natural choice. On the other hand, in situations where covariance information is needed, and computational resources not so limited, a Bayesian filter is appropriate. As such, this thesis does not advocate for the displacement of [2, 4, 12] and similar work. Rather, this thesis build on [12] by providing a means to use both a rate-gyro and camera to generate a homography estimate along with a covariance in a way that accounts for the limitations of the assumed process model of [12].

1.2 Organization

This thesis is structured as follows. First, preliminaries related to homography and matrix Lie groups are presented in Chap. 2. Then, Bayesian filtering is presented in Chap. 3. The nuances of estimating homography within a Bayesian framework are presented in Chap. 4. Furthermore, simulated and experimental data are used to validate the proposed algorithm and compare its performance relative to a state-of-the-art nonlinear deterministic observer are presented in Chap. 5. Conclusions and future work directions are presented in Chap. 6. The appendix of this thesis contains a more in-depth explanation of the aforementioned nonlinear deterministic observer.

Chapter 2

Preliminaries

2.1 Overview

Before introducing the problem of homography estimation, certain topics must be reviewed. In particular, the concepts of matrix Lie groups and homography will be reviewed in this chapter.

2.2 Homography

As shown in Figure 2.1, a homography mapping relates features on a plane from two distinct views. A point p_i lying on a plane has coordinates resolved in the camera frame $\mathcal{F}_a$ given by $\boldsymbol{\rho}_a^i = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$. The normalized image coordinates are

$$\mathbf{p}_a^i = \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} = \frac{1}{z}\boldsymbol{\rho}_a^i. \tag{2.1}$$

It is possible to obtain the projection of $\mathbf{p}_a^i$ on the image plane $\mathbf{q}_a^i$ by using the intrinsic parameter matrix $\mathbf{K}$, written as

$$\mathbf{q}_a^i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \mathbf{p}_a^i, \tag{2.2}$$

where f_u and f_v are the horizontal and vertical focal lengths and c_u and c_v are the optical center coordinates in pixels. As the camera's pose changes by $\mathbf{T}_{ab} \in SE(3)$ [17, Sec. 8.1], it is possible to relate a new observation $\mathbf{q}_b^i$ of p_i to $\mathbf{q}_a^i$ by a *homography matrix* $\mathbf{H}_{ab}$ [12], To

derive the homography matrix, the plane constraint resolved in $\mathcal{F}_a$ is utilized,

$$\mathbf{n}_a^\mathsf{T} \boldsymbol{\rho}_a^i + d_a = 0, \tag{2.3}$$

where $\mathbf{n}_a$ is the normalized plane direction resolved in $\mathcal{F}_a$, d_a is the orthogonal distance from the origin of $\mathcal{F}_a$ to the plane and $\boldsymbol{\rho}_a^i$ is the direction from the origin of $\mathcal{F}_a$ to p_i. Now, the pixel coordinates of p_i seen from the origin of $\mathcal{F}_a$ are given by

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{q}_a = \frac{1}{z_a} \mathbf{K} \boldsymbol{\rho}_a^i, \tag{2.4}$$

where z_a is the depth of point p_i. Substitution of (2.4) into (2.3) and solving for z_a yields

$$z_a = -\frac{d_a}{\mathbf{n}_a^\mathsf{T} \mathbf{K}^{-1} \mathbf{q}_a^i}, \tag{2.5}$$

which means that if the plane parameters are known, depth can be inferred from a single camera image. Next consider the following question: if the camera's pose changes by $\mathbf{T}_{ab}$, how will p_i look from the origin of $\mathcal{F}_b$? First, the direction to the point from the origin of $\mathcal{F}_b$ resolved in $\mathcal{F}_b$ is determined considering the previous information,

$$\begin{aligned} \boldsymbol{\rho}_b^i &= \mathbf{T}_{ab}^{-1} \boldsymbol{\rho}_a^i & (2.6) \\ &= \mathbf{C}_{ab}^\mathsf{T} \boldsymbol{\rho}_a^i - \mathbf{C}_{ab}^\mathsf{T} \mathbf{r}_a^{ba}, & (2.7) \end{aligned}$$

where $\mathbf{C}_{ab} \in SO(3)$ is the direction cosine matrix that encodes the orientation of $\mathcal{F}_a$ relative to $\mathcal{F}_b$, $\mathbf{r}_a^{ba}$ is the position of the origin of $\mathcal{F}_b$ relative to the origin of $\mathcal{F}_a$ resolved in $\mathcal{F}_a$, and

$$\mathbf{T}_{ab} = \begin{bmatrix} \mathbf{C}_{ab} & \mathbf{r}_a^{ba} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \in SE(3), \quad \mathbf{T}_{ab}^{-1} = \begin{bmatrix} \mathbf{C}_{ab}^\mathsf{T} & -\mathbf{C}_{ab}^\mathsf{T} \mathbf{r}_a^{ba} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \in SE(3) \tag{2.8}$$

is the robot pose. In pixel coordinates,

$$\begin{aligned} z_b \mathbf{K}^{-1} \mathbf{q}_b^i &= z_a \mathbf{C}_{ab}^\mathsf{T} \mathbf{K}^{-1} \mathbf{q}_a^i - \mathbf{C}_{ab}^\mathsf{T} \mathbf{r}_a^{ba}, & (2.9) \\ \mathbf{q}_b^i &= \frac{z_a}{z_b} \mathbf{K} \mathbf{C}_{ab}^\mathsf{T} \mathbf{K}^{-1} \mathbf{q}_a^i - \frac{z_a}{z_a z_b} \mathbf{K} \mathbf{C}_{ab}^\mathsf{T} \mathbf{r}_a^{ba}, & (2.10) \end{aligned}$$

Using (2.5) in $1/z_a$ in the second term of (2.10) results in

$$\begin{aligned} \mathbf{q}_b^i &= \frac{z_a}{z_b} \mathbf{K} \mathbf{C}_{ab}^\mathsf{T} \mathbf{K}^{-1} \mathbf{q}_a^i + \frac{z_a}{z_b} \mathbf{K} \mathbf{C}_{ab}^\mathsf{T} \mathbf{r}_a^{ba} \frac{\mathbf{n}_a^\mathsf{T} \mathbf{K}^{-1} \mathbf{q}_a^i}{d_a} & (2.11) \\ &= \frac{z_a}{z_b} \mathbf{K} \mathbf{C}_{ab}^\mathsf{T} \left(\mathbf{1} + \frac{1}{d_a} \mathbf{r}_a^{ba} \mathbf{n}_a^\mathsf{T} \right) \mathbf{K}^{-1} \mathbf{q}_a^i & (2.12) \\ &= \mathbf{K} \mathbf{H}_{ba} \mathbf{K}^{-1} \mathbf{q}_a^i, & (2.13) \end{aligned}$$

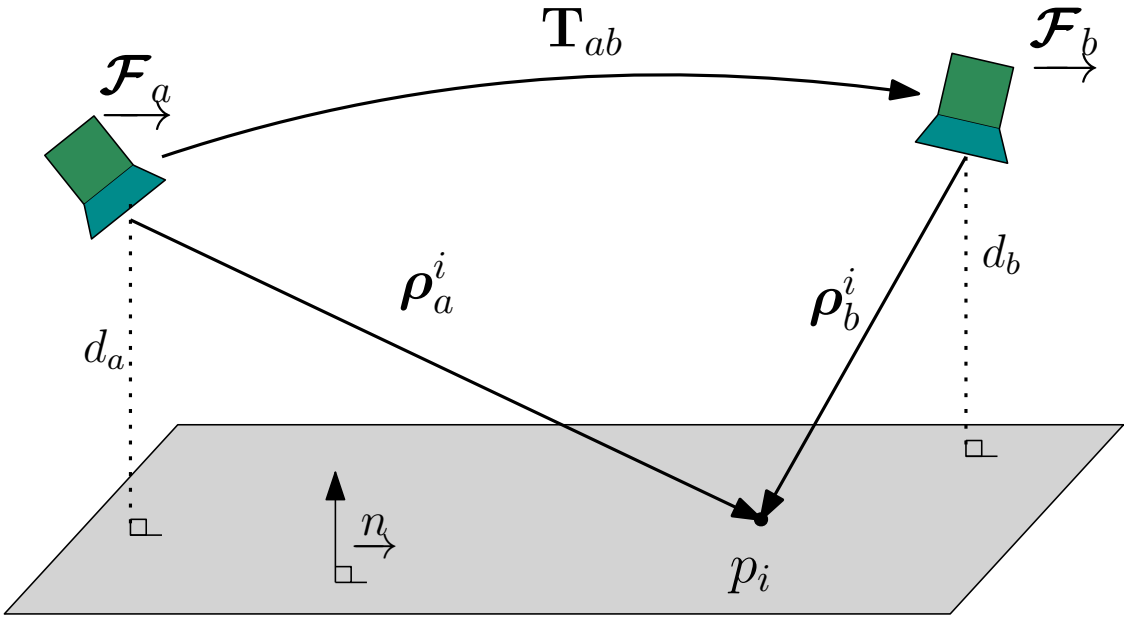


Figure 2.1: A change of perspective from a camera tracking a point on a plane can be described by homography transformation.

where

$$\mathbf{H}_{ba} = \frac{z_a}{z_b}\mathbf{C}_{ab}^{\mathsf{T}}\left(\mathbf{1} + \frac{1}{d_a}\mathbf{r}_a^{ba}\mathbf{n}_a^{\mathsf{T}}\right), \tag{2.14}$$

is the *homography* matrix. As can be observed, by knowing the pose transformation and the plane parameters, the projection of a point onto the image plane in a new position can be predicted. Also, since $\mathbf{q}_b^i$ has homogeneous coordinates, it is common to drop the z_a/z_b scale factor and change it by a scalar $\gamma \triangleq z_a/z_b$ since the true pixel coordinates can be readily recovered by dividing all terms by the bottom entry of the vector.

The homography inverse can be obtained following a similar procedure. It can be verified that

$$\mathbf{H}_{ba}^{-1} = \mathbf{H}_{ab} = \frac{z_b}{z_a}\mathbf{C}_{ab}\left(\mathbf{1} + \frac{1}{d_b}\mathbf{r}_b^{ab}\mathbf{n}_b^{\mathsf{T}}\right). \tag{2.15}$$

2.3 Homography Estimation from Point Correspondences

Given a set of pixel points and a set of corresponding points in two different images, obtained by a camera observing a plane, how is the homography $\mathbf{H}_{ab}$ between the two images obtained?

First, as seen in Sec. 2.2, the homography has 8 degrees of freedom, since it is defined only up to the scalar γ. Since each point correspondence $\mathbf{q}_a^i \leftrightarrow \mathbf{q}_b^i$ in pixel coordinates produces two constraints, coming from the x and y coordinates in the image, a minimum of 4 noncolinear points are required to define and compute a homography.

The most basic algorithm to compute a homography from 4 points is the Direct Linear Transform (DLT) algorithm [6]. First, the relation between a pair of points is

$$c \begin{bmatrix} u_b \\ v_b \\ 1 \end{bmatrix} = \mathbf{H}_{ba} \begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} \tag{2.16}$$

where in this case

$$\mathbf{H}_{ba} \triangleq \mathbf{K}\mathbf{H}_{ba}\mathbf{K}^{-1} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \tag{2.17}$$

is the *image homography*, and c is a scale factor. Equation (2.16) can be rewritten as

$$c \begin{bmatrix} u_b \\ v_b \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 u_a + h_2 v_a + h_3 \\ h_4 u_a + h_5 v_a + h_6 \\ h_7 u_a + h_8 v_a + h_9 \end{bmatrix}. \tag{2.18}$$

Given that c is unknown and unobservable, it is set equal to 1, and the system of equations in (2.18) can be rewritten as

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & u_a & v_a & 1 & -v_b u_a & -v_b v_a & -v_b \\ u_a & v_a & 1 & 0 & 0 & 0 & -u_b u_a & -u_b v_a & -u_b \end{bmatrix}}_{\mathbf{A}_i} \mathbf{h} = \mathbf{0}, \tag{2.19}$$

where $\mathbf{h}$ is the vectorization of all the elements of $\mathbf{H}_{ba}$. Given $i = 1, \ldots, n$ point correspondences of the form (2.19) results in a larger system of equations,

$$\underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_n \end{bmatrix}}_{\mathbf{A}} \mathbf{h} = \mathbf{0}, \tag{2.20}$$

which means solving for homography involves finding the null-space of $\mathbf{A}$. If 4 noncolinear point correspondences are used, $\mathbf{A} \in \mathbb{R}^{8\times 9}$ which means $\mathbf{A}$ has a column rank of 8, and the null-space is 1-dimensional. To find the null-space of $\mathbf{A}$, the singular value decomposition (SVD) of $\mathbf{A}$ is computed, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}}$, and the solution corresponds to the last column of $\mathbf{V}$. If more than 4 point correspondences are used and the points are exact, then $\mathbf{A}$ will still have a column rank of 8, since the extra points are redundant. A more realistic scenario is when

the points are noisy. In the noisy case, the problem can be set up as a least squares problem,

$$\mathbf{h}^\star = \arg\min_{\mathbf{h}} \quad \|\mathbf{A}\mathbf{h}\|_2 , \tag{2.21}$$

$$\text{such that} \quad \|\mathbf{h}\|_2 = 1, \tag{2.22}$$

where the constraint is applied to avoid the trivial solution of $\mathbf{h}^\star = \mathbf{0}$. To solve this least squares problem, let $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$. Then, $\|\mathbf{A}\mathbf{h}\|_2 = \left\|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}\mathbf{h}\right\|_2 = \left\|\mathbf{\Sigma}\mathbf{V}^\mathsf{T}\mathbf{h}\right\|_2$. Given $\|\mathbf{h}\|_2 = \left\|\mathbf{V}^\mathsf{T}\mathbf{h}\right\|_2$, a change of variable is proposed, $\mathbf{y} = \mathbf{V}^\mathsf{T}\mathbf{h}$, which results in

$$\mathbf{y}^\star = \arg\min_{\mathbf{y}} \quad \|\mathbf{\Sigma}\mathbf{y}\|_2 , \tag{2.23}$$

$$\text{such that} \quad \|\mathbf{y}\|_2 = 1. \tag{2.24}$$

Since $\mathbf{\Sigma}$ is a diagonal matrix with its entries in descending order, the solution must be $\mathbf{y}^\star = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}^\mathsf{T}$. It follows that $\mathbf{h}^\star$ equals the last column of $\mathbf{V}$.

2.4 Matrix Lie Groups

Matrix Lie groups are relevant in many robotics applications because they can better represent states, such as orientation and poses, and measurement models, achieving thus better performance, consistency and stability [18]. In this thesis, the focus will be on the special linear group, $SL(3)$, which is intimately related to the homography estimation problem.

2.4.1 Overview

A Lie group G is a smooth manifold whose elements, given a group operation $\circ : G \times G \to G$, satisfy the group axioms [18]. A Matrix Lie group is a Lie group whose elements are matrices and the group operation is matrix multiplication [19]. For any G, there exists an associated Lie algebra $\mathfrak{g}$, a vector space identifiable with elements of $\mathbb{R}^m$, where m is referred to as the degrees of freedom of G. The Lie algebra is related to the group through the exponential and logarithmic maps, denoted $\exp : \mathfrak{g} \to G$ and $\log : G \to \mathfrak{g}$. For matrix Lie groups, the exponential map is the matrix exponential and the logarithmic map is the matrix natural logarithm. The "vee" and "wedge" operators are denoted $(\cdot)^\vee : \mathfrak{g} \to \mathbb{R}^m$ and $(\cdot)^\wedge : \mathbb{R}^m \to \mathfrak{g}$, and are used to associate group elements with vectors with

$$\mathbf{X} = \exp(\boldsymbol{\xi}^\wedge), \qquad \boldsymbol{\xi} = \log(\mathbf{X})^\vee, \tag{2.25}$$

where $\mathbf{X} \in G, \boldsymbol{\xi} \in \mathbb{R}^m$. The adjoint operator $\mathrm{Ad} : G \times \mathfrak{g} \to \mathfrak{g}$ is defined as

$$\mathrm{Ad}(\mathbf{X})\mathbf{\Xi} = \mathbf{X}\mathbf{\Xi}\mathbf{X}^{-1}, \tag{2.26}$$

where $\boldsymbol{\Xi} \in \mathfrak{g}$. The adjoint matrix $\mathbf{Ad}(\mathbf{X})$ can be obtained by $(\mathrm{Ad}(\mathbf{X})\boldsymbol{\Xi})^\vee = \mathbf{Ad}(\mathbf{X})\boldsymbol{\xi}$. The Lie bracket $[\cdot,\cdot] : \mathfrak{sl}(3) \times \mathfrak{sl}(3) \to \mathfrak{sl}(3)$ is defined as

$$[\boldsymbol{\Xi}_1, \boldsymbol{\Xi}_2] = \boldsymbol{\Xi}_1\boldsymbol{\Xi}_2 - \boldsymbol{\Xi}_2\boldsymbol{\Xi}_1, \tag{2.27}$$

and a little adjoint matrix can be defined also as, $[\boldsymbol{\Xi}_1, \boldsymbol{\Xi}_2]^\vee = \mathbf{ad}(\boldsymbol{\xi}_1)\boldsymbol{\xi}_2$. The $(\cdot)^\odot : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ operator is also relevant in this thesis and it is defined as $\boldsymbol{\xi}^\wedge \mathbf{p} = \mathbf{p}^\odot \boldsymbol{\xi}$, where $\mathbf{p} \in \mathbb{R}^n$.

The most common Lie groups appearing in robotics are $SO(n)$, representing rotations in n-dimensional space, and $SE(n)$, representing poses. In this thesis, the focus is in the special linear group, $SL(3)$.

2.4.2 Uncertainty Representation

In this thesis, $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is used to denote a Gaussian random variable $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. For a random variable $\mathbf{X}$ evolving in a Lie group G, a "Gaussian" distribution can be represented on a vector space by two different ways,

$$\mathbf{X} = \exp(\boldsymbol{\xi}^\wedge)\bar{\mathbf{X}}, \tag{2.28}$$

$$\mathbf{X} = \bar{\mathbf{X}}\exp(\boldsymbol{\xi}^\wedge), \tag{2.29}$$

where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ [20], and the two uncertainty representations are defined as right and left-invariant respectively, although it can also be said they are left and right perturbations, respectively. Note that $\mathbf{X}$ is not normally distributed. Also, two additional related uncertainty definitions can be added,

$$\mathbf{X} = \exp(-\boldsymbol{\xi}^\wedge)\bar{\mathbf{X}}, \tag{2.30}$$

$$\mathbf{X} = \bar{\mathbf{X}}\exp(-\boldsymbol{\xi}^\wedge). \tag{2.31}$$

These two additional definitions are also right and left-invariant, respectively.

2.4.3 Linearization

An element of a group G can be represented as

$$\mathbf{X} = \exp(\boldsymbol{\xi}^\wedge). \tag{2.32}$$

The matrix exponential can be written as

$$\exp(\boldsymbol{\xi}^\wedge) = \sum_{i=0}^{\infty} \frac{1}{k!} (\boldsymbol{\xi})^\wedge. \tag{2.33}$$

For a small $\boldsymbol{\xi}$, terms of second order or higher can be neglected, yielding

$$\mathbf{X} \approx \mathbf{1} + \boldsymbol{\xi}^\wedge, \tag{2.34}$$

which will prove very useful in this thesis when linearizing various functions.

2.4.4 Lie Groups Derivatives

Recall that for a multivariate function $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian matrix is defined as [18]

$$\mathbf{J}(\bar{\mathbf{x}}) = \left.\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\bar{\mathbf{x}}} \triangleq \left.\begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial g_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial g_m(\mathbf{x})}{\partial x_n} \end{bmatrix}\right|_{\bar{\mathbf{x}}} \tag{2.35}$$

$$= \begin{bmatrix} \mathbf{j}_1 & \cdots & \mathbf{j}_n \end{bmatrix}, \tag{2.36}$$

where each of the columns $\mathbf{j}_i$ are defined in turn as

$$\mathbf{j}_i = \left.\frac{\partial \mathbf{g}(\mathbf{x})}{\partial x_i}\right|_{\bar{\mathbf{x}}} \triangleq \lim_{h \to 0} \frac{\mathbf{g}(\bar{\mathbf{x}} + h\mathbf{e}_i) - \mathbf{g}(\bar{\mathbf{x}})}{h}, \tag{2.37}$$

where $\mathbf{e}_i$ is the i^{th} vector of the natural basis of $\mathbb{R}^n$. These columns $\mathbf{j}_i$ can be compacted as,

$$\mathbf{J}(\bar{\mathbf{x}}) = \lim_{\mathbf{h} \to 0} \frac{\mathbf{g}(\bar{\mathbf{x}} + \mathbf{h}) - \mathbf{g}(\bar{\mathbf{x}})}{\mathbf{h}}, \tag{2.38}$$

where $\mathbf{h} \in \mathbb{R}^n$. This definition is only for compactness, since division by the vector $\mathbf{h}$ is undefined, and the computation of the Jacobian is done utilizing (2.37). Using these facts, it is possible to present the group Jacobian of a matrix Lie group G,

$$\mathbf{J}^{\ell}\left(\bar{\boldsymbol{\xi}}\right) = \left.\frac{\partial \log\left(\exp\left(\bar{\boldsymbol{\xi}}^{\wedge} + \boldsymbol{\xi}^{\wedge}\right)\exp\left(-\boldsymbol{\xi}^{\wedge}\right)\right)^{\vee}}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=0}, \tag{2.39}$$

$$\mathbf{J}^{r}\left(\bar{\boldsymbol{\xi}}\right) = \left.\frac{\partial \log\left(\exp\left(-\bar{\boldsymbol{\xi}}^{\wedge}\right)\exp\left(\bar{\boldsymbol{\xi}}^{\wedge} + \boldsymbol{\xi}^{\wedge}\right)\right)^{\vee}}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=0}, \tag{2.40}$$

where $\mathbf{J}^{\ell}(\cdot)$, $\mathbf{J}^{r}$ are the left and right group Jacobians, respectively. When the function $\mathbf{g} : G \to \mathbb{R}^m$ takes a Lie group element, using a right-invariant uncertainty representation, the left Jacobian is defined as

$$\mathbf{J}(\bar{\mathbf{X}}) = \left.\frac{D\mathbf{g}(\mathbf{X})}{D\mathbf{X}}\right|_{\bar{\mathbf{X}}} \triangleq \left.\frac{\partial \mathbf{g}(\exp(\boldsymbol{\xi}^{\wedge})\bar{\mathbf{X}})}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=\mathbf{0}} \tag{2.41}$$

$$= \lim_{\boldsymbol{\xi} \to 0} \frac{\mathbf{g}(\exp(\boldsymbol{\xi}^{\wedge})\bar{\mathbf{X}}) - \mathbf{g}(\bar{\mathbf{X}})}{\boldsymbol{\xi}}. \tag{2.42}$$

A right jacobian can be obtained by using a left-invariant uncertainty representation.

2.4.5 Special Linear Group

There are 6 degrees of freedom associated with rotation and translation, and 3 degrees of freedom associated with relative distance. However, a homography matrix only has 8 degrees

of freedom, and not 9, because it is defined only up to the scalar γ. One way to parameterize homography is by the Special Linear Group $SL(3)$,

$$SL(3) \equiv \{\mathbf{H} \in \mathbb{R}^{3\times 3} \mid \det \mathbf{H} = 1\}. \tag{2.43}$$

Any non-singular matrix $\mathbf{X} \in \mathbb{R}^{3\times 3}$ can be projected to $SL(3)$ by

$$\frac{\mathbf{X}}{(\det \mathbf{X})^{1/3}} \in SL(3). \tag{2.44}$$

As with any Lie group, there is a Lie algebra associated to $SL(3)$,

$$\mathfrak{sl}(3) \equiv \{\boldsymbol{\Xi} \in \mathbb{R}^{3\times 3} \mid \mathrm{tr}(\boldsymbol{\Xi}) = 0\}. \tag{2.45}$$

A vector can be uniquely associated to an element of $\mathfrak{sl}(3)$ by $(\cdot)^\wedge : \mathbb{R}^8 \to \mathfrak{sl}(3)$, which is defined as [21],

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \xi_4 + \xi_5 & -\xi_3 + \xi_6 & \xi_1 \\ \xi_3 + \xi_6 & \xi_4 - \xi_5 & \xi_2 \\ \xi_7 & \xi_8 & -2\xi_4 \end{bmatrix}. \tag{2.46}$$

Given this parametrization, the adjoint matrix can be defined as

$$\mathbf{Ad}(\mathbf{H}) = \mathbf{A}^\star \mathbf{B}(\mathbf{H}) \mathbf{A}, \tag{2.47}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{9\times 8}, \tag{2.48}$$

$$\mathbf{A}^\star = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{8\times 9}, \tag{2.49}$$

$$\mathbf{B}(\mathbf{H}) = \begin{bmatrix} \mathbf{H}_{11}\mathbf{H}^{-\mathsf{T}} & \mathbf{H}_{12}\mathbf{H}^{-\mathsf{T}} & \mathbf{H}_{13}\mathbf{H}^{-\mathsf{T}} \\ \mathbf{H}_{21}\mathbf{H}^{-\mathsf{T}} & \mathbf{H}_{22}\mathbf{H}^{-\mathsf{T}} & \mathbf{H}_{23}\mathbf{H}^{-\mathsf{T}} \\ \mathbf{H}_{31}\mathbf{H}^{-\mathsf{T}} & \mathbf{H}_{32}\mathbf{H}^{-\mathsf{T}} & \mathbf{H}_{33}\mathbf{H}^{-\mathsf{T}} \end{bmatrix}. \tag{2.50}$$

The little adjoint matrix associated with $SL(3)$ is defined as well, that being

$$\mathbf{ad}(\boldsymbol{\xi}) = \begin{bmatrix} 3\xi_4+\xi_5 & -(\xi_3-\xi_6) & \xi_2 & -3\xi_1 & -\xi_1 & -\xi_2 & 0 & 0 \\ \xi_3+\xi_6 & 3\xi_4-\xi_5 & -\xi_1 & -3\xi_2 & \xi_2 & -\xi_1 & 0 & 0 \\ \frac{\xi_8}{2} & -\frac{\xi_7}{2} & 0 & 0 & 2\xi_6 & -2\xi_5 & \frac{\xi_2}{2} & -\frac{\xi_1}{2} \\ -\frac{\xi_7}{2} & -\frac{\xi_8}{2} & 0 & 0 & 0 & 0 & \frac{\xi_1}{2} & \frac{\xi_2}{2} \\ -\frac{\xi_7}{2} & \frac{\xi_8}{2} & 2\xi_6 & 0 & 0 & -2\xi_3 & \frac{\xi_1}{2} & -\frac{\xi_2}{2} \\ -\frac{\xi_8}{2} & -\frac{\xi_7}{2} & -2\xi_5 & 0 & 2\xi_3 & 0 & \frac{\xi_2}{2} & \frac{\xi_1}{2} \\ 0 & 0 & \xi_8 & 3\xi_7 & \xi_7 & \xi_8 & -3\xi_4-\xi_5 & -(\xi_3+\xi_6) \\ 0 & 0 & -\xi_7 & 3\xi_8 & -\xi_8 & \xi_7 & \xi_3-\xi_6 & -(3\xi_4-\xi_5) \end{bmatrix}. \tag{2.51}$$

The projection matrix $\mathbf{B}$ that projects $\mathbf{w} \in \mathbb{R}^3$ to $\mathbb{R}^8$, such that $(\mathbf{B}\mathbf{w})^\wedge = \mathbf{w}^\wedge_{\mathfrak{sl}(3)} \in \mathfrak{sl}(3)$, can be derived by noting that $\mathbf{w}^{\times^\vee} = \begin{bmatrix} w_2 & -w_1 & w_3 & 0 & 0 & 0 & -w_2 & w_1 \end{bmatrix}^\mathsf{T}$. Thus, the projection

matrix is

$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{2.52}$$

2.5 Homography Kinematics

From [12], it is possible to find an expression for the kinematics of $\mathbf{H}_{ab}$. Consider the frame $\mathcal{F}_a$ as an inertial frame, which will serve as reference frame and $\mathcal{F}_b$ as the body frame. The body kinematics are given by

$$\dot{\mathbf{C}}_{ab} = \mathbf{C}_{ab}\boldsymbol{\omega}_b^{ba\times}, \tag{2.53a}$$

$$\dot{\mathbf{r}}_a^{ba} = \mathbf{C}_{ab}\mathbf{v}_b^{ba}, \tag{2.53b}$$

where $\boldsymbol{\omega}_b^{ba}$ is the angular velocity of the body w.r.t. the inertial frame resolved in $\mathcal{F}_b$, $\mathbf{v}_b^{ba}$ is the linear velocity of the body w.r.t. the inertial frame resolved in $\mathcal{F}_b$, and the operator $(\cdot)^\times : \mathbb{R}^3 \to \mathfrak{so}(3)$ where $\mathfrak{so}(3)$ is the Lie algebra of $SO(3) \subset SL(3)$. It can be verified that

$$\mathbf{n}_b = \mathbf{C}_{ab}^\mathsf{T}\mathbf{n}_a, \tag{2.54}$$

$$d_b = d_a + \mathbf{n}_a^\mathsf{T}\mathbf{r}_a^{ba}, \tag{2.55}$$

therefore, since the plane parameters relative to the reference frame do not change, meaning $\dot{\mathbf{n}}_a = \mathbf{0}, \dot{d}_a = 0$. It can be deduced that

$$\dot{d}_b = \mathbf{n}_a^\mathsf{T}\dot{\mathbf{r}}_a^{ba} \tag{2.56}$$

$$= \mathbf{n}_b^\mathsf{T}\mathbf{C}_{ab}^\mathsf{T}\mathbf{C}_{ab}\mathbf{v}_b^{ba} \tag{2.57}$$

$$= \mathbf{n}_b^\mathsf{T}\mathbf{v}_b^{ba}. \tag{2.58}$$

With these results in mind, it is possible to find an expression for the kinematics of $\mathbf{H}_{ab} = \gamma\mathbf{C}_{ab}\left(\mathbf{1} + \frac{1}{d_b}\mathbf{r}_b^{ab}\mathbf{n}_b^\mathsf{T}\right)$,

$$\dot{\mathbf{H}}_{ab} = \gamma\dot{\mathbf{C}}_{ab}\left(\mathbf{1} + \frac{1}{d_b}\mathbf{r}_b^{ab}\mathbf{n}_b^\mathsf{T}\right) + \gamma\mathbf{C}_{ab}\left(\frac{\dot{\mathbf{r}}_b^{ab}\mathbf{n}_b^\mathsf{T} + \mathbf{r}_b^{ab}\dot{\mathbf{n}}_b^\mathsf{T}}{d_b} - \frac{\dot{d}_b\mathbf{r}_b^{ab}\mathbf{n}_b^\mathsf{T}}{d_b^2}\right) + \frac{\dot{\gamma}}{\gamma}\mathbf{H}_{ab} \tag{2.59}$$

$$= \gamma\mathbf{C}_{ab}\boldsymbol{\omega}_b^{ba\times}\left(\mathbf{1} + \frac{1}{d_b}\mathbf{r}_b^{ab}\mathbf{n}_b^\mathsf{T}\right) + \gamma\mathbf{C}_{ab}\left(\frac{(-\boldsymbol{\omega}_b^{ba\times}\mathbf{r}_b^{ab} - \mathbf{v}_b^{ba})\mathbf{n}_b^\mathsf{T} + \mathbf{r}_b^{ab}\mathbf{n}_b^\mathsf{T}\boldsymbol{\omega}_b^{ba\times}}{d_b} - \frac{\mathbf{r}_b^{ab}\mathbf{n}_b^\mathsf{T}\mathbf{v}_b^{ba}\mathbf{n}_b^\mathsf{T}}{d_b^2}\right) + \frac{\dot{\gamma}}{\gamma}\mathbf{H}_{ab} \tag{2.60}$$

$$= \gamma\mathbf{C}_{ab}\left(\left[\mathbf{1} + \frac{\mathbf{r}_b^{ab}\mathbf{n}_b^\mathsf{T}}{d_b}\right]\boldsymbol{\omega}_b^{ba\times} - \left[\mathbf{1} + \frac{\mathbf{r}_b^{ab}\mathbf{n}_b^\mathsf{T}}{d_b}\right]\frac{\mathbf{v}_b^{ba}\mathbf{n}_b^\mathsf{T}}{d_b}\right) + \frac{\dot{\gamma}}{\gamma}\mathbf{H}_{ab} \tag{2.61}$$

$$= \mathbf{H}_{ab}\left(\boldsymbol{\omega}_b^{ba\times} - \frac{\mathbf{v}_b^{ba}\mathbf{n}_b^\mathsf{T}}{d_b} + \frac{\dot{\gamma}}{\gamma}\mathbf{1}\right) \tag{2.62}$$

$$= \mathbf{H}_{ab}\mathbf{U}_b^{ab}, \tag{2.63}$$

where $\mathbf{U}_b^{ab} \in \mathfrak{sl}(3)$ and is referred as the group velocity, which has the interpretation of being an infinitesimal variation of $\mathbf{H}_{ab}$. The fact that $\mathrm{tr}\,(\boldsymbol{\Xi}) = 0$ for $\boldsymbol{\Xi} \in \mathfrak{sl}(3)$ can be exploited to determine the value of the undefined scalar $\frac{\dot{\gamma}}{\gamma}$. Noting that $\mathrm{tr}\left(\mathbf{U}_b^{ab}\right) = 0$, it follows that

$$0 = \mathrm{tr}\left(\boldsymbol{\omega}_b^{ba\times} - \frac{\mathbf{v}_b^{ba}\mathbf{n}_b^\mathsf{T}}{d_b} + \frac{\dot{\gamma}}{\gamma}\mathbf{1}\right) \tag{2.64}$$

$$= -\frac{\mathbf{n}_b^\mathsf{T}\mathbf{v}_b^{ba}}{d_b} + 3\frac{\dot{\gamma}}{\gamma}, \tag{2.65}$$

which means

$$\mathbf{U}_b^{ab} = \left(\boldsymbol{\omega}_b^{ba\times} + \boldsymbol{\Gamma}_b^{ba}\right), \qquad \boldsymbol{\Gamma}_b^{ba} = -\frac{\mathbf{v}_b^{ba}\mathbf{n}_b^\mathsf{T}}{d_b} + \frac{\mathbf{n}_b^\mathsf{T}\mathbf{v}_b^{ba}}{3d_b}\mathbf{1}, \tag{2.66}$$

where $\mathbf{1}$ is the identity matrix. As can be noticed, $\boldsymbol{\Gamma}_b^{ba}$ depends on the plane parameters $\mathbf{n}_b$ and d_b, which are unmeasurable. A simple kinematic model can be derived if $\mathbf{s}_{ab} \triangleq \frac{\dot{\mathbf{r}}_a^{ba}}{d_b}$ is assumed to be constant [12]. Using $\mathbf{v}_b^{ba} = \mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}$,

$$\dot{\boldsymbol{\Gamma}}_b^{ba} = -\frac{\dot{\mathbf{C}}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}\mathbf{n}_b^\mathsf{T} + \mathbf{C}_{ab}^\mathsf{T}\ddot{\mathbf{r}}_a^{ba}\mathbf{n}_b^\mathsf{T} + \mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}\dot{\mathbf{n}}_b^\mathsf{T}}{d_b} + \frac{\dot{d}_b\mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}\mathbf{n}_b^\mathsf{T}}{d_b^2} \tag{2.67}$$

$$+ \frac{\dot{\mathbf{n}}_b^\mathsf{T}\mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba} + \mathbf{n}_b^\mathsf{T}\dot{\mathbf{C}}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba} + \mathbf{n}_b^\mathsf{T}\mathbf{C}_{ab}^\mathsf{T}\ddot{\mathbf{r}}_a^{ba}}{3d_b}\mathbf{1} - \frac{\dot{d}_b\mathbf{n}_b^\mathsf{T}\mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}}{9d_b^2}\mathbf{1}$$

$$= -\frac{\dot{\mathbf{C}}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}\mathbf{n}_b^\mathsf{T} + \mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}\dot{\mathbf{n}}_b^\mathsf{T}}{d_b} + \frac{\dot{\mathbf{n}}_b^\mathsf{T}\mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba} + \mathbf{n}_b^\mathsf{T}\dot{\mathbf{C}}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}}{3d_b}\mathbf{1}, \tag{2.68}$$

where the terms eliminated are due to the assumption on $\mathbf{s}_{ab}$. Then, substituting $\dot{\mathbf{C}}_{ab}$ using (2.53a), and $\dot{\mathbf{n}}_b$ using (2.53a) and (2.54),

$$\dot{\mathbf{\Gamma}}_b^{ba} = \frac{\boldsymbol{\omega}_b^{ba\times}\mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}\mathbf{n}_b^\mathsf{T} - \mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}\mathbf{n}_b^\mathsf{T}\boldsymbol{\omega}_b^{ba\times}}{d_b} + \frac{\mathbf{n}_b^\mathsf{T}\boldsymbol{\omega}_b^{ba\times}\mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba} - \mathbf{n}_b^\mathsf{T}\boldsymbol{\omega}_b^{ba\times}\mathbf{C}_{ab}^\mathsf{T}\dot{\mathbf{r}}_a^{ba}}{3d_b}\mathbf{1} \tag{2.69}$$

$$= \frac{\boldsymbol{\omega}_b^{ba\times}\mathbf{v}_b^{ba}\mathbf{n}_b^\mathsf{T} - \mathbf{v}_b^{ba}\mathbf{n}_b^\mathsf{T}\boldsymbol{\omega}_b^{ba\times}}{d_b} \tag{2.70}$$

$$= \left[\mathbf{\Gamma}_b^{ba}, \boldsymbol{\omega}_b^{ba\times}\right]. \tag{2.71}$$

It can be noted that $\mathbf{\Gamma}_{ab} \in \mathfrak{sl}(3)$. This simplified process model depends only on angular velocity, which is easily measured by a rate-gyro. However, assuming $\mathbf{s}_a$ is constant restricts the trajectory of the camera to have constant velocity parallel to the plane, or to exponentially converge towards the plane [12]. Without the assumption that $\mathbf{s}_a$ is constant, the kinematic model for $\mathbf{\Gamma}_b^{ba}$ is

$$\dot{\mathbf{\Gamma}}_b^{ba} = \left[\mathbf{\Gamma}_b^{ba}, \boldsymbol{\omega}_b^{ba\times}\right] + \mathbf{\Gamma}_b^{ba\,2} + \frac{\mathbf{C}_{ab}^\mathsf{T}\ddot{\mathbf{r}}_a^{ba}\mathbf{n}_b^\mathsf{T}}{d_b} - \frac{\mathbf{n}_b^\mathsf{T}\mathbf{C}_{ab}^\mathsf{T}\ddot{\mathbf{r}}_a^{ba}}{3d_b}\mathbf{1}. \tag{2.72}$$

Chapter 3

Bayesian Filtering

3.1 Overview

Bayesian inference provides the tools to make optimal decisions when uncertainties are provided. In the context of state estimation in robotics applications, Bayesian filters are used to find the optimal state from noisy, uncertain measurements. A probabilistic state space model is needed, composed of two parts [13].

1. **Process model**: The process model encodes the prior beliefs of how the state evolves in time. Markov assumption is used, implying that

$$\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \tag{3.1}$$

   where $\mathbf{u}_{k-1}$ is a process model input that typically provides information about the movement of the robot, such as accelerometer and rate-gyro readings.

2. **Measurement model**: The measurement model encodes the distribution of the measurements given the state,

$$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}_k). \tag{3.2}$$

   The measurements provide information of the state of the robot with respect to the environment, such as camera measurements or Global Positioning System (GPS) measurements.

The purpose of a Bayesian filter is to estimate the posterior distribution of the current state given the whole history of measurements and inputs,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0). \tag{3.3}$$

Notation-wise, unless stated otherwise, the indefinite integral sign is taken to be from $-\infty$ to

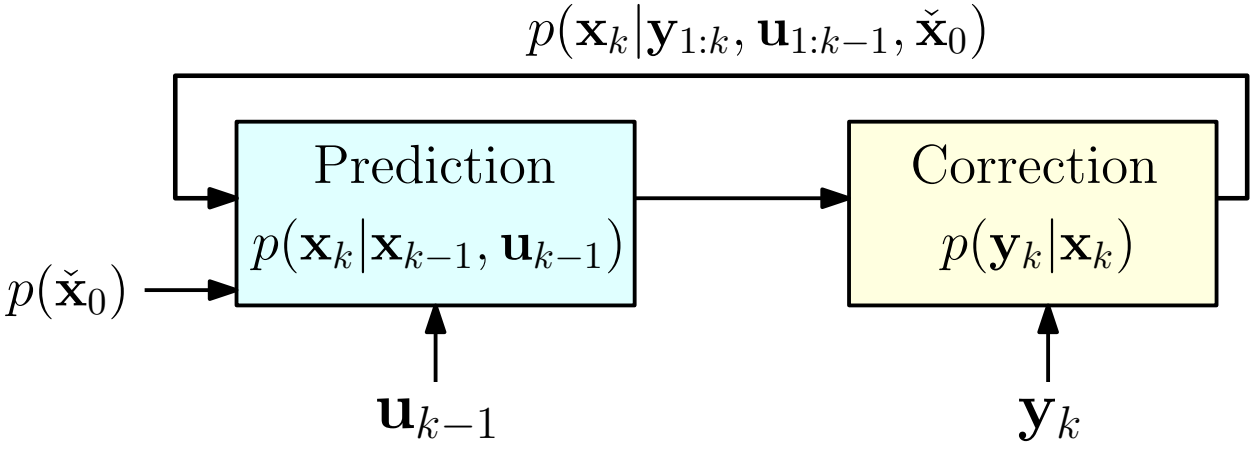


Figure 3.1: General structure of a Bayesian filter.

∞,

$$\int \mathbf{f}(\mathbf{x})\,\mathrm{d}\mathbf{x} \triangleq \int_{-\infty}^{\infty} \mathbf{f}(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{3.4}$$

with $\mathbf{f}(\mathbf{x})$ representing any integrable function. The structure of a Bayesian filter is shown in Figure 3.1. It consists of a prediction and correction steps. Initializing the state with a prior guess $p(\check{\mathbf{x}}_0)$, the exact equations for these steps are

1. **Prediction step**: Before a new measurement $\mathbf{y}_k$ arrives, the state's distribution is predicted using the process model,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0)\,\mathrm{d}\mathbf{x}_{k-1}. \tag{3.5}$$

2. **Correction step**: When a new measurement $\mathbf{y}_k$ arrives, the distribution is updated by

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0) = \frac{1}{c} p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0), \tag{3.6}$$

   where c is a normalization constant.

3.2 Gaussian Filter

In the context of filtering, it is difficult to compute the whole distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0)$, so a Gaussian filter assumes that the state distribution is Gaussian,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0) \approx \mathcal{N}\left(\mathbf{x}_k|\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k\right). \tag{3.7}$$

Therefore, only a mean and covariance are computed, ignoring higher order moments. This assumption suffices in many applications, but can be limiting in some cases, especially when the state distribution is multimodal [13].

To derive the Gaussian filter and its variants, the process model and measurement models can be posed alternatively as discrete-time nonlinear state space models, instead of a probabilistic state space model as in (3.1) and (3.2). In particular, the state-space form of the process and measurement models, respectively, are

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}), \quad \mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}), \tag{3.8}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{v}_k), \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{3.9}$$

where $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are zero-mean Gaussian noise processes with covariance matrices $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$, respectively.

Consider a Gaussian random variable $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ that is transformed by a nonlinearity $\mathbf{y} = \mathbf{g}(\mathbf{x})$. The mean $\bar{\mathbf{y}}$ and covariance $\mathbf{S}$ of such transformation are given by

$$\bar{\mathbf{y}} = \int \mathbf{g}(\mathbf{x}) \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \, \mathrm{d}\mathbf{x}, \tag{3.10a}$$

$$\mathbf{S} = \int (\mathbf{g}(\mathbf{x}) - \bar{\mathbf{y}}) (\mathbf{g}(\mathbf{x}) - \bar{\mathbf{y}})^{\mathsf{T}} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \, \mathrm{d}\mathbf{x}. \tag{3.10b}$$

Using the Gaussian assumption that $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}, \check{\mathbf{x}}_0) \approx \mathcal{N}\left(\mathbf{x}_k|\check{\mathbf{x}}_k, \check{\mathbf{P}}_k\right)$, it follows that the **prediction step** of the Gaussian filter is

$$\check{\mathbf{x}}_k = \int \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \mathcal{N}\left(\mathbf{x}_{k-1}|\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1}\right) \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}) \, \mathrm{d}\mathbf{x}_{k-1} \, \mathrm{d}\mathbf{w}_{k-1}, \tag{3.11}$$

$$\begin{aligned} \check{\mathbf{P}}_k = \int &\left(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) - \check{\mathbf{x}}_k\right) \left(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) - \check{\mathbf{x}}_k\right)^{\mathsf{T}} \times \\ &\mathcal{N}\left(\mathbf{x}_{k-1}|\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1}\right) \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}) \, \mathrm{d}\mathbf{x}_{k-1} \, \mathrm{d}\mathbf{w}_{k-1}. \end{aligned} \tag{3.12}$$

In the correction step, given that the joint probability density function $p(\mathbf{x}_k, \mathbf{y}_k)$ of the state and measurement is

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} \middle| \begin{bmatrix} \check{\mathbf{x}}_k \\ \bar{\mathbf{y}}_k \end{bmatrix}, \begin{bmatrix} \check{\mathbf{P}}_k & \mathbf{C}_k \\ \mathbf{C}_k^{\mathsf{T}} & \mathbf{S}_k \end{bmatrix} \right). \tag{3.13}$$

The posterior $p(\mathbf{x}_k|\mathbf{y}_k) \approx \mathcal{N}\left(\mathbf{x}_k|\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k\right)$ can be obtained in closed form using the Lemma

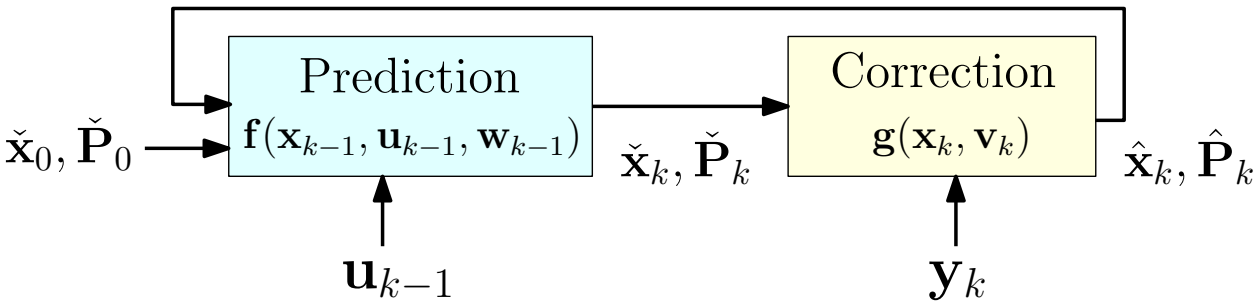


Figure 3.2: General structure of a Gaussian filter.

from [13, Lemma A.2]. Therefore, the **correction step** is,

$$\bar{\mathbf{y}}_k = \int \mathbf{g}(\mathbf{x}_k, \mathbf{v}_k) \mathcal{N}\left(\mathbf{x}_k | \check{\mathbf{x}}_k, \check{\mathbf{P}}_k\right) \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \, \mathrm{d}\mathbf{x}_k \, \mathrm{d}\mathbf{v}_k, \tag{3.14}$$

$$\begin{aligned} \mathbf{S}_k = \int & \left(\mathbf{g}(\mathbf{x}_k, \mathbf{v}_k) - \bar{\mathbf{y}}_k\right) \left(\mathbf{g}(\mathbf{x}_k, \mathbf{v}_k) - \bar{\mathbf{y}}_k\right)^{\mathsf{T}} \\ & \times \mathcal{N}\left(\mathbf{x}_k | \check{\mathbf{x}}_k, \check{\mathbf{P}}_k\right) \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \, \mathrm{d}\mathbf{x}_k \, \mathrm{d}\mathbf{v}_k, \end{aligned} \tag{3.15}$$

$$\begin{aligned} \mathbf{C}_k = \int & \left(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) - \check{\mathbf{x}}_k\right) \left(\mathbf{g}(\mathbf{x}_k, \mathbf{v}_k) - \bar{\mathbf{y}}_k\right)^{\mathsf{T}} \\ & \times \mathcal{N}\left(\mathbf{x}_k | \check{\mathbf{x}}_k, \check{\mathbf{P}}_k\right) \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \, \mathrm{d}\mathbf{x}_k \, \mathrm{d}\mathbf{v}_k, \end{aligned} \tag{3.16}$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}\left(\mathbf{y}_k - \bar{\mathbf{y}}_k\right), \qquad \mathbf{K} = \mathbf{C}_k \mathbf{S}_k^{-1}, \tag{3.17}$$

$$\hat{\mathbf{P}}_k = \check{\mathbf{P}}_k - \mathbf{K}\mathbf{S}_k\mathbf{K}^{\mathsf{T}}. \tag{3.18}$$

The structure of the filter is shown in Figure 3.2.

3.3 Extended Kalman Filter

The Extended Kalman Filter (EKF) is a type of Gaussian filter. In this filter, the approach taken to compute (3.10a) and (3.10b) is linearization [13],

$$\bar{\mathbf{y}} = \int \mathbf{g}(\mathbf{x})\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\,\mathrm{d}\mathbf{x} \tag{3.19}$$

$$\approx \int \left(\mathbf{g}(\boldsymbol{\mu}) + \left.\frac{\mathrm{d}\mathbf{g}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right|_{\boldsymbol{\mu}} \delta\mathbf{x}\right) \mathcal{N}(\delta\mathbf{x}|\mathbf{0}, \boldsymbol{\Sigma})\,\mathrm{d}\delta\mathbf{x} \tag{3.20}$$

$$= \mathbf{g}(\boldsymbol{\mu}), \tag{3.21}$$

$$\mathbf{S} = \int (\mathbf{g}(\mathbf{x}) - \bar{\mathbf{y}})\,(\mathbf{g}(\mathbf{x}) - \bar{\mathbf{y}})^{\mathsf{T}}\,\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\,\mathrm{d}\mathbf{x} \tag{3.22}$$

$$\approx \int \left(\left.\frac{\mathrm{d}\mathbf{g}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right|_{\boldsymbol{\mu}} \delta\mathbf{x}\right) \left(\left.\frac{\mathrm{d}\mathbf{g}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right|_{\boldsymbol{\mu}} \delta\mathbf{x}\right)^{\mathsf{T}} \mathcal{N}(\delta\mathbf{x}|\mathbf{0}, \boldsymbol{\Sigma})\,\mathrm{d}\delta\mathbf{x} \tag{3.23}$$

$$= \left.\frac{\mathrm{d}\mathbf{g}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right|_{\boldsymbol{\mu}} \boldsymbol{\Sigma} \left.\frac{\mathrm{d}\mathbf{g}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right|_{\boldsymbol{\mu}}^{\mathsf{T}}. \tag{3.24}$$

Given these results, the prediction and correction steps can be easily derived. The **prediction step** is given by

$$\mathbf{F}_{k-1} = \left.\frac{\partial \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})}{\partial \mathbf{x}_{k-1}}\right|_{\mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1}, \mathbf{w}_{k-1}=\mathbf{0}}, \tag{3.25}$$

$$\mathbf{L}_{k-1} = \left.\frac{\partial \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})}{\partial \mathbf{w}_{k-1}}\right|_{\mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1}, \mathbf{w}_{k-1}=\mathbf{0}}, \tag{3.26}$$

$$\check{\mathbf{x}}_k = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}), \tag{3.27}$$

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^{\mathsf{T}} + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^{\mathsf{T}}. \tag{3.28}$$

The **correction step** is given by

$$\mathbf{G}_k = \left.\frac{\partial \mathbf{g}(\mathbf{x}_k, \mathbf{v}_k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\check{\mathbf{x}}_k, \mathbf{v}_k=\mathbf{0}}, \tag{3.29}$$

$$\mathbf{M}_k = \left.\frac{\partial \mathbf{g}(\mathbf{x}_k, \mathbf{v}_k)}{\partial \mathbf{v}_k}\right|_{\mathbf{x}_k=\check{\mathbf{x}}_k, \mathbf{v}_k=\mathbf{0}}, \tag{3.30}$$

$$\mathbf{K} = \check{\mathbf{P}}_k\mathbf{G}_k(\mathbf{G}_k\check{\mathbf{P}}_k\mathbf{G}_k^{\mathsf{T}} + \mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^{\mathsf{T}})^{-1}, \tag{3.31}$$

$$\mathbf{z}_k = \mathbf{y}_k - \mathbf{g}(\check{\mathbf{x}}_k, \mathbf{0}), \tag{3.32}$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}\mathbf{z}_k, \tag{3.33}$$

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}\mathbf{G}_k)\check{\mathbf{P}}_k. \tag{3.34}$$

For a more in-depth discussion, the reader is referred to [13].

3.4 Iterated Extended Kalman Filter

The iterated EKF inherits the EKF prediction step, but its correction step is formulated as a nonlinear least squares solved with Gauss-Newton algorithm [22]. The iterated EKF evaluates multiple times the correction step of a typical extended Kalman filter, but updating the evaluation points at each time,

$$\mathbf{G}_k = \left.\frac{\partial \mathbf{g}(\mathbf{x}_k, \mathbf{v}_k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\check{\mathbf{x}}_k, \mathbf{v}_k=\mathbf{0}}, \tag{3.35a}$$

$$\mathbf{M}_k = \left.\frac{\partial \mathbf{g}(\mathbf{x}_k, \mathbf{v}_k)}{\partial \mathbf{v}_k}\right|_{\mathbf{x}_k=\check{\mathbf{x}}_k, \mathbf{v}_k=\mathbf{0}}, \tag{3.35b}$$

$$\mathbf{K} = \check{\mathbf{P}}_k \mathbf{G}_k (\mathbf{G}_k \check{\mathbf{P}}_k \mathbf{G}_k^\mathsf{T} + \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^\mathsf{T})^{-1}, \tag{3.35c}$$

$$\mathbf{z} = \mathbf{y}_k - \mathbf{g}(\hat{\mathbf{x}}_k, \mathbf{0}) + \mathbf{G}_k(\hat{\mathbf{x}}_k - \check{\mathbf{x}}_k), \tag{3.35d}$$

$$\delta\mathbf{x}_k = \mathbf{K}\mathbf{z}, \tag{3.35e}$$

and then updating the current estimate $\hat{\mathbf{x}}_k \leftarrow \check{\mathbf{x}}_k + \delta\mathbf{x}_k$. Equations (3.35) are continuously iterated, updating the estimate each time, until convergence. The covariance is updated at last when the estimate has converged, with $\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}\mathbf{G}_k)\check{\mathbf{P}}_k$. The estimate is initialized with $\hat{\mathbf{x}}_k \leftarrow \check{\mathbf{x}}_k$. It can be noted that the EKF is an iterated EKF with only one iteration.

3.4.1 Extension of Iterated EKF to Matrix Lie Groups

Thus far the Bayes' filter and its variants have been derived for states evolving in a vector space $\mathbf{x} \in \mathbb{R}^n$. When the state evolves in a matrix Lie group, some changes are needed to account for the non-Euclidean geometry of the state space [23]. The changes are applied here to the iterated EKF, which is relevant in this thesis.

As mentioned in Sec. 2.4.2, the uncertainty in a matrix Lie group element is represented locally. Using a right-invariant perturbation $\mathbf{X} = \exp(-\delta\boldsymbol{\xi}^\wedge)\bar{\mathbf{X}}$, and a process model of the form $\mathbf{F} : G \to G$, the **prediction step** is given by

$$\mathbf{A}_{k-1} = \left.\frac{D\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})}{D\mathbf{X}_{k-1}}\right|_{\mathbf{X}_{k-1}=\hat{\mathbf{X}}_{k-1}, \mathbf{w}_{k-1}=\mathbf{0}}, \tag{3.36}$$

$$\mathbf{L}_{k-1} = \left.\frac{\partial\mathbf{F}(\mathbf{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})}{\partial\mathbf{w}_{k-1}}\right|_{\mathbf{X}_{k-1}=\hat{\mathbf{X}}_{k-1}, \mathbf{w}_{k-1}=\mathbf{0}}, \tag{3.37}$$

$$\check{\mathbf{X}}_k = \mathbf{F}(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}), \tag{3.38}$$

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^\mathsf{T} + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^\mathsf{T}. \tag{3.39}$$

The **correction step** is given by

$$\delta\boldsymbol{\epsilon}_k = \log\left(\hat{\mathbf{X}}_k\check{\mathbf{X}}_k^{-1}\right)^\vee \tag{3.40}$$

$$\mathbf{J} = -\mathbf{J}^\ell(\delta\boldsymbol{\epsilon}_k) \tag{3.41}$$

$$\mathbf{G}_k = \left.\frac{D\mathbf{g}(\mathbf{X}_k,\mathbf{v}_k)}{D\mathbf{X}_k}\right|_{\mathbf{X}_k=\hat{\mathbf{X}}_k,\mathbf{v}_k=\mathbf{0}}, \tag{3.42}$$

$$\mathbf{M}_k = \left.\frac{\partial\mathbf{g}(\mathbf{X}_k,\mathbf{v}_k)}{\partial\mathbf{v}_k}\right|_{\mathbf{X}_k=\hat{\mathbf{X}}_k,\mathbf{v}_k=\mathbf{0}}, \tag{3.43}$$

$$\mathbf{K} = \mathbf{J}\check{\mathbf{P}}_k\mathbf{J}^\mathsf{T}\mathbf{G}_k(\mathbf{G}_k\mathbf{J}\check{\mathbf{P}}_k\mathbf{J}^\mathsf{T}\mathbf{G}_k^\mathsf{T} + \mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^\mathsf{T})^{-1}, \tag{3.44}$$

$$\mathbf{z}_k = \mathbf{y}_k - \mathbf{g}(\hat{\mathbf{X}}_k,\mathbf{0}) + \mathbf{G}_k\mathbf{J}\delta\boldsymbol{\epsilon}_k, \tag{3.45}$$

$$\delta\boldsymbol{\xi}_k = \mathbf{K}\mathbf{z}, \tag{3.46}$$

and the state must be updated at each iteration with

$$\hat{\mathbf{X}}_k \leftarrow \exp(-\delta\boldsymbol{\xi}_k)\check{\mathbf{X}}_k. \tag{3.47}$$

The covariance is updated at last when the estimate has converged, with

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}\mathbf{G}_k)\mathbf{J}\check{\mathbf{P}}_k\mathbf{J}^\mathsf{T}. \tag{3.48}$$

3.5 Sigma-Point Kalman Filter

The sigma-point transform generates a set of sigma-points $\boldsymbol{\xi}^i$ from a prior distribution, then passes them through a nonlinearity to then approximate the mean and covariance of the transformed distribution [13]. The way (3.10a) and (3.10b) are solved is

$$\bar{\mathbf{y}} = \int \mathbf{g}(\mathbf{x})\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})\,\mathrm{d}\mathbf{x} \tag{3.49}$$

$$= \int \mathbf{g}(\boldsymbol{\mu} + \sqrt{\boldsymbol{\Sigma}}\boldsymbol{\xi})\mathcal{N}(\boldsymbol{\xi}|\mathbf{0},\mathbf{1})\,\mathrm{d}\boldsymbol{\xi} \tag{3.50}$$

$$\approx \sum_{i=0}^{n} w^i\mathbf{g}(\boldsymbol{\mu} + \sqrt{\boldsymbol{\Sigma}}\boldsymbol{\xi}^i), \tag{3.51}$$

$$\mathbf{S} = \int (\mathbf{g}(\mathbf{x}) - \bar{\mathbf{y}})\,(\mathbf{g}(\mathbf{x}) - \bar{\mathbf{y}})^\mathsf{T}\,\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})\,\mathrm{d}\mathbf{x} \tag{3.52}$$

$$= \int \left(\mathbf{g}(\boldsymbol{\mu} + \sqrt{\boldsymbol{\Sigma}}\boldsymbol{\xi}) - \bar{\mathbf{y}}\right)\left(\mathbf{g}(\boldsymbol{\mu} + \sqrt{\boldsymbol{\Sigma}}\boldsymbol{\xi}) - \bar{\mathbf{y}}\right)^\mathsf{T}\mathcal{N}(\boldsymbol{\xi}|\mathbf{0},\mathbf{1})\,\mathrm{d}\boldsymbol{\xi} \tag{3.53}$$

$$\approx \sum_{i=0}^{n} w^i\left(\mathbf{g}(\boldsymbol{\mu} + \sqrt{\boldsymbol{\Sigma}}\boldsymbol{\xi}^i) - \bar{\mathbf{y}}\right)\left(\mathbf{g}(\boldsymbol{\mu} + \sqrt{\boldsymbol{\Sigma}}\boldsymbol{\xi}^i) - \bar{\mathbf{y}}\right)^\mathsf{T}. \tag{3.54}$$

Three popular sigma-point transformations are: unscented, spherical cubature, and Gauss-Hermite [13]. The differences between each transformation lie in which type of functions

they try to better approximate. For example, Gauss-Hermite computes the exact mean and covariance when the nonlinearity is a polynomial.

For the **prediction step**, the sigma-points are then calculated by

$$\tilde{\mathbf{P}}_{k-1} = \text{diag}(\hat{\mathbf{P}}_{k-1}, \mathbf{Q}_{k-1}) \triangleq \mathbf{L}\mathbf{L}^{\mathsf{T}}, \tag{3.55}$$

$$\begin{bmatrix} \delta\mathbf{x}^{i\mathsf{T}} & \delta\mathbf{w}^{i\mathsf{T}} \end{bmatrix}^{\mathsf{T}} = \mathbf{L}\boldsymbol{\xi}^i, \tag{3.56}$$

where $\boldsymbol{\xi}^i$ is the i^{th} unit sigma-point generated by one of the three available methods, along with a corresponding weight w^i. The sigma-points are propagated as follows,

$$\check{\mathbf{x}}_k^i = \mathbf{f}(\hat{\mathbf{x}}_{k-1} + \delta\mathbf{x}^i, \mathbf{u}_{k-1}, \delta\mathbf{w}^i), \tag{3.57}$$

$$\check{\mathbf{x}}_k = \sum_{i=1}^{n} w^i \check{\mathbf{x}}_k^i, \tag{3.58}$$

$$\check{\mathbf{P}}_k = \sum_{i=1}^{n} w^i \left(\check{\mathbf{x}}_k^i - \check{\mathbf{x}}_k\right)\left(\check{\mathbf{x}}_k^i - \check{\mathbf{x}}_k\right)^{\mathsf{T}}. \tag{3.59}$$

For the **correction step**, the sigma-points are calculated by

$$\tilde{\mathbf{P}}_k = \text{diag}(\check{\mathbf{P}}_k, \mathbf{R}_k) \triangleq \mathbf{L}\mathbf{L}^{\mathsf{T}}, \tag{3.60}$$

$$\begin{bmatrix} \delta\mathbf{x}^{i\mathsf{T}} & \delta\mathbf{v}^{i\mathsf{T}} \end{bmatrix}^{\mathsf{T}} = \mathbf{L}\boldsymbol{\xi}^i. \tag{3.61}$$

Propagating and obtaining new estimates,

$$\bar{\mathbf{y}}_k = \sum_{i=1}^{n} w^i \mathbf{y}_k^i, \quad \mathbf{y}_k^i = \mathbf{g}(\check{\mathbf{x}}_k + \delta\mathbf{x}^i, \delta\mathbf{v}_k^i), \tag{3.62}$$

$$\mathbf{S}_k = \sum_{i=1}^{n} w^i \left(\mathbf{y}_k^i - \bar{\mathbf{y}}_k\right)\left(\mathbf{y}_k^i - \bar{\mathbf{y}}_k\right)^{\mathsf{T}}, \tag{3.63}$$

$$\mathbf{C}_k = \sum_{i=1}^{n} w^i \delta\mathbf{x}^i \left(\mathbf{y}_k^i - \bar{\mathbf{y}}_k\right)^{\mathsf{T}}, \tag{3.64}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + \mathbf{K}\left(\mathbf{y}_k - \bar{\mathbf{y}}_k\right), \quad \mathbf{K} = \mathbf{C}_k\mathbf{S}_k^{-1}, \tag{3.65}$$

$$\hat{\mathbf{P}}_k = \check{\mathbf{P}}_k - \mathbf{K}\mathbf{S}_k\mathbf{K}^{\mathsf{T}}. \tag{3.66}$$

3.5.1 Extension of SPKF to Matrix Lie Groups

The SPKF can also be extended to matrix Lie groups [24]. A right-invariant perturbation $\mathbf{X} = \exp(-\delta\boldsymbol{\xi}^\wedge)\bar{\mathbf{X}}$, and a process model of the form $\mathbf{F} : G \to G$ is used as in Sec. 3.4.1. For the **prediction step**, once the sigma-points are computed, they are propagated as follows,

$$\check{\mathbf{X}}_k^i = \mathbf{F}(\exp(-\delta\mathbf{x}^i)\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}, \delta\mathbf{w}^i). \tag{3.67}$$

The mean $\check{\mathbf{X}}_k$ is computed in an iterated manner, where the first of the propagated states is set as the initial mean estimate, $\check{\mathbf{X}}_k \leftarrow \check{\mathbf{X}}_k^1$, and the error with the rest of the propagated states is computed with a weighted mean. The mean is updated until a convergence criteria is met,

$$\delta\boldsymbol{\xi}_j = \sum_{i=1}^{n} w^i \log\left(\check{\mathbf{X}}_k \check{\mathbf{X}}_k^{i\,-1}\right)^{\vee}, \quad \check{\mathbf{X}}_k \leftarrow \exp(-\delta\boldsymbol{\xi}_j)\check{\mathbf{X}}_k.$$

The covariance is computed given the converged mean estimate with

$$\delta\boldsymbol{\xi}^i = \log\left(\check{\mathbf{X}}_k \check{\mathbf{X}}_k^{i\,-1}\right)^{\vee}, \tag{3.68}$$

$$\check{\mathbf{P}}_k = \sum_{i=1}^{n} w^i \delta\boldsymbol{\xi}^i \delta\boldsymbol{\xi}^{i\mathsf{T}}. \tag{3.69}$$

For the **correction step**, once the sigma-points are computed, the updates are done as follows,

$$\bar{\mathbf{y}}_k = \sum_{i=1}^{n} w^i \mathbf{y}_k^i, \quad \mathbf{y}_k^i = \mathbf{g}\left(\exp(-\delta\mathbf{x}^{i\wedge})\check{\mathbf{X}}_k, \delta\mathbf{v}_k^i\right), \tag{3.70}$$

$$\mathbf{S}_k = \sum_{i=1}^{n} w^i \left(\mathbf{y}_k^i - \bar{\mathbf{y}}_k\right)\left(\mathbf{y}_k^i - \bar{\mathbf{y}}_k\right)^{\mathsf{T}}, \tag{3.71}$$

$$\mathbf{C}_k = \sum_{i=1}^{n} w^i \delta\mathbf{x}^i \left(\mathbf{y}_k^i - \bar{\mathbf{y}}_k\right)^{\mathsf{T}}, \tag{3.72}$$

$$\mathbf{z} = \mathbf{K}\left(\mathbf{y}_k - \bar{\mathbf{y}}_k\right), \quad \mathbf{K} = \mathbf{C}_k\mathbf{S}_k^{-1}, \tag{3.73}$$

$$\hat{\mathbf{X}}_k = \exp(-\mathbf{z}^{\wedge})\check{\mathbf{X}}_k, \tag{3.74}$$

$$\hat{\mathbf{P}}_k = \check{\mathbf{P}}_k - \mathbf{K}\mathbf{S}_k\mathbf{K}^{\mathsf{T}}. \tag{3.75}$$

3.6 Interacting Multiple Model

An IMM manages multiple models of a dynamic system in estimation tasks. The model at time step k is assumed to be among n possible models $\theta(k) = \{\theta_i\}_{i=1}^n$. The model switching is assumed a Markov chain with known transition probabilities

$$p_{ij} \triangleq P\left(\theta(k) = \theta_i | \theta(k-1) = \theta_j\right). \tag{3.76}$$

In robotics applications, the IMM is used to mix the estimates of multiple filters each with their own process model of the system, where the i^{th} process model $\mathbf{f}_i(\cdot)$ is written

$$\mathbf{x}_k^i = \mathbf{f}_i(\mathbf{x}_{k-1}^i, \mathbf{u}_{k-1}). \tag{3.77}$$

To avoid the exponential complexity of accounting for all possible n^k combination of models at time step k, the IMM runs n process models in parallel at all times, each with a weight $w_k^i \triangleq P(\theta(k) = \theta_i | \mathbf{y}_{1:k})$. When a new measurement $\mathbf{y}_k$ arrives, the IMM executes the following three steps.

1. *Interaction.* Mixing probabilities are calculated by

$$\mu_{k-1}^{ij} = P\left(\theta(k) = \theta_j | \theta(k-1) = \theta_i\right) P\left(\theta(k-1) = \theta_i | \mathbf{y}_{1:k-1}\right) \tag{3.78}$$

$$= \frac{1}{c} p_{ij} w_{k-1}^i, \tag{3.79}$$

   where c is a normalization constant. The mixing probabilities represent how likely a switch is to happen given the history of measurements or current knowledge of the trajectory. Every model i computes mixing probabilities with the rest of the models including itself.

2. *Mixing.* Since it is assumed the states are Gaussian distributed, a Gaussian mixture is carried out to update the state and covariance of each filter considering the mixing probabilities,

$$\check{\mathbf{x}}_{k-1}^i \leftarrow \sum_j^n \mu_{k-1}^{ij} \check{\mathbf{x}}_{k-1}^j, \tag{3.80}$$

$$\check{\mathbf{P}}_{k-1}^i \leftarrow \sum_j^n \mu_{k-1}^{ij} \left(\check{\mathbf{P}}_{k-1}^j + (\check{\mathbf{x}}_{k-1}^j - \check{\mathbf{x}}_{k-1}^i)(\check{\mathbf{x}}_{k-1}^j - \check{\mathbf{x}}_{k-1}^i)^\mathsf{T}\right). \tag{3.81}$$

3. *Weights update.* Once each filter has a mixed state estimate, the correction step is performed within each filter. The likelihood that each filter i generated the measurement $\mathbf{y}_k$ is computed to update the filters' weights,

$$\Lambda_k^i = p(\mathbf{y}_k | \mathbf{x}_{k-1}^i, \theta_{k-1}), \tag{3.82}$$

$$w_k^i = \frac{1}{c} \Lambda_k^i \sum_j^n w_{k-1}^j p_{ij}, \tag{3.83}$$

   where c is a normalization constant.

Figure 3.3 shows how these previous steps are taken in the case of using two Gaussian filters. A more in-depth description of the IMM can be found in [15, Sec. 11.6].

3.6.1 Extension of IMM to Matrix Lie Groups

The mixing step is straightforward for states in vector space. To mix distributions in Lie groups, three steps are carried out [25].

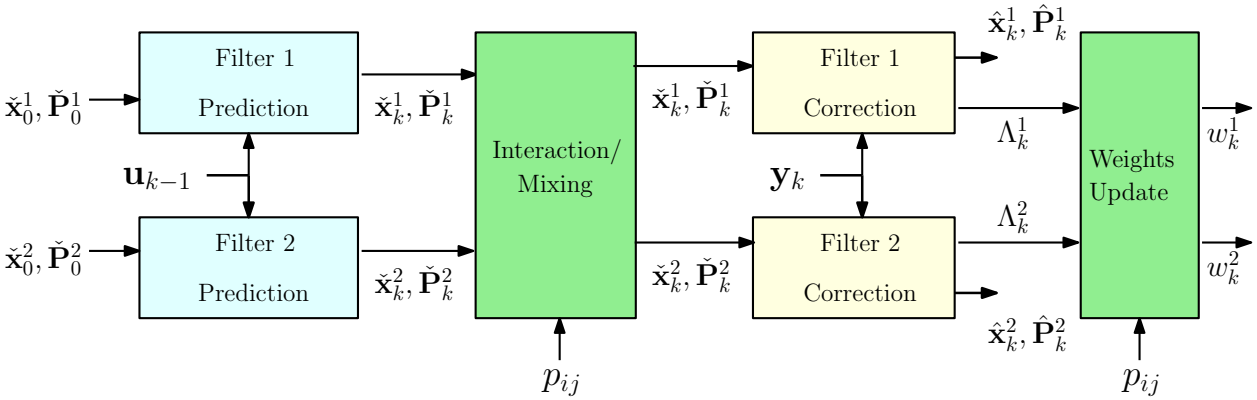


Figure 3.3: General structure of the IMM using two Gaussian filters.

1. *Distribution unfolding.* For states that evolve in a Lie group G, random variables are represented using the tools found in Sec. 2.4.2. For example,

$$\mathbf{X}_{k-1}^j = \exp\left(-\boldsymbol{\xi}_{k-1}^{j\,\wedge}\right) \check{\mathbf{X}}_{k-1}^j, \quad \boldsymbol{\xi}_{k-1}^j \sim \mathcal{N}(\mathbf{0}, \check{\mathbf{P}}_{k-1}^j). \tag{3.84}$$

   It can be noticed that the covariance $\check{\mathbf{P}}_{k-1}^j$ is defined in the Lie algebra $\mathfrak{g}$, and is relevant only to the tangent space of the mean $\check{\mathbf{X}}_{k-1}^j$. When performing the mixing for filter i, a reparametrization in vector space or "unfolding" of each estimate's $\mathbf{X}_{k-1}^j$ distribution is needed about the tangent space of $\check{\mathbf{X}}_{k-1}^i$, as shown in Fig. 3.4b. To this end,

$$\boldsymbol{\xi}_{k-1}^{ji} = \log\left(\check{\mathbf{X}}_{k-1}^i \mathbf{X}^{j\,-1}_{k-1}\right)^\vee \tag{3.85}$$

$$\approx \bar{\boldsymbol{\xi}}_{k-1}^{ji} + \mathbf{J}^r(\bar{\boldsymbol{\epsilon}}_{k-1}^{ji})^{-1} \boldsymbol{\xi}_{k-1}^j, \tag{3.86}$$

$$\bar{\boldsymbol{\xi}}_{k-1}^{ji} = \log\left(\check{\mathbf{X}}_{k-1}^i \check{\mathbf{X}}^{j\,-1}_{k-1}\right)^\vee, \tag{3.87}$$

$$\check{\mathbf{P}}_{k-1}^{ji} = \mathbf{J}^r(\bar{\boldsymbol{\xi}}_{k-1}^{ji})^{-1} \check{\mathbf{P}}_{k-1}^j \mathbf{J}^r(\bar{\boldsymbol{\xi}}_{k-1}^{ji})^{-\mathsf{T}}, \tag{3.88}$$

   where $\bar{\boldsymbol{\xi}}_{k-1}^{ji}$ and $\check{\mathbf{P}}_{k-1}^{ji}$ are the mean and covariance of (3.86). They define the reparametrized distribution in vector space of $\mathbf{X}_k^j$ about $\check{\mathbf{X}}_{k-1}^i$. Equation (3.86) is a first-order approximation of (3.85), with $\mathbf{J}^r(\bar{\boldsymbol{\xi}}_{k-1}^{ji})$ being the right group Jacobian of $SL(3)$ about $\bar{\boldsymbol{\xi}}_{k-1}^{ji}$, calculated using a backward finite-difference.

2. *Gaussian mixture.* Since the reprojected means and covariances $\bar{\boldsymbol{\xi}}_{k-1}^{ji}$ and $\check{\mathbf{P}}_{k-1}^{ji}$ describe a distribution in vector space, as can be seen in Figure 3.4c, (3.81) and (3.80) can be

applied to produce $\bar{\boldsymbol{\xi}}_{k-1}^{i}$ and $\check{\mathbf{P}}_{k-1}^{i}$,

$$\bar{\boldsymbol{\xi}}_{k-1}^{i} \leftarrow \sum_{j}^{n} \mu_{k-1}^{ij} \bar{\boldsymbol{\xi}}_{k-1}^{ji}, \tag{3.89}$$

$$\check{\mathbf{P}}_{k-1}^{i} \leftarrow \sum_{j}^{n} \mu_{k-1}^{ij} \left(\check{\mathbf{P}}_{k-1}^{ji} + (\bar{\boldsymbol{\xi}}_{k-1}^{ji} - \bar{\boldsymbol{\xi}}_{k-1}^{i})(\bar{\boldsymbol{\xi}}_{k-1}^{ji} - \bar{\boldsymbol{\xi}}_{k-1}^{i})^{\mathsf{T}} \right). \tag{3.90}$$

3. *Distribution folding.*To finalize the Gaussian mixture on $SL(3)$, these values are then projected onto the $SL(3)$ manifold as follows,

$$\check{\mathbf{X}}_{k-1}^{i} \leftarrow \exp\left(-\bar{\boldsymbol{\xi}}_{k-1}^{i}\right) \check{\mathbf{X}}_{k-1}^{i}, \tag{3.91}$$

$$\check{\mathbf{P}}_{k-1}^{i} \leftarrow \mathbf{J}^{r}(\bar{\boldsymbol{\xi}}_{k-1}^{i}) \check{\mathbf{P}}_{k-1}^{i} \mathbf{J}^{r}(\bar{\boldsymbol{\xi}}_{k-1}^{i})^{\mathsf{T}}. \tag{3.92}$$

3.7 Linearization then Discretization

In this chapter, a process model in discrete time (3.8) is assumed to derive the Bayes' filter and all its variants. However, in many cases, the process model is given in continuous time,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}), \qquad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\mathcal{Q}}\delta(t-\tau)), \tag{3.93}$$

where $\mathbf{w}$ is a white noise random process and $\boldsymbol{\mathcal{Q}}$ is the power spectral density of such signal. When carrying out a linearization-based prediction step as found in the EKF, to obtain a discrete-time model, the continuous-time model can be either discretized first and then linearized, or linearized first about the most recent state estimate $\bar{\mathbf{x}}_{k-1}$, the input $\bar{\mathbf{u}}_{k-1}$ and noise $\bar{\mathbf{w}}_{k-1}$ and then discretized. First, the Jacobians of (3.93) are needed to linearize the process model,

$$\mathbf{F} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{w}}}, \qquad \mathbf{B} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{w}}}, \qquad \mathbf{L} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{w}}}, \tag{3.94}$$

which yield a linear continuous-time model,

$$\delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x} + \mathbf{B}\delta\mathbf{u} + \mathbf{L}\delta\mathbf{w}, \qquad \delta\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\mathcal{Q}}\delta(t-\tau)), \tag{3.95}$$

which is then discretized. Van Loan's method [17], [26, Sec.4.7] allows carrying out the discretization of (3.95) by means of the matrix exponential. First,

$$\boldsymbol{\Psi} = \begin{bmatrix} \mathbf{F} & \mathbf{L}\mathbf{F}\mathbf{L}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{F}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.96}$$

is constructed. Then, the matrix exponential is computed using the Δt time step over which the discretization is carried out,

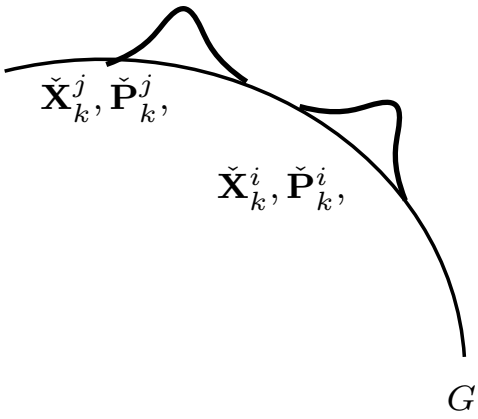
$$\mathbf{M} = \exp(\Delta t \boldsymbol{\Psi}) = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \star & \star \\ \mathbf{0} & \star & \star & \star \\ \mathbf{0} & \mathbf{0} & \star & \mathbf{M}_{34} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \star \end{bmatrix}, \tag{3.97}$$

where $\star$ denotes irrelevant nonzero entries. The linear discretized model is then

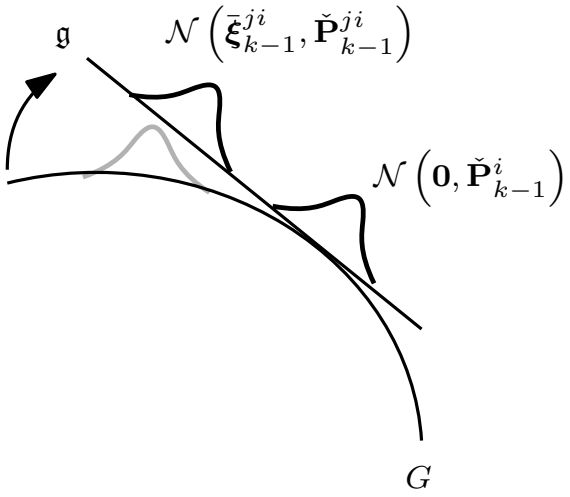
$$\mathbf{F}_d = \mathbf{M}_{11}, \qquad \mathbf{B}_d = \mathbf{M}_{34}, \qquad \mathbf{Q}_d = \mathbf{M}_{12}\mathbf{M}_{11}^{\mathsf{T}}, \tag{3.98}$$

$$\delta\mathbf{x}_{k+1} = \mathbf{F}_d \delta\mathbf{x}_k + \mathbf{B}_d \delta\mathbf{u}_k + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_d). \tag{3.99}$$

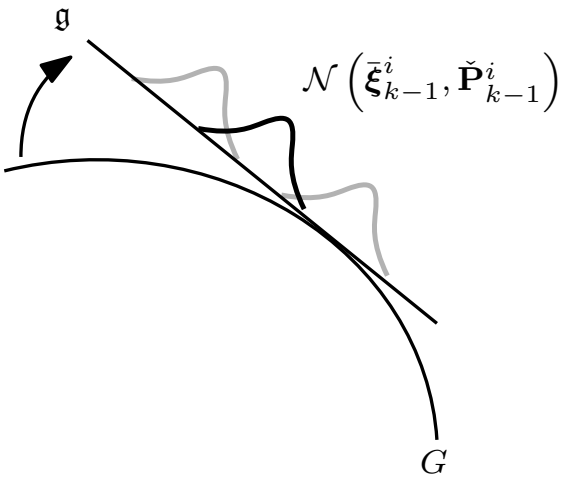
The Jacobians needed in the EKF's prediction step, are then obtained from this linear discretized model.



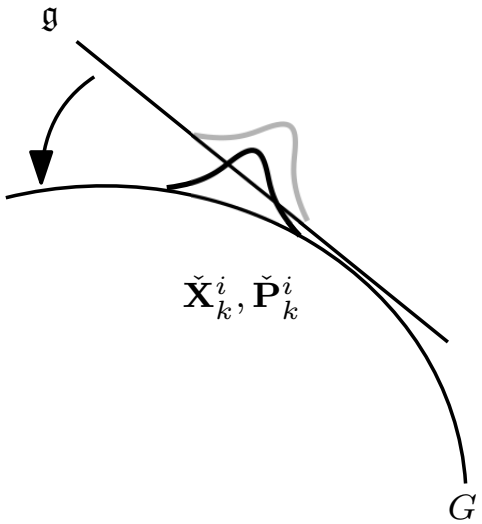
(a) Two distributions evolving in a Lie group G with their means $\check{\mathbf{X}}_k^i$ and $\check{\mathbf{X}}_k^j$ and their respective covariances. The goal is to mix both distributions.



(b) The distributions are first unfolded about the tangent plane of $\check{\mathbf{X}}_k^i$. The distributions evolve now in the Lie algebra $\mathfrak{g}$.



(c) The Gaussian distributions are mixed in vector space.



(d) The distribution is folded back into the manifold with the exponential map.

Figure 3.4: Gaussian mixture procedure for matrix Lie groups. Figure inspired by [25].

Chapter 4

Homography Estimation Using the IMM Filter

4.1 Overview

The states to be estimated are $\mathbf{H}_{ab}$ and $\mathbf{\Gamma}_{ab}$. An IMM is implemented employing two (iterated) EKFs. Both EKFs share the same measurement and process model, but have different process model noise levels, similar to [16]. Although the true noise levels are unknown in this work, the same principle will be used to manage how confident the filter is in the assumption that $\mathbf{s}_a$, defined in Sec. 2.5, is constant. To implement an EKF, the Jacobians of the process and measurement models must be obtained. A SPKF is also implemented using the same process and measurement models.

4.2 Process Model

The process model has access to a rate-gyro measurement $\mathbf{u} = \boldsymbol{\omega}_b^{ba} + \mathbf{w}$ resolved in the body frame $\mathcal{F}_b$. It is assumed that the rate-gyro is unbiased and is only corrupted by white Gaussian noise $\mathbf{w}(t) \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\mathcal{Q}}_g \delta(t-\tau)\right)$. From (2.63) and (2.71), the process model is

$$\dot{\mathbf{H}}_{ab} = \mathbf{H}_{ab}\left((\mathbf{u}^\times - \mathbf{w}^\times) + \mathbf{\Gamma}_{ab}\right), \tag{4.1}$$

$$\dot{\mathbf{\Gamma}}_{ab} = \left[\mathbf{\Gamma}_{ab}, \mathbf{u}^\times - \mathbf{w}^\times\right] + \mathbf{w}^{m\wedge}. \tag{4.2}$$

As previously discussed in Sec. 2.5, assuming $\mathbf{s}_a$ is constant in time comes at the cost of process model inaccuracies when the assumption is broken. One way to account for modeling errors is adding a noise term $\mathbf{w}^m(t) \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\mathcal{Q}}_m \delta(t-\tau)\right) \in \mathbb{R}^8$. The power spectral density (PSD) is modeled as $\boldsymbol{\mathcal{Q}}_m = \sigma_m^2 \mathbf{1}$. The size of σ_m^2 represents how confident the filter is on the motion model for $\mathbf{\Gamma}_{ab}$.

The IMM introduced in Sec. 3.6 is a common adaptive method that weights the available motion hypotheses on a Bayesian framework to produce an estimate. The idea is to have two similar models, with different values of σ_m^2. Low values accommodate for scenarios where the assumption that $\mathbf{s}_a$ is constant is respected, and higher values can deal with scenarios where the motion violates this assumption. The IMM should provide an estimate with better consistency properties than using a single filter in cases where the trajectory is more varied, switching from slow to aggressive maneuvers.

4.2.1 Jacobians of the Process Model

Recalling Sec. 2.4.2, consider the right-invariant error definitions,

$$\delta\mathbf{H} \triangleq \exp(\delta\boldsymbol{\xi}^\wedge) = \bar{\mathbf{H}}\mathbf{H}^{-1}, \tag{4.3}$$

$$\delta\boldsymbol{\gamma}^\wedge = \boldsymbol{\Gamma} - \bar{\boldsymbol{\Gamma}}, \tag{4.4}$$

Using (4.3) and (4.4) in (4.1) and (4.2) results in a perturbed process model,

$$-\delta\mathbf{H}_{ab}^{-1}\delta\dot{\mathbf{H}}_{ab}\delta\mathbf{H}_{ab}^{-1}\bar{\mathbf{H}}_{ab} + \delta\mathbf{H}_{ab}^{-1}\dot{\bar{\mathbf{H}}}_{ab} = \delta\mathbf{H}_{ab}^{-1}\bar{\mathbf{H}}_{ab}\left(\mathbf{u}^\times + \bar{\boldsymbol{\Gamma}}_{ab} + \delta\boldsymbol{\Gamma}_{ab} - \delta\mathbf{w}^\times\right), \tag{4.5}$$

$$\dot{\bar{\boldsymbol{\Gamma}}}_{ab} + \delta\dot{\boldsymbol{\Gamma}}_{ab} = \left[\bar{\boldsymbol{\Gamma}}_{ab} + \delta\boldsymbol{\Gamma}_{ab}, \mathbf{u}^\times - \delta\mathbf{w}^\times\right] + \delta\mathbf{w}^{m\wedge}. \tag{4.6}$$

After algebraic manipulations,

$$-\delta\dot{\mathbf{H}}_{ab}\delta\mathbf{H}_{ab}^{-1} = \bar{\mathbf{H}}_{ab}\left(\delta\boldsymbol{\Gamma}_{ab} - \delta\mathbf{w}^\times\right)\bar{\mathbf{H}}_{ab}^{-1}, \tag{4.7}$$

$$\delta\dot{\boldsymbol{\Gamma}}_{ab} = \left[\delta\boldsymbol{\Gamma}_{ab}, \mathbf{u}^\times\right] - \left[\bar{\boldsymbol{\Gamma}}_{ab}, \delta\mathbf{w}^\times\right] + \delta\mathbf{w}^{m\wedge}, \tag{4.8}$$

where $\delta\boldsymbol{\Gamma}_{ab}\delta\mathbf{w}^\times \approx \mathbf{0}$ has been used. Using $\mathbf{1} + \delta\boldsymbol{\xi}^{\mathrm{R}\wedge} \approx \delta\mathbf{H}^{\mathrm{R}}$ and $\boldsymbol{\gamma}^\wedge = \boldsymbol{\Gamma}$,

$$-\delta\dot{\boldsymbol{\xi}}_{ab}^\wedge\left(\mathbf{1} - \delta\boldsymbol{\xi}_{ab}^\wedge\right) = \bar{\mathbf{H}}_{ab}\left(\delta\boldsymbol{\Gamma}_{ab} - \delta\mathbf{w}^\times\right)\bar{\mathbf{H}}_{ab}^{-1}, \tag{4.9}$$

$$\delta\dot{\boldsymbol{\gamma}}_{ab}^\wedge = \left[\delta\boldsymbol{\gamma}_{ab}^\wedge, \mathbf{u}^\times\right] - \left[\bar{\boldsymbol{\gamma}}_{ab}^\wedge, \delta\mathbf{w}^\times\right] + \delta\mathbf{w}^{m\wedge}. \tag{4.10}$$

Assuming $\delta\dot{\boldsymbol{\xi}}_{ab}^\wedge\delta\boldsymbol{\xi}_{ab}^\wedge \approx \mathbf{0}$, the linearized process model evolving in the Lie algebra is

$$\delta\dot{\boldsymbol{\xi}}_{ab}^\wedge = -\mathrm{Ad}\left(\bar{\mathbf{H}}_{ab}\right)\delta\boldsymbol{\gamma}_{ab}^\wedge + \mathrm{Ad}\left(\bar{\mathbf{H}}_{ab}\right)\delta\mathbf{w}^\times, \tag{4.11a}$$

$$\delta\dot{\boldsymbol{\gamma}}_{ab}^\wedge = -\mathrm{ad}\left(\mathbf{u}\right)\delta\boldsymbol{\gamma}_{ab}^\wedge - \mathrm{ad}\left(\bar{\boldsymbol{\gamma}}_{ab}\right)\delta\mathbf{w}^\times + \delta\mathbf{w}_m^\wedge. \tag{4.11b}$$

When applying the linear $(\cdot)^\vee$ operator on (4.11), $\delta\mathbf{w}^\times$ has to be addressed. A projection matrix $\mathbf{B}$ can be found such that, $\delta\mathbf{w}^{\times\vee} = \mathbf{B}\delta\mathbf{w}$. As such, the linearized process model is thus

$$\delta\dot{\boldsymbol{\xi}}_{ab} = -\mathbf{Ad}\left(\bar{\mathbf{H}}_{ab}\right)\delta\boldsymbol{\gamma}_{ab} + \mathbf{Ad}\left(\bar{\mathbf{H}}_{ab}\right)\mathbf{B}\delta\mathbf{w}, \tag{4.12a}$$

$$\delta\dot{\boldsymbol{\gamma}}_{ab} = -\mathbf{ad}\left(\mathbf{u}\right)\delta\boldsymbol{\gamma}_{ab} - \mathbf{ad}\left(\bar{\boldsymbol{\gamma}}_{ab}\right)\mathbf{B}\delta\mathbf{w} + \delta\mathbf{w}^m. \tag{4.12b}$$

Setting $\delta\mathbf{x}_{ab} = \begin{bmatrix}\delta\boldsymbol{\xi}_{ab}^{\mathsf{T}} & \delta\boldsymbol{\gamma}_{ab}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$, and $\delta\mathbf{w}_{ab} = \begin{bmatrix}\delta\mathbf{w}^{\mathsf{T}} & \delta\mathbf{w}^{m\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$, the linearized process model can be written as

$$\delta\dot{\mathbf{x}}_{ab} = \underbrace{\begin{bmatrix}\mathbf{0} & -\mathbf{Ad}\left(\bar{\mathbf{H}}_{ab}\right)\\ \mathbf{0} & -\mathbf{ad}\left(\mathbf{u}\right)\end{bmatrix}}_{\mathbf{A}}\delta\mathbf{x}_{ab} + \underbrace{\begin{bmatrix}\mathbf{Ad}\left(\bar{\mathbf{H}}_{ab}\right)\mathbf{B} & \mathbf{0}\\ -\mathbf{ad}\left(\bar{\boldsymbol{\gamma}}_{ab}\right)\mathbf{B} & \mathbf{1}\end{bmatrix}}_{\mathbf{L}}\delta\mathbf{w}_{ab}. \tag{4.13}$$

At this point, the linearized process model given by (4.13) can be discretized using the matrix exponential [17, 26], as discussed in Sec. 3.7. The resulting discrete-time Jacobians are then used in a standard iterated EKF framework.

4.3 Measurement Model

From (2.13), given a point correspondence obtained from camera measurements of a point feature p_i lying on a plane, the pair can be related by a homography transformation $\mathbf{H}_k \triangleq \mathbf{H}_{ab}$ as follows,

$$\mathbf{y}_b^i = \mathbf{g}(\mathbf{H}_k^{-1}\mathbf{p}_a^i) + \mathbf{v}_k, \tag{4.14}$$

where $\mathbf{p}_a^i$ represents the measurement of p_i resolved in $\mathcal{F}_a$, in normalized image coordinates, $\mathbf{y}_b^i$ the noisy measurement in pixel coordinates of the same feature resolved in $\mathcal{F}_b$, and $\mathbf{v}_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_k\right) \in \mathbb{R}^2$ models white noise on the pixel measurement. Setting $\mathbf{r}_b^i \triangleq \mathbf{H}_k^{-1}\mathbf{p}_a^i = \begin{bmatrix}x & y & z\end{bmatrix}^{\mathsf{T}}$,

$$\mathbf{g}\left(\mathbf{r}_b^i\right) \triangleq \frac{1}{z}\mathbf{D}\mathbf{K}\mathbf{r}_b^i, \tag{4.15}$$

where $\mathbf{D} = \begin{bmatrix}\mathbf{1}_{2\times 2} & \mathbf{0}_{2\times 1}\end{bmatrix}$ and $\mathbf{K}$ is the previously introduced intrinsic parameter matrix to model a pinhole camera.

4.3.1 Jacobians of the Measurement Model

The linearized measurement model is,

$$\delta\mathbf{y}_b^i \approx \frac{\partial\mathbf{g}(\mathbf{r}_b^i)}{\partial\mathbf{r}_b^i}\left[\left.\frac{D\mathbf{r}_b^i(\mathbf{H}_k)}{D\mathbf{H}_k}\right|_{\bar{\mathbf{H}}_k} \quad \left.\frac{\partial\mathbf{r}_b^i(\boldsymbol{\Gamma}_k)}{\partial\boldsymbol{\Gamma}_k}\right|_{\bar{\boldsymbol{\Gamma}}_k}\right]\delta\mathbf{x}_k + \delta\mathbf{v}_k, \tag{4.16}$$

where

$$\frac{\partial\mathbf{g}(\mathbf{r}_b^i)}{\partial\mathbf{r}_b^i} = \frac{1}{z}\begin{bmatrix}f_u & 0 & -f_u\frac{x}{z}\\ 0 & f_v & -f_v\frac{y}{z}\end{bmatrix}. \tag{4.17}$$

To find the Jacobians of $\mathbf{r}_b^i$, a perturbation is again applied to linearize $\mathbf{r}_b^i$,

$$\bar{\mathbf{r}}_b^i + \delta\mathbf{r}_b^i = \bar{\mathbf{H}}_k^{-1}\delta\mathbf{H}_k\mathbf{p}_a^i. \tag{4.18}$$

Using $\mathbf{1} + \delta\boldsymbol{\xi}^{\mathrm{R}\wedge} \approx \delta\mathbf{H}^{\mathrm{R}}$,

$$\bar{\mathbf{r}}_b^i + \delta\mathbf{r}_b^i \approx \bar{\mathbf{H}}_k^{-1}\mathbf{p}_a^i + \bar{\mathbf{H}}_k^{-1}\delta\boldsymbol{\xi}_k^{\wedge}\mathbf{p}_a^i, \tag{4.19}$$

$$\delta\mathbf{r}_b^i = \bar{\mathbf{H}}_k^{-1}\mathbf{p}_a^{i\,\odot}\delta\boldsymbol{\xi}_k. \tag{4.20}$$

Thus, this yields

$$\left[\left.\frac{D\mathbf{r}_b^i(\mathbf{H}_k)}{D\mathbf{H}_k}\right|_{\bar{\mathbf{H}}_k} \quad \left.\frac{\partial\mathbf{r}_b^i(\boldsymbol{\Gamma}_k)}{\partial\boldsymbol{\Gamma}_k}\right|_{\bar{\boldsymbol{\Gamma}}_k}\right] = \left[\bar{\mathbf{H}}_k^{-1}\mathbf{p}_a^{i\,\odot} \quad \mathbf{0}_{3\times 8}\right]. \tag{4.21}$$

With these derived jacobians, the prediction and correction steps can be determined for an iterated EKF.

4.4 Robust Loss

Point matching in real applications is subject to outliers even after outlier-removal procedures like RANSAC. Robust M-estimation is a popular method that downweighs outliers in an optimization procedure to find the estimate. In a nonlinear least squares formulation,

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{e}(\mathbf{x})^{\mathsf{T}}\mathbf{e}(\mathbf{x}) = \frac{1}{2}r(\mathbf{x})^2, \tag{4.22}$$

a robust loss function is added to decrease the influence of outliers,

$$J'(\mathbf{x}) = \sum_i \rho\left(r_i(\mathbf{x})\right). \tag{4.23}$$

It can be shown that this redefined optimization problem is equivalent to an Iteratively Reweighted Least-Squares (IRLS) problem [27],

$$J'(\mathbf{x}) = \frac{1}{2}\sum_i w(r_i(\bar{\mathbf{x}}))r_i(\mathbf{x})^2, \tag{4.24}$$

where $\bar{\mathbf{x}}$ is the current best estimate and $w(\cdot)$ is a weight function associated to the robust loss function. An iterated EKF's correction step can be formulated as a weighted nonlinear least squares problem [22] solved by Gauss–Newton,

$$\mathbf{e}(\mathbf{x}_k)^{\mathsf{T}} = \left[\left(\check{\mathbf{x}}_k - \mathbf{x}_k\right)^{\mathsf{T}}\check{\mathbf{P}}_k^{-\frac{1}{2}} \quad \left(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k)\right)^{\mathsf{T}}\mathbf{R}_k^{-\frac{1}{2}}\right], \tag{4.25}$$

$$J(\mathbf{x}_k) = \frac{1}{2}\mathbf{e}(\mathbf{x}_k)^{\mathsf{T}}\mathbf{e}(\mathbf{x}_k) \tag{4.26}$$

$$= \frac{1}{2}\left(\check{\mathbf{x}}_k - \mathbf{x}_k\right)^{\mathsf{T}}\check{\mathbf{P}}_k^{-1}\left(\check{\mathbf{x}}_k - \mathbf{x}_k\right) + \frac{1}{2}\left(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k)\right)^{\mathsf{T}}\mathbf{R}_k^{-1}\left(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k)\right) \tag{4.27}$$

$$= \frac{1}{2}r_{\text{prior}}(\mathbf{x})^2 + \frac{1}{2}r_{\text{meas}}(\mathbf{x})^2, \tag{4.28}$$

Figure 4.1: Front view of an Intel Realsense D453i, shown by the red circle, used to obtain both camera and angular velocity measurements. It is mounted on an T-slot aluminum profile. The numerous small gray spheres are the markers that the OptiTrack system uses to track the sensor head and generate ground truth data.

If the robust loss is applied solely on the measurement errors, it is straightforward to modify the weighting matrix to account for that change,

$$J'(\mathbf{x}_k) = \frac{1}{2} r_{\text{prior}}(\mathbf{x})^2 + \rho\left(r_{\text{meas}}(\mathbf{x})\right) \tag{4.29}$$

$$= \frac{1}{2}\left(\check{\mathbf{x}}_k - \mathbf{x}_k\right)^{\mathsf{T}} \check{\mathbf{P}}_k^{-1}\left(\check{\mathbf{x}}_k - \mathbf{x}_k\right) + \frac{1}{2} w(r_{\text{meas}}(\bar{\mathbf{x}}))\left(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k)\right)^{\mathsf{T}} \mathbf{R}_k^{-1}\left(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k)\right) \tag{4.30}$$

$$= \frac{1}{2}\mathbf{e}(\mathbf{x}_k)^{\mathsf{T}} \begin{bmatrix} \mathbf{1} & \\ & w(r_{\text{meas}}(\bar{\mathbf{x}}))\mathbf{1} \end{bmatrix} \mathbf{e}(\mathbf{x}). \tag{4.31}$$

The *SC/DCS* robust loss was picked due to the properties explained in [27]. Its weight function is given by

$$w(r) = \begin{cases} \frac{4c^2}{(c+r^2)^2} & \text{if } r^2 \geq c \\ 1 & \text{otherwise.} \end{cases} \tag{4.32}$$

4.5 Experimental Setup

To collect experimental data, an Intel Realsense D435i is used, shown in Figure 4.1. It has an IMU included, which provides angular velocity measurements at 200 Hz. The left camera

Figure 4.2: The experimental setup. The Intel Realsense D435i is moved in this space, observing the pages spread in the floor, employed to generate distinctive features on a plane. The OptiTrack system tracks the sensor head maneuvering in this space. A close-up of the tracked plane is shown in the right image.

of the stereo rig is used for camera measurements of 640×480 pixels at 30 Hz. Ground truth data is collected using an OptiTrack optical motion capture system at 120 Hz. The whole setup is shown in Figure 4.2.

4.5.1 Sensor Calibration

From (4.14), it can be noticed that the intrinsic parameter matrix $\mathbf{K}$ from the camera is needed. Many tools are available to automate obtaining this matrix from camera measurements. Observations of a checkerboard with known dimensions is a common way to determine the camera intrinsics. In this thesis, the `Kalibr` toolbox was used to determine the camera intrinsics, which has the calibration solution from [28] implemented. The checkerboard shown in Figure 4.3 was used.

Additionally, from (2.63), it can be observed that the angular velocity measurement needs to be resolved in the camera's frame, but this measurement is offered by the IMU and therefore resolved in the IMU's frame. To transform the measurement to the camera's frame, the pose $\mathbf{T}_{bc}$ from the IMU's frame $\mathcal{F}_b$ to the camera's frame $\mathcal{F}_c$ must be known. It can be shown using (2.53a) that

$$\dot{\mathbf{C}}_{ac} = \mathbf{C}_{ac} \underbrace{\mathbf{C}_{bc}^{\mathsf{T}} \boldsymbol{\omega}_b^{ba\times} \mathbf{C}_{bc}}_{\boldsymbol{\omega}_c^{ca\times}}. \tag{4.33}$$

Again, `Kalibr` has the tools to perform a Camera-IMU calibration, by implementing the solution presented in [29]. It requires again camera measurements of a board with known

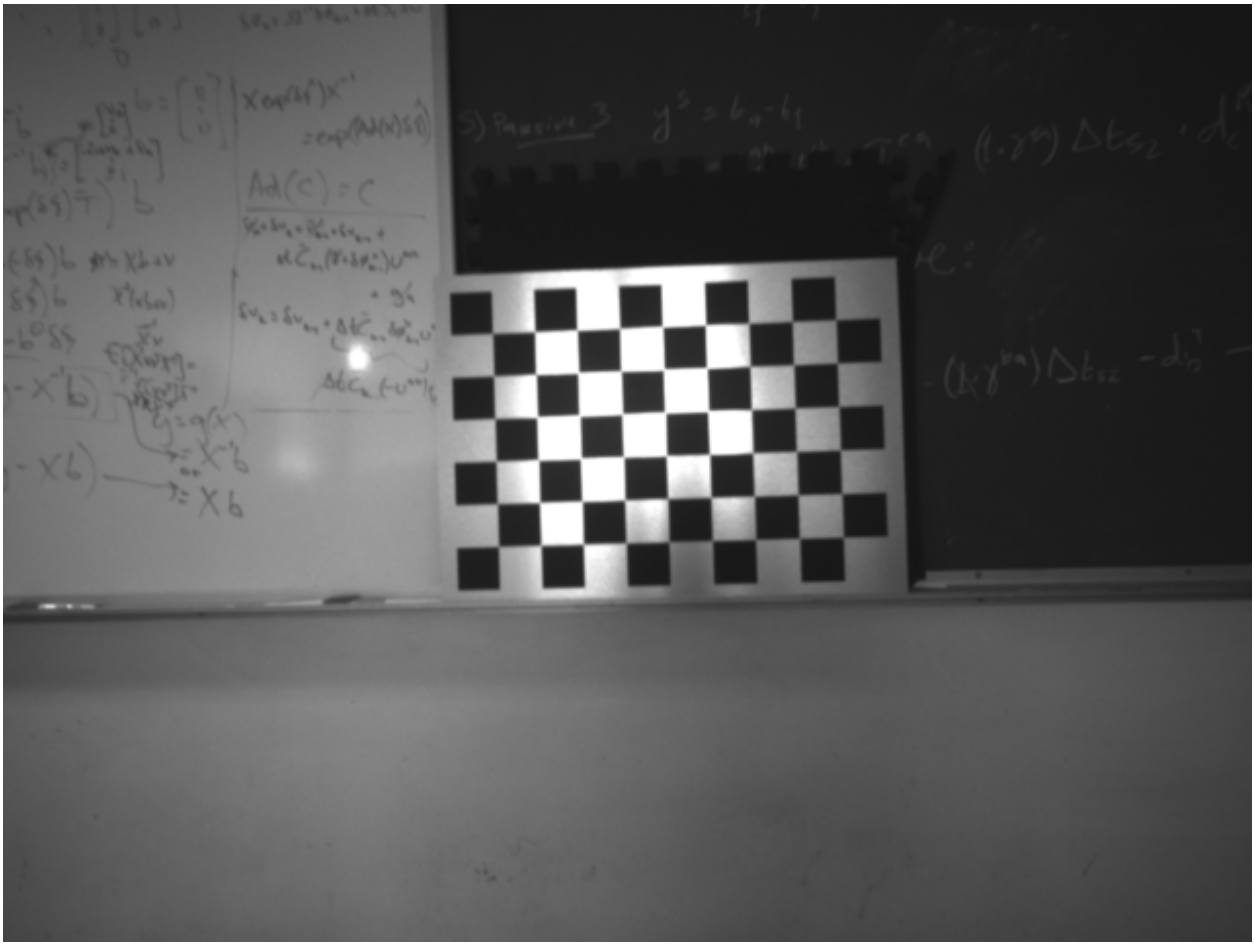


Figure 4.3: Checkerboard employed to calibrate the Intel Realsense D435i camera intrinsics parameters and the pose between the camera and the IMU.

dimensions and features, as well as accelerometer and angular velocity measurements, provided by the IMU.

4.5.2 Point Correspondences

Homography estimation as proposed in this thesis requires point correspondences from different camera measurements. To accomplish this task, ORB descriptors are used as implemented in `OpenCV` [30]. Descriptor-based feature matching consists of two steps [31].

1. *Detection.* Keypoints are found in this step, which are usually corners. Corners are regions in the image with large variation in intensity in all the directions. Many different methods exist to find corners, but in a descriptor-based approach, a *descriptor* is assigned to each detected corner. A descriptor essentially encodes the region surrounding the extracted keypoint.

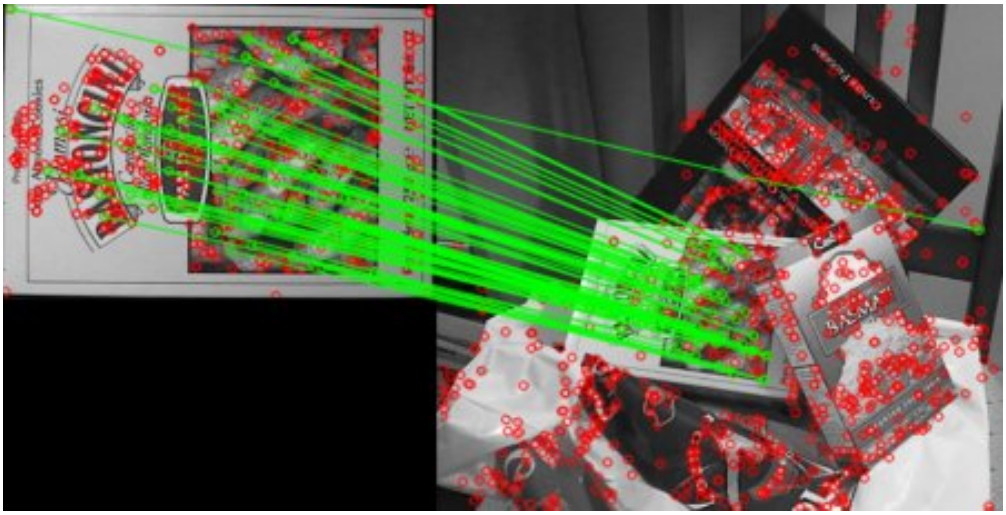


Figure 4.4: Examples of how detected keypoints, denoted with red, in two images with overlapping features get matched with a descriptor approach. Obtained from [33].

2. *Matching.* Once keypoints with their descriptors are computed for two images, the matching procedure consists in comparing the descriptors with a brute force approach to distinguish which keypoints are present in both images. Those descriptors that are similar by a norm sense, are considered to belong to the same feature.

After the matches are obtained, the distance ratio test from from [32, Sec. 7.1], is applied to remove spurious matches. In this thesis, the homography is estimated w.r.t. a reference frame, so that means all the features are matched against those from a reference image. No outlier rejection methods are used besides robust M-estimation, used in all filters. No other procedures are done for point correspondences.

Chapter 5

Results and Discussion

5.1 Overview

In this chapter, the proposed IMM filter is tested in simulation and on experimental data. The performance is compared relative to the observer of [2] and two versions of the iterated EKF. An implementation of the sigma point Kalman filter (SPKF) is also tested in simulation but not in experimental data since it has undesirable properties, later discussed. More details about the observer can be found in Appx. A.1 of this thesis.

5.2 Simulation Results

The proposed IMM filter is first tested in simulation. A set of trajectories are generated, where a camera tracks 4 points on a plane at all times, the minimum number of point matches to define a homography [6]. The rate-gyro and camera provide data at 90 Hz and 30 Hz, respectively. Gyroscope and camera measurements are corrupted by additive Gaussian noise, simulated using $\boldsymbol{\mathcal{Q}}_g = \sigma_g^2\mathbf{1}$, $\mathbf{R}_k = \sigma_r^2\mathbf{1}$, with $\sigma_g = 0.01\,\mathrm{rad\,s^{-1}}$, $\sigma_r = 1\,\mathrm{pixel}$. Initial uncertainty is set to $\hat{\mathbf{P}}_0 = 1\mathrm{e}{-1}\mathbf{1}$. Although an additive Gaussian noise model for pixel coordinates has shortcomings in real world conditions, such as accounting for outliers in the point correspondence procedure, it is a typical assumption in the robotics community [34]. Outliers in the point correspondence problem are handled in the experimental case with a robust loss function, described in Sec. 4.4.

100 Monte Carlo runs with varying initial conditions and noise realizations are performed for each trajectory to evaluate the filters' consistency and accuracy when the assumption that $\mathbf{s}_a$ is constant is broken in different ways. Among the tested filters are 2 versions of the iterated EKF. The first is *EKF tight*, which is confident in the assumption that $\mathbf{s}_a$ is constant, by setting $\sigma_m^2 = 1\mathrm{e}{-7}$. The second is *EKF loose*, with $\sigma_m^2 = 1\mathrm{e}{-1}$, which has little confidence in

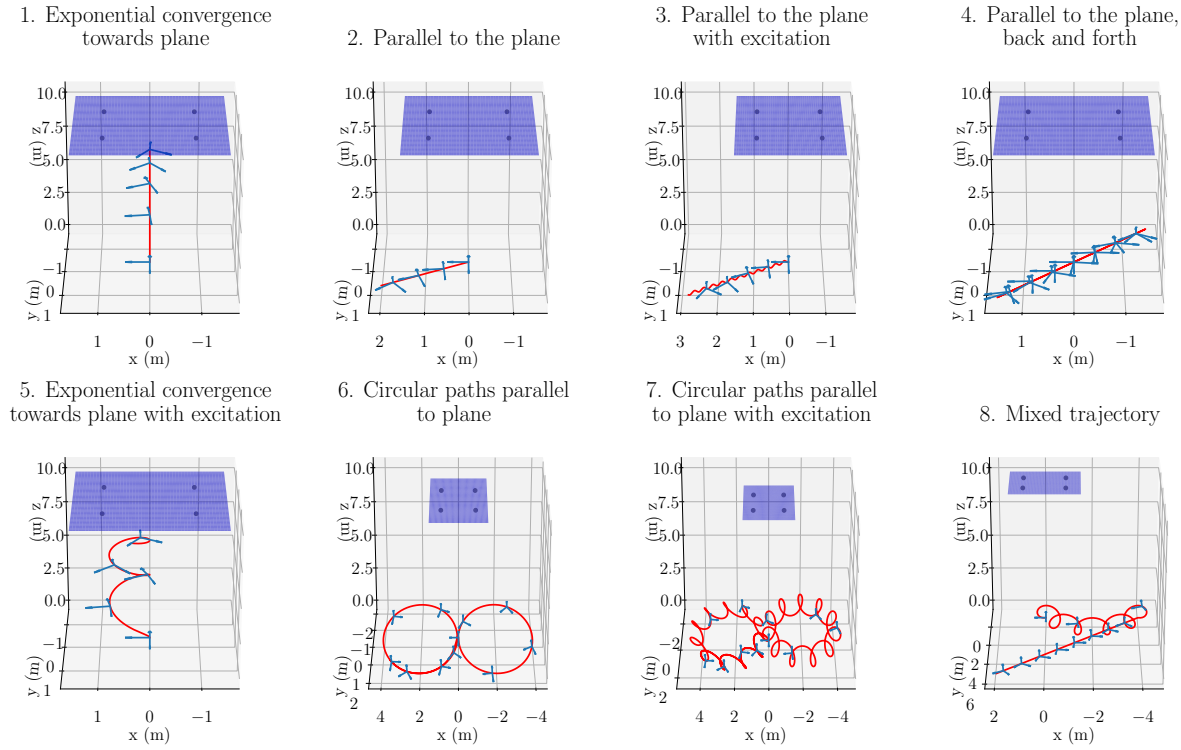


Figure 5.1: Simulated trajectories.

the assumption that $\mathbf{s}_a$ is constant. A SPKF is also included in the tests, with $\sigma_m^2 = 5\text{e}{-4}$. An IMM composed of both versions of the EKFs is also tested, as well as the observer from [2].

To evaluate the accuracy of the filters on each trajectory, the error

$$r_k = \| \log \left(\hat{\mathbf{H}}_k \mathbf{H}_k^{-1} \right)^{\vee} \|_2 \tag{5.1}$$

is used to compare homography estimates to the true homography value at each time step. In Figure 5.2, it is shown how r_k is distributed across all time steps in all trajectories from Figure 5.1. The average of r_k across all trials and then averaged across all time steps is displayed in Table 5.1. *EKF tight* performs the best on this metric when the trajectories respect the assumption that $\mathbf{s}_a$ is constant as in trajectories 1 and 2, or closely do, as in trajectory 3. In the remaining trajectories, the performance of *EKF tight* degrades. In general, the IMM offers the best performance when the assumption is less respected. Only in trajectory 7, when the assumption is severely broken, *EKF loose* outperforms the IMM by a small margin. The SPKF does not offer the best performance in any of the trajectories but it is still better than the observer of [2] and than the *EKF tight* when the assumption

that $\mathbf{s}_a$ is constant is broken, except in the last one, which might be due to the fact that the SPKF is not tuned for this trajectory. The observer of [2] has higher homography estimation error in all trials, since all the filters are characterizing the sensor noise properly and tuned accordingly. When tuning the observer it was observed that that modifying the gains did not change the observer's performance drastically. The observer is, in effect, constant gain, while the IMM filter changes the gain at each time step. Assessing consistency is done using the

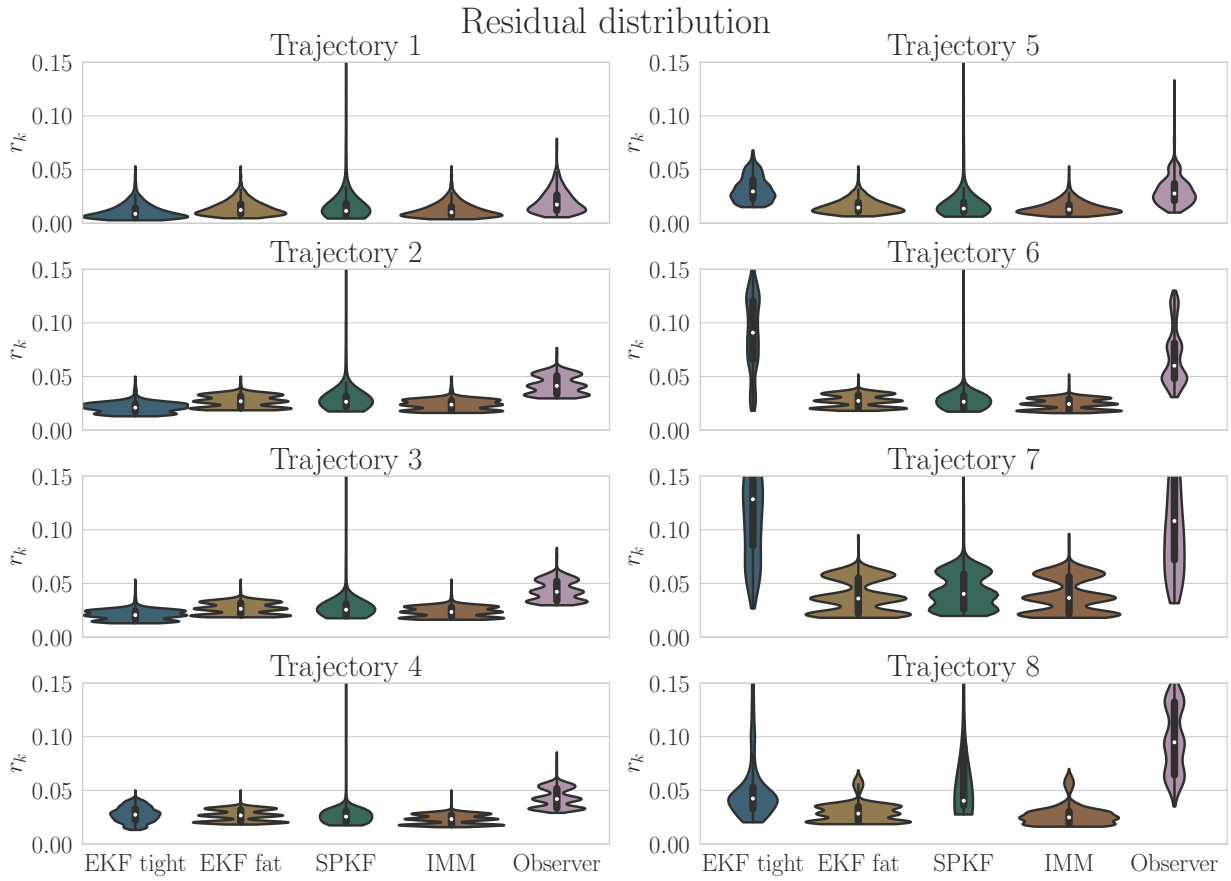


Figure 5.2: Violin plots to show distributions of r_k in tested trajectories from simulated data. EKF tight/loose, IMM, and observer are tested.

normalized estimation error squared (NEES) test [15, Sec. 5.4] on Monte Carlo runs. The NEES test involves computing a $\epsilon_k \sim \chi^2_8$ statistic using the error trajectory and the predicted covariance of such error, where χ^2_8 is the Chi-square distribution with 8 degrees of freedom,

$$\boldsymbol{\xi}_k = \log\left(\hat{\mathbf{H}}_k \mathbf{H}_k^{-1}\right)^{\vee}, \qquad \epsilon_k = \boldsymbol{\xi}_k^{\mathsf{T}} \hat{\mathbf{P}}_{hh,k}^{-1} \boldsymbol{\xi}_k, \tag{5.2}$$

where $\hat{\mathbf{P}}_{hh,k}$ is the block on the diagonal of $\hat{\mathbf{P}}_k$ corresponding to the homography state. To assess with $3\sigma \approx 99.73\%$ confidence if the estimator is consistent, ϵ_k should remain between an upper and lower threshold [15]. In Figure 5.6, ϵ_k is averaged across all 100 trials, and then

Table 5.1: Performance among the proposed filters and the observer. The metric used is r_k averaged across all Monte Carlo trials and then timesteps. A 100 Monte Carlo trials were run for each filter on each trajectory tested.

Traj.	$\mathbb{E}[r_k]$					% Diff. IMM to observer
	EKF tight	EKF fat	SPKF	IMM	Observer	
1	**0.0108**	0.0140	0.0184	0.0121	0.0201	39.5%
2	**0.0212**	0.0269	0.0316	0.0237	0.0424	44.1%
3	**0.0207**	0.0266	0.0294	0.0235	0.0433	45.7%
4	0.0274	0.0266	0.0272	**0.0231**	0.0429	46.1%
5	0.0319	0.0158	0.0194	**0.0143**	0.0308	53.4%
6	0.0897	0.0273	0.0290	**0.0242**	0.0684	64.6%
7	0.1319	**0.0383**	0.0442	0.0388	0.1268	69.4%
8	0.0501	0.0299	0.1064	**0.0272**	0.1062	74.4%

plotted as a function of time. *EKF tight* is a consistent estimator when the assumption that $\mathbf{s}_a$ is constant is respected, but as soon as the assumption is broken, the NEES values diverge, as shown in Figure 5.3. This is expected, since the errors are large. For *EKF loose*, the NEES value is below the lower threshold, producing inconsistent results since the covariance estimate is too large. However, having a large covariance is preferable to having the error be too large for the covariance, as shown in Figure 5.4. For the SPKF, the filter is sensitive to initializations with big error, showing an undesirable transient response in the first seconds of estimation, as shown in Figure 5.5. Out of the 100 Monte Carlo trials, 1 to 3 were not completed due to the filter's estimate diverging. For these reasons, the SPKF was not tested on experimental data. For the IMM, the NEES value goes below the lower threshold in some trajectories for the same reasons as *EKF loose* does, but stays closer to the lower threshold, indicating that the covariance is better modulated than *EKF loose*, as shown in Figure 5.6. Out of all filters tested, the IMM is the one with the most consistent results.

5.3 Experimental Results

An Intel Realsense D435i is used to collect data. Angular velocity measurements are provided by a built-in rate-gyro at 200 Hz. The left camera of the stereo rig is used for camera measurements of 640×480 pixels at 30 Hz. Ground truth data is collected using an OptiTrack optical motion capture system at 120 Hz. The noise parameters used are $\sigma_g = 0.022\,\mathrm{rad\,s^{-1}}$, $\sigma_r = 1\,\mathrm{pixel}$.

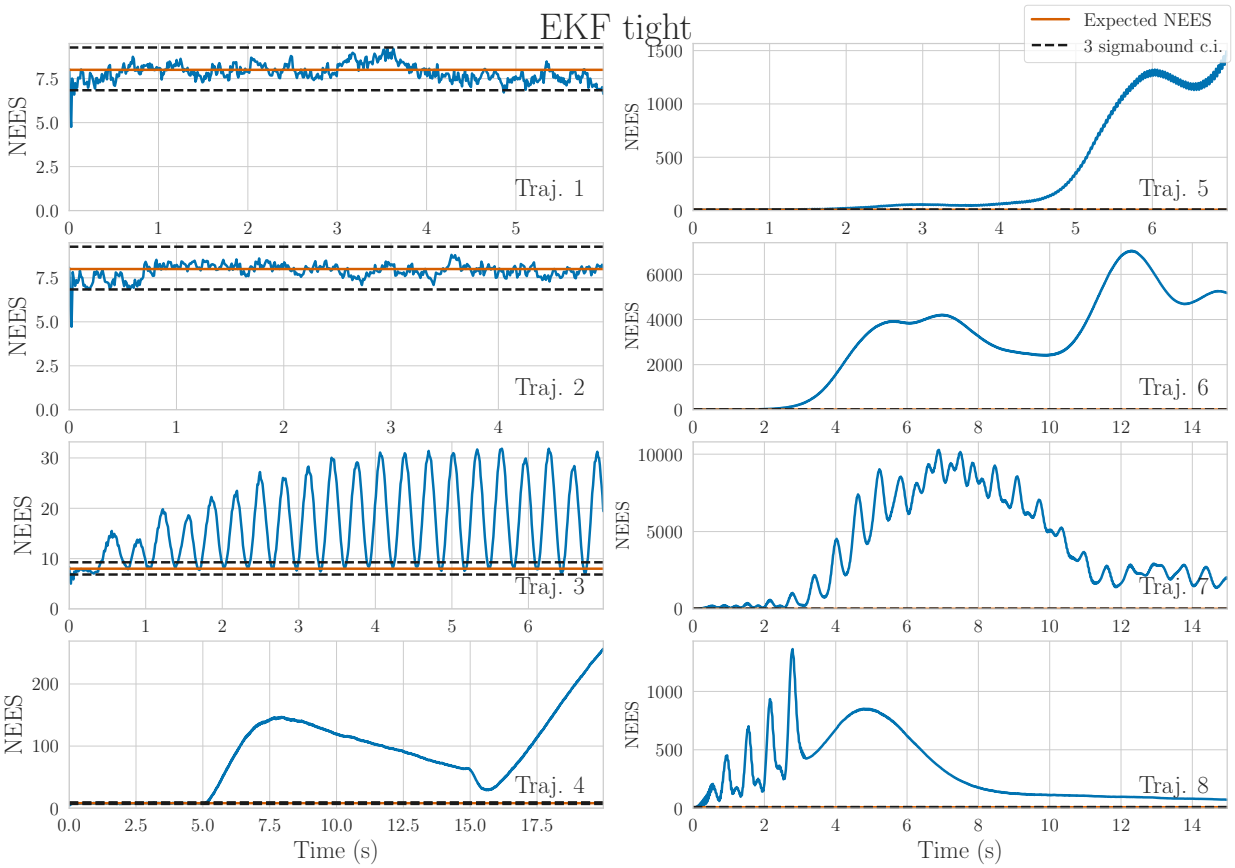


Figure 5.3: NEES plots for Monte Carlo trials of *EKF tight* approach.

Nine trials are recorded, all about 1 minute long. In each trial the camera tracks a plane about 1.5 m away, consisting of magazine pages spread on the floor, so salient features can be detected by the front-end system, as shown in Figure 5.10. The camera moves above this plane while rotating, with different paces in all trials to test how well the filters work when the assumption that $\mathbf{s}_a$ is constant in time is broken. The camera does not observe the plane for brief moments of time, and occlusions are also added in some trials to assess the robustness of the proposed approaches.

All the estimators tested, including the observer from [2], need a front-end to find point correspondences. ORB descriptors are used in its `OpenCV` implementation [30], as explained in 4.5.2. Since the goal is to estimate homography w.r.t. a reference frame, the feature points from each image are all matched against those from a reference image, which is picked from the first few frames in the trials. The outlier rejection used on each filter is the robust loss described in Sec. 4.4.

The state estimate is initialized as $\check{\mathbf{H}}_0 = \mathbf{1}, \check{\mathbf{\Gamma}}_0 = \mathbf{0}$ with $\check{\mathbf{P}}_0 = 10^{-4}\mathbf{1}$ for all trials since the reference frame was set as one of the frames recorded and the filter initialized at that same

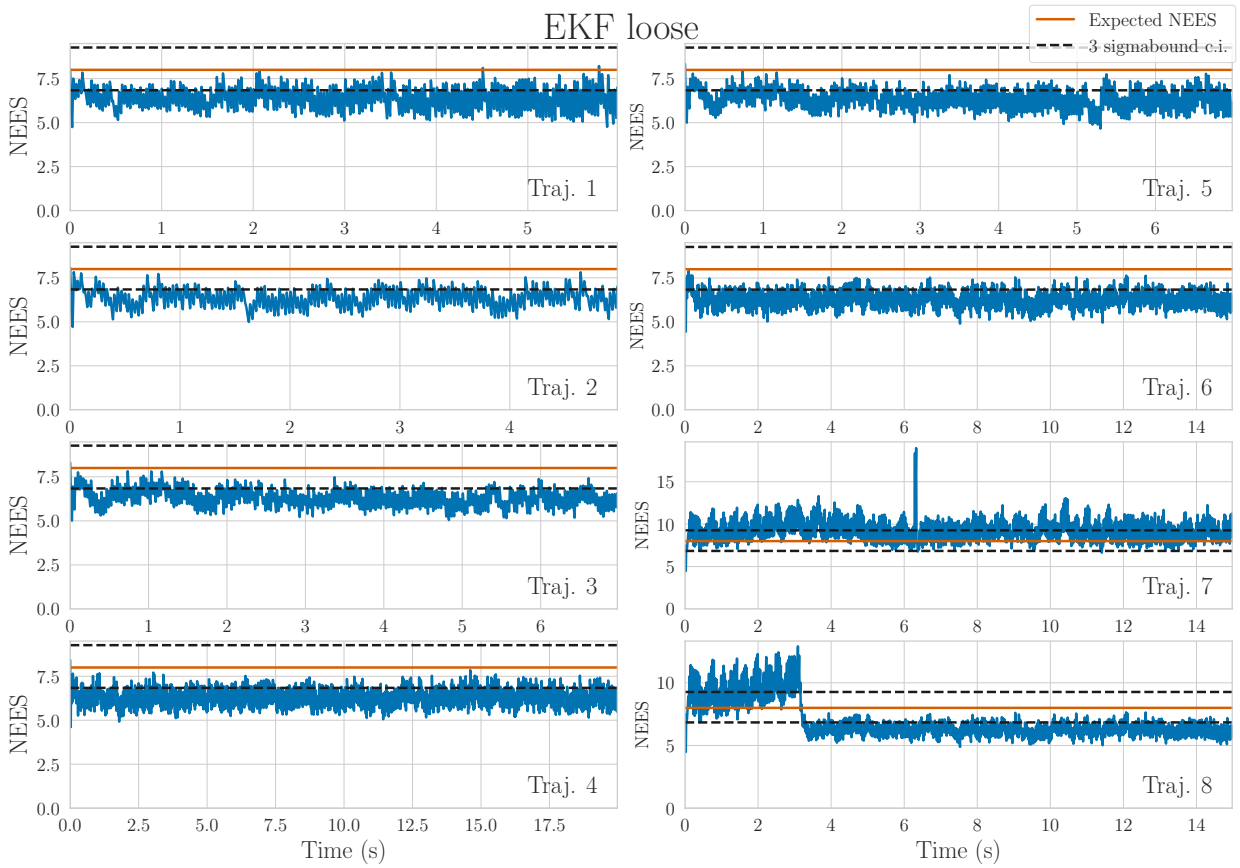


Figure 5.4: NEES plots for Monte Carlo trials of *EKF loose* approach.

frame, so the initial error is very small. The IMM is composed in this case of two iterated EKFs, the first one with $\sigma_m^2 = 10^{-6}$ and the second with $\sigma_m^2 = 1$. The robust loss function implemented to reject outliers is *SC/DCS* from [27], with $c = 9.5$, which was obtained heuristically. The transition probabilities are set as $\mathbf{\Pi} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$.

The error r_k defined in (5.1) is used for performance evaluation. In Table 5.2, it can be seen that in the recorded trajectories, the observer and the IMM have a comparable performance, with the observer performing the best in 4 out of 9 trajectories and the IMM in 4 out of 9. In Trajectory 9, the performance of the observer and the IMM is essentially the same.

To explain the different performance across the recorded trajectories, in Figure 5.7 it can be observed how in Trajectory 1, where the observer performs better, there is a period going from 41 s to 42 s where not enough features are tracked. When the plane is slowly seen again after that period, the observer is able to recover and maintain a low error, while the IMM error grows. Nonetheless, the IMM is still able to converge back to a reasonable homography estimate. Meanwhile in Trajectory 4, where the IMM performs better, in the time frames

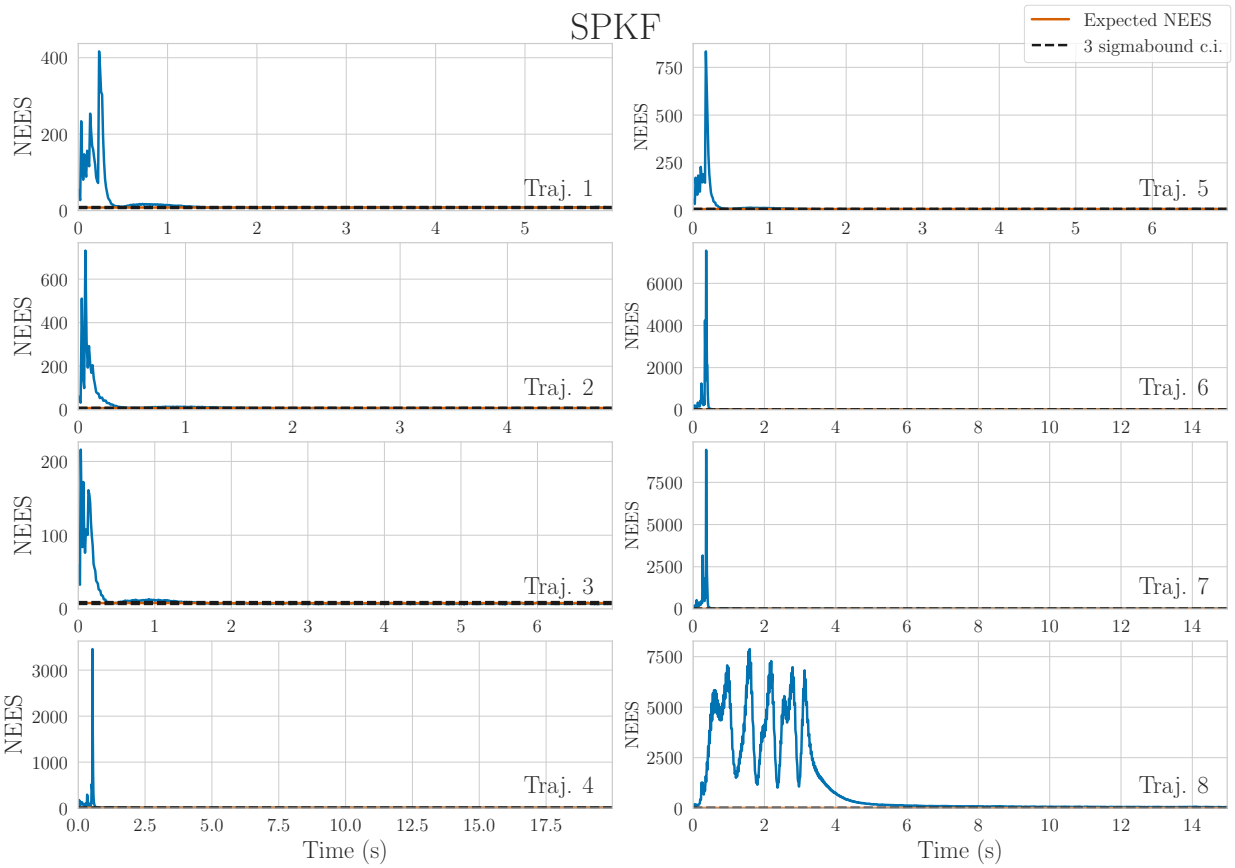


Figure 5.5: NEES plots for Monte Carlo trials of SPKF approach.

where the assumption that $\mathbf{s}_a$ is constant is violated to a larger degree, the observer's error is bigger than the IMM.

Additionally, in time frames where the assumption is violated or where there are not enough features to track, the IMM relies more on Model 2, which corresponds to the *EKF loose*, which is expected. In the first case, the process model is inaccurate, and it cannot be trusted. In the second case, the IMM grows uncertain of its estimate since it is mostly deadreckoned.

In Figure 5.8, the performance in Trajectories 5 and 6 is shown. Again, in Trajectory 5 there are many time frames where the observer has more error that the IMM and they align with time frames where the assumption that $\mathbf{s}_a$ is constant is violated. The time frames where the IMM error grows are the ones right after where there are not enough features tracked.

By analyzing also Figure 5.9, with Trajectories 7 and 8, a pattern appears. The IMM outperforms the observer in the situations where the assumption that $\mathbf{s}_a$ is constant is violated, as long as the number of point matches is enough to determine the homography. On the other hand, the observer performs best when there are not enough point correspondences

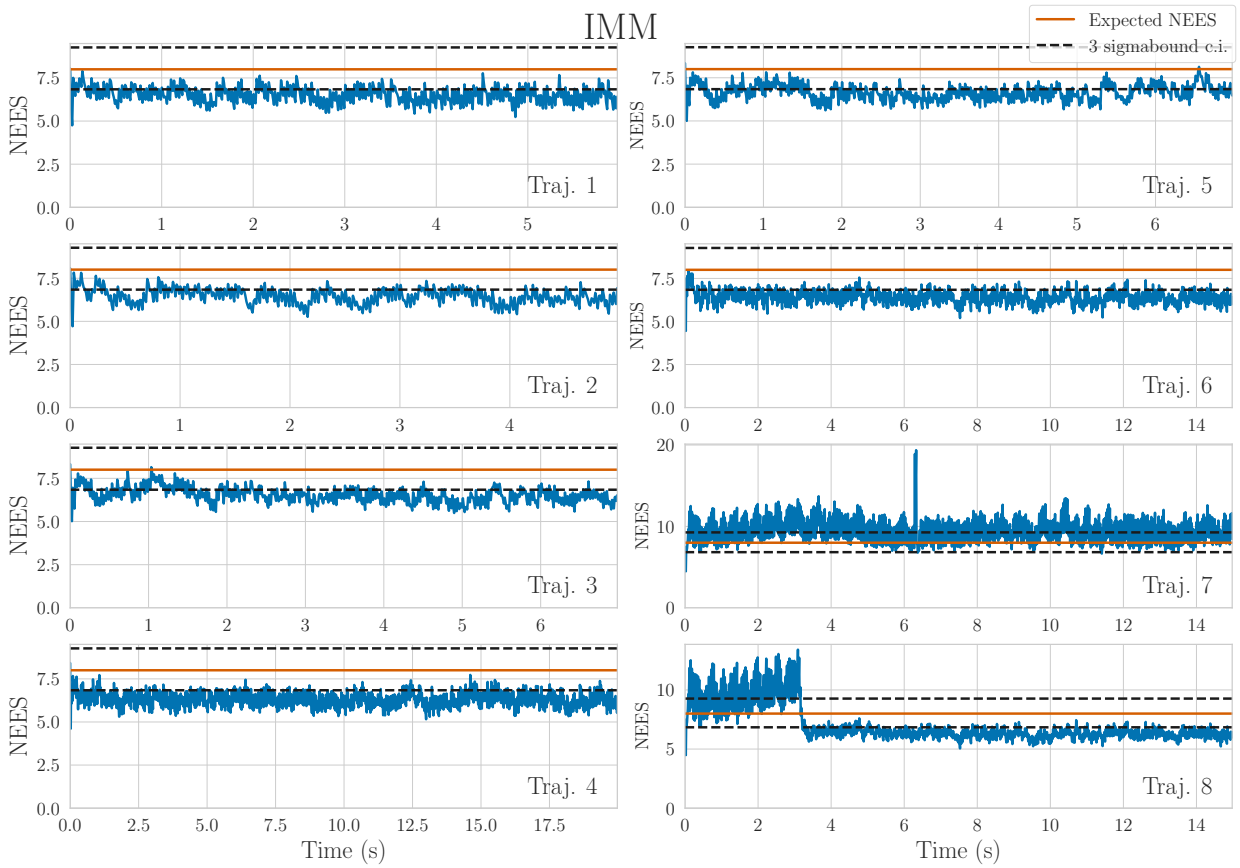


Figure 5.6: NEES plots for Monte Carlo trials of IMM approach.

for a short period of time.

Unlike in simulations, the IMM is not the clear winner over the observer in an experimental setting. However, even if the observer shows good robustness in an experimental setting, especially when the estimate drifts away, the IMM offers better performance in varied trajectories while also offering covariance information.

Table 5.2: Performance of EKF loose, IMM, and observer using experimental data. The error r_k is averaged across all time steps.

	$\mathbb{E}[r_k]$			
Traj.	EKF loose	IMM	Observer	% Diff. IMM/Obs.
1	0.0897	0.0896	**0.0589**	-52.1%
2	0.1256	0.0982	**0.0943**	-4.2%
3	Diverged	**0.0785**	0.2915	73.1%
4	0.0912	**0.0765**	0.1852	58.7%
5	0.0928	**0.0690**	0.1043	33.8%
6	0.0887	0.0634	**0.0485**	-30.9%
7	0.0994	**0.0801**	0.0948	15.6%
8	Diverged	0.0699	**0.0585**	-19.6%
9	0.1434	**0.1237**	**0.1238**	0.1%

Figure 5.7: IMM model probabilities depending on how the $\mathbf{s}_a$ is constant assumption is broken and number of matches in Trajectories 1 and 4. Mode 1 corresponds to $\sigma_m^2 = 10^{-6}$ and mode 2 corresponds to $\sigma_m^2 = 1$. Evolution of IMM and observer's error r_k is shown in bottom plots. Green shaded regions represent time frames where $\|\dot{\mathbf{s}}_a\| > \alpha = 155$. Orange shaded regions represent time frames where number of tracked features goes below 4. When $\|\dot{\mathbf{s}}_a\| > \alpha$, the observer performance suffers. When insufficient features are tracked, the IMM performance suffers.

Figure 5.8: IMM model probabilities depending on how the $\mathbf{s}_a$ is constant assumption is broken and number of matches in Trajectories 5 and 6. Mode 1 corresponds to $\sigma_m^2 = 10^{-6}$ and mode 2 corresponds to $\sigma_m^2 = 1$. Evolution of IMM and observer's error r_k is shown in bottom plots. Green shaded regions represent time frames where $\|\dot{\mathbf{s}}_a\| > \alpha = 155$. Orange shaded regions represent time frames where number of tracked features goes below 4. When $\|\dot{\mathbf{s}}_a\| > \alpha$, the observer performance suffers. When insufficient features are tracked, the IMM performance suffers.

Figure 5.9: IMM model probabilities depending on how the $\mathbf{s}_a$ is constant assumption is broken and number of matches in Trajectories 7 and 8. Mode 1 corresponds to $\sigma_m^2 = 10^{-6}$ and mode 2 corresponds to $\sigma_m^2 = 1$. Evolution of IMM and observer's error r_k is shown in bottom plots. Green shaded regions represent time frames where $\|\dot{\mathbf{s}}_a\| > \alpha = 155$. Orange shaded regions represent time frames where number of tracked features goes below 4. When $\|\dot{\mathbf{s}}_a\| > \alpha$, the observer performance suffers. When insufficient features are tracked, the IMM performance suffers.

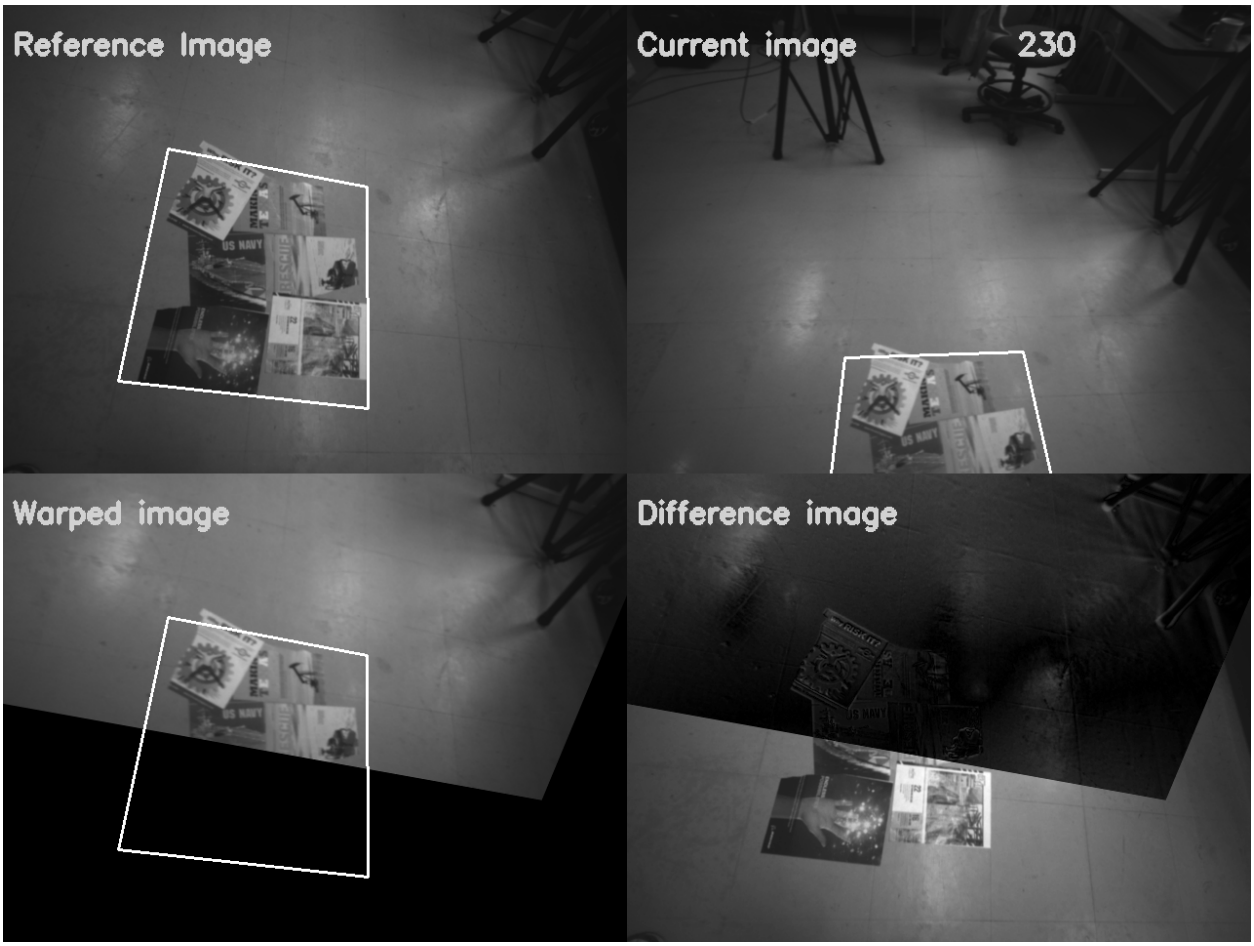


Figure 5.10: Experimental method which consisted of tracking features on the ground. The white square is used to visualize the estimated homography. The frame at the illustrated timestamp has the pattern, which is partially out of camera view, approximately realigned with the reference frame by the current homography matrix estimate. The current image is warped by the homography estimate on the left bottom.

Chapter 6

Closing Remarks and Future Work

6.1 Conclusion

In this thesis, the problem of estimating homography using rate-gyro and camera measurements is addressed. This thesis' novelty lies in the use of the Bayesian filtering framework in concert with a simplified process model. In particular, two iterated EKFs are used within an IMM filter. The approach was compared to a nonlinear deterministic observer in both simulation and experiments where better or comparable performance is realized, depending on the situation.

The proposed algorithm performs better than the observer in the situations it was designed for, that is, varied trajectories that may or may not break the assumption used. The observer on the other hand, shows good robustness in cases where the error between homography estimate and true value is big, showing good convergence properties.

Reiterating, this thesis does not propose to displace the homography observer, but rather present an alternative approach to estimate homography. In situations where it is known that the assumption will be constantly violated, an IMM could be a better choice over the observer. If it is known that many occlusions will pervade the dataset, with slow-varying trajectories, the observer could be preferable.

Also, the proposed Bayesian approach offers covariance information, unlike the observer. This opens the avenue for adaptive filtering, as in this thesis, but also post-processing procedures such as low-confidence estimate removal, smoothing, loop-closure detection, and quality control.

6.2 Future Work

This work can be improved upon in several directions. First, a smoothing algorithm that takes advantage of the covariance information could be applied for the IMM filter. This would provide a complete framework for estimating homography in real-time with an IMM filter and post-processing capabilities to remove outliers from the estimated values.

In this work, feature-based approaches where employed in the front-end to find point correspondences. The approach implemented using ORB descriptors needs to be further validated and compared to other feature-based approaches. Also, other options exist that were not explored. Direct methods, that take the whole image into account and not only selected points to find the best homography estimate can be explored. They have shown promise in textureless environments, that is, with few salient features. Underwater or indoor environments are examples that can tend to have planar features, but not many salient, distinctive features to track. A direct method could enable homography estimation in those environments.

Another area of improvement is implementing a better robust loss function in the correction step of the EKF. The robust loss was tuned manually and it is unable to adapt to distinct distributions of the residuals. An adaptive robust loss function is expected to offer better robustness.

Another limitation of this work is that no rate-gyro biases are estimated. Biases can degrade the performance of the estimator. Estimating them and taking them into account could improve this algorithm.

In this work, the state estimated was homography, and while it has many uses, other states such as attitude or position have more applications. In the future a Bayesian filter can be developed to take advantage of homography constraints to estimate a more "standard" state, such as attitude and position.

Appendices

Appendix A

Nonlinear Observer

A.1 Homography Estimation Employing an Observer

In [12], the authors proposed a nonlinear deterministic observer to estimate the homography including not only point correspondences, but also rate-gyro measurements via the kinematic model described in Sec. 2.5, with stability properties known *a priori*. The advantage of this approach is that it is more robust against occlusions, when not enough point correspondences are available to compute homography, as in the Direct Linear Transform (DLT) algorithm.

The observer, similar to the Bayesian filter structure, has a prediction step and a correction step. The observer uses (2.63) and (2.71) to include the rate-gyro measurements in the prediction step, more frequent, and correct them with point correspondences from a plane, which are less frequent. The observer is designed as

$$\dot{\hat{\mathbf{H}}}_{ab} = \hat{\mathbf{H}}_{ab}\left(\boldsymbol{\omega}_b^{ba\times} + \hat{\boldsymbol{\Gamma}}_b^{ba}\right) + k_p \mathbf{Z}_{ab}\hat{\mathbf{H}}_{ab} \tag{A.1a}$$

$$\dot{\hat{\boldsymbol{\Gamma}}}_b^{ba} = \left[\hat{\boldsymbol{\Gamma}}_b^{ba}, \boldsymbol{\omega}_b^{ba\times}\right] + k_i \mathrm{Ad}(\hat{\mathbf{H}}_{ab}^{\mathsf{T}})\mathbf{Z}_{ab} \tag{A.1b}$$

where k_p, k_i are constant gains, $\mathbf{Z}_{ab} \in \mathfrak{sl}(3)$ is the innovation term obtained from the point correspondences. The error in the estimation is going to be defined in a right invariant manner,

$$\tilde{\mathbf{H}}_{ab} = \hat{\mathbf{H}}_{ab}\mathbf{H}_{ab}^{-1} \tag{A.2}$$

$$\tilde{\boldsymbol{\Gamma}}_b^{ba} = \boldsymbol{\Gamma}_b^{ba} - \hat{\boldsymbol{\Gamma}}_b^{ba} \tag{A.3}$$

where $\hat{\mathbf{H}}_{ab}$ is the estimated value and $\mathbf{H}_{ab}$ corresponds to the true value of homography. The goal of the observer is to drive $\tilde{\mathbf{H}}_{ab} \to \mathbf{1}$ and $\tilde{\boldsymbol{\Gamma}}_b^{ba} \to \mathbf{0}$. In other words, to drive the error to zero.

A.1.1 Measurement Model

The measurement model employed in the observer uses $\mathbf{H}_{ab} \triangleq \mathbf{K}\mathbf{H}_{ab}\mathbf{K}^{-1}$ as well, that is, the *image homography*,

$$\hat{\mathbf{q}}_a^i = \frac{\hat{\mathbf{H}}_{ab}\mathbf{q}_b^i}{\left\|\hat{\mathbf{H}}_{ab}\mathbf{q}_b^i\right\|_2} \tag{A.4}$$

$$= \frac{\hat{\mathbf{H}}_{ab}\mathbf{H}_{ab}^{-1}\mathbf{q}_a^i}{\left\|\hat{\mathbf{H}}_{ab}\mathbf{H}_{ab}^{-1}\mathbf{q}_a^i\right\|_2} \tag{A.5}$$

$$= \frac{\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i}{\left\|\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i\right\|_2}, \tag{A.6}$$

where $\hat{\mathbf{q}}_a^i$ is the estimated pixel coordinates of the point p_i in the image plane resolved in the camera frame $\mathcal{F}_a$ and $\mathbf{q}_a^i$ is the true value.

A.1.2 Stability Proof of the Observer

The stability of the observer is proven in [12] using the Lyapunov stability tools. An important definition from [12] for the subsequent stability proofs is the following.

Definition A.1.1. A set of $n \geq 4$ measurements $\mathcal{M}_n$ of $\mathbf{q}_a^i$ where $i \in \{1, \cdots, n\}$ is *consistent* if there is $\mathcal{M}_4 \subset \mathcal{M}_n$ such that any combination of three vectors are linearly independent. This also implies that any $\mathbf{q}_a^i \in \mathcal{M}_4$ can be written as a linear combination of the remaining three vectors.

Theorem A.1.1. Consider a camera moving with the kinematic model described in (2.53) viewing a planar scene where $\mathcal{F}_a$ is static, ${\boldsymbol{\omega}_b^{ba}}^\times$ is bounded and the homography kinematics are described with (2.63) and (2.71). Consider the observer in (A.1), assume $\mathbf{H}_{ab}$ is bounded, and consider an innovation $\mathbf{Z}_{ab}$ defined as

$$\mathbf{Z}_{ab} = \sum_{i=1}^{n} \mathbf{g}_a^i \mathbf{q}_a^i \hat{\mathbf{q}}_a^{i\,\mathsf{T}}. \tag{A.7}$$

where $\mathbf{g}_a^i = \left(\mathbf{1} - \hat{\mathbf{q}}_a^i \hat{\mathbf{q}}_a^{i\,\mathsf{T}}\right)$. Then, if $\mathcal{M}_n$ is consistent, the equilibrium point $(\tilde{\mathbf{H}}_{ab}, \tilde{\mathbf{\Gamma}}_b^{ba}) = (\mathbf{1}, \mathbf{0})$ is asymptotically stable.

Proof. First, a Lyapunov function candidate is defined as

$$\mathcal{L}(\hat{\mathbf{H}}_{ab}, \hat{\mathbf{\Gamma}}_b^{ba}) = \sum_{i=1}^{n} \frac{1}{2} \left\|\hat{\mathbf{q}}_a^i - \mathbf{q}_a^i\right\|_2^2 + \frac{1}{2k_i} \left\|\tilde{\mathbf{\Gamma}}_b^{ba}\right\|_{\mathsf{F}}^2, \tag{A.8}$$

which is positive definite. Differentiating, the Lyapunov function candidate results in

$$\dot{\mathcal{L}}(\hat{\mathbf{H}}_{ab}, \hat{\mathbf{\Gamma}}_b^{ba}) = \sum_{i=1}^{n} \left(\hat{\mathbf{q}}_a^i - \mathbf{q}_a^i\right)^\mathsf{T} \dot{\hat{\mathbf{q}}}_a^i + \frac{1}{k_i} \mathrm{tr}\left(\tilde{\mathbf{\Gamma}}_b^{ba\,\mathsf{T}} \dot{\tilde{\mathbf{\Gamma}}}_b^{ba}\right) \tag{A.9}$$

$$= \sum_{i=1}^{n} \left(\hat{\mathbf{q}}_a^i - \mathbf{q}_a^i\right)^\mathsf{T} \dot{\hat{\mathbf{q}}}_a^i + \frac{1}{k_i} \mathrm{tr}\left(\tilde{\mathbf{\Gamma}}_b^{ba\,\mathsf{T}} \left(\left[\boldsymbol{\omega}_b^{ba\times}, \mathbf{\Gamma}_b^{ba}\right] - \left[\boldsymbol{\omega}_b^{ba\times}, \hat{\mathbf{\Gamma}}_b^{ba}\right] - k_i \mathrm{Ad}(\hat{\mathbf{H}}_{ab}^\mathsf{T}) \mathbf{Z}_{ab}\right)\right) \tag{A.10}$$

$$= \sum_{i=1}^{n} \left(\hat{\mathbf{q}}_a^i - \mathbf{q}_a^i\right)^\mathsf{T} \dot{\hat{\mathbf{q}}}_a^i + \frac{1}{k_i} \mathrm{tr}\left(\tilde{\mathbf{\Gamma}}_b^{ba\,\mathsf{T}} \left(\left[\boldsymbol{\omega}_b^{ba\times}, \tilde{\mathbf{\Gamma}}_b^{ba}\right] - k_i \mathrm{Ad}(\hat{\mathbf{H}}_{ab}^\mathsf{T}) \mathbf{Z}_{ab}\right)\right) \tag{A.11}$$

$$= \sum_{i=1}^{n} \left(\hat{\mathbf{q}}_a^i - \mathbf{q}_a^i\right)^\mathsf{T} \dot{\hat{\mathbf{q}}}_a^i - \mathrm{tr}\left(\tilde{\mathbf{\Gamma}}_b^{ba\,\mathsf{T}} \mathrm{Ad}(\hat{\mathbf{H}}_{ab}^\mathsf{T}) \mathbf{Z}_{ab}\right), \tag{A.12}$$

where $\tilde{\mathbf{\Gamma}}_b^{ba\,\mathsf{T}} \left(\left[\boldsymbol{\omega}_b^{ba\times}, \tilde{\mathbf{\Gamma}}_b^{ba}\right]\right)$ evaluates to $\mathbf{0}$ since the products in $\left[\boldsymbol{\omega}_b^{ba\times}, \tilde{\mathbf{\Gamma}}_b^{ba}\right]$ are orthogonal to $\tilde{\mathbf{\Gamma}}_b^{ba}$. As can be seen, $\dot{\hat{\mathbf{q}}}_a^i$ is needed. By using (A.4) and differentiating,

$$\dot{\hat{\mathbf{q}}}_a = \frac{\dot{\hat{\mathbf{H}}}_{ab} \mathbf{H}_{ab}^{-1} \mathbf{q}_a + \hat{\mathbf{H}}_{ab} \dot{\mathbf{H}}_{ab}^{-1} \mathbf{q}_a}{\left\|\tilde{\mathbf{H}}_{ab} \mathbf{q}_a\right\|_2} - \frac{\tilde{\mathbf{H}}_{ab} \mathbf{q}_a \mathbf{q}_a^\mathsf{T} \tilde{\mathbf{H}}_{ab}^\mathsf{T} \left(\dot{\hat{\mathbf{H}}}_{ab} \mathbf{H}_{ab}^{-1} \mathbf{q}_a + \hat{\mathbf{H}}_{ab} \dot{\mathbf{H}}_{\mathbf{ab}}^{\mathbf{-1}} \mathbf{q}_a\right)}{\left\|\tilde{\mathbf{H}}_{ab} \mathbf{q}_a\right\|_2^3} \tag{A.13}$$

$$= \mathbf{g}_a^i \frac{\dot{\hat{\mathbf{H}}}_{ab} \mathbf{H}_{ab}^{-1} \mathbf{q}_a + \hat{\mathbf{H}}_{ab} \dot{\mathbf{H}}_{\mathbf{ab}}^{\mathbf{-1}} \mathbf{q}_a}{\left\|\tilde{\mathbf{H}}_{ab} \mathbf{q}_a\right\|_2} \tag{A.14}$$

$$= \mathbf{g}_a^i \frac{\hat{\mathbf{H}}_{ab} \left(\boldsymbol{\omega}_b^{ba\times} + \hat{\mathbf{\Gamma}}_b^{ba}\right) \mathbf{H}_{ab}^{-1} \mathbf{q}_a + k_p \mathbf{Z}_{ab} \hat{\mathbf{H}}_{ab} \mathbf{H}_{ab}^{-1} \mathbf{q}_a - \hat{\mathbf{H}}_{ab} \left(\boldsymbol{\omega}_b^{ba\times} + \mathbf{\Gamma}_b^{ba}\right) \mathbf{H}_{ab}^{-1} \mathbf{q}_a}{\left\|\tilde{\mathbf{H}}_{ab} \mathbf{q}_a\right\|_2} \tag{A.15}$$

$$= \mathbf{g}_a^i \left(k_p \mathbf{Z}_{ab} - \mathrm{Ad}(\hat{\mathbf{H}}_{ab}) \tilde{\mathbf{\Gamma}}_b^{ba}\right) \hat{\mathbf{q}}_a. \tag{A.16}$$

Substituting (A.16) back into (A.12),

$$\dot{\mathcal{L}}(\hat{\mathbf{H}}_{ab}, \hat{\mathbf{\Gamma}}_b^{ba}) = \sum_{i=1}^{n} \left(\hat{\mathbf{q}}_a^i - \mathbf{q}_a^i\right)^\mathsf{T} \mathbf{g}_a^i \left(k_p \mathbf{Z}_{ab} - \mathrm{Ad}(\hat{\mathbf{H}}_{ab}) \tilde{\mathbf{\Gamma}}_b^{ba}\right) \hat{\mathbf{q}}_a - \mathrm{tr}\left(\tilde{\mathbf{\Gamma}}_b^{ba\,\mathsf{T}} \mathrm{Ad}(\hat{\mathbf{H}}_{ab}^\mathsf{T}) \mathbf{Z}_{ab}\right) \tag{A.17}$$

$$= -\sum_{i=1}^{n} \mathbf{q}_a^{i\,\mathsf{T}} \mathbf{g}_a^i \left(k_p \mathbf{Z}_{ab} - \mathrm{Ad}(\hat{\mathbf{H}}_{ab}) \tilde{\mathbf{\Gamma}}_b^{ba}\right) \hat{\mathbf{q}}_a - \mathrm{tr}\left(\mathrm{Ad}(\hat{\mathbf{H}}_{ab}^{-1}) \mathbf{Z}_{ab}^\mathsf{T} \tilde{\mathbf{\Gamma}}_b^{ba}\right) \tag{A.18}$$

$$= -\mathrm{tr}\left(\sum_{i=1}^{n} \hat{\mathbf{q}}_a \mathbf{q}_a^{i\,\mathsf{T}} \mathbf{g}_a^i \left(k_p \mathbf{Z}_{ab} - \mathrm{Ad}(\hat{\mathbf{H}}_{ab}) \tilde{\mathbf{\Gamma}}_b^{ba}\right) + \mathrm{Ad}(\hat{\mathbf{H}}_{ab}^{-1}) \mathbf{Z}_{ab}^\mathsf{T} \tilde{\mathbf{\Gamma}}_b^{ba}\right) \tag{A.19}$$

$$= -\mathrm{tr}\left(k_p \sum_{i=1}^{n} \hat{\mathbf{q}}_a \mathbf{q}_a^{i\,\mathsf{T}} \mathbf{g}_a^i \mathbf{Z}_{ab} + \mathrm{Ad}(\hat{\mathbf{H}}_{ab}^{-1}) \left[\mathbf{Z}_{ab}^\mathsf{T} - \sum_{i=1}^{n} \hat{\mathbf{q}}_a \mathbf{q}_a^{i\,\mathsf{T}} \mathbf{g}_a^i\right] \tilde{\mathbf{\Gamma}}_b^{ba}\right), \tag{A.20}$$

applying the proposed innovation in (A.7),

$$\dot{\mathcal{L}}(\hat{\mathbf{H}}_{ab}, \hat{\mathbf{\Gamma}}_b^{ba}) = -k_p \left\|\mathbf{Z}_{ab}\right\|_\mathsf{F}^2 \tag{A.21}$$

which shows that the derivative of the Lyapunov function is negative semidefinite and equals 0 when $\mathbf{Z}_{ab} = \mathbf{0}$. It can be seen that $\mathcal{L}(\hat{\mathbf{H}}_{ab}, \hat{\mathbf{\Gamma}}_b^{ba}) \leq \mathcal{L}(\hat{\mathbf{H}}_0, \hat{\mathbf{\Gamma}}_0)$. Considering also that $\boldsymbol{\omega}_b^{ba\times}$ is bounded, $\hat{\mathbf{H}}_{ab}$ and $\hat{\mathbf{\Gamma}}_b^{ba}$ are bounded. Therefore, to show by Barbalat's Lemma that $\mathbf{Z}_{ab} \to \mathbf{0}$ as $t \to \infty$, it remains to show that $\ddot{\mathcal{L}}(\hat{\mathbf{H}}_{ab}, \hat{\mathbf{\Gamma}}_b^{ba})$ is bounded,

$$\ddot{\mathcal{L}}(\hat{\mathbf{H}}_{ab}, \hat{\mathbf{\Gamma}}_b^{ba}) = -2k_p \mathrm{tr}\left(\mathbf{Z}_{ab}^\mathsf{T} \dot{\mathbf{Z}}_{ab}\right) \tag{A.22}$$

$$= -2k_p \mathrm{tr}\left(\mathbf{Z}_{ab}^\mathsf{T} \sum_{i=1}^{n} \left(\dot{\mathbf{g}}_a^i \mathbf{q}_a^i \hat{\mathbf{q}}_a^{i\,\mathsf{T}} + \mathbf{g}_a^i \mathbf{q}_a^i \dot{\hat{\mathbf{q}}}_a^{i\,\mathsf{T}}\right)\right) \tag{A.23}$$

$$= -2k_p \mathrm{tr}\left(\mathbf{Z}_{ab}^\mathsf{T} \sum_{i=1}^{n} \left(\left(\dot{\hat{\mathbf{q}}}_a^i \hat{\mathbf{q}}_a^{i\,\mathsf{T}} + \hat{\mathbf{q}}_a^i \dot{\hat{\mathbf{q}}}_a^{i\,\mathsf{T}}\right) \mathbf{q}_a^i \hat{\mathbf{q}}_a^{i\,\mathsf{T}} + \mathbf{g}_a^i \mathbf{q}_a^i \dot{\hat{\mathbf{q}}}_a^{i\,\mathsf{T}}\right)\right) \tag{A.24}$$

From the previous results, $\mathbf{Z}_{ab}$ and $\dot{\mathbf{Z}}_{ab}$ are bounded, showing $\ddot{\mathcal{L}}(\hat{\mathbf{H}}_{ab}, \hat{\mathbf{\Gamma}}_b^{ba})$ boundedness and $\mathbf{Z}_{ab} \to \mathbf{0}$.

Now, to prove that $\tilde{\mathbf{H}}_{ab} \to \mathbf{1}$, first consider

$$\mathbf{Z}_{ab}\tilde{\mathbf{H}}_{ab}^{-\mathsf{T}} = \sum_{i=1}^{n} \left(\mathbf{1} - \frac{\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i \mathbf{q}_a^{i\,\mathsf{T}} \tilde{\mathbf{H}}_{ab}^\mathsf{T}}{\left\|\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i\right\|_2^2}\right) \frac{\mathbf{q}_a^i \mathbf{q}_a^{i\,\mathsf{T}}}{\left\|\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i\right\|_2}. \tag{A.25}$$

Computing the trace,

$$\mathrm{tr}\left(\mathbf{Z}_{ab}\tilde{\mathbf{H}}_{ab}^{-\mathsf{T}}\right) = \sum_{i=1}^{n} \frac{1}{\left\|\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i\right\|_2^3} \left(\left\|\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i\right\|_2^2 \mathrm{tr}(\mathbf{q}_a^i \mathbf{q}_a^{i\,\mathsf{T}}) - \mathrm{tr}(\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i \mathbf{q}_a^{i\,\mathsf{T}} \tilde{\mathbf{H}}_{ab}^\mathsf{T} \mathbf{q}_a^i \mathbf{q}_a^{i\,\mathsf{T}})\right) \tag{A.26}$$

$$= \sum_{i=1}^{n} \frac{1}{\left\|\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i\right\|_2^3} \left(\left\|\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i\right\|_2^2 \left\|\mathbf{q}_a^i\right\|_2^2 - \left((\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i)^\mathsf{T} \mathbf{q}_a^i\right)^2\right). \tag{A.27}$$

When $\mathbf{Z}_{ab} \to \mathbf{0}$, the trace equals 0, therefore

$$\left\|\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i\right\|_2 \left\|\mathbf{q}_a^i\right\|_2 = (\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i)^\mathsf{T} \mathbf{q}_a^i, \tag{A.28}$$

which can be rewritten as

$$\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i = \lambda_i \mathbf{q}_a^i, \tag{A.29}$$

for a non-zero $\lambda_i = \left\|\tilde{\mathbf{H}}_{ab}\mathbf{q}_a^i\right\|_2$. This means that λ_i and $\mathbf{q}_a^i$ and the eigenvalues and eigenvectors of $\tilde{\mathbf{H}}_{ab}$, correspondingly. Using the fact that the measurements produce a consistent set, any

point measurement can be written as the linear combination of some three point measurements,

$$\lambda_k \mathbf{q}_a^k = \lambda_k \sum_{i=1}^{3} \alpha_i \mathbf{q}_a^i \tag{A.30}$$

$$= \sum_{i=1}^{3} \alpha_i \tilde{\mathbf{H}}_{ab} \mathbf{q}_a^i \tag{A.31}$$

$$= \sum_{i=1}^{3} \alpha_i \lambda_i \mathbf{q}_a^i \tag{A.32}$$

with $\alpha_i \neq 0$ such that $\left\|\mathbf{q}_a^k\right\|_2 = 1$. Since λ_k can be seen as a fourth eigenvalue of $\tilde{\mathbf{H}}_{ab}$, and $\det(\tilde{\mathbf{H}}_{ab}) = \prod_{i=1}^{n} \lambda_i = 1$, it follows that $\lambda_k = \lambda_i = 1$, so $\tilde{\mathbf{H}}_{ab} \to \mathbf{1}$ when $\mathbf{Z}_{ab} \to \mathbf{0}$. With this result in mind, by inspecting the Lyapunov function (A.8), it can be deduced that over time $\tilde{\mathbf{\Gamma}}_b^{ba}$ becomes a constant. To prove that it converges to $\mathbf{0}$, the derivative of $\tilde{\mathbf{H}}_{ab}$ is computed,

$$\dot{\tilde{\mathbf{H}}}_{ab} = \dot{\hat{\mathbf{H}}}_{ab} \mathbf{H}_{ab}^{-1} - \hat{\mathbf{H}}_{ab} \mathbf{H}_{ab}^{-1} \dot{\mathbf{H}}_{ab} \mathbf{H}_{ab}^{-1} \tag{A.33}$$

$$= \hat{\mathbf{H}}_{ab} \left(\boldsymbol{\omega}_b^{ba\times} + \hat{\mathbf{\Gamma}}_b^{ba} \right) \mathbf{H}_{ab}^{-1} + k_p \mathbf{Z}_{ab} \hat{\mathbf{H}}_{ab} \mathbf{H}_{ab}^{-1} - \hat{\mathbf{H}}_{ab} \left(\boldsymbol{\omega}_b^{ba\times} + \mathbf{\Gamma}_b^{ba} \right) \mathbf{H}_{ab}^{-1}, \tag{A.34}$$

since $\mathbf{Z}_{ab} = \mathbf{0}$ and $\tilde{\mathbf{H}}_{ab} = \mathbf{1}$ at the limit,

$$\lim_{t \to \infty} \dot{\tilde{\mathbf{H}}}_{ab} = -\mathrm{Ad}(\hat{\mathbf{H}}_{ab}) \tilde{\mathbf{\Gamma}}_b^{ba} = \mathbf{0}, \tag{A.35}$$

and considering $\hat{\mathbf{H}}_{ab}$ is bounded, $\tilde{\mathbf{\Gamma}}_b^{ba} \to \mathbf{0}$. □

Bibliography

[1] S. Benhimane and E. Malis, "Homography-based 2D visual tracking and servoing", *The Int. J. of Robotics Research*, vol. 26, no. 7, pp. 661–676, 2007.

[2] M.-D. Hua, J. Trumpf, T. Hamel, R. Mahony, and P. Morin, "Feature-based recursive observer design for homography estimation and its application to image stabilization", *Asian J. of Control*, vol. 21, no. 4, pp. 1443–1458, 2019.

[3] V. Grabe, H. H. Bülthoff, D. Scaramuzza, and P. R. Giordano, "Nonlinear ego-motion estimation from optical flow for online control of a quadrotor uav", *The Int. J. of Robotics Research*, vol. 34, no. 8, pp. 1114–1135, 2015.

[4] M.-D. Hua, N. Manerikar, T. Hamel, and C. Samson, "Attitude, linear velocity and depth estimation of a camera observing a planar target using continuous homography and inertial data", in *2018 IEEE Int. Conf. on Robotics and Auto. (ICRA)*, IEEE, 2018, pp. 1429–1435.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system", *IEEE Trans. on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[6] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[7] A. Agarwal, C. Jawahar, and P. Narayanan, "A survey of planar homography estimation techniques", *Centre for Visual Information Technology, Tech. Rep. II-IT/TR/2005/12*, 2005.

[8] J. Y. Kaminski and A. Shashua, "Multiple view geometry of general algebraic curves", *Int. J. of Computer Vision*, vol. 56, pp. 195–219, 2004.

[9] S. Benhimane and E. Malis, "Real-time image-based tracking of planes using efficient second-order minimization", in *2004 IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS)*, IEEE, vol. 1, 2004, pp. 943–948.

[10] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation", *arXiv preprint arXiv:1606.03798*, 2016.

[11] T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar, "Unsupervised deep homography: A fast and robust homography estimation model", *IEEE Robotics and Auto. Letters*, vol. 3, no. 3, pp. 2346–2353, 2018.

[12] T. Hamel, R. Mahony, J. Trumpf, P. Morin, and M.-D. Hua, "Homography estimation on the special linear group based on direct point correspondence", in *2011 50th IEEE Conf. on Decision and Control and European Control Conf.*, 2011, pp. 7902–7908.

[13] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, vol. 3.

[14] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Homography based Kalman filter for mosaic building: Applications to UAV position estimation", in *Proceedings 2007 IEEE Int. Conf. on Robotics and Auto.*, IEEE, 2007, pp. 2004–2009.

[15] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[16] X. R. Li and Y. Bar-Shalom, "A recursive multiple model approach to noise identification", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 30, no. 3, pp. 671–684, 1994.

[17] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

[18] J. Solà, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics", pp. 1–17, 2018. arXiv: `1812.01537`. [Online]. Available: `http://arxiv.org/abs/1812.01537`.

[19] B. C. Hall and B. C. Hall, *Lie groups, Lie algebras, and representations*. Springer, 2013.

[20] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems", *IEEE Trans. on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[21] E. Eade, "Lie groups for computer vision", *Cambridge Univ., Cambridge, UK, Tech. Rep*, vol. 2, 2014.

[22] M. A. Skoglund, G. Hendeby, and D. Axehill, "Extended Kalman filter modifications based on an optimization view point", in *2015 18th Int. Conf. on Information Fusion (Fusion)*, IEEE, 2015, pp. 1856–1861.

[23] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu, "From intrinsic optimization to iterated extended Kalman filtering on Lie groups", *J. of Mathematical Imaging and Vision*, vol. 55, pp. 284–303, 2016.

[24] M. Brossard, A. Barrau, and S. Bonnabel, "A code for unscented Kalman filtering on manifolds (UKF-M)", in *2020 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 5701–5708.

[25] J. Ćesić, I. Marković, and I. Petrović, "Mixture reduction on matrix Lie groups", *IEEE Signal Processing Letters*, vol. 24, no. 11, pp. 1719–1723, 2017.

[26] J. Farrell, *Aided navigation: GPS with high rate sensors*. McGraw-Hill, Inc., 2008.

[27] K. MacTavish and T. D. Barfoot, "At all costs: A comparison of robust cost functions for camera correspondence outliers", in *2015 12th Conf. on Computer and Robot Vision*, IEEE, 2015, pp. 62–69.

[28] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems", in *2013 IEEE Intell. Vehicles Symposium (IV)*, IEEE, 2013, pp. 473–480.

[29] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems", in *2013 IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS)*, IEEE, 2013, pp. 1280–1286.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF", in *2011 Int. Conf. on Computer Vision*, Ieee, 2011, pp. 2564–2571.

[31] F. Fraundorfer and D. Scaramuzza, "Visual odometry : Part ii: Matching, robustness, optimization, and applications", *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[32] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *Int. J. of Computer Vision*, vol. 60, pp. 91–110, 2004.

[33] O. S. C. Vision. "Feature matching". (2023), [Online]. Available: `https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html` (visited on 06/22/2023).

[34] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation", in *2020 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 4666–4672.