Particle Filtering on Lie Groups : Application to Navigation

Clément Chahbazian

université PARIS-SACLAY

Particle Filtering on Lie Groups

Application to Navigation

Filtrage Particulaire sur groupes de Lie
application à la navigation

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Sciences du traitement du signal et des images.
Graduate School : Sciences de l'ingénierie et des systèmes. Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Traitement de l'information et systèmes (Université Paris-Saclay, ONERA)**, sous la direction de **Christian MUSSO**, Directeur de recherche, le co-encadrement de **Karim DAHIA**, Ingénieur de recherche, et le co-encadrement de **Nicolas MERLINGE**, Ingénieur de recherche.

Thèse soutenue au Plessis-Robinson, le 9 mai 2023, par

Clément CHAHBAZIAN

Composition du jury

Membres du jury avec voix délibérative

Nicolas LE BIHAN Directeur de recherche, CNRS Gipsa-lab	Président
Audrey GIREMUS Professeure des universités, Université de Bordeaux (IMS)	Rapportrice & Examinatrice
Hichem SNOUSSI Professeur des universités, Université de Technologie de Troyes	Rapporteur & Examinateur
Silvère BONNABEL Professeur des universités, Mines ParisTech	Examinateur
François LE GLAND Directeur de recherche, INRIA Rennes	Examinateur

THESE DE DOCTORAT
NNT : 2023UPAST069

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Filtrage Particulaire sur Groupes de Lie, Application à la Navigation
Mots clés : Groupes de Lie, Estimation, Filtre Particulaire, Navigation

Résumé :
L'estimation bayésienne est une discipline importante dans un grand nombre de domaines scientifiques. Elle se base sur le théorème de Bayes qui permet d'associer une observation avec une connaissance a priori sur un évènement ou un paramètre. Cependant, ce théorème ne peut pas être résolu analytiquement en présence de fortes non-linéarités, aussi de nombreuses méthodes ont été développées pour le traiter numériquement, dont les filtres particulaires qui représentent les densités de probabilités avec un nuage de particules. Cette approche permet de traiter des problèmes fortement non-linéaires avec un cadre générique. Cependant, les filtres particulaires présentent des défis, tels que l'étape de ré-échantillonnage, la résolution de problèmes de grande dimension, ainsi que la charge calculatoire. En parallèle, de récentes études portant sur des algorithmes d'estimation dans les groupes de Lie ont montré l'intérêt de ces approches sur de nombreux aspects. En effet, représenter les variables d'estimation sur les groupes de Lie permet d'utiliser les propriétés algébriques et géométriques de ces espaces et amène à une gestion naturelle des incertitudes. Ainsi, les filtres obtenus présentent une amélioration de leur précision et de leur robustesse par rapport aux approches classiques. Cette thèse porte sur le domaine nouveau du filtrage particulaire dans les groupes de Lie. Elle propose un ensemble de filtres particulaires résolvant l'équation de Bayes dans les groupes de Lie, ainsi qu'une borne d'erreur minimale. De nouveaux algorithmes sont développés en particulier pour l'étape de ré-échantillonnage et pour la représentation des particules. Les méthodes proposées sont appliquées à la navigation de systèmes autonomes, qui exigent des algorithmes robustes pour estimer leur état (position, vitesse et attitude) afin d'effectuer leur contrôle et leur guidage. Les algorithmes de navigation utilisent les mesures d'une centrale inertielle. Cependant, les défauts de ce capteur génèrent une dérive temporelle des grandeurs cinématiques estimées. Il est donc nécessaire de les recaler en vol par les mesures de senseurs auxiliaires, via un processus de fusion de données. Les algorithmes proposés dans la thèse ont été testés sur des scénarios de navigation exigeants, et ont montré un gain significatif en précision et en robustesse par rapport aux méthodes classiques.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Particle Filtering on Lie Groups, Application to Navigation
Keywords: Lie Groups, Estimation, Particle Filter, Navigation

Abstract: Bayesian estimation is an important discipline in many scientific and technical domains. It is based on Bayes' theorem, which allows to associate an observation with an a priori knowledge about an event or a parameter. However, this theorem cannot be solved analytically in the case of strong non-linearities. Thus, many methods were developed to address this problem numerically. Among them, particle filters represent probability densities with a cloud of particles. This allows to solve strongly nonlinear problems with a generic approach. However, particle filters present several challenges, such as the resampling step, the resolution of high-dimensional problems, and the computational load. Moreover, studies on estimation algorithms in Lie groups demonstrated the interest of these approaches in many aspects. Indeed, representing the estimation variables on Lie groups allows the use of algebraic and geometric properties of these spaces and leads to a natural handling of uncertainties. Thus, filters on Lie groups show improved accuracy and robustness compared to conventional approaches. This thesis focuses on the new field of particle filtering on Lie groups. It establishes a class of particle filters solving Bayes' theorem on Lie groups by focusing on different aspects of these algorithms, such as the resampling step, the particle representation, and the lower error bound. Furthermore, the proposed methods are applied to the navigation of autonomous systems that need robust algorithms to estimate their state (position, velocity, attitude) in order to perform their control and guidance. An inertial measurement unit (IMU) is usually used to complete the navigation function. However, these sensors drift and need to be frequently updated with aiding sensor measurements, which requires a navigation filter for data fusion. Thus, the algorithms presented in this thesis are tested on challenging navigation scenarios, and demonstrated a significant gain in accuracy and robustness compared to conventional methods.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Accronyms

ARMSE	Average Root Mean Square Error
BCH	Baker-Campbell-Hausdorf formula
CRLB	Cramer-Rao Lower Bound
E-CRLB	Euclidean Cramer-Rao Lower Bound
EKF	Extended Kalman Filter
IEKF	Invariant Extended Kalman Filter
iid	Independent Indentically Distributed
ItEKF	Iterated Extended Kalman Filter
KPKF	Kalman-Particle Kernel Filter
LG	Lie Group
LG-CRLB	Lie Group Cramer-Rao Lower Bound
LPF	Laplace Particle Filter
PF	Particle Filter
RMSE	Root Mean Square Error
UKF	Unscented Kalman Filter

Calculus and Operators

$\det$	Determinant
∂_X	Euclidean derivative
$\dfrac{\partial}{\partial \epsilon}$	Euclidean derivative
$\| . \|_2$	L_2 norm
$\| . \|_M, M \in \mathbb{R}^{d \times d}$	Mahalanobis norm
$\mathrm{d}\epsilon$	Lebesgue measure
$\mathrm{d}X$	Haar measure

$\mathcal{W}^n$	Likelihood cluster weight
θ	Proportional resampling threshold
$\tilde{w}^i$	Particle weight (un-normalized)
$\tilde{W}^n$	Proposal density cluster weight
N	Number of steps in the estimation process
N_p	Number of particles
N_{eff}	Number of efficient particles [47]
N_{step}	Number of steps in the estimation process
N_{th}	Resampling threshold
N_c	Number of clusters in the prior density
w^i	Particle weight (normalized)
W^n	Prior density cluster weight
X^i	Particle

Lie Groups Analysis

$[\cdot, \cdot]$	Algebra hook
ϵ, ξ	Log-Euclidean error
$\exp_{\mathcal{G}}$	Group exponential
$\log_{\mathcal{G}}$	Group logarithm
$\text{Ad}_{\mathcal{G}}$	Group adjoint
$\text{ad}_{\mathcal{G}}$	Algebra adjoint
$\text{SE}(\text{d})$	Special Euclidean group of dimension d
$\text{SE}_{\text{p}}(\text{d})$	Multiple Special Euclidean group of dimension d with p vectors
$\text{SO}(\text{d})$	Special Orthogonal group of dimension d
∂_X	Lie group derivative
$\Phi_{\mathcal{G}}$	Exponential Jacobian

$\varphi_{\mathcal{G}}$	Logarithm Jacobian

Miscellaneous

$I_d \in \mathbb{R}^{d \times d}$	Identity matrix

Navigation

Ω	Rotation rate skew-symmetric matrix
ω	Rotation rate vector
$\Theta = (\psi, \theta, \phi)$	Euler angles: Yaw, Pitch, Roll
ba	Accelerometer bias vector
bg	Gyrometer bias vector
C	Rotation matrix
g	Gravity vector
p	Landmark position vector
v	Velocity vector
v_{DOP}	Doppler velocity
x	Position vector

Probabilities

$\mathbb{E}[.]$	Probabilistic expectation
$\mathbb{V}[.]$	Probabilistic variance
$\mathcal{N}_{\mathcal{G}}(X; \mu, P)$	Lie group normal density of mean μ and covariance P
$\mu \in \mathcal{G}$	Deterministic mean matrix
ν^q	Process noise model
ν^r	Measurement noise model
J	Fisher information matrix
P	Probabilistic covariance matrix
Q	Covariance matrix for the process noise model
R	Covariance matrix for the measurement noise model

Spaces

$\mathbb{R}^d$	Euclidean Space of dimension d
$\mathcal{C}$	Cluster of particles
$\mathcal{G}$	State matrix Lie group
$\mathcal{G}^n$	Augmented Lie group
$\mathcal{H}$	Measurement matrix Lie group
$\mathcal{M}$	Manifold
$\mathfrak{g}$	State Lie algebra
$\mathfrak{h}$	Measurement Lie algebra

Variables

$\hat{\mu} \in \mathcal{G}$	Probabilistic mean
$\mathcal{X}$	Augmented Lie group matrix
$\mu \in \mathcal{G}$	Lie group matrix
$\mu^* \in \mathcal{G}$	State *maximum a posteriori*
$a, b \in \mathbb{R}^d$	Euclidean variables
$u \in \mathbb{R}^n$	Model input vector
$X \in \mathcal{G}$	Lie group state random variable
$Y \in \mathcal{G}$	Lie group measurement random variable

Acknowledgements

I would like to express my deepest gratitude and appreciation to all those who have contributed to the completion of this thesis. Without their support, guidance, and encouragement, this work would not have been possible.

First and foremost, I would like to thank my thesis director, Christian Musso, and my supervisors, Karim Dahia and Nicolas Merlinge, for their invaluable guidance and mentorship throughout the tortuous journey of doctoral studies. Their expertise, patience, insightful feedback and humanity have been instrumental in shaping the direction of my work. I am truly grateful for their constant support and for believing in my abilities in any situation. They have my highest consideration and esteem as scientists and humans.

I extend my heartfelt appreciation to my thesis reviewers, Audrey Giremus and Hichem Snoussi, for their time, effort, and expertise. This also goes for the other members of my thesis committee, Nicolas Le Bihan, Silvère Bonnabel and François Le Gland. Their constructive criticisms, valuable suggestions, and rigorous examination of my work have immensely contributed to its improvement and enlightened my vision for future research works.

I would like to acknowledge ONERA, MBDA France and the Agency for Innovation and Defense, which joint endeavour created a broad and dynamic environment for my doctoral studies. The resources, facilities, and opportunities provided by these institutions have played a significant role in the successful completion of this thesis. I am much obliged to Jacques Blanc-Talon, Arnaud Lepers and Christophe Guilmart for their interest in my work and their insightful feedback throughout this thesis. I am grateful to Pascal Fély, Hervé Verrièle and Jean-Philippe Ovarlez for supporting my work at ONERA and Jean-Michel Allard for his friendly feedback and his expertise on navigation problems. My deepest gratitude also goes to Dominique Guiboug for her truthful guidance and support. Besides, I would like to extend my sincere appreciation to Jean-Christophe Antoni, Dominique Lucas and Lionel Buffard for supporting my doctoral work at MBDA and enabling promising perspectives by developing research tracks within the group.

My gratitude goes to all my colleagues from ONERA and MBDA France who have shared their knowledge, ideas, and expertise, thereby enriching this study. I want especially to thank Bénédicte Winter-Bonnet and Julien Marini who pioneered my thesis at MBDA France and trusted me to take up this doctoral position. I would like to include Kévin Honoré and Aurélien Blanc whose inputs have fostered a valuable industrial background to my work. This goes as well for my others colleagues and friends: Ethan (*aka*[1] Small Lake, *aka* The Pond, *aka* The Honey Badger, *aka* Ethanol), Pierre (*aka* Foreign Body), Daniel (*aka* The Nuclear Dutchman), Antoine (*aka* Boris, *aka* Litte Flat, *aka* Xiaoping, *aka* Mr Ping, *aka* [whateveryouwant]), Baudouin (*aka* Dédé), Clément

[1] *aka* : also known as

(*aka* Kenny), Guillaume LB (*aka* LBG, *aka* The Worst), Maxime (*aka* Maxou), Jean-Baptiste (*aka* 3bis), Guillaume S (*aka* Guigui, *aka* QDB), Emile (*aka* Milou), Antonin (*aka* Toni), Thibault (*aka* Tibo), Enzo, Ugo, Laure-Anne, Raphael, Bastien, Louis, Natacha, Luc, Ralph, François, Jean-Luc, Camille, Julius and many others.

My early research journey started long before this thesis and strongly contributed to its success. To that extent, I am extremely thankful to Hervé Jourdran and Gauthier Dakin for their proactive guidance and mentoring during my first research experience at the French Atomic Commission (CEA). I am also thankful to my former colleagues and friends from this institution, Stéphane, Ewan, Hugo, Rémy, Hoby and Eloise.
Besides, I acknowledge the significant influence of my time at Schlumberger-Doll Research (SDR) in the completion of this thesis. Hence, my gratefulness goes to Sepand Ossia, Stephane Vannuffelen and Arnaud Croux for their trust, guidance and dynamic mentorship. I also learned a lot from Timothy Osedach and Pedro Teixeira who shared their technical excellence and scientific knowledge with passion and humility. I extend my deepest gratitude to Lisa Schmitt, Yaning Zhang, Debra Davis and the amazing staff of SDR for allowing me to initiate my early career in an optimal environment. I am also sincerely grateful to John Leonard for our discussions and for taking the time to share his vision on robotics when I was seeking new research perspectives. Eventually, I am thankful to all my colleagues and fellow interns from SDR and Boston area: Kurran, Brindha, Yohann, Julien, Maud, Charles, Victoire, Myriam, Mona, Manson, Tongyang, Greg, Noyan, Nengxiu, and many others, for enlightening my journey in America with their friendship and brilliant minds.

I appreciate the role of my professors at Supélec and Lycée Gustave Eiffel of Bordeaux. Their mentorship spurred my genuine interest in science and technology and guided me through demanding academic curriculums. Hence, I want to thank Giorgio Valmorbida and Didier Dumur from Supélec Control Department, Sébastien Fayolle, François Motard, Sébastien Villacampa, François Batista, Jean-Philippe Quadri, and my other professors from Lycée Gustave Eiffel in Bordeaux.

I would like to express my deepest appreciation to Manon and our families for their support, love, and understanding throughout this long and challenging journey. Their encouragement, belief in my abilities, and kindness have been a constant source of strength. I express special acknowledgement to my parents who supported me with love and devotion in any of my decision, as well as for my siblings.

Finally, I would like to extend my gratitude to all my friends and family who stood by me before and during this thesis. Their encouragement, motivation, and occasional - yet efficient - distractions have made this journey enjoyable and luminous. This gratitude includes, amongst many others, Sacha, Sylvain, Matthieu, Yann, Guillaume, Olivier, Romain, Adélie, Léon, Blandine, Océane, Paula, Antoine, Othmane, Matt[2], Reenad[2], Cannelle[2], Leslie[2], Tomoya[2], Yan[2], Robert[2] and Tamara[2].

To everyone who has played a part, big or small, in my life I extend my heartfelt thanks. Your contributions have made an impact on my journey, which is reflected on this work.

Clément.

[2]Memories of Mondamin court jeans jackets, Boston and NYC nightlife, New Hampshire hikes, guitar evenings, Bagelsaurus mornings and Porter Square food courts.

1 - Introduction

1.1 . Context

Bayesian estimation is a broad statistical framework that has been widely used in various fields, such as engineering, social sciences and finance, since they often involve parameters or events subject to uncertainties. The Bayesian approach is based on the theory of probability and Bayes theorem, which provides a way to update the belief of an event when new information becomes available: it computes the posterior probability of the event by combining prior belief and the likelihood of an observation. This has several advantages over deterministic methods since it incorporates the knowledge of a model and uncertainty sources into the analysis process, leading to insightful results. Besides, it provides a natural and generic way to handle complex problems, where the parameters and events have strong dependencies.
Despite its advantages, Bayesian estimation raises some challenges that need to be addressed. The computation of the posterior density may involve high-dimensional integrals and strongly nonlinear models, which can be intractable analytically. However, advances in computational methods led to efficient techniques for solving Bayes theorem. Some of the most popular techniques are estimation filters, Monte Carlo Markov Chain algorithms and variational inference. In particular, estimation filters are an active research field since the second half of the 20th century. The most popular approaches like Extended Kalman Filter, Unscented Kalman Filter or Particle Filter led to mature industrial systems. Nonetheless, high-dimensional and strongly nonlinear scenarios are still challenging for these filters, and the implementation of a robust algorithm often involves major engineering challenges. Over the last decade, Lie groups gained increasing interest for Bayesian estimation. These spaces were introduced by Sophus Lie in the late 19th century and later developed by numerous major mathematicians and physicists. Lie groups have the geometric nature of a differential manifold and the algebraic structure of a group, which makes them highly relevant to describe physical systems. Indeed, this specificity enables to process the estimated variables in their natural space, which preserves their symmetries. Recent works demonstrated that this framework leads to substantial improvements on the accuracy and robustness of Bayesian filters. This thesis investigates theoretical aspects of particle filtering on Lie groups by addressing the Bayes theorem from different perspectives. This approach led to several filters and theoretical tools suited to solve a broad range of estimation problems. Besides, the methods developed through this work are applied to navigation, which consists in finding the position of a device with respect to a reference frame. This discipline has been fundamental to transportation and mobile systems for centuries. Modern techniques are based on estimation filters which calculate kinematics variables and systems parameters. Usually, these filters are designed on the Euclidean space $\mathbb{R}^d$ since a vectorial representation of the equations and variables is convenient for its vector-space properties. However, this approach overlooks the non-Euclidean and nonlinear nature of some systems, especially regarding the rotations. Indeed, angular uncertainties are usually treated with a linear approximation, assuming small errors between the estimate and the true state. Although

such approaches are often relevant, the filter performance is undermined when the sensors are strongly nonlinear or when the operating conditions do not provide a good estimate of the state. On the other hand, errors described on Lie groups are accurate on a large domain, which introduces an intrinsic robustness in the filters. Therefore, Lie groups provide a natural and powerful framework to design filters, which often leads to elegant and efficient solutions compared to their Euclidean counterparts.

1.2 . Aim of the Thesis

Recent studies proved the interest of Extended and Unscented Kalman filters on Lie groups for nonlinear estimation. In the wake of these encouraging results, this thesis focuses on the new field of Monte Carlo methods on Lie groups. The main goal is to derive, implement and test innovative algorithms, and put their performance in perspective with usual Monte Carlo algorithms such as Euclidean Particle Filters and existing Lie groups filters, when possible. In addition, the application of these filters focuses on long range navigation which accounts for terrestrial effects such as Coriolis force or variable gravity. More precisely, the thesis focuses on:

- The derivation of particle filters from the Bayes Filter (also called Optimal Filter) on Lie groups. These filters were designed to cope with strongly nonlinear estimation scenarios;
- The impact of the group chosen to design the filter, and the proper representations of the state variables and noises, depending on the application;
- Providing a clear framework to implement these filters.

1.3 . Contributions of the Thesis

This thesis establishes a new framework for Monte Carlo filters on Lie groups which is derived from the Bayes filter. In the light of the issues discussed in the previous section, these new approaches demonstrate their accuracy and robustness in strongly nonlinear and non-Gaussian problems. Besides, cross-comparatives studies between Particle and Kalman filters defined on the Euclidean space and their Lie groups counterparts proves the interest of representing the variables in Lie groups. Additional works on the recursive Cramer-Rao Lower Bound show that the Euclidean formulation can be naturally extended to unimodular matrix Lie groups. This thesis led to several publications detailed hereafter.

International journals:

- Clément Chahbazian, Karim Dahia, Nicolas Merlinge, Bénédicte Winter-Bonnet, Aurélien Blanc, Christian Musso, *Discrete Recursive Posterior Cramer-Rao Lower Bound on Lie Groups*, Automatica 2022 (minor revisions).

- Clément Chahbazian, Nicolas Merlinge, Karim Dahia, Bénédicte Winter-Bonnet, Aurélien Blanc, Christian Musso, *Revisited Extended Kalman Filter on Matrix Lie Groups*, IEEE Transactions on Robotics 2022 (in preparation for submission).

International and national conferences:

- Clément Chahbazian, Nicolas Merlinge, Karim Dahia, Bénédicte Winter-Bonnet, Julien Marini, Christian Musso, *Laplace Particle Filter on Lie Groups Applied to Angles-Only Navigation*, IEEE International Conference on Information Fusion 2021, pages 1-8.
- Clément Chahbazian, Karim Dahia, Nicolas Merlinge, Bénédicte Winter-Bonnet, Kévin Honore, Christian Musso, *Improved Kalman-Particle Kernel Filter on Lie Groups Applied to Angles-Only Navigation*, IEEE International Conference on Robotics and Automation 2021, pages 1689-1694.
- Clément Chahbazian, Nicolas Merlinge, Karim Dahia, Bénédicte Winter-Bonnet, Aurélien Blanc, Christian Musso, *Generalized Laplace Particle Filter on Lie Groups Applied to Ambiguous Doppler Navigation*, IEEE International Conference on Intelligent Robots, Kyoto, Japan, 2022, pp. 2387-2394.
- Christian Musso, Frédéric Dambreville, Clément Chahbazian, *Filtering and Sensors Optimization Applied to Angles-Only Navigation*, IEEE International Conference on Information Fusion 2021, pages 1-8.
- Clément Chahbazian, Karim Dahia, Nicolas Merlinge, Bénédicte Winter-Bonnet, Aurélien Blanc, Christian Musso, *Filtre particulaire sur groupes de Lie*, GRETSI 2022.

1.4 . Organization of the Document

This thesis unfolds in three parts and thirteen chapters, including the introduction chapter. Part I gives an overview of the state-of-the-art in unusual nonlinear estimation, an introduction to Lie groups for Bayesian estimation and the state-of-the-art of filters on Lie groups. Part II details the theoretical contributions of the thesis, that include new filters detailed from Chapter 5 to Chapter 8, a set of close-to-optimal resampling strategies on Lie groups in Chapter 9, and a recursive Cramer-Rao Lower Bound in Chapter 10. Finally, Part III applies the filters developed in this thesis to navigation scenarios. Chapter 11 sets the framework and equations of navigation. Chapter 12 compares the filters in different test cases, and Chapter 13 concludes this thesis.

Part I

Mathematical Preliminaries and State of the Art

2 - Bayesian Estimation

This chapter describes the Bayesian estimation framework which computes the posterior density of the state given a set of observations by combining the likelihood of a measurement with prior knowledge on the state. It also outlines some of the usual filters designed on the Euclidean space.

2.1 . Data Fusion and Bayesian Inference

Data fusion aims to extract relevant information out of uncertain data coming from heterogeneous sources. This broad domain applies to navigation which principle is to merge measurements of different nature to estimate the position of a vehicle.
Bayesian inference is a popular technique to perform data fusion. Its principle is to compute the posterior probability density of a hidden random variable from a prior dynamics model and likelihood function derived from an observation model. Let $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$ be two random vectors such that:

$$\begin{cases} x_{k+1} &= f(x_k, u_{k+1}, \nu_k^q), \\ y_{k+1} &= h(x_{k+1}, \nu_{k+1}^r), \end{cases} \tag{2.1}$$

where u_{k+1} is a vector of inputs and (ν_k^q, ν_{k+1}^r) are two centered noise vectors. The Bayesian framework is based on the following assumptions:

- The sequence $x_{1:k} = \{x_1, \cdots, x_k\}$ follows a Markovian stochastic process where the transition density writes:
$$p(x_{k+1}|x_{1:k}) = p(x_{k+1}|x_k). \tag{2.2}$$
- The observations $y_{1:k}$ are conditionally independent with respect to the state:
$$p(y_{1:k}|x_{1:k}) = \prod_{i=1}^{k} p(y_i|x_i). \tag{2.3}$$

Bayesian inference uses prior knowledge from the state space model and observations to infer the conditional density of the state x_k given the observations $y_{1:k}$.

The Bayes rule writes:

$$p(x_{k+1}|y_{1:k+1}) = \frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})}{p(y_{k+1}|y_{1:k})}, \tag{2.4}$$

where:

- $p(x_{k+1}|y_{1:k+1})$ is the posterior density estimated with the Bayesian inference;
- $p(y_{k+1}|x_{k+1})$ is the likelihood derived from the observation model;

- $p(x_{k+1}|y_{1:k})$ is the prior density, which relates to the knowledge of the propagated density with respect to the model;
- $p(y_{k+1}|y_{1:k})$ is the marginal likelihood, which can be seen as a scale parameter as it does not depend on x_k nor x_{k+1}.

The resolution of the Bayes rule is commonly referred to as the update step in filtering since it fuses the information from the measurement with the prior density. The latter can be obtained by the Chapman-Kolmogorov equation.

The Chapman-Kolmogorov equation writes:

$$p(x_{k+1}|y_{1:k}) = \int_{\mathbb{R}^d} p(x_{k+1}|x_k)p(x_k|y_{1:k})\mathrm{d}x_k, \tag{2.5}$$

where:

- $p(x_k|y_{1:k})$ is the posterior density at time k;
- $p(x_{k+1}|x_k)$ is the transition density which relates to the noisy dynamics model of the state-space system.

Following these considerations, the Optimal Filter (also called Bayes Filter) can be obtained as detailed in Algorithm 1. First, the prior density is propagated with the Chapman-Kolmogorov equation, leading to the prior density, which is updated with the Bayes rule.

Algorithm 1 Optimal Filter

Result: $p(x_k|y_{1:k})$, $k \in [1, N]$

Propagation step: $p(x_{k+1}|y_{1:k}) = \int_{\mathbb{R}^d} p(x_{k+1}|x_k)p(x_k|y_{1:k})\mathrm{d}x_k$

Update step: $p(x_{k+1}|y_{1:k+1}) \propto p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})$

Note that the Optimal Filter is only theoretical and cannot be exactly implemented in practice, except in the case of linear Gaussian systems, which leads to Kalman Filter. Indeed, in the general case, Optimal Filter contains integrals which are difficult to compute when the dimension is high. Hence, Bayesian estimation filters are numerical methods designed to solve the Optimal Filter with assumptions on the system. When the dimension of the state is small, nonlinear filtering equations can be solved by grid methods [19, 2]. The following sections discuss the main categories of filters.

2.2 . Estimators

2.2.1 . Mean and Covariance Matrix

Although Optimal Filter estimates the probability density $p(x_k|y_k)$, most algorithms only focus on its first and second statistical moments, also referred to as the mean and covariance matrix. These parameters are often sufficient to describe the behavior of p when it is sufficiently symmetric and unimodal.

In the sequel, a probability density function is called unimodal when it is concentrated around one predominant peak, involving one global maximum. On the opposite, a probability density function is called multimodal if it has several peaks.

Considering two vectors $x_k \in \mathbb{R}^d$ and $y_k \in \mathbb{R}^m$, the conditional expectation at time k writes:

$$\mathbb{E}\left[x_k|y_{1:k}\right] = \int_{\mathbb{R}^d} x_k p(x_k|y_{1:k}) \mathrm{d}x_k, \tag{2.6}$$

and the conditional covariance matrix:

$$\mathbb{V}\left[x_k|y_{1:k}\right] = \mathbb{E}\left[(x_k - \mathbb{E}\left[x_k|y_{1:k}\right])(x_k - \mathbb{E}\left[x_k|y_{1:k}\right])^T\right]. \tag{2.7}$$

For compact notations, and in order to depict the different densities involved in the Optimal Filter, the conditional expectation at different steps of the filtering process is denoted:

$$\begin{aligned} \hat{x}_{k|k} &= \mathbb{E}\left[x_k|y_{1:k}\right], \\ \hat{x}_{k+1|k} &= \mathbb{E}\left[x_{k+1}|y_{1:k}\right]. \end{aligned} \tag{2.8}$$

Similar notations are applied to the variance. Hence, the prior density gives:

$$\begin{aligned} P_{k|k} &= \mathbb{V}\left[x_k|y_{1:k}\right], \\ P_{k+1|k} &= \mathbb{V}\left[x_{k+1}|y_{1:k}\right]. \end{aligned} \tag{2.9}$$

The conditional mean and variance are often used in filtering as they are easy to obtain with slight approximations, but other estimators exist.

2.2.2 . Maximum a Posteriori

The *Maximum a Posteriori* (MAP) estimates the state vector with the highest probability *a posteriori*. It is defined as:

$$\hat{x}_{k+1} = \arg\max_{x_{k+1}} p(x_{k+1}|y_{1:k+1}). \tag{2.10}$$

Besides, when the system verifies the assumptions of Bayesian inference, the MAP maximizes the product of the likelihood and the prior density:

$$\begin{aligned} \arg\max_{x_{k+1}} p(x_{k+1}|y_{1:k+1}) &= \arg\max_{x_{k+1}} \frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})}{p(y_{k+1}|y_{1:k})}, \\ &= \arg\max_{x_{k+1}} p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k}). \end{aligned} \tag{2.11}$$

Hence, in filtering problems, the MAP is taken as:

$$\hat{x}_{k+1} = \arg\max_{x_{k+1}} p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k}). \tag{2.12}$$

Note that for linear Gaussian systems, the MAP is equal to the conditional expectation.

2.3 . Extended Kalman Filter

2.3.1 . Linear case

Kalman Filter [45] represents a major contribution in control and signal processing. It can be obtained from the Optimal Filter assuming that the system (2.1) is linear with centered Gaussian noises:

$$\begin{cases} x_{k+1} &= F_k x_k + \nu^q_{k+1}, \\ y_{k+1} &= H_{k+1} x_{k+1} + \nu^r_{k+1}, \end{cases} \tag{2.13}$$

where $\nu^q_{k+1} \sim \mathcal{N}(0, Q_{k+1})$ and $\nu^r_{k+1} \sim \mathcal{N}(0, R_{k+1})$.
Under these assumptions, the posterior density is a normal law which mean $\hat{x}_{k+1|k+1}$ and covariance matrix $P_{k+1|k+1}$ are tracked with Kalman Filter described in Algorithm 2. It is derived from Algorithm 1 since integrals can be calculated explicitely.

Algorithm 2 Kalman Filter

Result: $(\hat{x}_{k|k}, P_{k|k})$, $k \in [1, N]$
Propagation step:
$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k}$
$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_{k+1}$
Update step:
$K_{k+1} = P_{k+1|k} H_{k+1}^T \left(H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1} \right)^{-1}$
$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \left(y_{k+1} - H_{k+1} \hat{x}_{k+1|k} \right)$
$P_{k+1|k} = (I - K_{k+1} H_{k+1}) P_{k+1|k}$

Kalman Filter is optimal, which means that the posterior density is exactly obtained from these equations since the posterior density is determined with its two first statistical moments. Besides Kalman Filter stability is obtained under specific assumptions on the model [33].

2.3.2 . Extension to nonlinear systems

Extended Kalman Filter (EKF) follows the principle of Kalman Filter but applies to nonlinear systems. It approximates a solution of Optimal Filter under the assumption that the noises of (2.1) are centered and Gaussian. Therefore, it copes with nonlinear models such that:

$$\begin{cases} x_{k+1} &= f(x_k) + \nu^q_{k+1}, \\ y_{k+1} &= h(x_{k+1}) + \nu^r_{k+1}, \end{cases} \tag{2.14}$$

where $\nu^q_{k+1} \sim \mathcal{N}(0, Q_{k+1})$, $\nu^r_{k+1} \sim \mathcal{N}(0, R_{k+1})$, and (f, h) are two nonlinear functions. Note that additive noises are considered for simplicity, but non-additive noises are also suitable for EKF.
Although it has a similar structure to Kalman Filter, Extended Kalman Filter is based on strong approximations on the nonlinear system which prevent optimality and stability properties. Indeed, the matrices used in the filter are computed with a linearization of the propagation and update model:

$$F_k = \left. \frac{\partial f(x)}{\partial x} \right|_{\hat{x}_{k|k}}, \; H_{k+1} = \left. \frac{\partial h(x)}{\partial x} \right|_{\hat{x}_{k+1|k}}.$$

This linearization point has to be accurate so the filter works properly, especially if the functions are strongly nonlinear. Extended Kalman Filter is similar to Algorithm 2, where the matrices F and H are defined in (2.3.2). Its versatility and light implementation makes a popular choice for industrial applications. Nonetheless, it is still an active filed of research as its basic implementation can be substantially improved, depending on the estimation problem.

Without being exhaustive, a few improved implementations of EKF are:

- The Iterated Extended Kalman Filter (ItEKF) [8, 9] loops on the update step to update the linearization point and get to a close-to-optimal computation of H_{k+1};
- The Adaptive Extended Kalman Filter (AEKF) [43] address the case where the propagation and measurement noises are poorly known. It provides a framework to compute online Q_{k+1} and R_{k+1} for improved accuracy;
- The Ensemble Kalman Filter (EnKF) [37] draws particles from the prior density to compute the prior and posterior density with a sampling approximation;
- The Unscented Kalman filter (UKF) [44] uses a set of deterministic sample points (called sigma points) to estimate the covariance matrix.

2.4 . Monte Carlo Methods

Monte Carlo methods estimate the complete density where other filters focus only on the statistical moments. Hence, they can be adapted to demanding scenarios where no prior assumptions can be made on the shape of the estimated densities. Particle filters were introduced independently by Gordon with Bootstrap Filter [39] and Del Moral with Interacting Particle Filter [22]. Since then, many variants have been developed. Among them we can mention Auxiliary Particle Filter [69], the sequential Monte Carlo methods [36], sequential data assimilation [68], MCMC-based Particle Filter [50], Regularized Particle Filter [63], Kalman-Particle Kernel Filter [31] and Box Particle Filter [55].

2.4.1 . Principle

Particle Filter (PF) hinges on the Monte Carlo approximation which principle is to approximate the probability densities of Optimal Filter with weighted Dirac functions, referred-to as particles.

Formally, the Monte Carlo approximation writes:

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta_{x_k^i}(x_k). \tag{2.15}$$

This approximation is interpreted in the weak sense, meaning that for any measurable bounded function ϕ:

$$\int_{\mathbb{R}^d} \phi(x)p(x|y)\mathrm{d}x \approx \sum_{i=1}^{N_p} w^i \phi(x_i). \tag{2.16}$$

In the sequel, the generic nonlinear and non-Gaussian system is considered:

$$\begin{cases} x_{k+1} &= f(x_k, u_{k+1}, \nu_k^q), \\ y_{k+1} &= h(x_{k+1}, \nu_{k+1}^r), \end{cases} \tag{2.17}$$

with the assumptions of Bayesian inferenec.
Particle Filter uses the Monte Carlo approximation to solve the Optimal Filter. For the propagation step, the system evolves according to the dynamics of (2.17). The Chapman-Kolmogorov equation gives:

$$\begin{aligned} p(x_{k+1}|y_{1:k}) &\approx \int_{\mathbb{R}^d} p(x_{k+1}|x_k) \sum_{i=1}^{N_p} w_k^i \delta_{x_k^i}(x_k)\mathrm{d}x_k, \\ &\approx \sum_{i=1}^{N_p} w_k^i p(x_{k+1}|x_k^i), \\ &\approx \sum_{i=1}^{N_p} w_k^i \delta_{x_{k+1}^i}(x_{k+1}). \end{aligned}$$

where the propagated particles are $x_{k+1}^i = f(x_k^i, u_{k+1}, \nu_k^{q,i})$.

The propagation step is given by:

$$p(x_{k+1}|y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta_{x_{k+1}^i}(x_{k+1}) \tag{2.18}$$

Then, the estimated density is updated with the Bayes rule, according to the measurement model of (2.17):

$$\begin{aligned} p(x_{k+1}|y_{1:k+1}) &= \frac{p(y_{k+1}|x_{k+1})}{p(y_{k+1}|y_{1:k})} \sum_{i=1}^{N_p} w_k^i \delta_{f(x_k^i)}(x_{k+1}), \\ &= \sum_{i=1}^{N_p} w_k^i \frac{p(y_{k+1}|x_{k+1}^i)}{p(y_{k+1}|y_{1:k})} \delta_{f(x_k^i)}(x_{k+1}), \\ &= \sum_{i=1}^{N_p} w_{k+1}^i \delta_{f(x_k^i)}(x_{k+1}), \end{aligned}$$

where w_{k+1}^i are the updated weights:

$$w_{k+1}^i \propto w_k^i p(y_{k+1}|x_{k+1}^i). \tag{2.19}$$

Hence, in practice, the updated weights are factorized with the likelihood and normalized with the sum of the total weights:

$$\tilde{w}^i_{k+1} = w^i_k p(y_{k+1}|x^i_{k+1}), \tag{2.20}$$

$$w^i_{k+1} = \frac{\tilde{w}^i_{k+1}}{\sum_{i=1}^{Np} \tilde{w}^i_{k+1}}, \tag{2.21}$$

where $\tilde{w}$ denotes the non-normalized weights.

At any step, the mean and variance of the estimated density is approximated by the mean and variance of the sample of particles:

$$\hat{x}_{k|k} = \sum_{i=1}^{Np} w^i_k x^i_k. \tag{2.22}$$

$$P_{k|k} = \sum_{i=1}^{Np} w^i_k (x^i_k - \hat{x}_{k|k})(x^i_k - \hat{x}_{k|k})^T. \tag{2.23}$$

After a few successive updates, most particle weights tend to zero while a few tend to unity [39]. This phenomenon is called the degeneracy of Particle Filter, which is due to the augmentation of the variance of the normalized weights [47]. The degeneracy of the filter must be avoided: the more weights tend to zero, the less particles represent the estimated density, which leads to substantial inconsistencies and eventually the divergence of the filter. The number of efficient particles N_{eff} , introduced by Kong [47], monitors the degeneracy of the filter as it is linked to the variance of the unnormalized weights:

$$N_{eff} = \frac{N_p}{1 + \mathrm{Var}(\tilde{w}^i_k)}. \tag{2.24}$$

In practice, the efficient number of particles can be approximated with the weights of the normalized particles:

$$N_{eff} \approx \frac{1}{\sum_{i=1}^{N_p} w^i_k}. \tag{2.25}$$

Therefore, when the weights variance is high (i.e. when the filter degenerates), N_{eff} tends to unity. And when the variance is low (i.e. when the filter is consistent), it is close to N_p. In other words, the number of efficient particles depicts the number of particles which efficiently contribute to the filter.
To prevent degeneracy, a resampling step is triggered when it goes below a given threshold N_{th}. In general, common resampling strategies [49] duplicate the most probable particles (i.e. the heavy weights), and delete the least probable ones (i.e. the lower weights). Particle Filter is described in Algorithm 3 in its compact version.

Algorithm 3 Particle Filter

Result: $(\hat{x}_{k|k}, P_{k|k})$, $k \in [1, N]$
Propagation step: $x^i_{k+1} = f(x^i_k, u_{k+1}, \nu^q_k)$
Update step: $w^i_{k+1} \propto w^i_k p(y_{k+1}|x^i_{k+1})$
Resampling step: If $N_{eff} < N_{th}$ draw x^i_{k+1} using a resampling method and set $w^i_{k+1} = \frac{1}{N_p}$
Compute the moments: $\hat{x}_{k+1|k+1}$ and $P_{k+1|k+1}$

Theoretical analyses of the Monte Carlo errors of Particle Filter approximations can be found in [32, 49] where central limit theorems are provided.

2.4.2 . Sequential Importance Sampling

Another popular method for particle filters is the Sequential Importance Sampling (SIS) [35] where the particles are drawn according to a proposal density denoted $\tilde{q}$. This approach is especially useful to lower the variance of the Monte Carlo estimation or when it is difficult to sample the particles according to p. Therefore, if $\tilde{q}$ has to verify the following assumptions:

- The support of $\tilde{q}(x)$ must include the support of $p(x)$;
- The integral $\int \frac{p(x)}{\tilde{q}(x)} \mathrm{d}x$ is finite.

It is possible to build an estimator of $p(x)$ by sampling according to $\tilde{q}(x)$ since:

$$\mathbb{E}_p\left[\phi(X)\right] = \int \phi(x)p(x)\mathrm{d}x = \int \phi(x)\frac{p(x)}{\tilde{q}(x)}\tilde{q}(x)\mathrm{d}x = \mathbb{E}_{\tilde{q}}\left[\phi(X)\frac{p(X)}{q(X)}\right], \tag{2.26}$$

where ϕ is an integrable function.

Then, considering the Monte Carlo approximation from (2.15), the posterior density can be approximated from a sample generated with $\tilde{q}$:

$$p(x_{k+1}|y_{1:k+1}) \approx \sum_{i=1}^{N_p} w^i_{k+1}\delta_{x^i_{k+1}}(x_{k+1}), \tag{2.27}$$

where $w^i_{k+1} = \tilde{w}^i_{k+1}/\sum_{i=1}^{N_p}\tilde{w}^i_{k+1}$ and:

$$\tilde{w}^i_{k+1} = \frac{p(y_{k+1}|x^i_{k+1})p(x^i_{k+1}|y_{1:k})}{\tilde{q}(x^i_{k+1})}. \tag{2.28}$$

Note that in the literature, the proposal density $\tilde{q}$ is also called importance function.
The choice of $\tilde{q}$ has an impact on the Monte Carlo approximation. A good indicator of the quality of the proposal density is the variance of the unnormalized weights [64]:

$$\mathbb{V}\left[\tilde{q}\right] = \frac{1}{N_p}\left(\int \frac{p(x_k|y_k)^2}{\tilde{q}(x_k)}\mathrm{d}x_k - 1\right). \tag{2.29}$$

One can notice that the lowest variance is obtained for $\tilde{q}_{opt}(x_{k+1}) = p(x_{k+1}|y_{1:k+1})$ since the variance (2.29) would equal zero. Hence the optimal proposal density is the function to be estimated.

2.5 . Laplace Particle Filter

Laplace Particle Filter (LPF) is introduced in [61, 70], and uses a close-to-optimal resampling step based on the Laplace approximation for integrals. The resampling step is critical to particle filters as it occurs at a point where the particle cloud is about to degenerate. According to the previous section, choosing a good proposal density in the Monte Carlo approximation has a substantial impact on the estimation process. Hence, the idea of LPF is to compute a proposal density whose statistical moments are close to the ones of the posterior density.

2.5.1 . Laplace approximation

The Laplace approximation introduced in [79] and developed in [62, 61] for Particle Filter.
Let $\{h_\lambda\}_{\lambda>0}$ be a family of functions in $\mathcal{C}^4(\mathbb{R}^d)$ and consider the integral:

$$\mathcal{I}_\lambda = \int_{\mathbb{R}^d} \mathrm{e}^{-\lambda h_\lambda(x)} \mathrm{d}x. \tag{2.30}$$

Assuming that $\exists \lambda_0$ such that $\forall \lambda > \lambda_0$:

- $\mathcal{I}_\lambda$ converges:

$$\int_{\mathbb{R}^d} \mathrm{e}^{-\lambda h_\lambda(x)} \mathrm{d}x < \infty, \tag{2.31}$$

- h_λ admits a global minimum denoted $\hat{x}_\lambda$;
- the four first derivatives of h_λ are bounded with respect to λ.

Then, when $\lambda \to \infty$ [79]:

$$\mathcal{I}_\lambda = (2\pi)^{d/2} \mathrm{e}^{-\lambda h_\lambda(\hat{x}_\lambda)} \det \left[\lambda \frac{\partial^2 h_\lambda(\hat{x}_\lambda)}{\partial x^2} \right]^{-1/2} (1 + O(\lambda^{-1})). \tag{2.32}$$

The principle of this approximation is that when λ is large enough, the contribution of the integrated function is mostly at the vicinity of $\hat{x}_\lambda$, leading to the obtained formula with a second order approximation. Besides, this formula is even more accurate when the integrated function has a fast decrease.

2.5.2 . Estimation of the posterior moments

The idea developed in the sequel is to use the Laplace approximation to compute the first and second statistical moments of the posterior density (i.e. its mean and covariance matrix). This statistical moments will be used to parametrize the importance density which will be close to the posterior. The moment-generating function is given by:

$$M(a) = \mathbb{E}\left[\mathrm{e}^{a^T X} | y \right], \; a \in \mathbb{R}, \; X \sim p(x|y), \tag{2.33}$$

and the conditional expectation and the variance write:

$$\mathbb{E}[X|y] = M'(0), \tag{2.34}$$
$$\mathbb{V}[X|y] = M''(0) + M'(0)^T M'(0). \tag{2.35}$$

From this expression, [70] shows that applying the Laplace approximation (2.32) to the moment-generating function leads to accurate approximations of $\mathbb{E}[x|y]$ and $\mathbb{V}[x|y]$ with analytic expressions:

$$\mathbb{E}[X|y] \approx x^*, \tag{2.36}$$
$$\mathbb{V}[X|y] \approx J(x^*)^{-1}, \tag{2.37}$$

where x^* is the MAP, and $J(x^*)$ is the posterior Fisher information matrix computed at x^*.

Note that high-order corrective terms can be applied to x^* and $J(x^*)$ as described in [70]. In practice, very high accuracy is obtained with the first-order approximation.

2.5.3 . Application to Particle Filtering

As mentioned in this Section 2.4.2, the optimal proposal density $\tilde{q}_{opt}$ is the posterior density. Although it is impossible to obtain directly the posterior density, Laplace approximation enables to calculate a proposal density whose mean and covariance matrix are close to the posterior moments.
Thus, the resampled particles are drawn according to the proposal density $\tilde{q}$:

$$x^i \sim \tilde{q}\left(x; x^*, J(x^*)^{-1}\right), \tag{2.38}$$

where $\tilde{q}$ is usually taken as a normal law. In theory, any probability law could be chosen, provided that it copes with the assumptions of sequential importance sampling. The normal law is a popular choice as it is simple to obtain and often relevant to describe a process. Higher robustness can be obtained with Student laws as they have heavy tails and spread the particles in a broader domain. Finally, more complex densities (e.g. non-symmetric, mixture...) can be tailored to the nature of the problem for improved results.
After resampling the particles, the (non-normalized) weights are computed according to:

$$\tilde{w}^i_{k+1} = \frac{p(y_{k+1}|x^i_{k+1})p(x^i_{k+1})}{\tilde{q}(x^i_{k+1})}. \tag{2.39}$$

Finally, the Laplace Particle Filter algorithm is described in Algorithm 4.

Algorithm 4 Laplace Particle Filter

Result: $(\hat{x}_{k|k}, P_{k|k})$, $k \in [1, N]$
Propagation step: $x^i_{k+1} = f(x^i_k, u_{k+1}, \nu^q_k)$
Update step: $w^i_{k+1} \propto w^i_k p(y_{k+1}|x^i_{k+1})$
if $N_{eff} < N_{th}$ **then**
  MAP: $x^*_{k+1} = \arg\max_{x_{k+1}} p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})$
  Fisher information matrix: $J(x^*_{k+1}) = -\left.\dfrac{\partial^2 \log(p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k}))}{\partial x^2}\right|_{x_{k+1}=x^*_{k+1}}$
  Draw: $x^i_{k+1} \sim \tilde{q}\left(x_{k+1}; x^*_{k+1}, J(x^*_{k+1})^{-1}\right)$
  Set: $\tilde{w}^i_{k+1} = \dfrac{p(y_{k+1}|x^i_{k+1})p(x^i_{k+1})}{\tilde{q}(x^i_{k+1})}$
end if
Normalize: $w^i_{k+1} = \dfrac{\tilde{w}^i_k}{\sum_{i=1}^{N_p} \tilde{w}^i_{k+1}}$

Another approach of LPF would be to compute a proposal density at each update step. However, this approach can be computationally intensive, and applying the Laplace method at each resampling is sufficient to achieve good accuracy. Besides, in the framework of non-linear filtering, the measurement model often involves only a part of the state, which allows a simplified computation of the MAP [70, 58].

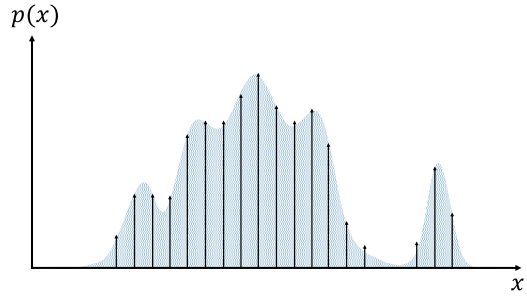
2.6 . Kalman-Particle Kernel Filter

2.6.1 . Principle and Hypotheses

Kalman-Particle Kernel Filter was introduced in [30, 31] and can be seen as an hybrid between Extended Kalman Filter and Particle Filter. The principle of this filter is to represent the particles with a mixture of weighted Gaussians instead of Dirac functions.
The particularity of KPKF is that the norms of the covariance matrices of the Kalman filter are small enough to justify the linearization of the equations.

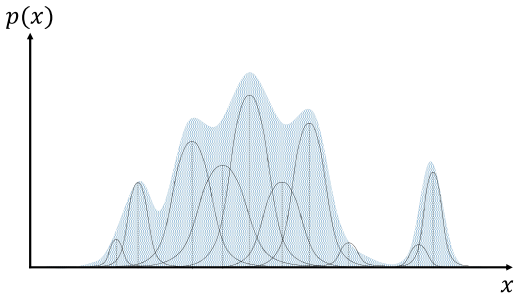
Particle Filter

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta_{x_k^i}(x_k).$$



Kalman-Particle Kernel Filter

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \phi(x_k; x_k^i, P_k^i).$$



This representation of the probability density leads to several improvements:

- The approximated density has a better coverage of the state-space;
- On a local scale, each particle can be processed with an Extended Kalman Filter;
- On a global scale, the cloud of particles can be treated with a Particle Filter process.

Therefore, without loosing the genericity of Particle Filter, the additional local Extended Kalman Filter guide the particles towards the areas of highest probability, preventing early divergence of the filter.

2.6.2 . Algorithm

Initialization

KPKF is initialized using the known density $p(x_0)$ of mean x_0 and covariance matrix P_0. The aim of this step is to draw a Gaussian mixture such that:

$$p(x_0) \approx \sum_{i=1}^{N_p} w_0^i \phi(x_0; x_0^i, h^2 P_0), \tag{2.40}$$

where h is the bandwidth factor [76] and the weights are $w_0^i = 1/N_p$. Assuming that $p(x_0)$ is a Gaussian density function, the means x_0^i have to be drawn such that:

$$x_0^i \sim \mathcal{N}(x_0, (1+h^2)^{-1} P_0), \tag{2.41}$$

which ensures that the covariance matrix of the mixture equals the covariance matrix of $p(x_0)$.

Update Step

The update, involves a Kalman and a Particle Filter process [30]:

$$p(x_{k+1}|y_{1:k+1}) = \sum_{i=1}^{N_p} w_{k+1}^i \phi(x_{k+1}, x_{k+1|k+1}^i, P_{k+1|k+1}^i), \tag{2.42}$$

where the updated particles are such that:

$$\begin{cases} x_{k+1|k+1}^i & = x_{k+1|k}^i + K_{k+1}^i (y_{k+1} - h(x_{k+1|k}^i)), \\ P_{k+1|k+1}^i & = P_{k+1|k}^i - K_{k+1}^i H_{k+1}^i P_{k+1|k}^i, \\ K_{k+1}^i & = P_{k+1|k}^i H_{k+1}^{i^T} \left(H_{k+1}^i P_{k+1|k}^i H_{k+1}^{i^T} + R_{k+1} \right)^{-1}, \end{cases} \tag{2.43}$$

and H_{k+1}^i is the local Jacobian:

$$H_{k+1}^i = \left. \frac{\partial h(x_{k+1})}{\partial x_{k+1}} \right|_{x_{k+1|k}^i}. \tag{2.44}$$

Finally, the weights are updated according the sample importance sampling approach from Section 2.4.2, where the proposal density is taken as the prior density:

$$w_{k+1}^i = w_k^i p(y_{k+1}|x_{k+1|k+1}^i). \tag{2.45}$$

Propagation Step

The particles are propagated according to the Chapman-Kolmogorov equation, and [30] shows that the propagated density writes:

$$p(x_{k+1}|y_{1:k}) = \sum_{i=1}^{N_p} w_k^i \phi(x_{k+1}; x_{k+1}^i, P_{k+1|k}^i), \tag{2.46}$$

and the parameters are propagated with local Extended Kalman Filter equations :

$$\begin{cases} x_{k+1|k}^i & = f(x_{k|k}^i, u_{k+1}), \\ P_{k+1|k}^i & = F_k^i P_{k|k}^i F_k^{i^T} + Q_{k+1}, \end{cases} \tag{2.47}$$

where the local Jacobian F_k^i is such that:

$$F_k^i = \left. \frac{\partial f(x_k, u_{k+1})}{\partial x_k} \right|_{x_{k|k}^i}. \tag{2.48}$$

Note that with KPKF, the contribution of the process noise appears only in the propagation of the covariance matrix and the particle means have a deterministic propagation. At this stage, after several consecutive propagations, the norms of the particle covariance matrices $P^i_{k+1|k}$ are no longer guaranteed to be small, which undermines the linearization of the measurement model. It is therefore important to resample the particles in order to preserve the structure of the Gaussian mixture with small covariance matrices norms.

Resampling Step

KPKF has two distinct resampling steps. A partial resampling method, introduced in [30, 67, 31], is used to ensure that the covariance matrix of the particles remain small. This partial resampling do not modifiy the particles weight.
When the variance of the particles weights is high, the particles degenerate according to the criterion of Kong [47], and a new set of particles is drawn.
Note that in the case of a Laplace resampling with a Gaussian proposal density, the particles covariance matrices are taken as $P^i_{k+1|k+1} = h^2 J(x^*)^{-1}$. Besides, the particle means are drawn according to a Gaussian of mean x^* and covariance matrix $(1+h^2)^{-1}J(x^*)^{-1}$, so the resampled mixture has a covariance matrix close to $J(x^*)^{-1}$.
It is demonstrated in [30] that the optimal value of h is the Silverman formula [76]:

$$h = \left[\frac{4}{(2+d)N_p}\right]^{\frac{1}{d+4}}, \tag{2.49}$$

where d is the dimension of the state space and N_p the number of particles.

3 - Introduction to Lie Groups

Lie groups represent a broad mathematical field which have found numerous applications to fundamental physics and computer science. This chapter is a pragmatic introduction to the theory of Lie groups for Bayesian estimation, with a specific focus on unimodular matrix Lie groups since their properties make them suitable to address filtering problems.

3.1 . General Definitions

A Lie group $\mathcal{G}$ is group which composition law denoted $\circ$ is differentiable. The algebraic properties of the group law $\circ$ verify the following axioms:

$$\begin{aligned} \text{Stability:} \quad & \forall g_1, g_2 \in \mathcal{G} : g_1 \circ g_2 \in \mathcal{G}; \\ \text{Neutral element:} \quad & \exists I \in \mathcal{G} \text{ such that } \forall g \in \mathcal{G} : g \circ I = I \circ g = g; \\ \text{Inverse:} \quad & \forall g \in \mathcal{G}, \exists g^{-1} : g \circ g^{-1} = g^{-1} \circ g = I; \\ \text{Associativity:} \quad & \forall g_1, g_2, g_3 \in \mathcal{G}, g_1 \circ (g_2 \circ g_3) = (g_1 \circ g_2) \circ g_3; \end{aligned} \tag{3.1}$$

The differentiability of the group law implies that $\mathcal{G}$ is a differential manifold, and every point admits a tangent space. In Lie groups theory, the tangent space at the identity plays a central role, and is specifically referred to as the Lie algebra denoted $\mathfrak{g}$. The Lie algebra $\mathfrak{g}$ is a vector space of dimension d. It admits an isomorphism with $\mathbb{R}^d$ which is usually defined with the hat and the vee maps [42]:

$$(hat)\,[\cdot]^\wedge : \mathbb{R}^d \to \mathfrak{g}, \text{and } (vee)\,[\cdot]^\vee : \mathfrak{g} \to \mathbb{R}^d. \tag{3.2}$$

Besides, Lie groups curved geometry can be encoded into the algebra through two mappings defined as the group exponential $\exp_{\mathcal{G}} : \mathfrak{g} \to \mathcal{G}$ and logarithm $\log_{\mathcal{G}} : \mathcal{G} \to \mathfrak{g}$ illustrated in Figure 3.1. The combination of the algebra isomorphisms with $\exp_{\mathcal{G}}$ and $\log_{\mathcal{G}}$ are denoted:

$$\exp_{\mathcal{G}}^{\wedge}(\cdot) \triangleq \exp_{\mathcal{G}}([\cdot]^\wedge) \; ; \; \log_{\mathcal{G}}^{\vee}(\cdot) \triangleq \log_{\mathcal{G}}([\cdot]^\vee). \tag{3.3}$$

The geometric and algebraic nature of Lie groups enables a quite simple and powerful framework. The group law gives simple and exact operations between elements which do not belong to a vector space. Then, the exponential and logarithm maps create a link between the manifold and Lie algebra, which is a vector space and can be mapped to the Euclidean space with an isomorphism. Hence, the exact operations happening in the Lie group can be projected to the Euclidean space where known theories apply (e.g. Kalman Filter).
The main advantage of using Lie groups compared to Riemannian manifolds is that there is only one tangent space to use: the Lie algebra. Indeed, in the case of Riemannian manifold, every tangent space have a different metric, which has to be calculated. In filtering, the transition between the manifold and the Euclidean space has to be performed at many different points, and the Lie group properties simplify greatly the approach as the exponential and logarithm apply to all the points.

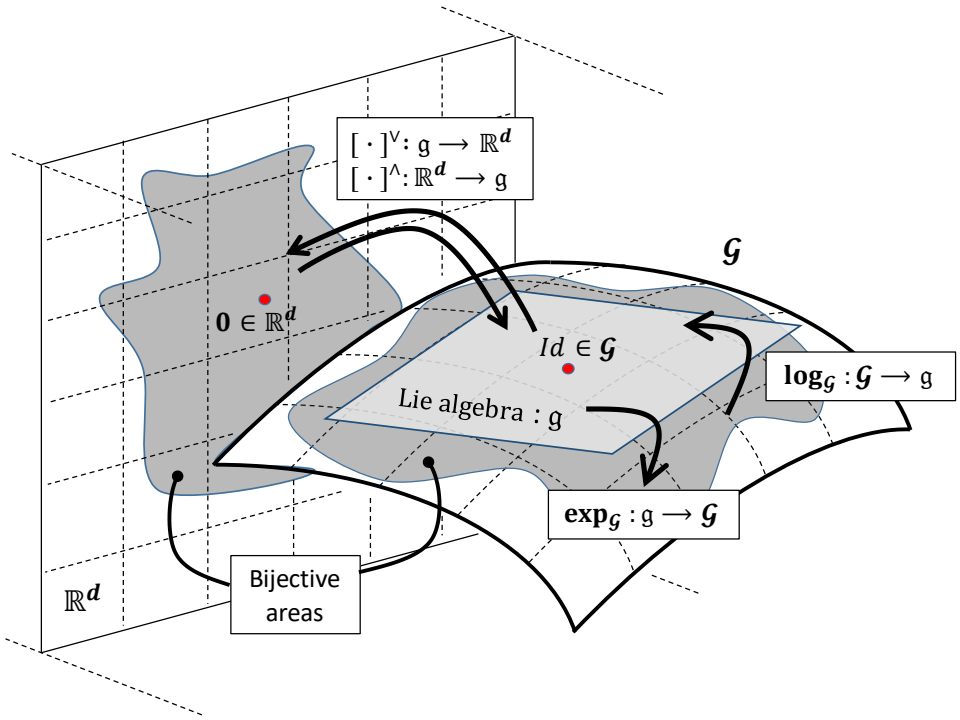


Figure 3.1: Illustration of the Lie group structure. The group exponential $\exp_{\mathcal{G}}^{\wedge}$ and logarithm $\log_{\mathcal{G}}^{\vee}$ define a bijection of $\mathcal{G}$ into $\mathbb{R}^d$, and the algebra $\mathfrak{g}$ is the tangent space at I_d.

3.2 . Matrix Lie Groups

This thesis focuses on matrix Lie groups for which $\mathcal{G}$ is a group of invertible square matrices, and the group operator $\circ$ is the usual matrix product. These Lie groups are particularly useful in nonlinear estimation as they are suited to represent the usual variables encountered in physical systems.
The exponential and logarithm are bijective at the vicinity of I_d (otherwise the exponential is surjective only) and their expressions reduce to matrix power series [42]:

$$\forall a \in \mathbb{R}^d : \exp_{\mathcal{G}}^{\wedge}(a) = \sum_{n=0}^{\infty} \frac{([a]^{\wedge})^n}{n!}, \tag{3.4}$$

$$\forall X \in \mathcal{G} : \log_{\mathcal{G}}^{\vee}(X) = \left[\sum_{n=1}^{\infty} (-1)^{n+1} \frac{(X - I_d)^n}{n} \right]^{\vee}. \tag{3.5}$$

Three groups are encountered in this thesis. The first one is the Special Orthogonal group of dimension $d \in \mathbb{N}$ and defined by:

$$\mathrm{SO}(d) = \left\{ C \in \mathbb{R}^{d \times d} | CC^T = I_d, \det[C] = 1 \right\}. \tag{3.6}$$

It is commonly referred to as the rotation matrices group. The algebra of $\mathrm{SO}(d)$ is the group of skew-symmetric matrices of dimension d :

$$\mathfrak{so}(d) = \left\{ M = [m_{ij}]_{i,j \in [1,d]} \, | m_{ij} = -m_{ji} \right\}. \tag{3.7}$$

The second one is the Special Euclidean group $\mathrm{SE}(d)$. It involves a rotation matrix $C \in \mathrm{SO}(d)$ and a vector $v \in \mathbb{R}^d$:

$$\mathrm{SE}(d) = \left\{ \begin{bmatrix} C & v \\ 0_{1,d} & 1 \end{bmatrix}, \; C \in \mathrm{SO}(d), \; v \in \mathbb{R}^d \right\}, \tag{3.8}$$

and its algebra is:

$$\mathfrak{se}(d) = \left\{ \begin{bmatrix} M & u \\ 0_{1,d} & 0 \end{bmatrix}, \; M \in \mathfrak{so}(d), \; u \in \mathbb{R}^d \right\}. \tag{3.9}$$

The Special Euclidean group can be used to represent homogeneous transformations with pose matrices (rotation and translation) which are ubiquitous in robotics and computer vision [3].

The last group used in this thesis is $\mathrm{SE}_p(d)$ formally introduced in [4, 6, 10]. This structure is a semi-direct product of $\mathrm{SO}(d)$ and $\mathbb{R}^p$. It involves a rotation matrix $C \in \mathrm{SO}(d)$ and $p \in \mathbb{N}$ vectors of $\mathbb{R}^d$:

$$\mathrm{SE}_p(d) = \left\{ \begin{bmatrix} C & v_1 \cdots v_p \\ 0_{p,d} & I_p \end{bmatrix}, \; C \in \mathrm{SO}(d), \; v_1, \cdots, v_p \in \mathbb{R}^d \right\}, \tag{3.10}$$

and its algebra is:

$$\mathfrak{se}_p(d) = \left\{ \begin{bmatrix} M & u_1 \cdots u_p \\ 0_{p,d} & 0_p \end{bmatrix}, \; M \in \mathfrak{so}(d), \; u_1, \cdots, u_p \in \mathbb{R}^d \right\}. \tag{3.11}$$

This group is popular since it enables to represent a pose and additional state variables such as the velocity in the group matix, leading to performance improvements described in the next chapters.

3.3 . Geometric and Algebraic Properties

This section focuses on the main tools for state estimation on matrix Lie groups.

3.3.1 . Group and log-Euclidean error

First, the group error is a fundamental concept. The group law is a natural approach to enable a meaningful comparison which stays in the group. Let $X_1, X_2 \in \mathcal{G}$ be two matrices. The group error between X_1 and X_2 can be defined in two different ways, as the group law generally does not commute:

$$\textbf{Left error: } \eta = X_1^{-1} X_2, \tag{3.12}$$
$$\textbf{Right error: } \eta = X_2 X_1^{-1}. \tag{3.13}$$

Hence, this definition ensures that the error stays in $\mathcal{G}$ given the stability property of the group law. The concept of group error can be extended to the log-Euclidean error:

$$\textbf{Left case: } \epsilon \triangleq \log_{\mathcal{G}}^{\vee}\left(X_1^{-1} X_2\right), \tag{3.14}$$
$$\textbf{Right case: } \epsilon \triangleq \log_{\mathcal{G}}^{\vee}\left(X_2 X_1^{-1}\right). \tag{3.15}$$

Hence, the log-Euclidean error fully encodes the group error in a vector defined in the Euclidean space. This formulation will be often used in the algorithms designed in this thesis.

3.3.2 . Adjoints

General commutativity is not possible on matrix Lie groups. However, it is a useful property in calculus, and is naturally used in vector spaces. To that extent, the group adjoint $\mathrm{Ad}_{\mathcal{G}}$ is a tool which was introduced to behave like a pseudo-commutator:

$$\forall X \in \mathcal{G}, \epsilon \in \mathbb{R}^d : X \exp_{\mathcal{G}}^{\wedge}(\epsilon) = \exp_{\mathcal{G}}^{\wedge}(\mathrm{Ad}_{\mathcal{G}}(X)\epsilon) X, \tag{3.16}$$

and its formal definition is [27]:

$$\forall X \in \mathcal{G}, a \in \mathbb{R}^d : \mathrm{Ad}_{\mathcal{G}}(X) a \triangleq \left[X [a]^{\wedge} X^{-1}\right]^{\vee}. \tag{3.17}$$

The group adjoint enables to define unimodular Lie groups whose properties will be used in the following sections.

The Lie algebra of a matrix Lie group also admits an adjoint operator denoted $\mathrm{ad}_{\mathcal{G}}$ which is defined by the Lie bracket:

$$\forall (a,b) \in \mathfrak{g} : \mathrm{ad}_{\mathcal{G}}(a)b \triangleq \left[[a]^{\wedge}, [b]^{\wedge}\right]^{\vee}, \tag{3.18}$$

where:

$$\left[[a]^{\wedge}, [b]^{\wedge}\right] = [a]^{\wedge}[b]^{\wedge} - [b]^{\wedge}[a]^{\wedge}. \tag{3.19}$$

Note that $\mathrm{Ad}_{\mathcal{G}}$ and $\mathrm{ad}_{\mathcal{G}}$ are related with the following equation [27]:

$$\forall a \in \mathbb{R}^d, \mathrm{ad}_{\mathcal{G}}(a) = \mathrm{logm}(\mathrm{Ad}_{\mathcal{G}}(\exp_{\mathcal{G}}^{\wedge}(a))), \tag{3.20}$$

where logm denotes the usual matrix logarithm.

3.3.3 . Unimodularity

At this point of the chapter, all the previous definitions apply to general matrix Lie groups. However, a more restrictive framework needs to be set to introduce the different calculus tools at the base of the filtering algorithms developed in this thesis. The unimodularity of a Lie group is a momentous property which has direct repercussions of most theoretical concepts introduced in the sequel. Indeed, Lie group theory covers a broad range of abstract mathematical aspects which go far beyond the scope of this thesis, and nonlinear estimation in general.
Unimodular matrix Lie groups represent a specific class of Lie groups which are suited to most problems encountered in nonlinear estimation. They ensure a simple framework which naturally extends usual calculus tools to matrix Lie groups, as discussed in [27]. A unimodular Lie group is defined by a unitary determinant of the group adjoint:

$$\forall X \in \mathcal{G} : \det\left[\mathrm{Ad}_{\mathcal{G}}(X)\right] = 1. \tag{3.21}$$

This class of groups shows interesting topological properties. First, a unimodular Lie group is compact [27]. Compact groups all carry a bi-invariant Haar measure, meaning that it is unchanged by a left or right multiplication. This enables a simplified definition of integrals and invariance properties which are alike the one encountered for the Euclidean space. Besides, the Haar measure can be easily normalized to define a probability density function on the group, which is highly relevant when working with stochastic systems.

In this thesis every reference to Lie group implies unimodular matrix Lie group, for which the properties described in the sequel apply. Hence they enable a convenient algebraic and geometric framework with a natural extension of most customary analysis tools.

3.3.4 . Group action

Operations between elements from the group and elements from outside the group can be useful (e.g. a matrix-vector product). Let $\mathcal{U}$ be a set of elements of the same nature (i.e. a set of vectors). A group action denoted $*$ is a left or right homomorphism:

$$\text{Left} : \mathcal{G} \times \mathcal{U} \to \mathcal{U}, \tag{3.22}$$

$$\text{Right} : \mathcal{U} \times \mathcal{G} \to \mathcal{U}, \tag{3.23}$$

such that $\forall u \in \mathcal{U}$, $X_1, X_2 \in \mathcal{G}$, the operator $*$ verifies the following properties:

$$\textbf{Identity invariance: } I_d * u = u * I_d = u \tag{3.24}$$
$$\textbf{Left associativity: } X_1 * (X_2 * u) = (X_1 X_2) * u \tag{3.25}$$
$$\textbf{Right associativity: } (u * X_2) * X_1 = u * (X_2 X_1). \tag{3.26}$$

The group action is specially useful to derive IEKF described in Chapter 4.

3.3.5 . Group Jacobian

Lie groups are often represented as smooth curved spaces, such as in Figure 3.1. This curvature is linked to the non-commutativity of the group law, which involves a distortion when a sum of vectors from the algebra are projected to the group with the group exponential. This curvature often appears in calculus, and is characterized by the Lie group Jacobian $\Phi_{\mathcal{G}}(\epsilon)$ defined as the derivative of the exponential with respect to the log-Euclidean error:

$$\forall \epsilon \in \mathbb{R}^d, \Phi_{\mathcal{G}}(\epsilon) \triangleq \frac{\partial \exp_{\mathcal{G}}^{\wedge}(\epsilon)}{\partial \epsilon} = \sum_{n=0}^{+\infty} \frac{\mathrm{ad}_{\mathcal{G}}(\epsilon)^n}{(n+1)!}. \tag{3.27}$$

This term often appears when operations on the algebra (or the Euclidean space) are projected to the group. This will be broadly discussed in the following section on the Baker-Campbell-Hausdorff formula. Note that this Jacobian is specific to the group and should not be confused with Jacobians of functions introduced in future sections.
The matrix inverse of $\Phi_{\mathcal{G}}$ is often useful. It is denoted $\varphi_{\mathcal{G}}$, and writes:

$$\forall \epsilon \in \mathbb{R}^d, \varphi_{\mathcal{G}}(\epsilon) = \sum_{n=0}^{+\infty} \frac{\mathcal{B}_n \mathrm{ad}_{\mathcal{G}}(\epsilon)^n}{n!}, \tag{3.28}$$

where $\mathcal{B}_n$ represent the Bernoulli numbers [1].

3.4 . The Baker-Campbell-Hausdorff Formula

The Baker-Campbell-Hausdorff formula, also referred to as BCH, plays a key role in information theory on Lie groups. Indeed, the non-commutativity of the group law implies that:

$$\log_{\mathcal{G}}^{\vee}\left(\exp_{\mathcal{G}}^{\wedge}(a)\exp_{\mathcal{G}}^{\wedge}(b)\right) = \mathrm{BCH}(a,b) \neq a+b. \tag{3.29}$$

The integral expression of BCH is popular in physics. It is valid at the neighborhood of the identity point of the group within the bounds of the inequality [42]:

$$(a,b) \in \mathbb{R}^d : \|a\|_2 \text{ and } \|b\|_2 < \frac{1}{2}\log\left(2 - \frac{\sqrt{2}}{2}\right). \tag{3.30}$$

Under this assumption, BCH writes:

$$\log_{\mathcal{G}}^{\vee}\left(\exp_{\mathcal{G}}^{\wedge}(a)\exp_{\mathcal{G}}^{\wedge}(b)\right) = a + \left[\int_0^1 \Psi(\exp_{\mathcal{G}}^{\wedge}(\mathrm{ad}_{\mathcal{G}}a)\exp_{\mathcal{G}}^{\wedge}(t\,\mathrm{ad}_{\mathcal{G}}b))\mathrm{d}t\right] b, \tag{3.31}$$

where Ψ is defined as:

$$\Psi(x) = x \sum_{i=0}^{\infty} \frac{(-1)^i (x - I_d)^i}{i+1}. \tag{3.32}$$

The Hausdorff Series is another widespread formulation of BCH for matrix Lie groups, and is detailed in [42]. The knowledge of the first terms provides an accurate approximation:

$$\log_{\mathcal{G}}^{\vee}\left(\exp_{\mathcal{G}}^{\wedge}(a)\exp_{\mathcal{G}}^{\wedge}(b)\right) \approx a + b + \frac{1}{2}[[a]^{\wedge}, [b]^{\wedge}]^{\vee} + \cdots \tag{3.33}$$

where $[\,\cdot\,,\,\cdot\,]$ denotes the Lie brackets. Besides, when a and b are close to 0, a zero-order approximation is often sufficient:

$$\text{if } a \approx 0 \text{ and } b \approx 0 : \log_{\mathcal{G}}^{\vee}\left(\exp_{\mathcal{G}}^{\wedge}(a)\exp_{\mathcal{G}}^{\wedge}(b)\right) \approx a + b. \tag{3.34}$$

Note taht other accurate formulations of BCH exist. The approach described in [46] shows that a linearization in a or b leads to two different first-order expressions of BCH formula.

If $\|b\|_2$ is small with respect to $\|a\|_2$:

$$\log_{\mathcal{G}}^{\vee}\left(\exp_{\mathcal{G}}^{\wedge}(a)\exp_{\mathcal{G}}^{\wedge}(b)\right) = b + \varphi_{\mathcal{G}}(b)a + O(\|a\|_2^2). \tag{3.35}$$

If $\|a\|_2$ is small with respect to $\|b\|_2$:

$$\log_{\mathcal{G}}^{\vee}\left(\exp_{\mathcal{G}}^{\wedge}(a)\exp_{\mathcal{G}}^{\wedge}(b)\right) = a + \varphi_{\mathcal{G}}(-a)b + O(\|b\|_2^2). \tag{3.36}$$

In practice, other expressions provided in [14] are highly relevant as these configurations often occur in calculus:

$$\log_{\mathcal{G}}^{\vee}\left(\exp_{\mathcal{G}}^{\wedge}(-a)\exp_{\mathcal{G}}^{\wedge}(a+b)\right) = \Phi_{\mathcal{G}}(-a)b + O(\|b\|^2), \tag{3.37}$$

$$\log_{\mathcal{G}}^{\vee}\left(\exp_{\mathcal{G}}^{\wedge}(a+b)\exp_{\mathcal{G}}^{\wedge}(-a)\right) = \Phi_{\mathcal{G}}(a)b + O(\|b\|^2). \tag{3.38}$$

3.5 . Differential and Integral Calculus

The usual derivative requires a slight adaptation to be applicable to Lie groups. Indeed, considering a smooth function $f : \mathcal{G} \to \mathbb{R}$, its derivative is: $\lim_{h\to 0}\left[f(X+h) - f(X)\right]/h$. However the $+$ operation is not group-stable: $X + h \notin \mathcal{G}$.
Another approach uses the directional derivative defined in the Lie algebra [28], which can be defined in two ways. Let $f : \mathcal{G} \to \mathcal{G}$ be a smooth function, the left derivative on $\mathcal{G}$ writes:

$$\partial_X f(X) \triangleq \lim_{t\to 0} \frac{f(X\exp_{\mathcal{G}}^{\wedge}(t\epsilon)) - f(X)}{t}, \tag{3.39}$$

where $\epsilon \in \mathbb{R}^d$. For more natural notations, this paper uses:

$$\frac{\partial}{\partial \epsilon} f(X\exp_{\mathcal{G}}^{\wedge}(\epsilon))\big|_{\epsilon=0} \triangleq \partial_X f(X). \tag{3.40}$$

In the sequel, the derivative is always implicitly computed for $\epsilon = 0$.
The derivative on Lie groups preserves important properties of the derivative on $\mathbb{R}^d$. Let f and g be two smooth functions on unimodular Lie groups. In that case, [27] shows that the customary product rule applies for ∂_X:

$$\partial_X(fg)(X) = \partial_X f(X)\, g(X) + f(X)\, \partial_X g(X). \tag{3.41}$$

Besides, the derivative chain rule applies as well:

$$\partial_X f(g(X)) = \left.\frac{\partial f(u)}{\partial u^T}\right|_{u=g(X)} \partial_X g(X). \tag{3.42}$$

The sequel focuses on the integrals on Lie groups. Let f be an integrable function defined on $\mathcal{G}$, a unimodular Lie group. Its integral is defined as [27]:

$$\int_{\mathcal{G}} f(X)\mathrm{d}X \triangleq \int_{\mathbb{R}^d} f(\exp_{\mathcal{G}}^{\wedge}(\epsilon))\, |\det\left[\Phi_{\mathcal{G}}(\epsilon)\right]|\, \mathrm{d}\epsilon, \tag{3.43}$$

where $\mathrm{d}X$ is a Haar measure, $\mathrm{d}\epsilon$ is a Lebesgue measure, $\Phi_{\mathcal{G}}(\epsilon)$ is defined in (3.27), and ϵ is the log-linear error on the group. This definition can be seen as the change of measure:

$$\mathrm{d}X \to |\det\left[\Phi_{\mathcal{G}}(\epsilon)\right]|\, \mathrm{d}\epsilon. \tag{3.44}$$

The integrals defined on unimodular Lie groups verify similar properties as the ones defined on the Euclidean space [27]. First, the integral is unchanged by the inverse change of measure:

$$\int_{\mathcal{G}} f(X)\mathrm{d}X = \int_{\mathcal{G}} f(X^{-1})\mathrm{d}X. \tag{3.45}$$

Another key property of the Haar measure for unimodular Lie group is its bi-invariance, meaning that the integral is unchanged by a left or right multiplication:

$$\forall \mu \in \mathcal{G} : \int_{\mathcal{G}} f(X)\mathrm{d}X = \int_{\mathcal{G}} f(\mu X)\mathrm{d}X = \int_{\mathcal{G}} f(X\mu)\mathrm{d}X. \tag{3.46}$$

3.6 . Probability densities

The concept of probability densities on Lie groups is broadly discussed in [81]. Previous works extended the customary probability density functions to unimodular matrix Lie groups. Indeed, this category of Lie groups present important properties for the concept of density to hold.

3.6.1 . General introduction

Let X be a random matrix on $\mathcal{G}$, a unimodular Lie group, and $p : \mathcal{G} \to [0, 1]$ be a probability density function. Then, according to [81], $\mathcal{G}$ has a Haar measure $\mathrm{d}X$ for which:

$$\int_G p(X)\mathrm{d}X = 1. \tag{3.47}$$

The probability density can be also defined with the change of measure presented in (3.44), leading to:

$$\int_{\mathbb{R}^d} p(\exp_{\mathcal{G}}^{\wedge}(\epsilon)) \left|\det\left[\Phi_{\mathcal{G}}(\epsilon)\right]\right| \mathrm{d}\epsilon = 1. \tag{3.48}$$

In this thesis, both formulations of the densities are used. For the sake of simplicity, the variables X and ϵ can be used in the same expressions when the context is clear. This notation abuse enables compact calculus and more readability of long expressions.

3.6.2 . Mean and covariance matrix

Let X be a random matrix and $p(X)$ a probability density function on $\mathcal{G}$, a unimodular Lie group. Its first moment μ, also referred to as the expectation or mean value, is defined by:

$$\mu \text{ such that } \int_{\mathcal{G}} \log_{\mathcal{G}}^{\vee}\left(\mu^{-1}X\right) p(X)\mathrm{d}X = 0. \tag{3.49}$$

This definition also represent the expectation of X:

$$\mu = \mathbb{E}\left[X\right]. \tag{3.50}$$

The mean is the same regardless the side of the product between μ and X. Indeed, consider the right formulation:

$$\int_{\mathcal{G}} \log_{\mathcal{G}}^{\vee}\left(X\mu^{-1}\right) p(X)\mathrm{d}X = 0. \tag{3.51}$$

The integral invariance property from (3.46) leads to the same definition of the mean. Note that other definition of the mean exist, but this is the only one which is consistent either on the left or the right. Similarly, the definition of covariance matrix holds in Lie groups as well.

The left covariance matrix P is defined as the second moment of p:

$$P \triangleq \int_{\mathcal{G}} \log_{\mathcal{G}}^{\vee}\left(\mu^{-1}X\right) \log_{\mathcal{G}}^{\vee}\left(\mu^{-1}X\right)^{T} p(X)\mathrm{d}X, \tag{3.52}$$

which can be denoted:

$$P = \mathbb{E}\left[\log_{\mathcal{G}}^{\vee}(\mu^{-1}X) \log_{\mathcal{G}}^{\vee}(\mu^{-1}X)^{T}\right] \tag{3.53}$$

Then, in the case of a right error, the covariance matrix writes:

$$P \triangleq \int_{\mathcal{G}} \log_{\mathcal{G}}^{\vee}\left(X\mu^{-1}\right) \log_{\mathcal{G}}^{\vee}\left(X\mu^{-1}\right)^{T} p(X)\mathrm{d}X, \tag{3.54}$$

and the expectation notation is:

$$P = \mathbb{E}\left[\log_{\mathcal{G}}^{\vee}(X\mu^{-1}) \log_{\mathcal{G}}^{\vee}(X\mu^{-1})^{T}\right] \tag{3.55}$$

Note that for the sake of brevity, the covariance matrix is denoted $P = \mathbb{V}[X]$, where the left or right definition will be chosen according to the side of the error.

The covariance matrix can be also defined with the expectation using the log-linear error. It has a general formulations which apply either to the left and right case:

$$P = \mathbb{E}\left[\epsilon\epsilon^T\right]. \tag{3.56}$$

Unlike the mean, the right formulation of the covariance matrix differs from the left as it involves a different error function: $\log_{\mathcal{G}}^{\vee}\left(\mu^{-1}X\right) \neq \log_{\mathcal{G}}^{\vee}\left(X\mu^{-1}\right)$.
In this thesis, the computation of the mean and covariance matrix of a finite sample will be required. The sequel describes this specific case:

A sample of matrices $X^i,\ i \in [1, N_p]$ on $\mathcal{G}$, with the corresponding weights $w^i,\ i \in [1, N_p]$ such that $\sum_{i=1}^{N_p} w^i = 1$, and is denoted:

$$\left\{X^i, w^i\right\}_{i\in[1,N_p]}. \tag{3.57}$$

Based on the definition (3.49), the mean of the sample is:

$$\mu \text{ such that } \sum_{i=1}^{N_p} w^i \log_{\mathcal{G}}^{\vee}\left(\mu^{-1}X^i\right) = 0. \tag{3.58}$$

The notion of mean on Lie groups is broadly discussed in [66]. note that the definition (3.58) has similar properties to the usual mean on the Euclidean space. It is bi-invariant, meaning that it is consistent with a left or right multiplication of the sample: $\forall g \in \mathcal{G}$ the mean of $\left\{gX^i, w^i\right\}_{i\in[1,N_p]}$ is $g\mu$.
The mean of a sample has to be carefully computed as Lie groups are not vector spaces. Hence, it cannot be calculated with a direct sum of the weighted particles as the "$+$" operator does not belong to the group:

$$\mu \neq \sum_{i=1}^{N_p} w^i X^i.$$

Another approach would be to use the group logarithm and exponential to bring the samples to the Euclidean space, compute the mean on the Euclidean space, and send this mean back to the group with the exponential: $\exp_{\mathcal{G}}^{\wedge}\left(\sum_{i=1}^{N_p} w^i \log_{\mathcal{G}}^{\vee}\left(X^i\right)\right)$. This expression is usually called the group exponential barycenter. Although this formulation is group-stable, it is not invariant under a right or left multiplication of the sample:

$$\forall g \in \mathcal{G} : g\exp_{\mathcal{G}}^{\wedge}\left(\sum_{i=1}^{N_p} w^i \log_{\mathcal{G}}^{\vee}\left(X^i\right)\right) \neq \exp_{\mathcal{G}}^{\wedge}\left(\sum_{i=1}^{N_p} w^i \log_{\mathcal{G}}^{\vee}\left(gX^i\right)\right).$$

As discussed in [66], the definition (3.58) is the only one which is consistent with the group geometry and shows the usual properties.

3.6.3 . Concentrated Gaussian

Let $X \in \mathcal{G}$ be a random matrix which follows a left concentrated Gaussian distribution on $\mathcal{G}$:

$$X \sim \mathcal{N}_{\mathcal{G}}^{L}(X; \mu, P) \; ; \; X = \mu \exp_{\mathcal{G}}^{\wedge}(\epsilon), \tag{3.59}$$

where $\epsilon \sim \mathcal{N}_{\mathbb{R}^d}(0, P)$. This definition holds when the density is concentrated around its mean, that is to say, all eigenvalues of P are small enough [28]. Its probability density function writes:

$$p(X) \approx \frac{1}{\sqrt{(2\pi)^d \det[P]}} e^{-\frac{1}{2}\left\|\log_{\mathcal{G}}^{\vee}(\mu^{-1}X)\right\|_P^2}. \tag{3.60}$$

This formulation is only an approximation as pointed out in [14], since a reparametrization term appears on the covariance matrix of the normalization factor. The exact form of the left Gaussian writes:

$$p(X) = \frac{1}{\sqrt{(2\pi)^d \det\left[\Phi_{\mathcal{G}}(\epsilon) P \Phi_{\mathcal{G}}^{T}(\epsilon)\right]}} e^{-\frac{1}{2}\left\|\log_{\mathcal{G}}^{\vee}(\mu^{-1}X)\right\|_P^2}, \tag{3.61}$$

where $\epsilon = \log_{\mathcal{G}}^{\vee}\left(\mu^{-1}X\right)$. However, since the density is assumed to be concentrated, $\Phi_{\mathcal{G}}(\epsilon)$ is close to identity. Thus, this term is often overlooked for simplicity.
Similarly, if X follows a right concentrated Gaussian on $\mathcal{G}$:

$$X \sim \mathcal{N}_{\mathcal{G}}^{R}(X; \mu, P) \; ; \; X = \exp_{\mathcal{G}}^{\wedge}(\epsilon)\mu, \tag{3.62}$$

where $\epsilon \sim \mathcal{N}_{\mathbb{R}^d}(0, P)$. The density is given by:

$$p(X) \approx \frac{1}{\sqrt{(2\pi)^d \det[P]}} e^{-\frac{1}{2}\left\|\log_{\mathcal{G}}^{\vee}(X\mu^{-1})\right\|_P^2}. \tag{3.63}$$

It is important to stress that the left and right Gaussian functions represent two different densities. Although they will have the same mean, their variance can significantly differ as $\log_{\mathcal{G}}^{\vee}\left(X\mu^{-1}\right) \neq \log_{\mathcal{G}}^{\vee}\left(\mu^{-1}X\right)$ in general. Also, for the sake of brevity and when the context is clearly indicated, left and right Gaussians on $\mathcal{G}$ are both denoted $\mathcal{N}_{\mathcal{G}}(X; \mu, P)$.

3.6.4 . Dirac Distribution

The definition of the Dirac distribution is momentous for Particle Filter on Lie groups as it represents the mathematical nature of a particle.
In the Euclidean case, it is formally defined as the Lebesgue measure $\delta(x)$ such that:

$$\int_{\mathbb{R}} f(x)\delta(x) = f(0). \tag{3.64}$$

Besides, the Dirac function can be translated at any point $a \in \mathcal{R}$ according to the following notation:

$$\delta_a(x) \triangleq \delta(x - a). \tag{3.65}$$

The framework of unimodular Lie groups admits an extension of the Dirac function as described in [27]. Let X and μ be two matrices in $\mathcal{G}$, a unimodular Lie group. They can be written as a

linear combination of the basis of the Euclidean space $[e_1 \cdots e_n]$ projected to the group with the group exponential:

$$\begin{aligned} X &= \exp_{\mathcal{G}}^{\wedge}\left(\sum_{n=1}^{d} a_n e_n\right), \\ \mu &= \exp_{\mathcal{G}}^{\wedge}\left(\sum_{n=1}^{d} b_n e_n\right), \end{aligned} \tag{3.66}$$

where $a_n, b_n \in \mathbb{R}$.
The Dirac function on $\mathcal{G}$ at point $\mu \in \mathcal{G}$ is defined as:

$$\delta_\mu(X) \triangleq \prod_{n=1}^{d} \delta_{b_n e_n}(a_n e_n). \tag{3.67}$$

Note that in the sequel, there will be no distinctions in the notation of a Dirac function on Lie groups and a Dirac function on the Euclidean space. When the inputs of δ are $\mathbb{R}^d$ vectors, definition (3.64) holds, and when they are Lie group matrices, definition (3.67) holds.

3.6.5 . Sampling

A sample of N elements on a Lie group $\mathcal{G}$ is denoted $\left\{X^i\right\}_{i\in[1,N]} = \left\{X^1, \cdots, X^N\right\}$. It is usually obtained from a centered sample on the Euclidean space projected on the Group with an exponential and a left or right multiplication of the mean. Let $\left\{\epsilon^i\right\}_{i\in[1,N]}$ be a centered sample such that $\epsilon^i \sim \pi(\epsilon; 0, P)$ where π is a probability law on the Euclidean space. Then, the sample of mean μ on the Group writes:

$$\begin{aligned} &\text{Left:} \quad X^i = \mu \exp_{\mathcal{G}}^{\wedge}\left(\epsilon^i\right), \\ &\text{Right:} \quad X^i = \exp_{\mathcal{G}}^{\wedge}\left(\epsilon^i\right) \mu. \end{aligned} \tag{3.68}$$

This sample copes with the geometry of the group and often has a curved shape, as illustrated in Figure 3.2.

3.7 . Fisher Information Matrix

Let p be a probability density function on an unimodular Lie group $\mathcal{G}$ and $\widehat{X}$ be an estimator of a random matrix X. Assuming that p is symmetric $p(X^{-1}) = p(X)$, [27] shows that its Fisher information matrix is defined by:

$$J \triangleq \int_{\mathcal{G}} \frac{1}{p(X)} \left(\frac{\partial p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon))}{\partial \epsilon}\right) \left(\frac{\partial p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon))}{\partial \epsilon}\right)^T \mathrm{d}X. \tag{3.69}$$

Besides, a covariance matrix on $\mathcal{G}$ defined as (3.54) verifies:

$$P \geq J^{-1}, \tag{3.70}$$

in the sense that the eigenvalues of $P - J^{-1}$ are positive.

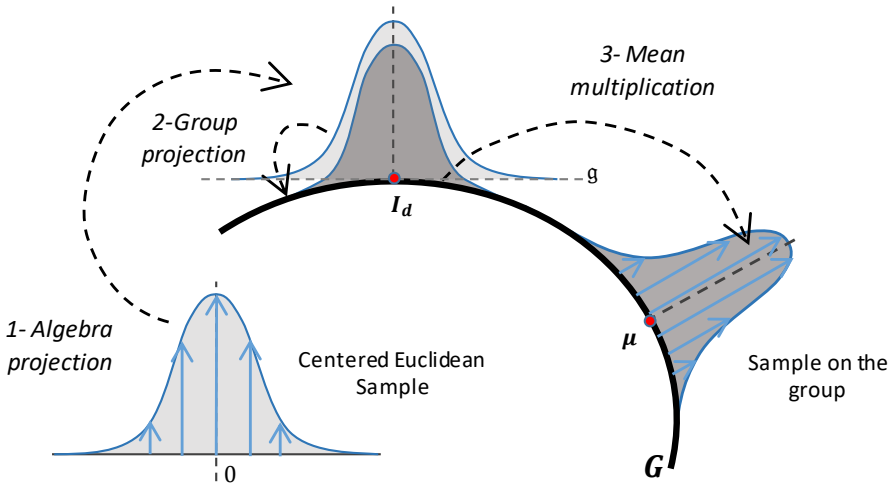


Figure 3.2: Illustration of the sampling process on Lie groups.

Lemma 1. *The Fisher information matrix J defined in* (3.69) *can be written in the alternative form:*

$$J = -\mathbb{E}\left[\frac{\partial^2 \log p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi))}{\partial \epsilon \partial \xi}\right]. \tag{3.71}$$

Proof. Consider the following derivative denoted $(\mathcal{D})$:

$$(\mathcal{D}) = \frac{\partial^2}{\partial \epsilon \partial \xi} \log p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi)). \tag{3.72}$$

The derivation with respect to ξ gives:

$$(\mathcal{D}) = \frac{\partial}{\partial \epsilon} \frac{1}{p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon))} \frac{\partial}{\partial \xi} p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi)). \tag{3.73}$$

Then, the derivation with respect to ϵ leads to two terms:

$$(\mathcal{D}) = \frac{1}{p(X)^2} \frac{\partial}{\partial \epsilon} p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon)) \frac{\partial}{\partial \xi} p(X \exp_{\mathcal{G}}^{\wedge}(\xi)) + \frac{1}{p(X)} \frac{\partial^2}{\partial \epsilon \partial \xi} p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi)). \tag{3.74}$$

By applying the expectation on the second term of (3.74) gives:

$$\begin{aligned} \mathbb{E}\left[\frac{1}{p(X)} \frac{\partial^2}{\partial \epsilon \partial \xi} p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi))\right] &= \int_G \frac{\partial^2}{\partial \epsilon \partial \xi} p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi)) \mathrm{d}X, \\ &= \frac{\partial^2}{\partial \epsilon \partial \xi} \int_G p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi)) \mathrm{d}X. \end{aligned} \tag{3.75}$$

The invariance property of the integral from (3.46) gives:

$$\int_G p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi)) \mathrm{d}X = \int_G p(X) \mathrm{d}X = 1, \tag{3.76}$$

leading to:

$$\mathbb{E}\left[\frac{1}{p(X)} \frac{\partial^2}{\partial \epsilon \partial \xi} p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi))\right] = 0. \tag{3.77}$$

By applying now the expectation to the first term of (3.74):

$$\begin{aligned} &\mathbb{E}\left[\frac{1}{p(X)^2} \frac{\partial p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon))}{\partial \epsilon} \frac{\partial p(X \exp_{\mathcal{G}}^{\wedge}(\xi))^T}{\partial \xi}\right] \\ &\qquad = \int_{\mathcal{G}} \frac{1}{p(X)} \left(\frac{\partial p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon))}{\partial \epsilon}\right) \left(\frac{\partial p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon))}{\partial \epsilon}\right)^T \mathrm{d}X, \end{aligned} \tag{3.78}$$

which concludes the proof. □

This alternative formulation is equivalent to the one proposed in [48], which can be seen as a Hessian on the group.

3.8 . Link with Riemannian manifolds

The properties introduced in this Chapter can be extended to Riemannian manifolds, as detailed in [75]. The main advantage of Lie groups is to have a unique bi-invariant metric with the group logarithm.

4 - Bayesian Filters on Lie Groups and Manifolds

Over the last decade, estimation filters on Lie groups has gained interest among researchers and engineers for a wide range of applications. This Chapter introduces some of the main methods developed in the literature. The works of Barrau and Bonnabel on a specific class of nonlinear systems on Lie groups led to the Invariant Extended Kalman Filter (IEKF) which was adapted to a new version of Rao-Blackwellized particle filter. At the same time, Giremus and Bourmaud developed the Extended Kalman Filter on Lie groups (LG-EKF), which provides a generic formulation for EKF on matrix Lie groups. Following the works of Barrau, Brossard developed the Unscented Kalman Filter on Lie groups which became a popular approach to solve Visual Inertial Odometry (VIO) and Visual Simultaneous Localization and Mapping (V-SLAM) problems. A different approach was developed by Snoussi et al. with Particle Filter where the system is constained on a Riemannian manifold.

4.1 . Invariant Filtering

4.1.1 . Invariant Kalman Filter

Invariant Extended Kalman Filter (IEKF) is one of the first breakthrough introduced by the use of Lie groups in estimation. When the observation and dynamics models verify hypothesis described in the sequel, IEKF uses geometric and algebraic properties of Lie groups to obtain a linear behavior for a class of nonlinear system described in [7]. This unique result in nonlinear estimation led to stability properties [4, 6].
Let $X_k \in \mathcal{G}$ be a matrix whose evolution is described by the general discrete observation system:

$$\begin{cases} X_{k+1} = f(X_k, u_{k+1}, \nu_k^q), \\ Y_{k+1} = h(X_{k+1}, \nu_{k+1}^r). \end{cases} \tag{4.1}$$

IEKF theory is based on two core assumptions on f and h such that the propagation and update errors are autonomous with respect to the estimated state. First, it is assumed that f verifies:

$$\forall X_1, X_2 \in \mathcal{G} : f(X_1, X_2) = f(X_1) f(I_d)^{-1} f(X_2). \tag{4.2}$$

In that case, f is group-affine [7]. Besides, it is assumed that h describes a group action of $\mathcal{G}$ on a set $\mathcal{S}$ as defined in (3.23):

$$\forall X \in \mathcal{G}, s \in \mathcal{S} : h(X_k) = X_k * s_k. \tag{4.3}$$

Then, the estimation error on the deterministic system (4.1) is autonomous with respect to the state [7]. This specific result enables to design a filter with invariance properties, which leads to stability results for this class of nonlinear systems [6]. Technical developments on IEKF can be found in [4].

4.1.2 . Imperfect Invariant Extended Kalman Filter

Although IEKF suits to a quite general framework, its implementation does not cope with the estimation of some parameters such as biases. In that case, [7] shows that IEKF framework does not apply as the dynamics model function is not group-affine.
This problem is soved with the "Imperfect" IEKF introduced in [4]. The state variables are represented by a matrix and a vector $(X_k \in \mathcal{G}, \theta_k \in \mathbb{R}^n)$ and follows the system:

$$\begin{cases} X_{k+1} &= f_{\theta_k}(X_k, u_{k+1}), \\ \theta_{k+1} &= g(\theta_k), \\ Y_{k+1} &= h(X_{k+1} * b, \theta_k), \end{cases} \tag{4.4}$$

where $b \in \mathbb{R}^p$, f_{θ_k} is groupe-affine in the sense of (4.2). The imperfect IEKF is built on the state error (in the left case):

$$e_k = \begin{bmatrix} \log_{\mathcal{G}}^{\vee}\left(\widehat{X}_k^{-1} X_k\right) \\ \theta_k - \hat{\theta}_k \end{bmatrix}. \tag{4.5}$$

This hybrid error describes a groupe-affine dynamic with the matrix product and includes the parameters on the Euclidean space. The matrix error on X is handled with IEKF and the vectorial error is handled with EKF. Thus, in practical navigation problems, the main kinematics variables are represnted in the matrix and the biases in the vector.
Although this filter does not have mathematical stability properties, the core of the state is estimated with an IEKF, making the imperfect IEKF highly accurate and robust in practice. Besides, (4.4) is suitable to most estimation problems, and generalizes IEKF framework.

4.1.3 . Invariant Rao-Blackwellized Particle Filter

Invariant Particle Filter is an extension of IEKF where specific parameters are identified with a Monte Carlo approach. Indeed, IEKF framework cannot handle biases often encountered in nonlinear estimation (e.g. accelerometer or gyrometer biases).
The approach developed in [5, 4] splits the state variables in a matrix X and a vector θ, following the continuous-discrete state space model:

$$\begin{cases} \dot{\theta} &= f(\theta, \nu), \\ \dot{X} &= g(\theta, X, u) + Xv, \\ Y_k &= h_\theta(X_k p + w_k), \end{cases} \tag{4.6}$$

where (ν, v, w_k) are noise vectors, and p is a vector. Besides, it is assumed that g is group-affine with respect to X in the sense of (4.2) and verifies:

$$g(\theta, X, u) - g(\theta, I, u)X = g(0, X, u) - g(0, I, u)X. \tag{4.7}$$

Under these assumptions, [5] describes a process where X is estimated with an IEKF and θ with a particle filter, which can be identified as an Invariant Rao-Blackwellized Particle Filter. This filter benefits from the properties of IEKF as the Kalman gains are independent from both θ and X. Hence, a single Kalman gain is sufficient to update all the particles during the update step.

4.2 . Extended Kalman Filter on Lie Groups

Extended Kalman Filter on Lie groups (LG-EKF) was introduced by Bourmaud et al. in [15]. It consists in a direct linearization of the group error, leading to a formalism close to usual EKF. Although LG-EKF does not have theoretical convergence properties, it comes with a natural robustness. Besides, to some extent, IEKF can be seen as a linear version of Extended Kalman Filter on Lie groups (LG-EKF).
Consider the discrete state space system:

$$\begin{cases} X_{k+1} &= f(X_k, u_{k+1}, w_k^q), \\ Y_{k+1} &= h(X_{k+1}, w_{k+1}^r), \end{cases} \tag{4.8}$$

where the state matrix X_k belongs to a matrix Lie group $\mathcal{G}$ and Y_{k+1} to a matrix Lie group $\mathcal{H}$. The approach described by [15] assumes that the system (4.8) writes:

$$\text{Left case: } f(X_k, u_{k+1}, \nu_k^q) = X_k \exp_{\mathcal{G}}^{\wedge}\left(\Omega(X_k, u_{k+1}) + \nu_{k+1}^q\right), \tag{4.9}$$

$$\text{Right case: } f(X_k, u_{k+1}, \nu_k^q) = \exp_{\mathcal{G}}^{\wedge}\left(\Omega(X_k, u_{k+1}) + \nu_{k+1}^q\right) X_k, \tag{4.10}$$

where n_k^q is a centered Gaussian noise vector and Ω a smooth function. Also, the observation model is assumed to have a Gaussian multiplicative noise:

$$h(X_k, \nu_k^r) = h(X_k) \exp_{\mathcal{G}}^{\wedge}(\nu_k^r). \tag{4.11}$$

The filter obtained with this system is described in Algorithm 5 where the propagation Jacobian is given by [14]:

$$F_k = Ad_{\mathcal{G}}\left(\exp_{\mathcal{G}}^{\wedge}\left(-\Omega(\hat{\mu}_{k|k}, u_{k+1})\right)\right) + \Phi_{\mathcal{G}}(-\Omega(\hat{\mu}_{k|k}, u_{k+1}))\mathcal{C}_k, \tag{4.12}$$

where:

$$\mathcal{C}_k = \frac{\partial \log_{\mathcal{G}}^{\vee}\left(\Omega(\hat{\mu}_{k|k} \exp_{\mathcal{G}}^{\wedge}(\epsilon), u_{k+1})\right)}{\partial \epsilon}. \tag{4.13}$$

The update Jacobian is a direct linearization of the measurement model:

$$H_{k+1} = -\frac{\partial \log_{\mathcal{H}}^{\vee}\left(h(\hat{\mu}_{k+1} \exp_{\mathcal{G}}^{\wedge}(\epsilon))^{-1} Y_{k+1}\right)}{\partial \epsilon}. \tag{4.14}$$

Algorithm 5 Left Lie Group Extended Kalman Filter

Result: $(\hat{\mu}_{k|k}, P_{k|k})$, $k \in [1, N]$
Inputs: $\hat{\mu}_0$, P_0, $y_{1:N}$
Propagation:
$\hat{\mu}_{k+1|k} = \hat{\mu}_{k|k} \exp_{\mathcal{G}}^{\wedge}\left(\Omega(\hat{\mu}_{k|k}, u_{k+1})\right)$
$P_{k+1|k} = F_k P_{k|k} F_k^T + \Phi_{\mathcal{G}}(\Omega(\hat{\mu}_{k|k}, u_{k+1}))Q_{k+1}\Phi_{\mathcal{G}}^T(\Omega(\hat{\mu}_{k|k}, u_{k+1}))$
Update:
$K_{k+1} = P_{k+1|k} H_{k+1}^T \left(H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}\right)^{-1}$
$\hat{\mu}_{k+1|k+1} = \hat{\mu}_{k+1|k} \exp_{\mathcal{G}}^{\wedge}\left(K_{k+1} \log_{\mathcal{H}}^{\vee}\left(h(\hat{\mu}_{k+1|k})^{-1} y_{k+1}\right)\right)$
$P_{k+1|k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k}$

4.3 . Unscented Kalman Filter on Lie Groups

Lie Group Unscented Kalman Filter (LG-UKF) hinges on the same principle of UKF developed by Uhlmann [44]. It was introduced by Brossard et al. [17] and demonstrated its interest for robotics applications [18, 51, 20].
Assume that the state matrix X belongs to a Lie group $\mathcal{G}$ of dimension d, ane follows a discrete process according to the following state-space model, where Y_k belongs to $\mathcal{H}$, a Lie group of dimension m:

$$\begin{cases} X_{k+1} &= X_k \exp_{\mathcal{G}}^{\wedge}\left(\Omega(X_k, u_{k+1}) + \nu_{k+1}^{q}\right), \\ Y_{k+1} &= h(X_{k+1}) \exp_{\mathcal{H}}^{\wedge}\left(\nu_{k+1}^{r}\right), \end{cases} \tag{4.15}$$

where $\nu_{k+1}^{q} \sim \mathcal{N}(0, Q_{k+1})$ and $\nu_{k+1}^{r} \sim \mathcal{N}(0, R_{k+1})$. The difference with the usual UKF is to draw the sigma-points in the Lie algebra, which copes with the geometry of the group.
Similar approaches exist where only the rotation matix is defined on the Lie group [20]. Also, [18] simplifies this algorithm by considering a vectorial measurement model (i.e. $\mathcal{H} = \mathbb{R}^m$) with additive noise.

4.4 . Particle Filter on Manifolds

Particle Filter on Manifold is related to this thesis as it focuses on Monte Carlo methods on non-Euclidean spaces. It was developed by Snoussi and al. in [77] and it sets a framework for particle filtering in Riemannian manifolds. The specificity of this approach is to define the stochastic processes in tangent spaces to the manifold. Then, the inner product of each tangent space can be used to define an error which is analogous to the log-Euclidean error.
Lie groups and Riemannian manifolds have several similarities. Indeed, every tangent space of the manifold admits an exponential and a logarithm which work in the same ways as for Lie groups. Besides, every tangent space at the manifold is a vector space with a bi-invariant metric, which enables to design filters. However, the main advantage of Lie group is the unicity of these elements. In the case of Riemannian manifolds, the exponential and logarithm differ from one tangent space to another, which involves the definition of a new metric for each tangent space.

Part II

Nonlinear Filtering on Lie Groups

5 - Bayes Optimal Filter on Lie Groups

This Chapter represents the first theoretical contribution of the thesis. It characterizes the Bayes Filter on Lie groups based on the probabilistic considerations of Chapter 3. It sets the framework of all the Chapters of Part II which are based on the assumptions described in the sequel.

5.1 . Framework and Assumptions

This section focuses on the filtering aspects of Bayesian fusion on Lie groups discussed in [81, 27]. All the filters proposed in the next chapters will be derived from this formulation.
Let $\{X_k\}_{k\in\mathbb{N}} \in \mathcal{G}$ be the discrete-time random process describing the evolution of a sequence of hidden states, according to a set of observations $\{Y_k\}_{k\in\mathbb{N}} \in \mathcal{H}$, where $\mathcal{G}$ and $\mathcal{H}$ are unimodular matrix Lie groups of respective dimensions d and m. Consider the generic state-space model:

$$\begin{cases} X_{k+1} &= f(X_k, u_{k+1}, \nu_k^q), \\ Y_{k+1} &= h(X_{k+1}, \nu_{k+1}^r), \end{cases} \tag{5.1}$$

where f and h are two nonlinear functions.
Similarly to the Euclidean case, the filtering problem on Lie groups lies in the estimation of the posterior density $p(X_k|Y_{1:k})$, where $Y_{1:k} = \{Y_1, \dots, Y_k\}$, under the following assumptions:

- The sequence $X_{1:k} = \{X_1, \cdots, X_k\}$ describes a Markov process which transition density is known:
$$p(X_{k+1}|X_{1:k}) = p(X_{k+1}|X_k). \tag{5.2}$$
- The observations $Y_{1:k}$ are conditionally independent given the state process:
$$p(Y_{1:k}|X_{1:k}) = \prod_{i=1}^{k} p(Y_i|X_i). \tag{5.3}$$
- The initial state probability density function $p(X_0)$ is known.

5.2 . Derivation of the Filter

The estimation process hinges on the Optimal Filter (or Bayes filter) which is the most general approach in Bayesian estimation. The Bayes filter applies to Lie groups as they enable the definition of probability densities [27]. The sequel assumes unimodular Lie groups for which the analysis and probability tools described in Chapter 3 apply. The propagation step is given by the Chapman-Kolmogorov equation which is obtained with a marginalization on the group:

$$p(X_{k+1}|Y_{1:k}) = \int_{\mathcal{G}} p(X_{k+1}|X_k)p(X_k|Y_{1:k})\mathrm{d}X_k, \tag{5.4}$$

where $p(X_{k+1}|X_k)$ is the transition density and $p(X_k|Y_{1:k})$ is the posterior density.

The update is performed by the Bayes rule, which is obtained following the same principles as the Euclidean space:

$$p(X_{k+1}|Y_{1:k+1}) = \frac{p(Y_{k+1}|X_{k+1})p(X_{k+1}|Y_{1:k})}{p(Y_{k+1}|Y_{1:k})}. \tag{5.5}$$

The Bayes rule computes the posterior density $p(X_k|Y_{1:k})$ from the fusion of the likelihood $p(Y_{k+1}|X_{k+1})$ derived from a measurement model, and the prior density $p(X_{k+1}|Y_{1:k})$ which represents the propagated density from (5.4).

Note that $p(Y_{k+1}|Y_{1:k})$ is the marginal likelihood. Ii is obtained with the following integral:

$$p(Y_{k+1}|Y_{1:k}) = \int_{\mathcal{G}} p(Y_{k+1}|X_{k+1})p(X_{k+1}|Y_{1:k})\mathrm{d}X_{k+1}. \tag{5.6}$$

The marginal likelihood is considered as a normalization term, and it is not computed in practice. The Optimal Filter on Lie group can be expressed with the following algorithm:

Algorithm 6 Lie Group Optimal Filter

Result: $p(X_k|Y_{1:k})$, $k \in [1, N]$

Propagation step: $p(X_{k+1}|Y_{1:k}) = \int_{\mathbb{R}^d} p(X_{k+1}|X_k)p(X_k|Y_{1:k})dX_k$

Update step: $p(X_{k+1}|Y_{1:k+1}) \propto p(Y_{k+1}|X_{k+1})p(X_{k+1}|Y_{1:k})$

5.3 . Estimators

Estimators play a key role in the filters of this thesis. As in the Euclidean case, the sequel focuses on the conditional mean and the *Maximum A Posteriori* (MAP).

5.3.1 . Conditional mean

The conditional mean at different steps of the filtering process is defined with the expectation on Lie groups introduced in Chapter 3:

$$\hat{\mu}_{k|k} \triangleq \mathbb{E}\left[X_k|Y_{1:k}\right], \tag{5.7}$$

$$\hat{\mu}_{k+1|k} \triangleq \mathbb{E}\left[X_{k+1}|Y_{1:k}\right]. \tag{5.8}$$

Because of the properties of unimodular Lie groups discussed in Chapter 3, these definitions are the same if X is a left or right random variable.
Then, the log-Euclidean error at different times is denoted:

Left case:

$$\epsilon_{k|k} \triangleq \log_{\mathcal{G}}^{\vee}\left(\hat{\mu}_{k|k}^{-1} X_k\right), \tag{5.9}$$

$$\epsilon_{k+1|k} \triangleq \log_{\mathcal{G}}^{\vee}\left(\hat{\mu}_{k+1|k}^{-1} X_{k+1}\right). \tag{5.10}$$

Right case:

$$\epsilon_{k|k} \triangleq \log_{\mathcal{G}}^{\vee}\left(X_k \hat{\mu}_{k|k}^{-1}\right), \tag{5.11}$$

$$\epsilon_{k+1|k} \triangleq \log_{\mathcal{G}}^{\vee}\left(X_{k+1} \hat{\mu}_{k+1|k}^{-1}\right). \tag{5.12}$$

Similarly, the conditional variance on Lie groups is denoted at different steps:

$$P_{k|k} \triangleq \mathbb{V}\left[X_k | Y_{1:k}\right] = \mathbb{E}\left[\epsilon_{k|k} \epsilon_{k|k}^T\right], \tag{5.13}$$

$$P_{k+1|k} \triangleq \mathbb{V}\left[X_{k+1} | Y_{1:k}\right] = \mathbb{E}\left[\epsilon_{k+1|k} \epsilon_{k+1|k}^T\right]. \tag{5.14}$$

Unlike the mean, the covariance matrix differs in the left and right cases as it involves two different errors.

5.3.2 . Maximum A Posteriori

The *Maximum A Posteriori* on Lie groups has a similar definition to the Euclidean case. It is the point $\mu^* \in G$ which maximizes the posterior density:

$$\mu_{k+1}^* = \arg\max_{X_{k+1}} p(X_{k+1} | Y_{1:k+1}). \tag{5.15}$$

Then, by applying the Bayes equation, a simplified expression of the MAP can be obtained as the product of the prior density and the likelihood:

$$\mu_{k+1}^* = \arg\max_{X_{k+1}} p(X_{k+1} | Y_{1:k}) p(Y_{k+1} | X_{k+1}). \tag{5.16}$$

5.4 . Conclusion

This Chapter introduces the Bayes Filter on unimodular matrix Lie groups with the definitions of the conditional expectation, conditional variance and *Maximum A Posteriori* on the group. This framework has strong similarities with its Euclidean counterpart, and enables the existence of all the tools required for nonlinear estimation. Therefore, this Chapter demonstrates that the Bayesian framework holds on unimodular matrix Lie groups, which establishes the theoretical background of the works introduced in this thesis. In the sequel, the Bayes Filter will be solved with different assumptions on the state space system. First, the assumption of Gaussian processes leads to the revisited Lie group Extended Kalman Filter (LG-EKF) described in in Chapter

6. Then, Optimal Filter will be solved without assumption on the model in Chapters 7 and 8, leading to new formulations of particle filters on Lie groups. Eventually, Chapter 9 focuses on the case where the noises are Gaussian and the estimated density is represented by a mixture of concentrated Gaussians.

6 - Extended Kalman Filter on Lie Groups Revisited

This Chapter solves the Optimal Filter on Lie groups presented in Chapter 5 assuming that the state X and the measurement Y are Gaussian random variables on $\mathcal{G}$ and $\mathcal{H}$. These assumptions lead to a formulation of Extended Kalman Filter on Lie groups (LG-EKF).
This filter was presented in previous works under similar hypothesis [14, 15] and is close to the Imperfect IEKF from [4]. The specificity of this chapter is to demonstrate that LG-EKF can be retreived from the Bayes filter equations on Lie groups. Also, it establishes a revisited approach of LG-EKF where the propagation noise follows a multiplicative model on the group. Besides, it describes a simplified formulation for the left and right Jacobians in the case where the state belongs to the multiple Special Euclidean group $\mathrm{SE}_p(3)$ defined in [10, 6].

6.1 . Problem statement

Let $\{X_k\}_{k\in\mathbb{N}} \in \mathcal{G}$ be the discrete-time process describing the evolution of a sequence of hidden states, according to a set of observations $\{Y_k\}_{k\in\mathbb{N}} \in \mathcal{H}$, where $\mathcal{G}$ and $\mathcal{H}$ are unimodular matrix Lie groups, and the assumptions of the optimal filter from Chapter 5 are verified.
A left propagation and measurement model writes:

$$\left\{\begin{array}{ll} X_{k+1} & = f(X_k)\exp_{\mathcal{G}}^{\wedge}\left(\nu_{k+1}^{q}\right), \\ Y_{k+1} & = h(X_{k+1})\exp_{\mathcal{G}}^{\wedge}\left(\nu_{k+1}^{r}\right), \end{array}\right. \tag{6.1}$$

and a right model writes:

$$\left\{\begin{array}{ll} X_{k+1} & = \exp_{\mathcal{G}}^{\wedge}\left(\nu_{k+1}^{q}\right) f(X_k), \\ Y_{k+1} & = \exp_{\mathcal{G}}^{\wedge}\left(\nu_{k+1}^{r}\right) h(X_{k+1}), \end{array}\right. \tag{6.2}$$

where $(\nu_{k+1}^{q}, \nu_{k+1}^{r})$ are centered noise vectors following centered Gaussian distributions:

$$\left\{\begin{array}{ll} \nu_{k+1}^{q} & \sim \mathcal{N}(0, Q_{k+1}), \\ \nu_{k+1}^{r} & \sim \mathcal{N}(0, R_{k+1}), \end{array}\right. \tag{6.3}$$

where Q_{k+1} and R_{k+1} are two definite semi-positive matrices.
The Lie group Extended Kalman filter tracks the expectation and covariance matrix of X_k. The notations of Chapter 5 detailed in (5.7), (5.11) and (5.13) are taken in the sequel.

6.2 . General solution

The solution to the Bayes filter leads to the Extended Kalman Filter on Lie group (LG-EKF), which tracks the temporal evolution of the statistical moments of the prior and posterior densities. The equations derived in the next sections can be summarized as the following algorithm.

Algorithm 7 Lie Group Extended Kalman Filter

Result: $(\hat{\mu}_{k|k}, P_{k|k})$, $k \in [1, N]$
Propagation step:
$\hat{\mu}_{k+1|k} = f(\hat{\mu}_{k|k}, u_{k+1})$
$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_{k+1}$
Update step:
$K_{k+1} = P_{k+1} H_{k+1}^T (H_{k+1} P_{k+1} H_{k+1}^T + R_{k+1})^{-1}$,
$P_{k+1|k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k}$,
Left case: $\hat{\mu}_{k+1|k+1} = \hat{\mu}_{k+1|k} \exp_{\mathcal{G}}^{\wedge} \left(K_{k+1} \log_{\mathcal{H}}^{\vee} \left(Y_{k+1}^{-1} h \left(\hat{\mu}_{k+1|k} \right) \right) \right)$,
Right case: $\hat{\mu}_{k+1|k+1} = \exp_{\mathcal{G}}^{\wedge} \left(K_{k+1} \log_{\mathcal{H}}^{\vee} \left(h \left(\hat{\mu}_{k+1|k} \right) Y_{k+1}^{-1} \right) \right) \hat{\mu}_{k+1|k}$.

The Jacobians F_k and H_{k+1} depend on the side of the Lie group error (either left or right) and are computed as:

- Left case:

$$\begin{aligned} F_k &= \frac{\partial}{\partial \epsilon_{k|k}} \log_{\mathcal{G}}^{\vee} \left(f(\hat{\mu}_{k|k})^{-1} f \left(\hat{\mu}_{k|k} \exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k|k} \right) \right) \right), \\ H_{k+1} &= \frac{\partial}{\partial \epsilon_{k+1|k}} \log_{\mathcal{H}}^{\vee} \left(Y_{k+1}^{-1} h \left(\hat{\mu}_{k+1|k} \exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k+1|k} \right) \right) \right). \end{aligned} \tag{6.4}$$

- Right case:

$$\begin{aligned} F_k &= \frac{\partial}{\partial \epsilon_{k|k}} \log_{\mathcal{G}}^{\vee} \left(f \left(\exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k|k} \right) \hat{\mu}_{k|k} \right) f(\hat{\mu}_k)^{-1} \right), \\ H_{k+1} &= \frac{\partial}{\partial \epsilon_{k+1|k}} \log_{\mathcal{H}}^{\vee} \left(h \left(\exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k+1|k} \right) \hat{\mu}_{k+1|k} \right) Y_{k+1}^{-1} \right). \end{aligned} \tag{6.5}$$

Note that in the specific case where $\mathcal{G} = \mathbb{R}^d$ and $\mathcal{H} = \mathbb{R}^m$, the expressions of the jacobians are equivalent to their definitions in the euclidean space. Indeed, the group law is the customary vector addition $+$ and the groups exponentials and logarithms boil down to the identity map.

In the general case, when $\mathcal{G}$ has no specific structure, the main implementation difficulty of a LG-EKF is the computation of the Jacobians from (6.4) or (6.5). The sequel establishes LG-EKF equations from Optimal Filter derived in Chapter 5.

6.2.1 . Propagation Step

Since the propagation model is Gaussian, the Chapman-Kolmogorov equation (5.4) writes:

$$p(X_{k+1}|Y_k) = \int_{\mathcal{G}} p(X_{k+1}|X_k) \phi_G(X_k; \hat{\mu}_{k|k}, P_{k|k}) \mathrm{d}X_k. \tag{6.6}$$

Proposition 1. *Let $X_k \in \mathcal{G}$ describing the discrete-time sequence either defined by* (6.1) *or* (6.2) *. The propagated density writes:*

$$p(X_{k+1}|Y_k) \approx \phi_{\mathcal{G}}(X_{k+1}; \hat{\mu}_{k+1|k}, P_{k+1|k}), \tag{6.7}$$

where $\hat{\mu}_{k+1|k}$ is the propagated mean:

$$\hat{\mu}_{k+1|k} = f(\hat{\mu}_{k|k}), \tag{6.8}$$

and $P_{k+1|k}$ is the propagated covariance matrix:

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_{k+1}. \tag{6.9}$$

Besides, the Jacobian F_k depends on the side of the error. In the left case, it writes:

$$F_k = \frac{\partial}{\partial \epsilon_{k|k}} \log_{\mathcal{G}}^{\vee} \left(f\left(\hat{\mu}_{k|k}\right)^{-1} f(\hat{\mu}_{k|k} \exp_{\mathcal{G}}^{\wedge}\left(\epsilon_{k|k}\right)) \right). \tag{6.10}$$

And in the right case, it writes:

$$F_k = \frac{\partial}{\partial \epsilon_{k|k}} \log_{\mathcal{G}}^{\vee} \left(f(\exp_{\mathcal{G}}^{\wedge}\left(\epsilon_{k|k}\right) \hat{\mu}_{k|k}) f\left(\hat{\mu}_{k|k}\right)^{-1} \right). \tag{6.11}$$

Proof: Proposition 1 is based on the Chapman-Kolmogorov equation (5.4) assuming a Gaussian process on $\mathcal{G}$. Hence the transition density $p(X_{k+1}|X_k)$ writes:

$$p(X_{k+1}|X_k) = \phi_G(X_{k+1}; f(X_k), Q_{k+1}), \tag{6.12}$$

and the prior density writes:

$$p(X_k|Y_{1:k}) = \phi_{\mathcal{G}}(X_k; \hat{\mu}_{k|k}, P_{k|k}), \tag{6.13}$$

the propagated density (6.6) writes:

$$p(X_{k+1}|Y_{1:k}) = \int_{\mathcal{G}} \phi_G(X_{k+1}; f(X_k), Q_{k+1}) \phi_{\mathcal{G}}(X_k; \hat{\mu}_{k|k}, P_{k|k}) \mathrm{d}X_k. \tag{6.14}$$

The sequel of the proof depends on the side of the error (left or right). Thus, the two cases are separated.

Left Case Linearization

In this development, the left concentrated Gaussian (3.60) is considered:

$$p(X_{k+1}|Y_{1:k}) = \int_{\mathcal{G}} \alpha_{k+1}^{-} \exp\left(-\frac{1}{2} \left\| \begin{matrix} \log_{\mathcal{G}}^{\vee}\left(f(X_k)^{-1} X_{k+1}\right) \\ \log_{\mathcal{G}}^{\vee}\left(\hat{\mu}_{k|k}^{-1} X_k\right) \end{matrix} \right\|_{\Sigma_{k+1}}^{2} \right) \mathrm{d}X_k, \tag{6.15}$$

where α^-_{k+1} is the normalization factor from the products of Gaussian density functions in (6.14):

$$\alpha^-_{k+1} = \left((2\pi)^{2d} \det\left[P_{k|k} Q_{k+1}\right]\right)^{-\frac{1}{2}}, \tag{6.16}$$

and Σ_{k+1} is such that:

$$\Sigma_{k+1} = \begin{pmatrix} Q_{k+1} & 0 \\ 0 & P_{k|k} \end{pmatrix}. \tag{6.17}$$

Focusing on the transition term, considering that $X_k = \hat{\mu}_{k|k} \exp^{\wedge}_{\mathcal{G}}(\epsilon_{k|k})$:

$$f(X_k)^{-1} X_{k+1} = f\left(\hat{\mu}_{k|k} \exp^{\wedge}_{\mathcal{G}}(\epsilon_{k|k})\right)^{-1} f(\hat{\mu}_{k|k}) \exp^{\wedge}_{\mathcal{G}}(\epsilon_{k+1|k}). \tag{6.18}$$

Therefore the linearization of the logarithm of the latter expression with respect to $\epsilon_{k|k}$ and $\epsilon_{k+1|k}$ at the linearization point $\epsilon_{k|k} = \epsilon_{k+1|k} = 0$ gives:

$$\begin{aligned} &\log^{\vee}_{\mathcal{G}}\left(f\left(\hat{\mu}_{k|k} \exp^{\wedge}_{\mathcal{G}}(\epsilon_{k|k})\right)^{-1} f(\hat{\mu}_{k|k}) \exp^{\wedge}_{\mathcal{G}}(\epsilon_{k+1|k})\right) \\ &\qquad = \epsilon_{k+1|k} - F_k \epsilon_{k|k} + O(\|\epsilon_{k|k}\|_2^2, \|\epsilon_{k+1|k}\|_2^2), \end{aligned} \tag{6.19}$$

where F_k is a Jacobian:

$$F_k \triangleq \frac{\partial}{\partial \epsilon_{k|k}} \log^{\vee}_{\mathcal{G}}\left(f(\hat{\mu}_{k|k})^{-1} f\left(\hat{\mu}_{k|k} \exp^{\wedge}_{\mathcal{G}}(\epsilon_{k|k})\right)\right). \tag{6.20}$$

The change from the Haar measure dX_k to the Lebesgue measure $d\epsilon_{k|k}$ can now be applied according to (3.44) with the previous linearization:

$$p(X_{k+1}|Y_{1:k}) = \int_{\mathbb{R}^d} \alpha^-_{k+1} \exp\left(-\frac{1}{2}\left\|\begin{bmatrix} I & -F_k \\ 0 & I \end{bmatrix}\begin{bmatrix} \epsilon_{k+1|k} \\ \epsilon_{k|k} \end{bmatrix}\right\|^2_{\Sigma_{k+1}}\right) \left|\det\left[\Phi_{\mathcal{G}}(\epsilon_{k|k})\right]\right| d\epsilon_{k|k}. \tag{6.21}$$

Right Case Linearization

The proof of the right case unfolds in the same way. Considering a right Gaussian on the group (3.63):

$$p(X_{k+1}|Y_{1:k}) = \int_{\mathcal{G}} \alpha^-_{k+1} \exp\left(-\frac{1}{2}\left\|\begin{matrix} \log^{\vee}_{\mathcal{G}}\left(X_{k+1} f(X_k)^{-1}\right) \\ \log^{\vee}_{\mathcal{G}}\left(X_k \hat{\mu}^{-1}_{k|k}\right) \end{matrix}\right\|^2_{\Sigma_{k+1}}\right) \mathrm{d}X_k. \tag{6.22}$$

The transition term in the logarithm gives:

$$X_{k+1} f(X_k)^{-1} \quad = \exp^{\wedge}_{\mathcal{G}}(\epsilon_{k+1|k})\, f(\hat{\mu}_{k|k}) f\left(\exp^{\wedge}_{\mathcal{G}}(\epsilon_{k|k})\, \hat{\mu}_{k|k}\right)^{-1}. \tag{6.23}$$

Proceeding with the same linearization as (6.19), and the change of measure gives the same result as (6.21) where F_k is the (right) propagation Jacobian defined as:

$$F_k \triangleq \frac{\partial}{\partial \epsilon_{k|k}} \log^{\vee}_{\mathcal{G}}\left(f\left(\hat{\mu}_{k|k} \exp^{\wedge}_{\mathcal{G}}(\epsilon_{k|k})\right) f(\hat{\mu}_{k|k})^{-1}\right). \tag{6.24}$$

Note that this change of measure is the same whether $\epsilon_{k|k}$ is defined with a right or a left error with the invariance property of the measure for unimodular Lie groups [27].

Propagated Mean and covariance matrix

Starting from this point, the proof is the same for the left and right case. Applying the Mahalanobis norm on (6.21) and the inversion of the matrix with the Schur complement gives:

$$p(X_{k+1}|Y_{1:k}) = \int_{\mathbb{R}^d} \alpha^-_{k+1} \exp\left(-\frac{1}{2}\left\|\begin{bmatrix}\epsilon_{k+1|k}\\ \epsilon_{k|k}\end{bmatrix}\right\|^2_{M_{k+1}}\right) \left|\det\left[\Phi_{\mathcal{G}}(\epsilon_{k|k})\right]\right| \mathrm{d}\epsilon_{k|k}, \tag{6.25}$$

where M_{k+1} inverse is:

$$M^{-1}_{k+1} = \begin{bmatrix} Q^{-1}_{k+1} & -Q^{-1}_{k+1}F_k \\ -F_k^T Q^{-1}_{k+1} & P^{-1}_{k|k} + F_k^T Q^{-1}_{k+1} F_k \end{bmatrix}. \tag{6.26}$$

The sequel of the proof computes M and its determinant. By applying the matrix inversion lemma detailed in Appendix B, (6.26) gives:

$$M_{k+1} = \begin{bmatrix} F_k P_{k|k} F_k^T + Q_{k+1} & F_k P_{k|k} \\ P_{k|k} F_k^T & P_{k|k} \end{bmatrix}, \tag{6.27}$$

and the determinant is:

$$\begin{aligned} \det M_{k+1} &= \det \begin{bmatrix} F_k P_{k|k} F_k^T + Q_{k+1} & F_k P_{k|k} \\ P_{k|k} F_k^T & P_{k|k} \end{bmatrix}, \\ &= \det\left(\begin{bmatrix} I & F_k \\ 0 & I \end{bmatrix} \begin{bmatrix} Q_{k+1} & 0 \\ 0 & P_{k|k} \end{bmatrix} \begin{bmatrix} I & 0 \\ F_k^T & I \end{bmatrix}\right), \\ &= \det \begin{bmatrix} Q_{k+1} & 0 \\ 0 & P_{k|k} \end{bmatrix}, \\ &= \det Q_{k+1} \det P_{k|k}. \end{aligned} \tag{6.28}$$

Hence, the normalization factor α^-_{k+1} from (6.16) also writes:

$$\alpha^-_{k+1} = \left((2\pi)^{2d} \det M_{k+1}\right)^{-\frac{1}{2}}. \tag{6.29}$$

Hence, the prior density writes:

$$p(X_{k+1}|Y_{1:k}) = \int_{\mathbb{R}^d} p(\epsilon_{k+1|k}, \epsilon_{k|k}) \left|\det\left[\Phi_{\mathcal{G}}(\epsilon_{k|k})\right]\right| \mathrm{d}\epsilon_{k|k}, \tag{6.30}$$

where $p(\epsilon_{k+1|k}, \epsilon_{k|k})$ is a Gaussian density such that:

$$p(\epsilon_{k+1|k}, \epsilon_{k|k}) \sim \mathcal{N}\left(\begin{bmatrix}\epsilon_{k+1|k}\\ \epsilon_{k|k}\end{bmatrix}; \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix} P_{k+1|k} & P_{k|k}F_k \\ F_k^T P_{k|k} & P_{k|k} \end{bmatrix}\right), \tag{6.31}$$

where the propagated matrix is defined as $P_{k+1|k} \triangleq F_k P_{k|k} F_k^T + Q_{k+1}$. This probability density can be factorized with the conditional probabilities formula:

$$p(\epsilon_{k|k}, \epsilon_{k+1|k}) = p(\epsilon_{k|k}|\epsilon_{k+1|k}) p(\epsilon_{k+1|k}), \tag{6.32}$$

and factorizing the integral of (6.30) gives:

$$p(X_{k+1}|Y_{1:k}) = p(\epsilon_{k+1|k}) \int_{\mathbb{R}^d} p(\epsilon_{k|k}|\epsilon_{k+1|k}) \left|\det\left[\Phi_{\mathcal{G}}(\epsilon_{k|k})\right]\right| \mathrm{d}\epsilon_{k|k}, \tag{6.33}$$

and the integral equals to unity as $p(\epsilon_{k|k}|\epsilon_{k+1|k})$ is a concentrated Gaussian for $\epsilon_{k|k}$:

$$\int_{\mathbb{R}^d} p(\epsilon_{k|k}|\epsilon_{k+1|k}) \left|\det\left[\Phi_{\mathcal{G}}(\epsilon_{k|k})\right]\right| \mathrm{d}\epsilon_{k|k} = 1. \tag{6.34}$$

Eventually, the Chapman-Kolmogorov equation boils down to:

$$p(X_{k+1}|Y_{1:k}) = p(\epsilon_{k+1|k}), \tag{6.35}$$

where $p(\epsilon_{k+1|k})$ denotes either a left or right concentrated Gaussian on $\mathcal{G}$, which mean is given by $\hat{\mu}_{k+1|k} = f(\hat{\mu}_{k|k})$ and covariance matrix $P_{k+1|k} = F_k P_{k|k} F_k^T + Q_{k+1}$. □.

6.2.2 . Update Step

The Kalman update is derived by applying the Bayes rule to the measurement model of (6.1) (6.2) and verifies the following proposition.

Proposition 2. *Let $X_{k+1} \in \mathcal{G}$ and $Y_{k+1} \in \mathcal{H}$ two random matrices describing the discrete-time sequence either defined by the state-space model* (6.1) *or* (6.2)*. Then, the posterior density is:*

$$p(X_{k+1}|Y_{k+1}) \approx \phi_{\mathcal{G}}(X_{k+1}; \hat{\mu}_{k+1|k+1}, P_{k+1|k+1}), \tag{6.36}$$

where the updated covariance matrix $P_{k+1|k+1}$ writes:

$$\begin{array}{rl} K_{k+1} & = P_{k+1|k} H_{k+1}^T (H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1})^{-1}, \\ P_{k+1|k+1} & = (I - K_{k+1} H_{k+1}) P_{k+1|k}. \end{array} \tag{6.37}$$

The Jacobian H_{k+1} depends on the side of the error. In the left case it writes:

$$H_{k+1} = \frac{\partial}{\partial \epsilon} \log_{\mathcal{H}}^{\vee} \left(Y_{k+1}^{-1} h \left(\hat{\mu}_{k+1|k} \exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k+1|k} \right) \right) \right). \tag{6.38}$$

And in the right case:

$$H_{k+1} = \frac{\partial}{\partial \epsilon} \log_{\mathcal{H}}^{\vee} \left(h \left(\exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k+1|k} \right) \hat{\mu}_{k+1|k} \right) Y_{k+1}^{-1} \right). \tag{6.39}$$

The updated mean also depends on the side of the error. In the left case:

$$\begin{array}{rl} \xi_{k+1} & = \log_{\mathcal{H}}^{\vee} \left(Y_{k+1}^{-1} h \left(\hat{\mu}_{k+1|k} \right) \right) \\ \hat{\mu}_{k+1|k+1} & = \hat{\mu}_{k+1|k} \exp_{\mathcal{G}}^{\wedge} \left(K_{k+1} \xi_{k+1} \right), \end{array} \tag{6.40}$$

and in the right case:

$$\begin{aligned} \xi_{k+1} &= \log_{\mathcal{H}}^{\vee}\left(h\left(\hat{\mu}_{k+1|k}\right) Y_{k+1}^{-1}\right) \\ \hat{\mu}_{k+1|k+1} &= \exp_{G}^{\wedge}(K_{k+1}\xi_{k+1})\hat{\mu}_{k+1|k}, \end{aligned} \tag{6.41}$$

Proof. The Bayes equation writes:

$$p(X_{k+1}|Y_{1:k+1}) = \frac{p(Y_{k+1}|X_{k+1})p(X_{k+1}|Y_{1:k})}{p(Y_{k+1}|Y_{1:k})}, \tag{6.42}$$

where the prior density and the likelihood are concentrated Gaussians on two unimodular Lie groups $\mathcal{G}$ and $\mathcal{H}$:

$$p(X_{k+1}|Y_{1:k}) = \phi_{\mathcal{G}}\left(X_{k+1}; \hat{\mu}_{k+1|k}, P_{k+1|k}\right), \tag{6.43}$$

$$p(Y_{k+1}|X_{k+1}) = \phi_{\mathcal{H}}\left(Y_{k+1}; h(X_{k+1}), R_{k+1}\right). \tag{6.44}$$

For the sake of clarity, this proof unfolds in 5 distinct steps aiming to derive the expression of the posterior density $p(X_{k+1}|Y_{1:k+1})$, which leads to the equations of LG-EKF. The first step details the linearization of the measurement model in the left and right case. The second step derives the update equation of the covariance matrix with the Kalman gain. The third step calculates the marginalized likelihood of the Bayes equation (5.5), leading to the expression of the posterior density detailed in the fourth step. The fifth step gives the update equation of the mean in the case of a left or right problem.

1- Linearization of the Measurement Model

This step of the proof applies (6.43) and (6.44) to the Bayes equation (5.5) and linearizes the measurement model.

(Left case) In the case of a left concentrated Gaussian density on Lie groups, the posterior density writes:

$$p(X_{k+1}|Y_{1:k+1}) = \alpha_{k+1}^{+} \exp\left(-\frac{1}{2}\left\| \begin{matrix} \log_{\mathcal{H}}^{\vee}\left(h\left(X_{k+1}\right)^{-1} Y_{k+1}\right) \\ \log_{\mathcal{G}}^{\vee}\left(\hat{\mu}_{k+1|k}^{-1} X_{k+1}\right) \end{matrix} \right\|_{\Lambda_{k+1}}^{2}\right), \tag{6.45}$$

where:

$$\Lambda_{k+1} = \begin{bmatrix} R_{k+1} & 0 \\ 0 & P_{k+1|k} \end{bmatrix}, \tag{6.46}$$

and the normalization constant α_{k+1}^{+} is the product of the normalization constants from the Gaussian density functions of (6.43) and (6.44):

$$\alpha_{k+1}^{+} = \left((2\pi)^{d+m} \det R_{k+1} \det P_{k+1|k}\right)^{-\frac{1}{2}}. \tag{6.47}$$

Then, given that a left random variable on $\mathcal{G}$ writes:

$$X_{k+1} = \hat{\mu}_{k+1|k} \exp_{\mathcal{G}}^{\wedge}\left(\epsilon_{k+1|k}\right), \tag{6.48}$$

the term $\log_{\mathcal{H}}^{\vee}\left(h\left(X_{k+1}\right)^{-1}Y_{k+1}\right)$ from (6.45) can be linearized such that:

$$\log_{\mathcal{H}}^{\vee}\left(h\left(\hat{\mu}_{k+1|k}\exp_{\mathcal{G}}^{\wedge}\left(\epsilon_{k+1|k}\right)\right)^{-1}Y_{k+1}\right)=z_{k+1}-H_{k+1}\epsilon_{k+1|k}+O(\|\epsilon_{k+1|k}\|^2), \tag{6.49}$$

where z_{k+1} is the innovation:

$$z_{k+1}=\log_{\mathcal{H}}^{\vee}\left(h\left(\hat{\mu}_{k+1|k}\right)^{-1}Y_{k+1}\right), \tag{6.50}$$

and H_{k+1} is the measurement Jacobian:

$$H_{k+1}=\frac{\partial}{\partial\epsilon_{k+1|k}}\log_{\mathcal{H}}^{\vee}\left(Y_{k+1}^{-1}h\left(\hat{\mu}_{k+1|k}\exp_{\mathcal{G}}^{\wedge}\left(\epsilon_{k+1|k}\right)\right)\right). \tag{6.51}$$

(Right case) In the case of a right concentrated Gaussian density, the posterior density is:

$$p(X_{k+1}|Y_{1:k+1})=\alpha_{k+1}^{+}\exp\left(-\frac{1}{2}\left\|\begin{matrix}\log_{\mathcal{H}}^{\vee}\left(Y_{k+1}h(X_{k+1})^{-1}\right)\\ \log_{\mathcal{G}}^{\vee}\left(X_{k+1}\hat{\mu}_{k+1|k}^{-1}\right)\end{matrix}\right\|_{\Lambda_{k+1}}^{2}\right), \tag{6.52}$$

By applying the same principle as for the left case, and given that a right random variable writes:

$$X_{k+1}=\exp_{\mathcal{G}}^{\wedge}\left(\epsilon_{k+1|k}\right)\hat{\mu}_{k+1|k}, \tag{6.53}$$

the expression can be linearized as:

$$\log_{\mathcal{H}}^{\vee}\left(Y_{k+1}h(X_{k+1})^{-1}\right)=z_{k+1}-H_{k+1}\epsilon_{k+1|k}+O(\|\epsilon_{k+1|k}\|^2), \tag{6.54}$$

where H_{k+1} is the right measurement Jacobian:

$$H_{k+1}=\frac{\partial}{\partial\epsilon_{k+1|k}}\log_{\mathcal{H}}^{\vee}\left(h\left(\exp_{\mathcal{G}}^{\wedge}\left(\epsilon_{k+1|k}\right)\hat{\mu}_{k+1|k}\right)Y_{k+1}^{-1}\right), \tag{6.55}$$

and the innovation z_{k+1} is defined by:

$$z_{k+1}=\log_{\mathcal{H}}^{\vee}\left(Y_{k+1}h\left(\hat{\mu}_{k+1|k}\right)^{-1}\right). \tag{6.56}$$

2- Kalman Gain and Covariance Matrix Update Equations

The sequel of the proof establishes the gain and covariance matrix update expressions, which are the same in the left and the right case. Given the previous linearized expressions (6.49) and (6.54), the posterior density writes:

$$p(X_{k+1}|Y_{1:k+1})=\alpha_{k+1}^{+}\exp\left(-\frac{1}{2}\left\|\begin{bmatrix}I & -H_{k+1}\\ 0 & I\end{bmatrix}\begin{bmatrix}z_{k+1}\\ \epsilon_{k+1|k}\end{bmatrix}\right\|_{\Lambda_{k+1}}^{2}\right). \tag{6.57}$$

Developing the square Mahalanobis norm in the exponential of (6.57) gives:

$$\left\|\begin{bmatrix}I & -H_{k+1}\\ 0 & I\end{bmatrix}\begin{bmatrix}z_{k+1}\\ \epsilon_{k+1|k}\end{bmatrix}\right\|_{\Lambda_{k+1}}^{2}=\begin{bmatrix}z_{k+1}\\ \epsilon_{k+1|k}\end{bmatrix}^{T}\begin{bmatrix}R_{k+1}^{-1} & -R_{k+1}^{-1}H_{k+1}\\ -H_{k+1}^{T}R_{k+1}^{-1} & P_{k+1|k}^{-1}+H_{k+1}^{T}R_{k+1}^{-1}H_{k+1}\end{bmatrix}\begin{bmatrix}z_{k+1}\\ \epsilon_{k+1|k}\end{bmatrix}. \tag{6.58}$$

The latter matrix can be inverted with (B.11):

$$\begin{bmatrix} R_{k+1}^{-1} & -R_{k+1}^{-1}H_{k+1} \\ -H_{k+1}^T R_{k+1}^{-1} & P_{k+1|k}^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \end{bmatrix}^{-1} = \begin{bmatrix} S_{k+1} & H_{k+1}P_{k+1|k} \\ P_{k+1|k}H_{k+1}^T & P_{k+1|k} \end{bmatrix}, \tag{6.59}$$

where $S_{k+1} \triangleq R_{k+1} + H_{k+1}P_{k+1|k}H_{k+1}^T$ is obtained with the Woodburry identity (B.12):

$$\left(R_{k+1}^{-1} + R_{k+1}^{-1}H_{k+1}(P_{k+1|k}^{-1} + H_{k+1}^T R_{k+1}^{-1}H_{k+1})^{-1}H_{k+1}^T R_{k+1}^{-1}\right)^{-1} = R_{k+1} + H_{k+1}P_{k+1|k}H_{k+1}^T. \tag{6.60}$$

Using the matrix factorization (B.3) from Appendix B, the latter expression gives:

$$\begin{bmatrix} S_{k+1} & H_{k+1}P_{k+1|k} \\ P_{k+1|k}H_{k+1}^T & P_{k+1|k} \end{bmatrix} = \begin{bmatrix} I & 0 \\ K_{k+1} & I \end{bmatrix} \begin{bmatrix} S_{k+1} & 0 \\ 0 & P_{k+1|k+1} \end{bmatrix} \begin{bmatrix} I & K_{k+1}^T \\ 0 & I \end{bmatrix}, \tag{6.61}$$

where $K_{k+1} \triangleq P_{k+1|k}H_{k+1}^T S_{k+1}^{-1}$ is commonly known as the Kalman gain, and the covariance matrix $P_{k+1|k+1} = P_{k+1|k} - K_{k+1}H_{k+1}P_{k+1|k}$ is the updated covariance matrix. The inverse of the expression (6.61) with the inversion formulas (B.13) and (B.14) gives:

$$\begin{bmatrix} S_{k+1} & H_{k+1}P_{k+1|k} \\ P_{k+1|k}H_{k+1}^T & P_{k+1|k} \end{bmatrix}^{-1} = \begin{bmatrix} I & -K_{k+1}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} S_{k+1}^{-1} & 0 \\ 0 & P_{k+1|k+1}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -K_{k+1} & I \end{bmatrix}. \tag{6.62}$$

Thus, the calculus from (6.58) to (6.62) lead to:

$$\left\| \begin{bmatrix} I & -H_{k+1} \\ 0 & I \end{bmatrix} \begin{bmatrix} z_{k+1} \\ \epsilon_{k+1|k} \end{bmatrix} \right\|_{\Lambda_{k+1}}^2 = \left\| \begin{bmatrix} I & 0 \\ -K_{k+1} & I \end{bmatrix} \begin{bmatrix} z_{k+1} \\ \epsilon_{k+1|k} \end{bmatrix} \right\|_{\Upsilon_{k+1}}^2, \tag{6.63}$$

where Υ_{k+1} writes:

$$\Upsilon_{k+1} = \begin{bmatrix} S_{k+1} & 0 \\ 0 & P_{k+1|k+1} \end{bmatrix}. \tag{6.64}$$

Thus, the product of the likelihood and prior densities is such that:

$$p(Y_{k+1}|X_{k+1})p(X_{k+1}|Y_{1:k}) \approx \alpha_{k+1}^{+} \exp\left(\left\| \begin{bmatrix} z_{k+1} \\ \epsilon_{k+1|k} - K_{k+1}z_{k+1} \end{bmatrix} \right\|_{\Upsilon_{k+1}}^2 \right), \tag{6.65}$$

where $(K_{k+1}, P_{k+1|k+1}, S_{k+1})$ are the customary matrices of the Kalman filter update:

$$S_{k+1} = H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}, \tag{6.66}$$

$$K_{k+1} = P_{k+1}H_{k+1}^T S_{k+1}^{-1}, \tag{6.67}$$

$$P_{k+1|k+1} = (I - K_{k+1}H_{k+1})P_{k+1|k}. \tag{6.68}$$

3- Calculus of the normalization term

The normalization density $p(Y_{k+1}|Y_{1:k})$ can be marginalized with respect to X_{k+1}:

$$p(Y_{k+1}|Y_{1:k}) = \int_{\mathcal{G}} p(Y_{k+1}|X_{k+1})p(X_{k+1}|Y_{1:k})\mathrm{d}X_{k+1}. \tag{6.69}$$

According to (6.65), and by applying the change of variable detailed in Chapter 3:

$$dX_{k+1} = \left|\det \Phi_{\mathcal{G}}(\epsilon_{k+1|k})\right| \mathrm{d}\epsilon_{k+1|k}, \tag{6.70}$$

this marginalization is approximated by:

$$p(Y_{k+1}|Y_{1:k}) \approx \int_{\mathbb{R}^d} \alpha_{k+1}^{+} \exp\left(-\frac{1}{2}\left\|\begin{bmatrix} z_{k+1} \\ \epsilon_{k+1|k} - K_{k+1}z_{k+1}\end{bmatrix}\right\|_{\Upsilon_{k+1}}^2\right) \left|\det \Phi_{\mathcal{G}}(\epsilon_{k+1|k})\right| \mathrm{d}\epsilon_{k+1|k}, \tag{6.71}$$

which can be split in two parts as the innovation term does not depend on $\epsilon_{k+1|k}$:

$$\begin{aligned} p(Y_{k+1}|Y_{1:k}) \approx \alpha_{k+1}^{+} \exp\left(-\frac{1}{2}\left\|z_{k+1}\right\|_{S_{k+1}}^2\right) \\ \int_{\mathbb{R}^d} \exp\left(-\frac{1}{2}\left\|\epsilon_{k+1|k} - K_{k+1}z_{k+1}\right\|_{P_{k+1|k+1}}^2\right) \left|\det \Phi_{\mathcal{G}}(\epsilon_{k+1|k})\right| \mathrm{d}\epsilon_{k+1|k}, \end{aligned} \tag{6.72}$$

The normalization constant α_{k+1}^{+} defined in (6.47) can be expressed with matrices S_{k+1} and $P_{k+1|k+1}$. Since the determinant of the block diagonal matrices is equal to unity (see Appendix B) the following determinant product writes:

$$\begin{aligned} \det R_{k+1}^{-1} \det P_{k+1|k}^{-1} &= \det \begin{bmatrix} R_{k+1}^{-1} & 0 \\ 0 & P_{k+1|k}^{-1} \end{bmatrix}, \\ &= \det \left(\begin{bmatrix} I & 0 \\ -H_{k+1}^T & I \end{bmatrix} \begin{bmatrix} R_{k+1}^{-1} & 0 \\ 0 & P_{k+1|k}^{-1} \end{bmatrix} \begin{bmatrix} I & -H_{k+1} \\ 0 & I \end{bmatrix} \right), \\ &= \det \begin{bmatrix} R_{k+1}^{-1} & -R_{k+1}^{-1}H_{k+1} \\ -H_{k+1}^T R_{k+1}^{-1} & P_{k+1}^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \end{bmatrix}. \end{aligned} \tag{6.73}$$

The developments from (6.58) to (6.62) give:

$$\begin{bmatrix} R_{k+1}^{-1} & -R_{k+1}^{-1}H_{k+1} \\ -H_{k+1}^T R_{k+1}^{-1} & P_{k+1}^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \end{bmatrix} = \begin{bmatrix} I & -K_{k+1}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} S_{k+1}^{-1} & 0 \\ 0 & P_{k+1|k+1}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -K_{k+1} & I \end{bmatrix}, \tag{6.74}$$

which leads to:

$$\begin{aligned} \det R_{k+1}^{-1} \det P_{k+1|k}^{-1} &= \det \left(\begin{bmatrix} I & -K_{k+1}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} S_{k+1}^{-1} & 0 \\ 0 & P_{k+1|k+1}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -K_{k+1} & I \end{bmatrix} \right), \\ &= \det S_{k+1}^{-1} \det P_{k+1|k+1}^{-1}. \end{aligned} \tag{6.75}$$

Thus, the term α_{k+1}^{+} also writes:

$$\alpha_{k+1}^{+} = \left((2\pi)^{d+m} \det S_{k+1} \det P_{k+1|k+1}\right)^{-\frac{1}{2}}. \tag{6.76}$$

Applying this expression to (6.71) and splitting the factors to the appropriate terms leads to:

$$p(Y_{k+1}|Y_{1:k}) \approx \frac{1}{\sqrt{(2\pi)^m \det S_{k+1}}} \exp\left(-\frac{1}{2}\left\|z_{k+1}\right\|^2_{S_{k+1}}\right) \int\limits_{\mathbb{R}^d} \frac{1}{\sqrt{(2\pi)^d \det P_{k+1|k+1}}} \exp\left(-\frac{1}{2}\left\|\epsilon_{k+1|k} - K_{k+1}z_{k+1}\right\|^2_{P_{k+1|k+1}}\right) \left|\det \Phi_{\mathcal{G}}(\epsilon_{k+1|k})\right| \mathrm{d}\epsilon_{k+1|k}. \tag{6.77}$$

The integral term of the latter expression equals to unity since it is the integral of a concentrated Gaussian density on $\mathcal{G}$:

$$\int\limits_{\mathbb{R}^d} \frac{1}{\sqrt{(2\pi)^d \det P_{k+1|k+1}}} \exp\left(-\frac{1}{2}\left\|\epsilon_{k+1|k} - K_{k+1}z_{k+1}\right\|^2_{P_{k+1|k+1}}\right) \left|\det \Phi_{\mathcal{G}}(\epsilon_{k+1|k})\right| \mathrm{d}\epsilon_{k+1|k} = 1. \tag{6.78}$$

Eventually, the Bayes rule normalization term writes:

$$p(Y_{k+1}|Y_{1:k}) \approx \frac{1}{\sqrt{(2\pi)^m \det S_{k+1}}} \exp\left(-\frac{1}{2}\left\|z_{k+1}\right\|^2_{S_{k+1}}\right). \tag{6.79}$$

4- Expression of the Posterior Density

Then, by applying the results from (6.79) and (6.65) to the Bayes rule (5.5), and using the expression of the constant α^+_{k+1} from (6.47) gives:

$$p(X_{k+1}|Y_{1:k+1}) \approx \frac{\frac{1}{\sqrt{(2\pi)^{d+m} \det R_{k+1} \det P_{k+1|k}}} \exp\left(-\frac{1}{2}\left\|\begin{bmatrix} z_{k+1} \\ \epsilon_{k+1|k} - K_{k+1}z_{k+1}\end{bmatrix}\right\|^2_{\Upsilon_{k+1}}\right)}{\frac{1}{\sqrt{(2\pi)^m \det S_{k+1}}} \exp\left(-\frac{1}{2}\left\|z_{k+1}\right\|^2_{S_{k+1}}\right)}, \tag{6.80}$$

which writes:

$$p(X_{k+1}|Y_{1:k+1}) \approx \frac{\sqrt{(2\pi)^m \det S_{k+1}}}{\sqrt{(2\pi)^{d+m} \det R_{k+1} \det P_{k+1|k}}} \exp\left(-\frac{1}{2}\left\|\begin{bmatrix} z_{k+1} \\ \epsilon_{k+1|k} - K_{k+1}z_{k+1}\end{bmatrix}\right\|^2_{\Upsilon_{k+1}} + \frac{1}{2}\left\|z_{k+1}\right\|^2_{S_{k+1}}\right). \tag{6.81}$$

Focusing on the multiplicative term of (6.81), the equality (6.75) leads to the simplification:

$$\frac{\sqrt{(2\pi)^m \det S_{k+1}}}{\sqrt{(2\pi)^{d+m} \det R_{k+1} \det P_{k+1|k}}} = \frac{1}{\sqrt{(2\pi)^d \det P_{k+1|k+1}}}. \tag{6.82}$$

The exponential term of (6.81), can be written as following:

$$\begin{aligned}
&\exp\left(-\frac{1}{2}\left\|\begin{bmatrix} z_{k+1} \\ \epsilon_{k+1|k} - K_{k+1}z_{k+1}\end{bmatrix}\right\|^2_{\Upsilon_{k+1}} + \frac{1}{2}\left\|z_{k+1}\right\|^2_{S_{k+1}}\right) \\
&\quad = \exp\left(-\frac{1}{2}\left\|z_{k+1}\right\|^2_{R_{k+1}}\right)\exp\left(-\frac{1}{2}\left\|\epsilon_{k+1|k} - K_{k+1}z_{k+1}\right\|^2_{P_{k+1|k}}\right)\exp\left(+\frac{1}{2}\left\|z_{k+1}\right\|^2_{R_{k+1}}\right), \\
&\quad = \exp\left(-\frac{1}{2}\left\|\epsilon_{k+1|k} - K_{k+1}z_{k+1}\right\|^2_{P_{k+1|k}}\right).
\end{aligned} \tag{6.83}$$

Combining the results of (6.82) and (6.83) to (6.81):

$$p(X_{k+1}|Y_{1:k+1}) \approx \frac{1}{\sqrt{(2\pi)^d \det P_{k+1|k+1}}}\exp\left(-\frac{1}{2}\left\|\epsilon_{k+1|k+1}\right\|^2_{P_{k+1|k+1}}\right). \tag{6.84}$$

Thus, the updated density is a Gaussian on $\mathcal{G}$ of covariance matrix $P_{k+1|k+1}$ where the updated log-Euclidean error is:

$$\epsilon_{k+1|k+1} = \epsilon_{k+1|k} - K_{k+1}z_{k+1}. \tag{6.85}$$

5- Mean Update Equation

(Left case) In the left case, the updated log-Euclidean error writes:

$$\epsilon_{k+1|k+1} = \log^\vee_{\mathcal{G}}\left(\hat{\mu}^{-1}_{k+1|k+1}X_{k+1}\right), \tag{6.86}$$

$$= \log^\vee_{\mathcal{G}}\left(\hat{\mu}^{-1}_{k+1|k+1}\hat{\mu}_{k+1|k}\exp^\wedge_{\mathcal{G}}\left(\epsilon_{k+1|k}\right)\right), \tag{6.87}$$

and from (6.85):

$$\log^\vee_{\mathcal{G}}\left(\hat{\mu}^{-1}_{k+1|k+1}\hat{\mu}_{k+1|k}\exp^\wedge_{\mathcal{G}}\left(\epsilon_{k+1|k}\right)\right) = \epsilon_{k+1|k} - K_{k+1}z_{k+1}, \tag{6.88}$$

which gives:

$$\hat{\mu}^{-1}_{k+1|k+1}\hat{\mu}_{k+1|k}\exp^\wedge_{\mathcal{G}}\left(\epsilon_{k+1|k}\right) = \exp^\wedge_{\mathcal{G}}\left(\epsilon_{k+1|k} - K_{k+1}z_{k+1}\right). \tag{6.89}$$

Then, by isolating the term $\hat{\mu}^{-1}_{k+1|k+1}$ on the left side and by inverting both sides of the latter equation:

$$\hat{\mu}_{k+1|k+1} = \hat{\mu}_{k+1|k}\exp^\wedge_{\mathcal{G}}\left(-\epsilon_{k+1|k} + K_{k+1}z_{k+1}\right)\exp^\wedge_{\mathcal{G}}\left(\epsilon_{k+1|k}\right), \tag{6.90}$$

$$\hat{\mu}_{k+1|k+1} = \hat{\mu}_{k+1|k}\exp^\wedge_{\mathcal{G}}\left(\mathrm{BCH}(-\epsilon_{k+1|k} + K_{k+1}z_{k+1}, \epsilon_{k+1|k})\right). \tag{6.91}$$

To conclude the proof and to obtain the form described in Proposition 2, the BCH terms have to be applied from (3.38):

$$\hat{\mu}_{k+1|k+1} \approx \hat{\mu}_{k+1|k}\exp^\wedge_{\mathcal{G}}\left(\Phi_{\mathcal{G}}(-\epsilon_{k+1|k})K_{k+1}z_{k+1}\right). \tag{6.92}$$

Finally, it is assumed that the covariance matrix is sufficiently small, and the Lie group Jacobian can be approximated by the identity. This zero-order approximation is suitable to all the studied scenarios:

$$\hat{\mu}_{k+1|k+1} \approx \hat{\mu}_{k+1|k} \exp_{\mathcal{G}}^{\wedge} \left(K_{k+1} z_{k+1} \right). \tag{6.93}$$

(Right case) In the right case, the error equation writes:

$$\epsilon_{k+1|k+1} = \log_{\mathcal{G}}^{\vee} \left(X_{k+1} \hat{\mu}_{k+1|k+1}^{-1} \right), \tag{6.94}$$

$$= \log_{\mathcal{G}}^{\vee} \left(\exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k+1|k} \right) \hat{\mu}_{k+1|k} \hat{\mu}_{k+1|k+1}^{-1} \right), \tag{6.95}$$

Following an approach similar to the left case gives:

$$\hat{\mu}_{k+1|k+1} = \exp_{\mathcal{G}}^{\wedge} \left(-\epsilon_{k+1|k} + K_{k+1} z_{k+1} \right) \exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k+1|k} \right) \hat{\mu}_{k+1|k}. \tag{6.96}$$

The introduction of BCH formula gives:

$$\hat{\mu}_{k+1|k+1} = \exp_{\mathcal{G}}^{\wedge} \left(\mathrm{BCH}(-\epsilon_{k+1|k} + K_{k+1} z_{k+1}, \epsilon_{k+1|k}) \right) \hat{\mu}_{k+1|k}, \tag{6.97}$$

and the zero-order approximation of BCH gives the desired formula:

$$\hat{\mu}_{k+1|k+1} \approx \exp_{\mathcal{G}}^{\wedge} \left(K_{k+1} z_{k+1} \right) \hat{\mu}_{k+1|k}. \tag{6.98}$$

□

6.3 . Jacobians on the Special Euclidean Group

In practice, filters on Lie groups often use specific structures for the state matrices depending on the estimation problem. This section focuses on the Special Euclidean group ($\mathrm{SE}_p(3)$) which is broadly used in robotics and signal processing problems. The sequel introduces substantial simplifications in the expressions of the propagation Jacobians (6.4) and (6.5). For the sake of simplicity, the temporal index k is overlooked in this section. T

6.3.1 . Expression of the Jacobians

Let $X \in \mathrm{SE}_p(3)$ be a random state matrix and $f : \mathrm{SE}_p(3) \to \mathrm{SE}_p(3)$ be a smooth function. Since f is in $\mathrm{SE}_p(3)$ it can be separated in several components such that:

$$f(X) = \begin{bmatrix} f_C(X) & f_1(X) \cdots f_p(X) \\ 0_{p,3} & I_p \end{bmatrix}, \tag{6.99}$$

where $f_C : \mathrm{SE}_p(3) \to \mathrm{SO}(3)$ is the component of f for the rotation and $f_i : \mathrm{SE}_p(3) \to \mathbb{R}^3, i \in [1, p]$ are the components of f for the vector variables. Recall that in the left case $X = \mu \exp_{\mathcal{G}}^{\wedge} (\epsilon)$ and in the right case $X = \exp_{\mathcal{G}}^{\wedge} (\epsilon) \mu$.

Theorem 1. *Let $X \in \mathcal{G}$, where $\mathcal{G} = \mathrm{SE}_p(3)$, be a state matrix following either a left or right state space model* (6.1) *and* (6.2)*. The propagation Jacobian $F \in \mathbb{R}^{d\times d}$ at the linearization point $\mu \in \mathcal{G}$ writes in the left case:*

$$F = \begin{bmatrix} \dfrac{\partial}{\partial \epsilon} \log^{\vee}_{\mathrm{SO}(3)}(f_C(\mu)^{-1} f_C(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon))) \\ f_C(\mu)^{-1} \dfrac{\partial}{\partial \epsilon} f_1(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)) \\ \vdots \\ f_C(\mu)^{-1} \dfrac{\partial}{\partial \epsilon} f_p(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)) \end{bmatrix}. \tag{6.100}$$

In the right case, F writes:

$$F = \begin{bmatrix} \dfrac{\partial}{\partial \epsilon} \log^{\vee}_{\mathrm{SO}(3)}(f_C(\exp^{\wedge}_{\mathcal{G}}(\epsilon)\,\mu) f_C(\mu)^{-1}) \\ \dfrac{\partial}{\partial \epsilon} f_1(\exp^{\wedge}_{\mathcal{G}}(\epsilon)\,\mu) - f_C(\exp^{\wedge}_{\mathcal{G}}(\epsilon)\,\mu) f_C(\mu)^{-1} f_1(\mu) \\ \vdots \\ \dfrac{\partial}{\partial \epsilon} f_p(\exp^{\wedge}_{\mathcal{G}}(\epsilon)\,\mu) - f_C(\exp^{\wedge}_{\mathcal{G}}(\epsilon)\,\mu) f_C(\mu)^{-1} f_p(\mu) \end{bmatrix}. \tag{6.101}$$

6.3.2 . Proof for the left case

The left Jacobian defined in Proposition 1 is the derivative of the smooth function g defined as:

$$g(\epsilon) \triangleq f(\mu)^{-1} f(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)). \tag{6.102}$$

Since g is also on $\mathrm{SE}_p(3)$, it has a similar decomposition to (6.99):

$$g(\epsilon) = \begin{bmatrix} g_C(\epsilon) & g_1(\epsilon) \cdots g_p(\epsilon) \\ 0_{p,3} & I_p \end{bmatrix}, \tag{6.103}$$

where:

$$\begin{aligned} g_C(\epsilon) &= f_C(\mu)^{-1} f_C(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)), \\ g_i(\epsilon) &= f_C(\mu)^{-1}(f_i(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)) - f_i(\mu)), i \in [1, p]. \end{aligned} \tag{6.104}$$

Applying the decomposition of g to (6.10):

$$\begin{aligned} F &= \frac{\partial}{\partial \epsilon} \log^{\vee}_{\mathcal{G}}(g(\epsilon)), \\ &= \frac{\partial}{\partial \epsilon} \log^{\vee}_{\mathcal{G}} \left(\begin{bmatrix} g_C(\epsilon) & g_1(\epsilon) \cdots g_p(\epsilon) \\ 0_{p,3} & I_p \end{bmatrix} \right). \end{aligned} \tag{6.105}$$

The definition of the logarithm of $\mathrm{SE}_p(3)$ and the algebra isomorphism detailed in Appendix A gives:

$$F = \frac{\partial}{\partial \epsilon} \begin{bmatrix} \log^{\vee}_{\mathrm{SO}(3)}(g_C(\epsilon)) \\ \varphi_{\mathrm{SO}(3)}(g_C(\epsilon)) g_1(\epsilon) \\ \vdots \\ \varphi_{\mathrm{SO}(3)}(g_C(\epsilon)) g_p(\epsilon) \end{bmatrix}. \tag{6.106}$$

The derivative product rule on the vectorial terms taken at $\epsilon = 0$ gives:

$$F = \frac{\partial}{\partial \epsilon} \begin{bmatrix} \log^{\vee}_{\mathrm{SO}(3)}(g_C(\epsilon)) \\ g_1(\epsilon) \\ \vdots \\ g_p(\epsilon) \end{bmatrix}, \tag{6.107}$$

as $g_i(0) = 0$, $g_C(0) = I_3$ and $\varphi_{\mathrm{SO}(3)}(I_3) = I_3$. Then, expressing the terms of g gives:

$$F = \frac{\partial}{\partial \epsilon} \begin{bmatrix} \log^{\vee}_{\mathrm{SO}(3)}(f_C(\mu)^{-1} f_C(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon))) \\ f_C(\mu)^{-1} \left(f_1(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)) - f_1(\mu) \right) \\ \vdots \\ f_C(\mu)^{-1} \left(f_p(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)) - f_p(\mu) \right) \end{bmatrix}, \tag{6.108}$$

leading to:

$$F = \frac{\partial}{\partial \epsilon} \begin{bmatrix} \log^{\vee}_{\mathrm{SO}(3)}(f_C(\mu)^{-1} f_C(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon))) \\ f_C(\mu)^{-1} f_1(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)) \\ \vdots \\ f_C(\mu)^{-1} f_p(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)) \end{bmatrix}, \tag{6.109}$$

which proves the left expression of F in Theorem 1. □

6.3.3 . Proof for the right case

The proof for the right case unfolds similarly as the left case from equation (6.105), (6.106) and (6.107), with g from (6.104) redefined as:

$$g(\epsilon) \triangleq f(\mu \exp^{\wedge}_{\mathcal{G}}(\epsilon)) f(\mu)^{-1}, \tag{6.110}$$

where:

$$\begin{aligned} g_C(\epsilon) &= f_C(\exp^{\wedge}_{\mathcal{G}}(\epsilon)\, \mu) f_C(\mu)^{-1}, \\ g_i(\epsilon) &= f_i(\exp^{\wedge}_{\mathcal{G}}(\epsilon)\, \mu) - f_C(\exp^{\wedge}_{\mathcal{G}}(\epsilon)\, \mu) f_C(\mu)^{-1} f_i(\mu), i \in [1, p]. \end{aligned} \tag{6.111}$$

□

6.4 . Conclusion

This chapter derives LG-EKF with the Bayesian approach on unimodular Lie groups provided in Chapter 5 and introduces a direct approach to calculate the Jacobians on the Special Euclidean group $\mathrm{SE}_p(3)$.
This revisited approach of LG-EKF differs from the literature as the stochastic processes of the propagation step are multiplicative in the group. Besides, the formulation of the model dynamics is directly defined from state-space model without the introduction of a dynamics function in the algebra as proposed in [15]. This allows a more natural approach to complex dynamics systems, and a framework closer to conventional EKF. The other contribution of this chapter develops a simplified formulation of the Jacobians when the state belongs to $\mathrm{SE}_p(3)$, which is a popular Lie group for navigation and tracking applications.

7 - Particle Filter on Lie Groups

In this Chapter, the Optimal filter from Chapter 5 is solved using a sampling Monte Carlo approximation on Lie group without further assumptions on the state nor the measurement. This generic approach leads to Particle Filter on Lie groups (LG-PF), which formulation is similar to Particle Filter described in Chapter 2. Hence, this Chapter describes a flexible background which can be adapted to multiple estimation problems.
This Chapter on LG-PF unfolds in three sections. First the problem is stated for general nonlinear and non-Gaussian discrete time systems. Then, a general algorithm is proposed, and its steps are detailed in a second section. Finally, a last section concludes the chapter.

7.1 . Problem Statement

This chapter focuses on the general discrete state-space system:

$$\begin{cases} X_{k+1} &= f(X_k, u_{k+1}, \nu_k^q), \\ Y_{k+1} &= h(X_{k+1}, \nu_{k+1}^r), \end{cases} \tag{7.1}$$

where u_{k+1} is an input vector and ν_k^q, ν_{k+1}^r are two centered noise vectors. Also, it is assumed that this system verifies the probabilistic framework and the assumptions described in Chapter 5. Particle Filter on Lie groups (LG-PF) solves Optimal Filter equations using a Monte-Carlo approximation of the probability densities.

Based on the definition of the Dirac impulse on Lie groups described in Chapter 3, the Monte Carlo approximation gives:

$$p(X_k|Y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta_{X_k^i}(X_k), \tag{7.2}$$

where w_k^i denotes the weight such that $\sum_{i=1}^{N_p} w_k^i = 1$.

7.2 . General Solution

Particle Filter on Lie groups formalism is alike usual particle filter. It is described in Algorithm 8.

Algorithm 8 Lie Group Particle Filter

Result: ($\hat{\mu}_{k|k}$, $P_{k|k}$), $k \in [1, N]$
Initialization step: $X_0^i \sim p(X_0),\ \forall i \in [1, N_p]$
Propagation step: $X_{k+1}^i = f(X_k^i, u_{k+1}, \nu_k^{q,i})$,
Update step: $\tilde{w}_{k+1}^i = w_k^i p(Y_{k+1}|X_{k+1}^i)$,
Normalization: $w_{k+1}^i = \tilde{w}_{k+1}^i / \sum_{i=1}^{N_p} w_{k+1}^i$,
if $N_{\text{eff}} < N_{\text{th}}$ **then**
  Resampling step: Draw $\left\{X_{k+1}^i, w_{k+1}^i\right\}$ with a resampling strategy.
end if
Output: $\hat{\mu}_{k+1|k+1} = \mathbb{E}\left[X_{k+1}|Y_{1:k+1}\right]$, $P_{k+1|k+1} = \mathbb{V}\left[X_{k+1}|Y_{1:k+1}\right]$

The sequel details every step of LG-PF.

7.2.1 . Initialization Step

The filter is initialized with a sample of N_p particles drawn from the density $p(X_0)$ chosen by the user. Sampling on matrix Lie groups is detailed in Chapter 3.
For instance, if the initial density is a left concentrated normal law on $\mathcal{G}$ such that $p(X_0) \sim \mathcal{N}_{\mathcal{G}}(X_0; \mu_0, P_0)$, a first sample of vectors are sampled as:

$$\epsilon_0^i \sim \mathcal{N}_{\mathbb{R}^d}(\epsilon; 0, P_0). \tag{7.3}$$

Then, this sample is projected to the group at the vicinity of I_d with the group exponential and shifted around the mean with a left multiplication:

$$X_0^i = \mu_0 \exp_{\mathcal{G}}^{\wedge}\left(\epsilon_0^i\right). \tag{7.4}$$

7.2.2 . Propagation Step

The propagation step is a shift of the prior particle sample according to the dynamics function f with the inputs u_{k+1} and the process noise $\nu_k^{q,i}$ denotes a realization of ν_k^q wich differs for every particle.
The propagation step is obtained with the Chapman-Kolmogorov equation (5.4) which can be developed with the Monte Carlo approximation (7.2):

$$\begin{aligned} p(X_{k+1}|Y_{1:k}) &= \int_{\mathcal{G}} p(X_{k+1}|X_k) p(X_k|Y_{1:k}) \mathrm{d}X_k, \\ &\approx \int_{\mathcal{G}} p(X_{k+1}|X_k) \sum_{i=1}^{N_p} w_k^i \delta_{X_k^i}(X_k) \mathrm{d}X_k, \\ &\approx \sum_{i=1}^{N_p} w_k^i \int_{\mathcal{G}} p(X_{k+1}|X_k) \delta_{X_k^i}(X_k) \mathrm{d}X_k. \end{aligned} \tag{7.5}$$

By definition of the Dirac impulse function:

$$\int_{\mathcal{G}} p(X_{k+1}|X_k)\delta_{X_k^i}(X_k)\mathrm{d}X_k = p(X_{k+1}|X_k^i). \tag{7.6}$$

Thus, the prior density becomes:

$$p(X_{k+1}|Y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta_{X_{k+1}^i}(X_{k+1}), \tag{7.7}$$

where X_{k+1}^i denotes the i^{th} propagated particle according to the dynamics of (5.1) :

$$\forall i \in [1, N_p] : X_{k+1}^i = f(X_k^i, u_{k+1}, n_k^{q,i}). \tag{7.8}$$

Note that the propagated weights are unchanged.

7.2.3 . Update Step

The update step consists in a weight update and normalization. The Bayes equation 5.5 is solved with the approximation (7.7) obtained in the prediction:

$$p(X_{k+1}|Y_{1:k+1}) \approx \sum_{i=1}^{N_p} \frac{p(Y_{k+1}|X_{k+1})}{p(Y_{k+1}|Y_{1:k})} w_k^i \delta_{X_{k+1}^i}(X_{k+1}). \tag{7.9}$$

By definition of the Dirac impulse:

$$p(Y_{k+1}|X_{k+1}^i) = p(Y_{1:k+1}|X_{k+1})\delta_{X_{k+1}^i}(X_{k+1}). \tag{7.10}$$

Thus, the new weights can be identified as:

$$\tilde{w}_{k+1}^i = \frac{p(Y_{k+1}|X_{k+1}^i)}{p(Y_{k+1}|Y_{1:k})} w_k^i. \tag{7.11}$$

After a normalization step:

$$w_{k+1}^i = \frac{\tilde{w}_{k+1}^i}{\sum_{i=1}^{N_p} \tilde{w}_{k+1}^i}, \tag{7.12}$$

the estimated posterior density writes:

$$p(X_{k+1}|Y_{1:k+1}) \approx \sum_{i=1}^{N_p} w_{k+1}^i \delta_{X_{k+1}^i}(X_{k+1}). \tag{7.13}$$

7.2.4 . Proposal density

Lie group particle filter also enables a Sequential Importance Sampling (SIS) approach. Given the properties of unimodular matrix Lie groups, it is possible to use an instrumental density $\tilde{q}$ to perform the updated of LG-PF. Assuming that the customary assumptions of SIS are verified:

- The support of $\tilde{q}$ includes the support of p;

- The integral $\int \frac{p(X)^2}{\tilde{q}(X)} \mathrm{d}X$ is finite.

then, the Monte Carlo approximation gives:

$$p(X_{k+1}|Y_{1:k+1}) \approx \sum_{i=1}^{N_p} w_{k+1}^i \delta_{X_{k+1}^i}(X_{k+1}), \tag{7.14}$$

where the weights are:

$$w_{k+1}^i = \tilde{w}_{k+1}^i / \sum_{i=1}^{N} \tilde{w}_{k+1}^i \text{ such that } \tilde{w}_{k+1}^i = \frac{p(Y_{k+1}|X_{k+1}^i) p(X_{k+1}^i|Y_{1:k})}{\tilde{q}(X_{k+1}^i)} \tag{7.15}$$

7.2.5 . Resampling step

A resampling step is performed if the particles are degenerated with the same principle described in [47] and recalled for Particle Filter in Chapter 2. Besides, the resampling strategy is unchanged compared to usual PF. Indeed, classic resampling steps focus on the weights of the particles which are independent from the nature and the space of which the particles belong to. In this thesis, the usual resampling is based on a multinomial strategy which duplicates the heavy weights and removes the weights close to zero. After the resampling, all the weights are set to $\frac{1}{Np}$.

7.3 . Conclusion

This chapter introduces Particle Filter on Lie groups (LG-PF) which solves on the Bayes Filter described in Chapter 5. The developments on LG-PF show that is is similar to conventional Particle Filter (PF) designed on the Euclidean space. The propagation step is a temporal integration of the dynamics model for every particle, and the posterior density is computed by updating the weights according to the likelihood or a chosen proposal density. Besides, conventional resampling strategies which only take into account the weight distribution of the estimated prior density can be applied.
LG-PF differs from PF on some aspects. The sampling of the particles has to account for the group geometry, which involves a specific process described in Chapter 3. Furthermore, the computation of the conditional mean and covariance matrix also differ from the Euclidean case. This filter is expected to provide a better representation of the probability densities on a large domain due to the definition of the variables on the Lie group. However, it is also expected that LG-PF faces similar challenges as PF such as the curse of dimensionality [71] or accurate the resampling when there is little overlap between the likelihood and the prior density. These issues will be addressed in the next chapters, leading to substantial improvements of LG-PF.

8 - Laplace Particle Filter on Lie Groups

This chapter completes Particle Filter on Lie Groups (LG-PF) introduced in Chapter 7. It follows the Bayesian estimation framework described in Chapter 5 and focuses on the resampling step. The ideas developed in this chapter are inspired by the works on the Laplace Particle Filter (LPF) [62, 72, 56] discussed in Chapter 2 in the Euclidean case. Indeed, LPF involves an optimization process which computes an accurate proposal density for the resampling step, leading to improved robustness and accuracy.
The algorithms introduced in the sequel are based on a similar principle, and lead to Laplace Particle Filter on Lie groups (LG-LPF). Specific attention has been paid to the optimization algorithm, focusing on its efficiency and simplicity of implementation.
The chapter unfolds in several sections. First, the problem is stated in Section 8.1. Then, Section 8.2 introduces a first solution in the case where the probability densities are unimodal, and Section 8.3 extends this solution when the probability densities are multimodal.

8.1 . Problem Statement

This chapter unfolds in the same framework as LG-PF with the generic system:

$$\begin{cases} X_{k+1} &= f(X_k, u_{k+1}, \nu_k^q), \\ Y_{k+1} &= h(X_{k+1}, \nu_{k+1}^r), \end{cases} \tag{8.1}$$

where u_{k+1} is an input vector and (ν_k^q, ν_{k+1}^r) are two centered noise vectors. The sequel aims to calculate a proposal density denoted $\tilde{q}$, which mean and covariance matrix are close to the statistical moments of the posterior density.
The two methods presented in this chapter follow the same principle. Laplace resampling step draw a new set of particles in the most probable areas, that is to say, at the vicinity of the highest values of the posterior density. According to the Bayes rule, the posterior density writes:

$$p(X_{k+1}|Y_{1:k+1}) \propto p(Y_{k+1}|X_{k+1})p(X_{k+1}|Y_{1:k}). \tag{8.2}$$

Therefore the areas which maximize the posterior density are close to the set of local extremums:

$$\tilde{\mu}_{k+1}^n = \left\{ \arg\max_{X_{k+1}} p(Y_{k+1}|X_{k+1})p(X_{k+1}|Y_{1:k}) \right\}_{n\in[1,\tilde{N}_c]}. \tag{8.3}$$

This represents a difficult optimization problem in practice, as it involves nonlinear models with possibly several local maximums. Hence, this chapter introduces efficient numerical methods to solve the maximization (8.3) with close-to-optimal solutions.
First, the problem is solved in the unimodal case, where the densities present a predominant peak. Then, it is addressed in the multimodal case, where the prior density and the likelihood may have several local maxima in various regions of the state space.

8.2 . Unimodal Laplace Resampling

In this section, it is assumed that the probability densities have a single peak. According to LPF theory, the proposal density mean and covariance matrix should be respectively close to the conditional expectation $\mathbb{E}\left[X_{k+1}|Y_{1:k+1}\right]$ and the conditional variance $\mathbb{V}\left[X_{k+1}|Y_{1:k+1}\right]$ [70]

A suitable definition for the proposal density is to take the mean and covariance matrix of $\tilde{q}$ as:

$$\mathbb{E}\left[X_{k+1}|Y_{1:k+1}\right] \approx \mu^*_{k+1}, \tag{8.4}$$
$$\mathbb{V}\left[X_{k+1}|Y_{1:k+1}\right] \approx J(\mu^*_{k+1})^{-1}, \tag{8.5}$$

where μ^*_{k+1} is the *Maximum A Posteriori* (MAP) and $J(\mu^*_{k+1})$ is the Fisher information matrix defined in Chapter 3, calculated at the MAP. Then, $\tilde{q}$ can be taken as a concentrated Gaussian on $\mathcal{G}$:

$$\tilde{q}(X_{k+1}) = \phi_{\mathcal{G}}\left(X_{k+1}; \tilde{\mu}_{k+1}, \tilde{P}_{k+1}\right), \tag{8.6}$$

where its parameters are taken as $\tilde{\mu}_{k+1} = \mu^*_{k+1}$ and $\tilde{P}_{k+1} = J(\mu^*_{k+1})^{-1}$.

Algorithm 8.2 details the full LG-LPF process. It is similar to LG-PF presented in Algorithm 8 from Chapter 7 for the initialization step, propagation step and update step. LG-LPF solve the Optimal Filter on matrix Lie groups from Chapter 5 in the case where the densities are represented with a sum of weighted Dirac functions. The novelty of this section is about the resampling step where an optimization algorithm computes an approximation of the MAP and the inverse of the Fisher information matrix, which are close to the conditional expectation and variance. Then, they are used as parameters of a proposal density $\tilde{q}$, from which a new set of particles is drawn. Then the weights are updated and normalized.

Algorithm 9 Lie Group Laplace Particle Filter (Unimodal)

Result: $(\hat{\mu}_{k|k}, P_{k|k})$, $k \in [1, N]$
Initialization step: $X^i_0 \sim p(X_0)$
Propagation step: $X^i_{k+1} = f(X^i_k, u_{k+1}, \nu^{q,i}_k)$,
Update step: $\tilde{w}^i_{k+1} = w^i_k p(Y_{k+1}|X^i_{k+1})$,
Normalization: $w^i_{k+1} = \tilde{w}^i_{k+1} / \sum_{i=1}^{N_p} \tilde{w}^i_{k+1}$,
if $N_{\text{eff}} < N_{\text{th}}$ **then**
  Optimization: $\tilde{\mu}_{k+1} = \mu^*_{k+1}$ and $\tilde{P}_{k+1} = J(\mu^*_{k+1})^{-1}$ from Algorithm 10
  Draw: $X^i_{k+1} \sim \tilde{q}\left(X_{k+1}; \tilde{\mu}_{k+1}, \tilde{P}_{k+1}\right)$
  Update weights: $\tilde{w}^i_{k+1} = \dfrac{p(Y_{k+1}|X^i_{k+1}) p(X^i_{k+1}|Y_{1:k})}{\tilde{q}\left(X_{k+1}; \tilde{\mu}_{k+1}, \tilde{P}_{k+1}\right)}$
  Normalization: $w^i_{k+1} = \tilde{w}^i_{k+1} / \sum_{i=1}^{N_p} \tilde{w}^i_{k+1}$,
end if
Output: $\hat{\mu}_{k+1|k+1} = \mathbb{E}\left[X_{k+1}|Y_{1:k+1}\right]$, $P_{k+1|k+1} = \mathbb{V}\left[X_{k+1}|Y_{1:k+1}\right]$

8.2.1 . Optimization

The *Maximum A Posteriori* (MAP) and the information matrix are obtained by solving the optimization problem (8.3). In practice, any relevant optimization algorithm is suitable to solve this problem. However, finding an exact solution could lead to a time consuming process since the models are nonlinear and the state can have a large dimension.
The core idea developed in this section is to solve the optimization problem with fitted Gaussians on the propagated density and the likelihood.

$$p(X_{k+1}|Y_{1:k}) \approx \phi_{\mathcal{G}}\left(X_{k+1}; \hat{\mu}_{k+1|k}, P_{k+1|k}\right), \tag{8.7}$$
$$p(Y_{k+1}|X_{k+1}) \approx \phi_{\mathcal{H}}\left(Y_{k+1}; h(\hat{\mu}_{k+1|k}), R_{k+1}\right). \tag{8.8}$$

This enables to use the Iterated Extended Kalman Filter on Lie groups (LG-ItEKF) introduced in [14], which leads to highly accurate solutions. The idea of Iterated Extended Kalman filter is to refine the update linearization point by repeating the Kalman update process, which leads to an approximation of the MAP. This process can also be encountered in the Euclidean space [9].

Then, the Bayes rule on the fitted Gaussian leads to the approximation:

$$p(X_{k+1}|Y_{1:k+1}) \approx \phi_{\mathcal{G}}\left(X_{k+1}; \hat{\mu}_{k+1|k}, P_{k+1|k}\right) \phi_{\mathcal{H}}\left(Y_{k+1}; h(\hat{\mu}_{k+1|k}), R_{k+1}\right) \tag{8.9}$$

Maximizing this density can be performed with LG-ItEKF for optimization presented in Algorithm 10 introduced in [14]. This algorithm solves the optimization problem (8.3) with a formalism which is identical to LG-EKF. Besides, the updated covariance matrix is a good approximation of the inverse of the Fisher information matrix. Indeed, extended Kalman filter estimated covariance matrix matches the inverse of the Fisher information matrix (i.e. the Cramer-Rao Lower Bound) when it is optimal. This process is illustrated in figure 8.1.

Algorithm 10 LG-ItEKF for Optimization

Inputs: $\hat{\mu}_{k+1|k}$, $P_{k+1|k}$, Y_{k+1}, ε
Initialize: $X_0 = \hat{\mu}_{k+1|k}$, $P_0 = P_{k+1|k}$
while $\left\|\log_{\mathcal{G}}^{\vee}\left(X_j^{-1} X_{j+1}\right)\right\| > \varepsilon$ **do**
$H_j = \dfrac{\partial \log_{\mathcal{H}}^{\vee}\left(h(X_j \exp_{\mathcal{G}}^{\wedge}(\epsilon))^{-1} Y_{k+1}\right)}{\partial \epsilon}$
$K_j = P_j H_j^T \left(H_j P_j H_j^T + R_{k+1}\right)^{-1}$
$P_j = (I_d - K_j H_j) P_j$
$X_j = \hat{\mu}_{k+1|k} \exp_{\mathcal{G}}^{\wedge}\left(K_j \log_{\mathcal{H}}^{\vee}\left(h(X_j)^{-1} Y_{k+1}\right)\right)$
end while
Maximum A Posteriori: $\mu_{k+1}^{*} = X_j$
Fisher Information matrix: $J(\mu_{k+1}^{*})^{-1} = P_j$

In practice, the convergence is obtained in a few iterations, depending on the threshold ε.

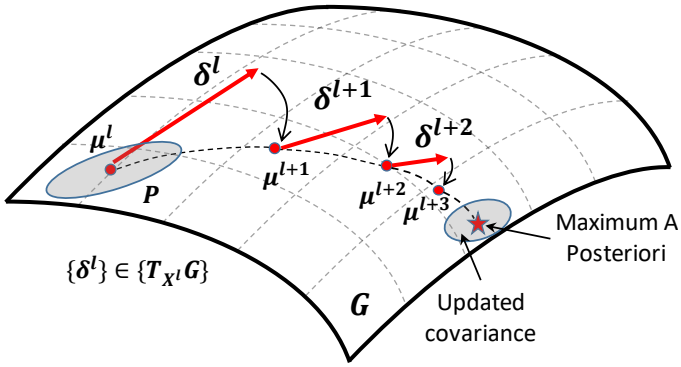


Figure 8.1: Illustration of the optimization process with LG-ItEKF presented in Algorithm 10

8.2.2 . Proposal Density

Once the parameters of the proposal density are obtained with Algorithm 10, a new set of particles is drawn:

$$X_{k+1}^i \sim \tilde{q}\left(X_{k+1}; \tilde{\mu}_{k+1}, \tilde{P}_{k+1}\right), \tag{8.10}$$

and the weights are computed through the importance sampling process:

$$\tilde{w}_{k+1}^i = \frac{p(Y_{k+1}|X_{k+1}^i)p(X_{k+1}^i|Y_{1:k})}{\tilde{q}\left(X_{k+1}^i; \tilde{\mu}_{k+1}, \tilde{P}_{k+1}\right)}. \tag{8.11}$$

Remark 1. *The proposal density $\tilde{q}$ can be chosen freely by the user. The closer it is from the posterior density the more accurate it is. Concentrated Gaussians are often chosen, although heavy-tailed densities (e.g. Student) could lead to more robustness [57].*

8.2.3 . Discussion

This section extends the approach of Laplace Particle Filter to matrix Lie groups with the probabilistic framework described in Chapter 3 and Chapter 5. The main interest of the proposed algorithm is the Gaussian approximation of the likelihood and the prior density which enables an optimization process based on LG-ItEKF detailed in Algorithm 10. This method provides a generic framework leading to accurate approximation of the MAP on the group. Besides, it does not require the explicit calculation of the Fisher information matrix as the updated covariance matrix of LG-ItEKF is a good approximation of its inverse. For this reason, even when $\mathcal{G}$ boils down to $\mathbb{R}^d$, the approach is simpler to implement than previous works on Euclidean Laplace Particle Filter (see Chapter 2).

8.3 . Multimodal Laplace Resampling

This section solves the optimization problem 8.3 assuming that the likelihood and the prior densities are multimodal (i.e. they present several peaks). When this is the case, using a unimodal proposal density for the resampling step would lead to a poor accuracy and could even overlook important components of the prior density as displayed in Figure 8.2. Thus, the sequel introduces a general formulation of the Laplace resampling by computing an accurate multimodal proposal density.

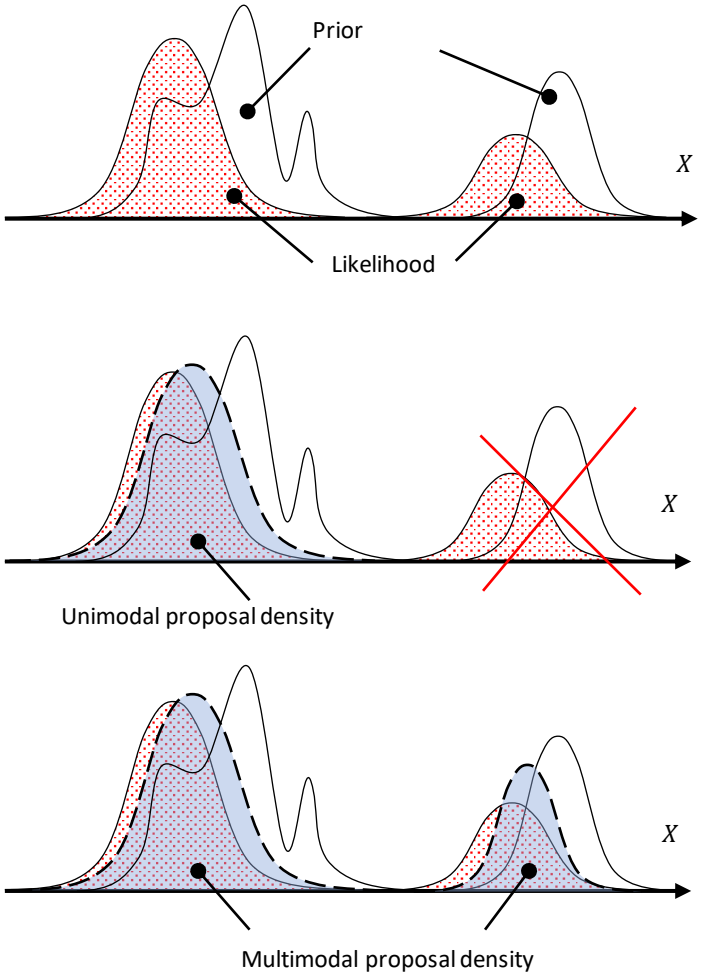


Figure 8.2: Illustration of the interest of a multimodal proposal density in the case of multimodal scenarios. A unimodal proposal density cannot properly represent a multimodal prior density.

8.3.1 . Notations

The solution described in the next sections requires the introduction of approximated Gaussian mixtures for the likelihood and the prior density.

$$p(X_{k+1}|Y_{1:k}) \approx \sum_{n=1}^{N_c} W_{k+1}^n \phi_{\mathcal{G}}(X_{k+1}; \mu_{k+1}^n, P_{k+1}^n), \tag{8.12}$$

$$p(Y_{k+1}|X_{k+1}) \approx \sum_{n=1}^{N_l} \mathcal{W}_{k+1}^n \phi_{\mathcal{H}}(Y_{k+1}; h^n(X_{k+1}), R_{k+1}^n). \tag{8.13}$$

Since the prior density and the likelihood present several peaks, the posterior density is expected to be multimodal as well. To that extent, this chapter aims to compute a multimodal proposal density defined by a kernel mixture, in order to resample the particles close to the areas of highest probability of the posterior density:

$$\tilde{q}(X_{k+1}) = \sum_{n=1}^{\tilde{N}_c} \widetilde{W}_{k+1}^n \phi_{\mathcal{G}}(X_{k+1}; \tilde{\mu}_{k+1}^n, \tilde{P}_{k+1}^n). \tag{8.14}$$

For the sake of clarity, the main elements are summarized in Table 8.1.

	Mean	covariance matrix	Weight	Number of kernels
Prior density	μ_{k+1}^n	P_{k+1}^n	W_{k+1}^n	N_c
Likelihood	$h^n(X_{k+1})$	R_{k+1}^n	$\mathcal{W}_{k+1}^n$	N_l
Proposal density	$\tilde{\mu}_{k+1}^n$	$\tilde{P}_{k+1}^n$	$\widetilde{W}_{k+1}^n$	$\tilde{N}_c$

Table 8.1: Summary of the fitted mixture parameters for the prior density, the likelihood and the proposal density.

8.3.2 . General solution

The resampling step of LG-LPF is based on the solution of the optimization problem (8.3). This problem is hard to address in practice, since the local maximums of the posterior density can be distributed on a large domain. Similarly to the previous section, the goal is to use the LG-ItEKF locally at the vicinity of the areas of highest probability. These areas can be determined with simple considerations on the Bayes equations. Since the posterior density is proportional to the product of the prior density and the likelihood, the areas of highest probability are found where the two latter densities overlap. This consideration is illustrated in Figure 8.3.

Algorithm 11 Generalized Lie Group Laplace Particle Filter

Initialization: sample $X_0^i \sim p(X_0)$;
Propagation: $X_{k+1}^i = f(X_k^i, \nu_{q,k}^i)$, $i \in [1, N_p]$
Update: $w_{k+1}^i \propto w_k^i p(Y_{k+1}|X_{k+1}^i)$
if $N_{\text{eff}} < N_{\text{th}}$ **then**
  Expectation Maximization on prior: $\{\mu_{k+1}^n, P_{k+1}^n, W_{k+1}^n\}_{n \in N_c}$
  Find the consistent pairs: Statistical test (8.25)
  for $n \subset \{\text{Consistentpairsindex}\}$ **do**
    Compute $\tilde{\mu}_{k+1}^n$ and $\tilde{P}_{k+1}^n$ with **Algorithm 10**.
    Compute $\widetilde{W}_{k+1}^n$ according to (8.26).
  end for
  Draw: $X_{k+1}^i \sim \tilde{q}(X_{k+1})$ according to (8.28)
  Re-initialize weights: w_{k+1}^i according to (8.30)
end if
Output: $\hat{\mu}_{k+1|k+1} = \mathbb{E}\left[X_{k+1}|Y_{1:k+1}\right]$, $P_{k+1|k+1} = \mathbb{V}\left[X_{k+1}|Y_{1:k+1}\right]$

The resampling step generalized LG-LPF described in Algorithm 11 is based on the previous statement. The first step fits Gaussian mixtures to the prior density and the likelihood with an Expectation Maximization algorithm (EM). Then, a screening strategy based statistical tests identifies the modes of prior density and the likelihood which are close to each other, defining the areas where there is a good overlap between the two densities. The overlapping modes of the prior density and the likelihood will be used as initialization points for local LG-ItEKF optimizations. Then, their solutions are taken as parameters of a multimodal proposal density which will resample the particles in the areas of highest probability. This process is illustrated in Figure 8.3. Similar approaches have been proposed in the Euclidian space framework [56, 59] using clustering algorithms.

8.3.3 . Approximation of the prior density

This section aims to approximate the prior density with a parametric Gaussian kernels mixture. The sequel describes a process which determines the parameters of (8.12).
The clusters of particles denoted $\left\{\mathcal{C}^1, ..., \mathcal{C}^{N_c}\right\}$ are identified with an Expectation-Maximization (EM) algorithm [52, 12]. Since this algorithm holds on the Euclidean space, the set of particles on the group is converted into a set of particles on the Euclidean space using the group logarithm:

$$\text{Left case: } \epsilon_{k+1}^i = \log_{\mathcal{G}}^{\vee}\left(\hat{\mu}_{k+1|k}^{-1} X_{k+1}^i\right), \tag{8.15}$$

$$\text{Right case: } \epsilon_{k+1}^i = \log_{\mathcal{G}}^{\vee}\left(X_{k+1}^i \hat{\mu}_{k+1|k}^{-1}\right). \tag{8.16}$$

The particles $\left\{\epsilon_{k+1|k}^i\right\}_{i \in [1, N_p]}$ are sorted in the set of cluster $\left\{\mathcal{C}^1, ..., \mathcal{C}^{N_c}\right\}$ using EM algorithm. The mean of the modes $\mu_{k+1}^n, n \in [1, N_c]$ are defined by:

$$\mu_{k+1}^n = \mathbb{E}\left[\left\{X_{k+1}^i\right\} \subset \mathcal{C}^n\right], \tag{8.17}$$

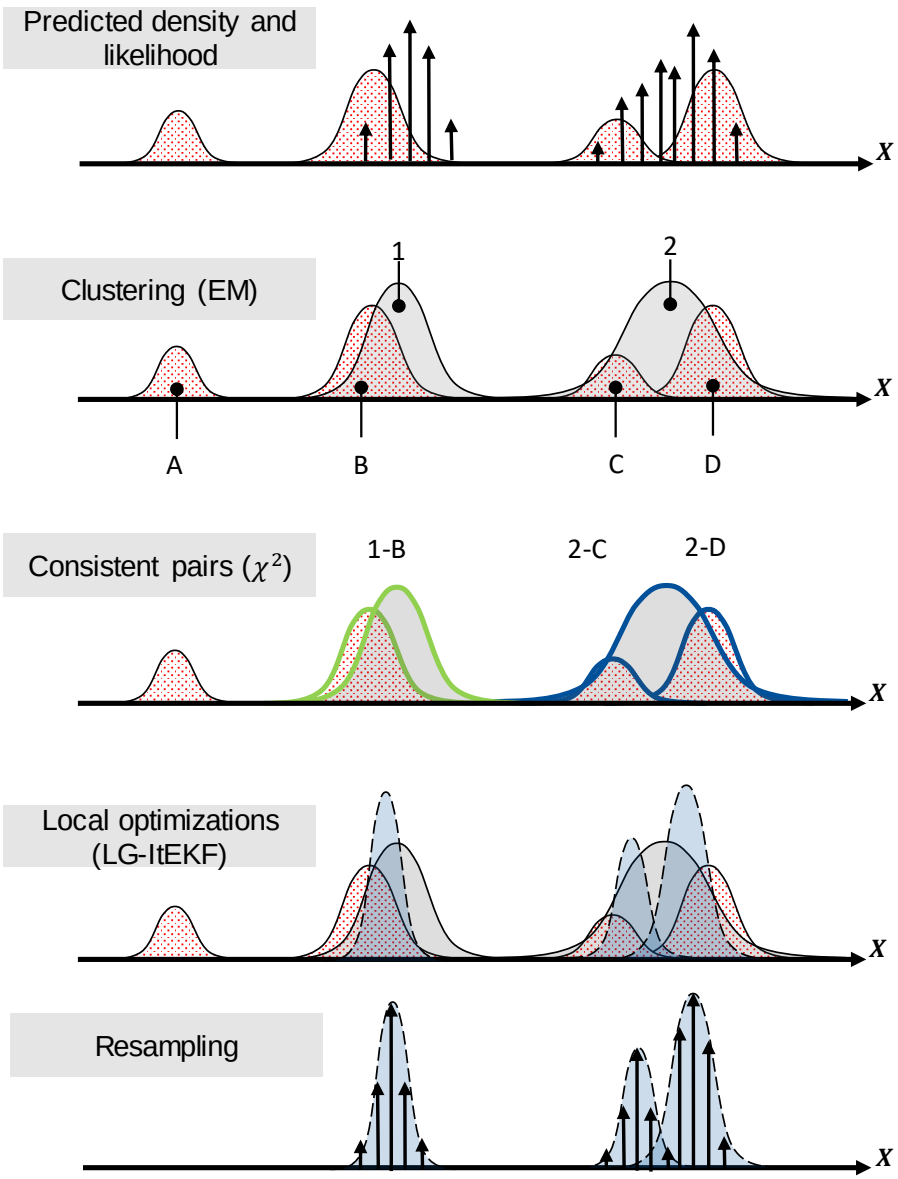


Figure 8.3: Illustration of the clustering and local optimization algorithm. Step 1 represents the approximated prior density with particles and the likelihood in red. Step 2 represents the mixture fitted to the prior with two kernels (1) and (2), and the likelihood with four kernels (A), (B), (C) and (D). Step 3 finds the consistent pairs based on a χ^2 with (8.25). Step 4 uses local optimizations based on LG-ItEKF for each consistent pair leading to the proposal density. In Step 5, the updated kernels of Step 4 are used for importance resampling.

and the covariance matrices $P_{k+1}^n, n \in [1, N_c]$ are given by:

$$P_{k+1}^n = \mathbb{V}\left[\left\{X_{k+1}^i\right\} \subset \mathcal{C}^n\right]. \tag{8.18}$$

Then, their weights W_p^n are defined as the total weight of the particles belonging to the cluster:

$$W_{k+1}^n = \sum_{X_{k+1}^i \subset \mathcal{C}^n} w_{k+1}^i. \tag{8.19}$$

Note that their sum is equal to unity:

$$\sum_{n=1}^{N_c} W_{k+1}^n = 1. \tag{8.20}$$

At the end of this process, the prior density function is approximated by a Gaussian mixture of N_c peaks:

$$p(X_{k+1}|Y_{1:k}) \approx \sum_{n=1}^{N_c} W_p^n \phi_{\mathcal{G}}\left(X_{k+1}; \mu_{k+1}^n, P_{k+1}^n\right). \tag{8.21}$$

8.3.4 . Likelihood definition

The likelihood is obtained from the physical model and properties of a sensor. Multimodal and non-Gaussian likelihood can occur in specific applications such as terrain-based navigation or ambiguous measurements.
This thesis focuses on the case where the likelihood can be approximated with a mixture of Gaussians on the groups of measurement $\mathcal{H}$:

$$p(Y_{k+1}|X_{k+1}) \approx \sum_{l=1}^{N_l} W_l^n \phi_{\mathcal{H}}\left(Y_{k+1}; h^l(X_{k+1}), R_{k+1}^l\right), \tag{8.22}$$

which fits most estimation scenarios.

8.3.5 . Computation of the Proposal Density

This section computes the parameters of $\tilde{q}$ defined in (8.14). According to the Bayes rule, the posterior density is proportional to the product of the prior density and the likelihood:

$$p(X_{k+1}|Y_{1:k+1}) \propto p(X_{k+1}|Y_{1:k}) p(Y_{k+1}|X_{k+1}). \tag{8.23}$$

Thus, the areas of highest probability are at the vicinity of the overlap between $p(X_{k+1}|Y_{1:k})$ and $p(X_{k+1}|Y_{1:k})$. Since the likelihood and the prior density are approximated with Gaussian mixtures, the optimization problem (8.3) writes:

$$\tilde{\mu}_{k+1}^n = \left\{ \arg\max_{X_{k+1}} \sum_{n=1}^{N_c} \sum_{l=1}^{N_l} W_{k+1}^n \mathcal{W}_{k+1}^l \phi_{\mathcal{G}}(X_{k+1}; \mu_{k+1}^n, P_{k+1}^n)) \phi_{\mathcal{H}}(Y_{k+1}; h^l(X_{k+1}), R_{k+1}^l) \right\}_{i \in [1, \tilde{N}_c]}. \tag{8.24}$$

As discussed in the previous sections, finding the exact solution to this problem is difficult, especially when the dimension of the state is large with a strongly nonlinear measurement model.
The proposed method is based on the idea that the solutions to the maximization problem (8.24) are close to the areas where there is a good overlap between the peaks of the prior density and the likelihood. This overlap is quantified by a χ^2 test between the kernels of the likelihood and the prior mixtures, and defines the notion of consistent pair, as illustrated in Figure 8.3.

A consistent pair is a set of two Gaussian kernels, one from the prior density and one from the likelihood, which are close according to a statistical test.
More precisely, let $X_{k+1} \in \mathcal{G}$ be a random matrix following (8.1), $\phi_{\mathcal{G}}(X_{k+1}; \mu^n_{k+1}, P^n_{k+1})$ be a kernel of the prior density and $\phi_{\mathcal{H}}(Y_{k+1}; h^m(X_{k+1}), R^m_{k+1})$ a kernel of the likelihood. Then, the two kernels are a consistent pair if:

$$\xi^T_{k+1} S^{-1}_{k+1} \xi_{k+1} < \mathcal{K}, \tag{8.25}$$

where $\mathcal{K}$ is chosen from the χ^2 test $p(\chi^2(m) \leqslant \mathcal{K}^2)$, d is the dimension of the Lie algebra, ξ_{k+1} is the log-linear error between Y_{k+1} and $h^m(\mu^n_{k+1})$, $S_{k+1} = H_{k+1} P^n_{k+1} H^T_{k+1} + R^m_{k+1}$, and H_{k+1} is the Lie group Jacobian of h^m at μ^n_{k+1}.

When the consistent pairs are obtained, an LG-ItEKF process is stated for each pair. Then, the updated mean and covariance matrix obtained at each optimization process are used as parameters for a Gaussian kernel of the proposal density $\tilde{q}$.
The last step determines the weights $\widetilde{W}^n_{k+1}$ of the proposal density. Each weight is defined by the product of the likelihood and the prior density taken at the posterior mean of each consistent pairs:

$$\widetilde{W}^n_{k+1} \propto p(Y_{k+1}|\tilde{\mu}^n_{k+1}) p(\tilde{\mu}^n_{k+1}|X_k), n \in [1, \tilde{N}_c]. \tag{8.26}$$

These weights are normalized such that:

$$\sum_{n=1}^{\tilde{N}} \widetilde{W}^n_{k+1} = 1. \tag{8.27}$$

Thus, the proposal density becomes:

$$\tilde{q}(X_{k+1}) = \sum_{n=1}^{\tilde{N}_c} \widetilde{W}^n_{k+1} \phi_{\mathcal{G}} \left(X_{k+1}; \tilde{\mu}^n_{k+1}, \tilde{P}^n_{k+1} \right). \tag{8.28}$$

Eventually, a new set of particles is drawn according to (8.28):

$$X^i_{k+1} \sim \tilde{q}(X_{k+1}), i \in [1, N_p], \tag{8.29}$$

and their weights are computed according to the sample importance sampling (SIS) process:

$$\tilde{w}^i_{k+1} = \frac{p(Y_{k+1}|X^i_{k+1}) p(X^i_{k+1}|Y_{1:k+1})}{\tilde{q}(X^i_{k+1})}. \tag{8.30}$$

Then, the weights are normalized:

$$w_{k+1}^i = \frac{\tilde{w}_{k+1}^i}{\sum_{i=1}^N \tilde{w}_{k+1}^i}. \tag{8.31}$$

8.3.6 . Discussion

This section extends LG-LPF to multimodal scenarios. This method holds under the assumption that the likelihood and the prior density can be approximated by Gaussian mixtures, which is the case of most estimation scenarios. In this chapter, Expectation Maximization is performed with the Matlab function $\mathbf{firgmdist}$ based on [53].

8.4 . Conclusion

This Chapter describes an adaptation of Laplace Particle Filter (LPF) to Lie groups in the case of unimodal and multimodal densities. It focuses on the resampling step that computes an accurate proposal which draws the particles close to the areas where the posterior density is highly probable. The proposal density is obtained by solving the Bayes equation of Chapter 5 with an optimization. Hence, LG-LPF solves Bayes filter with two approaches (filtering and optimization), which is expected to improve the filter's performance.
The optimization methods are also important contributions of this chapter since they involve Iterated Extended Kalman Filter on Lie groups (LG-ItEKF). This algorithm greatly simplifies the computations as LG-ItEKF provides a good approximation of the *Maximum A Posteriori* (MAP) and the Fisher information matrix on the group. Besides, implementing the optimization on the group shows better results in practice compared to conventional optimization in the Euclidean space [13].

9 - Kalman-Particle Kernel Filter on Lie Groups

This Chapter solves the Bayes filter with a hybrid approach based on Chapters 6 and 8. The main idea is to approximate the prior density with a weighted concentrated Gaussian mixture on Lie groups instead of Dirac functions as illustrated in Figure 9.1.

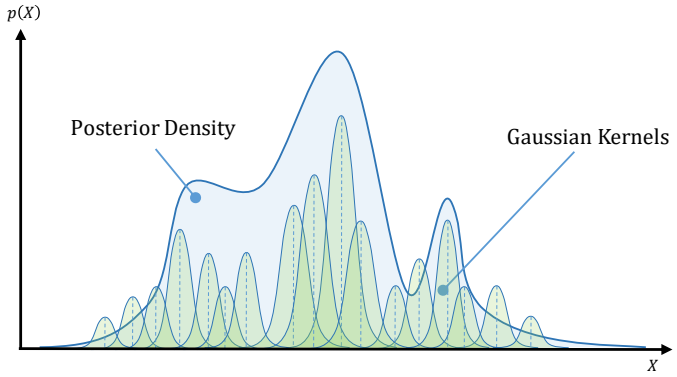


Figure 9.1: Representation of the estimated density with a weighted Gaussian mixture.

This approach was initially proposed in [30] for variables belonging to the Euclidean space, and named Kalman-Particle Kernel Filter (KPKF). The adaptation to Lie groups enables to develop a general Monte Carlo method which copes with high dimensions and shows improved accuracy, and is named Lie Group Kalman-Particle Kernel Filter (LG-KPKF). In the sequel, Section 9.1 states the problem. Then, Section 9.2 describes the general algorithm, and its steps are described in detail. Section 9.3 focuses on the specific case where the system is group-affine and Section 9.4 concludes the Chapter.

9.1 . Problem Statement

The following discrete state-space model is considered in this Chapter:

$$\begin{cases} X_{k+1} &= f(X_k, u_{k+1}, \nu_k^q), \\ Y_{k+1} &= h(X_{k+1}, \nu_{k+1}^r). \end{cases} \tag{9.1}$$

Also, it is assumed that this system verifies the probabilistic framework and the assumptions described in Chapter 5. LG-KPKF solves the Chapman Kolmogorov equation (5.4) and the Bayes rule (5.5) by approximating the prior density mean with a mixture of weighted Gaussians.

The definition of the concentrated Gaussian density on a matrix Lie group $\mathcal{G}$ is detailed in Chapter 3. The mixture approximation is:

$$p(X_k|Y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \phi_{\mathcal{G}} \left(X_k; \mu_{k|k}^i, P_{k|k}^i \right), \tag{9.2}$$

where w_k^i denotes the weights such that $\sum_{i=1}^{N_p} w_k^i = 1$ and $\phi_{\mathcal{G}}$ represents either a left or right concentrated Gaussian on $\mathcal{G}$.

Similarly, LG-KPKF assumes that the transition density and the likelihood are Gaussian functions:

$$p(X_{k+1}|X_k) = \phi_{\mathcal{G}} \left(X_{k+1}; f(X_k), Q_{k+1} \right), \tag{9.3}$$
$$p(Y_{k+1}|X_{k+1}) = \phi_{\mathcal{H}} \left(Y_{k+1}; h(X_{k+1}), R_{k+1} \right). \tag{9.4}$$

Although these assumptions are restrictive compared to LG-LPF, they are suitable to most applications. Since LG-KPKF can be seen as a hybrid between LG-EKF and LG-PF, a specific terminology is used. A particle refers to a weighted Gaussian kernel on the group. The quantity $\mu_{k|k}^i$ denotes the particle's mean and represents an abuse of notation since it is not a conditional expectation. The same remark applies to the particle's covariance matrix $P_{k|k}^i$. This abusive notation is kept in what follows for the sake of clarity with respect to the prediction step ($\mu_{k+1|k}^i$) and uptate ($\mu_{k+1|k+1}^i$) steps of LG-KPKF.
This formalism is kept as it is relevant to consider every particle as a local LG-EKF since their mean and covariance matrix follow a similar process.

9.2 . General Solution

The main steps of LG-KPKF are summarized in Algorithm 12. It shows a hybrid structure between LG-EKF and LG-LPF which involves the main steps of these two algorithms. Note that the resampling step of LG-KPKF is simpler than the one of KPKF dercribed in Chapter 2, since the linearization on the group is valid on a large domain. Hence, LG-KPKF does not have a partial resampling step.

Algorithm 12 Lie Group Kalman-Particle Kernel Filter

Result: $(\hat{\mu}_{k|k}, P_{k|k})$, $k \in [1, N]$
Initialization step:
$\mu_0^i \sim p(X_0; \mu_0, (1+h^2)^{-1} P_0)$, $P_0^i = h^2 P_0$
Propagation step $\forall i \in [1, N_p]$:
$\mu_{k+1|k}^i = f(\mu_{k|k}^i, u_{k+1})$
$P_{k+1|k}^i = F_k^i P_{k|k}^i {F_k^i}^T + Q_{k+1}$
Kalman update step $\forall i \in [1, N_p]$:
$S_{k+1}^i = H_{k+1} P_{k+1|k}^i {H_{k+1}^i}^T + R_{k+1}$,
$K_{k+1}^i = P_{k+1|k}^i {H_{k+1}^i}^T (S_{k+1}^i)^{-1}$,
$P_{k+1|k+1}^i = (I - K_{k+1} H_{k+1}) P_{k+1|k}^i$,
Left case: $\mu_{k+1|k+1}^i = \mu_{k+1|k}^i \exp_{\mathcal{G}}^{\wedge} \left(K_{k+1}^i \log_{\mathcal{H}}^{\vee} \left(Y_{k+1}^{-1} h \left(\mu_{k+1|k}^i \right) \right) \right)$,
Right case: $\mu_{k+1|k+1}^i = \exp_{\mathcal{G}}^{\wedge} \left(K_{k+1}^i \log_{\mathcal{H}}^{\vee} \left(h \left(\mu_{k+1|k}^i \right) Y_{k+1}^{-1} \right) \right) \mu_{k+1|k}^i$.
Weight update step: $\tilde{w}_{k+1}^i = w_k^i \phi_{\mathcal{H}} \left(Y_{k+1}; h \left(\mu_{k+1|k+1}^i \right), S_{k+1}^i \right)$,
Normalization: $w_{k+1}^i = \frac{\tilde{w}_{k+1}^i}{\sum_{i=1}^{N_p} w_{k+1}^i}$,
Resampling step: $\mu_{k+1}^i = \text{multinomial}(w_{k+1}^i)$, $w_{k+1}^i = \frac{1}{N_p}$,
Output: $\hat{\mu}_{k+1|k+1} = \mathbb{E}\left[X_{k+1} | Y_{1:k+1}\right]$, $P_{k+1|k+1} = \mathbb{V}\left[X_{k+1} | Y_{1:k+1}\right]$

9.2.1 . Initialization step

The initialization step draws an initial mixture of weighted Gaussian kernels. As discussed in [30] the particles are sampled with:

$$\mu_0^i \sim p(X_0; \mu_0, (1+h^2)^{-1} P_0) \tag{9.5}$$
$$P_0^i = h^2 P_0 \tag{9.6}$$

where (μ_0, P_0) are the mean and covariance matrix of $p(X_0)$ and h is the Silverman scaling parameter [76]. The sampling process on the group is described in Chapter 3.

9.2.2 . Propagation step

The propagation step solves the Chapman-Kolmogorov equation on Lie groups (5.4) with the mixture approximation from (9.2):

$$p(X_{k+1}|Y_{1:k}) \approx \int_{\mathcal{G}} \phi_{\mathcal{G}} \left(X_{k+1}; f(X_k), Q_{k+1} \right) \sum_{i=1}^{N_p} w_k^i \phi_{\mathcal{G}} \left(X_k; \mu_{k|k}^i, P_{k|k}^i \right) \mathrm{d}X_k. \tag{9.7}$$

Using the linearity property of integrals gives:

$$p(X_{k+1}|Y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \int_{\mathcal{G}} \phi_{\mathcal{G}} \left(X_{k+1}; f(X_k), Q_{k+1} \right) \phi_{\mathcal{G}} \left(X_k; \mu_{k|k}^i, P_{k|k}^i \right) \mathrm{d}X_k. \tag{9.8}$$

Eventually, the Chapman-Kolmogorov equation boils down to:

$$p(X_{k+1}|Y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i p(\epsilon_{k+1|k}^i), \tag{9.9}$$

where $p(\epsilon_{k+1|k}^i)$ denotes either a left or right concentrated Gaussian on $\mathcal{G}$. Hence the prior density is a concentrated Gaussian mixture on $\mathcal{G}$ such that:

$$p(X_{k+1}|Y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \phi_{\mathcal{G}}(X_{k+1}; \mu_{k+1|k}^i, P_{k+1|k}^i), \tag{9.10}$$

and its parameters are obtained with a LG-EKF propagation on each particle as defined in Chapter 6:

$$\mu_{k+1|k}^i = f(\mu_{k|k}^i), \tag{9.11}$$

$$P_{k+1|k}^i = F_k^i P_{k|k}^i {F_k^i}^T + Q_{k+1}, \tag{9.12}$$

and the particle weights remain unchanged.

9.2.3 . Update step

The update step proof is similar to the one of LG-EKF from Chapter 6. The general Bayes equation writes:

$$p(X_{k+1}|Y_{1:k+1}) = \frac{p(Y_{k+1}|X_{k+1})p(X_{k+1}|Y_{1:k})}{p(Y_{k+1}|Y_{1:k})}, \tag{9.13}$$

where the prior density and the likelihood are such that:

$$p(X_{k+1}|Y_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \phi_{\mathcal{G}}(X_{k+1}; \mu_{k+1|k}^i, P_{k+1|k}^i), \tag{9.14}$$

$$p(Y_{k+1}|X_{k+1}) = \phi_{\mathcal{H}}(Y_{k+1}; h(X_{k+1}), R_{k+1}). \tag{9.15}$$

Similarly to LG-EKF from Chapter 6, this proof unfolds in 5 steps and leads to the expression of the posterior density $p(X_{k+1}|Y_{1:k+1})$. First, the measurement model is linearized in the left and the right case at the vicinity of every particle mean. Then, the second step derives the expression of the updated covariance matrix of every particle. The third step calculates the marginalized likelihood $p(Y_{k+1}|Y_{1:k})$ and the fourth step leads to the expression of the posterior density. The fifth and last step leads to the update equation of the particles mean.

1- Linearization of the Measurement Model

(Left case) According to the Bayes rule, the posterior density writes:

$$p(X_{k+1}|Y_{1:k+1}) \propto \sum_{i=1}^{N_p} w_k^i \alpha_{k+1}^i \exp\left(-\frac{1}{2} \left\| \begin{matrix} \log_{\mathcal{H}}^{\vee}\left(h(X_{k+1})^{-1} Y_{k+1}\right) \\ \log_{\mathcal{G}}^{\vee}\left((\mu_{k+1|k}^i)^{-1} X_{k+1}\right) \end{matrix} \right\|_{\Lambda_{k+1}^i}^2 \right), \tag{9.16}$$

where:

$$\Lambda_{k+1}^i = \begin{bmatrix} R_{k+1} & 0 \\ 0 & P_{k+1|k}^i \end{bmatrix}, \tag{9.17}$$

and the normalization constant is:

$$\alpha_{k+1}^i = \left((2\pi)^{d+m} \det R_{k+1} \det P_{k+1|k}^i \right)^{-\frac{1}{2}}. \tag{9.18}$$

Similarly to the propagation step, the local left log-Euclidean error $\epsilon_{k+1|k}^i$ gives:

$$X_{k+1}^i = \mu_{k+1|k}^i \exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k+1|k}^i \right). \tag{9.19}$$

The term $\log_{\mathcal{H}}^{\vee} \left(h(X_{k+1})^{-1} Y_{k+1} \right)$ can be linearized locally around every particle mean:

$$\log_{\mathcal{H}}^{\vee} \left(h(X_{k+1})^{-1} Y_{k+1} \right) = z_{k+1}^i - H_{k+1}^i \epsilon_{k+1|k}^i + O(\|\epsilon_{k+1|k}^i\|^2), \tag{9.20}$$

where H_{k+1}^i is the measurement Jacobian:

$$H_{k+1}^i = \frac{\partial}{\partial \epsilon_{k+1|k}^i} \log_{\mathcal{H}}^{\vee} \left(Y_{k+1}^{-1} h \left(\mu_{k+1|k}^i \exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k+1|k}^i \right) \right) \right), \tag{9.21}$$

and z_{k+1}^i is the local innovation:

$$z_{k+1}^i = \log_{\mathcal{H}}^{\vee} \left(h \left(\mu_{k+1|k}^i \right)^{-1} Y_{k+1} \right). \tag{9.22}$$

(Right case) Considering now right densities:

$$p(X_{k+1}|Y_{1:k+1}) \propto \sum_{i=1}^{N_p} w_k^i \alpha_{k+1}^i \exp \left(-\frac{1}{2} \left\| \begin{matrix} \log_{\mathcal{H}}^{\vee} \left(Y_{k+1} h(X_{k+1})^{-1} \right) \\ \log_{\mathcal{G}}^{\vee} \left(X_{k+1} (\mu_{k+1|k}^i)^{-1} \right) \end{matrix} \right\|_{\Lambda_{k+1}^i}^2 \right), \tag{9.23}$$

where Λ_{k+1}^i and α_{k+1}^i are the same as the left case. By applying the sape approach as the left case, the linearized equation writes:

$$\log_{\mathcal{H}}^{\vee} \left(Y_{k+1} h(X_{k+1})^{-1} \right) = z_{k+1}^i - H_{k+1}^i \epsilon_{k+1|k}^i + O(\|\epsilon_{k+1|k}^i\|^2), \tag{9.24}$$

where H_{k+1}^i is the measurement Jacobian for the i^{th} particle:

$$H_{k+1}^i = \frac{\partial}{\partial \epsilon_{k+1|k}^i} \log_{\mathcal{H}}^{\vee} \left(h \left(\exp_{\mathcal{G}}^{\wedge} \left(\epsilon_{k+1|k}^i \right) \mu_{k+1|k}^i \right) Y_{k+1}^{-1} \right), \tag{9.25}$$

and z_{k+1}^i is the local innovation for the right error:

$$z_{k+1}^i = \log_{\mathcal{H}}^{\vee} \left(Y_{k+1} h \left(\mu_{k+1|k}^i \right)^{-1} \right). \tag{9.26}$$

2- Kalman Gain and Covariance Matrix Update Equations

The sequel of the proof is the same for the right and left case. It establishes the gain and covariance matrix update expressions for every particle. Given the previous linearized expressions, the posterior density writes:

$$p(X_{k+1}|Y_{1:k+1}) \propto \sum_{i=1}^{N_p} w_k^i \alpha_{k+1}^i \exp\left(-\frac{1}{2}\left\|\begin{bmatrix} I & -H_{k+1}^i \\ 0 & I \end{bmatrix}\begin{bmatrix} z_{k+1}^i \\ \epsilon_{k+1|k}^i \end{bmatrix}\right\|_{\Lambda_{k+1}^i}^2\right). \tag{9.27}$$

Following the same approach as the update equation proof for LG-EKF from Chapter 6 gives:

$$p(X_{k+1}|Y_{1:k+1}) \propto \sum_{i=1}^{N_p} w_k^i \alpha_{k+1}^i \exp\left(-\frac{1}{2}\left\|\begin{bmatrix} z_{k+1}^i \\ \epsilon_{k+1|k}^i - K_{k+1}^i z_{k+1}^i \end{bmatrix}\right\|_{\Upsilon_{k+1}^i}^2\right), \tag{9.28}$$

which also writes:

$$p(X_{k+1}|Y_{1:k+1}) \propto \sum_{i=1}^{N_p} w_k^i \alpha_{k+1}^i \exp\left(-\frac{1}{2}\left\|z_{k+1}^i\right\|_{S_{k+1}^i}^2\right) \exp\left(-\frac{1}{2}\left\|\epsilon_{k+1|k}^i - K_{k+1}^i z_{k+1}^i\right\|_{P_{k+1|k+1}^i}^2\right), \tag{9.29}$$

where $(K_{k+1}^i, P_{k+1|k+1}^i, S_{k+1}^i)$ are the customary matrices of the Kalman filter update:

$$S_{k+1}^i = H_{k+1}^i P_{k+1|k}^i {H_{k+1}^i}^T + R_{k+1}, \tag{9.30}$$

$$K_{k+1}^i = P_{k+1}^i {H_{k+1}^i}^T (S_{k+1}^i)^{-1}, \tag{9.31}$$

$$P_{k+1|k+1}^i = (I - K_{k+1}^i H_{k+1}^i) P_{k+1|k}^i. \tag{9.32}$$

and Υ_{k+1}^i writes:

$$\Upsilon_{k+1}^i = \begin{bmatrix} S_{k+1}^i & 0 \\ 0 & P_{k+1|k+1}^i \end{bmatrix}. \tag{9.33}$$

3- Expression of the Posterior Density

The posterior density is calculated by applying the Bayes rule (5.5) to (9.29):

$$p(X_{k+1}|Y_{1:k+1}) = \sum_{i=1}^{N_p} w_{k+1}^i \exp\left(-\frac{1}{2}\left\|\epsilon_{k+1|k}^i - K_{k+1}^i z_{k+1}^i\right\|_{P_{k+1|k+1}^i}^2\right) \tag{9.34}$$

where the new weights w_{k+1}^i are:

$$w_{k+1}^i \propto w_k^i \alpha_{k+1}^i \exp\left(-\frac{1}{2}\left\|z_{k+1}^i\right\|_{S_{k+1}^i}^2\right). \tag{9.35}$$

Hence, the weights of the posterior density are updated with a similar process to LG-PF and LG-LPF from Chapters 7 and 8. Besides, the particles covariance matrices are updated with the equation of LG-EKF from Chapter 6.

4- Mean Update Equation

The expression of the log-linear error for every particle of the posterior density is alike Section 6.2.2. Thus, the updated particles write:

$$\textbf{Left case: } \mu^i_{k+1|k} \exp^{\wedge}_{\mathcal{G}} \left(K^i_{k+1} z^i_{k+1}\right), \tag{9.36}$$

$$\textbf{Right case: } \exp^{\wedge}_{\mathcal{G}} \left(K^i_{k+1} z^i_{k+1}\right) \mu^i_{k+1|k}. \tag{9.37}$$

9.2.4 . Resampling step

The resampling step of LG-KPKF can be chosen freely by the user. It is triggered when the particles are degenerated according to the criterion [47]. In the case of a multinomial approach, the heavy-weighted particles mean and covariance matrix are duplicated. The weights are set to $1/N_p$.

In the case of Laplace resampling, LG-KPKF uses the Gaussian proposal density to resample the particle mean and covariance matrix. The general proposal density is given by (8.28) and recalled hereafter

$$\tilde{q}(X_{k+1}) = \sum_{n=1}^{\tilde{N}_p} W^n_{k+1} \phi_{\mathcal{G}} \left(X_{k+1}; \tilde{\mu}^n_{k+1}, \tilde{P}^n_{k+1}\right).$$

The means are sampled according to a Gaussian mixture which kernel covariance matrix with modified kernel covariance matrix:

$$X^i_{k+1} \sim \sum_{n=1}^{\tilde{N}_c} W^n_{k+1} \phi_{\mathcal{G}} \left(X_{k+1}; \tilde{\mu}^n_{k+1}, (1+h^2)\tilde{P}^n_{k+1}\right), \tag{9.38}$$

in order to have a resampled mixture consistent with the proposal density. The covariance matrix of the particles are taken from the covariance matrix of the proposal density. If a particle was sampled on the n^{th} kernel, its covariance matrix is taken as $h^2 \tilde{P}^n_{k+1}$, where h is a tuning parameter.

9.3 . Specific Case of Group-Affine Dynamics

Group-affine systems enable an autonomous propagation error in the sense that it does not depend on the estimated mean. Assume that the noises are Gaussian, and that f is group-affine:

$$\forall X_1, X_2 \in \mathcal{G} : f(X_1 X_2) = f(X_1) f(I_d)^{-1} f(X_2). \tag{9.39}$$

Note that the input vector u and the noise are omitted for the sake of brevity. Applying this property to the propagation Jacobian leads to a formulation which is independent from the linearization point.

In the group-affine case, the left and right Jacobian are the same for every particle and write:

$$\textbf{Left case: } F_k = \frac{\partial}{\partial \epsilon} \log_{\mathcal{G}}^{\vee} \left(f(I_d)^{-1} f(\exp_{\mathcal{G}}^{\wedge}(\epsilon)) \right). \tag{9.40}$$

$$\textbf{Right case: } F_k = \frac{\partial}{\partial \epsilon} \log_{\mathcal{G}}^{\vee} \left(f(\exp_{\mathcal{G}}^{\wedge}(\epsilon)) f(I_d)^{-1} \right). \tag{9.41}$$

Proof. In the left case the propagation Jacobian for every particle writes:

$$F_k^i = \frac{\partial}{\partial \epsilon} \log_{\mathcal{G}}^{\vee} \left(f(\mu_{k|k}^i)^{-1} f(\mu_{k|k}^i \exp_{\mathcal{G}}^{\wedge}(\epsilon)) \right). \tag{9.42}$$

Applying the group-affine property gives:

$$f(\mu_{k|k}^i)^{-1} f(\mu_{k|k}^i \exp_{\mathcal{G}}^{\wedge}(\epsilon)) = f(\mu_{k|k}^i)^{-1} f(\mu_{k|k}^i) f(I_d)^{-1} f(\exp_{\mathcal{G}}^{\wedge}(\epsilon)), \tag{9.43}$$

$$= f(I_d)^{-1} f(\exp_{\mathcal{G}}^{\wedge}(\epsilon)). \tag{9.44}$$

Therefore the left Jacobian writes:

$$F_k^i = \frac{\partial}{\partial \epsilon} \log_{\mathcal{G}}^{\vee} \left(f(I_d)^{-1} f(\exp_{\mathcal{G}}^{\wedge}(\epsilon)) \right). \tag{9.45}$$

A similar calculus leads to the Jacobian in the right case:

$$F_k^i = \frac{\partial}{\partial \epsilon} \log_{\mathcal{G}}^{\vee} \left(f(\exp_{\mathcal{G}}^{\wedge}(\epsilon)) f(I_d)^{-1} \right). \tag{9.46}$$

In both equations (9.45) and (9.46), the expression of the Jacobians are the same for every particle. □

Hence, when the state-space model is group-affine, the propagation Jacobians are all the same for KPKF. This represents an interesting property in practice, as the computation of the propagation Jacobians for every particle can be costly. Besides, the linearization is exact as it does not involve the estimated state.

9.4 . Conclusion

This chapter introduces the Kalman Particle Filter on Lie Groups (LG-KPKF) which can be seen as an hybrid approach between LG-EKF and LG-PF introduced in Chapters 6 and 7 respectively. This approach is expected to require less particles than conventional particle filters and should manage high-dimensional problems due to the use of local Kalman filters on every particle.
In addition, LG-KPKF is suitable to use Laplace resampling described in Chapter 8 for improved robustness and accuracy. Finally, the group-affine property on the dynamics model leads to a lighter and more efficient implementation of KPKF, which lowers the computational cost. In practice, many nonlinear systems encountered in robotic navigation hold a group-affine property [7]. This filter is expected to perform at close-to-optimal accuracy in the sense of the Optimal Filter introduced in Chapter 5, given the continuousness of the Gaussian mixture which enables an accurate approximation of the posterior density.

10 - Recursive Posterior Cramer-Rao Lower Bound

Cramer-Rao Lower Bound (CRLB) calculates the theoretical lowest asymptotic variance achievable by any estimator for a given scenario [73, 29]. It is ubiquitous in signal processing and control since it provides valuable insights for numerous applications. indeed, the CRLB evaluates the observability of a problem in the sense that one can unambiguously identify the hidden variables using noise-free measurements [34]. In the general case, if the CRLB is singular, the hidden state is not observable. Hence, CRLB has played a key role in industrial systems' robustness, dimensioning and performance evaluation. It found historical applications in tracking and navigation problems [74, 41]. Besides, the comparison of the root-mean square error (RMSE) of a filter with the CRLB shows if it is efficient. Since CRLB is the lowest attainable variance, the best performance is obtained when the RMSE equals CRLB. Note that this comparison holds only when the posterior is unimodal. In filtering problems, the posterior density mostly becomes unimodal asymptotically.
When designing a system, the bound gives valuable information on the highest accuracy expected from a data fusion process. If the specifications are below CRLB, even an optimal algorithm would be helpless to reach the requirements, and more accurate or additional sensors have to be used. Then, the bound is also a good means to quantify the impact of a configuration of sensors, which is helpful for a relevant design. On the opposite, if a specification is significantly higher than CRLB, the system may be too high-end for the expected applications. Thus, a simpler sensor setup (and probably cheaper) is sufficient to reach the desired accuracy.
Thichavsky et al. proposed a recursive formulation of the posterior CRLB on the Euclidean space (E-CRLB) [78]. The main asset of this formulation is the simplicity of its implementation, and its recursive form which enables to compute the lowest variance over a time sequence. Previous works on CRLB for matrix Lie groups were introduced in [11, 48]. However, these works do not account for the recursive aspect of the bound which are important when designing a filter.
Hence, this Chapter focuses on a recursive formulation the Cramer-Rao Lower Bound on matrix Lie groups (LG-CRLB). This enables to fully characterize the behavior of errors on Lie groups and it would help a robust design of filters on Lie groups for industrial applications.

10.1 . Problem Statement

Let X_k be a random matrix of an unimodular matrix Lie group $\mathcal{G}$ following the discrete-time state-space system:

$$\begin{cases} X_{k+1} &= f(X_k, u_{k+1}, \nu_k^q), \\ Y_k &= h(X_k, \nu_k^r), \end{cases} \tag{10.1}$$

where f and h are two nonlinear functions, (ν_k^r, ν_k^q) are centered noise vectors, and $\mathcal{H}$ is a unimodular matrix Lie group. According to the results of Chapter 3, two equivalent formulations of the Fisher information matrix on Lie groups can be used.

Expectation formulation:

$$J = -\mathbb{E}\left[\frac{\partial^2 \log p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon) \exp_{\mathcal{G}}^{\wedge}(\xi))}{\partial\epsilon\partial\xi}\right]. \tag{10.2}$$

Integral formulation:

$$J = \int_{\mathcal{G}} \frac{1}{p(X)} \left(\frac{\partial p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon))}{\partial\epsilon}\right) \left(\frac{\partial p(X \exp_{\mathcal{G}}^{\wedge}(\epsilon))}{\partial\epsilon}\right)^T dX. \tag{10.3}$$

The LG-CRLB is the inverse of the Fisher information matrix [27]. Based on the previous formulations, the sequel derives a recursive formulation of LG-CRLB in the case of unimodular matrix Lie groups.

10.2 . A Recursive Lower Bound on Lie Groups

The concepts developed in this section provide a natural formulation for the recursive Cramér-Rao Lower Bound of [78] when the state and the observations belong to unimodular matrix Lie groups.

10.2.1 . The State Recursive Formulation

Starting from the formulation of the Fisher information matrix, a recursive bound computes the information matrix of the discrete-time sequence $X_{0:k} = [X_0, \cdots, X_k]$, where the bound at time k uses the previous value at time $k-1$.

In the spirit of [78], the augmented state matrix is introduced:

$$\mathcal{X}_k = \text{diag}(X_0, \cdots, X_k) \in \mathcal{G}^{k+1}, \tag{10.4}$$

where $\mathcal{G}^{k+1}$ is defined from the Cartesian product of Lie groups detailed in Appendix C. The information matrix for $\mathcal{X}_k$ writes:

$$J(\mathcal{X}_k) = -\mathbb{E}\left[\frac{\partial^2 \log p\left(\mathcal{X}_k \exp_{\mathcal{G}^{k+1}}^{\wedge}(\epsilon_{0:k}) \exp_{\mathcal{G}^{k+1}}^{\wedge}(\xi_{0:k})\right)}{\partial\epsilon_{0:k}\partial\xi_{0:k}}\right], \tag{10.5}$$

where $\epsilon_{0:k} = [\epsilon_0^T, \cdots, \epsilon_k^T]^T \in \mathbb{R}^{(k+1)d}$. This consideration enables to extend the recursive bound of [78].

Proposition 3. *The Lie group Fisher information matrix of the state at time k verifies:*

$$J(X_k) = C_k - B_k^T A_k^{-1} B_k, \tag{10.6}$$

where the matrices (A_k, B_k, C_k) *are:*

$$A_k \triangleq -\mathbb{E}\left[\frac{\partial^2 \log p\left(\mathcal{X}_k \exp^{\wedge}_{\mathcal{G}^{k+1}}(\epsilon_{0:k}) \exp^{\wedge}_{\mathcal{G}^{k+1}}(\xi_{0:k})\right)}{\partial \epsilon_{0:k-1} \partial \xi_{0:k-1}}\right], \tag{10.7}$$

$$B_k \triangleq -\mathbb{E}\left[\frac{\partial^2 \log p\left(\mathcal{X}_k \exp^{\wedge}_{\mathcal{G}^{k+1}}(\epsilon_{0:k}) \exp^{\wedge}_{\mathcal{G}^{k+1}}(\xi_{0:k})\right)}{\partial \epsilon_{0:k-1} \partial \xi_{k}}\right], \tag{10.8}$$

$$C_k \triangleq -\mathbb{E}\left[\frac{\partial^2 \log p\left(\mathcal{X}_k \exp^{\wedge}_{\mathcal{G}^{k+1}}(\epsilon_{0:k}) \exp^{\wedge}_{\mathcal{G}^{k+1}}(\xi_{0:k})\right)}{\partial \epsilon_{k} \partial \xi_{k}}\right]. \tag{10.9}$$

Proof: The augmented state $\mathcal{X}_k$ has the following decomposition:

$$\mathcal{X}_k = \begin{bmatrix} \mathcal{X}_{k-1} & 0 \\ 0 & X_k \end{bmatrix}. \tag{10.10}$$

The Lie group derivative of the latter expression with respect to $\epsilon_{0:k}$ leads to the Fisher information matrix $J(\mathcal{X}_k)$:

$$J(\mathcal{X}_k) = \begin{bmatrix} A_k & B_k \\ B_k^T & C_k \end{bmatrix}, \tag{10.11}$$

where A_k, B_k and C_k are defined as in Proposition 3. Since these matrices are invertible, the Schur complement can be applied to the bottom right block which leads to the desired result.

□

10.2.2 . A Posterior Recursive Bound

The posterior information matrix is computed from the joint probability density $p(X_k, Y_k)$, where $Y_k \in \mathcal{H}$ is a measurement matrix and $\mathcal{H}$ a unimodular Lie group of dimension m.

Proposition 4. *The posterior information matrix on an unimodular Lie group* $\mathcal{G}$ *follows the recursive formula:*

$$J(X_{k+1}) = D_k^{22} - D_k^{21}(J(X_k) + D_k^{11})^{-1} D_k^{12} \tag{10.12}$$

where the matrices are defined as:

$$D_k^{11} \triangleq -\mathbb{E}\left[\frac{\partial^2 \log p\left(X_{k+1} | X_k \exp^{\wedge}_{\mathcal{G}}(\epsilon_k) \exp^{\wedge}_{\mathcal{G}}(\xi_k)\right)}{\partial \epsilon_k \partial \xi_k}\right], \tag{10.13}$$

$$D_k^{12} \triangleq -\mathbb{E}\left[\frac{\partial^2 \log p\left(X_{k+1} \exp^{\wedge}_{\mathcal{G}}(\epsilon_{k+1}) | X_k \exp^{\wedge}_{\mathcal{G}}(\xi_k)\right)}{\partial \epsilon_{k+1} \partial \xi_k}\right], \tag{10.14}$$

$$
\begin{aligned}
D_k^{22} \triangleq & -\mathbb{E}\left[\frac{\partial^2 \log p\left(X_{k+1}\exp_{\mathcal{G}}^{\wedge}(\epsilon_{k+1})\exp_{\mathcal{G}}^{\wedge}(\xi_k)|X_k\right)}{\partial\epsilon_{k+1}\partial\xi_k}\right] \\
& -\mathbb{E}\left[\frac{\partial^2 \log p\left(Y_{k+1}|X_{k+1}\exp_{\mathcal{G}}^{\wedge}(\epsilon_{k+1})\exp_{\mathcal{G}}^{\wedge}(\xi_{k+1})\right)}{\partial\epsilon_{k+1}\partial\xi_{k+1}}\right],
\end{aligned}
\tag{10.15}
$$

and $D_k^{21} = D_k^{12^T}$.

Proof. The recursive formulation is computed with the same approach as in the previous section. Considering the state augmented sequence $\mathcal{X}_k$, the measurement augmented sequence $\mathcal{Y}_k$ and proceeding as [78], the joint probability of $\mathcal{X}_k$ and $\mathcal{Y}_k$ can be factorized as:

$$
p(\mathcal{X}_{k+1}, \mathcal{Y}_{k+1}) = p(\mathcal{X}_k, \mathcal{Y}_k)p(X_{k+1}|X_k)p(Y_{k+1}|X_{k+1}). \tag{10.16}
$$

Applying the logarithm gives:

$$
\log p(\mathcal{X}_{k+1}, \mathcal{Y}_{k+1}) = \log p(\mathcal{X}_k, \mathcal{Y}_k) + \log p(X_{k+1}|X_k) + \log p(Y_{k+1}|X_{k+1}). \tag{10.17}
$$

Then, applying the Fisher information matrix leads to:

$$
J(\mathcal{X}_k) = \begin{bmatrix} A_k & B_k & 0 \\ B_k^T & C_k + D_k^{11} & D_k^{12} \\ 0 & D_k^{21} & D_k^{22} \end{bmatrix}, \tag{10.18}
$$

where (A_k, B_k, C_k) are defined in the previous section and the other matrices are given by (10.13) to (10.15). The sequel unfolds exactly as in [78]. Considering the Schur complement of the lower-right block:

$$
\begin{aligned}
J(X_{k+1}) &= D_k^{22} - \begin{bmatrix} 0 & D_k^{21} \end{bmatrix} \begin{bmatrix} A_k & B_k \\ B_k^T & C_k + D_k^{11} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ D_k^{21} \end{bmatrix}, \\
&= D_k^{22} - D_k^{21}\left(D_k^{11} + C_k + B_k^T A_k^{-1} B_k\right)^{-1} D_k^{12},
\end{aligned}
\tag{10.19}
$$

from which the expression of $J(X_k) = C_k + B_k^T A_k^{-1} B_k$ can be identified. □

10.3 . Application to Gaussian Nonlinear Systems

This section applies the previous results to Gaussian systems as they are often encountered in estimation.

10.3.1 . Definition of the system

In the sequel, we consider the left discrete-time system:

$$
\begin{cases} X_{k+1} &= f(X_k, u_{k+1})\exp_{\mathcal{G}}^{\wedge}\left(\nu_k^q\right), \\ Y_k &= h(X_k)\exp_{\mathcal{H}}^{\wedge}\left(\nu_k^r\right) \end{cases}
\tag{10.20}
$$

where the state and the measurement belong to two unimodular matrix Lie groups $\mathcal{G}, \mathcal{H}$, and $u_{k+1} \in \mathbb{R}^c$ is a command input. For the sake of simplicity, the command inputs will be omitted in the notations:

$$f(X_k) \triangleq f(X_k, u_{k+1}). \tag{10.21}$$

Besides, noise vectors are mutually independent and follow centered Normal laws $\nu_k^q \sim \mathcal{N}_{\mathbb{R}^d}(0, Q_k)$, $\nu_k^r \sim \mathcal{N}_{\mathbb{R}^m}(0, R_k)$, where:

$$Q_k = \mathbb{E}\{\nu_k^q \nu_k^{qT}\} \text{ and } R_k = \mathbb{E}\left\{\nu_k^r \nu_k^{rT}\right\}. \tag{10.22}$$

The transition and likelihood densities are assumed to be concentrated Gaussian on the Lie groups $\mathcal{G}$ and $\mathcal{H}$:

$$p(X_{k+1}|X_k) \sim \mathcal{N}_{\mathcal{G}}(X_{k+1}; f(X_k), Q_k), \tag{10.23}$$

$$p(Y_k|X_k) \sim \mathcal{N}_{\mathcal{H}}(Y_k; h(X_k), R_k). \tag{10.24}$$

10.3.2 . Computation of the recursive matrices

The sequel computes the Lie groups CRLB from Proposition 4. The definition of the concentrated Gaussian gives:

$$\log p(X_{k+1}|X_k) = c_1 + \frac{1}{2}\left\|\log_{\mathcal{G}}^{\vee}\left(f(X_k)^{-1}X_{k+1}\right)\right\|_{Q_k}^2, \tag{10.25}$$

where c_1 is a constant. Therefore, applying the generic expression of D_k^{11} from (10.13) leads to:

$$D_k^{11} = \mathbb{E}\left[F_k^T Q_k^{-1} F_k\right], \tag{10.26}$$

where F_k is a Jacobian defined as:

$$F_k = \frac{\partial}{\partial \epsilon} \log_{\mathcal{G}}^{\vee}\left(f(X_k \exp_{\mathcal{G}}^{\wedge}(\epsilon))^{-1} X_{k+1}\right), \tag{10.27}$$

Given the logarithm of the transition density (10.23) and (10.14), and using the previous definition from (10.27), the matrix D_k^{12} is:

$$D_k^{12} = \mathbb{E}\left[F_k^T Q_k^{-1}\right]. \tag{10.28}$$

Since the observation model is Gaussian on $\mathcal{H}$, the log-likelihood writes:

$$\log p(Y_k|X_k) = c_2 + \frac{1}{2}\left\|\log_{\mathcal{H}}^{\vee}\left(h(X_k)^{-1}Y_k\right)\right\|_{R_k}^2, \tag{10.29}$$

where c_2 is a constant. Applying the definition of D_k^{22} from (10.15) gives:

$$D_k^{22} = Q_k^{-1} + \mathbb{E}\left[H_{k+1}^T R_{k+1}^{-1} H_{k+1}\right] \tag{10.30}$$

where H_{k+1} is a Jacobian such that:

$$H_{k+1} = \frac{\partial}{\partial \epsilon} \log_{\mathcal{G}}^{\vee}\left(h(X_{k+1} \exp_{\mathcal{G}}^{\wedge}(\epsilon))^{-1} Y_{k+1}\right). \tag{10.31}$$

information matrix for the system (10.1) is: From the previous developments, the recursive matrices write:

$$D_k^{11} = \mathbb{E}\left[F_k^T Q_k^{-1} F_k\right], \tag{10.32}$$
$$D_k^{12} = \mathbb{E}\left[F_k^T Q_k^{-1}\right], \tag{10.33}$$
$$D_k^{22} = Q_k^{-1} + \mathbb{E}\left[H_{k+1}^T R_{k+1}^{-1} H_{k+1}\right]. \tag{10.34}$$

Note that (10.32) is alike the formulation from [78], with a different definition for Jacobians. In practice, a first approximation of the expectations is given by:

$$\mathbb{E}\left[F_k^T Q_k^{-1} F_k\right] \approx F_k^T Q_k^{-1} F_k, \tag{10.35}$$
$$\mathbb{E}\left[F_k^T Q_k^{-1}\right] \approx F_k^T Q_k^{-1}, \tag{10.36}$$
$$Q_k^{-1} + \mathbb{E}\left[H_{k+1}^T R_{k+1}^{-1} H_{k+1}\right] \approx Q_k^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}, \tag{10.37}$$

where F_k and H_{k+1} are calculated on the true state.

10.4 . Conclusion

This Chapter derives a recursive Cramer-Rao Lower Bound on unimodular Lie groups which is close to the method proposed by Tichavsky [78]. The proposed approach was applied to Gaussian systems on Lie groups, which leads to a direct computation of the bound using the Jacobians introduced in LG-EKF from Chapter 6. Note that if the state or the measurement belongs to $SE_p(3)$, the Jacobians calculus theorem of Chapter 6 holds.
This work on CRLB introduces a new tool to analyze and design algorithms on matrix Lie groups.

Part III

Application to Navigation

11 - Generalities on Navigation

This Chapter details the kinematics equations and the physics for long range and short range navigation taken from [40]. Besides, the last section presents the generic sensor models which were built on purpose for the numerical simulations presented in Chapter 12.

11.1 . Notations and Conventions

The notations and mathematical conventions used in this Chapter follow the same formalism as [40].
Let $[\alpha]$, $[\beta]$ and $[\gamma]$ be three distinct frames. The notations have the following conventions:

- $u^{\gamma}_{\beta\alpha} \in \mathbb{R}^3$ denotes the physical quantity u of frame $[\alpha]$ with respect to frame $[\beta]$ resolved in frame $[\gamma]$.
- $u^{\gamma} \in \mathbb{R}^3$ denotes the physical quantity u resolved in frame $[\gamma]$ which does not relate to a quantity between two frames (e.g. the gravity vector).
- $\Omega^{\gamma}_{\beta\alpha} \in \mathfrak{so}(3)$ denotes the rotation rate matrix of the frame $[\alpha]$ with respect to $[\beta]$ resolved in γ.
- $C^{\beta}_{\alpha} \in SO(3)$ denotes the rotation matrix from $[\alpha]$ to $[\beta]$

These elements have calculus properties described in the sequel. Vectors can be decomposed such as:

$$u^{\gamma}_{\beta\alpha} = u^{\gamma}_{\alpha\gamma} + u^{\gamma}_{\gamma\beta}, \tag{11.1}$$

besides $u^{\gamma}_{\beta\alpha} = -u^{\gamma}_{\alpha\beta}$.
Rotation rate matrices are skew-symmetric and can be denoted from a rotation rate vector $\omega = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^T$ such that:

$$\Omega^{\gamma}_{\beta\alpha} = [\omega^{\gamma}_{\beta\alpha}]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{11.2}$$

Note that rotation rate matrices can be decomposed like vectors:

$$\Omega^{\gamma}_{\beta\alpha} = \Omega^{\gamma}_{\alpha\gamma} + \Omega^{\gamma}_{\gamma\beta}, \tag{11.3}$$

and $\Omega^{\gamma}_{\beta\alpha} = -\Omega^{\gamma}_{\alpha\beta}$ and the matrix-vector product of a skew-symmetric matrix is equivalent to a vectorial product:

$$\forall u \in \mathbb{R}^3 : \omega \times u = [\omega]_{\times} u = \Omega u. \tag{11.4}$$

The composition of two rotation matrices writes:

$$C^{\gamma}_{\alpha} = C^{\gamma}_{\beta} C^{\beta}_{\alpha}, \tag{11.5}$$

and a rotation matrix applied to a vector is such that:

$$u^{\beta}_{\beta\alpha} = C^{\beta}_{\alpha} u^{\alpha}_{\beta\alpha}. \tag{11.6}$$

Besides, the time derivative of a rotation matrix writes:

$$\dot{C}^{\beta}_{\alpha} = C^{\beta}_{\alpha} \Omega^{\alpha}_{\beta\alpha}. \tag{11.7}$$

11.2 . Frames Definition

Long-range navigation requires the introduction of various frames:

- The Earth-Centered Inertial frame (ECI) [i];
- The Earth-Centered Earth-Fixed frame (ECEF) [e];
- The Navigation frame [n];
- The Body frame [b];
- The sensor frame [s];

which will be introduced in the sequel. Also, Figure 11.1 gives an illustration of these frames with respect to the Earth.

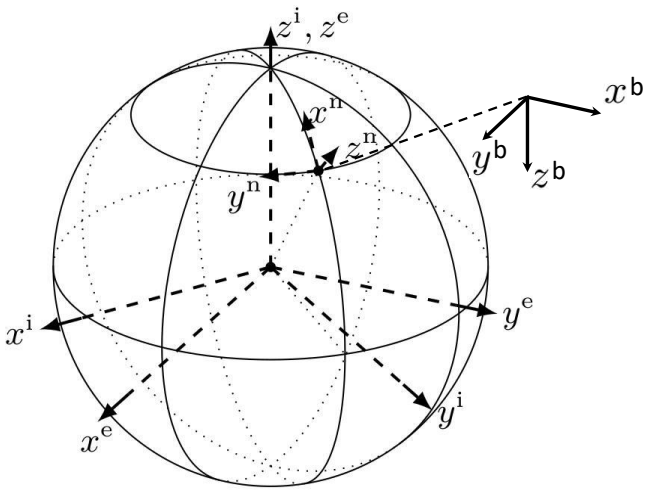


Figure 11.1: Visualization of the ECI [i], ECEF [e], Navigation [n] and Body [b] frames.

11.2.1 . The Earth-Centered Inertial Frame (ECI) denoted [i]

The Earth-Centered Inertial Frame (ECI) is defined as the frame which originates at the Earth center of mass, with the z axis pointing from the Earth's center of mass towards the North pole, and the x, y axis pointing towards two remote stars in the equatorial plan. One can point out that this frame is not strictly inertial since Earth is spinning around the Sun, and the Sun itself is spinning around the Solar system center of mass. But these effects are negligible considering the range and the duration of the missions, and are beyond the sensors' reach.

The Earth-Centered Earth-Fixed frame (ECEF) denoted [e]

The Earth-Centered Earth-Fixed frame (ECEF) is similar to the ECI, except its axis are attached to the rotating Earth. It originates at the Earth's center of mass, its z axis points towards the North pole, its x axis points towards the Greenwich meridian, and y is chosen to make a direct frame. Thus this frame is not inertial and is spinning around the z axis of the ECI frame.

The Local Tangent-Plane Frame (LTP) denoted [l]

The Local Tangent-Plane frame origins at a fixed point on Earth and its axes point to relevant directions defined by the user. This frame can be used for indoor or short range problems, which is often the case in robotics applications.

The Navigation Frame denoted [n]

The Navigation frame is attached to the vehicle and its origin can be an arbitrarily chosen reference point. In our application, this point is chosen as the projection of the Inertial Measurement Unit reference point on the WGS84 ellipsoid. Its axis are moving with respect to the vehicle and are pointing North, East and Down. Hence, this frame is often called NED.

The Body Frame denoted [b]

The Body Frame is attached to the vehicle and its origin can be an arbitrarily chosen reference point. Unlike the navigation frame, its axis are defined with respect to the vehicle. Several conventions exist, and a common choice is to use the IMU as reference point, direct the x axis toward the front, the y axis towards its right wing, and the z axis down.

The Sensor Frames [s]

In practice, each sensor has its own frame assumed fixed with respect to the rigid vehicle's body. Its origin is located on a precise point of the sensor. In practice, these frames differ from the body frame because of internal misalignments. Thus, the link between the body frame and each sensor's frame can be expressed by constant rotation matrices estimated after calibration.

11.3 . Gravity Modeling

The gravity vector resolved in the frame $[\alpha]$ is denoted g^{α}. It is defined as the acceleration sensed for a stationary accelerometer, which includes the gravitational force γ^{α} due to Earth's gravity field and the centrifugal force due to Earth's rotation.

11.3.1 . The navigation frame gravity

The local gravity vector resolved in the navigation frame is denoted g^n. It is directly computed from the vehicle's latitude L_b and geodetic height h_b. In these computations, g^n unique non-zero component is on the Down axis.
The gravity at sea level writes:

$$g_0(L_b) = g\frac{1 + a\sin^2 L_b}{\sqrt{1 - e^2\sin^2 L_b}}. \tag{11.8}$$

with the constants:

$$\begin{cases} g &= 9.7803253359m/s^2, \\ a &= 0.001931853, \\ e &= 0.0818191908425. \end{cases} \tag{11.9}$$

Then, the gravity vector in [n] is:

$$g^n(L_b, h_b) = g_0(L_b)\left[1 - \frac{2}{R_0}\left(1 + f + \frac{\omega_{ie}^2 R_0^2 R_P}{\mu}\right)h_b + \frac{3}{R_0^2}h_b^2\right]. \tag{11.10}$$

Note that the geodetic height h_b has the biggest impact on the gravity's accuracy. The later equation should be used for systems navigating as off a hundred meters above (or below) the sea level.

11.3.2 . The ECEF gravity

The gravity vector in the ECEF can be obtained with the rotation of the gravity vector in [n] from (11.10):

$$g^e(L_b, h_b) = C_n^e(L_b, h_b)g^n(L_b, h_b), \tag{11.11}$$

where the rotation matrix is given by:

$$C_n^e(L_b, h_b) = \begin{bmatrix} -\sin L_b\cos\lambda_b & -sin\lambda_b & -\cos L_b\cos\lambda_b \\ -\sin L_b\sin\lambda_b & \cos\lambda_b & -\cos L_b\sin\lambda_b \\ \cos L_b & 0 & -\sin L_b \end{bmatrix}. \tag{11.12}$$

11.3.3 . The LTP gravity

The gravity vector in the LTP is a constant down component:

$$g^l = \begin{bmatrix} 0 \\ 0 \\ 9.81\mathrm{m/s^2} \end{bmatrix}. \tag{11.13}$$

11.4 . Navigation Equations

11.4.1 . Navigation equations resolved in the navigation frame

The navigation frame enables to work with geodetic coordinates and have easier measurement equations when using sensors working in this very frame. This frame is often chosen for accurate long range - long duration applications.

Attitude equation in [n]

The attitude of the vehicle in the navigation frame is represented by C_b^n. Differentiating this matrix leads to:

$$\dot{C}_b^n = C_b^n \Omega_{nb}^b. \tag{11.14}$$

Splitting the skew symmetric rotation matrix in this expression gives:

$$\dot{C}_b^n = C_b^n (\Omega_{ib}^b - \Omega_{ie}^b - \Omega_{en}^b). \tag{11.15}$$

By applying the change of frame leads to:

$$\dot{C}_b^n = C_b^n \Omega_{ib}^b - (\Omega_{ie}^n - \Omega_{en}^n) C_b^n, \tag{11.16}$$

where Ω_{ib}^b denotes the rotation rate of the body frame with respect to the navigation frame, which is measured from an inertial measurement unit. Then, Ω_{ie}^n denotes the Earth rotation rate:

$$\Omega_{ie}^n = \omega_{ie} \begin{bmatrix} 0 & \sin(L_b) & 0 \\ -\sin(L_b) & 0 & -\cos(L_b) \\ 0 & \cos(L_b) & 0 \end{bmatrix}, \tag{11.17}$$

where $\omega_{ie} = 7.29211510^{-5} rad/s$. Finally, Ω_{en}^n denotes the induced rotation of the vehicle due to its movement on a spherical coordinate frame, and is often called the transport rate matrix:

$$\omega_{en}^n = \begin{bmatrix} \dfrac{v_{eb,E}^n}{R_E(L_b) + h_b} \\ \dfrac{-v_{eb,N}^n}{R_N(L_b) + h_b} \\ \dfrac{-v_{eb,E}^n \tan L_b}{R_E(L_b) + h_b} \end{bmatrix}, \tag{11.18}$$

and $\Omega_{en}^n = [\omega_{en}^n]_\times$.

Velocity equation in [n]

The velocity equation resolved in the navigation frame writes:

$$\dot{v}_{eb}^n = C_b^n f_{ib}^b + g^n(L_b) - (\Omega_{en}^n + 2\Omega_{ie}^n) v_{eb}^n, \tag{11.19}$$

where f_{ib}^b is the specific force measured by the IMU, $2\Omega_{ie}^n v_{eb}^n$ represent the Coriolis force, and $\Omega_{en}^n v_{eb}^n$ the transport rate.

Geographic position equation in [n]

The position equation in the geographic coordinates writes:

$$\begin{cases} \dot{L}_b &= \dfrac{v^n_{en,N}}{R_E(L_b)+h^b} \\ \dot{\lambda}_b &= \dfrac{v^n_{en,E}}{\cos(L_b)(R_N(L_b)+h_b)}. \\ \dot{h_b} &= -v^n_{en,D}, \end{cases} \tag{11.20}$$

Geographic navigation equation set in [n]

The navigation equations resolved in the navigation frame write:

$$\begin{cases} \dot{C}^n_b &= C^n_b\Omega^b_{ib} - (\Omega^n_{ie} - \Omega^n_{en})C^n_b, \\ \dot{v}^n_{eb} &= C^n_b f^b_{ib} + g^n(L_b) - (\Omega^n_{en} + 2\Omega^n_{ie})v^n_{eb}, \\ \dot{L} &= \dfrac{v^n_{eb,N}}{R_N(L_b)+h_b}, \\ \dot{\lambda} &= \dfrac{v^n_{eb,E}}{(R_E(L_b)+\dot{h}_b)\cos(L_b)}, \\ \dot{h}_b &= -v^n_{en,D}. \end{cases} \tag{11.21}$$

11.4.2 . Navigation equations resolved in the ECEF

The ECEF equations involve less non-linear terms than those resolved in the navigation frame:

$$\begin{cases} \dot{C}^e_b &= C^e_b\Omega^b_{ib} - \Omega^e_{ie}C^e_b, \\ \dot{v}^e_{eb} &= C^e_b f^b_{ib} + g^e(x^e_{eb}) + 2\Omega^e_{ie}v^e_{eb}, \\ \dot{x}^e_{eb} &= v^e_{eb}. \end{cases} \tag{11.22}$$

In this set of equations, $2\Omega^e_{ie}v^e_{eb}$ is the Coriolis term and $g^e(x^e_{eb})$ is the ECEF gravity vector.

11.4.3 . Navigation equations resolved in the LTP

The navigation equations in LTP are also referred-to as flat Earth equations. Since this frame is assumed to be inertial, there is no Coriolis term:

$$\begin{cases} \dot{C}^l_b &= C^i_b\Omega^b_{lb}, \\ \dot{v}^l_{lb} &= C^l_b f^b_{lb} + g^l, \\ \dot{r}^l_{lb} &= v^l_{lb} \end{cases} \tag{11.23}$$

11.5 . Generic Sensor Models

This section introduces the generic sensor models used for the simulations of this thesis. The first section introduces the Inertial Measurement Unit (IMU) which enables the time integration of the kinematic model. Section 2 and 3 detail two measurement models based on signals emited by fixed beacons. In the sequel, the true variables (i.e. unbiased and free of noise) are denoted with a bar.

11.5.1 . Inertial Measurement Unit

The Inertial Measurement Unit (IMU) gathers several sensors:

- Three gyrometers (one per axis) measuring the rotation rate ω_{ib}^b of the sensor with respect to the inertial frame;
- Three accelerometers (one per axis) measuring the inertial acceleration f_{ib}^b of the sensor with respect to the inertial frame.

In the case of short-range navigation, the IMU measurements are in the LTP [l].
In this thesis, it is assumed that measurements from the IMU are corrupted by centered Gaussian noises with constant biases:

$$\begin{array}{ll} \omega_{ib}^b & = \bar{\omega}_{ib}^b + b^g + \nu^g, \\ a_{ib}^b & = \bar{a}_{ib}^b + b^a + \nu^a, \end{array} \tag{11.24}$$

where $b^g \in \mathbb{R}^3$ are the gyrometers biases, $b^a \in \mathbb{R}^3$ are the accelerometers biases. In addition, the noises are assumed to be Gaussian and uncorrelated: $\nu^g \sim \mathcal{N}(0, Q^g) \in \mathbb{R}^3$, $\nu^a \sim \mathcal{N}(0, Q^a) \in \mathbb{R}^3$, and $\mathbb{E}\left[(\nu^g)^T \nu^a\right] = 0$.

11.5.2 . Angles of Arrival (AOA)

The Angle of Arrival sensor measures the azimuth θ and elevation φ of the line of sight between a beacon and the vehicle carrying the sensor as illustrated in Figure 11.2. The angular measurements equations are given by:

$$\left\{ \begin{array}{ll} \theta^b & = \arctan 2\left(\Delta_y^b, \Delta_x^b\right), \\ \varphi^b & = \arctan 2\left(-\Delta_z^b, \sqrt{(\Delta_x^b)^2 + (\Delta_y^b)^2}\right), \end{array} \right. \tag{11.25}$$

where $\Delta^b = C_e^b(p_{eb,k}^e - x_{eb}^e)$ is the relative distance between a landmark and the vehicle resolved in the body frame [b] defined in Figure 11.1, and $\arctan 2(y, x)$ is such that $\forall (x, y) \neq (0, 0)$:

$$\arctan 2(y, x) = \left\{ \begin{array}{ll} \operatorname{sign}(y) \arctan\left|\frac{y}{x}\right| & x > 0, \\ \operatorname{sign}(y)\frac{\pi}{2} & x = 0, \\ \operatorname{sign}(y)(\pi - \arctan\left|\frac{y}{x}\right|) & x < 0. \end{array} \right. \tag{11.26}$$

In this thesis, the noise model of the Angle of Arrival sensor is assumed to be Gaussian:

$$\begin{bmatrix} \theta^b \\ \varphi^b \end{bmatrix} = \begin{bmatrix} \bar{\theta}^b \\ \bar{\varphi}^b \end{bmatrix} + \nu^r, \tag{11.27}$$

where $\nu^r \sim \mathcal{N}(0, R)$ and $R \in \mathbb{R}^{2\times 2}$.

11.5.3 . Doppler Velocity or Frequency of Arrival (FOA)

A Doppler sensor computes the frequency shifts of signals emitted from a beacon on the ground. The Doppler shift Δf of a signal emitted at frequency f^0 is such that:

$$\Delta f = \frac{v_{DOP}}{c} f^0, \tag{11.28}$$

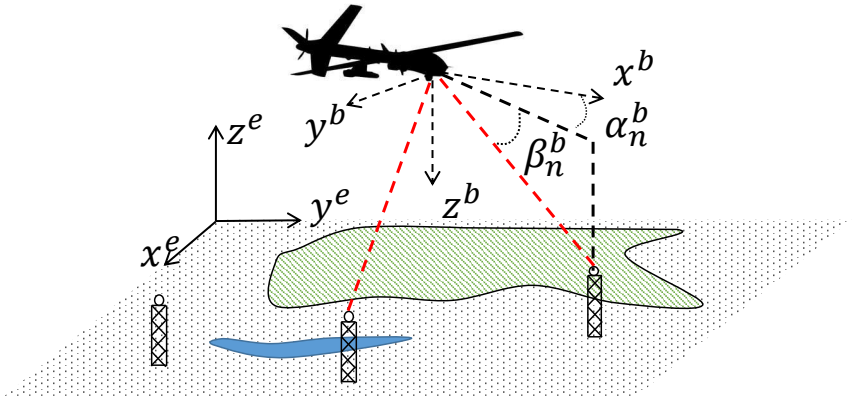


Figure 11.2: Representation of angle of arrival measurements

where c is the speed of light in vacuum and v_{DOP} is the projection of the vehicle velocity on the line of sight of the beacon described by the unitary vector $\overrightarrow{e}$:

$$v_{\mathrm{DOP}} = \overrightarrow{v_{eb}^{e}} \cdot \overrightarrow{e}. \tag{11.29}$$

The line of sight vector is obtained with the scaled distance vector pointing towards the beacon:

$$\vec{e} = \frac{\overrightarrow{p_{eb}^{e}} - \overrightarrow{x_{eb}^{e}}}{||\overrightarrow{p_{eb}^{e}} - \overrightarrow{x_{eb}^{e}}||}, \tag{11.30}$$

where p_{eb}^{e} denotes the position of the beacon. In this thesis, the noise model of the Doppler sensor is assumed to be Gaussian:

$$v_{\mathrm{DOP}} = \bar{v}_{\mathrm{DOP}} + \nu^r, \tag{11.31}$$

where $\nu^r \sim \mathcal{N}(0, \sigma_R)$.

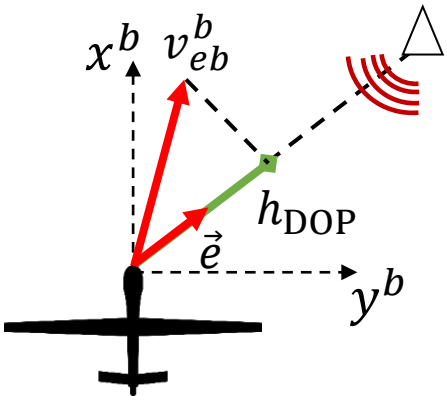


Figure 11.3: Representation of Doppler measurements

12 - Filters Comparative Study

This Chapter provides a numerical analysis of the filters introduced in this thesis. They are compared to state-of-the-art algorithms, either designed on Lie groups or on the Euclidean space. The filters will be compared on several simulation scenarios to assess their performance on different operational contexts. Section 12.1 provides the evaluation criteria for the filters. Then, Section 12.2 tests LG-LPF from Chapter 7 in unimodal and multimodal scenarios. Section 12.3 compares LG-KPKF with LG-EKF respectively from Chapters 7 and 9. Then, Section 12.4 studies the revisited LG-EKF from Chapter 6 and the Section 12.5 discusses the results of the Posterior Cramer-Rao Lower Bound on Lie groups introduced in Chapter 10.

12.1 . Evaluation Criteria

This section first describes a convergence criteria based on the Cramer-Rao Lower Bound. Then, it introduces the Root Mean Square Error formulation for variables on Lie groups and on the Euclidean space. Eventually, it describes the Average Root Mean Square Error (ARMSE) which quantifies the accuracy of a filter.

12.1.1 . Cramer-Rao Lower Bound and Convergence Rate

The robustness of a filter is assessed by the convergence rate, which is the ratio (in percent) of convergent runs with respect to the total number of Monte Carlo runs.
A run is considered convergent if the mean position component of the state is contained inside the confidence ellipsoid Γ_k computed from the Cramer-Rao Lower Bound (CRLB) [78]:

$$\Gamma_k = \left\{ x^e_{eb,k} | (x^e_{eb,k} - \widehat{x}^e_{eb,k})^T J_k (x^e_{eb,k} - \widehat{x}^e_{eb,k}) \leq \kappa \right\}, \tag{12.1}$$

where the threshold κ is chosen from the test $p(\chi^2(d) \leq \kappa^2) = 0.99$, d is the dimension of the state vector and J_k the Fisher information matrix for the position at step k.
The interest of using the J to determine the convergence is that it is independent from the measurements. Besides, [78] enables a recursive computation.

12.1.2 . Root Mean Square Error

The accuracy and speed of convergence of a filter is assessed with the Root Mean Square Error (RMSE). The Lie group Root Mean Square Error (LG-RMSE) is defined as:

$$\mathrm{LG-RMSE}(k) = \sqrt{\frac{1}{N_{\mathrm{conv}}} \sum_{m=1}^{N_{\mathrm{conv}}} \left\| \log^\vee_{\mathcal{G}} \left(\mu_k^{-1} \widehat{\mu}_{k|k,m} \right) \right\|_2^2}, \tag{12.2}$$

where N_{conv} is the number of convergent runs according to (12.1). The log-Euclidean error vector ϵ_k is decomposed with respect to each component of the state matrix:

$$\text{Left case: } \epsilon_k = \log^\vee_{\mathcal{G}} \left(\mu_k^{-1} \hat{\mu}_{k|k} \right), \tag{12.3}$$

$$\text{Right case: } \epsilon_k = \log^\vee_{\mathcal{G}} \left(\hat{\mu}_{k|k} \mu_k^{-1} \right). \tag{12.4}$$

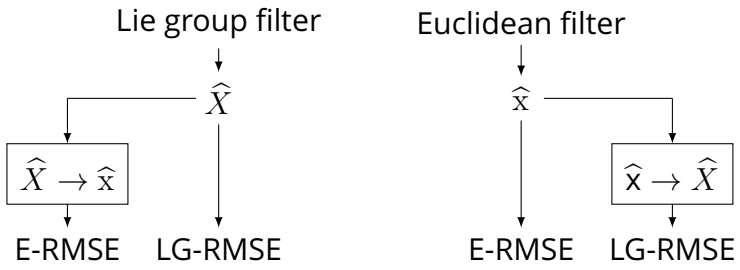


Figure 12.1: Illustration of the estimation scheme. The LG-RMSE and E-RMSE are calculated for both UKF and LG-ItEKF using the exact transformation (12.7) between a state matrix from $\mathrm{SE}_p(3)$ and a state vector from $\mathbb{R}^d$. The true trajectory enables to compute the LG-RMSE, the E-RMSE and their respective CRLB.

The Euclidean Root Mean Square Error (E-RMSE) is computed from the state vector x_k:

$$\mathrm{E-RMSE}(k) = \sqrt{\frac{1}{N_{\mathrm{conv}}} \sum_{m=1}^{N_{\mathrm{conv}}} \|\mathrm{x}_k - \widehat{\mathrm{x}}_{k,m}\|_2^2}. \tag{12.5}$$

The Euclidean error is denoted δ_k:

$$\delta_k = \mathrm{x}_k - \widehat{\mathrm{x}}_{k|k}. \tag{12.6}$$

Note that it is possible to derive an Euclidean error from a Lie group matrix by extracting its variables into a vector and *vice versa*:

$$X = \begin{bmatrix} C & u_1 & \cdots \\ 0_{1\times 3} & 1 & \\ \vdots & & \ddots \end{bmatrix} \leftrightarrow \mathrm{x} = \begin{bmatrix} \Theta \\ u_1 \\ \vdots \end{bmatrix}, \tag{12.7}$$

where Θ represents the Euler attitude angles of the rotation matrix C. Similarily, a log-Euclidean error can be computed from an usual filter defined on the Eudlidean space according to the process described in Figure 12.1 .

12.1.3 . Average Root Mean Square Error

The Average Root Mean Square Error (ARMSE) is computed from the RMSE over the time period T of N_T steps ranging from k_{ini} to k_{end}:

$$\mathrm{ARMSE} = \frac{1}{N_T} \sum_{k=k_{\mathrm{ini}}}^{k_{\mathrm{end}}} \mathrm{RMSE}(k). \tag{12.8}$$

In the sequel, the ARMSE is computed from the Euclidean RMSE (E-RMSE) to compare the accuracy of the filters.

12.2 . Laplace Particle Filter on Lie Groups

This section describes the numerical results obtained with Laplace Particle Filter on Lie Groups (LG-LPF) on two navigation scenarios presented in [25] and [26]. The novelty of this filter is to represent the probability densities on Lie groups and resample the particles with an accurate proposal density. The first scenario is unimodal, which enables to test the unimodal resampling strategy from Section 8.2. Then, the second scenario is designed to be multimodal, which requires the use of the strategy introduced in Section 8.3.

12.2.1 . Long Range Angles-Only Scenario

In this first scenario, LG-LPF from Chapter 7 is compared with two filters on the Euclidean space: the Laplace Particle Filter (LPF) [70] and the Regularized Particle Filter (RPF) [60]. The state follows a long-range kinematics model in the ECEF, and the aiding measurements are angles of arrival described in Section 11.5.3.

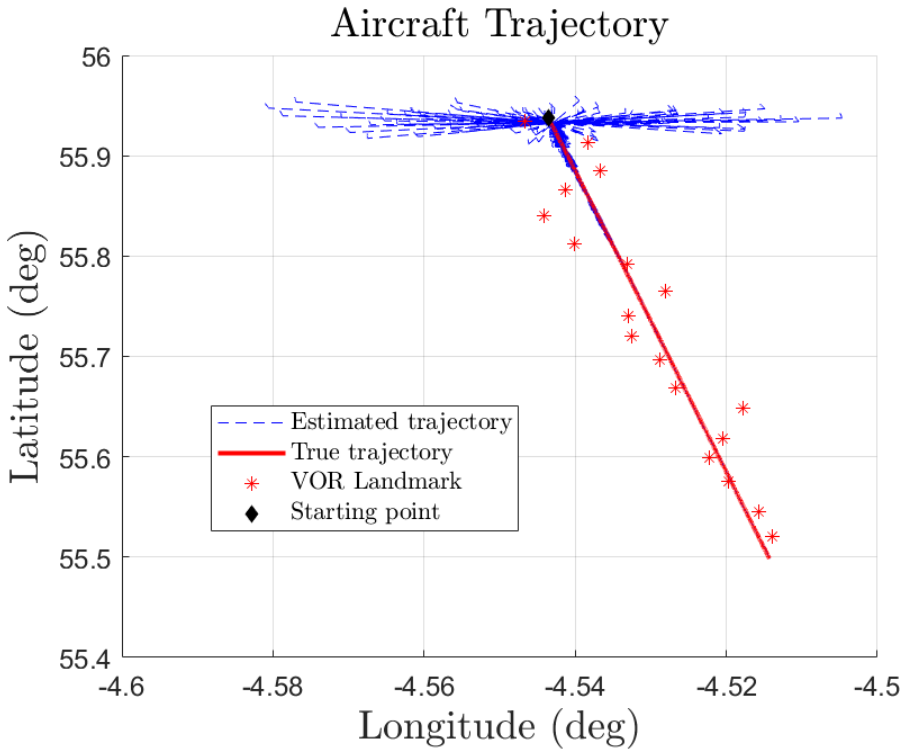


Figure 12.2: Illustration of the vehicle's true trajectory (red) with the trajectory of every Monte Carlo run (blue dashed), and the landmarks ($*$).

The vehicle has a straight line trajectory illustrated in Figure 12.2. In this scenario, the state matrix estimated with LG-LPF belongs to $\mathrm{SE}_2(3)$ and the particles at time k are defined as:

$$\forall i \in [1, N_p] : X_k^i = \begin{bmatrix} C_{b,k}^{e,i} & v_{be,k}^{e,i} & x_{be,k}^{e,i} \\ 0_{1,3} & 1 & 0 \\ 0_{1,3} & 0 & 1 \end{bmatrix}. \tag{12.9}$$

On the other hand, the state vector estimated by LPF and RPF is given by:

$$\forall i \in [1, N_p] : \mathrm{x}_k^i = \begin{bmatrix} \Theta_{b,k}^{e,i} \\ v_{be,k}^{e,i} \\ x_{be,k}^{e,i} \end{bmatrix}. \tag{12.10}$$

The choice of $\mathrm{SE}_2(3)$ allows couplings between the attitude matrix C_b^e and the other variables, which is expected to bring more consistency during the updates.
The propagation of the particles is obtained with a time discretization of time step dt. Thus, the propagation step of LG-LPF writes:

$$\begin{cases} C_{b,k+1|k}^{e,i} = C_{b,k}^{e,i} \exp_{\mathrm{SO(3)}} \left(\mathrm{dt}(\Omega_{ib}^b - C_{e,k}^{b,i} \Omega_{ie}^e C_{b,k}^{e,i}) + \nu_{r,k}^i \right), \\ v_{eb,k+1|k}^{e,i} = v_{eb,k}^{e,i} + \mathrm{dt} \left(C_k^i f_{ib}^b + g^e(x_k^i) - 2\Omega_{ie}^e v_k^i \right) + \nu_{v,k}^i, \\ x_{eb,k+1|k}^{e,i} = x_{eb,k}^{e,i} + \mathrm{dt} v_{eb,k}^{e,i} + \nu_{x,k}^i, \end{cases} \tag{12.11}$$

where the stochastic processes are Gaussian:

$$\forall k > 0, \forall i \in [1, N_p], \nu_k^i = \begin{bmatrix} \nu_{r,k}^i & \nu_{v,k}^i & \nu_{x,k}^i \end{bmatrix}^T \sim \mathcal{N}(0, Q_k). \tag{12.12}$$

The measurements are angles of arrival from known landmarks displayed in Figure 12.2. Hence, the measurements belong to the space $\mathcal{H} = \mathbb{R}^2$, which is a Lie group such that $\log_{\mathcal{H}} = \exp_{\mathcal{H}} = I_2$. Thus the likelihood of the angle of arrival measurement model writes:

$$p(Y_{k+1}|X_{k+1}^i) = \frac{\exp\left(-\frac{1}{2} \left\| Y_{k+1} - h(X_{k+1}^i) \right\|_{R_{k+1}}^2 \right)}{\sqrt{(2\pi)^2 \det[R_{k+1}]}}, \tag{12.13}$$

where h is computed for each landmark with (11.25) and R is the measurement noise matrix.

$$H = \begin{bmatrix} [\Delta^b(X)]_\times & 0_{3,3} & -I_3 \end{bmatrix} \begin{bmatrix} -\frac{\Delta_y^b}{\rho^2} & \frac{\Delta_x^b}{\rho^2} & 0 \\ \frac{\Delta_x^b \Delta_z^b}{\rho ||\Delta^b||^2} & \frac{\Delta_y^b \Delta_z^b}{\rho ||\Delta^b||^2} & \frac{-\rho}{||\Delta^b||^2} \end{bmatrix}, \tag{12.14}$$

where $[u]_\times$ is the skew-symmetric matrix of the vector u.
The simulations are run for 100 Monte Carlo realizations and their parameters are gathered in Table 12.1. The Average Root Mean Square Error (ARMSE) for the last minute of flight is displayed in Table 12.2. On the left side of Table 12.2, the comparison between RPF, LPF and LG-LPF shows that Laplace resampling significantly increases the convergence rate. This was expected since the particles are resampled in accurate areas, which improves the weights consistency and slows the filter's degeneracy.
In addition, comparing LG-LPF and LPF shows that the first approach substancially increases the accuracy of the filter on every state variable. This improvement was also expected as Lie groups enable a better representation of rotation matrices. Besides, the group $\mathrm{SE}_2(3)$ involves curvature terms in the densities which behave like a natural constraint for the particles. Also, the

Sensor Parameters			
Sensor rates (Hz)	IMU: 50Hz	AOA: 1Hz	
IMU noise (1σ)	Gyrometer: $2\,\mathrm{deg}/h$	Accel: $10^{-3}m/s^2$	
RDF noise (1σ)	Azimuth: $0.6°$	Elevation: $0.6°$	
Filter parameters			
Initial uncertainties	Attitude	Velocity	Position
Nominal (1σ)	$0.115°$	$10ms^{-1}$	$1km$
Poor (1σ)	$11.50°$	$50ms^{-1}$	$10km$
Process noise (1σ)	Attitude: $20°$	Velocity: $10^{-2}m/s$	Position $10^{-2}m$
Update noise	Azimuth: $2.8°$	Elevation: $2.8°$	
Resampling threshold		$N_{\mathrm{th}} = 0.6N_p$	

Table 12.1
Simulation and filters parameters for the two scenarios.

Filter	**RPF**	**LPF**	**LG-LPF**	**LG-LPF**	**LG-LPF**	**LG-LPF**
N_p	500	500	500	100	500	1000
Convergent runs	47%	100%	100%	78%	84%	82%
Position (m)	66.0	12.6	7.89	17.4	8.86	8.78
Velocity (m/s)	2.21	0.95	0.69	2.1	0.85	0.71
Yaw (°)	0.048*	0.299	0.092	0.311	0.082	0.041
Pitch (°)	0.032*	0.142	0.088	0.248	0.099	0.079
Roll (°)	0.026*	0.101	0.072	0.282	0.092	0.073

Table 12.2: Comparison of the ARMSE for RPF, LPF and LG-LPF with a nominal initialization (left side) and a poor initialization (right side).
* These values are not relevant considering the low convergence rate.

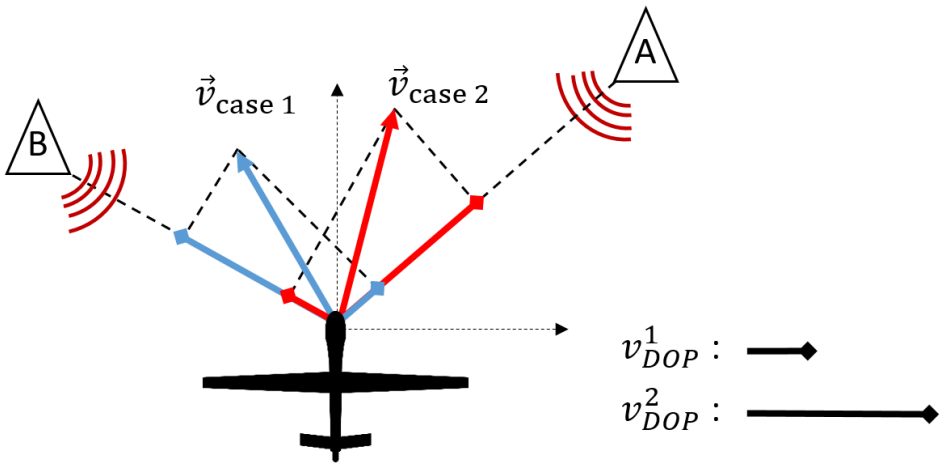


Figure 12.3: Illustration of an abiguous Doppler measurement in the case of two emitters.

optimization based on LG-ItEKF copes with the geometry of the group, leading to a smoother process.
The right side of Table 12.2 presents the results of LG-LPF with different numbers of particles in the specific case of a poor initialization. These results show that the filter converges even with a low number of particles.
Note that only LG-LPF converged with these large initialization errors. Besides, the result of RPF regarding the ARMSE of attitude variables are strongly affected by the survivor bias: only the runs having a small attitude uncertainty were able to converge, which leads to lower ARMSE than LG-LPF. This means that RPF is not robust to attitude errors, which is confirmed by the low convergence rate.

12.2.2 . Short Range Ambiguous Doppler Scenario

The scenario presented in the sequel involves ambiguous Doppler measurements leading to a multimodal likelihood. The ambiguity of Doppler measurements is illustrated in Figure 12.4. The generalized LG-LPF presented in Chapter 7 is suited to such scenarios as it introduces a multimodal proposal density to resample the particles at the most probable areas. It is compared to LPF having the same resampling strategy. These results were published in [25]. The propagation equations are given by the short-range navigation model described in Chapter 11. They are defined with respect to the Local Tangent Plane frame (LTP) denoted [l]:

$$\begin{cases} C^{l,i}_{b,k+1} = C^{l,i}_{b,k} \exp_{\mathrm{SO(3)}} \left(\mathrm{dt}(\Omega^b_{lb,k} + \nu^i_{r,k}) \right), \\ v^{l,i}_{lb,k+1} = v^{l,i}_{lb,k} + \mathrm{dt} \left(C^{l,i}_{b,k} f^b_{lb,k} + g^l + \nu^i_{v,k} \right), \\ x^{l,i}_{lb,k+1} = x^{l,i}_{lb,k} + \mathrm{dt}(v^{l,i}_{lb,k} + \nu^i_{x,k}). \end{cases} \tag{12.15}$$

Regarding the measurement model, the Doppler shift for one signal is in $\mathbb{R}$ and the left Lie group Jacobian for the j^{th} signal is given by:

$$H^j_{k+1} = \begin{bmatrix} 0_{1,3} & \frac{(\Delta^j_{k+1})^T}{||\Delta^j_{k+1}||} C^l_{b,k+1} & -J_{\Delta^j_{k+1}} C^l_{b,k+1} \end{bmatrix}. \tag{12.16}$$

Sensor Parameters			
Sensor rates (Hz)	IMU: 50Hz	Doppler: 1Hz	
IMU noise (1σ)	Gyrometer: $2\,\mathrm{deg}/h$	Accelerometer: $10^{-3}m/s^2$	
Doppler noise (1σ)	$v_{\mathrm{DOP}} = 10m/s$		
Altimeter noise (1σ)	$20m$		
Filter parameters			
Initial uncertainties	Attitude	Velocity	Position
1(σ)	$0.6°$	$10ms^{-1}$	$1km$
Process noise (1σ)	Attitude: $2°/h$	Velocity: $10^{-2}m/s$	Position $10^{-6}m$
Update noise	10 m/s		
Resampling threshold		$N_{\mathrm{th}} = 0.1N_p$	

Table 12.3
Simulation and filters parameters for the two scenarios.

Where $\Delta^j = p^{e,j}_{eb,k+1} - x^e_{eb,k+1} = [\Delta^j_{x,k}, \Delta^j_{y,k}, \Delta^j_{z,k}]^T$, and :

$$J_{\Delta^j_{k+1}} = \begin{bmatrix} \frac{1}{||\Delta^j_{k+1}||} - \frac{(\Delta^j_{x,k+1})^2}{||\Delta^j_{k+1}||^3} & -\frac{\Delta^j_{x,k+1}\Delta^j_{y,k+1}}{||\Delta^j_{k+1}||^3} & -\frac{\Delta^j_{x,k+1}\Delta^j_{z,k+1}}{||\Delta^j_{k+1}||^3} \\ -\frac{\Delta^j_x\Delta^j_{y,k+1}}{||\Delta^j_{k+1}||^3} & \frac{1}{||\Delta^j_{k+1}||} - \frac{(\Delta^j_{y,k+1})^2}{||\Delta^j_{k+1}||^3} & -\frac{\Delta^j_{y,k+1}\Delta^j_{z,k+1}}{||\Delta^j_{k+1}||^3} \\ -\frac{\Delta^j_{x,k+1}\Delta^j_{z,k+1}}{||\Delta^j_{k+1}||^3} & -\frac{\Delta^j_{y,k+1}\Delta^j_{z,k+1}}{||\Delta^j_{k+1}||^3} & \frac{1}{||\Delta^j_{k+1}||} - \frac{(\Delta^j_{z,k+1})^2}{||\Delta^j_{k+1}||^3} \end{bmatrix} \tag{12.17}$$

The measurement Jacobian for all the beacons writes:

$$H = \begin{bmatrix} H_1^T & \dots & H_N^T \end{bmatrix}^T . \tag{12.18}$$

Assuming that the measurement vector writes $y = [y_1, ..., y_{N_b}]^T$, where N_b represents the number of beacons, each component of y represents the Doppler shift of a signal coming from a specific beacon. Since the Doppler sensor is not able to assign the signal to its source, a measurement component $y_j, j \in [1, N_b]$ can be assigned to every beacon $p^j, j \in [1, N_b]$. Thus, there exist N_b factorial equiprobable combinations of beacons, and each one of them generates a mode in the likelihood:

$$g(X_{k+1}) \propto \sum_{n=1}^{N_b!} \exp\left(-\frac{1}{2}||y^n_{k+1} - h^n(X_{k+1})||^2_{R_{k+1}}\right). \tag{12.19}$$

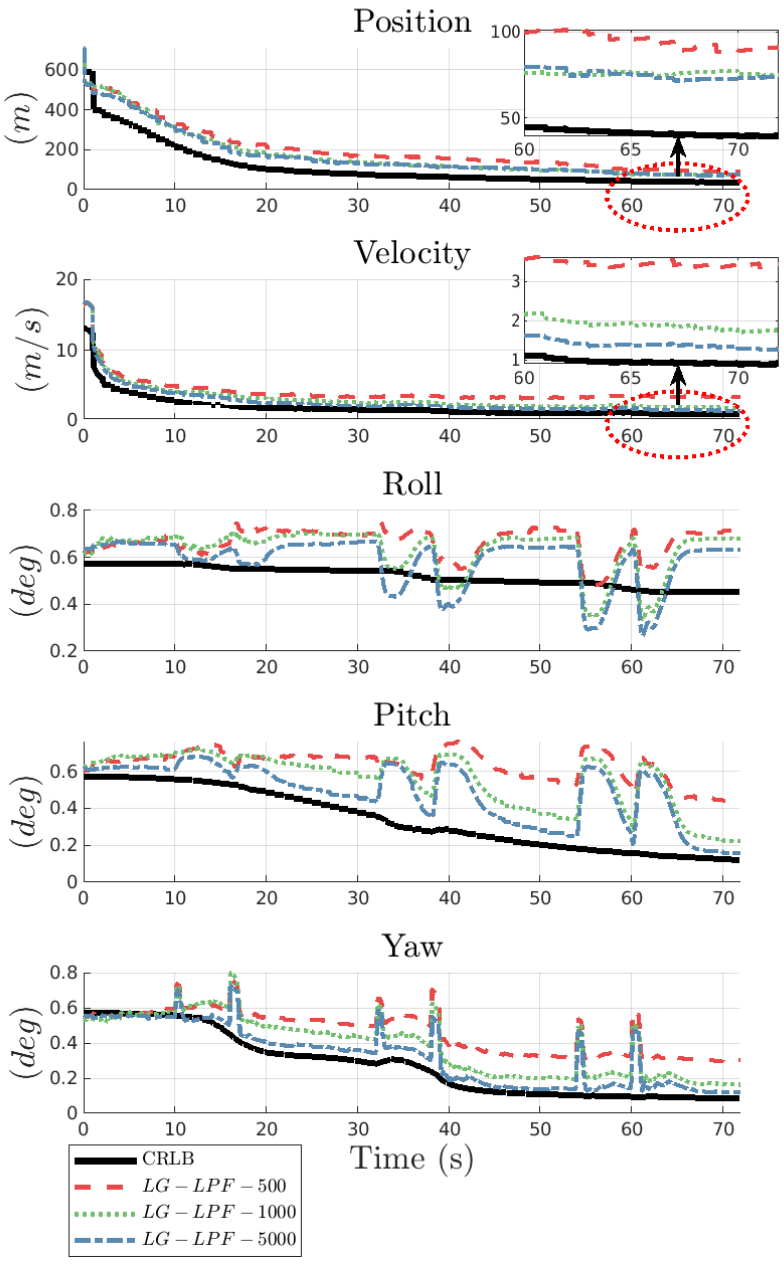


Figure 12.4: RMSE for LG-LPF with 500, 1000 and 5000 particles compared to the Cramer Rao Lower Bound (CRLB). The roll has low observability in this scenario, as indicated by the slow convergence of the CRLB on this variable. The synchronized peaks on the attitude variables result from integration errors due to strong dynamics. The position and velocity plots correspond to the norm L_2 of the RMSE of their three axes.

Filters	LG-LPF			LPF	
N_p	500	1000	5000	1000	5000
Position (m)	91.6	75.9	73.3	235.2	189.8
Velocity (m/s)	3.43	1.82	1.35	8.90	5.31
Roll (°)	0.704	0.673	0.625	4.400	3.531
Pitch (°)	0.453	0.256	0.189	1.233	0.782
Yaw (°)	0.305	0.173	0.130	0.577	0.574
Convergent runs	86%	94%	98%	64%	92%

Table 12.4: Comparison of LG-LPF and LPF. The ARMSE is computed for the last 10 seconds of simulation and only for the convergent runs in the sense of (12.1). The position and velocity values correspond to the norm L_2 of the ARMSE of their three axes.

The results detailed in Table 12.4 show that LG-LPF substantially improves the robustness and precision of LPF for every variable. This improvement was expected since the error function defined on Lie group is accurate on a large domain, which improves the cluster's consistency. Besides, the local optimizations benefit from the properties of LG-ItEKF which is known to be more robust and accurate than ItEKF.
Similarly to the previous scenario, LG-LPF shows improved performance for a lower amount of particles. This is linked to the improved accuracy of the importance function and the natural constraint involved by the group geometry. The plot of the Root Mean Square Error (RMSE) compared to the Euclidean Cramer Rao Lower Bound (CRLB) shows that the filter is close to optimal performance even with 500 particles. Besides, the CRLB is computed in the case where the source of the signals are known, whereas the filter cannot associate a given measurement with a source. This result demonstrates the capacity of this method to obtain good performance without knowing the source of the signals. The peaks on the RMSE of the attitude angles in Figure 12.4 are due to integration errors as they occur during maneuvers in the trajectory.

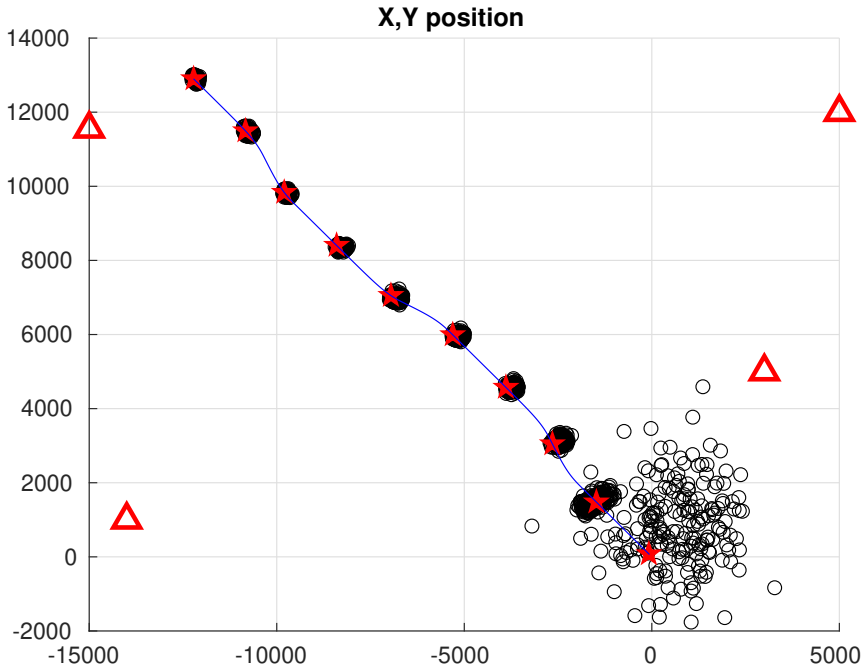


Figure 12.5: Illustration of the true trajectory horizontal position (blue line), the four beacons (triangles) and the particle cloud represented at different times. The red stars represent the true state at the time where the particles were captured.

This scenario confirms the interest of LG-LPF over LPF, as it shows substantial improvements on the accuracy and robustness for a lower amount of particles. This is due to the Lie group representation of the variables which enables accurate computations between the rotation matrices. Besides, the correlations and the curved geometry of the group behaves like a natural constraint on the particles, which limits LG-LPF divergence.

12.3 . Kalman-Particle Kernel Filter on Lie groups

This section compares Kalman-Particle Kernel Filter on Lie groups (LG-KPKF) described in Chapter 9 with LG-EKF described in Chapter 6. This comparison aims to assess the interest of LG-KPKF over LG-EKF in the case where computational resources are not a limitation. Indeed, LG-KPKF is expected to require more computations as it is based on several local LG-EKF. On the other hand, the Gaussian mixture representation of the estimated density with the Laplace method is expected to substancially improve the accuracy of LG-KPKF compared to LG-EKF.
The comparison is based on a short-range angles-only navigation scenario which parameters are described in Table 12.5. Each particle mean is propagated according to the flat Earth model

Sensor Parameters				
Sensor rates (Hz)	IMU:	100 Hz	Angles:	1 Hz
IMU noise (1σ)	Gyro:	$2^\circ/h$	Accel:	$10^{-4}\ m/s^2$
Angles noise (1σ)	Azimuth:	0.6°	Elevation:	0.6°
Filter parameters				
Initial errors ($\mathbf{1\sigma}$)	Attitude 11.50°	Velocity $10\ m/s$	Position $1\ km$	Landmarks $10\ m$
Process noise (1σ)	Attitude 0.6°	Velocity $10^{-4}\ m/s$	Position $\emptyset$	Landmarks $\emptyset$
Update noise (1σ)	Azimuth	1.8°	Elevation	1.8°
Resampling threshold		$N_{\text{th}} = 0.6N_p$		

Table 12.5
Simulation and filters parameters for the two scenarios.

equations:

$$\begin{cases} C_{b,k+1}^{l,i} = C_{b,k}^{l,i} \exp_{\mathrm{SO}(3)}\left(\mathrm{d}t\Omega_{lb,k}^{b}\right), \\ v_{lb,k+1}^{l,i} = v_{lb,k}^{l,i} + \mathrm{d}t\left(C_k^i f_{lb,k}^{b} + g^l\right), \\ x_{lb,k+1}^{l,i} = x_{lb,k}^{l,i} + \mathrm{d}t v_{lb,k}^{l,i}, \end{cases} \tag{12.20}$$

and every particle covariance is propagated according to the Ricatti equation as detailed in Chapter 9:

$$P_{k+1|k}^{i} = F_k P_{k|k}^{i} F_k^T + Q_{k+1}, \tag{12.21}$$

where F_k is such that:

$$F_k = \begin{bmatrix} -\Omega_{lb,k}^{b} & 0 & 0 & 0 \\ -[f_{lb,k}^{b}]_\times & -\Omega_{lb,k}^{b} & 0 & 0 \\ 0 & I_3 & -\Omega_{eb,k}^{b} & 0 \\ 0 & 0 & 0 & -I_{N_l} \otimes \Omega_{lb,k}^{b} \end{bmatrix}. \tag{12.22}$$

Note that $\otimes$ denoted the Kronecker product, and F_k is the same for every particle as the kinematics equation model is group-affine and the framework described in Chapter 9 Section 3 applies.
The simulation scenario was run for 100 Monte-Carlo runs. Table 12.6 gathers the ARMSE of LG-EKF and LG-KPKF for 100, 500 and 1000 particles. LG-KPKF shows an improvement on the convergence rate over LG-EKF, even with 100 particles. This was expected as the prior density has a more accurate representation due to the Gaussian mixture.
Furthermore, LG-KPKF has a higher accuracy than LG-EKF on every state variable. This improvement is due to the mixture representation and Laplace resampling which guides the particles

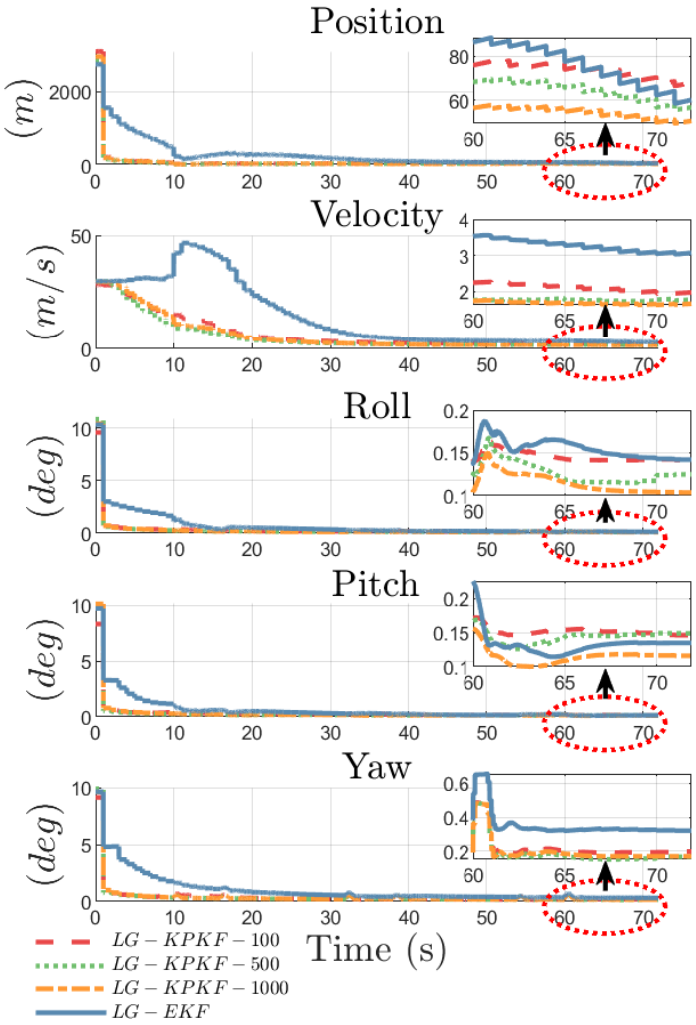


Figure 12.6: Comparison between LG-KPKF and LG-EKF Euclidean RMSE.

Filters	LG-KPKF			LG-EKF
N_p	100	500	1000	N.A.
Position (m)	71.3	58.1	52.7	69.13
Velocity (m/s)	2.04	1.72	1.66	3.15
Yaw (°)	0.195	0.160	0.171	0.326
Pitch (°)	0.150	0.133	0.115	0.131
Roll (°)	0.142	0.100	0.107	0.149
Convergent runs	90%	98%	98%	87%

Table 12.6: Comparison of the ARMSE and convergence rate of the LG-KPKF and LG-EKF for different amounts of particles. ARMSE are computed starting 60s of simulation and only on the convergent runs in the sense of (12.1).

near the *Maximum A Posteriori*.

The comparison of RMSE plots in Figure 12.6 is consistent with the results of Table 12.6, as LG-KPKF RMSE plots are lower than LG-EKF at convergence.
Focusing at the beginning of the plots, LG-KPKF has a faster convergence than LG-EKF on every variable. Regarding the position and the attitude angles, LG-KPKF is close to the Cramer-Rao Lower Bound at the first measurement, where LG-EKF takes around 30 seconds to converge (e.g. 30 measurement updates). This improvement is due to the Laplace method for resampling which computes a proposal density close to the posterior density. Such result shows the interest of LG-KPKF in application where a fast convergence of the estimation filter is critical.
Therefore, LG-KPKF shows competitive results compared to LG-EKF, and presents improvements on the asymptotic accuracy and convergence rate. Besides, its fast convergence and robustness make it suited for challenging initialization scenarios. Arguably, LG-KPKF has a higher computational cost than LG-EKF. Nevertheless it can be used in targeted phases of an estimation process.

12.4 . Extended Kalman Filter on Lie groups

This section focuses on the revisited formulation of LG-EKF(A) described in Chapter 6. It is compared to the formulation of LG-EKF(B) proposed in [15] and Euclidean Extended Kalman Filter.
The scenario is based on a long-range trajectory, which is similar to the one of a commercial aircraft. The aiding sensor measures the angle of arrival of signals emitted by VOR beacons which are visible within a radius of 50km around the aircraft. The position of the beacons is assumed to be known with an uncertainty of 10m, and they are located at the position of active VOR in UK.

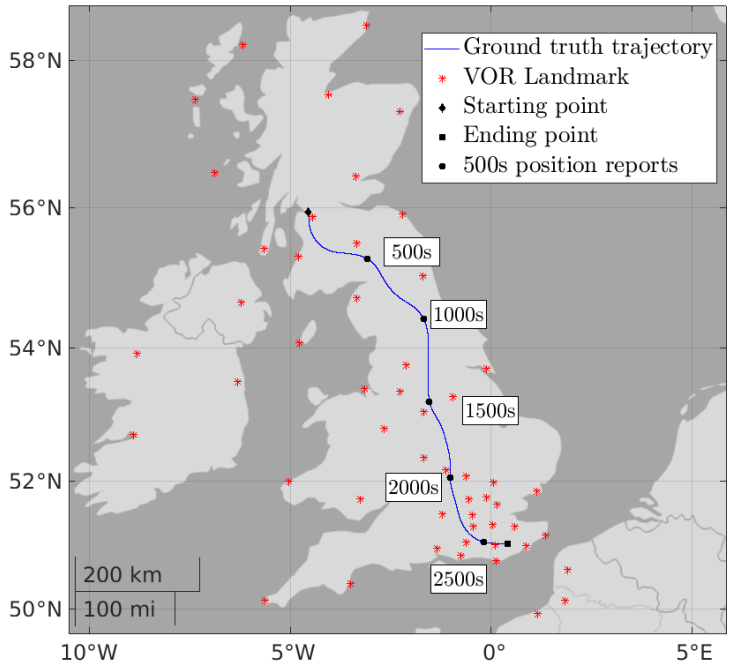


Figure 12.7: Illustration of the long-range trajectory (blue) with the VOR landmarks (red marks) used for Angle of Arrival updates.

In this scenario, the state variables are represented in the ECEF. The state matrix of LG-EKF(A) with the formulation proposed in Chapter 6 is:

$$X = \begin{bmatrix} C_b^e & v_{eb}^e & x_{eb}^e & ba & bg & p_1^e & \cdots & p_{N_b}^e \\ 0_{4+N_b,3} & & & [I_{4+N_b}] & & & & \end{bmatrix}, \tag{12.23}$$

where the frames are detailed in Chapter 11, $p_1^e, \cdots, p_{N_b}^e$ denote the position of the beacons on the map, and (ba, bg) respectively denote accelerometer and gyrometer biases. On the other hand, LG-EKF(B) from [15] splits the state matrix in two diagonal blocks with the rotation matrix in the top left corner and the other state variables in the bottom-right block:

$$X = \begin{bmatrix} C_b^e & 0_{3,1} & & \cdots & & & & 0_{3,1} \\ 0_{3,3} & I_3 & v_{eb}^e & x_{eb}^e & ba & bg & p_1^e & \cdots & p_{N_b}^e \\ 0_{4+N_b,3} & 0_{4+N_b,3} & & & [I_{4+N_b}] & & & & \end{bmatrix}. \tag{12.24}$$

Note that this structure is equivalent to the space $\mathrm{SO}(3) \times \mathbb{R}^{d-3}$.
The state vector of EKF is:

$$\mathrm{x} = \begin{bmatrix} \Theta_{be}^T & {v_{eb}^e}^T & {x_{eb}^e}^T & ba^T & bg^T & {p_1^e}^T & \cdots & {p_{N_b}^e}^T \end{bmatrix}^T, \tag{12.25}$$

where Θ_{be} represents the Euler attitude angles.
The propagation of the mean are based on the ECEF kinematics model:

$$\begin{cases} C^e_{b,k+1} = C^e_{b,k} \exp^{\wedge}_{\mathrm{SO}(3)} \left(\mathrm{dt}[\omega^b_{ib,k+1} + w^q_g - bg_k - \omega^b_{ie,k}] \right), \\ v^e_{eb,k+1} = v^e_{eb,k} + \mathrm{dt} \left(C^b_{e,k}(f^b_{ib,k+1} - ba_k) + g^e(x^e_{eb,k+1}) - a^b_{ie,k} \right), \\ x^e_{eb,k+1} = x^e_{eb,k} + \mathrm{dt} v^e_{eb,k}, \\ ba_{k+1} = ba_k, \\ bg_{k+1} = bg_k, \\ p^e_{n,k+1} = p^e_{n,k}, n \in [1, N_b]. \end{cases} \tag{12.26}$$

The propagation Jacobian of LG-EKF(A) is:

$$F_k = \begin{bmatrix} A_1 & 0_{3,3} & 0_{3,3} & 0_{3,3} & A_2 \\ A_3 & A_4 & 0_{3,3} & A_5 & 0_{3,3} \\ 0_{3,3} & \mathrm{dt}E_k & E_k & 0_{3,3} & 0_{3,3} \\ 0_{3,3} & 0_{3,3} & 0_{3,3} & E_k & 0_{3,3} \\ 0_{3,3} & 0_{3,3} & 0_{3,3} & 0_{3,3} & E_k \end{bmatrix}, \tag{12.27}$$

where $\Omega_k \triangleq \mathrm{dt}[\omega^b_{k+1} - bg_k - C^e_{b,k+1}\omega^e_{ie}]$, $E_k = \exp^{\wedge}_{\mathcal{G}}(\Omega_k)$, and the blocks $A_i \in \mathbb{R}^3, i \in [1,5]$ are such that:

$$\begin{aligned} A_1 &= E_k, \\ A_2 &= -\mathrm{dt}\varphi_{\mathrm{SO}(3)}(-\Omega_k)C^e_b, \\ A_3 &= -dtE_k[a^b]_\times, \\ A_4 &= E_k(I_3 - 2\mathrm{dt}[C^b_e\omega^e_{ie}]_\times), \\ A_5 &= -\mathrm{dt}E_kC^e_b. \end{aligned} \tag{12.28}$$

On the other hand, the propagation Jacobian of LG-EKF(B) is calculated according to [15].
Regarding the update step, the Jacobian of the AOA measurement model is computed from the results of Chapter 6. Let $\Delta^b : G \to \mathbb{R}^3$ be the smooth function such that $h(X) = h(\Delta^b(X))$:

$$\Delta^b(X) = \begin{bmatrix} \Delta^b_1(X) \\ \vdots \\ \Delta^b_{N_b}(X) \end{bmatrix} \triangleq \begin{bmatrix} C^b_e(p^e_{eb,1} - x^e_{eb}) \\ \vdots \\ C^b_e(p^e_{eb,N_b} - x^e_{eb}) \end{bmatrix}. \tag{12.29}$$

Hence the derivation chain rule for the left case gives:

$$H = \left.\frac{\partial h(\Delta)}{\partial \Delta}\right|_{\Delta^b(X)} J_\Delta, \tag{12.30}$$

where J_Δ is such that:

$$J_\Delta \triangleq \left.\frac{\partial \Delta^b(X \exp_{\mathrm{SE}_p(3)}(\epsilon))}{\partial \epsilon}\right|_{\epsilon=0}. \tag{12.31}$$

Then, taking $\Delta^b = \begin{bmatrix} \Delta^b_x & \Delta^b_y & \Delta^b_z \end{bmatrix}^T$ and $\rho = \sqrt{(\Delta^b_y)^2 + (\Delta^b_x)^2}$:

$$\left.\frac{\partial h(\Delta)}{\partial \Delta}\right|_{\Delta=\Delta^b(X)} = \begin{bmatrix} -\frac{\Delta^b_y}{\rho^2} & \frac{\Delta^b_x}{\rho^2} & 0 \\ \frac{\Delta^b_x\Delta^b_z}{\rho||\Delta^b||^2} & \frac{\Delta^b_y\Delta^b_z}{\rho||\Delta^b||^2} & \frac{-\rho}{||\Delta^b||^2} \end{bmatrix}, \tag{12.32}$$

Filters	LG-EKF(A)	LG-EKF(B)[15]	EKF
Position (m)	**74.8**	104.5	195.3
Velocity (m/s)	**1.06**	1.69	2.90
Roll (mrad)	**0.66**	1.24	1.77
Pitch (mrad)	**0.50**	0.88	1.17
Yaw (mrad)	**0.91**	0.91	1.79
Accel Bias ($\mathrm{m/s^2}$)	**0.022**	0.024	0.033
Gyro Bias (mrad/s)	**0.149**	0.285	0.288

Table 12.7: Comparison of the ARMSE for the Euclidean Extended Kalman Filter (EKF), the Lie Group Extended Kalman Filter (B) from [15] and the Lie-Group Extended Kalman Filter proposed in this paper (A). The ARMSE is computed on the whole trajectory for 50 Monte Carlo runs.

and J_Δ writes:

$$J_\Delta = \begin{bmatrix} [\Delta_1^b(X)]_\times & 0 & -I_3 & 0 & I_3 & & 0 \\ \vdots & \vdots & \vdots & \vdots & & \ddots & \\ [\Delta_{N_b}^b(X)]_\times & 0 & -I_3 & 0 & 0 & & I_3 \end{bmatrix}, \tag{12.33}$$

where the zeros refer to null blocks of consistent size according to the state and measurement model.

The results gathered in Table 12.7 give the ARMSE of LG-EKF(A), LG-EKF(B) and EKF on the whole trajectory and for 50 Monte Carlo runs. Both formulations of LG-EKF present higher accuracy than EKF. This was expected as Lie groups enable a better formulation of the attitude with a rotation matrix, which improves the estimation accuracy of all the state variables.

Comparing the two Lie group filters, LG-EKF(A) shows a substantial accuracy improvement on every state variable compared to LG-EKF(B), except for the yaw for which both filters have the same performance. This behavior was also expected since the error on the matrix Lie group $\mathrm{SE}_p(3)$ involves couplings between the attitude and the vectorial variables. These couplings enable the filter to get most of the available information and lead to more consistent state and covariance updates. Note that this improvement is linked to the state representation that involves a different error function.

An unexpected improvement is observed in the estimation of the gyroscope bias. The plot of E-RMSE for Euclidean bias in Figure 12.8 shows that LG-EKF(A) enables the asymptotic convergence of the gyroscope bias. This point is interesting as the biases are often difficult to estimate [38]. The comparison of the RMSE on the other variables show that the representation of LG-EKF(A) greatly limits the divergence of the filter in the area where the measurements provide limited information, around 1000s after the beginning of the simulation. This effect can be observed on the attitude variables in Figure 12.10. The limited divergence of LG-EKF(A) filter is due to the couplings involved by its error function.

This study demonstrates the interest of LG-EKF over EKF for long-range applications. Besides, the formulation LG-EKF(A) introduced in Chapter 6 leads to significant improvements compared to LG-EKF(B), which are due to the different group representation.

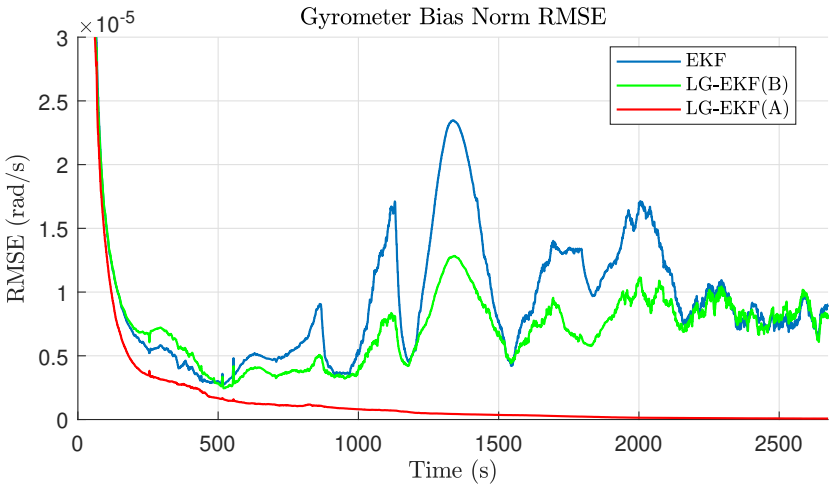


Figure 12.8: Root Mean Square Error of Gyrometer Biases

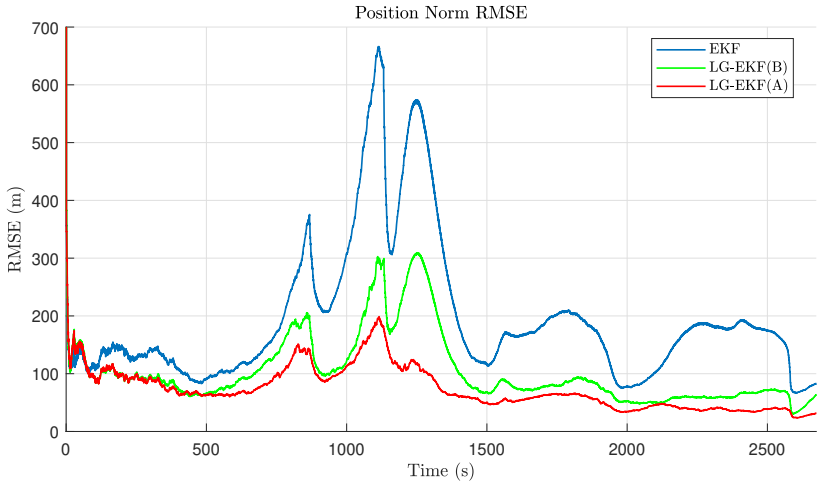


Figure 12.9: Root Mean Square Error of the position norm.

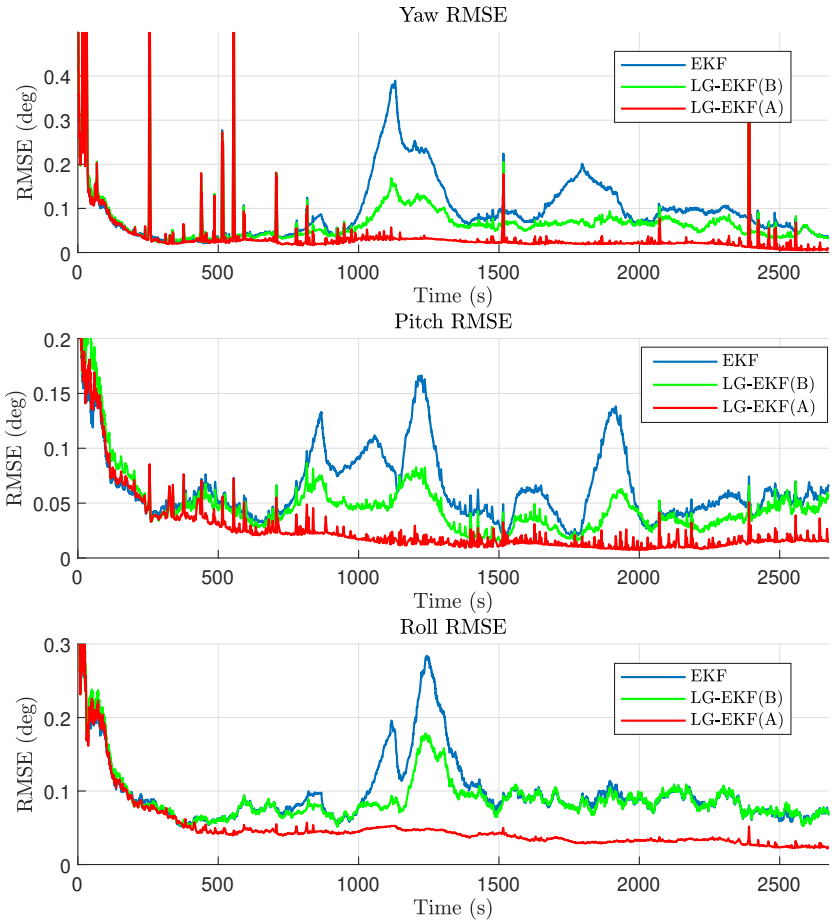


Figure 12.10: Root Mean Square Error of the attitude represented by Euler angles.

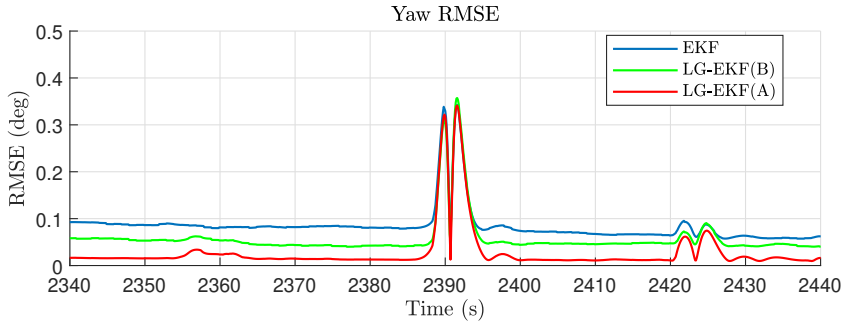


Figure 12.11: Zoom on a peak of the Root Mean Square Error of the attitude. The peaks are due to integration errors.

The representation of the state is an important aspect of this thesis. Mathematically, all the state representations are equivalent as they are based on the same propagation and update models. In practice, some choice of representation lead to better results as they enable strong correlations between the state variables.

12.5 . Cramer-Rao Lower Bound

This section illustrates the developments of Chapter 10 on the recursive Posterior Cramer Rao Lower Bound on matrix Lie groups. The results are based on a short-range navigation scenario with angle of arrival and Doppler updates. Hence, the system is Gaussian and uses the equations 10.32. The Cramer-Rao Lower Bound on Lie groups (LG-CRLB)from Chapter 10 is compared to the Euclidean Cramer-Rao Lower Bound from [78] in Figure 12.12 for the attitude variables, Figure 12.14 for the velocity variables and Figure 12.15 for the position variables. The consistency of the bound on Lie groups is assessed by comparison with the Root Mean Square Error (RMSE) from two filters:

- An Iterated Extended Kalman Filter on Lie groups (LG-ItEKF) designed on $\mathrm{SE}_p(3)$ [15, 16]. Its state is represented by the matrix X_k from (12.7);

- An Unscented Kalman Filter (UKF) designed on $\mathbb{R}^d$ [80]. Its state is represented by the vector x_k from (12.7);

The LG-RMSE for LG-ItEKF is calculated directly from the estimated state matrix and the E-RMSE for LG-ItEKF is calculated from the estimated state vector obtained from the transformation (12.7). Similarily, E-RMSE from UKF is calculated directly from the estimated state vector, and the LG-RMSE for UKF is obtained from (12.7). This process is illustrated for both filters in Figure 12.1. The attitude angles RMSE are displayed in Figure 12.12. For both Euclidean and Lie group case, the RMSE shows a good consistency with the CRLB. Also, the first attitude components of the LG-CRLB and the E-CRLB are similar. This was expected since the rotation matrix convention starts its first rotation around the *z* axis, which defines the yaw. Hence, the yaw and the first component of the attitude angles log-Euclidean error define the same quantity. The corresponding plots are overplayed with a zoom in Figure 12.13. Although these variables are mathematically identical, the LG-CRLB plot is slightly higher than the of the E-CRLB. This phenomenon is connected to the log-Euclidean error formulation, which is exact, while the error on Euler angles is accurate only at the vicinity of the true state. This discrepancy is known to yield over-optimistic observability assessment which is part of the inconsistency problem of the EKF. Indeed, similarly to the Euclidean Extended Kalman Filter, the Euclidean bound is too optimistic.
In the case of the velocity errors displayed in Figure 12.14, the two bounds have very different behaviors. Again, the Lie groups bound shows a strong sensitivity to the turns of the vehicles compared to the Euclidean bound, which is due to the couplings implied by the log-Euclidean error between the velocity and the attitude angles. The bound on the position variables shows a

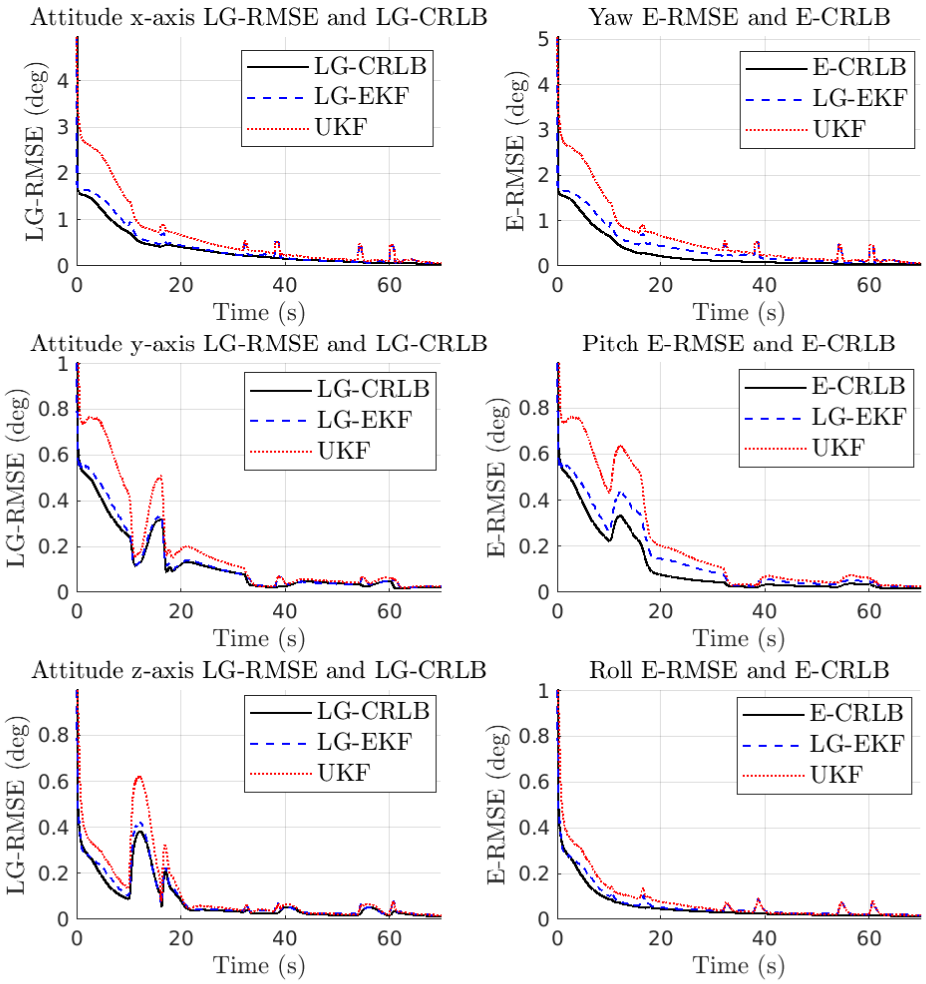


Figure 12.12: Illustration of the Euclidean and Lie groups attitude angles CRLB on the navigation scenario. The bounds show a similar behavior for the yaw due to the rotation matrix convention.

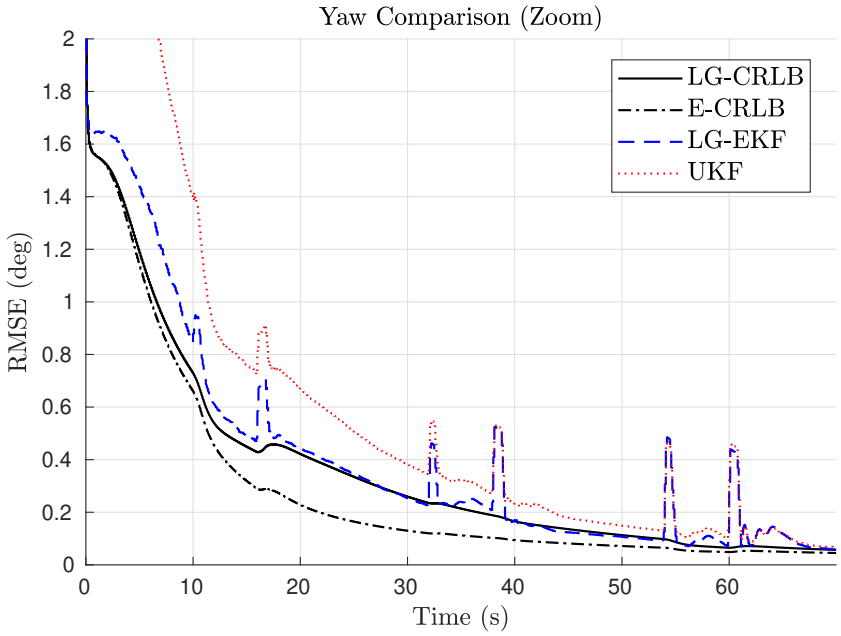


Figure 12.13: Comparison of the Yaw and the first component of the attitude angles log-Euclidean error (zoom). Although the representation of this variable is mathematically equivalent for Euler angles and rotations matrices due to the angles convention. The LG-CRLB shows a higher level, implying a better consistency without false observability.

similar behavior in Figure 12.15. This was predictable as the log-Euclidean error on the position is also coupled with the attitude angles.

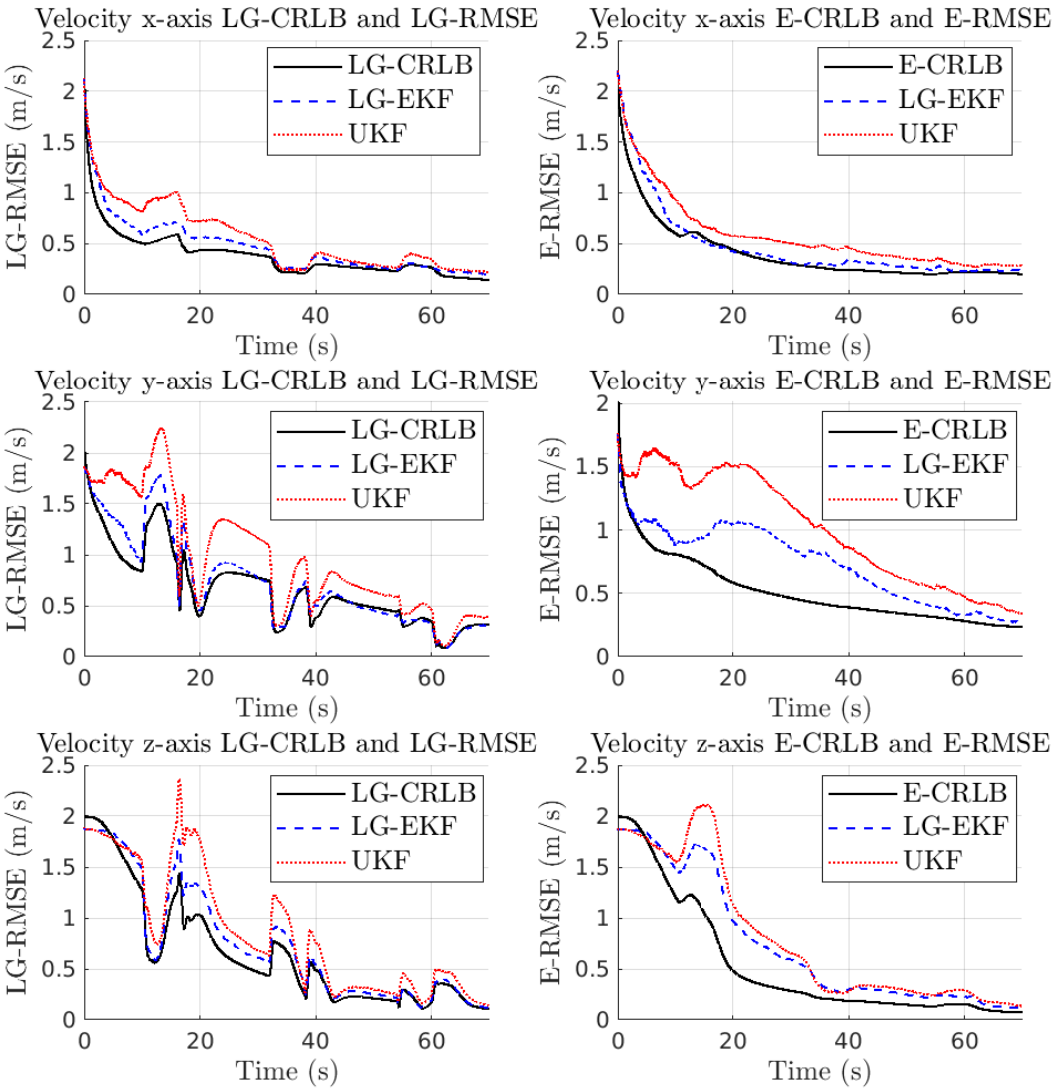


Figure 12.14: Comparison of the Lie group (left) and Euclidean (right) CRLB for the velocity errors.

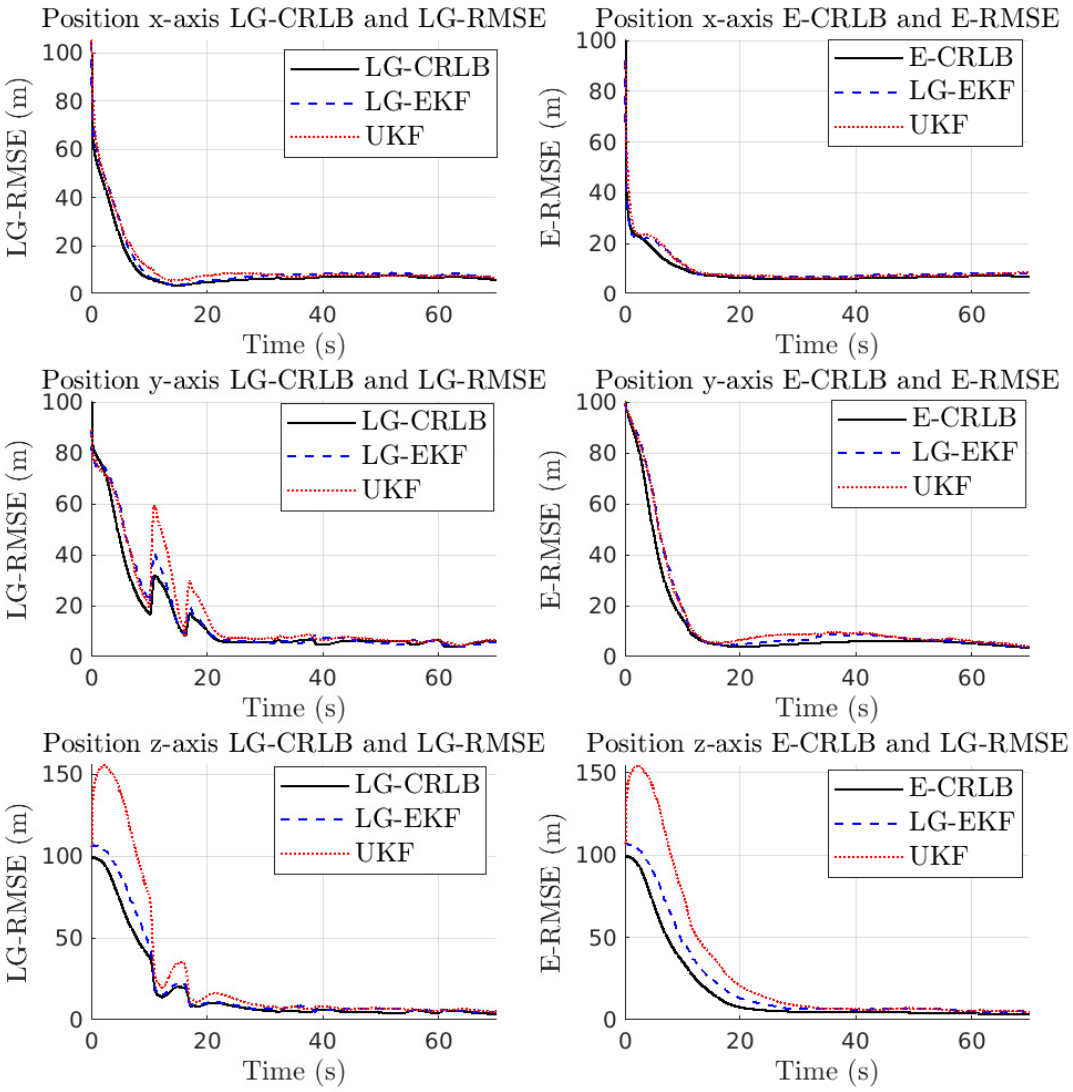


Figure 12.15: Comparison of the Lie group (left) and Euclidean (right) CRLB for the position errors.

13 - Conclusion of the Thesis

Filtering techniques on Lie groups gained increasing interest over the last decade in the fields of transportation and robotics. These approaches solve strongly nonlinear problems with improved accuracy and robustness compared to usual filters designed with variables belonging to the Euclidean space. This thesis has explored the new field of particle filtering on Lie groups by focusing on several aspects of these algorithms.

- First, the Bayes Filter is solved in the case where prediction and measurement models are described by Gaussian noises on Lie groups. This leads to a formulation of Extended Kalman Filter on Lie groups (LG-EKF) which is similar to the version introduced by Bourmaud et al. , but it differs in the way that the noises and linearizations are handled. In particular, a formulation for the Jacobians on the Special Euclidean group $\mathrm{SE}_p(3)$ is detailed for accurate and simplified calculus. This revisited LG-EKF presents substantial improvements in the estimation process.
- Then, the Bayes Filter on Lie groups is solved in the general case by approximating the estimated density with a sum of weighted Dirac functions, leading to Particle Filter on Lie groups (LG-PF). The formulation of filter is similar to its couterpart on the Euclidean space. In practice, LG-PF only differs from PF on the sampling step and the computation of the statistical moments.
- The thesis also developed the resampling step of LG-PF with an approach similar to Laplace Particle Filter (LPF), which consists in resampling the particles according to an accurate proposal density. This method was first extended to Lie groups assuming that the densities are unimodal (i.e. they present only one predominant peak), leading to Laplace Particle Filter on Lie Groups (LG-LPF). Then, this approach is generalized to the case where the probability densities are multimodal (i.e. they present several peaks), by approximating the prior density and the likelihood with Gaussian mixtures on the group. The optimization processes for the proposal density are based on an Iterated Extended Kalman Filter on Lie groups (LG-ItEKF) which enables a simplified optimization framework with accurate results. LG-LPF demonstrated substantial improvements compared to equivalent filters on the Euclidean space, especially on the robustness and accuracy. Besides, LG-LPF works with a very limited number of particles, which greatly limits its computational load.
- Another contribution of this thesis is the Kalman Particle Kernel Filter on Lie Groups (LG-KPKF) which is tailored to solve strongly nonlinear problems with large dimensions. Such problems can be difficult to solve with EKF because of the nonlinearities and with PF because of the curse of dimensionality. The core idea of LG-KPKF is to approximate the densities with a sum of weighted Gaussian functions instead of Dirac functions. This approach enables to process each particle mean and covariance with a local LG-EKF and

the weight is processed with a Lie group particle filter. This filter is derived by solving the Bayes Filter assuming that the estimated density is represented by a sum of weighted Gaussians on the group in the unimodal case. LG-KPKF was also studied in the case where the propagation model is group-affine [7], leading to substantial simplifications in the propagation step. This approach demonstrated improved accuracy and robustness compared to LG-EKF even with a small number of particles. Another important result is its fast convergence, which makes it interesting for applications requiring high accuracy .

- The thesis introduces a recursive formulation for the Cramer-Rao Lower Bound on unimodular matrix Lie groups. Although previous works derived the CRLB on Lie groups [11, 48], the recursive formula on these spaces, is a novelty and enables an efficient computation of the bound. The computation of this bound shows a consistent behavior compared the Lie Groups Root Mean Square Error (LG-RMSE) of a LG-ItEKF.

The filtering framework developed in this thesis provides a robust and flexible solution to nonlinear estimation problems on Lie groups. The theoretical foundations of the particle filters and recursive Cramer-Rao lower bound on Lie groups have been established, and their effectiveness has been demonstrated through various simulations. The results highlight the advantages of using particle filters on Lie groups for navigation. In particular, their ability to accurately estimate the state of nonlinear systems and track the underlying dynamics has been demonstrated, even with a very low amount of particles compared to usual approaches. It important implications for a wide range of navigation applications, including autonomous vehicles, robotics, and satellite navigation. Also, the particle filters developed in this thesis can find broader applications in image and signal processing.
In future works, the framework developed in this thesis can be extended to incorporate additional factors such as general density functions for the noises processes or refined optimization solutions. Additionally, the methods can be applied to more complex and challenging navigation problems, such as multi-sensor fusion and cooperative navigation. A robustness study will have to be conducted to evaluate, for example, the impact of the Gaussian noise assumption on the behavior of the proposed filters.
In conclusion, this thesis provides a significant contribution to the field of nonlinear and non-Gaussian estimation and demonstrates the potential of particle filters on Lie groups for solving complex estimation problems.

A - Usual Lie Groups

A.1 . The Special Orthogonal Group $\mathrm{SO(d)}$

The Special Orthogonal group of dimension $d \in \mathbb{N}$ is defined by:

$$\mathrm{SO}(d) = \left\{ C \in \mathbb{R}^{d \times d} | CC^T = I_d, \det[C] = 1 \right\}. \tag{A.1}$$

The case $\mathrm{SO}(3)$ represents the group of 3d rotation matrices which is broadly used in robotics and navigation. Its algebra is $\mathfrak{so}(d)$ which denotes the group of real skew-symmetric $d \times d$ matrices:

$$\mathfrak{so}(d) = \left\{ M = [m_{ij}]_{i,j \in [1,d]} \, | m_{ij} = -m_{ji} \right\}. \tag{A.2}$$

A.1.1 . Algebra isomorphisms

Let $u \in \mathbb{R}^3$ be a vector denoting attitude angles. The associated skew-symmetric matrix is:

$$[u]^\wedge = [u]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}, \tag{A.3}$$

and its reciprocal isomorphism is denoted $[M]^\vee$ where $M \in \mathfrak{so}$

A.1.2 . Exponential and Logarithm

The Rodrigues formula gives a closed-lood formulation of the exponential and logarithm on $\mathrm{SO}(3)$.
Let $u \in \mathbb{R}^3$:

$$\exp^\wedge_{\mathrm{SO}(3)}(u) = I_3 + \frac{\sin(\|u\|)}{\|u\|}[u]_\times + \frac{1 - \cos(\|u\|)}{\|u\|^2}[u]^2_\times. \tag{A.4}$$

Let $C \in \mathrm{SO}(3)$ be a rotation matrix. The logarithm writes:

$$\log^\vee_{\mathrm{SO}(3)}(C) = \frac{\theta}{2\sin(\theta)}[C - C^T]^\vee, \tag{A.5}$$

where:

$$\theta = \arccos\left(\frac{\mathrm{tr}(C) - 1}{2}\right). \tag{A.6}$$

A.1.3 . Adjoints

The algebra adjoint writes:

$$\mathrm{ad}_{\mathrm{SO}(3)}(u) = [u]^\wedge, \tag{A.7}$$

and the group adjoint writes:

$$\mathrm{Ad}_{\mathrm{SO}(3)}(C) = C. \tag{A.8}$$

A.2 . The Special Euclidean Group $\mathbf{SE_p(d)}$

The Special Euclidean group of dimension $d \in \mathbb{N}$ with $p \in \mathbb{N}$ vectors is defined by:

$$SE_p(d) = \left\{ \begin{bmatrix} C & v_1 \cdots v_p \\ 0_{p,d} & I_p \end{bmatrix}, C \in \mathrm{SO}(d), v_1, \cdots, v_p \in \mathbb{R}^d \right\}, \tag{A.9}$$

and its algebra by:

$$\mathfrak{se}_p(d) = \left\{ \begin{bmatrix} M & u_1 \cdots u_p \\ 0_{p,d} & u_p \end{bmatrix}, M \in \mathfrak{so}(d), u_1, \cdots, u_p \in \mathbb{R}^d \right\}, \tag{A.10}$$

The sequel describes useful elements for state estimation on $SE_p(3)$.

A.2.1 . Algebra isomorphism

Let $u \in \mathbb{R}^{3+3p}$ such that $u = \begin{bmatrix} u_C^T & u_1^T & \cdots & u_p^T \end{bmatrix}^T$ with $u_C,\ u_1,\ \cdots,\ u_p \in \mathbb{R}^3$:

$$[\mathbf{u}]^\wedge = \begin{bmatrix} [u_C]_\times & u_1 \cdots u_p \\ 0_{p,3} & I_p \end{bmatrix}. \tag{A.11}$$

The reciprocal map is denoted $[\,]^\vee$.

A.2.2 . Exponential and Logarithm

Let $u_C,\ u_1\ \cdots\ u_p \in \mathbb{R}^3$ be a set of vectors and $u = \begin{bmatrix} u_C^T & u_1^T & \cdots & u_p^T \end{bmatrix}^T$. The exponential on $SE(3)$ writes:

$$\exp^\wedge_{SE_p(3)}(u) = \begin{bmatrix} \exp^\wedge_{\mathrm{SO}(3)}(u_C) & \Phi_{\mathrm{SO}(3)}(u_C)u_1 \cdots \Phi_{\mathrm{SO}(3)}u_p \\ 0_{p,3} & I_p \end{bmatrix}, \tag{A.12}$$

where $\Phi_{\mathrm{SO}(3)}$ is the group Jacobian defined in (3.27) for $SO(3)$ [3][65]:

$$\Phi_{\mathrm{SO}(3)}(u_C) = I_3 + \frac{(1-\cos(\|u_C\|))}{\|u_C\|^2}[u_C]_\times + \frac{\|u_C\| - \sin(\|u_C\|)}{\|u_C\|^3}[u_C]^2_\times \tag{A.13}$$

Let $X \in SE_p(3)$ be a matrix defined as (A.9). The logarithm in its closed form writes:

$$\log_{SE_p(3)}(X) = \begin{bmatrix} \log_{\mathrm{SO}(3)}(C) & \varphi_{\mathrm{SO}(3)}(C)u_1 \cdots \varphi_{\mathrm{SO}(3)}(C)u_p \\ 0_{p,3} & I_p \end{bmatrix}, \tag{A.14}$$

where $\varphi_{\mathrm{SO}(3)}$ is the inverse group Jacobian defined in (3.28) for $SO(3)$:

$$\varphi_{\mathrm{SO}(3)}(C) = I_3 - \frac{1}{2}w + \frac{1}{\theta^2}\left[1 - \frac{\theta\sin(\theta)}{2(1-\cos(\theta))}\right]w^2, \tag{A.15}$$

with $w = \log_{\mathrm{SO}(3)}(C)$, and $\theta = \arccos\left(\frac{\mathrm{tr}(C)-1}{2}\right)$.

A.2.3 . Adjoints

Let $X \in \mathfrak{se}_p(3)$ be a matrix such that:

$$X = \begin{bmatrix} M & u_1 \cdots u_p \\ 0_{p,d} & 0_p \end{bmatrix} \tag{A.16}$$

The algebra adjoint writes:

$$\mathrm{ad}_{SE_p(3)}(X) = \begin{bmatrix} M & 0 & \cdots & 0 \\ [v_1]_\times & M & \cdots & 0 \\ \vdots & 0 & \ddots & 0 \\ [v_p]_\times & 0 & \cdots & M \end{bmatrix}. \tag{A.17}$$

Let $X \in SE_p(3)$ be a matrix such that:

$$X = \begin{bmatrix} C & v_1 \cdots v_p \\ 0_{p,d} & 0_p \end{bmatrix} \tag{A.18}$$

The group adjoint writes:

$$\mathrm{Ad}_{SE_p(3)}(X) = \begin{bmatrix} C & 0 & \cdots & 0 \\ [v_1]_\times C & C & \cdots & 0 \\ \vdots & 0 & \ddots & 0 \\ [v_p]_\times C & 0 & \cdots & C \end{bmatrix}. \tag{A.19}$$

B - Matrix Calculus

This Appendix describes matrices identities.

B.1 . Basic Properties

Let U be a square matrix. A first property is the inverse of the transposed matrix:

$$(M^{-1})^T = (M^T)^{-1}. \tag{B.1}$$

Assume that U has the following block structure:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

If D is invertible, U can be factorized such that:

$$M = \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} I & 0 \\ D^{-1}C & I \end{bmatrix}, \tag{B.2}$$

where I and 0 represent identity and zero blocks of appropriate size. Similarly, if A is invertible, M can be factorized such that:

$$M = \begin{bmatrix} I & 0 \\ CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & D - CA^{-1}B \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix}. \tag{B.3}$$

B.2 . Inverse

Let U be an invertible marix such that:

$$U = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

The inverse of U is obtained from (X, Y, Z, U) by solving the following system:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X & Y \\ Z & U \end{bmatrix} = I, \tag{B.4}$$

and it writes:

$$U^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}. \tag{B.5}$$

Similarly, the inverse of U can be obtained by solving:

$$\begin{bmatrix} X & Y \\ Z & U \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = I, \tag{B.6}$$

which leads to:

$$U^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}. \tag{B.7}$$

Since the inverse matrices from (B.5) and (B.7) are equal, the blocks of U verify the following equalities:

$$A^{-1}B(D - CA^{-1}B)^{-1} = (A - BD^{-1}C)^{-1}BD^{-1}, \tag{B.8}$$

$$D^{-1}C(A - BD^{-1}C)^{-1} = (D - CA^{-1}B)^{-1}CA^{-1}. \tag{B.9}$$

Hence, from (B.8) and (B.9), the inverse of U also writes:

$$U^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}, \tag{B.10}$$

and:

$$U^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}. \tag{B.11}$$

Another useful equality is the Woodburry identity. Let (X, Y, Z, U) be four matrices where X is invertible:

$$(X + YZU)^{-1} = X^{-1} - X^{-1}Y\left(Z^{-1} + UX^{-1}Y\right)^{-1}ZX^{-1}. \tag{B.12}$$

Also, the inversion equalities are used in this thesis:

$$\begin{bmatrix} I & A \\ 0 & I \end{bmatrix}^{-1} = \begin{bmatrix} I & -A \\ 0 & I \end{bmatrix}, \tag{B.13}$$

$$\begin{bmatrix} I & 0 \\ B & I \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -B & I \end{bmatrix}. \tag{B.14}$$

B.3 . Determinant

Basic properties of determinants which will be useful in this paper are first reminded:

- The determinant of the identity is 1:

$$\det I_d = 1. \tag{B.15}$$

- The determinant of a triangle or block triangle matrix is the product of the determinant of the diagonal terms:

$$\det \begin{bmatrix} A & B \\ 0 & D \end{bmatrix} = \det \begin{bmatrix} A & 0 \\ C & D \end{bmatrix} = \det(A)\det(D). \tag{B.16}$$

- The determinant of the inverse is the inverse of the determinant:

$$\det A^{-1} = \frac{1}{\det A}. \tag{B.17}$$

Considering the factorization (B.2):

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} = \begin{bmatrix} A - BD^{-1}C & B \\ 0 & D \end{bmatrix}, \tag{B.18}$$

the determinant of U writes:

$$\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(A - BD^{-1}C) \det(D). \tag{B.19}$$

C - Cartesian Product of Lie Groups

This section introduces the main definitions and properties of Cartesian products of matrix Lie groups. Let $X \in \mathcal{G}$ and $Y \in \mathcal{H}$ be two elements of matrix Lie groups $\mathcal{G}$ and $\mathcal{H}$ of dimensions d and m.
The Cartesian product of $\mathcal{G}$ and $\mathcal{H}$ is defined by:

$$\mathcal{G} \times \mathcal{H} \triangleq \left\{ Z = \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix}, \forall X \in \mathcal{G}, Y \in \mathcal{H} \right\}. \tag{C.1}$$

Define $\mathcal{F} = \mathcal{G} \times \mathcal{H}$, then the space $\mathcal{F}$ verifies the following properties:

1. $\mathcal{F}$ is a matrix Lie group.
2. If $\mathcal{G}$ and $\mathcal{H}$ are unimodular Lie groups, then $\mathcal{F}$ is a unimodular Lie group.
3. The Lie algebra of $\mathcal{F}$ is defined by:
$$\mathfrak{f} = \left\{ z = \begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix} \forall x \in \mathfrak{g}, y \in \mathfrak{h} \right\}. \tag{C.2}$$
4. $\mathcal{F}$ exponential verifies:
$$\exp_{\mathcal{F}}^{\wedge}(\cdot) = \begin{bmatrix} \exp_{\mathcal{G}}^{\wedge}(\cdot) & 0 \\ 0 & \exp_{\mathcal{H}}^{\wedge}(\cdot) \end{bmatrix}. \tag{C.3}$$
5. $\mathcal{F}$ logarithm verifies:
$$\log_{\mathcal{F}}^{\vee}(\cdot) = \begin{bmatrix} \log_{\mathcal{G}}^{\vee}(\cdot) & 0 \\ 0 & \log_{\mathcal{H}}^{\vee}(\cdot) \end{bmatrix}. \tag{C.4}$$
6. The product of $n \in \mathbb{N}$ Lie groups $\mathcal{G}_1 \times \cdots \times \mathcal{G}_n$ verifies all the previous properties.

Proof:

1. By definition, Z is an invertible square matrix which copes with the group axioms. The group action is still the matrix product which is differentiable, and $\mathcal{F}$ is a differential manifold. Hence, $\mathcal{F}$ is a matrix Lie group.
2. Let $Z_1, Z_2 \in \mathcal{F}$ be two Lie group matrices. Since $\mathcal{F}$ is a matrix Lie group, its group adjoint writes:
$$\begin{aligned} \mathsf{Ad}_{\mathcal{F}}(Z_1)Z_2 &= Z_1 Z_2 Z_1^{-1} \\ &= \begin{bmatrix} X_1 & 0 \\ 0 & Y_1 \end{bmatrix} \begin{bmatrix} X_2 & 0 \\ 0 & Y_2 \end{bmatrix} \begin{bmatrix} X_1^{-1} & 0 \\ 0 & Y_1^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathsf{Ad}_{\mathcal{G}}(X_1)X_2 & 0 \\ 0 & \mathsf{Ad}_{\mathcal{H}}(Y_1)Y_2 \end{bmatrix}. \end{aligned} \tag{C.5}$$
Thus $\det\left[\mathsf{Ad}_{\mathcal{F}}(Z)\right] = \det\left[\mathsf{Ad}_{\mathcal{G}}(X)\right] \det\left[\mathsf{Ad}_{\mathcal{H}}(Y)\right]$.
If $\mathcal{G}$ and $\mathcal{H}$ are unimodular, $\det\left[\mathsf{Ad}_{\mathcal{F}}(Z)\right] = 1$ and $\mathcal{F}$ is an unimodular Lie group.

3. The algebra of a direct product of Lie groups is the direct product of their algebras.

4. Let $c = [a^T b^T]^T \in \mathbb{R}^{d+m}$ be a vector, where $a \in \mathbb{R}^d$, $b \in \mathbb{R}^m$. The definition of the group exponential gives:

$$\begin{aligned} \exp_{\mathcal{F}}^{\wedge}(c) &= \sum_{n=0}^{\infty} \frac{([c]^{\wedge})^n}{n!} \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \begin{bmatrix} [a]^{\wedge} & 0 \\ 0 & [b]^{\wedge} \end{bmatrix}^n \\ &= \begin{bmatrix} \exp_{\mathcal{G}}^{\wedge}(a) & 0 \\ 0 & \exp_{\mathcal{H}}^{\wedge}(b) \end{bmatrix} \end{aligned} \tag{C.6}$$

5 The group logarithm proof is similar to 4.

6 Let $\mathcal{G}_1, \cdots, \mathcal{G}_n$ n be a set of matrix Lie groups. The proofs of 1 to 5 prove 6 for $\mathcal{G}_1 \times \mathcal{G}_2$, then consider the induction $(\mathcal{G}_1 \times \cdots \times \mathcal{G}_{n-1}) \times \mathcal{G}_n$.

A specific case used in this thesis is n Cartesian products of the same group $\mathcal{G}$, denoted $\mathcal{G}^n$, and which elements write:

$$\operatorname{diag}(X_{0:k}) \triangleq \begin{bmatrix} X_0 & 0 & \cdots & 0 \\ 0 & X_1 & 0 & \cdots \\ 0 & \cdots & \ddots & \cdots \\ 0 & \cdots & 0 & X_k \end{bmatrix} \in \mathcal{G}^{k+1}, \tag{C.7}$$

where $\mathcal{G}^{k+1}$ is a unimodular Lie group of dimension $(k+1)d$. □

D - Résumé en Français - French Summary

Contexte

L'estimation Bayésienne regroupe un ensemble de méthodes statistiques qui permettent de traiter des données ou des évènements comportant des incertitudes. Ce domaine se base sur le théorème de Bayes, qui permet de fusionner les données issues d'observations avec une connaissance *a priori* sur l'évènement. L'avantage de cette approche est de pouvoir prendre en compte les incertitudes de modélisation d'un système, ce qui amène à des résultats plus cohérents et plus précis. Cependant, cette approche théorique peut être difficile à mettre en œuvre lorsque le système traité est fortement non linéaire ou de grande dimension. En effet, seul les systèmes linéaires et Gaussiens amènent à une solution exacte : le filtre de Kalman [45].
Ainsi, de nombreuses méthodes numériques ont été développées pour résoudre le théorème de Bayes dans un cadre non linéaire, qui se présente fréquemment dans la pratique. Parmi ces méthodes, les filtres d'estimation tels que le filtre de Kalman étendu (EKF), représentent des solutions efficaces et matures. En outre, les filtres particulaires (PF) permettent de traiter des cas fortement non linéaires et non-Gaussiens en représentant la densité de probabilité estimée par un nuage de particules [54, 70, 56]. Cependant, les filtres d'estimation classiques présentent des limitations dans leur tolérance aux non linéarités du système et aux incertitudes.
Des études récentes sur des filtres implémentés dans les groupes de Lie[1] ont démontré l'intérêt d'utiliser ces espaces pour représenter les variables d'estimation. En effet, les groupes de Lie possèdent la double nature de groupe et de variété différentielle, ce qui représente un cadre d'intérêt lorsque des grandeurs difficiles à traiter (telles que les rotations) entrent dans la conception du filtre [14, 4, 21]. Ainsi, cette thèse se place dans la lignée des travaux qui ont été réalisés ces dernières années, en étendant le formalisme des groupes de Lie aux filtrage particulaire.
Les travaux de cette thèse s'appliquent à la navigation d'un système mobile, qui englobe les techniques permettant l'estimation de sa position, de sa vitesse et de son orientation. Cette discipline utilise un ensemble de capteurs embarqués dont les mesures sont imparfaites, ainsi que des modèles cinématiques comportant des incertitudes, ce qui en fait une application classique du problème d'estimation Bayésienne. Ces dernières années, les mesures à traiter sont devenues de plus en plus complexes à cause de nouvelles problématiques industrielles et opérationnelles. L'utilisation combinée du filtrage particulaire et de la théorie des groupes de Lie développée dans cette thèse s'est donc révélé être une voie prometteuse pour adresser ces problématiques nouvelles.

[1] Les groupes de Lie sont des espaces non-Euclidiens (courbes) ayant des propriétés mathématiques particulières. Leur principal intérêt est de permettre une meilleure représentation des variables d'un système non-linéaires et d'exploiter les symétries d'un modèle.

Objectifs Scientifiques

L'objectif de la thèse est d'établir les bases du filtrage particulaire sur groupes de Lie et d'en étudier l'intérêt pour des problèmes de navigation optimale. Cela implique l'adaptation des principaux aspects du filtre particulaire à ce nouveau formalisme. Une attention particulière est portée à la faisabilité des méthodes proposées, en terme d'implémentation, de performances et de coût de calcul.

Méthodes et développements

Un filtre de navigation a pour principe d'estimer les variables cinématiques d'un système en se basant sur un modèle de mouvement et un modèle de mesure associé à un capteur. Ces modèles sont décrits par des processus aléatoires qui sont traités par une approche Bayésienne [2] adaptée aux groupes de Lie. Cette nouvelle approche développée dans la thèse a mené à l'algorithme de base du filtre particulaire sur groupes de Lie, nommé LG-PF pour Lie Group Particle Filter. Par ailleurs, une attention particulière a été portée à l'étape de ré-échantillonnage qui intervient lorsque l'ensemble des particules doit être régénéré. En se basant sur la théorie du filtre particulaire Laplacien (LPF), une approche innovante a été proposée pour générer des particules dans les zones où la densité a posteriori est la plus probable. L'approche se formalise comme un problème d'optimisation qui se résout en utilisant un filtre de Kalman itéré sur groupes de Lie [14]. Ce nouveau filtre, nommé LG-LPF (Lie Group Laplace Particle Filter) a fait l'objet d'une première publication [26] et a démontré un gain significatif en performances par rapport aux algorithmes classiques sur un scénario de recalage par angles d'arrivée. Ces résultats on été confirmés dans une seconde publication portant sur des mesures Doppler [23]. Par ailleurs, cette approche a été généralisée dans des cas où les densités de probabilités présentent de fortes multi modalités. La méthode proposée établit une recherche des zones les plus probables sur un critère statistique, et étend des méthodes de partitionnement (clustering) aux groupes de Lie, ce qui constitue également une nouveauté introduite par la thèse. L'application de ces travaux à un scénario de navigation optimale par mesures Doppler ambigües a donné lieu à une seconde publication [25], et a démontré que l'utilisation des groupes de Lie améliore la précision et la robustesse du filtre particulaire, même pour un nombre faible de particules. Malgré les résultats encourageants du LG-LPF, ce filtre est toujours sujet au fléau de la dimension [3]. Pour cela, les travaux de thèse se sont tournés vers le LG-KPKF (Lie Groups Kalman-Particle Kernel Filter), qui a pour principe de représenter la densité de probabilités de l'état par une mixture de Gaussiennes sur groupes de Lie. L'application du formalisme Bayésien à cette nouvelle formulation montre que chaque particule peut être traitée par un filtre de Kalman étendu local, et l'ensemble des poids est régi par un filtre particulaire. Ainsi, les filtres locaux guident les particules vers les zones les plus probables et l'approche particulaire per-

[2] L'inférence Bayésienne est une méthode permettant de calculer la probabilité de divers évènements étant donné des observations.

[3] Le fléau de la dimension est un phénomène qui se présente lorsqu'un grand nombre de variables (supérieur à six) sont estimées par le filtre particulaire.

met de traiter des problèmes non-Gaussiens. De plus, le LG-KPKF a été étudié dans le cas où le modèle cinématique vérifie des propriétés de symétrie définies dans la théorie du filtre de Kalman étendu invariant (IEKF), améliorant significativement les performances du filtre. Cette nouvelle approche a fait l'objet d'une publication [24] et a démontré l'intérêt du LG-KPKF par rapport à des algorithmes de premier plan de la littérature (LG-EKF [14] et LG-LPF).
Un autre aspect des travaux de thèse s'est porté sur l'étude dune borne statistique sur groupes de Lie. Ces bornes calculent l'erreur asymptotique minimale atteignable pour un problème d'estimation (quelle que soit la méthode employée pour le résoudre), ce qui représente un outil puissant pour évaluer les performances d'un algorithme et concevoir un système robuste. Des études préliminaires ont été réalisées à propos de la borne de Cramer-Rao sur groupes de Lie [48, 11]. Cependant, les formulations proposées dans la littérature n'incluaient pas les aspects temporels des simulations. Ainsi, une borne de Cramer-Rao récursive sur groupes de Lie matriciels a été introduite dans cette thèse. Plus spécifiquement, les travaux de thèse ont montré que la formulation proposée est une extension naturelle de la borne proposée par Tichavsky [78]. Ce nouvel élément théorique a fait l'objet d'un article soumis dans la revue Automatica.
Les travaux de thèse sur le filtre particulaire ont permis des développements sur des méthodes connexes. En particulier, le filtre de Kalman étendu sur groupes de Lie (LG-EKF) est un algorithme de référence dans la littérature et surclasse le filtre de Kalman étendu classique. Cependant, les approches proposées dans la littérature présentent des formulations problématiques pour l'estimation des biais des capteurs inertiels. Une formulation adaptée a été proposée dans la thèse, apportant un gain dans l'estimation des biais gyrométriques et qui se traduit par une amélioration globale de la précision du filtre. De plus, un calcul analytique des matrices Jacobiennes sur groupes de Lie a été établi, ce qui a permis d'obtenir une implémentation plus simple et plus précise du filtre tout en diminuant la charge de calcul. Ces travaux ont fait l'objet d'un article en finalisation pour la revue IEEE Transactions on Robotics.

Conclusion et Perspectives

Cette thèse a permis d'établir les bases théoriques du filtre particulaire sur groupes de Lie, qui représente un champ de techniques d'estimation nouvelles. Par ailleurs, un ensemble de filtres innovants a été implémenté sur des problèmes de navigation exigeants. Les résultats obtenus ont justifié l'intérêt de l'utilisation des groupes de Lie, menant à une amélioration nette de la robustesse et de la précision des algorithmes pour un nombre réduit de particules (un à deux ordres de grandeur inférieur). De plus, la mise en place de la borne de Cramer-Rao récursive dans les groupes de Lie constitue une avancée significative pour caractériser le comportement des filtres développés.

Bibliography

[1] G. B. Arfken. Mathematical methods for physicists. *American Journal of Physics*, 35, 1966.

[2] F. Bao, Y. Cao, C. Webster, and G. Zhang. A hybrid sparse-grid approach for nonlinear filtering problems based on adaptive-domain of the zakai equation approximations. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):784–804, 2014.

[3] T. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[4] A. Barrau. *Non-linear State Error Based Extended Kalman Filters with Applications to Navigation*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2015.

[5] A. Barrau and S. Bonnabel. Invariant particle filtering with application to localization. *53rd IEEE Conference on Decision and Control*, pages 5599–5605, 2014.

[6] A. Barrau and S. Bonnabel. The invariant extended Kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, 62:1797–1812, 2017.

[7] A. Barrau and S. Bonnabel. Linear observed systems on groups. *Syst. Control. Lett.*, 129:36–42, 2019.

[8] B. M. Bell and F. W. Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Trans. Autom. Control.*, 38:294–297, 1993.

[9] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Y. Siegwart. Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36:1053 – 1072, 2017.

[10] S. Bonnabel. Symmetries in observer design: Review of some recent results and applications to ekf-based slam. In Krzysztof Kozłowski, editor, *Robot Motion and Control 2011*, pages 3–15, London, 2012. Springer London.

[11] S. Bonnabel and A. Barrau. An intrinsic cramér-rao bound on lie groups. In *GSI*, 2015.

[12] S. Borman. The expectation maximization algorithm-a short tutorial. *Submitted for publication*, 41, 2004.

[13] N. Boumal. An introduction to optimization on smooth manifolds. To appear with Cambridge University Press, Jun 2022.

[14] G. Bourmaud. *Estimation de pramètres évoluant sur des groupes de Lie : application à la cartographie et à la loalisation d'une caméra monoculaire*. PhD thesis, Université de Bordeaux, 2015.

[15] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu. Discrete Extended Kalman Filter on Lie groups. *21st European Signal Processing Conference (EUSIPCO 2013)*, pages 1–5, 2013.

[16] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu. From intrinsic optimization to iterated extended Kalman filtering on Lie groups. *Journal of Mathematical Imaging and Vision*, 55:284–303, 2015.

[17] M. Brossard. *Deep learning, inertial measurements units, and odometry : some modern prototyping techniques for navigation based on multi-sensor fusion*. Phd thesis, Université Paris sciences et lettres, September 2020.

[18] M. Brossard, S. Bonnabel, and J.-P. Condomines. Unscented Kalman filtering on Lie groups. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2485–2491, 2017.

[19] Z. Cai, F. Le Gland, and H. Zhang. *An adaptive local grid refinement method for nonlinear filtering*. PhD thesis, INRIA, 1995.

[20] T. Cantelobre, C. Chahbazian, A. Croux, and S. Bonnabel. A real-time unscented kalman filter on manifolds for challenging auv navigation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2309–2316, 2020.

[21] D. Caruso. *Improving Visual-Inertial Navigation Using Stationary Environmental Magnetic Disturbances*. PhD thesis, Université Paris-Saclay, 2018.

[22] H. Carvalho, P. Del Moral, A. Monin, and G. Salut. Optimal nonlinear filtering in gps/ins integration. *IEEE Transactions on Aerospace and Electronic Systems*, 33:835–850, 1997.

[23] C. Chahbazian, K. Dahia, N. Merlinge, B. Winter-Bonnet, A. Blanc, and C. Musso. Filtre particulaire sur groupes de lie. In *GRETSI 2022*, 2022.

[24] C. Chahbazian, K. Dahia, N. Merlinge, B. Winter-Bonnet, K. Honore, and C. Musso. Improved Kalman-particle kernel filter on lie groups applied to angles-only uav navigation. *IEEE International Conference on Robotics and Automation*, pages 1689–1694, 2022.

[25] C. Chahbazian, N. Merlinge, K. Dahia, B. Winter-Bonnet, A. Blanc, and C. Musso. Generalized Laplace Particle Filter on Lie Groups Applied to Ambiguous Doppler Navigation. *IEEE International Conference on Intelligent Robots (IROS)*, 2022.

[26] C. Chahbazian, N. Merlinge, K. Dahia, B. Winter-Bonnet, J. Marini, and C. Musso. The Laplace particle filter on Lie groups applied to angles-only navigation. *IEEE International Conference on Information Fusion*, 2021.

[27] G. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2*. Birkhäuser Boston, MA, 2012.

[28] G. Chirikjian and M. Kobilarov. Gaussian approximation of non-linear measurement models on Lie groups. *53rd IEEE Conference on Decision and Control*, pages 6401–6406, 2014.

[29] H. Cramér. *Mathematical methods of statistics*. Princeton University Press, 1999.

[30] K. Dahia. *Nouvelles méthodes en filtrage particulaire-Application au recalage de navigation inertielle par mesures altimétriques*. PhD thesis, Université Joseph Fourier, 2005.

[31] K. Dahia, C. Musso, D. Pham, and J. Guibert. Application of the kalman-particle kernel filter to the updated inertial navigation system. *2004 12th European Signal Processing Conference*, pages 601–604, 2004.

[32] P. Del Moral and A. Guionnet. Central limit theorem for nonlinear filtering and interacting particle systems. *Annals of Applied Probability*, pages 275–297, 1999.

[33] J. Deyst and C. Price. Conditions for asymptotic stability of the discrete minimum-variance linear estimator. *IEEE Transactions on Automatic Control*, 13(6):702–705, 1968.

[34] P. Dodin, P.; Minvielle, and J-P. Le Cadre. Re-entry vehicle tracking observability and theoretical bound. In *2005 7th International Conference on Information Fusion*, volume 1, pages 8–pp. IEEE, 2005.

[35] A. Doucet, Simon J. Godsill, and Christophe Andrieu. On sequential simulation-based methods for bayesian filtering. *Statistics and Computing*, 1998.

[36] A. Doucet and N. De Freitasand N. Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

[37] G. Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

[38] A. Fornasier, Ng. Yonhon, R. Mahony, and S. Weiss. Equivariant filter design for inertial navigation systems with input measurement biases. *arXiv preprint arXiv:2202.02058*, 2022.

[39] N. Gordon, D. Salmond, F. Adrian, and M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. 1993.

[40] P. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*. Artech, 2013.

[41] J. P. Helferty and D. R. Mudgett. Optimal observer trajectories for bearings only tracking by minimizing the trace of the cramer-rao lower bound. *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 936–939 vol.1, 1993.

[42] J. Hilgert and K.-H. Neeb. *Structure and Geometry of Lie Groups*. Springer New York, NY, 2011.

[43] Y. Huang, Y. Zhang, Z. Wu, N. Li, and J. A. Chambers. A novel adaptive kalman filter with inaccurate process and measurement noise covariance matrices. *IEEE Transactions on Automatic Control*, 63:594–601, 2018.

[44] S. Julier and J. Uhlmann. Comment on" a new method for the nonlinear transformation of means and covariances in filters and estimators"[with authors' reply]. *IEEE Transactions on Automatic Control*, 47(8):1408–1409, 2002.

[45] R. E. Kalman. A new approach to linear filtering and prediction problems" transaction of the asme journal of basic. 1960.

[46] S. Klarsfeld and J. A. Oteo. The baker-campbell-hausdorff formula and the convergence of the magnus expansion. *Journal of Physics A*, 22:4565–4572, 1989.

[47] A. Kong, J. S. Liu, and W. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89:278–288, 1994.

[48] S. Labsir. *Méthodes statistiques fondées sur les groupes de Lie pour le suivi d'un amas de débris spatiaux.* Phd thesis, Université de Bordeaux, December 2020.

[49] F. Le Gland. cours sod333 filtrage bayésien et approximation particulaire.

[50] Y. Lin, L. Miao, and Z. Zhou. An improved mcmc-based particle filter for gps-aided sins in-motion initial alignment. *IEEE Transactions on Instrumentation and Measurement*, 69(10):7895–7905, 2020.

[51] G. M. Magalhães, Y. Caceres, J. Bosco Ribeiro do Val, and R. Santos Mendes. Ukf on lie groups for radar tracking using polar and doppler measurements. In *Congresso Brasileiro de Automática-CBA*, volume 1, 2018.

[52] G. J. McLachlan, S. X. Lee, and S. I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

[53] G. J. McLachlan and D. Peel. Finite mixture models. In *Wiley Series in Probability and Statistics*, 2000.

[54] N. Merlinge. *State estimation and trajectory planning using particle box particle kernels*. PhD thesis, Université Paris-Saclay ; Coventry University, 2018.

[55] N. Merlinge, K. Dahia, H. Piet-Lahanier, J. Brusey, and N. Horri. A box regularized particle filter for state estimation with severely ambiguous and non-linear measurements. *Autom.*, 104:102–110, 2019.

[56] A. Murangira. *Nouvelles approches en filtrage particulaire. Application au recalage de la navigation inertielle*. Phd thesis, Université de Technologie de Troyes - UTT, March 2014.

[57] A. Murangira, C. Musso, and K. Dahia. A mixture regularized rao-blackwellized particle filter for terrain positioning. *IEEE Transactions on Aerospace and Electronic Systems*, 52:1967–1985, 2016.

[58] C. Musso. *Contributions aux méthodes numériques pour le filtrage et la recherche opérationnelle*. Hdr thesis, Université de Toulon, 2017.

[59] C. Musso, F. Champagnat, and O. Rabaste. Improvement of the laplace-based particle filter for track-before-detect. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1095–1102. IEEE, 2016.

[60] C. Musso, N. Oudjane, and F. L. Gland. Improving regularised particle filters. In *Sequential Monte Carlo Methods in Practice*, 2001.

[61] C. Musso, P. Bui Quang, and F. Gland. Introducing the Laplace approximation in particle filtering. *14th International Conference on Information Fusion*, pages 1–8, 2011.

[62] C. Musso, P. Bui Quang, and A. Murangira. A laplace-based particle filter for track-before-detect. *2015 18th International Conference on Information Fusion (Fusion)*, pages 1657–1663, 2015.

[63] N. Oudjane. *Stabilité et approximations particulaires en filtrage non linéaire application au pistage*. Phd thesis, Université de Rennes, 2000.

[64] C. Palmier. *New particle filters for underwater terrain-aided navigation using multi-sensor fusion*. Phd thesis, Université de Bordeaux, December 2021.

[65] F. C. Park. *The optimal kinematic design of mechanisms*. PhD thesis, Division of Engineering and Applied Science, Harvard University, Cambridge, MA, 1991.

[66] X. Pennec and V. Arsigny. Exponential Barycenters of the Canonical Cartan Connection and Invariant Means on Lie Groups. In F. Barbaresco, A. Mishra, and F. Nielsen, editors, *Matrix Information Geometry*, pages 123–168. Springer, May 2012.

[67] D. Pham, K. Dahia, and C. Musso. A Kalman-particle kernel filter and its application to terrain navigation. *Sixth International Conference of Information Fusion, 2003. Proceedings of the*, 2:1172–1179, 2003.

[68] D. T. Pham. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly weather review*, 129(5):1194–1207, 2001.

[69] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[70] P. B. Quang. *Aproximation particulaire et méthode de Laplace pour le filtrage Bayésien*. PhD thesis, Université de Rennes, 2013.

[71] P. B. Quang, C. Musso, and F. Le Gland. An insight into the issue of dimensionality in particle filtering. In *2010 13th International Conference on Information Fusion*, pages 1–8. IEEE, 2010.

[72] P. Bui Quang, C. Musso, and F. Gland. Particle filtering and the Laplace method for target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 52:350–366, 2016.

[73] C. R. Rao. Information and the accuracy attainable in the estimation of statistical parameters. In Samuel Kotz and Norman L. Johnson, editors, *Breakthroughs in Statistics: Foundations and Basic Theory*, pages 235–247, New York, NY, 1992. Springer New York.

[74] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[75] S. Said, L. Bombrun, Y. Berthoumieu, and J. H. Manton. Riemannian gaussian distributions on the space of symmetric positive definite matrices. *IEEE Transactions on Information Theory*, 63:2153–2170, 2015.

[76] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[77] H. Snoussi and A. Mohammad-Djafari. Particle Filtering on Riemannian Manifolds. *AIP Conference Proceedings (Vol. 872, No. 1, pp. 219-226)*, 2006.

[78] P. Tichavský, C. Muravchik, and A. Nehorai. Posterior Cramer-Rao bounds for discrete-time nonlinear filtering. *IEEE Trans. Signal Process.*, 46:1386–1396, 1998.

[79] L. Tierney, R. E. Kass, and J. B. Kadane. Fully exponential laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association*, 84:710–716, 1989.

[80] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.

[81] K. C. Wolfe and M. Mashner. Bayesian fusion on lie groups. *Journal of Algebraic Statistics*, 2(1), 2011.