HAL
open science

Smoothing algorithms for navigation, localisation and mapping based on high-grade inertial sensors

Paul Chauchat

To cite this version:

Paul Chauchat. Smoothing algorithms for navigation, localisation and mapping based on high-grade inertial sensors. Robotics [cs.RO]. Université Paris sciences et lettres, 2020. English. NNT : 2020UPSLM005 . tel-02887295

HAL Id: tel-02887295

https://pastel.hal.science/tel-02887295v1

Submitted on 2 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES ParisTech

Smoothing algorithms for navigation, localisation and mapping based on high-grade inertial sensors

Algorithmes de lissage pour la navigation, la localisation et la cartographie, basés sur des capteurs inertiels haute qualité

Soutenue par

Paul CHAUCHAT

Le 26 février 2020

École doctorale n°621

Ingénierie des Systèmes, Matériaux, Mécanique, Énergétique

Spécialité

Informatique temps réel, robotique, automatique

Composition du jury :

Brigitte d'ANDREA NOVEL Professeur, IRCAM	*Présidente du jury*
Luc JAULIN Professeur, ENSTA Bretagne	*Rapporteur*
James R. FORBES Associate Professor, McGill University	*Rapporteur*
Manon KOK Assistant Professor, TU Delft	*Examinateur*
Axel BARRAU Ingénieur de Recherche, Safran	*Examinateur*
Silvère BONNABEL Professeur, Université de Nouvelle-Calédonie	*Directeur de thèse*

MINES ParisTech | PSL

Acknowledgements/Remerciements

What a journey. It would not have been the same, or existed at all, without numerous people whom I want to acknowledge here.

First and foremost, I am grateful to both of my reviewers, James Forbes and Luc Jaulin, for accepting to read and evaluate the work carried out during these three years. They provided insightful remarks and suggestions to make this manuscript clearer and easier to understand. I also want to thank Brigitte d'Andréa-Novel and Manon Kok for presiding and being part of my jury respectively.

Since all the remaining persons I'd like to acknowledge are french speakers, I will switch to my native language.

Une thèse ce n'est pas seulement un thésard et un sujet, c'est aussi un encadrement. Je souhaite à toute personne se lançant dans l'aventure d'en avoir un d'aussi bonne qualité que moi. Silvère, Axel, un grand merci. De m'avoir pris en thèse, déjà, alors que j'arrivais totalement ignorant des sujets de navigation et d'estimation d'état. Merci d'avoir été présents lorsqu'il le fallait, malgré la distance, pour passer les murs qui se présentaient, tout en me laissant une grande liberté d'action. Je n'ai pas été l'étudiant le plus facile à encadrer, mais vous vous êtes toujours montrés encourageants, optimistes, et réalistes à la fois, et c'est malheureusement trop rare dans le monde académique pour ne pas le souligner.

Cet encadrement a certes facilité les choses, il n'empêche qu'une thèse reste une épreuve, principalement solitaire, avec des hauts et un certain nombre de bas, qu'on surmonte difficilement seul. Heureusement, j'ai pu compter sur bon nombre de soutiens pour sublimer les victoires et écouter patiemment les coups de gueule. En première ligne, mes compagnons d'armes, ces apprentis chercheurs qui pensaient s'être embarqués sur une croisière et se retrouvent sur un Vendée Globe, experts avant l'heure du confinement dès qu'ils prononcent les mots "article" et "deadline". Mais qui, miraculeusement, trouvent le temps pour aller prendre un café, ou une bière, lorsqu'ils sentent que l'un des leurs flanche. Et chez qui l'encouragement d'un seul passe avant sur les lamentations de beaucoup. Oui, merci infiniment chers doctorants et collègues de labo. Ceux du quotidien, personnels CAOR ou Terra parqués dans l'aquarium et le bocal du labo d'élec, Martin pour les longues discussions sur nos sujets respectifs; Hassan, Xavier, Yann, Antoine, Sofiane, Sami, Marin, Emmanuel et Martin, un de plus, pour les baffes de King of Tokyo et les virées au Pantalon. Et évidemment Hugues, Scapin du ping-pong, lobbeur de génie, incarnation de la diablerie (quand il n'a pas volley), grand ami des sapins et de la fraîche, et fantastique trésorier de Dopamines. Merci aussi aux autres, un peu plus loin mais qu'on croisait plus ou moins souvent, Michelle et Daniele, inséparable duo au subtil équilibre entre fantasque philosophie du karma et pragmatisme; et les gens de la V026.

Mais j'ai passé la moitié de mon temps autre part, dans les vertes et lointaines contrées de Châteaufort et Magny-les-Hameaux, et j'aurais bien plus subi ces périples s'il n'y avait pas eu la plus fine des équipes de doctorants sur place. Merci Mina, Elisa, Michel, Edouard, Yann, et par extension Sébastien, pour votre folie, votre honnêteté, votre intégrité. J'aurai rarement eu autant de discussions aussi passionnées, passionnantes, mais qui finissent (presque) toujours par un bon fou rire.

Le monde des doctorants ne se résume heureusement pas à son labo, et je salue tous ceux que j'ai pu rencontrer durant les aléas de la thèse. Se retrouver à gérer Dopamines, c'était peut-être forcer ces aléas me direz-vous. Que ce soit l'expertise vin chaud et urgences acquise lors des semaines ski, la découverte de la grande couronne à Champagne-sur-Seine et Fontainebleau, les nouvelles têtes aux Afterworks ou les présentations du Pint of Science, j'ai passé de sacrés bons moments avec vous. Merci aussi aux aventuriers de MT180, qui doit être l'un des rares concours où la solidarité est tellement forte qu'on ne se souvient qu'il y a un enjeu que lorsqu'ils annoncent les gagnants. Et merci aux doctorants de l'UNC pour l'accueil chaleureux qu'ils m'ont réservé lors de mes trois semaines de passage à l'autre bout du monde.

Merci aussi aux autres collègues du quotidien, la plupart anciens doctorants qui nous rappelaient tous les jours que, oui, c'est possible de finir. Notamment ces deux équipes dont le nom change tellement souvent qu'il sera plus clair des les appeler team Dohy et team Sylvaine, qui ont égayé ces 36 mois passés sur un sujet de niche d'afterworks et soirées jeux.

J'ai une pensée particulière pour mes nombreux colocataires successifs, Martin, Ludo, Charles, Xavier, Lamisse, qui ont dû me supporter durant ces soirées et weekends de boulot, à tenter de faire fonctionner

des bouts de code, crier victoire trop tôt, rédiger en urgence pour louper des deadlines, ou simplement tenter de comprendre mon sujet. Puis organiser des barbec/raclettes/autres en tout genre dès que le temps le permettait. Pouvoir me reposer sur vous dans les pires moments, perdu dans cette tour des hauts d'Issy, aura été un soulagement bien plus important que ce que vous ne l'imaginez. J'inclus aussi, évidemment, toutes celles et ceux qui nous ont rejoint pour quelques semaines ou quelques mois, dans la vigie face à la terrasse, parfois tellement régulièrement qu'on hésite à leur faire une carte de fidélité. Ensemble, ça aura été plus facile de surmonter ces soirées devant films et séries, dont on taira le nom pour préserver l'honneur et la dignité des participants.

Il paraît qu'il y a eu une vie avant la thèse, et mieux encore, que certaines et certains ont été présents depuis tout ce temps. A vous, je dois des excuses, car vous avez entendu bien plus de fois que de raison "J'peux pas, j'ai thèse", signe d'une tendance exacerbée à me barricader derrière le boulot, ce qui fut une erreur. Incroyable mais vrai, vous êtes encore là. Vous, les centraliens et centraliennes, tant de gens avec qui on refaisait le monde, perdus sur les bords du parc de Sceaux, qu'ils viennent de CeC, du Bad, du C7, mais surtout du 2G, cette belle bande d'abrutis, moi compris, qui m'aura enseigné parmi les leçons les plus importantes sur l'amitié, le respect et la confiance. Et, en continuant à remonter le temps, ceux avec qui j'ai grandi, du collège à la prépa, en traversant bon gré mal gré diverses aventures capillaires et vestimentaires.

J'aimerais aussi chaleureusement remercier nombre de personnes qui ne liront certainement pas ces lignes, mais qu'on oublie trop souvent, alors qu'ils sont parmi les premiers à éveiller notre curiosité pour ce qui nous entoure et la science, oui, un immense merci à tous ces profs qui m'ont appris, appris à être curieux, appris à réfléchir.

Mes avant-derniers remerciements vont aux âmes curieuses qui sont parvenues jusqu'ici. Mes derniers, les plus importants, vont à ma famille, tribu protéiforme s'il en est, qui s'est enrichie au fil du temps, et qui m'a toujours apporté un soutien inconditionnel sans jamais tenter d'influencer mes choix, et m'a laissé construire mon parcours avec une confiance totale. Comment ne pas finir par mon père, toi qui qui m'as conduit depuis toutes ces années, patiemment, discrètement, mais solidement. Une force tranquille, toujours droit dans la tempête, qui m'a porté de l'enfance à l'âge adulte. Merci, papa.

Contents

1 Introduction **7**
1.1 Highlights of the thesis . . . 7
1.2 Points marquants de la thèse . . . 8
1.3 Industrial context . . . 9
1.3.1 Inertial Navigation Systems since the 1950's . . . 9
1.3.2 Safrantech's experimental autonomous car . . . 9
1.4 Sensor fusion: global vs. local . . . 10
1.4.1 Inertial-aided navigation with global sensors . . . 10
1.4.2 Inertial-aided navigation with local perception . . . 11
1.4.3 Keeping memory of the past in the presence of non-linearities . . . 12
1.4.4 The smoothing framework, as studied in this thesis . . . 13
1.5 Contributions and organisation of the thesis . . . 14
1.5.1 Contributions . . . 14
1.5.2 Organisation of the document . . . 16
1.5.3 Publications . . . 17

I From Invariant Filtering to Invariant Smoothing **18**

2 Autonomous errors on Lie groups **19**
2.1 Mathematical preliminaries . . . 20
2.1.1 Lie Groups . . . 20
2.1.2 Uncertainties on Lie Groups . . . 22
2.1.3 Uncertainties of group actions . . . 22
2.2 The Lie groups of navigation . . . 22
2.2.1 The group of 2D rotations $SO(2)$ . . . 23
2.2.2 The group of rotations $SO(3)$ . . . 23
2.2.3 The group of 2D direct spatial isometries $SE(2)$ . . . 23
2.2.4 The group of 3D direct spatial isometries $SE(3)$ . . . 24
2.2.5 The group of double direct spatial isometries $SE_2(3)$ . . . 24
2.3 Group-affine dynamics and group actions lead to autonomous errors . . . 25
2.4 Examples of exactly and imperfect group-affine dynamics in navigation . . . 27
2.4.1 Attitude estimation . . . 27
2.4.2 Pose estimation . . . 28
2.4.3 Unbiased inertial navigation . . . 28
2.4.4 Adding noise and the biases . . . 29
2.5 Conclusion . . . 30

3 Invariant Filtering and geometric constraints **31**
3.1 The Linear Kalman Filter and the Extended Kalman Filter . . . 32
3.1.1 The linear Kalman filter and its optimality properties . . . 32
3.1.2 The Extended Kalman Filter . . . 33
3.2 The Invariant Extended Kalman Filter . . . 33
3.2.1 Considered noisy systems . . . 34
3.2.2 Invariant filtering equations, stability and consistency properties . . . 34
3.3 Kalman filtering and geometric constraints . . . 35
3.3.1 The Linear Kalman Filter may preserve side information . . . 35
3.3.2 Non-linear case . . . 36
3.4 Invariant Kalman filtering with geometric constraints . . . 36
3.4.1 Geometric constraints . . . 36

3.4.2 An example: equivariant constraints . . . 37
3.4.3 Illustration in terms of attitude estimation . . . 37
3.4.4 The L-IEKF naturally respects the constraints . . . 37
3.4.5 Direct corollary for the R-IEKF . . . 38
3.4.6 Graphical illustration of the theorem and discussion . . . 39
3.5 Examples . . . 39
3.5.1 Car position and heading estimation . . . 40
3.5.2 Attitude estimation . . . 41
3.5.3 Unbiased inertial navigation on flat-earth . . . 41
3.6 Conclusion . . . 42

4 Invariant smoothing 44
4.1 An introduction to the smoothing framework . . . 45
4.1.1 From filtering to smoothing . . . 45
4.1.2 Smoothing as a non-linear least-squares problem . . . 46
4.1.3 Solving the optimisation with iterative methods . . . 47
4.1.4 Smoothing inside a sliding window . . . 47
4.1.5 Smoothing on manifold . . . 48
4.2 Left-invariant smoothing . . . 48
4.2.1 Considered systems . . . 48
4.2.2 Factors definition . . . 49
4.2.3 Final algorithm and benefits of the proposed approach . . . 50
4.2.4 L-IS and the iterated IEKF . . . 50
4.3 Application of L-IS to mobile robot localisation . . . 51
4.3.1 Considered problem: robot localisation . . . 51
4.3.2 Compared smoothing frameworks . . . 52
4.3.3 Simulation results . . . 52
4.3.4 Experimental setting . . . 52
4.3.5 Experimental results . . . 55
4.3.6 Discussion . . . 55
4.3.7 Invariant Smoothing for biased inertial navigation: Extension for imperfect group-affine dynamics . . . 57
4.4 IMU preintegration in the context of invariant smoothing . . . 57
4.4.1 Classical preintegration theory . . . 57
4.4.2 Preintegration as a group-affine property: summary of [17,18] . . . 58
4.4.3 Defining the preintegrated IMU factors . . . 59
4.4.4 Jacobians computation . . . 60
4.4.5 Summary of the differences between preintegrated IMU factors . . . 61
4.4.6 Simulation results in inertial navigation: L-IS vs IEKF . . . 61
4.5 Other Invariant Smoothings: Right-Invariant and Hybrid Invariant Smoothing . . . 61
4.5.1 Using the right-invariant parametrisation: Right-Invariant Smoothing . . . 61
4.5.2 Considered systems . . . 62
4.5.3 Linearised system of R-IS . . . 62
4.5.4 An alternative to L-IS: Hybrid Invariant Smoothing . . . 63
4.6 Conclusion . . . 63

5 Invariant smoothing in the face of geometric constraints and unobservability related inconsistency 67
5.1 Invariant Smoothing also respects geometric constraints . . . 68
5.1.1 A 2D localisation illustrative example . . . 68
5.1.2 Invariant Smoothing respects deterministic group affine dynamics . . . 70
5.1.3 Conclusion . . . 73
5.2 Unobservability from a smoothing point of view . . . 73
5.2.1 Standard observability theory: The observability matrix and false observability . . . 74
5.2.2 Unobservability seen through a smoothing update . . . 76
5.2.3 The impact of the prior covariance illustrated on a toy example . . . 77
5.2.4 Studying a reduced example for SLAM and inertia-aided pose-SLAM . . . 78

5.2.5 Conclusion . . . 79
5.3 Conclusion . . . 80

6 Real-world experiments **81**
6.1 Experimental setup . . . 82
6.1.1 Safrantech's autonomous vehicle prototype . . . 82
6.1.2 Experimental tracks . . . 82
6.2 Inertia-GPS fusion experiment . . . 84
6.2.1 Experiments using the actual IMU increments . . . 84
6.2.2 Stronger biases could degrade performances . . . 85
6.3 Fusing inertia and relative translations . . . 86
6.3.1 Filtering vs Smoothing . . . 86
6.3.2 Assessing the importance of the right-invariant parametrisation . . . 86
6.4 Conclusions of the experiments . . . 89

II Navigation with highly precise sensors **92**

7 The shortcomings of standard smoothing with highly precise sensors **93**
7.1 Standard factor graph based smoothing through MAP . . . 94
7.1.1 Main considered problem . . . 94
7.1.2 Resolution of the linearised optimisation problem . . . 95
7.2 The limits of factor graphs and MAP . . . 95
7.2.1 Potential issue n°1: computational complexity . . . 95
7.2.2 Potential issue n°2: ill-conditioned information matrix . . . 95
7.3 A solution without matrix inversion: the robust batch solver . . . 96

8 SC-BIFM : A new solver based on the Kalman smoother and stochastic cloning **98**
8.1 The linear Kalman smoother as a least-squares solver . . . 99
8.1.1 The standard Kalman smoother . . . 99
8.1.2 The Backward Information Forward Marginal (BIFM) variant of the Kalman smoother 100
8.1.3 Summary of the approach . . . 101
8.2 Proposed method: the BIFM Kalman smoother with stochastic cloning . . . 102
8.2.1 Stochastic cloning for filtering . . . 102
8.2.2 Proposed "SC-BIFM" (stochastic cloning for smoothing) . . . 104
8.2.3 Current limits of the proposed SC-BIFM . . . 106
8.3 Generalisation to all linear least-squares . . . 106
8.3.1 Dealing with spanning-trees of invertible factors . . . 107
8.3.2 Regrouping factors and variables to propagate . . . 107
8.4 Numerical illustration of the methodology . . . 108
8.4.1 Ideal case numerical experiment . . . 109
8.4.2 Numerical experiment with low noise . . . 109

9 Real-world experiments **111**
9.1 Experimental setup . . . 111
9.1.1 Considered models and parameters . . . 111
9.1.2 Implementation . . . 112
9.2 Experimental comparison of linear solvers illustrating the decay of standard methods . . . 112
9.3 Experimental comparison of corresponding localisation algorithms . . . 114
9.4 Conclusions regarding SC-BIFM . . . 114

10 Invariant SC-BIFM : a non-linear Kalman smoother for precise navigation **116**
10.1 Information filter on Lie groups: exactly transposing the IEKF . . . 117
10.2 Rauch-Tung-Striebel Kalman smoother on Lie groups . . . 118
10.3 Adapting the Information form of Kalman filtering on Lie groups for BIFM . . . 119
10.4 Numerical validation of non-linear BIFM . . . 121
10.5 Conclusion . . . 122

11 Conclusion of the thesis **123**

A Right-Invariant Smoothing factors **125**
A.1 Prior factor 125
A.2 Group-affine factors 125
A.3 Measurement factors 125

Chapter 1

Introduction

This thesis investigates localisation algorithms based on sensor fusion, namely filtering and especially smoothing, in the context of high-grade inertial sensors. The first part deals with the nonlinear consequences of the use of high-grade inertial sensors, and demonstrates how the non-linear structure of both filtering and smoothing algorithms may be improved by leveraging the invariant filtering framework. The second part deals with the problems incurred by the linear solvers that are used at each step of nonlinear smoothing algorithms as a result of having highly precise sensors. It introduces a novel least-squares linear solver that solves the issues.

1.1 Highlights of the thesis

- The use of inertial measurements units (IMU) to navigate is called inertial navigation. The fusion of IMU with other sensors or sources of information has been performed mainly through the use of the extended Kalman filter (EKF) over the last fifty years. In the industry, EKF is the standard fusion algorithm for IMU and GNSS. This type of fusion is referred to as global navigation.
- The fusion of IMU with external perception sensors, such as vision or laser sensors, falls within the field of Simultaneous Localisation and Mapping (SLAM), a well researched topic in robotics over the past two decades. We refer to this type of fusion as local navigation.
- Local navigation is considered more challenging in terms of fusion than global navigation, and has prompted the development of optimisation based fusion methods (namely smoothing) as a more accurate, but also more numerically demanding, alternative to the EKF. Nevertheless, those methods may be run in real time thanks to modern computers and specific solvers.
- This thesis was conducted in collaboration with Safran, which is the number one company for inertial navigation systems (INS) in Europe, and the number three worldwide, through a CIFRE convention, and therefore focuses on inertial navigation and SLAM, especially when using high-grade IMUs.
- It first focuses on an extension of the invariant EKF (IEKF), a variant of the EKF that has proved well suited to inertial navigation. Some further properties of this filter are proved regarding geometric constraints.
- The optimisation based (smoothing) counterpart of IEKF is introduced, namely invariant smoothing. It comes with properties the standard smoothing algorithms do not share. Simulations based on real data for both a mobile robot and an autonomous car illustrate the interest of the approach.
- In a second part, a novel linear least-squares solver is proposed for the linear systems that appear at each iteration of most smoothing algorithms, in the case where sensors used in the fusion problem are very precise, as occurs when using high-grade industrial IMUs. Once again, real world experiments illustrate the interest of the approach.

Cette thèse porte sur les algorithmes de localisation reposant sur la fusion de capteurs, en particulier le filtrage et le lissage, lorsque des senseurs inertiels de haute qualité sont présents. La première partie s'intéresse aux aspects non-linéaires de l'utilisation de ces capteurs, et démontre comment la structure non-linéaire du filtrage et du lissage peuvent être améliorés grâce au formalisme du filtrage invariant. La seconde partie s'intéresse aux problèmes rencontrés par les solveurs linéaires utilisés à chaque étape des algorithmes de lissage non-linéaires du fait de la présence de capteurs très précis. Un nouveau solveur pour moindres carrés linéaires permettant de résoudre ces problèmes est introduit.

1.2 Points marquants de la thèse

- La navigation inertielle repose sur l'utilisation de centrales inertielles (CI). Sur les cinquante dernières années, la fusion de CI avec d'autres capteurs ou sources d'information a été principalement réalisée à travers le filtre de Kalman étendu (EKF). Industriellement, ce filtre est devenu un standard pour la fusion de l'inertie et du GNSS. Ce type de fusion est appelé ici navigation globale.
- La fusion de l'inertie avec des capteurs de perception externes, comme la vision ou des scanners laser, rentre dans le cadre de la localisation et cartographie simultanées (SLAM), un domaine de recherche très actif en robotique durant les deux dernières décennies. Ce type de navigation est appelé ici navigation locale.
- La navigation locale est considérée comme plus difficile que la navigation globale, et a conduit au développement de méthodes de fusion basées sur de l'optimisation (dites lissage), alternatives plus précises mais aussi plus coûteuses que l'EKF. Néanmoins, des solveurs spécifiques permettent de les appliquer en temps réel sur les moyens de calcul actuels.
- Cette thèse a été menée en collaboration avec Safran, entreprise numéro un des systèmes de navigation inertielle (SNI) en Europe et numéro trois dans le monde, via une convention CIFRE. Elle se concentre donc sur la navigation inertielle et le SLAM, en particulier basés sur des SNI de haute qualité.
- Elle s'intéresse en premier lieu à une extension de l'EKF invariant (IEKF), une variante de l'EKF qui s'est révélée bien adaptée à la navigation inertielle. De nouvelles propriétés du filtre par rapport à des contraintes géométriques sont prouvées.
- L'homologue de l'IEKF basé optimisation (lissage) est introduit, à savoir le lissage invariant. Il présente des propriétés que les méthodes de lissage classiques ne partagent pas. L'intérêt de cette approche est illustrée par des simulations basées sur des données réelles provenant d'un robot mobile ainsi que d'un véhicule autonome.
- Dans un second temps, un nouveau solveur de moindres carrés linéaires est proposé pour les systèmes linéaires apparaissant à chaque itération de la majorité des algorithmes de lissage, dans le cas où certains capteurs du problème de fusion sont très précis, par exemple avec des SNI de haute qualité. Ici encore, des expériences sur données réelles illustrent l'intérêt de la méthode.

1.3 Industrial context

1.3.1 Inertial Navigation Systems since the 1950's

With the progress of transportation systems came the need of an ever increasing localisation accuracy. In this regard, the 1960s and the Apollo program were major milestones both on the software and the hardware sides with the emergence of the Kalman filter and the use of an inertial navigation system to guide the Saturn V rockets.

The Kalman filter [74] is a major tool in state estimation, which allows for unbiased sensor fusion minimum variance in the case of linear systems with Gaussian noises, ensuring powerful convergence properties. Its extension to non-linear cases, such as inertial navigation, known as the Extended Kalman Filter (EKF), became a de facto standard in navigation system up until today, although it is known to lose the optimality properties of the linear case in general.

Inertial Measurement Units (IMU) are navigation tools usually composed of several gyroscopes and accelerometers, measuring the body's orientation and specific acceleration (ie without the gravity) respectively. Given an initial position, speed and orientation, they allow tracking the evolution of this navigation triplet through time. Since the estimation is based on a triple integration of measurements subject to noise, the output will drift with time, with an amplitude depending on the sensor's quality. Therefore, inertial navigation progressed both by improving the inertial sensors, and by coupling them with other sensors, such as GNSS receivers (e.g., GPS antenna), to correct the accumulated drift. In this context, Safran, through its Electronics and Defense branch, is the No. 1 company in Europe and No. 3 worldwide for Inertial Navigation Systems (INS). It offers a range of IMUs of various accuracy, up until the Marine grade which needs almost no correction.

Historically, INS heavily relied on the EKF, fully aware of its shortcomings, because of its ease of use, and its practical reliability in a number of aerospace applications thanks to the accumulated experience. That is why Safran developed, in collaboration with Mines Paristech, a novel version of the EKF, namely the Invariant EKF (IEKF) based on the theory of Lie groups. Its theoretical and practical advantages were brought to light by the PhD thesis of Axel Barrau [9], in particular with high-grade IMUs, which serves as one of the starting points of this work.

The other motivation for this thesis is the recent evolution of the complimentary sensors INS are made of. Indeed, while GPS measurements have been at the heart of these systems for decades, the rise of applications needing localisation in places where the GPS is either unavailable or unreliable (e.g., tunnels and urban canyons) lead to the use of other complimentary sensors for local navigation. In particular, the fusion of inertia and vision, known as visual-inertial odometry or sometimes visual INS (VINS), recently became highly popular thanks to the decrease of inertial sensors' cost. Lidar-inertial navigation is also on the rise. GPS and visual measurements fundamentally differ, as the first one is a global observation of the system at a single time, i.e., of a single state, while the second often represents a relative observation between the states at two different times, for example a displacement. In this context the EKF is not sufficient anymore, and this lead to the development of new algorithmic tools to deal with these systems, in particular the Multi-State Constrained Kalman Filter (MSCKF) [86] and smoothing [59,80]. Both are based on the same idea, keeping old states in the estimation to be able to handle the relative measurements.

For those reasons, the present work focuses on the extension of the invariant framework to smoothing, and the ensuing computational issues which might appear when dealing with high-grade IMUs which were disregarded up until now.

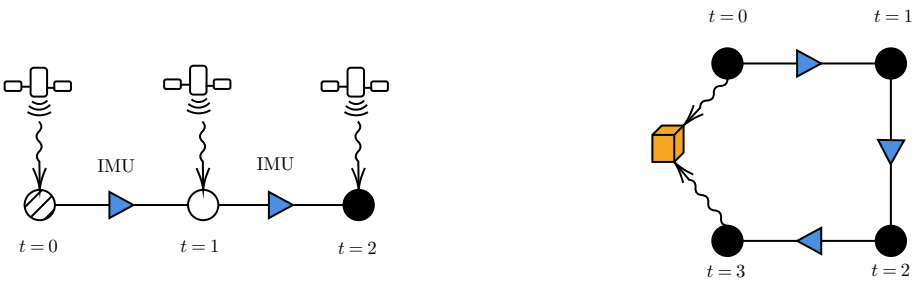
1.3.2 Safrantech's experimental autonomous car

Inside Safrantech, Safran's Research & Development center, an autonomous vehicle team was created roughly along the beginning of this thesis. They set up their own autonomous car prototype, illustrated on Figure 1.1, equipped with a number of navigation sensors, including a precise IMU. The creation of this prototype was an opportunity, both from the research and industrial points of view. Indeed, along with the hardware setting, the car prototype needs a navigation toolbox. It was decided to develop it mostly internally at Safrantech. Therefore, in order to effectively test and assess the interest of the different algorithms, this work also helped in the making of the toolbox, with the objective that it should be easily usable by Safran's navigation teams. In turn, having this toolbox allowed the prototype to serve as a test bench for the various estimation algorithms considered throughout these three years.

Figure 1.1: Autonomous vehicle prototype developed at Safrantech, which was used in the experiments.

1.4 Sensor fusion: global vs. local

For centuries, navigation was mostly thought as a way to localise oneself in a global frame, that is, compared to an identified reference position, e.g., the North pole. To this end, star tracking devices and compasses have been fundamental tools, which are still used today. The emergence of Global Navigation Satellite Systems (GNSS), such as the GPS, was a revolution in this field, as it allows getting position information on almost the whole planet. In the same time, local navigation became increasingly popular, in particular in the automation field, which required precise relative position information in places where no global positioning was available, or was not accurate enough, such as in buildings or tunnels. The problem of localising itself in a map which is being created, known as Simultaneous Localisation and Mapping (SLAM), is now a major research domain.



(a) Global navigation

(b) Local navigation

Figure 1.2: Global and local navigation representations. Global navigation relies on absolute information, e.g., GPS measurements, and the influence of former states on the current one gradually disappears. Local navigation is based on relative information, e.g., here seeing the same cube at $t = 0$ and $t = 3$, preventing the state to forget its past.

1.4.1 Inertial-aided navigation with global sensors

Navigation based on the fusion of inertial measurements and global observations have been used for decades in the guidance of spacecrafts, starting with the Apollo program during which the astronauts

manually tracked stars to correct the IMU's drift. If the IMU is precise enough, usually from tactical grade, it can even bring some global information by measuring the Earth's rotation. A particular property of global navigation, is that the state gradually loses its dependency to the past ones, i.e., it tends to "forget" where it comes from. This is intuitive when thinking about GPS measurements: if we have position information on ten minutes of navigation, knowing the first position becomes almost irrelevant. Mathematically, this comes from the assumption that the last GPS measurement received only depends on the current state, not the other ones, which is called the Markovian assumption and will be detailed in Section 4.1.2. This is illustrated on Figure 1.2(a).

1.4.2 Inertial-aided navigation with local perception

Local navigation can not forget its past

Navigation based on local sensors, such as cameras, or laser sensors (LiDar) is fundamentally different from the global case, because of two reasons:

- There is information which will remain unknown
- The past is never forgotten

Think of a building that is being discovered. Knowing that it lies in Paris or New-York has no impact on the fact that one is currently at the second floor. Therefore, there will be no way to discover where one is globally, or in which direction north is. The whole navigation is done with respect to the building's entrance. A more radical example is the follwing: put a robot inside a cube with a target for it to see. Even if you rotate or move the cube, the robot will see the exact same thing. The problem is called symmetric (a more rigorous definition will be given in Section 5.2), as the cube's position and orientation cannot be deduced from observations. This symmetry is at the heart of the problem, and drove a significant amount of work in this field [49]. In general, local navigation is thus carried out *with respect to its initial state*, that is, it is fixed and defines the reference frame in which the mobile will be located. For convenience, it is usually fixed at the origin. For instance, saying the robot is at point $(2,1,0)$ at time $t = 10s$, means that it is 2 meters away along the first axis (say forward), and 1 meter along the second (say left), of its initial position. Using inertial sensors slightly tempers this, since it is sensitive to gravity and therefore helps finding the vertical, yet the global heading remains unknown. This naturally leads to the fact that the system never forgets its past. Indeed, knowing that the hall the system goes through is the building's entrance is crucial in navigation, as it allows correcting some drift that the estimation suffered, a process called "loop closing" [3]. This is illustrated on Figure 1.2(b).

SLAM or Bundle adjustment? The different types of local navigation

Local navigation exists under various forms, depending on the sensors used and the chosen framework. Concerning the sensors, it ranges from mono sensor (usually visual or LiDar) to multi-sensor navigation, with or without an IMU. There are mainly two different frameworks, the ones explicitly estimating the local map through features, and those implicitly converting them into relative measurements of the state. Looking at Figure 1.2(b), the former would explicitly seek the orange cube, while the latter would turn the measurements of the cube from times 0 and 3 to a relative measurement between the corresponding states. Table 1.1 gives an overview of the zoology of methods

Note that the nomenclature between odometry and SLAM is not yet stabilised, as some papers relying on implicit relative measurements (here denoted as odometry) use the term SLAM, e.g., [94]. Some authors proposed simplified names, for instance in [78] the terms bundle-adjustment is used to cover visual-inertial odometry.

Nomenclature used in this thesis: To avoid confusion, **SLAM is used as a general term** to cover the whole local navigation framework. When more specific problems are addressed, the involved sensors will be given, and the methods will be either called **pose-graph based** or **features based**, depending on whether or not the features are explicitly estimated. Note that it will mostly be the former, and that this name comes from the notion of factor graph, tightly bound to SLAM, which will be detailed in Section 4.1.

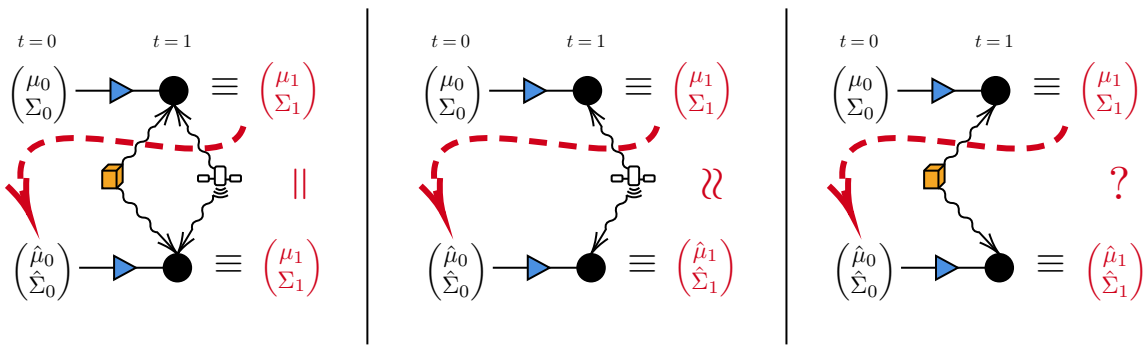
Method name	Camera	LiDar	IMU	Estimates features	Example
VO (mono or stereo) - BA	1 or more				[54]
V-SLAM (mono or stereo)	1 or more			×	[87]
LO		×			[104]
L-SLAM		×		×	[105]
VLO	1 or more	×			[68]
VINS or VI-SLAM	1 or more		×	×	[91]
VIO	1 or more		×		[53]
LIO		×	×		[56]
VILO	1 or more	×	×		[94]

Table 1.1: Zoology of local navigation methods. The letters in the acronym have the following meaning: BA - Bundle Adjustment, V - Visual, I - Inertial, L - LiDar/Laser, O - Odometry, NS - Navigation System

The particular case of inertia-odometer fusion

The fusion of IMU and odometers does not perfectly follow the classification given above. Indeed, although it is a local navigation system with symmetries, it resembles much more the inertia-GPS fusion , as it is also based on unary measurements. This is due to the fact that it does not use perceptive sensors such as cameras or lasers.

1.4.3 Keeping memory of the past in the presence of non-linearities



(a) Linear systems

(b) Non-linear global navigation

(c) Non-linear local navigation

Figure 1.3: Illustration of the differences between navigation for linear and non-linear systems. (a) Whether global or local, navigation for linear systems is optimally by simply computing the mean and covariance of the distribution at each time. Even if new measurements could help better estimating the past, it will have no influence on the distribution of the current estimate. (b) In global navigation with non-linear systems, since the influence of former states on the current one gradually disappears, correcting the former states will only slightly modify the current estimate. Reestimation of former states is not necessary in general (c) Since local navigation never forgets its past, the correction of former states can greatly influence the current estimate. Reestimation of former states can be crucial.

For linear systems, the past lies in the covariance

There is a case in which the memory of the past can be fully and efficiently maintained along the trajectory: if the system is linear, the noises follow Gaussian distributions and are mutually independent (i.e., the measurements are Markovian, they only depend on the state they are related to). Indeed, if these assumptions are satisfied, then the state which is to be estimated keeps on having a Gaussian distribution, relative to the obtained measurements. This is highly practical as this distribution, also called the normal distribution, only depends on two parameters, a vector $\mu \in \mathbb{R}^d$ called the mean of the

distribution, and a positive definite matrix $\Sigma \in \mathbb{R}^{d \times d}$ called its covariance. Indeed, its density is given by

$$p_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{d/2}|\det(\Sigma)|^{1/2}} \exp(-\frac{1}{2}\mu^T \Sigma^{-1} \mu) \tag{1.1}$$

Therefore, this distribution is denoted as $\mathcal{N}(\mu, \Sigma)$. The fact that the state is at all times normally distributed means that these two parameters *fully encode the influence of the past on the current state*. In practice, only the mean and the covariance need to be computed at each step, turning the estimation problem into one of linear algebra. This is the heart of the well-known Kalman filter [74].

Non-linear systems need relinearisation

The main challenge of navigation in general is that the considered systems are usually non-linear. Therefore, even if the noises are supposed to be gaussian, the state is not normally distributed anymore. This makes the state estimation problem much harder in general. Efficient techniques exist to approximate the problem, two in particular are often used in navigation. On the one hand, filters based on standard first-order linearisations of the system, which maintain a mean and a covariance. On the other hand, particle filters which approximate the state's distribution through a finite set of samples. Both have achieved remarkable results in global navigation. Although they were among the first proposed solutions for local navigation problems, thanks to their efficiency, they exhibited major flaws which made them inconsistent in the long-run [3,71]. This is due to the fact that these methods are well-suited for systems which forget their past [44], a property which is not satisfied by SLAM. This is illustrated on Figure 1.3.

What makes the linearisation-based filters fail is the fact that they rely on approximated linearisation of the system, to propagate and update their estimate, in particular its covariance. Indeed, if it was linearised around the true state, then no problem would occur [48]. To cope with these errors and avoid their accumulation, a solution is to relinearise them according to the additional information obtained later. In turn, this requires the past states to be explicitly reestimated, which gave rise to smoothing, a class of methods which is at the heart of this thesis.

Smoothing, a modernised old idea

Smoothing methods all share the same basic idea: casting the estimation as an optimisation problem, in the form of non-linear least-squares. Least-squares have been a well-known tool for centuries, as this is in essence the method Gauss used to determine Ceres' orbit in the late XVIIIth century. Indeed, they naturally arise when devising statistical estimators based on the Maximum Likelihood or Maximum A Posteriori in the presence of Gaussian noises. It was also the proposed method in the pioneer work of [80] on local navigation. However, it long proved intractable in practice, which is why filter-based methods were preferred for a time [49].

The theoretical shortcomings inherent to the filtering methods used at the time, along with the increase of computational power and some major breakthroughs considering smoothing complexity allowed to consider it in practice. Indeed, the least-squares of smoothing are *sparse*. That is, the theoretical numerical complexity rises linearly with the number of states, and not quadratically. The development of sparsity-preserving algorithms for smoothing made it the most used framework for local navigation [34]. Its successes lead to some applications in global navigation [106].

Non-linear filtering can also encode memory of the past

Recent advances in the theory of non-linear filters showed that, when considering distributions inspired from the Gaussian but specifically designed to take symmetries into account, theoretical results similar to those of the linear case could be derived [23]. This was first applied in global navigation [24] to derive more efficient Kalman-like methods. In turn, this showed that there was a kind of filters which could efficiently encode the past of the system, and carry out reliable estimation of local navigation in the long run [11]. This framework, based on the theory of Lie groups, is known as Invariant Filtering, and especially the Invariant Extended Kalman Filter (IEKF).

1.4.4 The smoothing framework, as studied in this thesis

A general navigation system can usually be decomposed into two more or less independent parts, the front-end and the back-end. The former processes the raw data given by the sensors in order to create

measurements corresponding to a certain mathematical model. The latter is fed with them and outputs the useful navigation information. Typically, GNSS antenna turning the continuous signals into pseudo-measurements and then position measurements, or stereo image processing tools tracking the range and bearing of visual features are part of the navigation front-end. On the contrary, the EKF is a back-end.

This thesis focuses on the back-end part of navigation. As will be carefully explained in Section 4.1, the estimate of smoothing is defined through a non-linear optimisation problem. In practice, the back-end must carry out a variety of tasks in order to produce a reliable estimate with tractable computational complexity, such as dealing with spurious data or choosing which one to drop in the long term, but these aspects are not considered here.

In this work, only the core part of smoothing is considered: the formulation and solving of the non-linear optimisation problem. Since it is not linear, iterative methods have to be used to find the estimate. That is, at each iteration, the system is linearly approximated around the current guess, leading to a linear system whose solution provides the next correction to be applied. Therefore, it appears that a smoothing algorithm can be divided into two main parts, the non-linear and the linear ones. The former dictates how the Jacobians and errors are computed, and how the updates are handled, while the latter is the solver, only concerned with linear algebra, as illustrated on Figure 1.4. This thesis brings forth new results for both aspects separately.

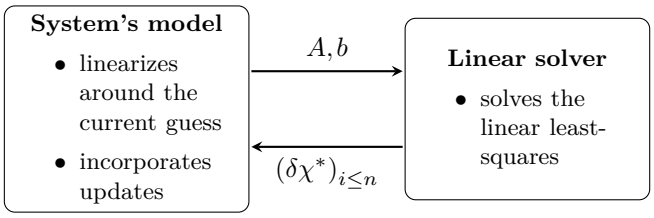


Figure 1.4: Simplified schematic representation of the two main components of a smoothing algorithm considered here, the non linear model of the system which handles the transitions from the manifold to the tangent space and back, and the solver which inverts the obtained linear system.

For the readers interested in the aspects of the smoothing back-end that are not considered here, it is recommended to focus on the SLAM bibliography. Indeed, although SLAM does not cover the whole navigation domain, most of the current state-of-the-art back-end techniques initially came from it. Several comprehensive and well-documented reviews were published on SLAM in general. For a historical point of view of its origins, see [3,49]. This was updated by two recent reviews. On the one hand, [97] completed the work of Durrant-Whyte and Bailey with recent advances, listing in particular a number of SLAM back-ends. On the other hand, [34] gives an overview of the various questions which were raised by SLAM research, its various applications, and the remaining challenges. Back-end in general is covered in [70], and its extension covering solely the optimisation aspects [69]. [66] specifically covers the latest developments of visual-inertial odometry. A review of SLAM from the autonomous car point of view is given by [28,96], the latter including a security aspect. Finally, [65] lists a large number of existing SLAM algorithms existing with different sensor configurations.

1.5 Contributions and organisation of the thesis

This thesis builds on the two recent advances in local and global navigation described in the preceding Section, smoothing and Invariant filtering. Both are solutions which were proposed to overcome the inherent difficulties of different aspects of navigation. While smoothing allowed to efficiently tackle SLAM, Invariant filtering proved highly efficient in the context of precise global navigation [13,14]. The starting point of this thesis was the following question: **Can we combine the benefits of smoothing and the Invariant framework, in particular when precise motion sensors are present?**

1.5.1 Contributions

This initial question lead to study both aspects of smoothing: its non-linear formulation and its linear part appearing in the solving process. Nevertheless, this work started by further exploring the properties

of Invariant filtering. Therefore, this thesis proposes three contribution: one on the IEKF, and one for each aspect of smoothing. All three focus on precise inertial navigation.

Invariant Filtering with geometric constraints

The first contribution of this work is about Invariant filtering. The IEKF is particularly well-suited for a particular class of systems, called linear-observed systems on groups [18]. This thesis reports another property of the IEKF, its ability to respect geometric constraints without having to artificially enforce them. This can be seen as a limit-case of highly precise sensors, in the presence of which the IEKF had been shown to outperform the EKF [14]. It shows that, using the invariant framework, there exists a continuum between highly precise and deterministic motion.

Invariant Smoothing

The main contribution of this work on the non-linear aspects of smoothing is the extension of Invariant Filtering to this framework. Building on the properties of this filter lead to Invariant Smoothing. Just as there exist two versions of the IEKF, the left and right ones, two Invariant smoothing algorithms can be designed: Left-Invariant smoothing (L-IS) and Right-Invariant smoothing (R-IS). In fact, a third one, called here Hybrid IS, can be defined. L-IS exhibits a particularly interesting feature in global navigation: the matrix involved in the linear system that is to be solved in smoothing can be considered, under mild assumptions, *independent from the current estimate*. This ensured not only computational advantages, but also better behavior in practice compared to other non-invariant methods: quicker convergence and the avoidance of some local minima.

The application of Invariant smoothing to inertial navigation is studied in details. At first glance, IMUs are not well-suited for smoothing-based estimation, as their output frequency is usually substantially higher than that of the other sensors, which would imply to repropagate past states through all the increments at each estimation step. Thanks to the notion of the notion of preintegration, providing means of computing this repropagation in constant times, the estimation was made tractable [53,81]. Preintegration was shown to be a natural property in the Invariant framework [17,18], simplifying its expressions. Leveraging this feature allows outlining the differences between the existing non-linear - but non-invariant - smoothing methods and the ones presented here.

Further theoretical properties of Invariant Smoothing are presented. First, the ability of the IEKF to respect geometric constraints is extended to Invariant Smoothing. Simulations show that this can have a massive impact when using very accurate sensors. Then the observability of local inertial navigation is studied, proving the superiority of R-IS over L-IS in this context. Moreover, considering observability from a smoothing point of view unveiled the unexpected impact of the prior covariance.

A new linear solver for highly precise navigation

The third main contribution of this work, independent of the first ones, lies in the linear part of smoothing. Indeed, its estimate is computed through iterative methods, which invert successive linear systems. The existing solvers managed to leverage the intrinsic sparsity of navigation problems to bring efficient solutions. However, they are all based on formulations involving the inverse of the covariance matrices associated to each sensor. Thus they are fundamentally unable to consider perfect motion sensors, for instance. While that is only an ideal case, this also means that these methods will fail given accurate enough sensors. Since this is exactly the focus of this thesis, this observation made it necessary to develop a new linear solver which would be adapted to such systems. Moreover, this solver had to keep a complexity close to the ones of the state-of-the-art, in order to still be tractable. Therefore, it also had to be sparsity-preserving. This was achieved thanks to three components. First, the fact that the Kalman smoother actually computes the minimum of a linear least-squares problem thanks to its optimality properties [92]. Second, another formulation of the Kalman smoother, which avoids inverting the prior and propagation covariance matrices, called Backward Information Forward Marginal (BIFM). While BIFM was proposed in the signal processing domain [79], another simpler formulation is presented here, explicitly defining the forward Kalman filter, the backward information filter, and their final fusion. Finally, the method of stochastic cloning, which allows turning a relative observation problem into one suitable for standard Kalman filtering techniques [85]. This new solver is thus called Stochastic-Cloning BIFM (SC-BIFM). It is validated both in simulation and in an experiment on real data which actually revealed the limits of the existing methods.

The BIFM formulation of the Kalman smoother also allowed to devise a new non-linear estimation algorithm leveraging the IEKF and the Information filter on Lie groups [50]. Indeed, the BIFM is made of a Kalman filter, an information one, and a final fusion step. It thus appeared natural to extend it to non-linear filters. However, it was not as obvious to do so while not inverting the covariance matrices at hand. Indeed, in this case, the linear part of the estimation comes from first-order (sometimes exact) linearisations, which define *updates* of the estimates. Once this was realised, this lead to a new estimation method for systems on Lie groups, which is based on update refinements.

Industrial application: experimenting on the autonomous prototype

A significant part of this thesis was dedicated to testing the proposed methods on industrial data of Safran. In particular, one of the objectives was to implement them into the toolbox used by the autonomous navigation Lab. The aim of the experimental phase was thus two-fold: validating the proposed methods, and their match with the toolbox' structure. Indeed, it was not just a question of extracting the collected data, try each estimator on them and post-process the result independently; but a question of integrating them directly into the pipeline of the toolbox. This way, it ensures that the code which has been validated will serve as a basis for the future evolution of the autonomous prototype. While this necessarily confronted the standard software engineering issues, this allowed to take an active part into structuring the toolbox and creating versatile code, easily extendable to take into account new sensors, and usable by the engineers. Having this experimental platform containing a high-grade IMU enabled to test the advantages and limits of each methods. For Invariant smoothing, it outlined the benefits of using the exponential update strategy, and the superiority of R-IS in local navigation. In addition, experiments run on simulated data showed that L-IS overcame some of the shortcomings the IEKF has in alignment problems in the presence of strong biases. For SC-BIFM, experiments run with single precision computations showed both the limits of the standard methods, which actually failed to conduct the estimation, and the robustness of this new linear solver, which managed to compute the estimate with sufficient accuracy during the whole trajectory.

1.5.2 Organisation of the document

This thesis is separated into two parts.

Part I is dedicated to the extension of Invariant filtering to the smoothing framework. Chapter 2 introduces the mathematical notions and the models considered in the theory of Invariant filtering. The IEKF and its properties, including its respect of geometrical constraints, are presented in Chapter 3. Then this framework is extended to derive Invariant Smoothing in Chapter 4. L-IS exhibits interesting properties in practice, including a reduced (and sometimes cancelled) dependency to the estimate, which brings better behavior in experiments. The recent invariant view on preintegration theory is leveraged to be applied to IS and compare it with other non-linear smoothing methods. Further theoretical results on Invariant Smoothing are derived in Chapter 5. First, the properties of the IEKF with respect to constraints, exposed in Chapter 3, are extended to Invariant Smoothing. Then, a novel look on unobservability of smoothing is proposed, highlighting the role of the prior covariance, especially in inertial navigation. R-IS proves much more adapted to local navigation than L-IS. Chapter 6 reports the results of experiments conducted on real-world data and which illustrate Invariant Smoothing's properties.

Part II focuses on the linear solver used in smoothing, and their behavior in the presence of highly precise sensors. Their shortcomings in such cases are outlined in Chapter 7. Then the new proposed solver, SC-BIFM, is first detailed and then validated in simulation in Chapter 8, where it is shown that it is indeed both sparsity-preserving and can tackle very low or even zero propagation noise. Real-world experiments, assessing both the decay of standard solvers and the robustness of the proposed one, are reported in Chapter 9. Finally, the ideas of SC-BIFM are leveraged to devise a new non-linear estimation method on Lie groups. This second part is largely independent from the first one. In particular, it does not require any Lie groups knowledge.

1.5.3 Publications

This thesis lead to the publication of two articles in the proceedings of international peer-reviewed conferences, the submission of a third one, and the submission of one journal paper to an international journal

- Paul Chauchat, Axel barrau and Silvère Bonnabel, Kalman Filtering with a class of state equality constraints, In IEEE 56th Annual Conference on Decision and Control, CDC 2017, Melbourne, Australia
- Paul Chauchat, Axel barrau and Silvère Bonnabel, Invariant Smoothing on Lie Groups, In IEEE/RSJ International Conerence on Intelligent Robots and Systems, IROS 2018, Madrid, Spain
- Paul Chauchat, Axel barrau and Silvère Bonnabel, Invariant Smoothing in the face of geometric constraints, IEEE/RSJ International Conerence on Intelligent Robots and Systems, IROS 2020, submitted
- Paul Chauchat, Axel barrau and Silvère Bonnabel, Factor graph based smoothing without matrix inversion for highly precise localization, IEEE Transactions on Control Systems and Technology (TCST), accepted.

In addition to these academic publications, the presented work led to a patent application currently pending.

Part I

From Invariant Filtering to Invariant Smoothing

Chapter 2

Autonomous errors on Lie groups

Highlights

- The main points which serve the theory of invariant Kalman filtering are reviewed. In particular group-affine systems, which is a class of dynamical systems that come with remarkable mathematical properties playing a great role when trying to estimate the state of such systems. Inertial navigation systems are -to some extent- group-affine.
- All group-affine systems share the property that they propagate a non-linear error between the state's estimate and the true one independently of their values. Moreover, coupled with observations in the form of group action, this error becomes autonomous, which is in contrast with the usual features of the conventional EKF. This property is at the heart of stability guarantees of the IEKF which will be presented in Chapter 3.

Points marquants

- Les principes de base de la théorie du filtrage de Kalman invariant sont rappelés, en particulier les systèmes groupe-affines. Cette classe de systèmes dynamiques présente des propriétés mathématiques remarquables jouant un grand rôle dans l'estimation de l'état de tels systèmes. La navigation inertielle est - dans une certaine mesure - groupe-affine.
- Tout système groupe-affine propage une erreur non-linéaire entre le vrai état et son estimé indépendamment de leurs valeurs respectives. De plus, couplé à des observations modélisées par des actions de groupe, cette erreur devient autonome, ce qui les distingue des propriétés usuelles de l'EKF conventionnel. Cette propriété est au cœur des résultats de stabilité de l'IEKF qui sera présenté au Chapitre 3.

Introduction (En/Fr)

This chapter introduces the mathematical concepts that will be used throughout this part. It starts with a brief introduction to Lie groups, in which the states of navigation systems typically live, as is shown with some examples. Then, the crucial notions of group action and group-affine dynamics, the latter being fairly recent [18], are presented. Finally, the properties of systems defined using these notions are outlined, and in particular the fact that they can lead to errors being autonomous, a strong feature which will be leveraged in the design of estimators in the following chapters. Some examples of navigation systems which fit into this framework are given, in particular (unbiased) inertial navigation.

Ce chapitre introduit les concepts mathématiques utilisés tout au long de cette partie. Les groupes de Lie sont brièvement introduits, avec des exemples montrant que les systèmes de navigation y évoluent naturellement. Puis les notions cruciales d'action de groupe, et de dynamique groupe-affine sont présentées, cette dernière étant récente [18]. Enfin, les propriétés des systèmes basés sur ces notions sont soulignées, en particulier les erreurs autonomes vers lesquelles ils mènent, une caractéristique forte sur laquelle s'appuiera la conception des estimateurs des chapitres suivants. Quelques exemples de systèmes de navigation rentrant dans ce cadre sont donnés, en particulier la navigation inertielle (sans biais).

2.1 Mathematical preliminaries

2.1.1 Lie Groups

In this section we recall the definitions and basic properties of matrix Lie groups, Lie algebra and random variables on Lie groups. A matrix Lie group $G \subset \mathbb{R}^{n\times n}$ is a set of square invertible matrices that is a group, i.e., the following properties hold:

$$I_N \in G;\ \forall \chi \in G, \chi^{-1} \in G;\ \forall \chi_1, \chi_2 \in G, \chi_1\chi_2 \in G \tag{2.1}$$

Matrix Lie groups are particular examples of Lie groups, which are sets being

- groups, i.e., have an inner product, an inverse function, and a neutral element (e.g., matrix product, matrix inverse and identity for matrix Lie groups);
- manifolds, i.e., one can derive on them (think of a disk for instance);
- and such that the inner product and the inverse function are smooth for the manifold structure. For matrices, this is ensured thanks to the closed group theorem [55].

Although Lie group theory can get quite involved, all the considered sets in this thesis are matrix Lie groups, therefore we will restrict to them.

Locally about the identity matrix I_N, the group G can be identified with a vector space $\mathbb{R}^d$ using the matrix exponential map $\exp_m(.)$, where $d = \dim G$. Indeed, to any $\xi \in \mathbb{R}^d$ one can associate a matrix $\xi^\wedge$ of the tangent space of G at I_N, called the Lie algebra $\mathfrak{g}$. We then define the exponential map $\exp : \mathbb{R}^d \to G$ for Lie groups as

$$\exp(\xi) = \exp_m\left(\xi^\wedge\right), \tag{2.2}$$

Locally, around 0, it is a bijection, and one can define the Lie logarithm map $\log : G \to \mathbb{R}^d$ as its inverse: $\log(\exp(\xi)) = \xi$. This is illustrated on Figure 2.1.

Major assumption

In the remainding of this thesis, it will be assumed that the considered quantities lie in the subspaces of G and $\mathfrak{g}$ on which exp and log are bijections. In pratice, this will only have an impact on the rotational parts of the navigation states at play. For instance, for 2D rotations, whose logarithm is given by their angle of rotation, this simply means that only angles in $[0, 2\pi[$ will be considered.

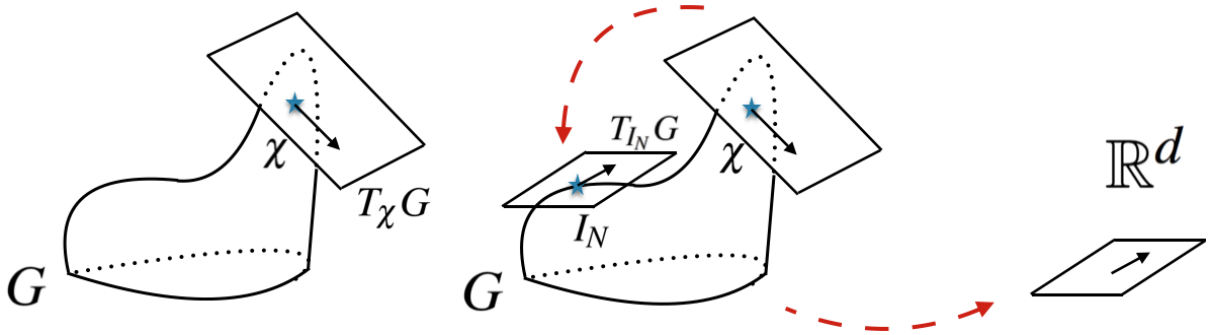


Figure 2.1: From [16]. G is a curved space. Left and right multiplications offer two ways to identify the tangent space $T_\chi G$ at χ with the tangent space at Identity $T_{I_N} G$, called the Lie algebra $\mathfrak{g}$. In turn, the application $\xi \mapsto \xi^\wedge$ provides a linear bijection between the Euclidean space $\mathbb{R}^d$ and $\mathfrak{g}$.

The Lie algebra $\mathfrak{g}$ is endowed with a bilinear map $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, called Lie brackets. For a matrix Lie algebra $\mathfrak{g} \subset \mathbb{R}^{n\times n}$, the brackets are based on the commutator of matrices: $[A, B]_\mathfrak{g} = AB - BA$. Using this, they are defined for $\xi, \zeta \in \mathbb{R}^d$ through $[\xi, \zeta]^\wedge = [\xi^\wedge, \zeta^\wedge]_\mathfrak{g}$. As Lie groups are not necessarily commutative, in general we have $\exp(x)\exp(y) \neq \exp(x+y)$ for $x, y \in \mathfrak{g}$. However, the vector z such that $\exp(x)\exp(y) = \exp(z)$ satisfies the Baker-Campbell-Hausdorff (BCH) formula [55] $z = BCH(x, y) =$

$x+y+r(x,y)$, where $r(x,y)$ contains higher order terms made of nested Lie brackets. When studying groups, maps of particular interest are those which respect the structure given by the group product, called morphism, and especially the bijective ones, called automorphism.

Definition 1. *An automorphism of G is an invertible map $\phi : G \to G$ such that:*

$$\forall a,b \in G \ \phi(ab) = \phi(a)\phi(b)$$

In particular, this implies that $\phi(Id) = Id$ and $\phi(a^{-1}) = \phi(a)^{-1}$ for all $a \in G$.

A last classical tool is the (inner) automorphism $\Psi_a \in Aut(G)$ defined for each $a \in G$ as $\Psi_a : g \mapsto aga^{-1}$. Its differential at the identity element Id of G is called *adjoint operator* and denoted by $Ad_a : \mathfrak{g} \mapsto \mathfrak{g}$. It satisfies:

$$\forall a \in G, u \in \mathfrak{g}, \ a\exp(u)a^{-1} = \exp(Ad_a u).$$

This is of paramount interest, because it gives the correction term which apply to commutation, by simply rewriting $\forall a \in G, u \in \mathfrak{g}, \ a\exp(u) = \exp(Ad_a u)a$.

The estimators which are proposed in this work come from the striking properties of a specific class of systems, called linear observed systems on groups [18], which rely on the notions of group automorphisms, given in Definition 1, and group actions, which are the extension of linear maps on vector spaces:

Definition 2. *Let G be a group and Y an arbitrary set. A left (resp. right) group action is an operation denoted by $\star$*

$$\begin{aligned} G \times Y \to Y, \qquad & (x,b) \mapsto x \star b \\ \textit{(resp.)} \ Y \times G \to Y, \qquad & (b,x) \mapsto b \star x \end{aligned}$$

which satisfies for all $x_1, x_2 \in G, \ b \in Y$:

$$x_2 \star (x_1 \star b) = (x_2 x_1) \star b, \qquad Id \star b = b \tag{2.3}$$

$$\textit{(resp.)} \ (b \star x_2) \star x_1 = b \star (x_2 x_1), \qquad b \star Id = b \tag{2.4}$$

Although it can seem abstract, this definition encompasses three very natural observation types which cover almost all cases considered in this thesis, but which are defined on different spaces. Consider $\chi, \chi_1, \chi_2 \in G$:

- Actions on a vector $d \in \mathbb{R}^n$: $h(\chi) = \chi d$ and $h(\chi) = \chi^{-1} d$.
- Actions on a state $\alpha \in G$: $h(\chi) = \alpha\chi^{-1}$ and $h(\chi) = \chi^{-1}\alpha$.
- Actions of the cartesian product $G \times G$ on a state: $h(\chi_1, \chi_2) = \chi_1 \alpha \chi_2^{-1}$ and $h(\chi_1, \chi_2) = \chi_2^{-1} \alpha \chi_1$.

The first explicitly exploits the fact that χ lives on a matrix Lie group. d is chosen to retrieve a part of χ or χ^{-1}, one of its columns for instance, and can thus model absolute position measurements as in Section 3.5.1, or observations of known vectors (gravity, Earth magnetic field) in its own frame. On the other hand, the second and last one are strictly intrinsic, and will be used to model full state, or relative state measurements.

The case of partial relative measurements

In some cases, the relative observations do not involve whole states, but only parts of it. The most common example is relative pose measurement in inertial navigation. In this case, the observation model writes

$$h(\chi_1, \chi_2) = \pi(\chi_2^{-1})\pi(\chi_1), \qquad \pi : G \to H, \tag{2.5}$$

where $H \subset G$ is a subgroup. This is also engulfed in the group action framework, as show the following result.

Proposition 1. *If $\star_H$ is an operator which defines a group action on a subgroup H of G, then for all morphism $\phi : G \to H$,*

$$\tilde{h}(\chi) = \phi(\chi) \star_H \phi(\alpha) \tag{2.6}$$

defines a group action on G.

Group action	Transformed noise	Innovation noise covariance
$h_L(\chi) = \chi d,\ d\ \in \mathbb{R}^n$	$\hat{\chi}^{-1} v$	$\hat{\chi}^{-1} N \hat{\chi}^{-T}$
$h_R(\chi) = \chi^{-1} d$	$\hat{\chi} v$	$\hat{\chi} N \hat{\chi}^{T}$
$h_L(\chi) = \alpha \chi^{-1},\ \alpha \in G$	v	N
$h_R(\chi) = \chi^{-1} \alpha$	v	N
$h_L(\chi_1, \chi_2) = \chi_1 \alpha \chi_2^{-1}$	$Ad_{\hat{\chi}_2}^{-1} v$	$Ad_{\hat{\chi}_2}^{-1} N Ad_{\hat{\chi}_2}^{-T}$
$h_R(\chi_1, \chi_2) = \chi_2^{-1} \alpha \chi_1$	$Ad_{\hat{\chi}_1} v$	$Ad_{\hat{\chi}_1} N Ad_{\hat{\chi}_1}^{T}$

Table 2.1: Innovation noise for three types of group actions. Each type is considered in its left- and right-group action form, denoted respectively by the subscript L and R

2.1.2 Uncertainties on Lie Groups

To define random variables on Lie groups, we cannot apply the usual approach of additive noise for $\chi_1, \chi_2 \in G$ as G is not a vector space, i.e., generally $\chi_1 + \chi_2 \notin G$ does not hold. In contrast, we adopt the framework of [7], see also [12], which is slightly different from the pioneering approach of [38,39]. Indeed, we define the probability distribution $\chi \sim \mathcal{N}_L(\bar{\chi}, \mathbf{P})$ for the random variable $\chi \in G$ as

$$\chi = \bar{\chi} \exp(\xi),\ \xi \sim \mathcal{N}(\mathbf{0}, \mathbf{P}), \tag{2.7}$$

where $\mathcal{N}(.,.)$ is the classical Gaussian distribution in Euclidean space and $\mathbf{P} \in \mathbb{R}^{q \times q}$ is a covariance matrix. In the sequel, we will refer to (2.7) as the left-invariant Gaussian distribution on G, owing to the fact that the discrepancy $\bar{\chi}^{-1}\chi$ which is invariant to left multiplications $(\bar{\chi}, \chi) \mapsto (\Gamma\bar{\chi}, \Gamma\chi)$, is the exponential of a Gaussian ξ. In (2.7), the noise-free quantity $\bar{\chi}$ is viewed as the mean, and the dispersion arises through left multiplication with the exponential of a Gaussian random variable. Similarly, the distribution $\chi \sim \mathcal{N}_R(\bar{\chi}, \mathbf{P})$ can be defined through right multiplication as

$$\chi = \exp(\xi)\, \bar{\chi},\ \xi \sim \mathcal{N}(\mathbf{0}, \mathbf{P}). \tag{2.8}$$

It should be stressed out that these probability density functions are defined directly in the vector space $\mathbb{R}^d$ such that both $\mathcal{N}_L(.,.)$ and $\mathcal{N}_R(.,.)$ are not Gaussian distributions.

2.1.3 Uncertainties of group actions

The estimation carried out by the back-end relies on the noise associated to the measurements. Therefore, the uncertainties associated to measurements modeled by group actions, for instance (2.31) must be assessed. The case of vectorial group actions, which was originally considered in [14], and will be covered by (3.9), simply uses an additional gaussian noise. In general, the uncertainty model will depend on the set Y in which the measurements live. In this work, all considered measurements live in groups (which also include vector spaces), and uncertainties are modeled in a same way as in Section 2.1.2:

$$h(\chi, v) = (\chi \star b) \exp(v) \tag{2.9}$$

However, as the $\star$ operator has in general no link with the group operation of Y, the noise associated to $\hat{\chi}^{-1} \star h(\chi, v)$ or $h(\chi, v) \star \hat{\chi}^{-1}$ must be computed in each case. Therefore, we write in the general case

$$\hat{\chi}^{-1} \star h(\chi, v) = (\hat{\chi}^{-1}\chi \star b) \exp(\hat{v}) \qquad h(\chi, v) \star \hat{\chi}^{-1} = (b \star \chi\hat{\chi}^{-1}) \exp(\hat{v}) \tag{2.10}$$

Table 2.1 gives the value of $\hat{v}$ for the models we are interested in. Since they are all linear transformations of the initial noise v_k, obtaining their covariance is straightforward.

2.2 The Lie groups of navigation

The groups of 2D and 3D rotations $SO(2)$ and $SO(3)$, and of 2D and 3D rigid transformations $SE(2)$ and $SE(3)$ are classically used in robotics, see [6,7,39]. In this work, we consider the group $SE_2(3)$, which is a natural extension of them. It was recently introduced for IMU-based navigation in [14], and used for legged robot state estimation [62]. They are all matrix Lie groups, meaning that they can be embedded in matrix spaces, thus deriving their respective group operations from the matrix multiplication. Quaternions and dual quaternions are also often chosen as efficient Lie group representations, especially for attitudes and poses [5,51,57], but also features [83].

2.2.1 The group of 2D rotations $SO(2)$

The simplest Lie group used in navigation is the 2D special orthogonal group $SO(2)$, which encodes the heading R of a body living on a planar world.

$$SO(2) := \{R \in \mathbb{R}^{2\times 2} | R^T R = I_2,\ \det R = 1\}$$

This group is commutative, and admits a 1D representation through its canonical bijection with $[0, 2\pi[$, given by

$$R : \theta \mapsto \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

Its tangent space is given by $\mathfrak{so}(2) = \{\alpha J, \alpha \in \mathbb{R}\}$, where $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. Its exponential coincides with the angle-rotation matrix bijection R defined above.

2.2.2 The group of rotations $SO(3)$

The most classical Lie group used in navigation is the 3D special orthogonal group $SO(3)$, which encodes the orientation R of a rigid body in space:

$$SO(3) := \{R \in \mathbb{R}^{3\times 3} | R^T R = I_3,\ \det R = 1\}$$

Unlike the 2D case, $SO(3)$ is not commutative. The tangent space around identity is $\mathfrak{so}(3) = \{A \in \mathcal{M}_3(\mathbb{R}), A = -A^T\}$, the space of skew-symmetric matrices. Since 3×3 skew-symmetric matrices are isomoprhic to $\mathbb{R}^3$, the isomorphism $\omega \mapsto \omega^\wedge$ between $\mathbb{R}^3$ and $\mathfrak{g}$ is simply given by $\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}^\wedge = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}_\times = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$. The exponential mapping is given by well-known Rodrigues formula:

$$\exp(\omega) = I_3 + \frac{\sin(||\omega||)}{||\omega||}(\omega)_\times + 2\frac{\sin(||\omega||/2)^2}{||\omega||^2}(\omega)_\times^2$$

Its inverse log is the vector associated to the skew-symmetric matrix defined as:

$$\log(R) = \frac{\theta}{2\sin(\theta)}(R - R^T), \qquad \theta = \arccos\left(\frac{Tr(R) - 1}{2}\right)$$

Finally, Ad_R is simply given by:

$$Ad_R = R$$

2.2.3 The group of 2D direct spatial isometries $SE(2)$

$SO(2)$ is extended as the group of 2D direct spatial isometries, $SE(2)$, which is used to represent the set of planar orientation-position (x, θ) pairs, called poses, through a matrix embedding

$$SE(2) := \{T = \begin{pmatrix} R(\theta) & x \\ 0_{1,2} & 1 \end{pmatrix} \in \mathbb{R}^{3\times 3} \mid (\theta, x) \in [0, 2\pi[\times\mathbb{R}^2\}.$$

As $SO(3)$, $SE(2)$ is not commutative. The exponential, logarithm and adjoint operators on $SE(2)$ are defined for $\chi \in SE(2)$ and $\xi \in \mathfrak{se}(2)$ as

$$\exp(\xi) = \begin{bmatrix} R(\xi^3)\ \mathbf{V}_{\xi^3} \begin{bmatrix} \xi^1 \\ \xi^2 \end{bmatrix} \\ \mathbf{0} \qquad 1 \end{bmatrix}, \log(\chi) = \begin{bmatrix} \theta \\ \mathbf{V}_\theta^{-1} x \end{bmatrix}, \tag{2.11}$$

$$\text{where } \mathbf{V}_\alpha = \frac{1}{\alpha}\begin{bmatrix} \sin(\alpha) & -1 + \cos(\alpha) \\ 1 - \cos(\alpha) & \sin(\alpha) \end{bmatrix} \tag{2.12}$$

$$Ad_\chi = \begin{bmatrix} R(\theta) & \mathbf{0} \\ -Jx & 1 \end{bmatrix} \tag{2.13}$$

2.2.4 The group of 3D direct spatial isometries $SE(3)$

As in the planar case, $SO(3)$ is extended as the group of 3D direct spatial isometries, $SE(3)$, which is used to represent the spatial poses, through a matrix embedding

$$SE(3) := \{T = \begin{pmatrix} R & x \\ 0_{1,3} & 1 \end{pmatrix} \in \mathbb{R}^{4\times 4} \mid (R,x) \in SO(3)\times\mathbb{R}^3\}.$$

We thus get the following standard pose composition operation $(R_1,X_1)\cdot(R_2,X_2) = (R_1R_2, R_1X_2+X_1)$.

The tangent space around identity is $\mathfrak{se}(3) = \left\{ \begin{pmatrix} (\omega)_\times & u \\ 0_{1,3} & 0 \end{pmatrix}, \omega, u \in \mathbb{R}^3 \right\}$. An isomorphism between $\mathbb{R}^6$ and $\mathfrak{se}(3)$ is given by $\begin{pmatrix} \omega \\ u \end{pmatrix}^\wedge = \begin{pmatrix} (\omega)_\times & u \\ 0_{1,3} & 0 \end{pmatrix}$. The exponential mapping is given by the formula: $\exp\begin{pmatrix} \omega \\ u \end{pmatrix} = I_4 + S + \frac{1-\cos(||\omega||)}{||\omega||^2}S^2 + \frac{||\omega||-\sin(||\omega||)}{||\omega||^3}S^3$, where $S = \begin{pmatrix} \omega \\ u \end{pmatrix}^\wedge$.

Let $\theta = \sqrt{\omega^T\omega}$, $A = \frac{\sin(\theta)}{\theta}$, $B = \frac{1-\cos(\theta)}{\theta^2}$, $C = \frac{1-A}{\theta^2}$. Then we can rewrite

$$\exp\begin{pmatrix} \omega \\ u \end{pmatrix} = \begin{pmatrix} I_3 + A(\omega)_\times + B(\omega)_\times^2 & (I_3 + B(\omega)_\times + C(\omega)_\times^2)u \\ 0_{1,3} & 1 \end{pmatrix} = \begin{pmatrix} \exp_{SO(3)}(\omega) & (I_3 + B(\omega)_\times + C(\omega)_\times^2)u \\ 0_{1,3} & 1 \end{pmatrix} \tag{2.14}$$

In intrinsic form, we thus get

$$\exp(\omega,u) = (\exp_{SO(3)}(\omega), (I_3 + B(\omega)_\times + C(\omega)_\times^2)u) \tag{2.15}$$

This allows to write the logarithm in closed-form as

$$\log(R,x) = (\log_{SO(3)}(R), (I_3 - \frac{1}{2}(\omega)_\times + \frac{1}{\theta^2}(1-\frac{A}{2B})(\omega_\times)^2)(x))$$

with $\omega = \log_{SO(3)}(R)$.

Finally, $Ad_{R,x}$ is given by:

$$Ad_{R,x} = \begin{pmatrix} R & 0_{3,3} \\ (x)_\times R & R \end{pmatrix}$$

2.2.5 The group of double direct spatial isometries $SE_2(3)$

$SE(3)$ has a natural extension, which was introduced to our best knowledge in this form in [14], and is a subgroup of the Galilean group. It is named "double direct spatial isometries" and used to describe *extended poses*, i.e. position X, velocity V and orientation R

$$SE_2(3) := \{T = \begin{pmatrix} R & v & x \\ 0_{1,3} & 1 & 0 \\ 0_{1,3} & 0 & 1 \end{pmatrix} \mid (R,v,x) \in SO(3)\times\mathbb{R}^3\},$$

The group operation derived from this matrix representation can be rewritten in intrinsic form $(R_1,V_1,X_1)\cdot(R_2,V_2,X_2) = (R_1R_2, R_1V_2+V_1, R_1X_2+X_1)$.

The derivations made for $SE(3)$ are seamlessly extended to $SE_2(3)$, as the position and velocity parts play equivalent roles. Therefore, we have

$$\exp(\omega,v,u) = (\exp_{SO(3)}(\omega), (I_3 + B(\omega)_\times + C(\omega)_\times^2)v, (I_3 + B(\omega)_\times + C(\omega)_\times^2)u) \tag{2.16}$$

The logarithm is similarly obtained This allows to write the logarithm in closed-form as

$$\log(R,v,x) = (\log_{SO(3)}(R), N(\omega)(x), N(\omega)(v))$$

with $\omega = \log_{SO(3)}(R)$, $\theta = \sqrt{\omega^T\omega}$, and $N(\omega) = (I_3 - \frac{1}{2}(\omega)_\times + \frac{1}{\theta^2}(1-\frac{A}{2B})(\omega_\times)^2)$.

Finally, $Ad_{R,x}$ is given by:

$$Ad_{R,x} = \begin{pmatrix} R & 0 & 0 \\ (v)_\times R & R & 0 \\ (x)_\times R & 0 & R \end{pmatrix}$$

Remark 1. *Here the extended poses were defined to take into account the mobile's velocity. However, other quantities can be represented this way, such as features [11,29].*

2.3 Group-affine dynamics and group actions lead to autonomous errors

This section summarises the main results of [14,18]. We first present the notion of group-affine dynamics which is at the very heart of the invariant framework. It was introduced in [14] and further explored in [8,18] with continuous dynamics. Here we consider the simpler yet as powerful case of discrete dynamics, for which a comprehensive introduction is given in [16], and transpose the existing results. Then, the advantage of considering group actions as observations is outlined. In the following, we consider a dynamical discrete system on a Lie group G:

$$\chi_{i+1} = f_i(\chi_i), \tag{2.17}$$

Definition 3. *Consider two different trajectories $(\chi_n)_{n\geq 0}$ and $(\upsilon_n)_{n\geq 0}$ of (2.17). The left- and right-invariant errors are defined as*

$$\eta_n^L = \upsilon_n^{-1}\chi_n \qquad \eta_n^R = \chi_n^{-1}\upsilon_n \tag{2.18}$$

These group elements are natural ways to measure the discrepancy between those trajectories on the group G They get their name from the fact that they are invariant to left (resp. right) multiplications $(\chi,\upsilon) \mapsto (\Gamma\chi,\Gamma\upsilon)$ (resp. $(\chi,\upsilon) \mapsto (\chi\Gamma,\upsilon\Gamma)$) by some group element Γ.

Just as the linear Kalman filter is best suited to linear systems, due to the autonomous propagation of the associated error, these non-linear error variables are particularly well adapted to a certain, but broad, class of dynamical models. Originally only dynamics that were invariant to the group action were considered - hence the name, the IEKF being rooted in [22,23] - and the class of dynamics to be considered has recently been much extended, they are called group-affine dynamics [14]. They are to Lie groups what affine maps are to vector spaces, and, although restrictive, this class contains many cases of interest for navigation, especially in attitude and pose estimation, and inertial navigation with known biases, as will be explained in Section 2.4.

Definition 4. *A discrete dynamics f defined on a group G is called group-affine if it satisfies*

$$\forall a,b \ f(ab) = f(a)f(Id)^{-1}f(b), \tag{2.19}$$

where Id is the identity matrix, i.e. the neutral element of G.

Note that $f(Id)^{-1}$ is the inverse of $f(Id)$ in G, and not the reciprocal image of Id, $f^{-1}(Id)$, as will be considered later.

The use of invariant errors and the notion of group-affine dynamics is advocated by two properties: the *autonomous propagation* and the *log-linearity* of the error.

Autonomous propagation of the error

Theorem 1. *For the system (2.17), the left- and right-invariant errors, both denoted by η here, follow autonomous dynamics if and only if the dynamics f_i is group-affine for each i.*

In that case, both errors follow dynamics of the form

$$\eta_{i+1} = g_i(\eta_i) \tag{2.20}$$

$$\text{For the left-invariant error, } g_i(\eta^L) = f_i(Id)^{-1}f_i(\eta^L) \tag{2.21}$$

$$\text{For the right-invariant error, } g_i(\eta^R) = f_i(\eta^L)f_i(Id)^{-1} \tag{2.22}$$

and in both cases g_i is a group automorphism.

Proof. If the dynamics f_i are all group-affine, then it is easy to check that (2.20)-(2.22) are indeed satisfied.

The converse is based on the fact that the shifts $a \mapsto ab$, for any $b \in G$ are bijections. Suppose that, for any pair of trajectories $(\chi_j)_j$ and $(\bar{\chi}_j)_j$, their associated left-invariant error $\eta_j = \chi_j^{-1}\bar{\chi}_j$ (the right-invariant case is similar), satisfies $\eta_{i+1} = g_i(\eta_i)$. Then, replacing η_{i+1} by its definition, we get:

$$g_i(\eta_i) = f_i(\chi_i)^{-1}f_i(\bar{\chi}_i) \tag{2.23}$$

Evaluating in a case where $\chi_i = Id$, implying that $\eta_i = \bar{\chi}_i$, brings:

$$g_i(\eta_i) = f_i(Id)^{-1} f_i(\eta_i) \tag{2.24}$$

Reinjecting in (2.23), isolating $f_i(\bar{\chi}_i)$, and using that $\bar{\chi}_i = \chi_i \eta_i$ in general, this finally leads to

$$f_i(\chi_i \eta_i) = f_i(\chi_i) f_i(Id)^{-1} f_i(\eta_i) \tag{2.25}$$

The converse is straightforward. □

In particular, the fact that g_i is an automorphism leads the following characterisation of the group-affine dynamics.

Corollary 1. *from [18] A dynamics f is group-affine iff there exist a group automorphism Θ and a group element κ such that*

$$\forall a \in G,\ f(a) = \Theta(a) \cdot \kappa \tag{2.26}$$

Remark 2. *If the group is a vector space equipped with the addition, this condition simply means that f must be affine. Indeed,* (2.19) *rewrites $f(a+b) = f(a) + f(b) - f(0)$. In turn, the characterisation* (1) *directly gives $f(a) = Ma + d$, with M an invertible matrix, and d a vector.*

Log-linear propagation the error Following (2.18), we are interested in errors which write

$$\eta_i = \hat{\chi}_i^{-1} \chi_i = \exp(\xi_i) \tag{2.27}$$

Remark 3. *The opposite error, $\eta_i = \chi_i^{-1} \hat{\chi}_i$ could also be considered. In this case, the true state would be recovered as $\chi_i = \hat{\chi}_i \exp(-\xi_i)$. This would simply change* (2.33)*, where η would get updated on the right, and the signs of measurements Jacobians computed throughout this thesis, but it would lead to strictly equivalent algorithms and the same estimates.*

For this case, the group-affine systems exhibit a second strong property: ξ_i follows a linear propagation. Although the proof in the continuous case is tedious [14], in the discrete case it is a direct consequence of the correspondance between Lie groups and Lie algebras given by:

Proposition 2. *[55, Proposition 16.4] for every Lie group homomorphism $g : G \to H$, we have $g \circ \exp = \exp \circ dg_{Id}$, where dg_{Id} is the tangent map of g at the neutral element of G.*

Since $dg_{i_{Id}}$ is linear, it is associated to a matrix F. Added to the assumption that the exponential can be considered a bijection, we get

Theorem 2. *Let $\eta_i = \exp(\xi_i)$ be either the left- or right-invariant error between two trajectories following group-affine dynamics f_i. Suppose that η_i always lies close enough to the identity, so that* exp *is bijective. Then we have*

$$\exp(\xi_{i+1}) = \exp(F_i \xi_i) \tag{2.28}$$

where F_i is independent *of the current estimate. This also implies that*

$$\xi_{i+1} = F_i \xi_i \tag{2.29}$$

This is illustrated by the following commutative diagram

$$\begin{tikzcd} \mathfrak{g} \arrow[r, "\exp"] \arrow[d, "g_i"'] & G \arrow[d, "F_i"] \\ \mathfrak{g} \arrow[r, "\exp"'] & G \end{tikzcd}$$

In practice, this will help computing the propagation of an updated state $\chi \exp(\xi)$ or $\exp(\xi)\chi$ from the propagation of χ.

Corollary 2. *Let f be group-affine, g^L, g^R the associated dynamics for the left- and right-invariant errors respectively, and F^L, F^R the respective matrices of Theorem 2.*

$$\forall \chi, \xi,\ f(\chi \exp(\xi)) = f(\chi) \exp(F^L \xi), \qquad f(\exp(\xi)\chi) = \exp(F^R \xi) f(\chi) \tag{2.30}$$

In the following, F^L and F^R are called the jacobians of the dynamics f.

Noise-free autonomous update with group actions The interest of group actions is that, in the noise-free case, they yield innovations which are also independent of the estimate. In turn, the error is both propagated and updated autonomously, leading to the stability properties shown in [14]. Indeed, suppose the observation is a left group action $\star : G \times Y \to Y$ for some set Y. Only this case is detailed, the right ones being extremely similar. The observation writes

$$z = h(\chi) = \chi \star b.$$

The innovation reflects the divergence between the measured observation and the expected one. In the linear case it writes $i_k = z_k - H\hat{X}_k$. Although intuitive, this choice is powerful as the innovation only depends on the error: $i_k = H(X_k - \hat{X}_k)$. To recover this property for a general group action and the non-linear error $\eta_k = \hat{\chi}_k^{-1}\chi_k$, the innovation is computed, in a noise-free setting, as

$$i_k = \hat{\chi}^{-1} \star z_k = \hat{\chi}^{-1} \star (\chi \star b) = \eta \star b \tag{2.31}$$

Let L_k be an update function, leading to a corrected estimate

$$\hat{\chi}^+ = \hat{\chi} L_k(i_k) \tag{2.32}$$

In turn, just as in the linear case, the error is updated autonomously. This will also allow the associated Jacobian to be state independent.

$$\eta^+ = L_k(\eta \star b)^{-1}\eta \tag{2.33}$$

Linear observed systems on group

These results bring the definition of linear observed systems on groups, which will benefit from autonomous errors.

Definition 5. *A system with state $\chi_i \in G$, where G is a group, is a linear observed system if*

- *Its dynamics are group-affine, i.e., satisfies* (2.19)*, or equivalently is of the form* (2.26)
- *It is observed through either left or right group actions, i.e., satisfying either* (2.3) *or* (2.4)*.*

For left (resp. right) group actions, the systems are called left (resp. right) linear observed systems on group.

2.4 Examples of exactly and imperfect group-affine dynamics in navigation

As it turns out, many dynamics of interest in navigation are group-affine. This is true in particular for attitude estimation, 2D and 3D pose estimation, which are based on left-invariant dynamics, and unbiased inertial navigation, which is neither left nor right-invariant.

2.4.1 Attitude estimation

Attitude estimation is the problem of tracking the 3D orientation of a mobile system. It is represented at each step i by a rotation matrix $R_i \in SO(3)$, which maps the coordinates of a vector expressed in the body frame to those in the static frame. Its evolution, in discrete form, is given by

$$R_{i+1} = f_i(R_i) = R_i\Omega_i, \tag{2.34}$$

$\Omega_i = \exp(dt\omega_i)$, ω_i being the mobile's angular velocity, supposed constant during the sampling period dt.

This dynamics is said to be left-invariant (if $\tilde{R}_0 = \Gamma R_0$, then $\tilde{R}_i = \Gamma R_i$), and is obviously of the form (2.26), i.e., group-affine.

The associated jacobian, appearing in (2.29), is computed for the left- and right-invariant error through

$$\begin{aligned} f_i(R_i\exp(\xi_i)) &= R_i\exp(\xi_i)\Omega_i & f_i(\exp(\xi_i)R_i) &= \exp(\xi_i)R_i\Omega_i && (2.35)\\ &= f_i(R_i)\exp(Ad_{\Omega_i^{-1}}\xi_i) & &= \exp(\xi_i)f_i(R_i), && (2.36)\end{aligned}$$

i.e., we have with the notations of Corollay 2

$$F_i^L = Ad_{\Omega_i^{-1}} \qquad F_i^R = Id. \tag{2.37}$$

Remark 4. *In this case, $\Omega_i^{-1} = \Omega_i^T$, facilitating the computations. However, since this derivation holds for other systems, the inverse is kept in* (2.37) *so that it can be generalised.*

2.4.2 Pose estimation

Pose estimation consists in tracking the orientation and position of a mobile system, either in a planar setting, or in three dimensions. Both cases have similar Lie groups embedding, as is described thereafter.

2D case

In two dimensions, the pose of a system is given by its heading θ and its position x. In general, the dynamics are given by:

$$\theta_{i+1} = \theta_i + dt\omega_i, \qquad x_{i+1} = x_i + dtR(\theta_i)u_i, \tag{2.38}$$

where $R(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$ is the 2D rotation matrix associated to α, and ω_i, u_i are the angular and linear velocities.

This system can be seen as living in the Lie group $SE(2)$ with a state χ_i verifying

$$\chi_i = \begin{pmatrix} R(\theta_i) & x_i \\ 0 & 1 \end{pmatrix}, \quad \chi_{i+1} = \chi_i U_i, \quad U_i = \begin{pmatrix} R(dt\omega_i) & dt\ u_i \\ 0 & 1 \end{pmatrix} \tag{2.39}$$

3D case

In three dimensions, the setting is similar, the main difference being that the heading is replaced by the 3D attitude R_i. The dynamics are then given by:

$$R_{i+1} = R_i\Omega_i, \qquad x_{i+1} = x_i + dt\ R(\theta_i)u_i, \tag{2.40}$$

where $\Omega_i = \exp_{SO(3)}(dt\omega_i)$.

This system can be seen as living in the Lie group $SE(3)$, with an embedding extremely similar to (2.39) in the 2D case, with a state χ_i verifying

$$\chi_i = \begin{pmatrix} R_i & x_i \\ 0 & 1 \end{pmatrix}, \quad \chi_{i+1} = \chi_i U_i, \quad U_i = \begin{pmatrix} \Omega_i & dt\ u_i \\ 0 & 1 \end{pmatrix} \tag{2.41}$$

Jacobians of the dynamics

Both cases, once embedded in a Lie group, share the same dynamics structure $\chi_{i+1} = \chi_i U_i$, which has the same form as (2.34) for attitude estimation. Therefore, the jacobians of the dynamics are given by

$$F_i^L = Ad_{U_i^{-1}} \qquad F_i^R = Id. \tag{2.42}$$

2.4.3 Unbiased inertial navigation

Inertial navigation tracks the attitude, the velocity and the position of a vehicle evolving in the 3D space, given by R_i, v_i and x_i respectively, thanks to the measurements of accelerometers and gyroscopes it is endowed with. These sensors have in practice inner biases, i.e. are polluted with noise of non-zero mean. Here, they are considered to be zero, or already known and corrected in advance, hence the name unbiased. As will be detailed, the dynamics of unbiased inertial navigation are neither left- nor right-invariant, as were those of the other examples. Nevertheless, they still fit into the group-affine framework, which was one of the main findings of [14]. A new and simpler proof, based on new theoretical insights, was given in [18]. This is the one presented here.

The mobile is endowed with accelerometers and gyroscopes whose measures are denoted respectively by u_i and ω_i. The dynamics then reads:

$$R_{i+1} = R_i\Omega_i, \quad v_{i+1} = v_i + dt(g + R_i u_i), \quad x_{i+1} = x_i + dt\ v_i, \tag{2.43}$$

where g denotes the gravitational field.

The navigation triplet (R, v, x) is embedded in the Lie group $SE_2(3)$, through

$$\chi_i = \begin{pmatrix} R_i & v_i & x_i \\ 0 & & I_2 \end{pmatrix} \tag{2.44}$$

Let f_i denote the dynamics, such that $\chi_{i+1} = f_i(\chi_i)$. The main result of [18] is that, in this noise and bias free case, (2.43) can be rewritten in compact and intrinsic form through group multiplication as

$$f_i(\chi_i) = \gamma\phi(\chi_i)\upsilon_i, \tag{2.45}$$

where $\gamma = (Id, dt\ g, 0)$, $\upsilon_i = (\exp(dt(\omega_i)), dt\ a_i)$, and $\phi : (R, v, x) \mapsto (R, v, x + dt\ v)$. That is, this mechanisation is a sequence of three steps, one internal change of position by the velocity, one local update from the increments, and one global update from the gravity. Thanks to this expression, the following result can be stated:

Theorem 3. *The map ϕ of* (2.45) *is a group automorphism, and in turn a system governed by* (2.45) *has group-affine dynamics.*

Proof. Let $\chi_1 = (R_1, v_1, x_1)$ and $\chi_2 = (R_2, v_2, x_2)$ be two group elements. Then we have

$$\phi(\chi_1\chi_2) = (R_1R_2, v_1 + R_1v_2, x_1 + R_1v_2 + dt\ (v_1 + R_1v_2)) = \phi(\chi_1)\phi(\chi_2)$$

Moreover, it is clearly invertible, of inverse map $(R, v, x) \mapsto (R, v, x - dt\ v)$. Therefore, ϕ is a group automorphism.

To complete the proof, (2.45) must be cast in the form of (2.26). This is done thanks to the Ad operator:

$$\gamma F(\chi_i)\upsilon_i = Ad_\gamma \circ F(\chi_i)\ (\gamma\ \upsilon_i)$$

Since Ad is also a group automorphism, Corollary 1 brings the desired result. □

The jacobians will be given in Section 4.4, along with a more in depth study of (2.45).

Remark 5. *For navigation applications which need to localise in a geocentric frame, and not just with a flat earth model, the Earth rotation must be taken into account. Being a change of the global frame, it intervenes along the gravity. This once again elegantly fits the described theory, as the gravitation term γ simply becomes $\gamma = (\Omega_e^T, dt\ g, 0)$, where Ω_e represents the rotation of the Earth during the time step dt. See the gyro-compassing example of [15] for instance, for more details.*

2.4.4 Adding noise and the biases

In practice, the inertial increments are polluted by biased noise, with potentially time-varying biases. Therefore, these biases need to be estimated online, and the state is extended into the 5-tuple (R, v, x, b^g, b^a), where $b^g \in \mathbb{R}^3$ is the gyroscopes bias, and $b^a \in \mathbb{R}^3$ that of the accelerometers. To account for the possible evolution of the biases, they are considered constant up to some noise. Moreover, considering noisy gyrometers and accelerometers measurements, the dynamics become

$$f(R, v, p, b^g, b^a) = \begin{cases} R\exp_{SO(3)}(dt(\omega - b^g + w_g)) \\ v + dt(R(a - b^a + w_a) + g) \\ p + dt\ v \\ b^g + w_{b^g} \\ b^a + w_{b^a}, \end{cases} \tag{2.46}$$

where $w_g, w_a, w_{b^g}, w_{b^a}$ are white noises on $\mathbb{R}^3$. However, this dynamics in only *group-affine for fixed biases*. That is, in general, the good properties of the Invariant framework may be lost. Still, filters can be defined, considering the state space $SE_2(3) \times \mathbb{R}^3 \times \mathbb{R}^3$, and which proved far superior to standard EKFs [16]. In this case, the left-invariant error writes

$$\eta^L = \begin{pmatrix} (R, v, x)^{-1}(\hat{R}, \hat{v}, \hat{x}) \\ \hat{b}^g - b^g \\ \hat{b}^a - b^a \end{pmatrix} \tag{2.47}$$

The right-invariant error is defined similarly

$$\eta^R = \begin{pmatrix} (\hat{R}, \hat{v}, \hat{x})(R, v, x)^{-1} \\ \hat{b}^g - b^g \\ \hat{b}^a - b^a \end{pmatrix} \tag{2.48}$$

Such mixed errors have been successfully used in filters for inertial navigation [30], target tracking [90], and in smoothing for attitude tracking [102].

2.5 Conclusion

This chapter presented the mathematical framework of invariant filtering, based on Lie groups, and the notions of group-affine dynamics and group actions. These concepts allowed defining non-linear errors between two trajectories of a broad class of systems, whose evolution does not depend on the states' actual values. Their propagation only depends on the dynamics of the system, and their update, in the noise free case, only on the observation model. These errors therefore evolve autonomously. Attitude and pose estimation, as well as unbiased inertial navigation, were shown to fit in the group-affine framework.

Chapter 3

Invariant Filtering and geometric constraints

Highlights

- This chapter is dedicated to the introduction of the IEKF, a reminder of its properties and complementary results that were obtained in the field of invariant Kalman filtering.
- It is proved that if the dynamics are deterministic (and the measurements noisy), then the state is bound to live in a subspace of the state space. If the dynamics are group-affine, then the IEKF guarantees the estimate also lives in this subspace. Hence IEKF estimates are consistent with the information one has about the system. By contrast, when using a standard EKF the estimate may very well step out of this subspace.
- Although there is always process noise in practice, the case of deterministic dynamics is worth studying as a limit case of high precision inertial navigation, where the amount of noise in the IMU equations is actually very low.

Points marquants

- Ce chapitre commence par une introduction de l'IEKF et un rappel de ses propriétés, puis présente de nouveaux résultats complémentaires obtenus sur le filtrage de Kalman invariant.
- Il est prouvé que si la dynamique est déterministe (et les mesures bruitées), alors l'état est tenu d'évoluer dans un sous-espace de l'espace d'état. Si la dynamique est groupe-affine, alors l'IEKF garantit que l'estimé évoluera lui-aussi dans ce sous-espace. Ainsi, les estimés fournis par l'IEKF sont cohérents avec l'information que l'on a sur le système. Par contraste, l'estimé d'un EKF standard peut sortir de ce sous-espace.
- Bien qu'il y ait en pratique du bruit de processus, le cas d'une dynamique déterministe mérite d'être étudié comme cas limite de la navigation inertielle de haute précision, où le bruit entrant dans les équations de l'IMU sont dans les faits très faibles.

Introduction (En/Fr)

The Kalman filter (KF) and its extended version (EKF) have appeared in the 1960s, and played a big role in the guidance of spacecraft during the space age. It has been the state of the art for industrial applications since the 1960s, notably for navigation. However, due to the non-linear nature of the navigation equations, and in particular to the fact the orientation of the aircraft (i.e., the attitude) does not live in a vector space, the EKF may have some shortcomings.

The Invariant Extended Kalman Filter (IEKF) is a relatively recent variant of the EKF meant to account for the non-linearities of the state space when devising EKFs on Lie groups.

First, the KF and its optimality properties are introduced, followed by the EKF which loses these features. Then the equations and theoretical results of the IEKF are given following [14,16]. The interest of Invariant Filtering is illustrated on a particular case of filtering with constraints, where it is shown that it once again regains and generalises the nice properties of the linear KF, in that it naturally satisfies a

broad range of geometric constraints without having to artificially enforce them.

Le filtre de Kalman (KF) and sa version étendue (EKF) sont apparues dans les années 60 et ont joué un grand rôle dans le guidage des engins spatiaux de l'ère spatiale. Il est longtemps resté l'état de l'art, notamment pour la navigation. Cependant, la non-linéarité des équations de navigation, venant en particulier du fait que l'orientation du mobile ne vit pas dans un espace vectoriel, rendent l'EKF parfois insuffisant.

Le filtre de Kalman étendu invariant (IEKF) est une variante récente de l'EKF faite pour prendre en compte les non-linéarités de l'espace d'état lors de la conception d'EKF sur groupe de Lie.

Le KF et ses propriétés d'optimalité sont introduits, suivis par l'EKF qui perd ces caractéristiques. Puis les équations et résultats théoriques de l'IEKF sont introduits suivant [14,16]. L'intérêt du Filtrage Invariant est illustré sur un cas particulier de filtrage sous constraintes, pour lequel il est montré qu'il permet à nouveau de regagner et généraliser les bonnes propriétés du KF linéaire. En effet, il satisfait un large ensemble de contraintes sans sans avoir à les forcer explicitement.

3.1 The Linear Kalman Filter and the Extended Kalman Filter

This section recalls the definitions of the linear Kalman filter [74], and its standard extension to non-linear systems.

3.1.1 The linear Kalman filter and its optimality properties

Consider a classical discrete linear system in $\mathbb{R}^p$:

$$X_{i+1} = F_i X_i + u_i + w_i, \tag{3.1}$$

$$Z_k = H_k X_k + V_k, \tag{3.2}$$

where $X_i \in \mathbb{R}^p$ is the state of the system at time i, u_i a given vector, $(z_k)_{k\geq 0}$ the corresponding observations and $w_i \sim \mathcal{N}(0, Q_i)$, $V_k \sim \mathcal{N}(0, N_k)$ are independent Gaussian noises polluting the dynamics at step i and the observation k. F_i and H_k are the matrices defining the dynamics of the system and the function of the system observed through Z_k, respectively.

Let $X_0 \sim \mathcal{N}(\bar{X}, P_0)$ be a Gaussian prior on the state at time 0. Then the distribution of X_i, given the propagation matrices F_i and the observations Z_k stays Gaussian, thanks to the following two standard properties:

- If $X \sim \mathcal{N}(\bar{X}, P)$, $w \sim \mathcal{N}(\bar{w}, Q)$, and F, u are given matrix and vector, then $FX + u + w$ is a Gaussian, of mean $F\bar{X} + u + \bar{w}$ and covariance $FPF^T + Q$
- If $X \sim \mathcal{N}(\bar{X}, P)$, $V \sim \mathcal{N}(\bar{0}, N)$, and H is a given matrix, then $(X, HX + V)$ forms a Gaussian vector. In turn, $\mathbb{E}[X|HX + V]$ is also a Gaussian, whose mean and covariance are given by the conditioning formulas, which lead to the Kalman filter equations, defined thereafter

Given a Gaussian prior, the Kalman filter defines an estimate $\hat{X}_i$ at each step i through the following initialisation and two step sequence:

$$\hat{X}_{0|0} \sim \mathcal{N}(\bar{X}, P_0) \qquad \text{Initialisation} \tag{3.3}$$

$$\begin{cases} \hat{X}_{i+1|i} &= F_i \hat{X}_{i|i} + u_i \\ P_{i+1|i} &= F_i P_i F_i^T + Q_i \end{cases} \qquad \text{Propagation} \tag{3.4}$$

$$\begin{cases} S_{i+1} = H_{i+1} P_{i+1|i} H_{i+1}^T + N_{i+1} \\ K_{i+1} = P_{i+1|i} H_{i+1}^T S_{i+1}^{-1} \\ \hat{X}_{i+1|i+1} = \hat{X}_{i+1|i} + K_{i+1}(Z_{i+1} - H_{i+1}\hat{X}_{i+1|i}) \\ P_{i+1|i+1} = (I - K_{i+1} H_{i+1}) P_{i+1|i} \end{cases} \qquad \text{Update} \tag{3.5}$$

The linear Kalman filter is said to be optimal thanks to the following result.

Theorem 4. *For all i, $(\hat{X}_{i|i}, P_{i|i})$ and $(\hat{X}_{i+1|i}, P_{i+1|i})$ are the mean and covariance of $\mathbb{P}(X|\bar{X}, u_{0:i-1}, z_{1:i})$ and $\mathbb{P}(X|\bar{X}, u_{0:i}, z_{1:i})$ respectively, where $u_{0:i-1} = u_0 \ldots, u_{i-1}$, and likewise for $u_{0:i}, z_{1:i}$. Moreover, these distributions are gaussians.*

The linear Kalman filter exhibits several optimality properties, it is in particular an unbiased estimator with minimum variance. These properties mostly come from two of its features: the particular Kalman gain K_i used, and the fact that its error is autonomous: it does not depend on the measurements u_i nor on the observations Z_i. Indeed, for two independent instances of this system X^1 and X^2, let the error be

$$e_i = X_i^1 - X_i^2$$

Then it satisfies

$$\begin{aligned} e_{i+1|i} &= F_i e_i \\ e_{i|i} &= e_{i+1|i} - K_{i+1} H_{i+1} e_{i+1|i} \end{aligned}$$

Therefore, the error evolves solely according to the noises and their statistical properties, which are supposed to be known in this context. In particular, this also holds for its associated covariance P_i. This autonomy property is crucial, as will be pointed out in invariant filtering in Section 3.2.

3.1.2 The Extended Kalman Filter

The original Kalman filter was designed for linear systems with linear observations. To handle non-linear dynamics or observations, an extension of the algorithm was proposed, called the Extended Kalman Filter (EKF). Consider the following non-linear system

$$X_{i+1} = f_i(X_i, u_i) + w_i, \tag{3.6}$$

$$Z_k = h_k(X_k) + V_k, \tag{3.7}$$

Then the EKF follows similar steps as the linear version, up to the following differences:

- The estimate is propagated as $\hat{X}_{i+1|i} = f_i(\hat{X}_{i|i}, u_i)$
- The update writes $\hat{X}_{i+1|i+1} = \hat{X}_{i+1|i} + K_{i+1}(Z_{i+1} - h_{i+1}(\hat{X}_{i+1|i}))$
- F_i and H_{i+1} in (3.4), (3.5) are replaced by $\hat{F}_i$ and $\hat{H}_{i+1}$, the jacobians of f_i and h_{i+1} computed at the estimates $\hat{X}_{i|i}$ and $\hat{X}_{i+1|i}$ respectively

This last point is crucial, as it means that the estimated covariance depends on the current estimate. That, in addition to the fact that the EKF outputs only an approximation of $\mathbb{P}(X_i|\bar{X}, Z_1, \ldots, Z_i)$ which is not a Gaussian anymore, is well-known to make the EKF lose the optimality properties of the linear case. In particular, the error e_i defined in Section 3.1.1 is not autonomous anymore.

3.2 The Invariant Extended Kalman Filter

The Invariant Extended Kalman Filter (IEKF) is a Kalman like non-linear observer devised to estimate systems on Lie groups: its goal is to track the mean and covariance of a $\mathcal{N}_L$ or $\mathcal{N}_R$ distribution, defined in Section 2.1.2. It was not designed to be applied to all possible systems: just as the KF is adapted to linear systems, the IEKF is well-suited for the so called linear observed systems on groups, which were formalised in [18] and recalled in Definition 5. These systems rely on the group-affine dynamics and group actions-based observations defined in Section 2.3. Generally speaking, the IEKF is close to the LG-EKF [25,26], however these did not leverage its links with group affine dynamics and group actions, and therefore autonomous errors. The IEKF can be seen as an extension of the Multiplicative EKF for more general state spaces [43]. Non-euclidian state spaces have also been used for other Kalman-like filters, such as the Unscented Kalman filter (UKF), exhibiting improvements compared to its standard version [33,52].

3.2.1 Considered noisy systems

As it is sufficient for the remainder of this Chapter, and for the sake of simplicity, we only consider the group actions on $\mathbb{R}^n$ mentioned at the end of Section 2.1.1. Therefore, we are interested in the following noisy dynamical system, where f_i denotes group-affine dynamics and $d \in \mathbb{R}^n$ a given vector.

$$\chi_{i+1} = f_i(\chi_i)\exp(w_i), \qquad w_i \sim \mathcal{N}(0, Q_i) \tag{3.8}$$

$$\begin{cases} z_k = \chi_k d + v_k \\ \quad (OR) \\ z_k = \chi_k^{-1} d + v_k \end{cases} \qquad v_k \sim \mathcal{N}(0, N_k) \tag{3.9}$$

The former and latter observation models are left- and right-equivariant respectively, each associated to a variant of the IEKF: the Left- and Right-IEKF, denoted L-IEKF and R-IEKF.

3.2.2 Invariant filtering equations, stability and consistency properties

The L-IEKF The L-IEKF estimates $\chi \sim \mathcal{N}_L(\hat{\chi}, P)$ through the standard propagation-update sequence of Kalman filters, with a modified innovation. Indeed, following (2.31), the action of the inverse of the estimate on the measurement is taken in order to make the error appear, and top equation of (3.9) rewrites

$$\hat{\chi}_k^{-1} z_k = \eta_k^{-1} d + \hat{\chi}_k^{-1} v_k. \tag{3.10}$$

Linearising Equation (3.10) through the first order approximation $\exp(\xi) \approx \mathbb{I} + \xi\hat{}$ allows defining a Kalman gain as in the conventional EKF theory, leading to the following equations.

$$\begin{cases} \hat{\chi}_{i+1|i} &= f_i(\hat{\chi}_{i|i}) \\ P_{i+1|i} &= F_i P_i F_i^T + Q_i \end{cases} \qquad \text{Propagation} \tag{3.11}$$

$$\begin{cases} S_{i+1} = H_{i+1} P_{i+1|i} H_{i+1}^T + \hat{N}_{i+1} \\ K_{i+1} = P_{i+1|i} H_{i+1}^T S_{i+1}^{-1} \\ \hat{\chi}_{i+1|i+1} = \hat{\chi}_{i+1|i} \exp\left[K_{i+1}(\hat{\chi}_{i+1|i}^{-1} z_{i+1} - d)\right] \\ P_{i+1|i+1} = (I - K_{i+1} H_{i+1}) P_{i+1|i} \end{cases} \qquad \text{Update} \tag{3.12}$$

where H_{i+1} is the matrix defined by $H_{i+1}\xi = \xi\hat{}d$ and $\hat{N}_{i+1}$ denotes the covariance matrix of the observation noise $\hat{v}_{i+1} = \hat{\chi}_{i+1}^{-1} v_{i+1}$. The minus signs comes from the considered error: $\eta = \chi^{-1}\hat{\chi}$, meaning that $\chi = \hat{\chi}\eta^{-1}$.

The R-IEKF The R-IEKF estimates $\chi \sim \mathcal{N}_R(\hat{\chi}, P)$. The R-IEKF differs from the L-IEKF only in the definition of H_{i+1}, the noise considered noise covariances, and the update:

$$Q_i \leftarrow Ad_{f(\hat{\chi}_{i|i})} Q_i Ad_{f(\hat{\chi}_{i|i})}^T \tag{3.13}$$

$$\hat{\chi}_{i+1|i+1} = \exp\left[K_{i+1}(\hat{\chi}_{i+1|i} z_{i+1} - d)\right] \hat{\chi}_{i+1|i} \tag{3.14}$$

$$H_{i+1}\xi = -\xi\hat{}d \tag{3.15}$$

$$\hat{N}_{i+1} \text{ denotes the covariance of } \hat{\chi}_{i+1|i} v_{i+1} \tag{3.16}$$

Stability and consistency properties of the IEKF

The fact that the IEKF is devised for linear observed systems on groups allows recovering autonomous error evolution, coming from (2.28) and (2.33). In turn, this leads the IEKF to be a locally asymptotically stable observer (i.e., with noise turned off) [14], just as the linear KF [47]. Note that this property is defined only theoretically, without taking into account the Lie group representation which was chosen. In particular, this does not consider shortcomings due to numerical approximations, e.g., the fact the numerical product of two rotation matrices might not be exactly one anymore.

Moreover, the exact log-linearity of the error dynamics allows the estimated covariance to be consistently propagated. A particularly important result following this property is the fact that it solves the EKF-SLAM inconsistency issues, as was shown in [11,29].

3.3 Kalman filtering and geometric constraints

The results of this section have been first presented at the 56th IEEE Conference on Design and Control (CDC) [36], in continuous form and for a thinner class of systems. They have then been extended in [8] to all group-affine dynamics in continuous time. Herein, the results of [36] and their extension to group-affine dynamics in discrete time are presented, to match the discrete time context retained in this manuscript.

The IEKF has been shown to literally outperform the EKF when the process noise is very small in simulations [14]. This section gives a better understanding of this fact. To be able to derive mathematical result, we need to consider the *limit case* where the motion sensors are ideal, that is, noise free, and when we have as well some prior deterministic information on the state (that is, the initial covariance matrix is rank deficient). Even if the mathematical guarantees obtained below do not strictly apply to the case of highly precise - but not perfect - sensors, they provide a strong indication that the IEKF is particularly suited for this setting, as previously observed [14]. The impact of a rank-deficient initial covariance matrix was first studied only in the very particular case of a non-holonomic car with GPS measurements [13].

Here the motion is assumed noise free (i.e., no process noise) and the measurements noisy. Furthermore, the covariance matrix of the initial state is supposed to be rank deficient. Combined with noise-free dynamics this implies that at all times 1- the covariance matrix is rank deficient and 2- the state can only "reach" a well-characterized submanifold of the state space, that is, the "physical" state space is constrained at all times. Unfortunately, due to the fact it is based on approximations, the EKF (and the other Kalman variants such as the unscented Kalman filter) fail to capture those constraints. On the other hand, the IEKF embraces the Lie group structure of the state space, and is shown to perfectly ensure properties 1 and 2 above. These properties had been shown "manually" on a non-holonomic car example in [13], and then leveraged to derive some global convergence properties of the filter for that case. These results were theorised in [36] and generalised in [8]. Here, a brief and simpler version is detailed, in the discrete case, to keep consistency with the rest of this thesis.

3.3.1 The Linear Kalman Filter may preserve side information

We recall in this section a known property of linear Kalman filtering regarding its ability to handle partially deterministic information if no process noise is added. Although theoretical, this limit case is pivotal to filter robustness as will be illustrated on a simple example in the non-linear case.

Consider the model (3.4), (3.5) of Section 3.1.1, with process noise turned off, i.e., $Q_i = 0$ for all i. Make the additional assumption that the initial distribution of the state lies in an affine subspace V_0 of $\mathbb{R}^p$:

$$\exists a \in \mathbb{R}^p,\ X_0 \in a + V_0. \tag{3.17}$$

It can then be immediately deduced from (3.17) that X_i lives in a known affine subspace at any step i:

$$X_i \in A_i a + V_i, \tag{3.18}$$

where $A_i = \prod_{j \leq i} F_j$, and the vector space is defined as $V_i = A_i V_0 = \{A_i x, x \in V_0\}$.

Remark 6. *In the framework of classical Kalman theory, system* (3.1)-(3.2) *is pathological, inasmuch as the state propagation is deterministic. This limit case is intended to give insight into the way a filter handles initial hard constraints such as* (3.17) *when the process noise is not sufficient to balance the ill-conditioning of the initial covariance matrix. Section 3.5 will show this situation can be extremely troublesome for non-linear systems.*

Initial linear information of the form (3.17) on a linear system is flawlessly captured by a Kalman filter (KF), as illustrated by the following proposition.

Proposition 3. *Let $\hat{X}_0$ and P_0 be respectively the initial estimate and covariance matrix of a Kalman Filter tracking System* (3.1)-(3.2) *with no propagation noise, i.e., $w_i = 0$ and $Q_i = 0$, and assume they are consistent with condition* (3.17) *in the following sense:*

$$\hat{X}_0 \in a + V_0, \qquad \mathrm{Im}(P_0) \subset V_0, \tag{3.19}$$

Then, the state $\hat{X}_i$ and covariance matrix P_i returned at any time by the Kalman filter are consistent with (3.18)*, i.e., we have:*

$$\hat{X}_i \in a + V_i, \quad \mathrm{Im}(P_{i|i}) \subset \mathrm{Im}(P_{i|i-1}) \subset V_i, \tag{3.20}$$

Proof. As it is noise free, the conservation of the property during propagation is clear. Before an update, this implies that $\mathrm{Im}(K_i) \subset \mathrm{Im}(P_{i|i-1}) \subset V_i$. In turn, it follows that $\mathrm{Im}(P_{i|i}) \subset V_i$, and that $x_{i|i}$ remains in the subspace, as the update writes $P_{i|i} = (I - K_i H_i) P_{i|i-1}$ and $\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(Z_i - H_i \hat{x}_{i|i-1})$. □

Conditions (3.19) and (3.20) are easily interpreted: initial error covariance over a direction outside of V_0 (resp. V_i) is zero at time 0 (resp. i). Thus, Proposition 3 implies that at all times the estimate of the Kalman filter remains in the subspace the state lives in (if initialised in V_0), and its covariance keeps reflecting the absence of dispersion of the probability distribution of the state outside of this subspace.

Remark 7. *Note that the result still holds even if the observations are not linear, as the matrix H_i always appears inside the product $K_i H_i$, which thus guarantees the respect of the constraints, whatever the observation model and the innovation are.*

3.3.2 Non-linear case

In the non-linear case, the initial subset in which the state lives may not be a vector space. But even if it is, it is distorted by the dynamics. Therefore, linearisations do not lead to updates that remain in that space. In turn, this leads to degraded performance of the EKF, even in the presence of small process noise, as illustrated in the simulations of [13] and [14].

The aim of this section is to show that although the property above, along with Proposition 3, seems to be reserved to linear systems, it has in fact a counterpart for group-affine dynamics and carries over to the Invariant EKF. This was already proved "manually" for a particular example in [13]. The results of the latter paper will prove to be a particular case of the general theory developed herein.

3.4 Invariant Kalman filtering with geometric constraints

3.4.1 Geometric constraints

A multiplicative counterpart of (3.17) and (3.18) can be derived, based on subgroups. We thus consider the non-linear system with deterministic dynamics, and observations (3.9).

$$\chi_{i+1} = f_i(\chi_i, u_i), \tag{3.21}$$

Following (2.26) from Corollary 1, let κ_i, ϕ_i be such that $f_i(\chi) = \phi_i(\chi)\kappa_i$, and F_i such that $f_i(\chi \exp(\xi)) = f_i(\chi)\exp(F_i \xi)$ according to Theorem 2.

Proposition 4. *Let $\hat{\chi}_0$ and P_0 be respectively the initial estimate and covariance matrix of an IEKF for system* (3.21)-(3.9)*. Suppose that initially the state is known to lie in a particular affine subgroup*

$$\chi_0 \in a \cdot G_0 = \{ax, x \in G_0\}. \tag{3.22}$$

where $a \in G$, and $G_0 \subset G$ is a subgroup such that

$$G_0 = \exp(L_0),$$

for L_0 a Lie subalgebra of $\mathfrak{g}$.

Then at all times we have necessarily

$$\forall i \geq 0, \ \chi_i \in a_i G_i \tag{3.23}$$

where $a_i = f_{i-1} \circ \cdots \circ f_0(a)$ and $G_i = \phi_{i-1} \circ \cdots \circ \phi_0(G_0)$. In particular, we have

$$G_i = \exp(F_{i-1} \cdots F_0 L_0) = \exp(L_i). \tag{3.24}$$

In this discrete case, the proof by induction is straightforward. For all i, G_i is clearly a subgroup of G. Moreover, induction also proves that L_i is Lie subalgebra of $\mathfrak{g}$, thanks to Theorem 2. Indeed, since ϕ_i is a group automorphism, F_i is a Lie algebra morphism, i.e., satisfies $F_i[\xi_1, \xi_2] = [F_i \xi_1, F_i \xi_2]$. Therefore, if L_{i-1} is a Lie subalgebra, then so will L_i.

3.4.2 An example: equivariant constraints

Suppose that initially the state is known to satisfy m equivariant constraints of the form

$$\chi_0 b^p = c^p, \tag{3.25}$$

for some $(b^p, c^p)_{1\leq p\leq m} \subset \mathbb{R}^n$. It can be rewritten in the form of (3.22) as (3.25) is equivalent to

$$\chi_0 \in a \cdot G_0 = \{ax, x \in G_0\},$$

where a is an element of G verifying $a \cdot b^p = c^p$ for all p, and G_0 is the stabilizer subgroup of G with respect to $(b^p)_p$, i.e., $G_0 = \{x, \forall p,\ xb^p = b^p\}$. Proposition 4 leads to the following result.

Corollary 3. *Consider a group-affine dynamics with initial condition* (3.25)*. For all* $1 \leq i \leq k$*, let* b_i^p *and* c_i^p *be defined by*

$$b_0^p = b^p \qquad b_{i+1}^p = \kappa_i^{-1}\phi_i(b_i^p) \tag{3.26}$$

$$c_0^p = c^p \qquad c_{i+1}^p = \phi_i(c_i^p). \tag{3.27}$$

Then at all times we have necessarily

$$\forall i \geq 0,\ \chi_i \in \{\chi \in G \mid \forall i,\ \chi b_i^p = c_i^p\}. \tag{3.28}$$

3.4.3 Illustration in terms of attitude estimation

As announced, let us present now a specific case modeling an attitude estimation problem, introduced in Section 2.4.1. The state is represented by a rotation matrix $R_i \in SO(3)$. Recall that, if ω_i denotes the perfectly measured angular velocity, the dynamics read:

$$R_{i+1} = R_i \Omega_i, \tag{3.29}$$

where $\Omega_i = \exp(dt\omega_i)$. In this case, using the notations of Corollary 1, $\phi_i(\chi) = \chi$, and $\kappa_i = \Omega_i$.

A geometric constraint of the form of (3.25) for this system can mean that, when initialising, the vehicle was able to measure in its frame a known vector with certainty, say the direction of a distant star thanks to a high-definition camera. Denoting by s_{fixed} and s_0 the direction of the star in the fixed and the initial mobile frame respectively, this reads:

$$R_0^T s_{fixed} = s_0 \Leftrightarrow R_0 s_0 = s_{fixed} \tag{3.30}$$

Thus, Corollary 3 states that the true system always knows the true direction of the star, i.e., satisfies, for s_i such that $s_{i+1} = \Omega_i^T s_i$,

$$\forall i,\ R_i^T s_{fixed} = s_i. \tag{3.31}$$

Therefore, the system only has one degree of freedom left, the rotation around the axis defined by the direction of that star.

3.4.4 The L-IEKF naturally respects the constraints

This section shows the chapter's main result, stating that the geometric constraints defined by Equation (3.22) are propagated by the L-IEKF the same way as in Proposition 4, without having to incorporate them in the filter as hard constraints like (artificial) perfect measurements [95]. We therefore look at systems whose outputs are of the form

$$z_k^L = \chi_k \cdot d + V_k \tag{3.32}$$

Theorem 5. *Consider the L-IEKF described by* (3.11) *and* (3.12)*, with initial equality constraint* (3.22)*. It implies the constraint* (3.23) *at all times, with* G_i, L_i *given by* (3.24)*. Now, note that* $\hat{\chi}_i \in a_i G_i$ *rewrites* $\eta_i \in G_i$ *due to definition* (2.18)*,* η_i *standing for* $\eta_{i|i-1}$ *and* $\eta_{i|i}$ *equivalently. Suppose now that the filter is initialised such that*

$$\eta_0 \in G_0, \qquad \mathrm{Im}(P_0) \subset L_0 \tag{3.33}$$

Then the L-IEKF estimate and covariance satisfy

$$\forall i,\ \eta_i \in G_i, \qquad \mathrm{Im}(P_i) \subset L_i \tag{3.34}$$

The second equality indicates the covariance output by the L-IEKF correctly encodes an absence of dispersion outside of the submanifold (3.23)*.*

Proof. To prove the theorem, it is enough to show the following four implications:

(i) $\eta_{i|i} \in G_i \Rightarrow \eta_{i+1|i} \in G_{i+1}$

(ii) $\mathrm{Im}(P_{i|i}) \subset L_i \Rightarrow \mathrm{Im}(P_{i+1|i}) \subset L_i$

(iii) $\mathrm{Im}(P_{i|i-1}) \subset L_i \Rightarrow \mathrm{Im}(P_{i|i}) \subset L_i$ for $i > 0$

(iv) $[\eta_{i|i-1} \in G_i] \wedge \mathrm{Im}(P_{i|i-1}) \subset L_i \Rightarrow \eta_{i|i} \in G_i$

Proof of (i): As the considered system has deterministic dynamics, this directly comes from Proposition 4 and (3.11).
Proof of (ii): Since the propagation is noise free, the propagation writes

$$P_{i|i-1} = F_{i-1} P_{i-1|i-1} F_{i-1}^T.$$

Since we assume that $\mathrm{Im}(P_{i-1|i-1}) \subset L_{i-1}$, then $\mathrm{Im}(P_{i|i-1}) \subset F_{i-1} L_{i-1} = L_i$.

Proof of (iii) and (iv): Equation (3.12) rewrites in terms of error:

$$\eta_{i|i} = \eta_{i|i-1} \exp\left[K_i(\eta_{i|i-1}^{-1} d - d + \hat{V}_i)\right]. \tag{3.35}$$

As $K_i = P_{i|i-1} H_i^T S_i^{-1}$ from (3.12), the image of K_i is included in that of $P_{i|i-1}$, triggering that Im $K_i \subset L_i$. Since $P_{i|i} = (I - K_i H_i) P_{i|i-1}$, this directly leads to $\mathrm{Im}(P_{i|i}) \subset L_i$.

It also follows that

$$\exp(\left[K_i(\eta_{i|i-1}^{-1} d - d + \hat{V}_i)\right] \in G_i$$

Therefore, the updated error lies in G_i:

$$\eta_i \in \eta_{i|i-1} G_i = G_i$$

according to the assumption of (iv), which concludes the proof. □

Remark 8. *This proof highlights the three key IEKF components that are the chosen class of dynamics, the use of the Kalman gain, and the exponential update.*

Moreover, as in the linear case, note that the results would still hold if another type of observation was chosen, further emphasising the importance of having a state-space structure adapted to the dynamics.

3.4.5 Direct corollary for the R-IEKF

If the output is of the form

$$z_k^R = \chi_k^{-1} \cdot d + V_k, \tag{3.36}$$

rather than (3.32), then it is said to be right-equivariant, and one should use a right-invariant EKF (R-IEKF), see [14]. The results remain entirely valid then, by symmetry.

Theorem 6. *If the hypotheses of Theorem 5 are satisfied, with output* (3.36) *instead of* (3.32)*, then the R-IEKF estimates also verify* (3.34)*.*

Proof. If χ follows a group-affine dynamics, then also does χ_i^{-1}. Using $\chi_i^{-1} \in G$ as the state variable, the output (3.36) becomes left-equivariant as (3.32). And the R-IEKF update for the variable χ_i^{-1} is exactly the L-IEKF update. Theorem 5 then applies. □

3.4.6 Graphical illustration of the theorem and discussion

As it was already known, part of what makes the IEKF work where a filter with linear update such as the EKF fails mostly is where the linearisation is done. Indeed, the EKF tries to linearise on a non-linear space by embedding the state in the ambient vector space. Think again of $SO(3)$: there is no simple way of expressing a rotation as the sum of another rotation and some matrix. The IEKF however linearises on the Lie algebra of the system, which is a linear space in its own right, the exponential map being just a translation between the two. When one writes $\chi = \exp(\xi)$, ξ is the axis of rotation of χ, the angle being the vector's norm, and summing rotation vectors makes perfect sense. It was already the idea behind the Multiplicative EKF (MEKF) [43].

The second main argument which makes the proof work is the fact that the image of a Lie sub-algebra by the exponential map is a subgroup of the associated Lie group. This comes from the Baker-Campbell-Hausdorff formula, which states that if $X, Y \in \mathfrak{g}$, then $e^X e^Y = e^Z$ where Z is a series of X, Y and nested Lie bracket terms. Z thus stays in the sub-algebra.

This is illustrated by Figure 3.1, which gives a schematic view of the difference among the linear, the MEKF and IEKF updates, for an estimate lying on the subgroup represented by the circle. The IEKF update, through the exponential map, makes the estimate move along the circle. Since the covariance is expressed in the Lie algebra, its alignment stays consistent with the subgroup (3.23). On the contrary, the MEKF gives no guarantee that the estimate will stay in the subgroup. In the meantime, the EKF updates along a straight line in the direction of the covariance, so there are no guarantee that the estimate will even remain in G, or that the covariance will stay consistent with the curvature of the space.

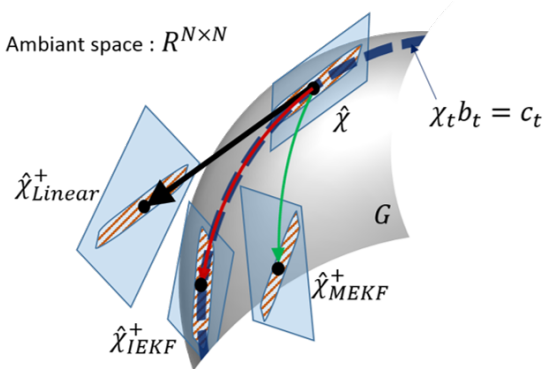


Figure 3.1: Schematic difference among the linear, the MEKF and the IEKF updates, illustrating Theorem 5. The dotted curve represents the subgroup (3.23), and the hatched areas the covariances, originally consistent with (3.23). The IEKF stays on the subgroup, while the MEKF and the EKF update respectively leave the subgroup and even the group G.

As already said, the interest is that when process noise is low, the hard constraint becomes useless. However, the state will live near the manifold defined by (11), and so will the IEKF estimate due to the filter's very structure! It is easily understood that small process noise will lead to a situation being close to the one of Figure 3.1 indeed (the system smoothly depends on the process noise amplitude).

3.5 Examples

This section presents three examples illustrating the implications of Theorems 5 and 6. The first one shows that the result of [13] now appears as a direct application of Theorem 5. The second one presents the implications of Theorem 6 for the attitude estimation example of Section 3.4.3, and illustrates what happens when noise is turned on. The last one focuses on unbiased inertial navigation, for a general system, and a legged robot [61].

3.5.1 Car position and heading estimation

Recall and transcription of the results of [13] in the discrete case

Consider the simple case of a non-holonomic car with perfect odometry, unknown heading and noisy position measurements. Suppose the initial position of the car is known. The dynamics are given by:

$$\theta_{i+1} = \theta_i + \omega_i, \qquad x_{i+1} = \begin{pmatrix} \cos(\theta_i)u_i \\ \sin(\theta_i)u_i \end{pmatrix}, \tag{3.37}$$

where θ_i is the heading of the car, x_i its position vector, and ω_i, u_i are the angular and linear velocities. Noisy position measurements $z_k = x_k + V_k$ are corrupted by white noise V_k.

As was then proven in [13], if $R(\theta)$ denotes the rotation matrix of angle θ, and $\hat{\theta}_i, \hat{x}_i$ denote the IEKF estimates, then $R(\theta_i)^T x_i = R(\hat{\theta}_i)^T \hat{x}_i = b_i$, where b_i is defined through the dynamics

$$b_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad b_{i+1} = R(\omega_i)^T (b_i + \begin{pmatrix} u_i \\ 0 \end{pmatrix}). \tag{3.38}$$

Figure 3.2, reproduced from [13], displays the trajectories of the true car and both EKF and IEKF estimates, for $\omega_i \equiv 0$. The IEKF car estimate is always traveling on a ray that passes through the origin, while that is not true for the EKF. At the first update, the EKF will stop satisfying this property, in part because it does not use the exponential update. Indeed, while the IEKF updates by "rotating" on the circle the true state lives in, the EKF will move tangentially to it, and leave it.

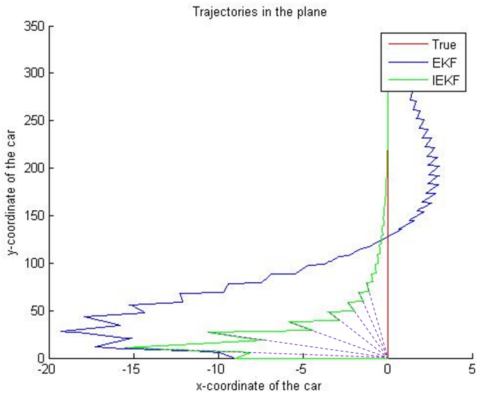


Figure 3.2: From [13]. First numerical example of Theorem 5: True, IEKF and EKF estimates trajectories of a car with perfect odometry, known initial position and uncertain initial heading. The car gets position measurement every 10 time steps. The odometry indicates $\omega_i \equiv 0$, thus we know the car is necessarily following a ray emanating from its original position, just not with which angle. The IEKF encodes correctly this information, as its estimates always move along rays emanating from the initial position indeed.

Translation in the Lie group formalism

The translation of this pose estimation problem in the Lie group $SE(2)$ is given in Section 2.4.2, through a state χ_i verifying

$$\chi_i = \begin{pmatrix} R(\theta_i) & x_i \\ 0 & 1 \end{pmatrix}, \quad \chi_{i+1} = \chi_i U_i, \quad U_i = \begin{pmatrix} R(\omega_i) & \begin{matrix} u_i \\ 0 \end{matrix} \\ 0 & 1 \end{pmatrix} \tag{3.39}$$

Moreover, the position measurements and initial known position are respectively given by

$$z_k = \chi_k \ (0\ 0\ 1)^T, \quad \chi_0 \ (b_0\ 1)^T = (0\ 0\ 1)^T \tag{3.40}$$

Thus, Proposition 4 implies that for $\beta_0 = \begin{pmatrix} b_0 & 1 \end{pmatrix}^T$ and $\beta_{i+1} = U_i^{-1}\beta_i$, the state satisfies $\chi_i \beta_i = \begin{pmatrix} 0 & 0 & -1 \end{pmatrix}^T$ for all i, which is exactly what is stated in Section 3.5.1.

In turn, the result of [13] showing that the L-IEKF preserves the latter property is in fact a direct corollary of Theorem 5.

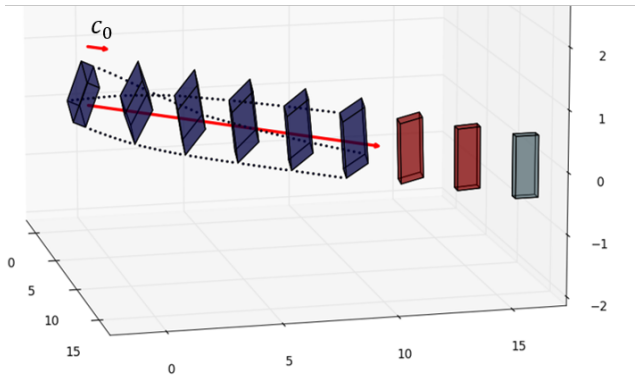


Figure 3.3: Second numerical example. Bricks represent the rotation matrices associated to the error between the true trajectory, and the estimated one. Dots materialise rotations from one block to the next one, by highlighting the trajectory of each of the base vectors. The arrow represents the (known) direction of the distant star, c_0. The rotations induced by the first five updates of the IEKF are shown (blue), followed by the results after the 15th and 30th update (red) ,which show the convergence of the error to the identity matrix (light blue). It can be clearly seen that they turn around c_0.

3.5.2 Attitude estimation

This example builds upon the work of [10]. Indeed, it did not consider the structure of the initial covariance, and in particular the impact of rank-deficiency. Consider that the system receives at times t_k, a measurement z_k of the gravity field g and the earth magnetic field b through a triplet of accelerometers and magnetometers, i.e.,

$$z_k = (R_k^T g + V_k^g; R_k^T b + V_k^b), \tag{3.41}$$

with V_k^g, V_k^b two centered noises in $\mathbb{R}^3$. It has thus right-equivariant outputs, and the system is suitable for the design of a R-IEKF. The corresponding filter's equations for this problem were already written in [10]. They are similar to (3.12), with the update consisting of a left multiplication by a term of the form $\exp(K_k z)$, with $z \in \mathbb{R}^n$, that is, a rotation around the axis $K_k z$. Thus, Theorem 6 implies that, if initialised correctly, the estimate will correctly know the direction of the star at all times whatever the motion. Figure 3.3 displays the results of numerical experiments. As stated, all the updates consist of rotations sharing the same axis, the direction of the star denoted by c_0.

Impact of noise

We also performed simulations with process noise having standard deviation equal to 0.02 degrees/s. During all the experiment, the angle between c_0 and the axis of the updates' rotations was of course impacted and did not remain null. However, it never exceeded 0.01 degree, and the estimated star direction remained in a sharp cone, as could be expected from the IEKF's geometrical structure.

3.5.3 Unbiased inertial navigation on flat-earth

Considered dynamics

Consider a vehicle evolving in the 3D space, whose attitude, speed and position, given by R_i, v_i and x_i respectively are tracked with accelerometers and gyroscopes. Their measures are denoted respectively by u_i and ω_i. As explained in Section 2.4.3, the dynamics then reads:

$$R_{i+1} = R_i \Omega_i, \quad v_{i+1} = v_i + dt(g + R_i u_i), \quad x_{i+1} = x_i + dt\ v_i, \tag{3.42}$$

where $\Omega_i = \exp_{SO(3)}(dt\omega_i)$ and g denotes the gravitational field. This can be embedded in the Lie group $SE_2(3)$ to rewrite the dynamics in compact form

$$\chi_i = \begin{pmatrix} R_i & v_i & x_i \\ 0 & & I_2 \end{pmatrix}, \quad \chi_{i+1} = \gamma \phi(\chi_i) \upsilon_i, \tag{3.43}$$

where $\gamma = (Id, dt\ g, 0)$, $\upsilon_i = (\exp(dt(\omega_i)), dt\ a_i)$, and $F : (R, v, x) \mapsto (R, v, x + dt\ v)$.

Initial configuration

Suppose the inertial unit is turned on while the vehicle stays at a fixed known location, that is $\hat{x}_0 = x_0$ and $\hat{v}_0 = v_0 = 0$. It thus only need estimate its attitude $\hat{R}_i$. From the Lie group point of view, this rewrites:

$$\eta = \chi^{-1}\hat{\chi} = (R^T\hat{R}, 0, 0),$$

i.e., lies in a determined subgroup G_0 of G. In particular, $G_0 = \{\exp(\omega, 0, 0),\ \omega \in \mathbb{R}^3\}$. The accelerometer outputs $-R_i^{-1}g$, thus we can directly see from the dynamics of (3.42) that the estimate will follow a uniformly accelerated trajectory along the direction $(Id - \hat{R}_0 R_0^{-1})g$, which is constant between two successive updates. For instance, the estimate stays still for $\hat{R}_i = R_i$, and goes down if $\hat{R}_i R_i^{-1} g = -g$. For any given time i, the estimated position is known to lie on a sphere with radius r proportional to i^2, centered at $x_0 - (0, 0, r/2)$. Moreover, its speed is collinear to its position vector, the quotient of the norms of both vectors being proportional to i: $\eta_i \in G_i = \{(\hat{R}_0 R_0^{-1}, i(Id - \hat{R}_0 R_0^{-1})g, \frac{i^2}{2}(Id - \hat{R}_0 R_0^{-1})g\}$. Notice that G_i is not exhaustively represented by the sphere of Figure 3.4, as each of the points corresponds to a single velocity but an infinity of attitudes, parametrised by the rotations around g, i.e., the heading.

Consequence for the IEKF

Suppose that the system observes its position at some steps. Similarly to the case of the non-holonomic car of Section 3.5.1, in the matrix Lie group formulation, outputs are of the form of $z_k = \chi_k(0, 0, 0, 0, 1)^T$. These are suited for the design of an L-IEKF. Then, if properly initialised, Theorem 5 implies, on the first hand, that the estimated covariance stays distributed accordingly during both propagation and update steps, and on the second hand, that an update moves the estimate on the sphere, while respecting the common direction shared by its speed and position vectors, and the proportion between their norms. Properly initialised simply means here that $\operatorname{Im} P_0 \subset \{(\omega, 0, 0),\ \omega \in \mathbb{R}^3\}$. Figure 3.4 illustrates simulation results of this L-IEKF for navigation on flat earth. It displays the straight line described by the movement of the estimate until the first measurement is received, and the induced update. The pre- and post-update velocities are also depicted. It appears clearly that each velocity describes a line joining the origin with the associated estimated position.

The practical example of a walking robot of [61,62]

The IEKF was successfully applied to the estimation of the navigation state of a legged robot, along with its contact points in [62]. The authors recently extended their work in [61], illustrating at the same time the properties of the IEKF which is presented here. Indeed, Figures 6, and especially 7, support the fact that the IEKF will respect the true distribution of a robot having an unknown initial heading and moving forward along a straight line. This is close to the situation depicted in Section 3.5.1. The Figure 7 of this paper shows that, if the initial heading is completely unkonwn, then the estimated position distribution given by the IEKF is a "thick" growing circle, whose radius is given by the walking mean speed, and width by the initial position and propagation uncertainties.

3.6 Conclusion

This chapter presented the IEKF, and recalled its theoretical stability properties, which make it a particularly adapted tool for navigation. Moreover, it highlighted the shortcomings of Extended Kalman Filtering for high-accuracy navigation problems through the degenerate situation of an infinitely accurate geometric prior. Without resorting to artificial process noise, the EKF fails to propagate this information. On the contrary, the IEKF proved to be a robust response to the issue of assimilating precise non-linear prior information, i.e., rank-deficient initial covariance. These situations are especially important in inertial navigation where they have to be handled carefully. This further advocates its use for attitude estimation, and more generally state estimation in navigation, when using very precise motion sensors.

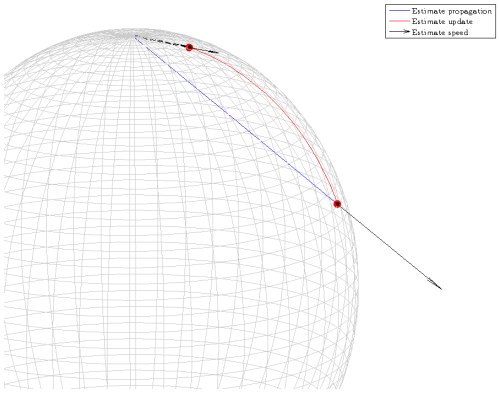


Figure 3.4: Numerical example of Theorem 5 applied to navigation on flat earth: IEKF propagation (in blue) and first update (in red) of the estimate on $SE_2(3)$ with deterministic dynamics and known initial position and velocity but uncertain initial attitude. The true vehicle stays still during the attitude estimation, thus the inertia indicates $\omega_i = 0$ and $u_i = -g$. The effect of the update on the estimated position was represented as a circle segment on the sphere of the positions of the elements of G_i (light grey). The estimated velocities at the time of the first measurement, before and after the update, are drawn as arrows at the corresponding locations. A dotted line extends the post-update velocity to highlight the fact that it stays collinear with the position vector. For the sake of simplicity, attitudes were not explicitly shown and the true state was taken as the group identity, so that the error and the estimate coincide.

Chapter 4

Invariant smoothing

Highlights

- Smoothing is a back-end model which tries to overcome the limitations of standard filtering in local navigation, but that can also be applied to global navigation.
- Invariant Smoothing is introduced as an extension of the Invariant filtering framework to the estimation of a full trajectory.
- For global navigation, Left-Invariant Smoothing (L-IS) reduces, and sometimes almost cancels, the dependency of the system's information matrix on the current estimate, quickening convergence and making the smoother more consistent for group-affine systems.
- In inertial navigation, L-IS solves the issues that the IEKF may encounter when the biases are large.
- Right-Invariant Smoothing (R-IS) is introduced for local navigation. It does not share the same properties as L-IS, and will be studied in more details in the subsequent chapter. Another variant of L-IS, called Hybrid Invariant Smoothing, is also presented.
- The recently revisited theory of preintegrated IMU measurements is applied to Invariant Smoothing to outline the differences between the existing methods and the ones proposed here.

Points marquants

- Le lissage est une méthode d'estimation dont le but est de surmonter les limites du filtrage standard en navigation locale, mais qui peut aussi s'appliquer à la navigation globale.
- Le Lissage Invariant est introduit comme extension de la théorie du filtrage invariant à l'estimation d'une trajectoire complète.
- Pour la navigation globale, le Lissage Invariant à Gauche (L-IS) réduit, voire parfois annule, la dépendance de la matrice d'information du système en l'estimé actuel, accélérant la convergence et produisant des estimés plus réguliers pour les systèmes groupe-affine.
- En navigation inertielle, le L-IS résout les problèmes rencontrés par l'IEKF lorsque les biais sont importants.
- Le Lissage Invariant à Droite (R-IS) est proposé pour la navigation globale. Il ne partage pas les mêmes propriétés que le L-IS, et sera étudié plus en détails dans le chapitre suivant. Une autre variante du L-IS, appelée Lissage Invariant Hybride, est aussi présentée.
- La théorie de la préintégration des mesures inertielles, récemment revisitée, est appliquée au Lissage Invariant pour souligner les différences entre les méthodes existantes et celles proposées ici.

Introduction (En/Fr)

Part of the present chapter, mostly Sections 4.2 and 4.3, was presented at the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), see [37].

As explained in Section 4.1, the shortcomings of filtering methods for SLAM lead to use the smoothing paradigm, a shift which was then also taken for other navigation problems. The recent results of the invariant filtering framework outlined in Chapters 2 and 3 convinced us to build upon it and derive a specific smoothing framework adapted to robot localisation, called Invariant Smoothing, and which is one of the two main contributions of this thesis. It turns out that, for *left-linear observed systems on group* (see Definition 5), the dependence of the whole information matrix of the system with respect to the linearisation points is reduced, or even removed under mild conditions and small approximations. This brings immediate benefits: 1) The number of iterations to convergence is reduced 2) The consistency of the smoothing estimate is improved. These two features are illustrated on simulated and experimental 2D localisation problems. A simulated biased inertial navigation problem further outlines the advantages of L-IS over the IEKF in cases where the dynamics are group-affine only up to an unknown parameter, here the biases. Finally, the theory is extended to the right-invariant error. It does not exhibit the same almost independence property, however Section 5.2 will show that it is better suited for local navigation, in particular because of its observability properties.

Une partie du présent chapitre, principalement les Sections 4.2 et 4.3, ont été présentées à la conférence 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), voir [37].

Comme expliqué dans la Section 4.1, les limites des méthodes de filtrage pour le SLAM ont mené à l'utilisation du paradigme de lissage, un changement aussi observé pour d'autres problèmes de navigation. Les récents résultats du filtrage invariant soulignés dans les Chapitres 2 et 3 nous ont convaincu de nous appuyer dessus pour dériver un cadre spécifique de lissage adapté à la localisation de robots, appelé Lissage Invariant, qui est l'une des deux principales contributions de cette thèse. Il s'avère que, pour les *systèmes linéairement observés à gauche sur groupe* (voir Définition 5), la dépendance de la matrice d'information complète du système par rapport au point de linéarisation est réduite, et même disparaît avec de légères hypothèses et faibles approximations. Cela amène des bénéfices immédiats : 1) Le nombre d'itérations jusqu'à convergence est réduit 2) La cohérence du lissage est améliorée. Ces deux propriétés sont illustrées sur des problèmes de localisation 2D simulés et expérimentaux. Un problème simulé de navigation inertielle biaisée souligne de plus l'avantage du L-IS comparé à l'IEKF dans les cas où la dynamique est groupe-affine à un paramètre inconnu près, ici les biais. Enfin, la théorie est étendue à l'erreur invariant à droite. Ce cas ne présente pas les mêmes propriétés d'indépendance, mais la Section 5.2 montrera qu'il est mieux adapté à la navigation locale, en particulier du fait de ses propriétés d'observabilité.

4.1 An introduction to the smoothing framework

4.1.1 From filtering to smoothing

For linear systems, the Kalman filter and smoothing coincide, in the sense that, at step i, their estimates $\hat{X}_i$ coincide, as shown by Theorem 4. This fact will be leveraged in the second part of this thesis, see Sections 8.1. As already mentioned, in a non-linear context the EKF is well-known to lose optimality. Recall that this was the reason of the degeneracy of the EKF-SLAM. A first solution was proposed in the form of the iterated EKF, which aimed at finding the best linearisation point at each update. It later appeared that it is in fact the application of a Gauss-Newton scheme to the following optimisation problem [19]:

$$\underset{\chi_n}{\operatorname{argmin}} \|\hat{\chi}_{n|n-1} - \chi_n\|^2_{P_{n|n-1}} + \|h(\chi_n) - z_n\|^2_{N_n} = \underset{\chi_n}{\operatorname{argmax}} \mathbb{P}(\chi_n | \hat{\chi}_{n|n-1}, z_n) \tag{4.1}$$

Still, this is only an *approximation* of the maximum of the distribution we seek, which is $\mathbb{P}(\chi_n | z_0, \cdots, z_n)$. Nevertheless, once this step is taken, the next one comes naturally: avoiding marginalisation by directly estimating the maximum we seek, through

$$\underset{\chi}{\operatorname{argmax}} \mathbb{P}(\chi_0, \cdots, \chi_n | z_0, \cdots, z_n) \tag{4.2}$$

Method	Distribution maximised
Linear KF	$\mathbb{P}(\chi_n \vert z_0, \cdots, z_n)$
EKF	Unknown in general
iterated-EKF	$\mathbb{P}(\chi_n \vert \hat{\chi}_{n\vert n-1}, z_n)$
Smoothing	$\mathbb{P}(\chi_0, \cdots, \chi_n \vert z_0, \cdots, z_n)$

Table 4.1: Filtering as an approximation of smoothing

where the state χ contains the whole trajectory [45,80]. That is, the chosen estimate is the Maximum a Posteriori (MAP) estimator. Therefore, the various definitions of non-linear Kalman filtering can be seen as approximations of smoothing. This is summarised in Table 4.1

4.1.2 Smoothing as a non-linear least-squares problem

State estimation in navigation and robotics relies on two standard assumptions to make it tractable:

Gaussianity All noise is supposed to follow zero-mean Gaussian distributions.

Conditional independence All the measurements are supposed to be realisations of independent random variables, i.e., the noises of the various sensors are independent, and so are the successive measures of a same sensor. This amounts to say the measurements only depend on the states they are related to, i.e. are assumed Markovian.

This Markovian assumption is powerful, as it is the one leading to the fact that the past is forgotten in global navigation, but it is usually satisfied in practice. When it is not, it can be accounted for during pre-processing (the front-end), or by modelling, as with the IMU biases.

Under these hypothesis, smoothing boils down to a non-linear least-squares problem.

Indeed, all the measurements, be it the prior, propagations or observations, can be written as a set of random variables in the form

$$z_k = \lambda_k(\chi, \epsilon_k), \tag{4.3}$$

where ϵ_k is a zero-mean gaussian noise, following the first assumption. For all k, this noise can be isolated in (4.3) as

$$\epsilon_k = \psi_k(\chi, z_k) \tag{4.4}$$

Let $z_{0:n}$ denote the set $(z_0, \ldots, z_n)$. The estimate is defined as $\chi^* = \operatorname{argmax}_\chi \mathbb{P}(\chi | z_{0:n})$ According to Bayes theorem, this distribution is given, up to a multiplicative constant, by

$$\mathbb{P}(\chi | z_{0:n}) \propto \mathbb{P}(z_{0:n} | \chi)\mathbb{P}(\chi) \tag{4.5}$$

Injecting (4.4) leads to

$$\begin{aligned}
\mathbb{P}(\chi | z_{0:n}) &\propto \mathbb{P}(w_{0:n}) && (4.6)\\
&= \prod_k \exp(-\frac{1}{2}\|\psi_k(\chi, z_k)\|^2_{\Sigma_k}) && (4.7)\\
&= \exp(-\frac{1}{2}\sum_k \|\psi_k(\chi, z_k)\|^2_{\Sigma_k}) && (4.8)
\end{aligned}$$

where $\|e\|^2_\Sigma = e^T \Sigma^{-1} e$ is the Mahalanobis distance. Therefore, defining the estimate via the maximisation (4.2) is equivalent to defining

$$\chi^* = \operatorname*{argmin}_\chi \sum_k \|\psi_k(\chi, z_k)\|^2_{\Sigma_k}, \tag{4.9}$$

Remark 9. *The notation $\mathbb{P}(z_{0:n}|\chi)\mathbb{P}(\chi)$ covers different modeling perspectives, which are equivalent and lead to the same estimate. Indeed, the early approaches separate the measurements between propagation, to go from i to $i+1$, and observations (GPS, range and bearing measurement, etc.). The propagation measurements and the prior on the initial state would form $\mathbb{P}(\chi)$. Another way to see it is to consider all measurements as various observations under a unified framework such as* (4.4)*. While that is slightly more general, the MAP is well-defined only if the associated non-linear least squares is well-posed, which boils down to having a kind of propagation between the states.*

4.1.3 Solving the optimisation with iterative methods

From a theoretical point of view, this non-linear optimisation problem has been known for decades, but solving it has long proved intractable in general. To the author knowledge, all existing navigation solvers rely on iterative linearisations of (4.9), i.e., the estimate is obtained by a series of first-order perturbations from an initial guess. For instance, given an estimate $\hat{\chi}$, the following update can be computed

$$\delta\chi^* = \underset{\delta\chi}{\operatorname{argmin}} \sum_k \|\psi_k(\hat{\chi}+\delta\chi, z_k)\|^2_{\Sigma_k} \tag{4.10}$$

$$\approx \underset{\delta\chi}{\operatorname{argmin}} \sum_k \|\psi_k(\hat{\chi}, z_k) + J_k\delta\chi\|^2_{\Sigma_k} \tag{4.11}$$

$$= \underset{\delta\chi}{\operatorname{argmin}} \|A\delta\chi - b\|^2_{\Sigma}, \tag{4.12}$$

where the J_k are the jacobians of the ψ_k at the current estimates, collected in a matrix A, the right-hand side b gathers the currently estimated values of the ψ_k, and Σ is the block-diagonal matrix made of the Σ_k. This update scheme is the Gauss-Newton (GN) algorithm used for instance in [45], however other schemes based on linearisation can also be considered, such as gradient descent or Levenberg-Marquardt. To sum up, these three ways of looking for updates are

$$\text{Gradient-descent (GD)}\delta\chi^* = -\lambda A^T\Sigma^{-1}b,\ \lambda > 0 \tag{4.13}$$

$$\text{Gauss-Newton (GN)}A^T\Sigma^{-1}A\delta\chi^* = A^T\Sigma^{-1}b \tag{4.14}$$

$$\text{Levenberg-Marquardt (LM)}(A^T\Sigma^{-1}A+\lambda D)\delta\chi^* = A^T\Sigma^{-1}b,\ \lambda > 0,\ D \in \{Id, diag(A^T\Sigma^{-1}A)\}. \tag{4.15}$$

Using iterative methods leads to relinearising the jacobians when needed, and in turn this makes the smoothing approach much more robust to inconsistencies than the standard EKF approach.

Remark 10. *These schemes are not second-order methods, however they have a much lower computational complexity than Newton's method for instance. To ensure a better convergence, they should theoretically be coupled with a line-search scheme [1], however the major successes of the various SLAM solvers which exist showed that it is not necessary in practice.*

Solving the normal equations (4.14) or its LM variant (4.15) is the main task to be addressed when implementing factor graph optimisation. Indeed, one is faced with a system of linear equations. Most popular methods rely on factorisations of the information matrix defined by $\mathcal{I} := A^T\Sigma^{-1}A$ in the form of $A^T\Sigma^{-1}A = LL^T$ with L lower triangular (Cholesky) or through a QR factorisation. Matrix L is referred as the "square root" of the information matrix, and allows to solve the problem by first solving $L\eta = A^T\Sigma^{-1}b$ and then $L^T\delta\chi^* = \eta$ by back-substitution, see [45].

Smoothing through factor graphs What made smoothing so successful is its computational complexity. Indeed, although the full trajectory is considered, the linear systems which are to be inverted, (4.14) or (4.15), are heavily *sparse*. Indeed, the observations z_k each involve only a limited number of states, usually one (e.g., prior, GPS measurement), or two (e.g., relative measurement, propagation). We can thus construct a bipartite graph associated to the non-linear least-squares, that contains variable nodes $X_i \in \chi$ and factor nodes associated to the observations and linked with the variables they involve. Each factor may be viewed as a constraint between variables of the state χ one seeks to estimate. This structure is called a factor graph and allows encoding the linearised system at each step [45,46]. It is at the heart of the existing solvers, which combine graph theory and linear algebra to be as efficient as possible [54,73,76,78,93]. An example of a factor graph and the filling of its associated A matrix is given in Figure 4.1.

4.1.4 Smoothing inside a sliding window

Smoothing estimates a full trajectory. To allow for long-term estimation, the number of states that are kept, or at least actively estimated, must be controlled. Indeed, although it is not quadratic, the computational cost still grows with the trajectory's length. To bound this cost, sliding windows are often used [40], but other sparsification techniques have been proposed, usually based on information

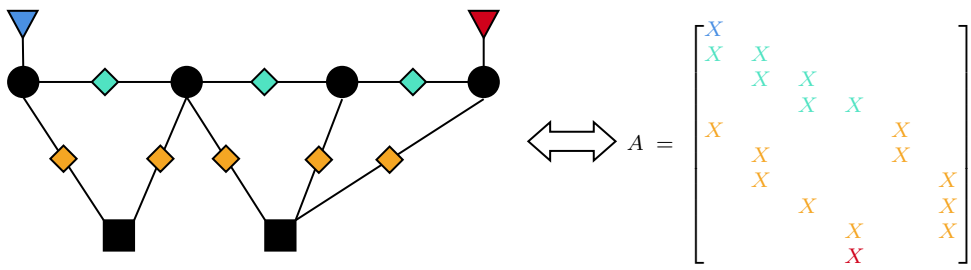


Figure 4.1: Example of a factor graph with four navigation states (black circles), two features (black squares), a prior (blue triangle), propagation factors (turquoise losanges), relative observation (orange losanges) and a unary measurement (red triangle). The filling of the matrix A, made by ordering first the navigation states and then the features, is denoted with the X blocks, colored to match the related factors.

theory [35,41,64,100]. Here, the simple approach of a sliding window is chosen. That is, the number of estimated states is bound by a fixed value, and once the estimate reaches it, each time a new state is added, the oldest one is marginalised out.

4.1.5 Smoothing on manifold

A large body of literature is dedicated to smoothing of systems which do not live in vector spaces but on a manifold $\mathcal{M}$, such as rotation matrices for instance. The main idea is to replace the $-$ and $+$ operators of the vector space by adapted ones $\boxminus$, $\boxplus$, which highlight the fact that $\mathcal{M}$ is not a vector space and that linearisation takes place on its tangent space $T\mathcal{M}$, where $\delta\chi$ should live. Thus, these operators are defined such that $\boxminus : \mathcal{M} \times \mathcal{M} \to T\mathcal{M}$ and $\boxplus : \mathcal{M} \times T\mathcal{M} \to \mathcal{M}$. In practice, this impacts the way the jacobians J_k are defined, and the way $\hat{\chi}$ is updated.

When the manifold is a Lie group, the group multiplication and its intrinsic operators log and exp naturally yield a family of such operators, through the following definitions:

$$\boxplus_L(\chi, \xi) = \exp(\xi)\chi \qquad \boxminus_L(\chi_1, \chi_2) = \log(\chi_1 \chi_2^{-1}) \tag{4.16}$$

$$\boxplus_R(\chi, \xi) = \chi \exp(\xi) \qquad \boxminus_R(\chi_1, \chi_2) = \log(\chi_2^{-1} \chi_1) \tag{4.17}$$

Note that, these operators were used in the definitions of Section 2.1.1. The choice of $\boxplus$ and $\boxminus$ are part of the non-linear part of smoothing, i.e., how the system should be linearised, and are thus included in the System's model block of Figure 1.4. The usual approaches of smoothing on manifold are focused on leveraging the intrinsic tools of a given structure of the state space (Riemannian manifold or Lie group). For instance in [27], the proposed algorithm only required the state to belong to a Lie group, but no assumption is made regarding the dynamics and observation functions. In the present chapter, the equations of the system are assumed to have specific properties with respect to the chosen group structure. This reduces the application field, but ensures in turn striking properties.

4.2 Left-invariant smoothing

In this section, we consider smoothing applied to the linear observed systems on groups from [18] and introduced in Section 3.2. It builds on the invariant filtering framework, hence its name. The remarkable result is that the dependency of the information matrix of smoothing algorithms (partially) disappears, owing to the interplay between the group structure and the system's properties.

4.2.1 Considered systems

Invariant Smoothing focuses on the modeling part of the non-linear estimation process, i.e., the left-hand side of Figure 1.4. The remainder of this section focuses on observations in the form of left group actions, i.e., satisfying (2.3). Just as in the case of the IEKF, this prompts the use of the left multiplication based

uncertainty model (2.7), leading to *Left-Invariant Smoothing* (L-IS). Therefore, given a current guess of one of the trajectory's states, $\hat{\chi}_i$, the cost function will be linearised, noting ξ_i the searched parameters, and $\Xi = (\xi_i)_i$, using

$$\chi_i = \hat{\chi}_i \exp(\xi_i), \qquad \chi = \hat{\chi} \exp(\Xi). \tag{4.18}$$

Let Y be the space in which the observations live, and suppose it is a Lie group. The noisy systems to which L-IS applies are modeled according to:

$$\chi_{i+1} = f_i(\chi_i) \exp(w_i), \qquad w_i \sim \mathcal{N}(0, Q_i) \tag{4.19}$$

$$z_k = \chi \star b_k \ \exp_Y(v_k) \qquad v_k \sim \mathcal{N}(0, N_k) \tag{4.20}$$

where f_i is a group-affine dynamics, and $b_k \in Y$.

4.2.2 Factors definition

Propagation factors

Isolating the noise in (4.19), we get $\exp(w_i) = f_i(\chi_i)^{-1}\chi_{i+1}$. Let $\hat{\chi}_i, \hat{\chi}_{i+1}$ be some guesses, and let us write the true χ_i, χ_{i+1} using the Lie-group based error model (4.18). This yields

$$\exp(w_i) = f_i(\chi_i)^{-1}\chi_{i+1} \tag{4.21}$$

$$= f_i(\hat{\chi}_i \exp(\xi_i))^{-1}\hat{\chi}_{i+1} \exp(\xi_{i+1}) \tag{4.22}$$

Using (2.19), and then (2.28) this is equal to

$$\begin{aligned}
&= (f_i(\hat{\chi}_i) f_i(Id)^{-1} f_i(\exp(\xi_i)))^{-1}\hat{\chi}_{i+1} \exp(\xi_{i+1}) \\
&= (f_i(\hat{\chi}_i) g_i^L(\exp(\xi_i)))^{-1}\hat{\chi}_{i+1} \exp(\xi_{i+1}) \\
&= \exp(-F_i^L \xi_i) f_i(\hat{\chi}_i)^{-1}\hat{\chi}_{i+1} \exp(\xi_{i+1})
\end{aligned}$$

Let $a_i = \log(f_i(\hat{\chi}_i)^{-1}\hat{\chi}_{i+1})$ be the estimation error for the guess. Then, taking the logarithm on both sides and using the BCH formula, only keeping the first order terms in ξ *and* a_i leads to

$$w_i \approx a_i - F_i^L \xi_i + \xi_{i+1} \tag{4.23}$$

The major advantage (and very remarkable feature) of this parametrisation is the fact that, if the dynamics is accurate enough the jacobians *do not* depend on the current estimate since F_i^L is independent of $\hat{\chi}$ from Theorem 2. This was also derived in the particular case of $SE(3) \times \mathbb{R}^n$ in [75].

Measurement factors

Based on (2.31), and as was done for the IEKF in (3.10), the measurement factors are obtained through the following quantities

$$\hat{\chi}^{-1} \star z_k = (\hat{\chi}^{-1}\chi) \star b_k \ \exp_Y(\hat{v}_k), \tag{4.24}$$

where $\hat{v}_k$ is the noise associated to this innovation, and depends on the group action, see Section 2.1.3. Isolating this noise, injecting (4.18), and linearising lead to

$$\exp_Y(\hat{v}_k) = (\exp(\Xi) \star b_k)^{-1}\hat{\chi}^{-1} \star z_k \tag{4.25}$$

$$\hat{v}_k \approx \log_Y((\exp(\Xi) \star b_k)^{-1}(\hat{\chi}^{-1} \star z_k)) \tag{4.26}$$

$$\approx c_k - H_k \Xi \tag{4.27}$$

where $c_k = \log_Y(\hat{\chi}^{-1} \star z_k)$, H_k is the matrix given by a Taylor expansion of the matrix exponential in (2.2) of $\log_Y(\exp(\Xi) \star b_k)$.

The case of vectorial measurements For vectorial measurements, the above equations are greatly simplified. Indeed, each measurement is only related to a single state, and (4.24) becomes

$$\hat{\chi}_k^{-1} v_k = \hat{\chi}_k^{-1} y_k - \exp(\xi_k) d. \tag{4.28}$$

Therefore, the error term c_k involved in the linearisation (4.27) write $c_k = \hat{\chi}_k^{-1} y_k$, while the matrix H is defined by $H\xi = \xi^\wedge d$.

Prior factor

Finally, the factor associated to the prior must also be assessed. Indeed, the noise on the prior now writes

$$\exp(x_0) = \bar{\chi}_0^{-1}\hat{\chi}_0 \exp(\xi_0) \tag{4.29}$$

Contrarily to the propagation factor, here the first-order terms of the BCH formula cannot be reduced to the sum of its arguments, as the term $\log(\bar{\chi}_0^{-1}\hat{\chi}_0)$ will tend to grow over the iterations, sometimes to a point where the jacobian of the Lie group [6] must be taken into account. We thus have in general

$$x_0 \approx p_0 + J_0\xi_0 \Leftrightarrow J_0^{-1}x_0 \approx J_0^{-1}p_0 + \xi_0 = p_0 + \xi_0 \tag{4.30}$$

where $p_0 = \log(\bar{\chi}_0^{-1}\hat{\chi}_0)$, and J_0 is defined as $BCH(p_0,\xi) = p_0 + J_0\xi + o(\|\xi\|^2)$, which implies $J_0 p_0 = p_0$. If the prior error decreases, for instance in a sliding window context, then J_0 can sometimes be approximated by Id.

4.2.3 Final algorithm and benefits of the proposed approach

Regrouping (4.30), (4.23) and (4.27), the linear least-squares problem to be solved at each iteration writes

$$\begin{aligned}\Xi^* = (\xi_i^*)_i = &\underset{(\xi_i)_i}{\operatorname{argmin}} \|p_0 + \xi_0\|^2_{J_0^{-1}P_0J_0^{-T}} \\ &+ \sum_i \|a_i - F_i^L\xi_i + \xi_{i+1}\|^2_{Q_i} + \sum_k \|c_k + H_k\Xi\|^2_{\hat{N}_k}\end{aligned} \tag{4.31}$$

where $\hat{N}_k = \hat{\chi}_k^{-1}N_k\hat{\chi}_k^{-T}$ is the covariance of $\hat{v}_k$, see Table 2.1.

Remark 11. *Note that this linear problem is not exactly in the form of* (4.12)*, as the measurement factors were not defined with isolated noise v_k, but $\hat{v}_k$. Nevertheless, this is only a linear transform of the noise, which can be transfered to the jacobians and residual vectors of the linearised factors to boil down to* (4.4)*.*

Injecting (4.31) into the smoothing framework, we get Algorithm 1 for the left multiplication based parametrisation (4.18) (as opposed to $\chi = \exp(\xi)\hat{\chi}$). The main advantage of this framework are the following facts: (i) The current estimates do not appear in the definition of the propagation and observation jacobians, in (4.23) and (4.27) respectively (matrices F_i^L and H_k), although the latter comes at the expense of a modified covariance for the measurement factors. (ii) However, this dependency disappears if the measurement covariance N_k is such that $N_k = \chi N_k \chi^T$ for all χ. Although this condition can seem restrictive at first, it often boils down to an isotropy assumption, as will be the case for the model of Section 4.3. This leaves only the prior covariance of (4.29) to be a function of the estimate. (iii) If the initial guess is good, this can also be harmlessly relaxed by approximating $J_0 \approx Id$. If conditions $(i)-(iii)$ are met, the information matrix associated to (4.31) can be considered independent from the current estimate.

Remark 12. *Note that this work does not take into account how the state is represented as a member of a Lie group and thus overlooks the possible numerical issues which could come at play, e.g. when multiplying rotation matrices.*

4.2.4 L-IS and the iterated IEKF

L-IS relies on a framework of optimisation on Lie groups through a Gauss-Newton scheme. The iterated EKF can be seen as a GN scheme applied to a non-linear least-squares problem on a vector field [19]. This was lifted to a geometric framework to define an iterated EKF on Lie groups [27]. This could be seen as a particular case of L-IS, with a sliding window of width 1. This is only true given a particular order of marginalisation-update-propagation. Indeed, in the iterated IEKF, the initial prior on the current estimate is zero. For this to be satisfied, the operations must be carried out according to the following order

Store a new GPS measurement → Propagate → Marginalise the oldest state → Update

while L-IS would carry out an update as soon as the measurement is received. Note, however, that Invariant Smoothing with window size 1 does coincide with filtering in the case of relative measurements (e.g., relative poses), as it is necessary to wait for the subsequent state to be able to compute the update.

Algorithm 1: Smoothing for a group-affine system with a left multiplication based parametrisation (4.18)

Input: $(\bar{\chi})_{1\leq i\leq n}, \mathbf{P}_0, (f_i)_i, (Q_i)_i, (z_k)_k, (N_k)_k$;
Initialisation
1 $\forall i$, Set $\hat{\chi}_i^0 = \bar{\chi}_i$,
Until convergence do
2 Linearize around $(\hat{\chi}_i^k)_i$ according to (4.31) ;
3 Solve for $\Xi = (\xi_i)_i$;
4 Update: $\hat{\chi}_i^{k+1} := \hat{\chi}_i^k \exp(\xi_i)$
5 $\forall i$, Set $\chi_i^* = \hat{\chi}_i^k$;
Output: $(\chi_i)_i^*$;

4.3 Application of L-IS to mobile robot localisation

4.3.1 Considered problem: robot localisation

To evaluate the performances of the developed approach, tests were conducted for a wheeled robot localisation problem, using the standard non-linear equations of the 2D differential drive car modeling the position $x_i \in \mathbb{R}^2$ and heading $\theta_i \in \mathbb{R}$ of the robot. The odometer velocity is integrated between two time steps to give a position shift $u_i \in \mathbb{R}^2$ and the angular shift $\omega_i \in \mathbb{R}$ is measured through (differential) odometry and/or gyroscopes. The discrete noisy model writes:

$$\begin{aligned} \theta_{i+1} &= \theta_i + \omega_i + w_i^\omega \\ x_{i+1} &= x_i + R(\theta_i)(u_i + w_i^x) \end{aligned} \tag{4.32}$$

where w_i^ω is the angular measurement error, w_i^x contains both the odometry and transversal shift errors, and $R(\theta) \in SO(2)$ denotes the planar rotation of angle θ. The vehicle also gets noisy position measurements (through e.g., GNSS) with standard deviation σ of the form

$$y_k = x_{t_k} + v_k, \qquad v_k \sim \mathcal{N}(\mathbf{0}, N_k = \sigma^2 I_2) \tag{4.33}$$

The state (and the increments) can be embedded in the matrix Lie group $SE(2)$, which is detailed in Section 2.2.3, using the homogeneous matrix representation

$$\chi_i =: \begin{pmatrix} R(\theta_i) & x_i \\ 0_{1\times 2} & 1 \end{pmatrix}, \; U_i =: \begin{pmatrix} R(\omega_i) & u_i \\ 0_{1\times 2} & 1 \end{pmatrix}$$

and letting $w_i = \begin{pmatrix} w_i^\omega & R(\omega_i)^T w_i^x \end{pmatrix}$ the stacked noise vector on the increments, we have using a first order expansion of the exponential map of $SE(2)$ (the increment noises are small if the time step is small)

$$\exp(w_i) \approx \begin{pmatrix} R(w_i^\omega) & R(\omega_i)^T w_i^x \\ 0_{1\times 2} & 1 \end{pmatrix}.$$

Thus, the dynamics and the observations respectively rewrite in the desired form:

$$\chi_{i+1} = \chi_i U_i \exp(w_i) = f_i(\chi_i)\exp(w_i) \tag{4.34}$$

$$y_k = \chi_{t_k} \begin{pmatrix} 0_{2\times 1} \\ 1 \end{pmatrix} + \begin{pmatrix} v_k \\ 0 \end{pmatrix} \tag{4.35}$$

This system is group-affine and its measurement covariance is rotation-invariant (i.e., isotropic), therefore (i) and (ii) of 4.2.3 hold. Moreover, we have, with the notations of Section 4.2 and according to Section 2.1.1,

$$g_i^L(\chi) = \Psi_{U_i^{-1}}(\chi) \tag{4.36}$$

4.3.2 Compared smoothing frameworks

In Sections 4.3.3 and 4.3.4, the proposed smoothing method is compared with two non-invariant parametrisations, but which account for the non linear structure of the state space, from respectively [58] and [53], and a standard linear parametrisation. The two former were developed for applications such as pose-SLAM or VIO, without directly considering navigation with absolute measurements. They were reimplemented in a batch setting, so that the focus was put exclusively on the parametrisation. Writing the propagation linearised factor in the general form $a_i + F_i^i \xi_i + F_i^{i+1} \xi_{i+1}$, the following jacobians were used, see [53,58] for the respective residuals' definitions:

	F_i^i	F_i^{i+1}
L-IS	$-Ad_{U_n^{-1}}$	Id
Lin	$\begin{bmatrix} -I_2 & -\hat{R}_i u_i \\ & -1 \end{bmatrix}$	Id
[58]	$\begin{bmatrix} -\Omega_i^T \hat{R}_i^T & J\Omega_i^T \hat{R}_i^T(\hat{x}_{i+1}-\hat{x}_i) \\ & -1 \end{bmatrix}$	$\begin{bmatrix} \Omega_i^T \hat{R}_i^T & \\ & 1 \end{bmatrix}$
[53]	$\begin{bmatrix} I_2 & J\hat{R}_i^T(\hat{x}_{i+1}-\hat{x}_i) \\ & 1 \end{bmatrix}$	$-\begin{bmatrix} \hat{R}_i^T \hat{R}_{i+1} & \\ & 1 \end{bmatrix}$

The observation factors are constructed using the following jacobians H_k, errors c_k and covariances:

	H_k	c_k	covariance
L-IS	$\begin{bmatrix} I_2 & 0 \end{bmatrix}$	$\hat{R}_{t_k}^T(y_k - \hat{x}_{t_k})$	$\hat{R}_{t_k}^T N_k \hat{R}_{t_k}$
Lin, [58]	$\begin{bmatrix} I_2 & 0 \end{bmatrix}$	$y_k - \hat{x}_{t_k}$	N_k
[53]	$\begin{bmatrix} \hat{R}_{t_k} & 0 \end{bmatrix}$	$y_k - \hat{x}_{t_k}$	N_k

4.3.3 Simulation results

The four smoothing methods of the previous section were compared. The vehicle is initialised at the true position but with a wrong heading of $-3\pi/4$, and an initial covariance of $diag(0.0025, 0.0025, (-3\pi/4)^2)$. The robot moves along a line at $7m/s$, where the odometry is polluted by a white noise of covariance $(0.1(m/s)^2, 0.01(rad/s)^2)$, and acquired at a rate of $10Hz$. Position measurements of covariance $0.01m^2$ are received every five steps. In this batch setting, (iii) of 4.2.3 does not hold for the invariant framework.

The iterations of the four optimisation schemes are displayed on Figure 4.2. As can be seen, the invariant parametrisation converges faster than all the other ones, and much more smoothly. This is important for a sliding window smoothing setting, as restricting the number of solver iterations at each step will have a weaker impact on the estimation.

4.3.4 Experimental setting

The same four methods are compared on experimental data obtained in an experiment conducted at the Centre for Robotics, MINES ParisTech. A small wheeled robot, called Wifibot and photographed in Figure 4.3, is equipped with independent odometers on the left and right wheels. We made it follow an arbitrary trajectory for 80 seconds. The OptiTrack motion capture system, a set of seven highly precise cameras, provides the ground truth with sub-millimeter precision at a rate of 120 Hz. This choice allows us to directly compare our results with other recent algorithms based on the invariant framework, namely the Left-UKF-LG, which had already been tested on these data [33], the Invariant EKF, and both non-invariant smoothing methods from Section 4.3.3. To keep the computational complexity acceptable, the smoothing is done in a sliding window, the oldest state being marginalised out when a new one is added. This choice makes (iii) hold in this case, leading to an information matrix fully independent from the estimate. The effect of the window's size is also studied here. The raw odometer inputs were provided to each method. Artificial position measurements are delivered by adding a Gaussian noise to the ground truth, at a rate of $1.35Hz$. For each setting, 100 Monte-Carlo simulations are run, each algorithm being initialised identically at each run. The trajectory followed by the robot, along with results of the various algorithms for one of the runs, are displayed in Figure 4.4. Three series of experiments were conducted, to study the evolution of the error with respect to the measurement covariance, the window size, and the initial error respectively. In the first case, the window size was fixed to 5, and the measurement covariance ranged from 10^{-1} to $10^{-5}m^2$, with the first state being initialised randomly or with a fixed value. In the former case, $(\bar{x}_0, \bar{\theta}_0) \sim \mathcal{N}((0,0,0), \mathrm{diag}(1/8, 1/8, (\pi/4)^2)$. In the latter, it was put at $\bar{x}_0 = (1/4, 1/4)$ and $\bar{\theta}_0 = \pi/4$, with the same covariance. In the second one, the measurement covariance was fixed to

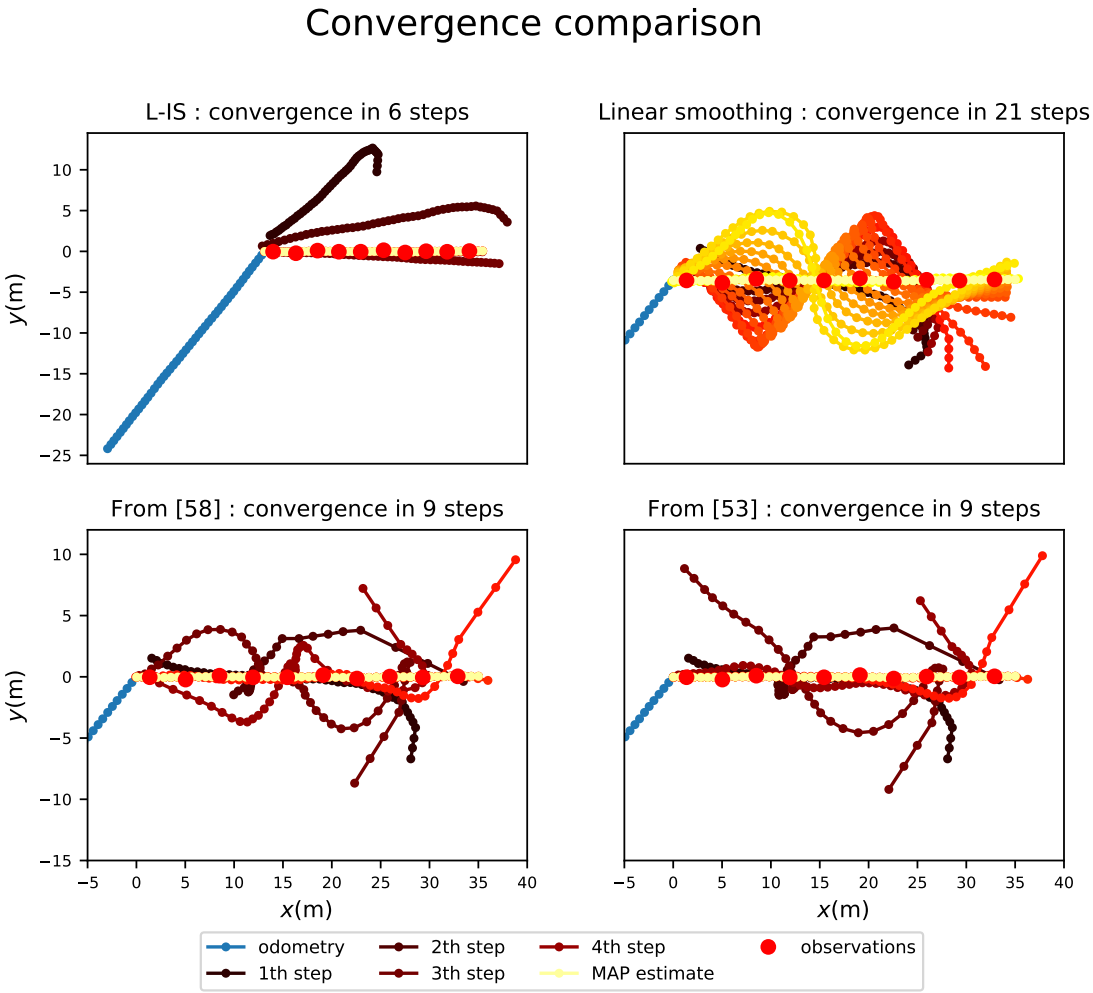


Figure 4.2: Comparison of the iterations of the smoothing algorithms based on four parametrisations: invariant (top-left), linear (top-right), from [58] (bottom-left), from [53] (bottom-right). Thanks to its being linearised independently from the estimate, the invariant version converges faster and in a more "sensible" fashion. However all have the same limit, the MAP estimate, i.e. find the minimum of the cost function.

$10^{-1}m^2$, while the window size ranged from 5 to 13. For the last one, the window size was set to 5, the measurement covariance to $10^{-5}m^2$. The initial estimate was put at $\bar{x}_0 = (1/4, 1/4)$ and $\bar{\theta}_0$ spanned $]-\pi, \pi[$ with only one Gauss-Newton per iteration, then it was fixed at $\bar{\theta}_0 = 9\pi/10$ with one and seven GN iterations. Figures 4.5 and 4.6 show the averaged Root Mean Square Error (RMSE) of the position and heading for the two first experiments. Figure 4.7 shows the RMSE of the heading for the last one.

Figure 4.3: Wifibot robot in its testing arena, surrounded by Optitrack cameras

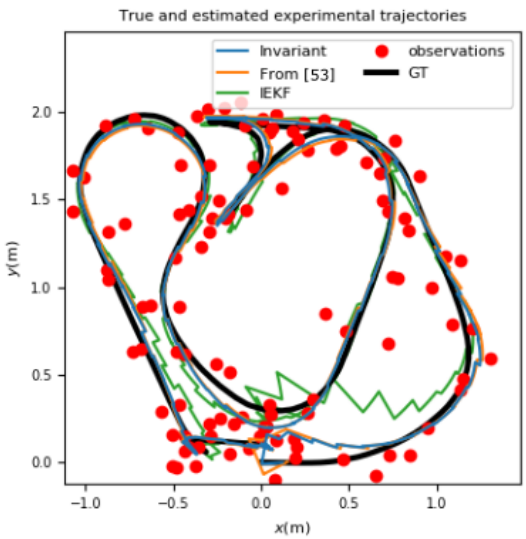


Figure 4.4: Trajectories of the wifibot robot, as captured by the OptiTrack system and estimated by three of the studied methods: invariant smoothing, smoothing based on [53], both using a sliding window of size 5, and the IEKF, with artificial noisy measurements obtained by adding simulated moderate noise - $\sigma = 0.1m$ - to the ground truth position, showed by the red dots.

Remark 13. *Average computation times of the full estimate* with one GN iteration per step*, on a laptop with Intel i5-5300 2.3 GHz CPU, are given in the following table.*

	Invariant	*From [53]*	*From [58]*	*IEKF*
Time (s)	*0.82*	*0.81*	*0.83*	*0.62*

It appears that the invariant smoothing, in a batch setting, does not induce extra computational load using exp *and* log*. Further gain could be expected with more GN iterations per step, and it could also prove more efficient in an incremental setting by taking advantage of the independence of the information matrix from the linearisation points to prefactor the information matrix [73].*

4.3.5 Experimental results

Figure 4.5 displays the evolution of the error with respect to the measurement noise covariance, when the initial state is fixed and when it is not. The first observation to be made is that, for a noise up to moderate ($10^{-2}m^2$), smoothing methods in general outperform the filtering ones, especially in terms of heading. This was expected, as the former explicitly contains the constraints between successive poses. If the initial state is fixed at $(1/4, 1/4, \pi/4)$, all smoothing approaches achieve almost identical results validating, in this case, the assumption made when deriving the propagation factor. However, invariant smoothing appeared to be the most robust to noise on the initial state, especially for low measurement covariance. This is further investigated in the third experiment.

The impact of the window size, illustrated in Figure 4.6, seems clear: increasing the size of the window is beneficial to all smoothing methods for strong noise. Still, none of them beats the UKF-LG in terms of position, highlighting its robustness. However, for lighter noise, the impact was negligible, as the odometry's uncertainty was quickly reached.

Finally, the invariant smoothing proved to be the more robust to initial heading errors such that $|\bar{\theta}_0| > \pi/2$, as shown by Figure 4.7, top, while both non invariant methods behaved identically. The two other graphs explain this difference: it appears that non-invariant smoothing methods fall into a local minimum where the estimate is completely turned around. Indeed, when increasing the number of GN iterations, the non-invariant experiments clearly form two clusters, one with the expected heading RMSE, and one with a RMSE close to π. Note that it is not a problem of the non-invariant methods, as they were devised for problems relative to the original position, i.e. without such an influence of the initial error. Moreover, this confirms the results of Section 4.3.3, as the invariant smoothing reaches the optimal RMSE quicker than the non-invariant ones, in the right cases.

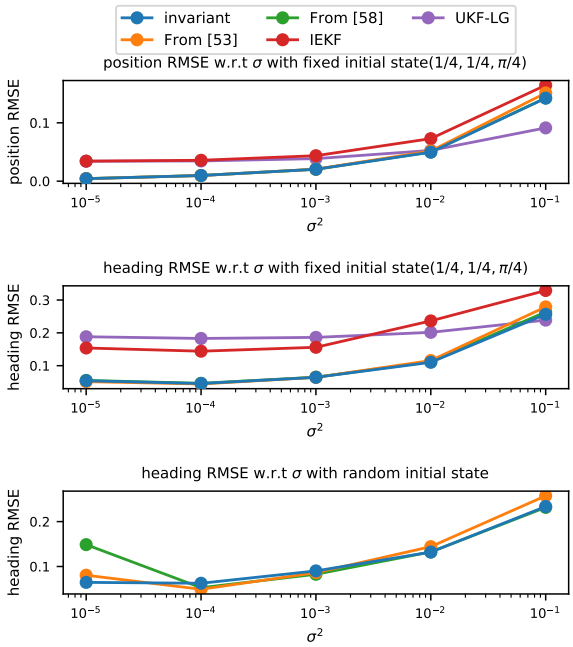


Figure 4.5: Top and middle: average position and heading RMSE for the five estimation methods w.r.t σ^2, with fixed initial state. Down: heading RMSE w.r.t. σ^2, with random initial state. The window size was set to 5 for the smoothing methods.

4.3.6 Discussion

For this experiment, the proposed method managed to combine the advantages of the smoothing approach and of the invariant framework. Indeed, on the first hand it provided better estimates than the filtering methods thanks to the smoothing paradigm. On the other hand, owing to the mathematical invariances leveraged by the algorithm, their computation are based on the measured odometry inputs, and not the estimated ones. This reduces the need for relinearisation, while ensuring more robustness to initial heading

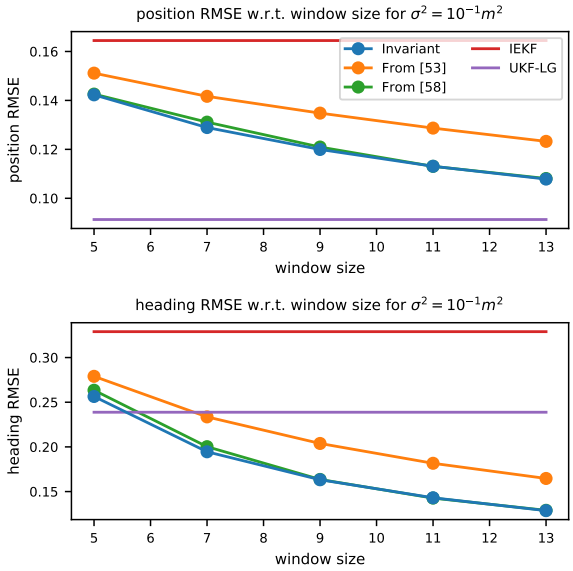


Figure 4.6: Average position and heading RMSE for the smoothing methods w.r.t. the window size, for $\sigma^2 = 10^{-1}m^2$. The RMSE of the filtering methods are shown as a comparison.

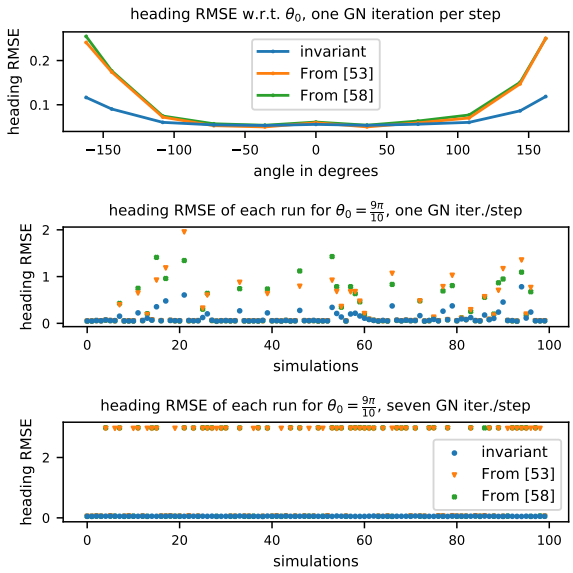


Figure 4.7: Heading RMSE for the smoothing methods w.r.t the initial angle $\bar{\theta}_0$ for $\sigma^2 = 10^{-5}m^2$ and a window size of 5. Top: $\bar{\theta}_0$ spans $]-\pi,\pi]$ with one GN iteration per step. Middle: RMSE for each run with $\bar{\theta}_0 = 9\pi/10$ with one GN iteration. Down: RMSE for each run with $\bar{\theta}_0 = 9\pi/10$ with seven GN iterations.

error, especially when non-invariant methods exhibited local minima with shifted heading estimates. However, this also means that the method will be dependent of the odometry's uncertainties.

4.3.7 Invariant Smoothing for biased inertial navigation: Extension for imperfect group-affine dynamics

As explained in Section 2.4.4, in practice inertial navigation needs to estimate the state's biases in addition to its attitude, speed and position. These biases are considered as additional parameters and treated linearly, i.e., the state lives in $SE_2(3) \times \mathbb{R}^3 \times \mathbb{R}^3$, see (2.47) and (2.48). $SE_2(3) \times \mathbb{R}^3 \times \mathbb{R}^3$ is also a Lie group, whose Lie algebra is $\mathfrak{se}_2(3) \times \mathbb{R}^3 \times \mathbb{R}^3$, and whose exponential, logarithm and adjoint operators are easily obtained from those of $SE_2(3)$

$$\exp_{SE_2(3)\times\mathbb{R}^3\times\mathbb{R}^3}(\omega, w, u, b^g, b^a) = \exp_{SE_2(3)}(\omega, w, u), b^g, b^a \tag{4.37}$$

$$\log_{SE_2(3)\times\mathbb{R}^3\times\mathbb{R}^3}(R, v, x, b^g, b^a) = \log_{SE_2(3)}(R, v, x), b^g, b^a \tag{4.38}$$

$$Ad_{R,v,x,b^g,b^a} = \begin{pmatrix} Ad_{R,v,x} & \\ & I_6 \end{pmatrix} \tag{4.39}$$

In turn, the state's dynamics are only *almost* group-affine, as (2.19) is not satisfied for varying biases. Nonetheless, an extended version of Invariant Smoothing can be derived. This was done for attitude and gyro bias estimation in [102]. The main difference is that, in this case, the jacobians F_i^L intervening in (4.23) will depend on the increments and the estimated biases, see Section 4.4.4 for the detailed formulas. However, as high-grade IMUs have stable biases, biased inertial navigation still stays close to the ideal group-affine case, as advocated by the numerical results reported thereafter and in Chapter 6.

4.4 IMU preintegration in the context of invariant smoothing

This section builds on the embedding of inertial navigation in a Lie group setting, introduced in Section 2.2.5, to get new insights on preintegrated IMU measurements. Indeed, it was recently shown in [17,18] that it allows retrieving the preintegrated factors of [53] with a highly reduced complexity. Then, links and differences, both theoretical and practical, between the standard method and the one coming from the invariant framework are evaluated.

4.4.1 Classical preintegration theory

In the standard smoothing framework the residuals and jacobians of the non-linear factors need to be rederived at each iteration, in particular the propagation ones

$$\|f(\chi_i) \boxminus \chi_{i+1}\|^2_{Q_i}.$$

In general though, computing the propagated state $f(\chi_i)$ directly depends on the starting point, χ_i. With the difference of clock frequencies between IMUs and other sensors, this would mean reintegrating many increments for each factor, which has no closed-form, thus ending up with a very inefficient smoothing process. However, it was realised that this caveat can be avoided by slightly changing (at first sight) the non-linear factor we seek to estimate. Indeed, the inertial propagation writes in general

$$R_{i+1} = R_i \widetilde{\Delta R}_i \tag{4.40}$$

$$v_{i+1} = v_i + g\Delta t + R_i \widetilde{\Delta v}_i \tag{4.41}$$

$$x_{i+1} = x_i + v_i \Delta t + 1/2 g \Delta t^2 + R_i \widetilde{\Delta x}_i, \tag{4.42}$$

where the terms decorated with a tilde depend solely *on the increments*, thus getting named *preintegrated* measurements [81]. Therefore, it was suggested to build the residuals with the preintegrated measurements, and their estimates, which are given by

$$\begin{aligned} \widehat{\Delta R}_i &= \hat{R}_i^T \hat{R}_{i+1} \\ \widehat{\Delta v}_i &= \hat{R}_i^T (\hat{v}_{i+1} - \hat{v}_i - g\Delta t) \\ \widehat{\Delta x}_i &= \hat{R}_i^T (\hat{x}_{i+1} - \hat{x}_i - \hat{v}_i \Delta t - 1/2 g \Delta t^2) \end{aligned} \tag{4.43}$$

4.4.2 Preintegration as a group-affine property: summary of [17,18]

Preintegration is a useful feature for smoothing systems. Indeed, it allows reducing the number of propagation factors while giving a faster way to compute the propagation function [81]. It turns out that, as shown in [18], it is a natural property of group-affine dynamics thanks to the following result.

Proposition 5 (from [18])**.** *Consider a discrete group-affine dynamics $\chi_{i+1} = \phi_i(\chi_i) \cdot a_i$. Then two consecutive propagations can be made into one discrete step of another group-affine dynamics:*

$$\chi_{i+2} = \phi_{i+1} \circ \phi_i(\chi_i) \cdot (\phi_{i+1}(a_i) a_{i+1}) \tag{4.44}$$

Thus, the system can be preintegrated on k steps recursively as:

$$\chi_{i+k} = \phi_{i+k-1:i}(\chi_i) \cdot b_{i+k-1:i} \tag{4.45}$$

$$\phi_{i+k-1:i} = \prod_{j=i}^{i+k-1} \phi_j \tag{4.46}$$

$$b_{i+k-1:i} = \prod_{j=i}^{i+k-1} \phi_{+k-1:j+1} a_j \tag{4.47}$$

where $\phi_{k:j} = Id$ if $j > k$.

Application to inertial navigation

This useful result on group-affine dynamics has a direct application in inertial navigation. Indeed, recall from Section 2.2.5 that unbiased inertial navigation is made group-affine when navigation triplets $\chi = (R, v, p)$ are embedded into the $SE_2(3)$ Lie group. Then, the corresponding product operation is given by

$$(R_1, v_1, p_1) \cdot (R_2, v_2, p_2) = (R_1 R_2, v_1 + R_1 v_2, p_1 + R_1 p_2) \tag{4.48}$$

Proposition 5 can be applied along with Theorem 3 to rederive the preintegrated measurements. However in this case it is simpler and faster to simply prove it by induction using (2.45), which gave $f_i(\chi_i) = \gamma \phi(\chi_i) \upsilon_i$:

Proposition 6 (from [17,18])**.** *Let $m > 0$, $\upsilon_1, \cdots, \upsilon_m$ be increments. There exist Γ_m, Υ_m, where Γ_m depends only on the gravity and Υ_m only on the increments and the biases, such that*

$$f_m \circ \cdots \circ f_1(\chi) = \Gamma_m \ \phi^m(\chi) \ \Upsilon_m \tag{4.49}$$

Uncertainty propagation

This preintegration theory also eases the description of uncertainty propagation, by proposing the uncertainty representation

$$\tilde{u} = u \cdot \exp(w), \tag{4.50}$$

where $\tilde{u}$ encodes the true increments, u is the IMU measurement, and w a white noise on $\mathfrak{se}(3)$. [17] gives a simple formula based on a exact non-linear noise propagation and a first-order expansion using the Baker Campbell Haussdorff (BCH) formula [6], which writes

$$\chi_{i+2} \approx \alpha_{i+1} \phi^2(\chi_i) \phi(u_i) u_{i+1} \exp(Ad_{u_{i+1}^{-1}} F w_i + w_{i+1}), \tag{4.51}$$

where F denotes here the matrix associated with function ϕ, that is, transformation $(\delta_R, \delta_v, \delta_x) \mapsto (\delta_R, \delta_v, \delta_x + \Delta t \delta_v)$. This noise propagation equation does not exactly match that given in Appendix A of [53], because of an additional rotation term due to the retained Lie group multiplication, and the impact of the noise v_w on the velocity of the estimate is through $R_u v_w$.

Taking the biases into account: exponential update of the preintegrated measurements [17]

To account for biases correction in the preintegration factors, first-order updates are used, as in [53]. As seen in (4.49), the Jacobians of f and Υ_m with respect to the biases coincide. We can thus update Υ_m with the precomputed $\partial_b f$, with no computational overload, and obtain the new residual and jacobian which should be used for estimation. Thus we have, following the methodology of [17] the following formula for an update ξ:

$$\Upsilon_m \leftarrow \Upsilon_m \exp(\partial_b f \xi) \tag{4.52}$$

Remark 14. *This update strategy neglects the changes of $\partial_b f$ and the covariance of the residual, which also depend on the biases. Therefore, one should be careful with large bias updates, which sometimes require re-integration.*

4.4.3 Defining the preintegrated IMU factors

The result of [17,18] highlights a simple and understandable description of the preintegrated inertial measurements, as we get for all m

$$(\Gamma_m F^m(\chi))^{-1} f_m \circ \cdots \circ f_1(\chi) = \Upsilon_m \tag{4.53}$$

In fact, (4.53) encodes the preintegrated inertial measurements (4.43),given in [53] and currently used in inertial navigation systems, such as GTSAM [46]. Indeed, the velocity part of the left-hand side writes $R_i^T(v_{i+1} - v_i - g\Delta t)$, which is exactly the preintegrated velocity measurement defined in their work. The same goes for the position and rotation parts. In itself, (4.53) is the starting point of the definition of preintegrated IMU factors. Here, the similarities and differences of these factors between the frameworks of Invariant Smoothing [37] and that of [53] are all derived from this very equation.

In the following, let χ_i and χ_{i+1} denote two successive states, Γ_i, Υ_i, φ such that

$$\chi_{i+1} = f(\chi_i) = \Gamma_i \varphi(\chi_i) \Upsilon_i \tag{4.54}$$

Under the Gaussian and conditional independence assumptions, defining the factors boils down to isolating the noise in each equation of the system and linearising accordingly [46]. Therefore, the definition of the residual comes from the uncertainty model that is chosen for the system. Using the general notations, one wants to define them accordingly, depending on the true increments $\tilde{\Upsilon}_i$:

$$\tilde{\Upsilon}_i = \Upsilon_i \boxplus w_\upsilon \tag{4.55}$$

$$r_{IMU} = \|(\Gamma_i \varphi(\chi_i))^{-1} \chi_{i+1} \boxminus \Upsilon_i\|_Q^2 \tag{4.56}$$

where Q is the covariance associated to the factor, which was given in Section 4.4.2.

Theoretical differences between I-S and standard preintegration

In the Invariant Smoothing framework, the noise model and the residual definition are naturally given by the $\boxplus_L$ and $\boxminus_L$ operators given in Section 4.1.5, that is

$$r_{IMU}^{IS} = \|\log(\Upsilon_i^{-1}(\Gamma_i \varphi(\chi_i))^{-1} \chi_{i+1})\|_{Q_{IS}}^2 = \|\log(f(\chi_i)^{-1} \chi_{i+1})\|_{Q_{IS}}^2 \tag{4.57}$$

where f is the propagation between steps i and $i+1$.

On the contrary, as introduced in Section 4.4.1, the standard preintegrated IMU residual is defined "line by line" from (4.53), i.e.,

$$r_{IMU}^F = \left\| \begin{matrix} \log_{SO(3)}(R^T_{(\Gamma_i \varphi(\chi_i))^{-1}\chi_{i+1}} R_{\Upsilon_i}) \\ v_{(\Gamma_i \varphi(\chi_i))^{-1}\chi_{i+1}} - v_{\Upsilon_i} \\ p_{(\Gamma_i \varphi(\chi_i))^{-1}\chi_{i+1}} - p_{\Upsilon_i} \end{matrix} \right\|_{Q_F}^2 \tag{4.58}$$

where $R_{(\cdot)}, v_{(\cdot)}, p_{(\cdot)}$ denote the rotational, velocity and position parts of the state respectively.

Practical differences

In fact, both definitions are quite close. The major difference comes from the use of the Lie group logarithm in (4.57). The term inside the logarithm does not exactly match that of (4.58). Indeed, looking at the velocity part for instance, it writes $R_{\Upsilon_i}^T(v_{(\Gamma_i\varphi(\chi_i))^{-1}\chi_{i+1}} - v_{\Upsilon_i})$. That is, there is an additional rotation involved. Since we consider Mahalanobis distances and not the Euclidian one, this rotation does play a role in the value of the residual. However, this will be counterbalanced in the way the jacobians and covariances are defined, therefore it has no impact in practice. Thus, if the IMU used are precise enough to neglect the effect of the logarithm, we see that both residuals almost coincide.

4.4.4 Jacobians computation

States' update

Invariant Smoothing uses the Lie group exponential through left multiplication, while in GTSAM the considered retraction is in the spirit of (4.48). For a state $\chi = (R, v, p)$ and an update $\xi = (\delta_R, \delta_v, \delta_x)$, we thus have

- For Invariant Smoothing: $\chi \boxplus \xi = \chi \cdot \exp(\xi)$
- For GTSAM: $\chi \boxplus \xi = (R\exp_{SO(3)}(\delta_R), v + R\delta_v, x + R\delta_x)$

Remark 15. *In the published work of [53], the retraction updates the velocity linearly. Since the only difference is the use of an invertible matrix, this is in fact equivalent. We thus do not distinguish both retractions in the following.*

Linearisation of the preintegrated IMU residual

There are actually two ways of defining jacobians in Invariant smoothing. In the case of inertial dynamics, it turns out that one involves solely the increments and biases, and is used by L-IS, while the other is computed from the estimates and the gravity only, and is used by [53]. Let Δ_{IMU} be the error used in the Invariant Smoothing residual, i.e., such that (4.57) is given by $\|\Delta_{UMI}\|^2_{Q_{IS}}$. Rewriting it with a perturbation of the estimates $\chi = \hat{\chi}\exp(\xi)$ leads to

$$\Delta_{IMU} = \Upsilon_i^{-1}\exp(-F\xi_i)\varphi(\hat{\chi})^{-1}\Gamma_i^{-1}\hat{\chi}_{i+1}\exp(\xi_{i+1}) \tag{4.59}$$

To regroup and obtain only first-order terms, the perturbations must be isolated, which can be done in two ways:

$$\Delta_{IMU} = \exp(-Ad_{\Upsilon_i^{-1}}F\xi_i)f(\hat{\chi}_i)^{-1}\hat{\chi}_{i+1}\exp(\xi_{i+1}) \tag{4.60}$$

$$= f(\hat{\chi}_i)^{-1}\hat{\chi}_{i+1}\exp(-Ad_{\hat{\chi}_{i+1}^{-1}(\Gamma_i\varphi(\hat{\chi}))}F\xi_i)\exp(\xi_{i+1}), \tag{4.61}$$

In particular, (4.61) gives jacobians which are equivalent to those of [53] up to rotations, which are accounted for in the definition of the residual and the uncertainty propagation. Therefore, it appears that, as far as the jacobians are concerned, the invariant framework leads to two possible choices: one based on the *measured* increments, the other relying on the *estimated* ones. This is generalised for all group-affine dynamics in Section 4.5.4, and the impact of this choice in practice is experimentally evaluated in Section 6.2.

Taking the biases into account: expanding the jacobians

Biases are added to the state and treated as linear parameters. That is, the state (R, v, p, b^g, b^a) lives on $SE_2(3) \times \mathbb{R}^3 \times \mathbb{R}^3$. The new jacobian for invariant smoothing writes, for a *single* inertial increment

$$\begin{pmatrix} Ad_{\upsilon^{-1}}F & dtJ_{\text{bias}} \\ & I \end{pmatrix}, \quad J_{\text{bias}} = \begin{pmatrix} Z & \\ & \tilde{\Omega}^T \end{pmatrix}, \tag{4.62}$$

where $Z = d_{\exp_{SO(3)}}(dt(\omega - b^g)$ is the differential of the exponential on $SO(3)$ at $dt(\omega - b^g)$. This jacobian is not independent from the state anymore, as the biases are involved in it. However, it still does not depend on the navigation state.

4.4.5 Summary of the differences between preintegrated IMU factors

Thanks to the preceding sections, it appears that the preintegrated IMU factors of Invariant Smoothing and the standard ones given in [53] show three main differences in general, and two in practice:

- The definition of the residuals slightly differs, but are close especially in the context of precise IMUs
- The linearised system varies depending on the chosen parametrisation, using either measured increments, or estimated ones
- Invariant smoothing relies on the exponential on $SE_2(3)$, while [53] retracts on $SO(3) \times \mathbb{R}^3 \times \mathbb{R}^3$

4.4.6 Simulation results in inertial navigation: L-IS vs IEKF

The impact of L-IS on Inertia-GPS fusion was first evaluated in an alignment scenario using an industrial simulator. This setup is similar to the one studied in Section 4.3, with a fusion of propagation and global observations. As outlined in Section 2.4.4, biased inertial dynamics are not group-affine anymore, if the biases are not fixed. This fact should be in favor of L-IS. Indeed, the biases are the only part of the state involved in the jacobians, and are fairly constant inside a sliding window. Therefore, relinearising them at each iteration should bring the problem closer to the ideal case of group-affine dynamics. This is evaluated here. The estimate was initialised at the first GPS position, with roughly the right pitch and roll, zero velocity, zero biases, and a random heading. The initial uncertainty was set accordingly. Although the exact parameters of the IMU cannot be made public, the accelerometer's bias can be fairly large (around $0.2m/s^2$), but stable:

The GPS measurements are taken with an isotropic uncertainty of $3m$. For each initial heading, we assess the estimators' ability to converge to the right value, while staying inside the $3-\sigma$ bounds of their covariance. Since the jacobians only rely on the estimated biases, the IEKF and L-IS share almost the same covariance for the latest state, therefore we only display that of filtering. The advantage of L-IS on the IEKF clearly appears on Figure 4.8, which represents the evolution of the heading and bias errors of both method, initialised with an error of $90°$ and biases at zero, along with the $3-\sigma$ bounds of the IEKF. Here, L-IS is taken in a batch setting, i.e., no state is marginalised, leading to a final estimate containing 75 states. Both methods have a stable heading estimate until the system starts moving forward. Then, while L-IS seamlessly converges to an acceptable, and consistent, heading value, the IEKF gets stuck in another local minimum, from which it cannot escape due to an estimated uncertainty too low. This stems from the high biases values which come and counter the theoretical invariant properties: distinguish heading error from accelerometers' bias error at the beginning is hard, and the IEKF can "fall on the wrong side". Here, it wrongly estimates the bias on the Y-axis, and does not recover from it. L-IS, on the other hand, carries out a consistent estimation. Its X-axis bias slightly goes off the $3-\sigma$ bound, but only for very short periods, and never far enough to harm the estimation. The reason is that, after each update, it maintains a roughly constant bias along its estimated trajectory, and updates its jacobians accordingly. Thus it can much more easily carry out the distinction between bias and heading error, and in turn come closer to the ideal invariant case.

This first result raises two questions: what about the other initial heading errors, and what if the window size is constrained? Answers are given by Figure 4.9, which shows the heading error after 13000 time steps of L-IS with growing sliding window sizes for initial angle error spanning $[-\pi, \pi]$. The smoothing algorithms are denoted by L-IS-(window size). As can be seen, for windows small enough, L-IS alternates between failure and consistent estimation, but once it exceeds a certain value, it always succeeds. There thus seems to be a critical window width from which the estimation always succeeds. Moreover, here this size allows marginalising the initial state only after the heading has converged.

4.5 Other Invariant Smoothings: Right-Invariant and Hybrid Invariant Smoothing

4.5.1 Using the right-invariant parametrisation: Right-Invariant Smoothing

As explained in [14], the right-invariant error is particularly well-suited for local and relative measurements, e.g. odometers, relative poses, known features, which are modeled by right group actions (2.4).

Similarly to the R-IEKF in Section 3.2, the right-invariant parametrisation of the search parameter

$$\chi = \exp(\xi)\hat{\chi} \tag{4.63}$$

leads to Right-Invariant Smoothing (R-IS). Its main equations are given here, their derivation has been moved to Appendix A.

4.5.2 Considered systems

Logically, the systems of interest for R-IS do not involve left group action observations as (4.20), but rather right group actions

$$z_k = b_k \star \chi \exp_Y(v_k) \qquad v_k \sim \mathcal{N}(0, N_k) \tag{4.64}$$

Remark 16. *One could argue that, for the sake of symmetry, the noise should be considered as entering "by the left", just like the search parameter. While the question of the noise model is an interesting and hard one, we chose to stick with this definition for the following reasons*

- *This has no impact for vectorial measurements*
- *This is closer to the physical reality of the sensors. Relative measurements will be, in general, corrupted by noise entering the local sensor (camera, odometer, etc.), and should therefore be modeled as a local perturbation. When the state is represented by the transformation between the global frame and its local frame, say (R, x), this corresponds to noise entering "by the right".*

4.5.3 Linearised system of R-IS

The derivation of the factors is very similar to those of the left-invariant case of Section 4.2, and its details have been moved to Appendix A. The resulting linear least-squares that appears in R-IS is

$$\begin{aligned}\Xi^* = (\xi_i^*)_i = &\operatorname*{argmin}_{(\xi_i)_i} \|p_0 + J_0^R \xi_0\|^2_{Ad_{\hat{\chi}_0} P_0 Ad^T_{\hat{\chi}_0}} \\ &+ \sum_i \|a_i^R - F_i^R \xi_i + \xi_{i+1}\|^2_{Ad_{\hat{\chi}_{i+1}} Q_i Ad^T_{\hat{\chi}_{i+1}}} + \sum_k \|c_k - H_k \Xi\|^2_{\hat{N}_k}\end{aligned} \tag{4.65}$$

where

$$BCH(\xi, p_0) = p_0 + J_0^R \xi + o(\|\xi\|^2) \tag{4.66}$$

$$a_i^R = \log(\hat{\chi}_{i+1} f(\hat{\chi}_i)^{-1}) \tag{4.67}$$

$$c_k = \log_Y(z_k \star \hat{\chi}^{-1}) \tag{4.68}$$

$$\log_Y(b_k \star \exp(\Xi)) = H_k \Xi + o(\|\Xi\|^2), \tag{4.69}$$

F_i^R corresponds to (2.28) for the right-invariant error, and $\hat{N}_k$ depends on the observation model.

One of the main differences with L-IS is the fact that the covariance of the propagation factor is corrected by $Ad_{\hat{\chi}_{i+1}}$, and the prior covariance by $Ad_{\hat{\chi}_0}$. Thus, approximate independence of the information matrix to the estimate which had been obtained with the left-invariant error is lost here. However, we argue that this is balanced by the consistency induced by this parametrisation, as shown in Section 5.2.4.

This leads to Algorithm 2 for the right-invariant based parametrisation.

Factors for relative state measurements

A particularly important type of observations modeled by right group actions are relative state measurements:

$$(\chi_1, \chi_2) \star \alpha = \chi_2^{-1} \alpha \chi_1,$$

where α is usually taken as the identity. For these, the measurement factor for R-IS is given by

$$\| -\xi_1 + Ad_{\alpha^{-1}} \xi_2 + \log(\alpha^{-1}((\hat{\chi}_1, \hat{\chi}_2)^{-1} \star z))\|_{Ad_{\hat{\chi}_1} N_k Ad^T_{\hat{\chi}_1}} \tag{4.70}$$

Algorithm 2: Smoothing for a group-affine system with a right-invariant based parametrisation

Input: $(\bar{\chi})_{1\leq i\leq n}, \mathbf{P}_0, (f_i)_i, (Q_i)_i, (z_k)_k, (N_k)_k$;
Initialisation
1 $\forall i$, Set $\hat{\chi}_i^0 = \bar{\chi}_i$
Until convergence do
2 Linearize around $(\hat{\chi}_i^k)_i$ according to (4.65) ;
3 Solve for $(\xi_i)_i$;
4 Update: $\forall i,\ \hat{\chi}_i^{k+1} := \exp(\xi_i)\hat{\chi}_i^k$,
5 $\forall i$, Set $\chi_i^* = \hat{\chi}_i^k$;
Output: $(\chi_i)_i^*$;

Usually, only partial relative measurements are available (for instance relative pose in inertial navigation). This is still a group action, as shown by Proposition 1. In that case, let π be the projector on the reduced group on which the relative measurement takes place (e.g. $SE(3)$ for relative poses), and Π its tangent map. Then the group action writes $(\chi_1, \chi_2) \star \alpha = \pi(\chi_2)^{-1}\alpha\pi(\chi_1)$, and the factor becomes

$$\| -\Pi\xi_1 + Ad_{\alpha^{-1}}\Pi\xi_2 + \log(\alpha^{-1}((\hat{\chi}_1, \hat{\chi}_1)^{-1} \star z))\|_{Ad_{\pi(\hat{\chi}_1)}N_k Ad^T_{\pi(\hat{\chi}_1)}} \tag{4.71}$$

4.5.4 An alternative to L-IS: Hybrid Invariant Smoothing

This section generalises the realisation made in Section 4.4.4, offering an alternative way of deriving propagation factors for left-invariant error based smoothing. Consider a general group-affine dynamics in the form of (2.26):

$$f(a) = \Theta(a) \cdot \kappa \tag{4.72}$$

Let Φ be the matrix such that $\Theta \circ \exp = \exp \circ \Phi$. Then we have,

$$f(\chi_i)^{-1}\chi_{i+1} = (\Theta(\hat{\chi}_i)\exp(\Phi\xi_i)\kappa)^{-1}\hat{\chi}_{i+1}\exp(\xi_{i+1}) \tag{4.73}$$
$$= \kappa^{-1}\exp(-\Phi\xi_i)\Theta(\hat{\chi}_i)^{-1}\hat{\chi}_{i+1}\exp(\xi_{i+1}) \tag{4.74}$$

To be able to linearise, only first-order terms must appear. There are actually two ways of regrouping these terms, which boil down to "choosing a side". One leads to L-IS, not the other:

$$f(\chi_i)^{-1}\chi_{i+1} = \exp(-Ad_{\kappa^{-1}}\Phi\xi_i)\kappa^{-1}\Theta(\hat{\chi}_i)^{-1}\hat{\chi}_{i+1}\exp(\xi_{i+1}) \qquad \text{L-IS} \tag{4.75}$$
$$= \kappa^{-1}\Theta(\hat{\chi}_i)^{-1}\hat{\chi}_{i+1}\exp(-Ad_{\hat{\chi}_{i+1}^{-1}\Theta(\hat{\chi}_i)}\Phi\xi_i)\exp(\xi_{i+1}) \qquad \text{Hybrid-IS} \tag{4.76}$$

When $\hat{\chi}_{i+1} = f(\hat{\chi}_i)$, both jacobians coincide. In fact, L-IS relies on the *measured* value of κ, while the alternative involves the *estimated* one, as was already noted for inertial measurements in (4.60), (4.61). Therefore, it appears that there is a third Invariant Smoothing method, close to L-IS but relying more on the estimate than the measurements, which is called Hybrid Invariant Smoothing. The impact of the differences between these three methods in real estimation problems are assessed in Section 6.2 and 6.3.2.

4.6 Conclusion

This chapter presented the extension of the invariant Kalman filtering framework to smoothing. As with the IEKF, two main smoothing algorithms, L-IS and R-IS, were proposed, depending on the measurement models at stake. A third one, an alternative to L-IS, was also brought to light. For L-IS, this resulted in Jacobians being independent from the current estimate, thus ensuring sound behavior of the estimate. To validate the results, this was applied to a robot localisation problem, in simulations, and using experimental data. On the one hand, faster convergence to the MAP estimate than the other smoothing approaches was achieved in simulation. On the other hand, the invariant smoothing provided estimates similar to that of other smoothing methods, while being as fast to compute, and better than invariant filtering methods. It also appeared more robust to errors on the initial state, avoiding local minima

of the non-invariant methods. For imperfect group-affine dynamics, such as biased inertial navigation, L-IS proved superior to the IEKF, which had a hard time distinguishing between bias and heading errors, thanks to its being able to relinearise past states using the newly computed bias values. The independence of the information matrix from the current estimate is lost in R-IS, however this will ensure consistency properties, studied in Chapter 5.

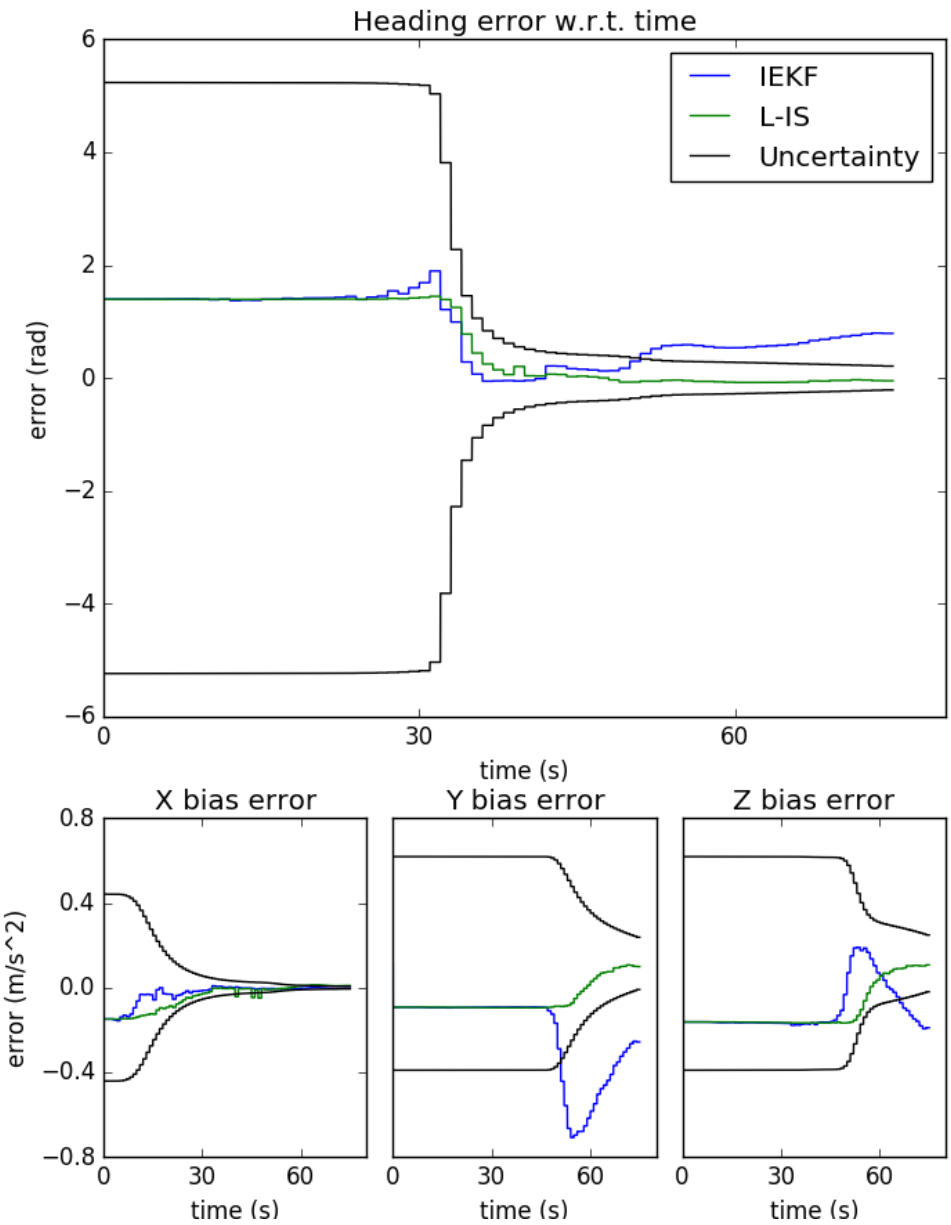


Figure 4.8: Comparison of the IEKF and batch L-IS on an alignment problem using inertia and GNSS information. At first the system stays still, with unknown heading, and starts moving forward after a minute. GPS positions are obtained every 2 second. We can see that the IEKF converges to a wrong solution, where the yaw and biases have been wrongly corrected, and turns inconsistent, as its error goes out of the $3-\sigma$ interval given by its covariance, which leads to subsequent estimation problems. In the meantime, L-IS manages to converge to a consistent estimate thanks to its relinearisation of previous states.

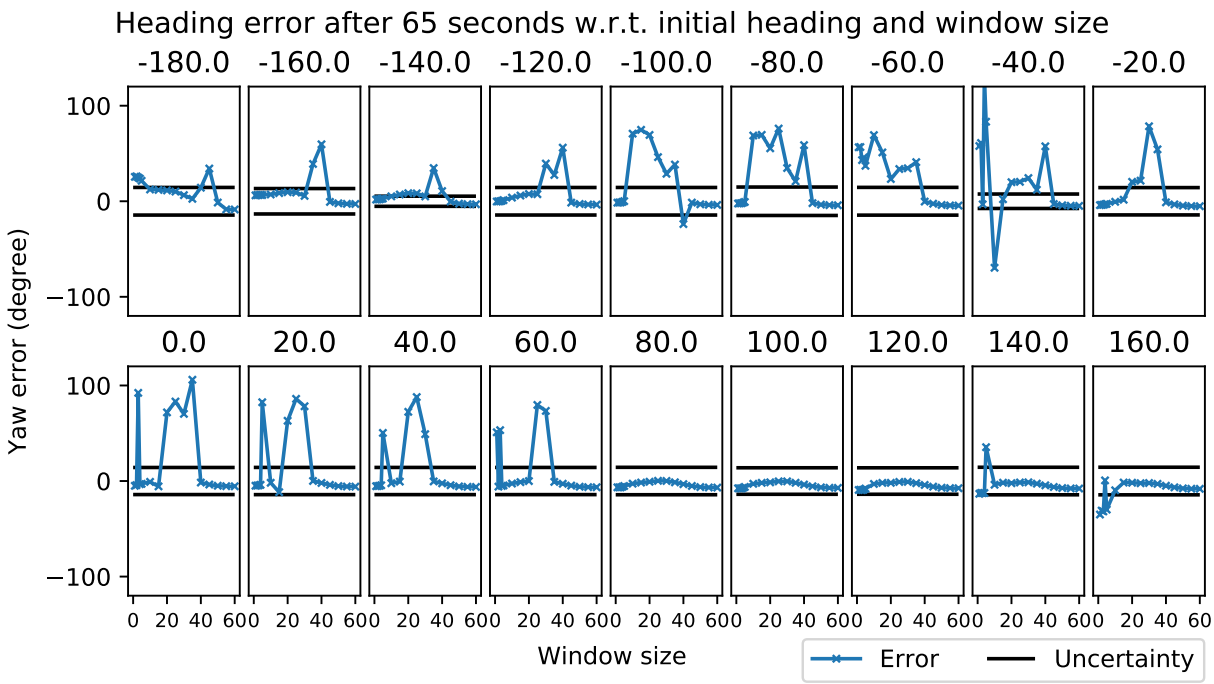


Figure 4.9: Heading error after $65s$ w.r.t. the sliding window width for initial yaw errors spanning $[-\pi, \pi]$, and compared to the estimated $3-\sigma$ bound. Once the window is large enough, the estimation always stays consistent.

Chapter 5

Invariant smoothing in the face of geometric constraints and unobservability related inconsistency

Highlights

- This chapter digs into the properties of Invariant Smoothing concerning geometrical constraints and the systems having symmetries.
- The results on deterministic dynamics of Chapter 3 are extended to Invariant Smoothing. The impact on low noise cases is outlined in simulation.
- Observability is studied from the point of view of smoothing updates. R-IS is shown to be consistent in local navigation (but not L-IS). It also brings to light the unexpected role of the prior covariance in respecting the unobservable directions.

Points marquants

- Ce chapitre explore les propriétés du Lissage Invariant en rapport avec les contraintes géométriques et les systèmes possédant des symétries.
- Les résultats sur les dynamiques déterministes du Chapitre 3 sont étendues au Lissage Invariant. L'impact sur les cas de faible bruit est illustré en simulation.
- L'observabilité est étudiée du point de vue des mises à jour du lissage. R-IS est prouvé être cohérent pour la navigation locale (mais pas L-IS). Cette étude met aussi en évidence le rôle inattendu de la covariance du prior dans le respect des directions inobservables.

Introduction (En/Fr)

This chapter, which notably presents the results of an article that was submitted to IROS 2020, further studies the properties of the developed theory of invariant smoothing and its application to navigation. Can we expect similar properties as those of the IEKF? And only them?

First, the theoretical result on geometrical constraints for the IEKF from Section 3.4 is extended to L-IS: if the trajectory lies in a known affine subgroup of the state space, then the iterations of L-IS will respect this. Once again, the exponential update is key to obtain this result. Then, the notion of the (un)observability of a system, which plays a key role in local navigation, is revisited from a smoothing point of view in Section 5.2. In particular, R-IS is shown to be consistent in pure and inertia-aided pose-SLAM, and the role of the covariance prior, usually not considered, is outlined in local inertial navigation.

Ce chapitre présente, entre autres, les résultats d'un article soumis à IROS 2020, portant sur l'étude des propriétés de la théorie du lissage invariant et son application à la navigation. Peut-on s'attendre à des propriétés similaires à celle de l'IEKF ? Et seulement celles-ci ?

Tout d'abord, le résultat théorique concernant les contraintes géométriques et l'IEKF de la Section 3.4 est étendu au L-IS : si la trajectoire se situe dans un sous-groupe affine connu de l'espace d'état, alors les itérations du L-IS l'y maintiendront. A nouveau, la mise à jour exponentielle est essentielle pour obtenir ce résultat. Puis, la notion d'(in)observabilité d'un système, cruciale pour la navigation locale, est revisitée du point de vue du lissage dans la Section 5.2. En particulier, R-IS se révèle cohérent pour le pose-SLAM avec et sans inertie, et le rôle de la covariance du prior, habituellement non considérée, est souligné pour la navigation inertielle locale.

5.1 Invariant Smoothing also respects geometric constraints

Chapter 3 showed that the IEKF is able, in a noise free dynamics setting, to preserve side information in the form of geometric constraints. While this is obviously an ideal case, it helped understanding why invariant filtering outperforms the standard EKF when process noise is very small in simulation [14], and advocated the use of the IEKF when using precise sensors. It is of particular interest to know what it becomes in the context of smoothing. Indeed, from a theoretical point of view, two questions arise: Does standard smoothing, thanks to its iterations and relinearisations, solve the problems the EKF had in the ideal case? Does (L-)IS exhibit properties similar to that of the IEKF? From a practical point of view, there is indeed the question of dealing with precise sensors, but also that of the imperfect group-affine dynamics. As exhibited in Section 4.4.6, the IEKF suffered from large biases, while L-IS seamlessly handled them. Can it be explained by a respect of some constraints?

5.1.1 A 2D localisation illustrative example

The property of L-IS presented in this section is first illustrated on a simple example. Simulations were conducted using the same setup as Section 3.5.1 and 4.3.3: a 2D localisation problem based on perfect odometry and GPS-like position measurements. The mobile is supposed to follow a straight line, with unknown initial heading. The state $\chi = (\theta, x)$ and the measurements y_k are modeled as

$$\begin{aligned} \theta_{i+1} &= \theta_i + w_i^\omega, & w_i^\omega &\sim \mathcal{N}(0, \sigma_\omega^2) \\ x_{i+1} &= x_i + R(\theta_i)(u + w_i^x), & w_i^x &\sim \mathcal{N}(0, \sigma_x^2) \\ y_k &= x_{t_k} + v_k, & v_k &\sim \mathcal{N}(\mathbf{0}, N_k = \sigma^2 I_2) \end{aligned}$$

where u is a given position shift, $R(\theta) \in SO(2)$ denotes the planar rotation of angle θ. To illustrate the theoretical result, the process noise σ_ω and σ_x are put to zero. Experiments with low but non-zero values are also conducted. The Lie group embedding of this system is given in Section 4.3.1, and is omitted here. In the framework of Invariant Smoothing, a propagation covariance Q_i is needed. Here we have $Q_i = 0$, thus the estiamtion problem appears ill-defined. It can still be inverted, however, using an alternative, but exact, rewriting of the solution which does not require inverting Q_i. This will be detailed in Lemma 1, in Section 5.1.2.

The same four smoothing methods as in Section 4.3.3 are compared here. The parameters are chosen as follows: vehicle initialised at the true position but with a wrong heading of $-3\pi/4$, an initial covariance of $diag(0, 0, (-3\pi/4)^2)$. The robot moves along a line at $7m/s$, the odometry being acquired at a rate of $1Hz$. For the low noise experiment, the odometry was polluted by a white noise of covariance $(0.001(m/s)^2, 0.0001(rad/s)^2)$. Position measurements of covariance $0.1m^2$ are received every five steps.

Expected behavior: maintaining a straight trajectory of a given length

When noise is turned off, $\theta_i = \theta_0$ for all i, therefore the true system satisfies, at step i:

$$x_i = x_0 + i\ R(\theta_0)u, \tag{5.1}$$

and in particular, for all θ_0, $\|x_i - x_0\|$ is constant. Added to the fact that the heading stays constant, this means that the true trajectory necessarily is a straight line whose length, given by u, is fixed. If the initial smoothing guess does satisfy this property, is it maintained during the subsequent iterations?

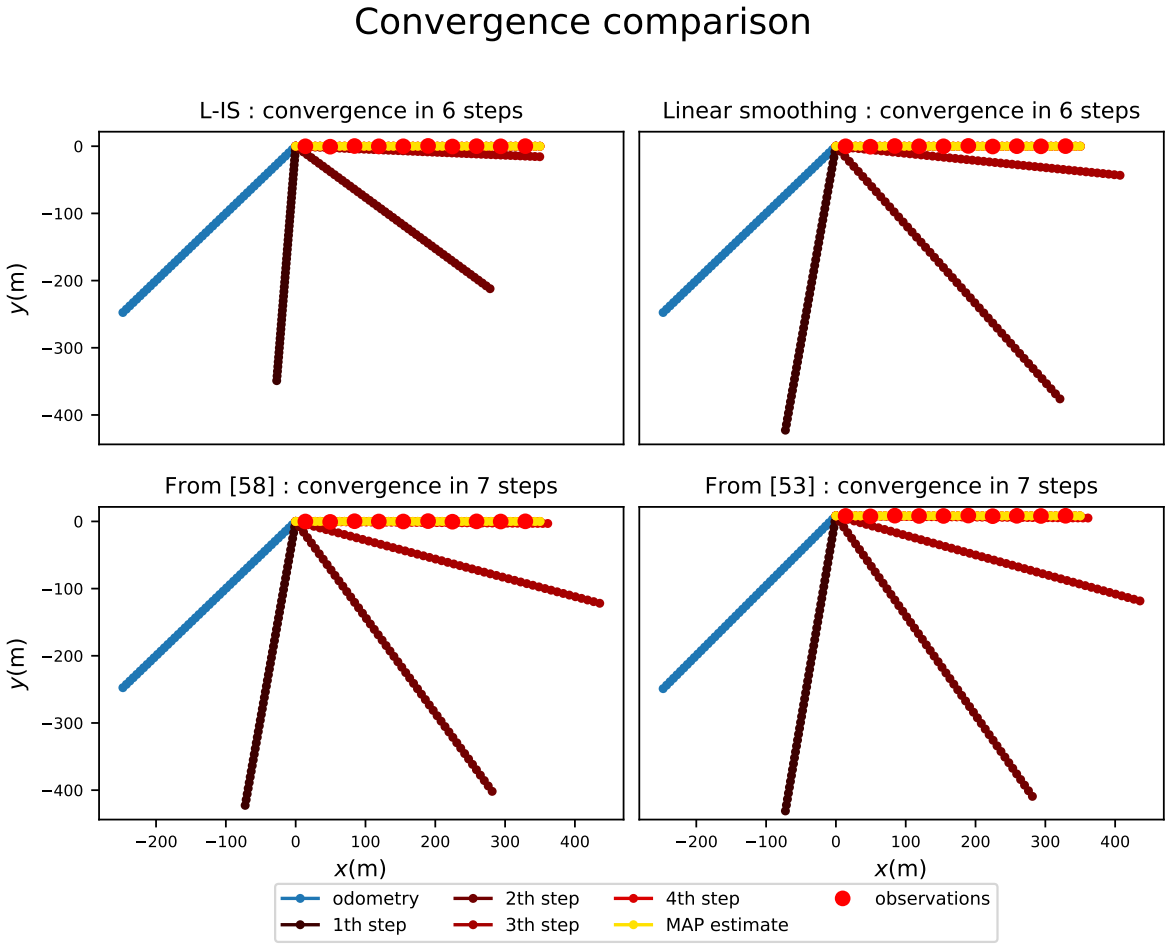


Figure 5.1: Illustration of Theorem 7. The estimates of four smoothing methods, including L-IS, in case of perfectly known initial position and propagation, are displayed. All methods maintain estimated trajectories as straight lines, however, only L-IS manages to keep the lines of the same length.

Simulation results

The successive estimates given by the four compared methods are displayed on Figure 5.1. It appears that all methods manage to estimate straight lines at each iterations, however only L-IS outputs successive lines of the same length. This means that the other methods necessarily break the conditions $x_{i+1} = x_i + R(\theta_0)u$ during the iterations. This is confirmed on Figure 5.2, which also reveals the heading of each state during the iterations. It then appears that these are collinear to the difference of successive positions only for L-IS. Mathematically, this means that all non-invariant smoothing methods actually induce estimates associated to an *infinite cost*.

Consequences in low noise cases

In [14], it was shown that the EKF could run into troubles and give wrong estimations in case of very low noise levels. In smoothing, relinearisation should temper this behavior, but convergence might still prove difficult. Results reported in Figure 5.3 confirm this fact. Indeed, in this low noise setting, with $\sigma_\omega^2 = 1e-3(rad/s)^2$, $\sigma_x^2 = 1e-4(m/s)^2$, the non-invariant methods have a much harder time converging than L-IS. This means that early stopping of the GN scheme could lead to poor estimate behavior, for two reasons. First, after marginalisation, the prior would have very low position covariance while the error would be significant. Second, since the Jacobians depend on the state, the propagated covariance

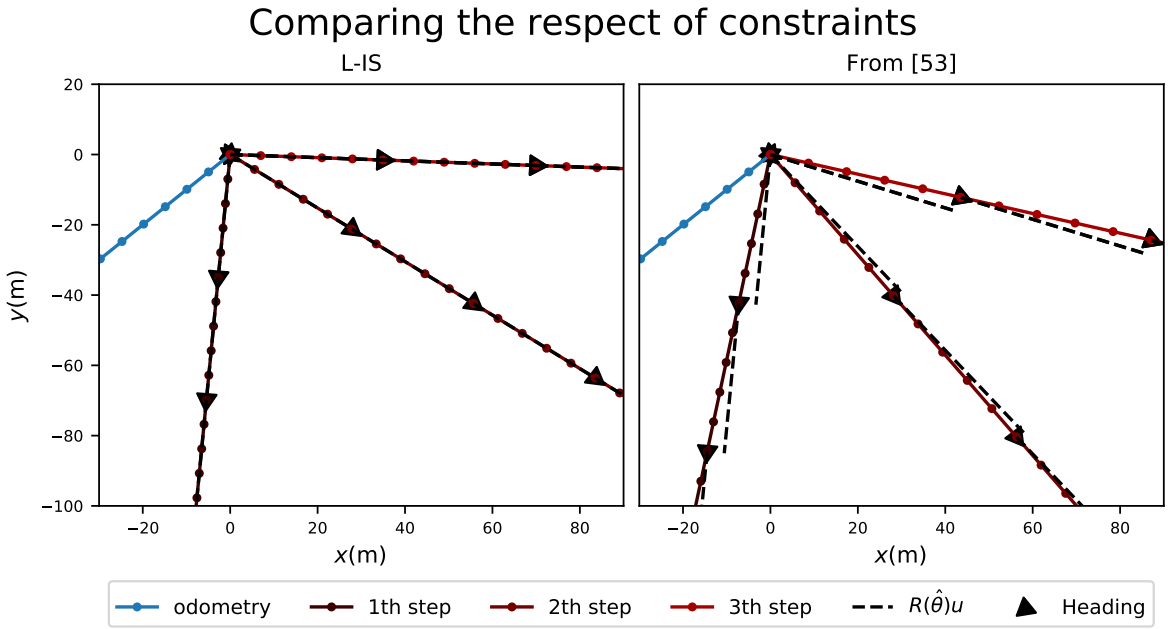


Figure 5.2: Illustration of Theorem 7. Zooms on the estimates of L-IS and [53] from Figure 5.1. For one in five points of each trajectory, the heading is represented as a triangle, and prolonged by a dotted line to assess the angle with the actual trajectory. Indeed, both should coincide. It reveals that only the heading output by L-IS coincide with the direction given by the positions.

would carry wrong information.

5.1.2 Invariant Smoothing respects deterministic group affine dynamics

The results presented in the previous section hinted that L-IS may have particular properties which are not shared by other smoothing methods. It turns out that Invariant Smoothing also satisfies the geometric constraints the IEKF was shown to respect in Section 3.4. Indeed, consider the same system as for L-IS in Section 4.2, but with noise-free dynamics:

$$\begin{aligned} \chi_0 &= \bar{\chi}\exp(w_0) & w_0 &\sim \mathcal{N}(0, P_0) \\ \chi_{i+1} &= f_i(\chi_i), & & \\ z_k &= \chi_k \star b_k \ \exp_Y(v_k) & v_k &\sim \mathcal{N}(0, N_k) \end{aligned} \tag{5.2}$$

where f_i are group-affine dynamics, and $b_k \in Y$. Suppose moreover that χ_0 lies in an affine subgroup, i.e., $w_0 \in L_0$, a Lie-subalgebra of $\mathfrak{g}$. Then Proposition 4 ensures that the true state, for all step i, satisfies

$$\chi_i \in \bar{\chi}_i G_i, \qquad G_i = \exp(L_i) = \exp(F_{i-1} \cdots F_0 L_0), \tag{5.3}$$

where $\bar{\chi}_i = f_{i-1} \circ \cdots \circ f_0(\bar{\chi})$, and F_i is the matrix associated to the log-linearity of f_i (see Theorem 2), i.e., such that $f_i(\chi\exp(\xi)) = f_i(\chi)\exp(F_i\xi)$. In particular, denoting as Θ_j the automorphism associated to f_j (see (2.26) from Corollary 1), $G_i = \Theta_{i-1} \circ \cdots \circ \Theta_0(G_0)$.

Then the estimate given by L-IS preserves all these constraints:

Theorem 7. *Consider the system described by* (5.2)*, estimated by L-IS. Let* $\hat{\chi} = (\hat{\chi}_i)_i$ *represent the current estimate, and* $\Xi = (\xi_i)_i$ *the update computed by L-IS. Then*

$$\forall i,\ \hat{\chi}_{i+1} = f_i(\hat{\chi}_i) \Rightarrow \forall i,\ \hat{\chi}_{i+1}\exp(\xi_{i+1}) = f_i(\hat{\chi}_i\exp(\xi_i)) \tag{5.4}$$

Moreover, suppose the initial state is constrained in an affine Lie group $\chi_0 \in \bar{\chi}G_0$*, where* $G_0 = \exp(L_0)$ *for* L_0 *a Lie sub-algebra of* $\mathfrak{g}$*. Then the L-IS estimate also satisfies*

$$\mathrm{Im}(P_0) \subset L_0 \wedge \forall i,\ \hat{\chi}_i \in \bar{\chi}_i G_i \Rightarrow \forall i,\ \hat{\chi}_i\exp(\xi_i) \in \bar{\chi}_i G_i, \tag{5.5}$$

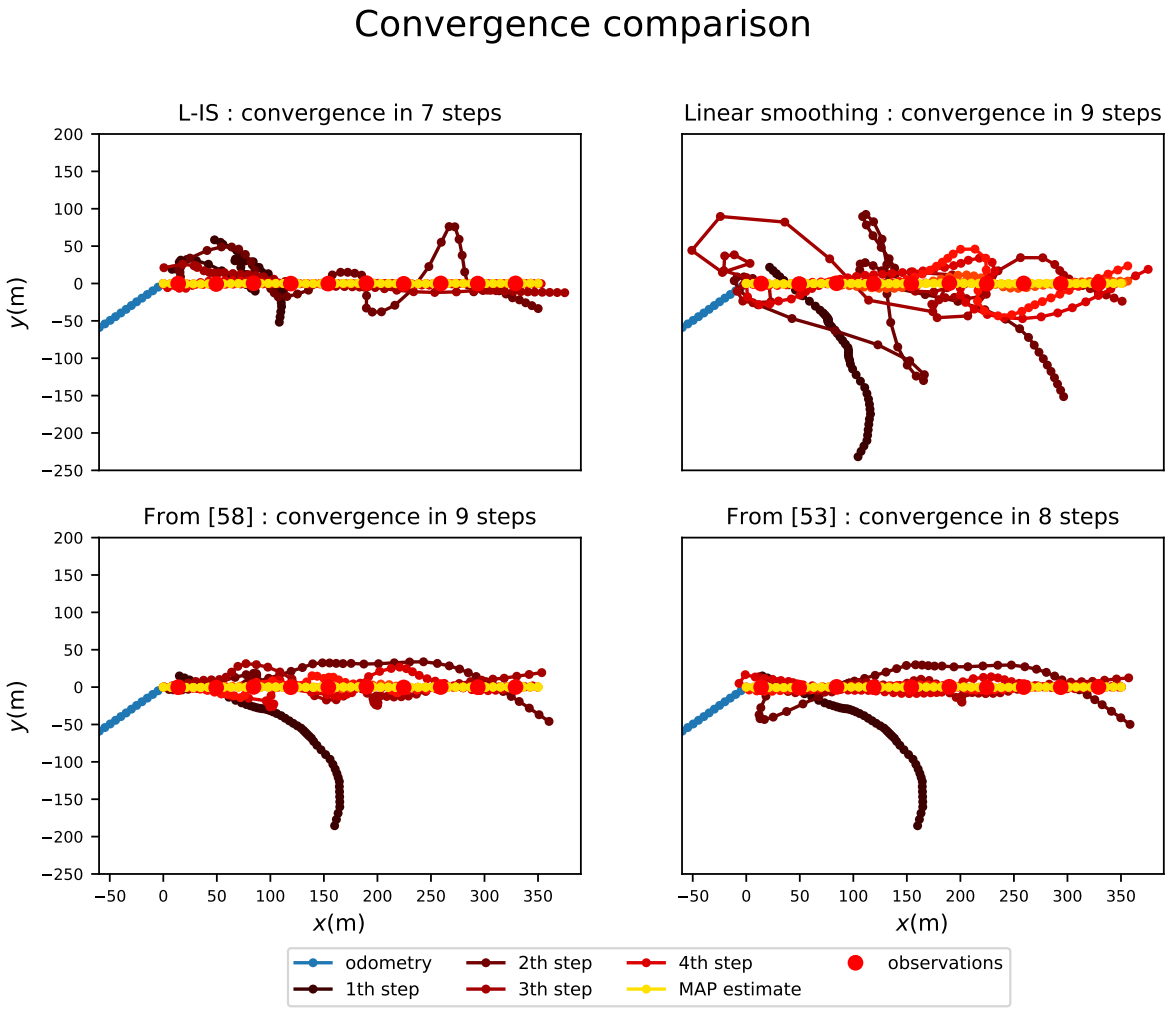


Figure 5.3: Consequences of Theorem 7 with low noise. The estimates of the same methods as Figure 5.1 are once again compared, this time with a low noise setting. Initial position covariance is $0.0025m^2$ for each direction, that of the linear and angular velocities are 1e-3$(m/s)^2$ and 1e-4$(rad/s)^2$ respectively. Measurement covariance is $0.1m^2$. L-IS converges faster than the other methods.

Before proving this Theorem, we can state and briefly prove its direct corollary about R-IS.

Corollary 4. *If $z_k = b_k \star \chi_k \exp_Y(v_k)$ is modeled by a right group action, then R-IS satisfies the constraints similar to those of L-IS:*

$$\forall i,\ \hat{\chi}_{i+1} = f_i(\hat{\chi}_i) \Rightarrow \forall i,\ \exp(\xi_{i+1})\hat{\chi}_{i+1} = f_i(\exp(\xi_i)\hat{\chi}_i) \tag{5.6}$$

$$\mathrm{Im}(P_0) \subset L_0 \wedge \forall i,\ \hat{\chi}_i \in G_i\bar{\chi}_i \Rightarrow \forall i,\ \exp(\xi_i)\hat{\chi}_i \in G_i\bar{\chi}_i, \tag{5.7}$$

where $G_i = \exp(F_{i-1}\cdots F_0 L_0)$, with F_j such that $f_j(\exp(\xi)\chi) = \exp(F_j\xi)f_j(\chi)$

Proof. Consider the new state variable $\eta = \chi^{-1}$. Then z_k is modeled by left group actions for η. Moreover, it can be easily checked that the associated dynamics stay group-affine. Therefore, the estimates of $\hat{\eta} = \hat{\chi}^{-1}$ given by L-IS will satisfy the given constraints: $\hat{\eta}_i \in (\bar{\chi}^{-1})G_i$. In turn, so will $\hat{\chi}$. □

The proof of this Theorem relies on the following linear algebra Lemma, which will also prove useful in Section 5.2 and in Part II.

Lemma 1. *Consider the well-posed linear least-squares problem with Jacobian A, covariance P, and error b, i.e., such that the correction satisfies*

$$\Xi^* = \underset{\Xi}{\mathrm{argmin}}\, \|A\Xi - b\|_P^2 = (A^T P^{-1} A)^{-1} A^T P^{-1} b \tag{5.8}$$

There exists a partition of A, P and b into two parts,

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}, \quad P = diag(P_1, P_2), \quad b = \begin{pmatrix} b_1 & b_2 \end{pmatrix}^T, \tag{5.9}$$

such that A_1 is invertible. Thanks to the matrix inversion lemma, the least-square problem's solution can be rewritten in a Kalman filtering fashion as

$$\Xi^* = A_1^{-1}\begin{pmatrix} I - KJ & K \end{pmatrix}\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{5.10}$$

$$J = A_2 A_1^{-1} \qquad K = P_1 J^T (J P_1 J^T + P_2)^{-1} \tag{5.11}$$

Proof. The existence of the partition directly comes from the fact that the problem is well-posed, thus A is full-rank. For all Ξ, the least-squares cost then rewrites

$$\|A\Xi - b\|_P^2 = \|A_1\Xi - b_1\|_{P_1}^2 + \|A_2\Xi - b_2\|_{P_2}^2 = \|\Xi - A_1^{-1}b_1\|_{A_1^{-1}P_1A_1^{-T}}^2 + \|A_2\Xi - b_2\|_{P_2}^2$$

Therefore, solving this problem is equivalent to applying a Kalman update on the prior $A_1^{-1}b_1$ with prior covariance $A_1^{-1}P_1A_1^{-T}$, innovation b_2 and measurement covariance P_2 (see the optimisation point of view of the Kalman update recalled in Section 4.1.1), which gives the result. □

The idea behind this partition is that P_1 will contain the covariances which should not be inverted. Typically, in this case it will be composed of the initial prior and the propagation factors, making A_1 invertible. Suppose that there are less constrained factors than the dimension of the state. Then the SVD of A_1 can be used, or A_1 can simply be filled until it gets invertible which is possible since the problem is well-posed. The case in which there would be more constrained factors than the dimension of the state is out of scope here, because it would represent an over-constrained problem. This Lemma allows proving Theorem 7.

Proof of Theorem 7

Proof. Let the cost of the linear least-squares to be inverted for the next L-IS step be given by $\|A\Xi - b\|_P^2$, for all Ξ. The proof is based on a partition of the state as in Lemma 1, where A_1 contains the prior and propagation factors, and is therefore lower triangular. Its inverse is given by

$$A_1^{-1} = \begin{pmatrix} Id & & \\ (F_{i-1}\cdots F_j)_{i,j} & \ddots & \\ & & Id \end{pmatrix} \tag{5.12}$$

where the lines and columns are numbered $i,j \in [|0,n-1|]$. As defined in Section 4.2.2, the prior Jacobian is Id, and its right-hand side is $a_0 = \log(\bar{\chi}^{-1}\hat{\chi}_0)$. For the propagation factor from i to $i+1$, the Jacobian is F_i and the right-hand side $\log(f_i(\hat{\chi}_i)^{-1}\hat{\chi}_{i+1}) = 0$. Therefore, it is clear that $b_1 = (a_0^T, 0, \cdots, 0)^T$, where $a_0 \in L_0$. A_1 having only non-zero blocks on the diagonal and the first (block) subdiagonal, its inverse is triangular. Thus, A_1^{-1} and P_1 write

$$A_1^{-1} = \begin{pmatrix} Id & & \\ (F_{i-1}\cdots F_j)_{i,j} & \ddots & \\ & & Id \end{pmatrix} \qquad P_1 = \mathrm{diag}(J_0^{-1}P_0J_0^{-T}, 0, \cdots, 0), \tag{5.13}$$

where the lines and columns are numbered $i,j \in [|0,n-1|]$. In particular, only the first line of K is non-zero, and written K_0. It follows that

$$\Xi = A_1^{-1}\begin{pmatrix} a_0 - K_0Jb_1 + K_0b_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

The facts that J_0^{-1}, P_0 stabilise L_0 and $\mathrm{Im}(K_0) \subset \mathrm{Im}(P_0)$ ensure that $\xi_0 \in L_0$. The structure of A_1^{-1} then guarantees that, for all i, $\xi_i = F_{i-1}\xi_{i-1}$ and $\xi_i \in L_i$.

Since the update is based on the Lie exponential, and f_i are group affine, the result directly follows. □

Why non-invariant smoothings fail

In Section 3.4, what allowed the IEKF to respect geometric constraints were the independence of the propagation Jacobians from the estimate, the use of the Kalman gain, and the exponential update. In the present case, the Kalman gain is replaced by the use of GN iterations, which are shared by all smoothing methods. The Jacobians' argument does not hold so well: if $\hat{\chi}_{i+1} = f_i(\hat{\chi}_i)$, then the Jacobians used by L-IS and [53] coincide exactly. In particular, L-IS and H-IS are identical in this case. However, the log-linearity of group-affine dynamics is still crucial. Therefore, it appears that the exponential update is the key ingredient here.

5.1.3 Conclusion

This section proves a result for Invariant Smoothing under geometric constraints which is the counterpart of that for the IEKF given in Section 3.4: with noise-free dynamics, if the true trajectory lies in a known subgroup, the current estimate also lies in it, and the prior covariance reflects this knowledge, then so will the next estimate. This gives insights as to why Invariant Smoothing behaves well in the presence of high-quality motion sensors, and in particular inertial ones, as evidenced already in Section 4.4.6, and later in Section 6. Indeed, although taking the biases into account makes imperfect group-affine dynamics, since they are stable in time, for a given value the system turns almost group-affine.

5.2 Unobservability from a smoothing point of view

The notion of observability asks whether the parameters of a system can be effectively estimated given a series of measurements. That is, forgetting about the noise, does there exist two different situations in which a system would give the same outputs? In SLAM this is obviously the case: since the environment is initially unknown, everything is computed with respect to the initial position and heading of the vehicle, see Figure 5.4 (reproduced from [16]).

Consider a system χ whose noisy dynamics write at step i, $\chi_{i+1} = f_i(\chi_i, w_i)$ and whose outputs are given by $h_i(\chi_i, v_i)$, for w_i and v_i some noises. Mathematically, the symmetries of the system are based on the following definition [22,88].

Definition 6. *Let $\phi_.$ be an invertible group-action of some group G' on the state space G, i.e., for all $\alpha \in G'$, $\phi_\alpha : G \to G$ is bijective. Then a system governed by f_i and h_i is said to be totally invariant under*

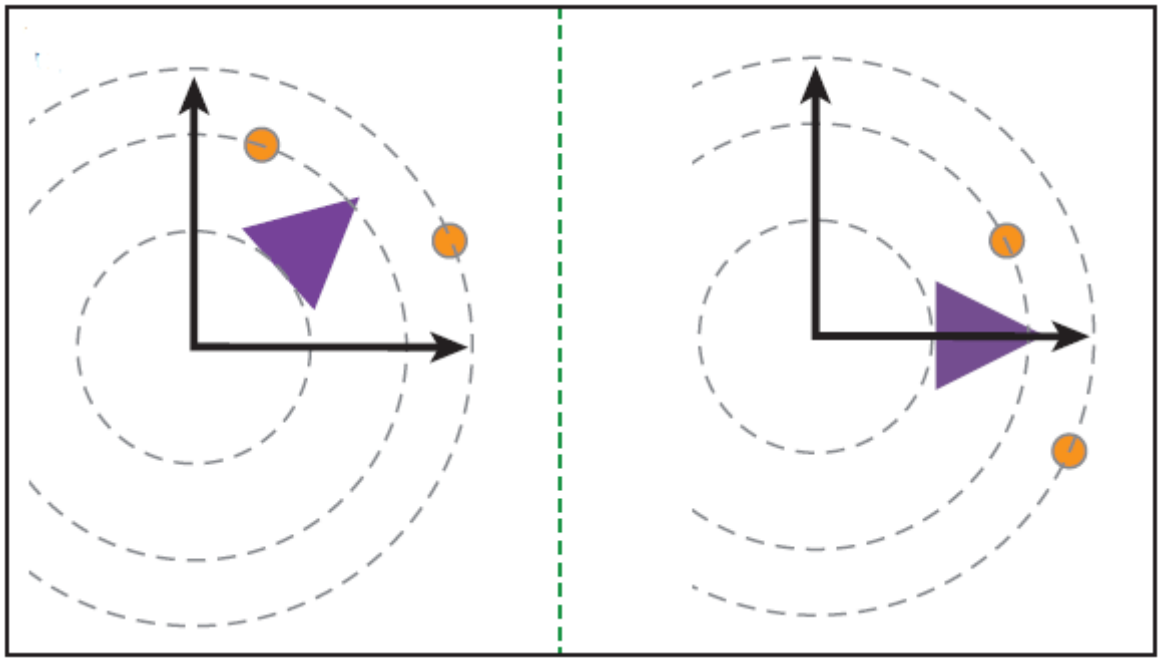


Figure 5.4: Reproduced from [16]. The SLAM problem is unobservable because a global translation or rotation of the robot and the landmarks is impossible to perceive: A robot that starts from respectively the configurations from the left and right pannel above, and that is fed with a series of control inputs that make it move along an arbitrary trajectory, will never be able to distinguish between the two initial conditions, since landmarks measurements are relative.

the action of ϕ if the dynamics f_i and observation h_i are equivariant and invariant w.r.t. ϕ respectively. That is, for all α, i, χ, w, v:

$$\phi_\alpha(f_i(\chi, w)) = f_i(\phi_\alpha(\chi), w) \tag{5.14}$$

$$h_i(\phi_\alpha(\chi), v) = h_i(\chi, v) \tag{5.15}$$

The actions $\phi_.$ under which a system is unobservable create indistinguishable trajectories. Indeed, let $(\chi_i)_i$ be the true trajectory of the system, for a given noise w_i. Then, for all α, the trajectory $(\lambda_i)_i = (\phi_\alpha(\chi_i))_i$ satisfies

$$\lambda_{i+1} = f_i(\lambda_i, w), \qquad h_i(\lambda_i, v) = h_i(\chi_i, v),$$

and there is no way to differentiate one from the other. Ideally, one would like any estimator of this system to be able to reflect the existence of these symmetries. This is important in order to ensure the consistency of the estimator, which is the fact that it does not underestimate its error. Typically, an estimate is said to be consistent when its error lies inside the $3-\sigma$ enveloppe of its covariance. In this regard, not respecting the symmetries might lead to overconfident estimates, which can then turn inconsistent. Classically, assessing the behavior of an estimator with respect to symmetries has been done using the of observability matrix, and the notion of false observability.

5.2.1 Standard observability theory: The observability matrix and false observability

Consider a trajectory $(\chi_i)_i$ of a system governed by f_i (noise turned off) and h_i. The observability matrix is defined using linearisations around this true value

$$\mathcal{O}((\chi_i)_i) = \begin{pmatrix} H_0 \\ H_1 \Phi_{1,0} \\ \vdots \\ H_k \Phi_{k,0} \end{pmatrix} \tag{5.16}$$

where H_i is the observation Jacobian at step i, and $\Phi_{i,0} = F_{i-1} \cdots F_0$ the state transition matrix from 0 to i, *computed around the true state* $(\chi_i)_i$. It studies the first-order impact of small updates of the initial state on the output of the system. A non-trivial kernel thus implies that there exists a non-zero update has no impact on the output, i.e., is not observable. These are called the unobservable directions of the system. In particular, as stated in [29], if the system is invariant under the action of ϕ, then it induces an unobservable direction. This link between symmetries and unobservable directions is formalised through the following result.

Proposition 7 (From [29])**.** *If a system governed by f_i and h_i is invariant under the action of $\phi_.$, then, for any true trajectory $(\chi_i)_i$, the directions infinitesimally spanned by $\phi_.$ at these points , i.e., $\frac{\partial}{\partial \alpha}\phi_\alpha|_{Id}(\chi_0)$, lie in the kernel of $\mathcal{O}((\chi_i)_i)$.*

Since the observability matrix relies on the Jacobians of the system, its kernel depends on the chosen parametrisation $\boxplus$. For instance, a global rotation around the gravity does not have the same form for the left- and right-invariant errors (but they are related by Ad_χ).

Notation 1. *Let $v_1, \cdots, v_p$ be a basis of the vector space $\{\frac{\partial}{\partial \alpha}\phi_\alpha|_{Id}(\chi_0),\ \phi_.$ is a symmetry of the system $\}$. The matrix of this basis is denoted as*

$$M_{\boxplus}(\chi_0) = \begin{bmatrix} v_1 & \cdots & v_p \end{bmatrix} \tag{5.17}$$

In particular, since the vectors composing $M_{\boxplus}(\chi_0)$ have been shown to form unobservable directions, it satisfies

$$\mathcal{O}((\chi_i)_i) M_{\boxplus}(\chi_0) = 0 \tag{5.18}$$

Considering pose-graph like settings

The way the observability matrix is formulated makes it adapted to systems having unary measurements, i.e. measurements of the current state only. A simple way to extend it to n-ary measurements is to extend the state's dimension, by including at all time all the states in the trajectory, or at least all the past ones. This is formalised in the notion of Stochastic Cloning [85], which will be thoroughly detailed and heavily used in Part II. It is a treatment close to that of multiple robots sharing relative measurements [89]. As far as the observability matrix is concerned, this simply means that the Jacobian F_i must be padded with identity blocks for the past states, and H_i will contain non-zero blocks only for the states involved in the measurement. This is illustrated thereafter on a toy example.

An illustrative toy example

Consider the decoupled system, of state $(z,x) \in \mathbb{R}^n \times \mathbb{R}^p$, and its outputs given by

$$\begin{bmatrix} z_{i+1} \\ x_{i+1} \end{bmatrix} = \begin{bmatrix} z_i \\ f_i(x_i) \end{bmatrix} = g_i(\begin{bmatrix} z_i \\ x_i \end{bmatrix}) \qquad y_i = x_i - x_{i-1} \tag{5.19}$$

It is clear that this system is symmetric in z_0: any global transformation in z, $(z,x) \mapsto (\phi(z),x)$, leads to a new trajectory producing the same outputs. For the standard vectorial parametrisation $(z,x) = (\hat{z}+\delta_z, \hat{x}+\delta_x)$, the propagation Jacobian F_i, and the ones for the observation, H_i and H_{i-1}, write

$$F_i = \begin{bmatrix} Id & \\ & F_i^x \end{bmatrix}, \qquad H_i = -H_{i-1} = \begin{bmatrix} 0 & Id \end{bmatrix},$$

where F_i^x is the Jacobian of f_i, computed at $\hat{x}_i$. Since they are relative measurements, expend the state such that:

$$X_0 = \begin{bmatrix} z_0 \\ x_0 \end{bmatrix}, \qquad X_{i+1} = \begin{bmatrix} g(X_i) \\ X_i \end{bmatrix} \tag{5.20}$$

$$\tilde{F}_i = \begin{bmatrix} F_i \\ Id \end{bmatrix}, \qquad \tilde{H}_i = \begin{bmatrix} H_i & H_{i-1} & 0 \end{bmatrix}, \tag{5.21}$$

In turn, the observability matrix writes

$$\mathcal{O}((z_i, x_i)_i) = \begin{pmatrix} 0 & F_0^x - Id \\ \vdots & \vdots \\ 0 & (F_{k-1}^x - F_{k-2}^x) \cdots F_0^x \end{pmatrix} \tag{5.22}$$

Consider for instance the action $\phi_\alpha(z,x) \mapsto (z + \alpha v, x)$ defined for $\alpha \in \mathbb{R}, v \in \mathbb{R}^n$, under which the system is invariant. Then the direction infinitesimally spanned by $\phi_\cdot$, $(v,0)$ is an unobservable direction. In addition, suppose that the system is not invariant under a transformation impacting x. The unobservable directions coming from the symmetries thus give

$$M_+(z_0, x_0) = \begin{bmatrix} Id \\ 0 \end{bmatrix} \tag{5.23}$$

And it is straightforward to check that (5.18) is satisfied indeed. The system would have more unobservable directions, depending on the dynamics f_i for the x part. If f_i is the identity map, then the initial state is completely unobservable.

The problem of false observability

An estimate suffers from false observability when it gains information along unobservable directions, thus artificially reducing its uncertainty. It occurs when the observability matrix computed *around the estimated state*, $\mathcal{O}((\hat{\chi}_i)_i)$, has a different kernel than $\mathcal{O}((\chi_i)_i)$, usually of lower dimensions. It was the case for early EKF-SLAM for instance, due to the noise and Kalman updates [67]. In the long run, this lead to an overconfident, and therefore inconsistent, estimation.

Indeed, SLAM has at least 4 or 6 unobservable directions, depending on whether inertial sensors are used. If not, the full initial pose is unobservable. With an IMU, only the yaw, i.e., the rotation around the gravity, is unobservale in addition to the initial position. For early EKF-SLAM, the unobservable directions linked to global translations of the system were respected, but not those related to global rotations. This leads to an artificial decrease in the estimate's covariance along the unobservable directions, making the estimate unreliable in the long run. This issue has plagued early filtering approaches, namely EKF-SLAM and FastSLAM [4,71]. This also helped smoothing to establish itself, its relinearisation tempering the harm of these symmetry problems. False observability, which has become a well-known issue, received much attention and many solutions were proposed, based on constraints on the estimated observability matrix [63,67]. Parametrisation-based approaches allowed designing naturally consistent estimators, as introduced in [11] and generalised in [29].

All these works focused on the observability (or sometimes the Fisher information) matrix. In this section, the focus is put on another view of (un)observability, through the smoothing updates. The role of the prior covariance in respecting symmetries, which was not, to the author's knowledge, considered yet, is brought forth. It is shown that the prior's structure may transform the directions which will receive no update. Although this will not reduce the number of these directions, and thus is substantially less harmful than false observability, this could have impacts on the long run, or when fusing global and local navigation.

5.2.2 Unobservability seen through a smoothing update

Most recent local navigation systems (mostly VIO and SLAM) respect the system's observability properties [66]. In this context, R-IS has a clear advantage over L-IS, since only the former respects the unobservability of pose-SLAM (which uses relative poses as observations), as will be shown on a simple but easily generalisable example. Contrary to filtering which marginalises the past states, in a smoothing setting (without sliding windows) the initial state is still estimated. Therefore, one would like each computed update to respect the unobservable directions. One standard approach is to fix the initial state to zero and remove it from the estimation. However, this cannot always be done. Indeed, consider a car which navigates with GPS until entering a tunnel, where it will rely on local navigation only. Then the initial state cannot be given an arbitrary value and be considered perfectly known. Therefore this Section considers cases in which the initial state is kept in the estimation. In pose-SLAM for instance, where the full initial pose is unknown, one would like the applied correction to stay zero. For VIO systems, one would like that only the initial roll and pitch be corrected. It appears that, even if the estimated observability matrix has the right properties, this might not be the case.

More formally, consider a linearised system, with Jacobian A, covariance P, and error b, i.e., such that the correction satisfies

$$\Xi^* = \underset{\Xi}{\operatorname{argmin}} \|A\Xi - b\|_P^2 = (A^T P^{-1} A)^{-1} A^T P^{-1} b \tag{5.24}$$

Naturally, one would want the state correction to reflect the unobservability, i.e.,

$$M_{\boxplus}(\hat{\chi}_0)^T \delta^*_{X_0} = 0, \tag{5.25}$$

To get the property which the Jacobians must satisfy, the least square's solution is rewritten in the fashion of Lemma 1. Consider the partition of A, P and b into two parts, one collecting the initial prior and the propagation factors, the other made of the observation factors:

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}, \quad P = diag(P_1, P_2), \quad b = \begin{pmatrix} b_1 & b_2 \end{pmatrix}^T \tag{5.26}$$

Recall that, thanks to Lemma 1, the least-square problem's solution can be rewritten in a Kalman filtering fashion as

$$\Xi^* = A_1^{-1} \begin{pmatrix} I - KJ & K \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{5.27}$$

$$J = A_2 A_1^{-1} \qquad K = P_1 J^T (J P_1 J^T + P_2)^{-1} \tag{5.28}$$

$\delta^*_{X_0}$ is given by the first block line of $A_1^{-1} \begin{pmatrix} I - KJ & K \end{pmatrix}$ applied to b. Without loss of generality, we can assume the prior to have identity Jacobian. Then, because of the structure of A_1^{-1} given in (5.13), only the first block line of K, noted K_0, is important. Letting P_0 and a_0 be the covariance and error term of the prior respectively, we get

$$\delta^*_{X_0} = -a_0 + K_0(z - Jb_1) \tag{5.29}$$

Suppose that the prior respects the unobservability, i.e., $M_{\boxplus}(\hat{\chi}_0)^T a_0 = 0$. The wanted property becomes equivalent to $M_{\boxplus}(\hat{\chi}_0)^T K_0(z - Jb_1) = 0$. As this should be true for any measurement error z which might occur, and $z \mapsto (JP_1J^T + P_2)^{-1}(z - Jb_1)$ is an automorphism, this becomes

$$M_{\boxplus}(\hat{\chi}_0)^T P_0 [J^T]_0 = 0 \tag{5.30}$$

A major difference with standard observability analysis appeared: the prior's covariance is involved in this criteria. Therefore, it appears that **only the unobservable subpsaces which are preserved by the prior's covariance can be effectively respected**. This fact seems natural if we think in terms of correlations: if the initial heading and velocity are correlated for some reason, then updating one will automatically update the other, which might seem to break unobservability. However, this can also have an impact without correlations, for instance in inertial local navigation, as will be shown in Section 5.2.4.

The link with standard observability analysis

The standard observability analysis requires that $\mathcal{O}((\hat{\chi})_i) M_{\boxplus}(\hat{\chi}_0) = 0$. However, the structure of A_1^{-1} given in (5.13) show in particular

$$[A_1^{-T}]_0 = \begin{bmatrix} Id & F_0^T & \cdots & F_{i-1}^T \cdots F_0^T \end{bmatrix} \tag{5.31}$$

Then, given the structure of A_2, it turns out that $\mathcal{O}((\hat{\chi})_i)$ is the first column of J, and in turn that

$$\mathcal{O}((\hat{\chi})_i) M_{\boxplus}(\hat{\chi}_0) = 0 \Leftrightarrow M_{\boxplus}(\hat{\chi}_0)^T [J^T]_0 = 0 \tag{5.32}$$

Finally, it appears that the standard observability criterion is a particular case of (5.30) where $P_0 = Id$.

5.2.3 The impact of the prior covariance illustrated on a toy example

Consider the toy system defined by 5.19. Then, the first line of J^T being the transpose of $\mathcal{O}((\hat{\chi})_i)$, it is clear that it produces no information on z. However, the prior covariance P_0 is applied to compute the smoothing update. In particular, if there is a correlation between x and z, then P_0 writes $P_0 = \begin{bmatrix} P_{zz} & P_{zx} \\ P_{xz} & P_{xx} \end{bmatrix}$, and this leads to a non-zero update of z. This is completely normal, and an expected behavior in this case. However, this underlines the need to carefully initialise the prior covariance in order to respect the symmetries. Here, it is trivially done as it sufficies to put no correlation between x and z. In the case of inertial navigation, it is trickier, as appears in Section 5.2.4.

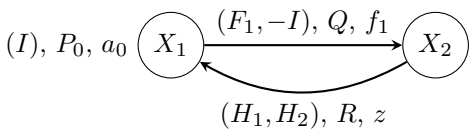


Figure 5.5: Linearised factor graph of a reduced example. Arrows indicate constraints between nodes, labels give the corresponding Jacobians, covariances and errors. A prior P_0 is given on X_0.

5.2.4 Studying a reduced example for SLAM and inertia-aided pose-SLAM

Consider the system whose factor graph is given in Figure 5.5. The system is linearised from an arbitrary initial guess. The Jacobian, inverse covariance matrix, and error vector are directly obtained as:

$$A = \begin{pmatrix} -I & \\ F_1 & -I \\ H_1 & H_2 \end{pmatrix} \qquad \begin{matrix} P = diag(P_0,\ Q,\ R) \\ b = \begin{pmatrix} a_0 & f_1 & z \end{pmatrix}^T \end{matrix} \tag{5.33}$$

In this case, Equation (5.30) becomes

$$M_{\boxplus}(\hat{\chi}_0)^T P_0 (H_1^T + F_1^T H_2^T) = 0 \tag{5.34}$$

In particular, the fact that L-IS relies on Jacobians which depend on the noisy measurements makes (5.34) very unlikely to be fulfilled in general. R-IS however behaves much better, due to its propagation and observation Jacobians being independent of both the estimate and the measurements, as shown in the subsequent paragraphs.

Consistency of R-IS for pose-SLAM

In standard pose-graphs, the whole initial pose is unobservable, that is, δ_{χ_0} must always be 0. This is seamlessly respected by R-IS, thanks to its unusually simple propagation and observation jacobians presented in Section 4.5.1:

Theorem 8. *For standard pose-SLAM, the GN smoothing algorithm based on the right-invariant error parametrisation satisfies* (5.34)*, and is therefore consistent. Moreover, the smoothing updates respect this unobservability*

Proof. Indeed, the observation model of pose-SLAM is given by (4.70), with $\alpha = Id$. Therefore, we have $F_1 = H_1 = -H_2 = Id$, and thus $[J^T]_0 = 0$, leading to the desired result. □

Restricted consistency of inertial pose-SLAM

The unobservable directions of VINS are known to be the global initial position and rotation around the gravity [63]. Using the left-invariant error, these unobservable directions are given by

$$M_{\boxplus_L}(\hat{\chi}_0) = \begin{pmatrix} R_0^T g & \\ R_0^T (g)_\times v_0 & \\ R_0^T (g)_\times x_0 & I_3 \end{pmatrix} \tag{5.35}$$

The unobservable directions for the right-invariant parametrisation are easily derived as

$$M_{\boxplus_R}(\hat{\chi}_0)^T = M_{\boxplus_L}(\hat{\chi}_0)^T Ad_{\hat{\chi}_0^{-1}} \tag{5.36}$$

Denote, in the lines of (4.54), the inertial propagation $f(\chi) = \Gamma\phi(\chi)\Upsilon$. Then its Jacobian and that of the observation are given by

$$F = Ad_\Gamma \Phi, \qquad H_1 = -H_2 = \begin{pmatrix} Id & \\ & Id, \end{pmatrix} \tag{5.37}$$

In turn, this allows computing $[J^T]_0 = H_1^T + F_1^T H_2^T$

$$[J^T]_0 = \begin{pmatrix} 0 & a(g)_\times \\ 0 & b\ Id \\ 0 & 0 \end{pmatrix}, \tag{5.38}$$

where a, b are real numbers depending on the preintegration time Δt. Recalling, from Section 4.5.1, that the prior covariance in R-IS is linked with that of L-IS through $P_0^R = Ad_{\hat{\chi}_0} P_0^L Ad_{\hat{\chi}_0}^T$, (5.30) becomes

$$M_{\boxplus_R}(\hat{\chi}_0)^T P_0^R [J^T]_0 = M_{\boxplus_L}(\hat{\chi}_0)^T P_0^L Ad_{\hat{\chi}_0^T} [J^T]_0 \tag{5.39}$$

The correction by $Ad_{\hat{\chi}_0}$ is part of R-IS, and not a parameter to tune as is P_0^L. Therefore, it must be considered even for the standard observability study. Computations give

$$Ad_{\hat{\chi}_0}^T [J^T]_0 = \begin{pmatrix} 0 & a\hat{R}_0^T (g)_\times - b\hat{R}_0^T (\hat{v}_0)_\times \\ 0 & b\hat{R}_0^T \\ 0 & 0 \end{pmatrix}, \tag{5.40}$$

and, in turn, it is easy to verify that

$$M_{\boxplus_L}(\hat{\chi}_0)^T Ad_{\hat{\chi}_0}^T [J^T]_0 = 0, \tag{5.41}$$

confirming that the chosen parametrisation preserves the standard observability conditions of VINS. Nevertheless, the update $\delta_{\chi_0}^*$ actually computed *might not respect it.* This is because the prior covariance P_0^L comes into play in (5.34). Indeed, in order to satisfy this condition, P_0 should need to preserve the orthogonal space of the vector corresponding to $(\hat{R}_0^T g, \hat{R}_0^T (g)_\times \hat{v}_0, \hat{R}_0^T (g)_\times \hat{x}_0)$. Assuming that P_0 is diagonal, as is usually the case, this implies that P_0^L should be isotropic on the full navigation state, unless $\hat{v}_0$ and $\hat{x}_0$ are zero. The latter is the case in standard VIO systems for instance, usually initialised while standing still, but might not be if other global measurements are available. Theorem 9 summarises and completes these results for the biases.

Theorem 9. *For unbiased inertial pose-graphs, the matrix $M_{\boxplus}((\hat{\chi}_i)_i)$ contains the global initial position, and rotation around the gravity. R-IS verifies a restricted version of the consistency property:*

$$M_{\boxplus_L}(\hat{\chi}_0)^T Ad_{\hat{\chi}_0}^T [J^T]_0 = 0 \tag{5.42}$$

Suppose that P_0^L encodes no correlation between attitude, velocity and position. In order to fully satisfy (5.34)*, it must also be assumed one of the following*

(i) The initial position and velocity are zero, or

(ii) P_0^L is isotropic

Moreover, this stays true in the biased case.

Proof. There is only the part related to the biases to be shown. Since they are not involved in the relative pose measurement, the Jacobians H_1 and H_2 are padded with zeros. Therefore, only the navigation part of F_0 is used in the computation of $[J^T]_0$. Referring to the structure of the Jacobian with biases (4.62), the result follows directly. □

5.2.5 Conclusion

This section advocates the use of R-IS over L-IS in a local navigation context, because it does not suffer from false observability. However, considering observability from a smoothing point of view, it appeared that the prior covariance may play a role in the respect of the system's symmetries. This is particularly true in unbiased inertial navigation, for which only a part of the original state, its position and yaw, are unobservable. In this case, even a diagonal covariance P_0^L does not ensure the respect of these symmetries by the GN updates. Indeed, an additional isotropy condition is sometimes needed. This is highly unsatisfactory, as it imposes an equality constraint on values whose comparison would have no physical meaning (an angle error versus a position error for instance). Moreover, it is unit dependent. This echoes the recent work of [98,99], which studied generalised inverses other than the Moore-Penrose one, respecting other invariances and in particular unit changes, i.e., diagonal transformations. Still, the use of the Moore-Penrose inverse directly derives from the considered cost function, a least-squares problem. There thus lies an open question of whether this cost should be adapted to lead to these new generalised inverses, and if this is even desirable.

5.3 Conclusion

In this chapter the behavior of Invariant Smoothing was studied in two particular cases: in the presence of geometric constraints, and with unobservable directions. In the former case, L- and R-IS exhibit properties similar to that of the IEKF from Chapter 3. In particular, with deterministic dynamics, they are the only smoothing methods giving estimates associated to a finite cost. Concerning systems with symmetries, leading to unobservable directions, such as SLAM or VIO, only R-IS satisfies the standard consistency properties, i.e., respects the kernel of the observability matrix. However, considering not only this matrix but the update computed at each iteration, the unexpected role of the prior covariance in the system's consistency was unveiled.

Chapter 6

Real-world experiments

Highlights

- L- and R-IS and their properties are evaluated on the experimental data coming from a vehicle equipped in particular with a high-grade IMU.
- In global navigation with low biases, little to no difference arise between invariant and non-invariant smoothing methods.
- With strong biases, the exponential update used in Invariant Smoothing appears crucial to keep consistency.
- In local navigation, as outlined in Chapter 5, L-IS loses its consistency while R-IS maintains it. Moreover, on a challenging case, R-IS maintains long-term reliability of the estimation, while [53] slowly degrades, which was not expected.
- The impact of successive sliding window size expansions on the estimation output is decreasing, and becomes negligible at some point.

Points marquants

- Les lissages invariants à gauche et à droite sont évalués sur des données expérimentales provenant d'un véhicule équipé d'une centrale inertielle de haute qualité.
- En navigation globale, avec de faibles biais, peu, voire aucune différence, n'apparaît entre les méthodes de lissage invariantes et non-invariantes.
- Avec des biais plus forts, la mise à jour exponentielle utilisée pour le lissage invariant apparaît cruciale pour la cohérence de l'estimation.
- En navigation locale, comme souligné dans le Chapitre 5, L-IS perd sa cohérence tandis que R-IS la maintient. Par ailleurs, dans un cas difficile, R-IS maintient une fiabilité à long terme de l'estimation, tandis [53] se dégrade lentement, ce qui n'était pas attendu.
- L'impact, sur l'état estimé, d'extensions successives de la taille de la fenêtre glissante décroît, et devient négligeable à partir d'un certain point.

Introduction (En/Fr)

The goal of this thesis is to develop and evaluate navigation algorithms for systems based on precise inertial sensors. The theory of Invariant Smoothing developed in the previous sections and the related numerical results advocate its use in some cases. Its interest for the main use case of this work still needs to be measured. The creation of an autonomous vehicle prototype at Safrantech was a perfect opportunity to do so, as it came with a high-grade IMU and formed as such a particularly well fitted test bed.

Three main conclusions can be drawn from the conducted experiments. As far as global navigation is concerned, smoothing methods are of interest only when heavy biases are at hand. In this case, the

exponential update of L-IS (and its variants) seems crucial for the estimate to remain well-behaved, especially before the heading becomes effectively observable if the IMU is gyro-compassing (i.e., allows observing the global heading after a few hours thanks to the Earth rotation). For local navigation, out of L-IS, the method based on [53], and R-IS, only the latter manages to keep a consistent estimation in the long term. Finally, in both global and local navigation, expanding the sliding window stops impacting the output at some point. In particular, when considering the low native biases of the embedded IMU, smoothing sometimes appeared almost as a waste of computational power compared to the IEKF.

Le but de cette thèse est de développer et évaluer des algorithmes de navigation pour des systèmes reposant sur des senseurs inertiels précis. La théorie du Lissage Invariant développée dans les précédentes sections et les résultats numériques obtenus encouragent son utilisation dans certains cas. Son intérêt pour la principale application de ce travail doit encore être mesurée. La création d'un prototype de véhicule autonome à Safrantech a été l'opportunité parfaite pour ce faire : contenant une IMU haute qualité, il forme un banc d'essai particulièrement adapté.

Trois conclusions principales peuvent être tirées des expériences conduites. Concernant la navigation globale, les méthodes de lissage ne présentent un intérêt que lorsque les biais se révèlent importants. Dans ce cas, la mise à jour du L-IS (et ses variantes) semble cruciale afin que l'estimé reste stable, en particulier avant que l'orientation ne devienne effectivement observable si l'IMU est gyro-compassante (i.e., permet d'observer l'orientation globale après quelques dizaines de minutes grâce à la rotation de la Terre). Pour la navigation locale, entre L-IS, la méthode basée sur [53], et R-IS, seul le dernier réussit à maintenir une estimation cohérente sur le long terme. Enfin, que ce soit en navigation globale ou locale, étendre la fenêtre glissante n'impacte plus la sortie au-delà d'une certaine taille. En particulier, lorsque les faibles biais réels de la centrale inertielle embarquée sont considérés, le lissage apparaît parfois presque comme un gaspillage de puissance de calcul comparé à l'IEKF.

6.1 Experimental setup

6.1.1 Safrantech's autonomous vehicle prototype

The autonomous vehicle developed at Safrantech and presented in Section 1.3.2 served as a test bed for the methods which were proposed during this thesis. In particular, thanks to the embedded sensors whose description is given thereafter, both global and local navigation could be evaluated

Epsilon 10 IMU The car is equipped with a high-grade IMU Epsilon 10 from the company Safran Electronics and Defense. The increments are acquired at 100 Hz. The propagation noise associated to the IMU is given by

$$\begin{aligned}
\sigma_\omega &= 0.1\ ^\circ/h \\
\sigma_a &= 0.0015\ m/s^2\ (0.15mg),\ \sigma_{b^\omega} = 2.5e-5\ ^\circ/s/\sqrt{s}, \\
\sigma_{b^a} &= 8e-4\ m/s^{-2}/\sqrt{s}.
\end{aligned}$$

RTK GPS The RTK antenna outputs positions which we consider with a constant uncertainty of $1m$.

LiDar A Velodyne VLP32C is mounted on top of the car. The 3D laser scans between keyframes were preprocessed to obtain relative transformations using scan matching algorithms. Scan matching algorithms such as iterative closest point (ICP) [20] return relative orientation and translation. As the gyro only drifts of less than a few degrees in 1 hour, the level of uncertainty associated to the relative rotation between LiDAR scans computed by the ICP is much higher than the gyro's uncertainty. Therefore, in the subsequent fusion it is as if only relative translations were computed. They were considered with an uncertainty of $10cm$.

6.1.2 Experimental tracks

The car acquired experimental data through a number of runs around the building of Safrantech. It drove around in a peri-urban area, including medium-speed portions ($90km/h$) and roundabouts. The length of the runs vary, but they all cover at least $10km$. Examples of two the experimental trajectories are given in Figure 6.1.

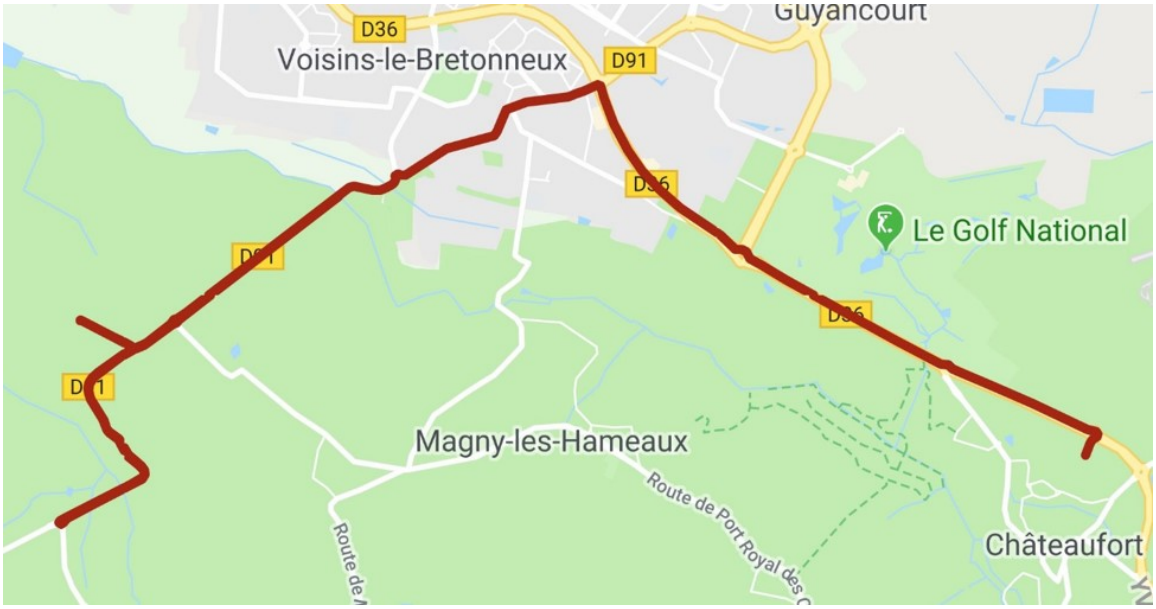


Figure 6.1: 11 km long trajectory followed by the car, taken from Google Maps.

Figure 6.2: 12.5 km long trajectory followed by the car, taken from Google Maps.

Usually the recording started while the car was still in the parking lot. In some cases, recording starts during the run. This allowed evaluating the estimators' capacity to correct their initial speed.

6.2 Inertia-GPS fusion experiment

This section presents the results obtained by various estimation methods on an inertia-GPS alignment problem, in which the car starts still with an unknown heading. This is similar to the case studied in Section 4.4.6, but with more accurate GPS data. This simulation example had outlined the usefulness of L-IS over the IEKF. That is why the focus is only put here on smoothing methods. Four of them are compared:

- L-IS
- L-IS using the retraction of GTSAM (see Section 4.4.4), called NoExp
- L-IS using the Jacobian (4.61), called Hybrid
- Smoothing based on [53]

Moreover, two sets of experiments were conducted. On the one hand, using the IMU true increments. On the other hand, with inflated biases. Indeed, the biases turned out to be quite low in the experiments, and the simulations of Section 4.4.6 showed that bigger biases (but still in the range of what could be expected) could harm the estimation. Therefore it seemed interesting to check how each method would behave in this case.

Since this was an alignment experiment, the chosen trajectory starts with the car standing still for 90s, before starting to move forward. Only the outputs of the estimation on the three first minutes of the run are reported. GPS measurements were taken every second. For all the experiments, the estimate is initialised at the first GPS measurement, with zero velocity, zero biases, roll and pitch as if the IMU lay on a flat floor. The state uncertainty was initialised accordingly,

$$\begin{aligned}
&P_0 = \mathrm{diag}((I_2, \sigma_R^0), \sigma_v^0\ I_3, \sigma_x^0\ I_3, \sigma_{b^\omega}^0\ I_3, \sigma_{b^a}^0\ I_3),\\
&\sigma_R^0 = 100\ ^\circ,\ \sigma_v^0 = 100\ m/s,\ \sigma_x^0 = 100\ m,\ \sigma_{b^\omega}^0 = 7e-4\ ^\circ/s, \sigma_{b^a}^0 = 7e-2\ m/s^2
\end{aligned}$$

For each experiment, the algorithms were compared in two setups, in batch, optimising the whole trajectory, and using sliding windows of various sizes.

6.2.1 Experiments using the actual IMU increments

First, the four smoothing alternatives were compared using the inertial increments actually recorded by the car. In this case, not many differences appeared, be it in batch or sliding window setting. Indeed, given the IMU precision and the weak biases encountered in practice, in the order of $0.04 m/s^2$, this alignment problem turned out to be fairly easy. Results are reported on Figures 6.3 (left) for the batch setting and 6.4 (top) for sliding windows. The latter gives the evolution of the heading error with respect to time for each estimator. In all cases the error stays at its initial value until the car starts moving, time at which it quickly converges to zero. The former confirms these good behaviors, since it shows similar converging speeds for each method. Looking at the details, one could see that L-IS maintains trajectories which seem to be more consistent during the iterations, similarly to, but not as striking as what could be observed in the 2D case in Section 4.3.3.

In terms of computational load, all three algorithms run in about the same time per GN step, see Table 6.1. Since they come from a shared implementation, it means that the small differences between them, e.g., using the Lie group exponential and logarithm, and the way the Jacobians are recomputed, even out. This confirms what had been observed in the 2D case in 6.1.

The results observed in the sliding window setting showed almost no impact of the length of the sliding window. This point was further investigated, in the case of L-IS only. The maximum distance between trajectories estimated with growing window sizes and the one associated to a window of size 1 was computed. This simple comparison scheme has the advantage of defining a norm. These distances were computed for the data of two different runs, and results are reported in Figure 6.6. The conclusions are clear: the window size has a limited impact on the output, and, beyond a certain width, expanding the

	From [53]	Invariant	Hybrid
GN iteration (s)	0.098	0.10	0.11

Table 6.1: Mean computation time of a GN step of smoothing algorithms with 150 states

window is completely ineffective. This could be expected: being a global navigation problem, the impact of past states gradually fades over time. The propagated covariance carries most of this information, and as soon as enough GPS measurements are available to calibrate the biases, adding more of them into the state should not impact it.

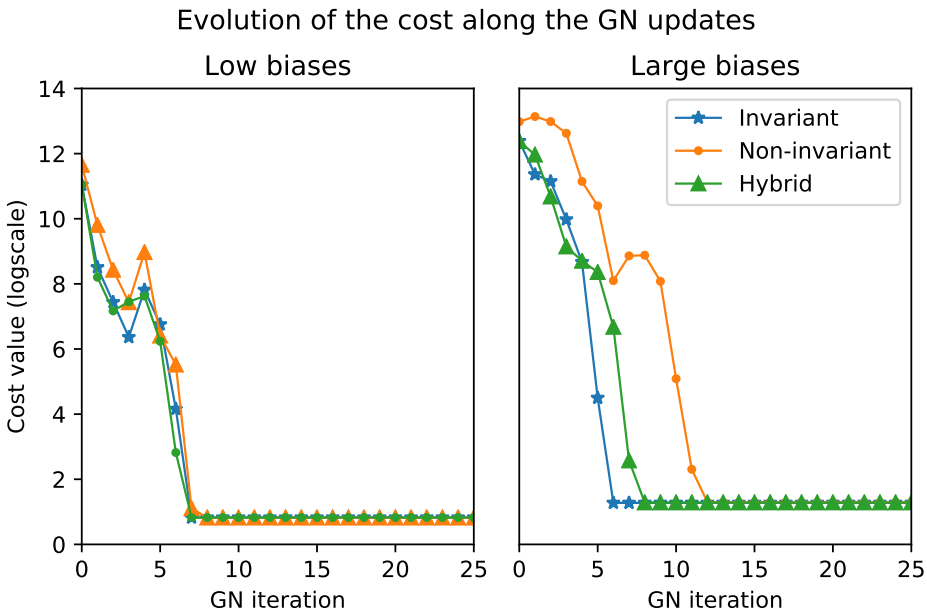


Figure 6.3: Evolution of the cost associated to each smoothing method along the GN iterations, in log scale. Left: With the IMU's true measurements, the three methods behave similarly. Right: With inflated bias and drift, Invariant and Hybrid smoothing converge much faster than the non-invariant method, with a slight advantage for the first one.

6.2.2 Stronger biases could degrade performances

As outlined in the previous section, the actual biases of the IMU equipped in the car are quite low compared to what could be expected. Note that IMUs are not considered to be high-grade when their biases are low, but according to their stability. This is consistent with the framework developed herein: having stable biases means that, once they are calibrated, the system gets closer to the ideal case of group-affine dynamics given in Section 2.3. Therefore, the same alignment experiment was conducted after inflating each increment with a constant additional bias, of $0.1m/s^2$ and $0.001°/s$.

These inflated biases strongly affected the estimation. This can be seen in batch setting first, in which case the parametrisation of [53] has a much harder time converging, than L-IS and the Hybrid versions. Between the two latter, L-IS reaches the optimum slightly faster, as Figure 6.3(right) shows.

The most interesting, and intriguing, behaviors are observed in the sliding window setting. It is reported in Figure 6.4, on the rows below the top one. It shows the evolution of the heading error with respect to the chosen algorithm, the window size, and the number of GN iterations carried out at each step. Far from staying still as in the previous section, if the window exceeds a certain size, erratic behaviors emerge.

This originates from the fact that, during the still phase, the system is in fact not observable, but the measurement noise make it appear very weakly observable. Therefore the cost function may have very wide plateaus, and the first GN iterations may be unstable. This clearly appears on Figure 6.4,

middle, which represent the yaw error along the trajectory. All methods look unstable in this part, but this is not necessarily a failure: since the car has not moved, the covariance remains large. Therefore, most estimators quickly converge once the car accelerates, although not all of them. For instance, that of [53] with a window of size 30 ends up stuck in a local minima after the convergence phase. More GN iterations allow the estimators to reduce their volatility. However, this too not always improves the estimator's behavior nor its convergence. While L- and H-IS are stabilised, non-invariant smoothing gets even worse. If the window reaches a sufficient size, however, all smoothing methods manage to carry out the estimation, except for NoExp, which still fails if too many GN iterations are carried out. Interestingly, in this experiment H-IS appeared superior to L-IS.

Contrary to the experiment using the actual increments, the window size has a noticeable influence on the results, for all methods. However, this impact seems to be located around the converging phase. To assess this, along with the maximum distance to the trajectory estimated by the estimator of length 1, the distance to the final point is also computed. The evolution of both metrics with respect to the window size is given in Figure 6.6. The effect on the maximum distance is clear on at least one of the trajectories. However, in all cases, the final distance between each estimated trajectory and that of the estimator of length 1 is negligible. It thus appears that, depending on the setting, Invariant Smoothing does not bring noticeable improvements over Invariant Filtering. This is not always the case, as outlined in Section 4.4.6. However, this should be an important information for practitioners: *choosing the right parametrisation is more important than expanding the sliding window in global navigation.*

6.3 Fusing inertia and relative translations

This section presents the results obtained by various estimation methods given an inertia-relative-translations fusion system. Its goal is to illustrate the differences between L-IS, R-IS, and [53]. The relative translations between states i and j follow the standard model

$$t_{ij} = R_i^T (t_j - t_i) + v_{ij}, \tag{6.1}$$

with v_{ij} a white noise.

The laser scans were transformed into relative translations obtained at 4 Hz.

Since it is a local navigation problem, the initial state is taken as the origin of global frame, the biases and initial velocity are set to zero, and the uncertainty was initialised the following way,

$$\begin{aligned} &P_0 = \mathrm{diag}(\sigma_R^0 I_3, \sigma_v^0\ I_3, 0, \sigma_{b^\omega}^0\ I_3, \sigma_{b^a}^0\ I_3), \\ &\sigma_R^0 = 1\ ^\circ,\ \sigma_v^0 = 100\ m/s,\ \sigma_x^0 = 0.1\ m,\ \sigma_{b^\omega}^0 = 7e-4\ ^\circ/s, \sigma_{b^a}^0 = 7e-2\ m/s^2 \end{aligned}$$

The initial heading uncertainty was not set exactly to 0, as the quality of the gyroscope makes it potentially sensible to the Earth's rotation, which gives hints about the heading after some time.

Remark 17. *This experiment does not consider H-IS for the sake of readability. Indeed, it is fairly close to L-IS and shares the same false observability issues. Therefore, it behaves almost identically and presents no particular interest here.*

6.3.1 Filtering vs Smoothing

Filtering is a particular case of smoothing, with a sliding window of length 1 (see Section 4.2.4). Therefore we focus on the more general question of the impact of the window size on the estimation. To this end, we compute the maximal distance between the trajectories estimated by R-IS with window lengths growing from 1 (filtering) to 50 with the filtering estimate, for two different runs, on Figure 6.7. In both cases, it clearly appears that the impact of expanding the window decreases with its length. Moreover, the distance between the trajectories reaches $7m$ in one case, and about $2m$ in the other, on runs of about $3km$. In turn, the estimation varies of 0.2% and 0.06% respectively. Considering the increase in computational cost, filtering seems to be a good compromise in this case.

6.3.2 Assessing the importance of the right-invariant parametrisation

As exposed in Section 5.2.4, R-IS is a consistent local navigation estimator while L-IS is not. This strong discrepancy between both invariant estimators must be illustrated, as must the performance of

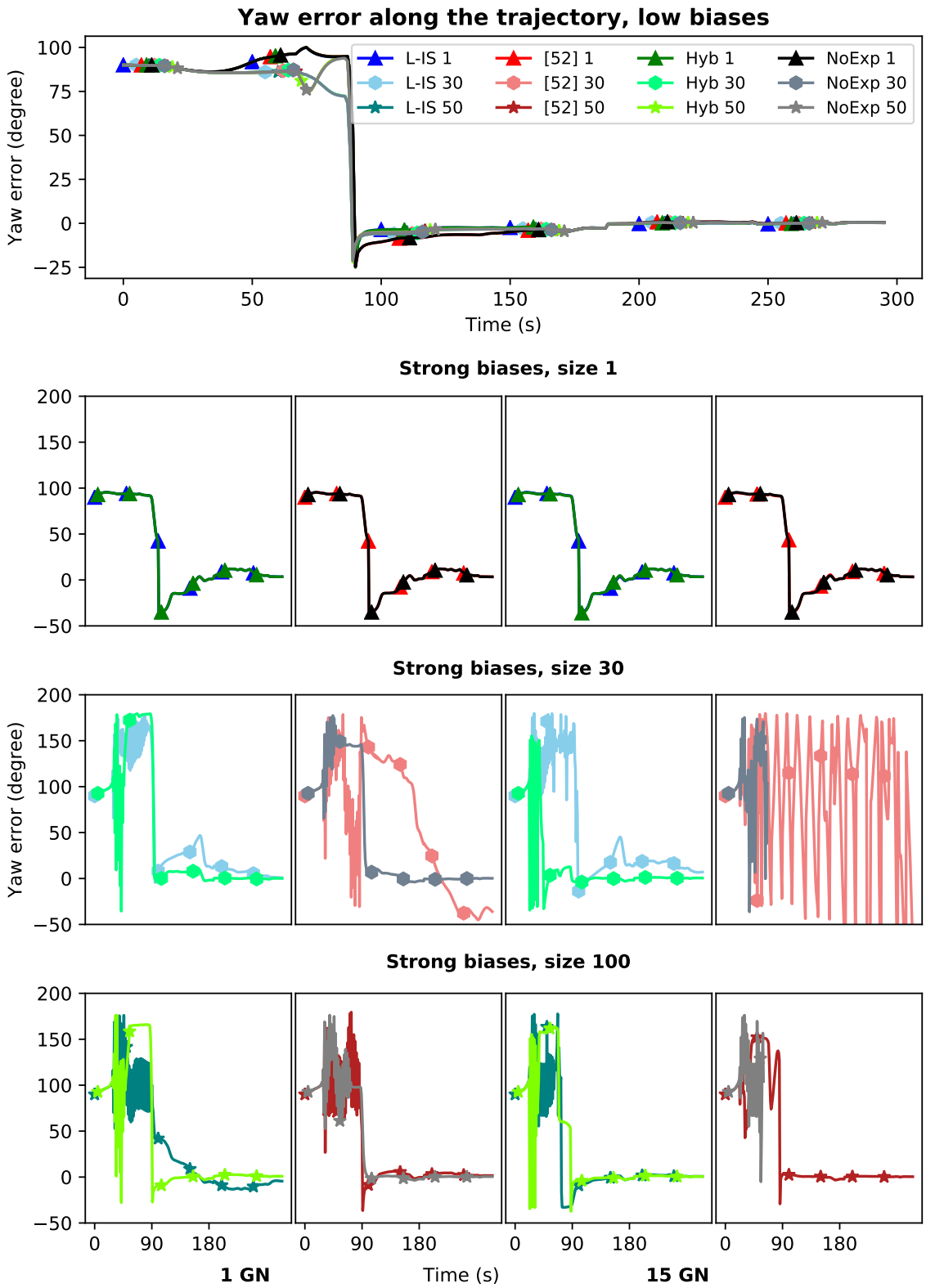


Figure 6.4: Evolution of the yaw error along the trajectory for four different smoothing methods, two invariant and two non-invariant, with respect to the window size. First row: with the IMU's actual increments. All smoothing methods perform almost equivalently, whatever the window size. The other rows report the alignment estimation with inflated biases. They must be read as follows: each row shows the result for a given window size (top: 1, middle: 30, bottom: 50). The two left-most and the two right-most columns show what happens for 1 GN and 15 GN iterations per step respectively. In each case, the left panel displays both invariant versions, and the right one the non-invariant ones. For a window size of one, L-IS and Hyb coincide, and so do [53] and NoExp. In the presence of these biases, the still phase looks unstable, because discerning heading error from bias error is hard. Unexpectedly, expanding the window can destabilise the estimation and degrade the results. Increasing the number of GN iterations can temper the oscillations, but may also lead to the non-invariant methods diverging and even failing at some point, e.g., NoExp with window size 30 and 50 and 15 GN iterations. The exponential update thus seems crucial.

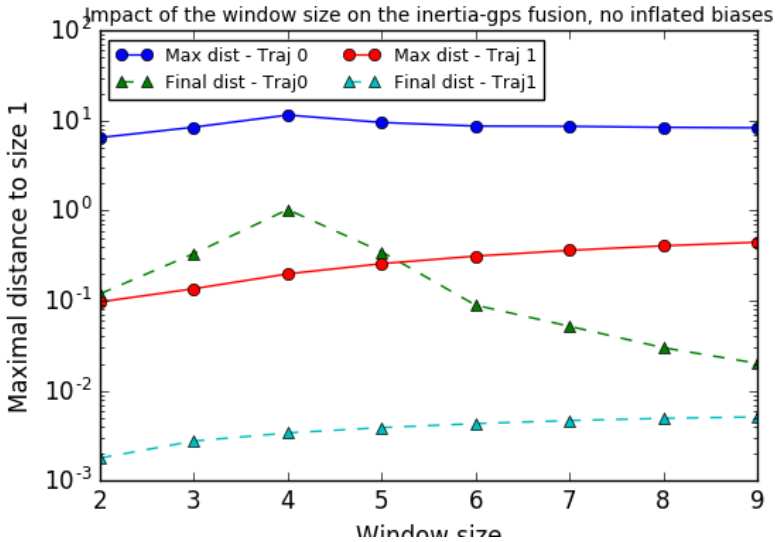


Figure 6.5: Evolution of the maximal and final distances between L-IS estimates using various window sizes and the estimate using size 1, in logscale. The result on two different runs are displayed. In a case, the maximal slightly grows but stays at low values. In the other, it directly starts at around $5m$, and stays stable. In both cases, the final distances are quite negligible, between 1 and $50cm$, indicating that the difference between the methods narrow once the first transitory phase is finished.

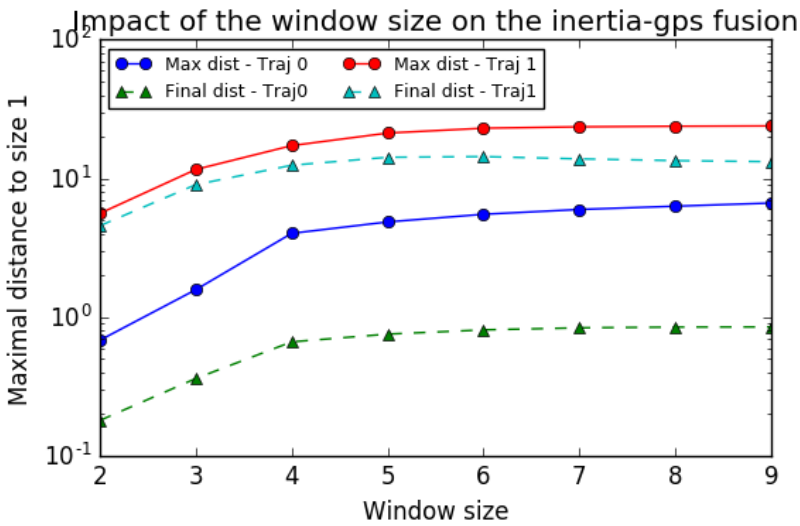


Figure 6.6: Evolution of the maximal and final distances between L-IS estimates using various window sizes and the estimate using size 1, with inflated biases, in logscale. The result on two different runs are displayed. The maximal difference grows but quickly reaches a limit value. However, the final distance always stays at much lower values, indicating that the difference between the methods narrow once the first transitory phase is finished.

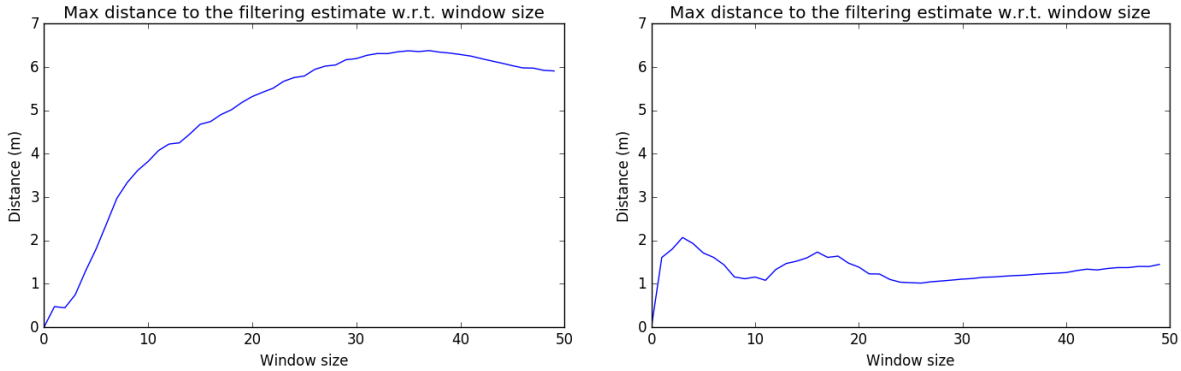


Figure 6.7: Impact of extending the window size on the estimated trajectory in local navigation. For two different experimental runs, the maximum distance between the outputs of a sliding window estimator with window 0 (i.e. with a single active state) and others with growing windows is computed. In both cases, the impact stays below 0.2%, and seems to converge, or at least stay bounded with the window width.

R-IS compared to a non-invariant but consistent estimator, based on [53]. To this end, the LiDar-inertial odometry (LIO) estimate is computed with the data from two different runs of the experimental car. This time, a sliding window of size 10 was considered. To isolate the effect of the right-invariant parametrisation, the smoothing scheme based on [53] was made using an exponential update. The GPS measurements are also given as a reference. The results are displayed on Figure 6.8 and 6.9, which respectively compare the three methods on the first four minutes of each run, and only R-IS and [53] on one of the full runs. On the former, it clearly appears L-IS provides a poor estimation, confirming its inconsistency exhibited in Section 5.2.4. This most probably comes from the observation Jacobians, however this may come from the fact that it is the least accurate sensor. Indeed, considering the IMU's characteristics, it is likely that both version of the Jacobians given in (4.60) and (4.61) are close to one another. R-IS and [53] (using the exponential udpate) seem equivalent during the first four minutes of both runs. They do not on the long term though, as clearly appears on the latter figure, on which the trajectory estimated by [53] slowly turns inconsistent and ends up being strongly degraded.

Remark 18. *We wanted to compare directly the results of the estimate given by GTSAM. However, despite our best efforts, we did not manage to obtain a stable estimation. Indeed, using the Levenberg-Marquardt solver provided by the toolbox was the only way to get a somewhat stable estimate, but this required too much computations for the whole trajectory. Switching to iSAM2 made the estimation diverge after less than 10 measurements. We cannot guarantee that we correctly used the toolbox, and although no definitive conclusion can be drawn, it would seem that iSAM2 has trouble dealing with this level of precision.*

6.4 Conclusions of the experiments

The experiments carried out on the autonomous vehicle prototype of Safrantech allowed evaluating the interest of one of the main contributions of this thesis, Invariant Smoothing. It brought to light the following conclusions regarding navigation with precise inertial sensors

- When dealing with strong biases in global navigation, the exponential update used by Invariant Smoothing makes it much more stable.
- Further experiments must be carried out to explore the differences between L-IS and H-IS.
- R-IS proves far superior than L-IS in local navigation due to its consistency properties. Moreover, it also shows higher stability on the long run compared to other consistent, but non-invariant, smoothing methods. However, two different explanations can be given: a better handling of the Earth's rotation, or the system's observability properties seen from the smoothing point of view as in Section 5.2.2.

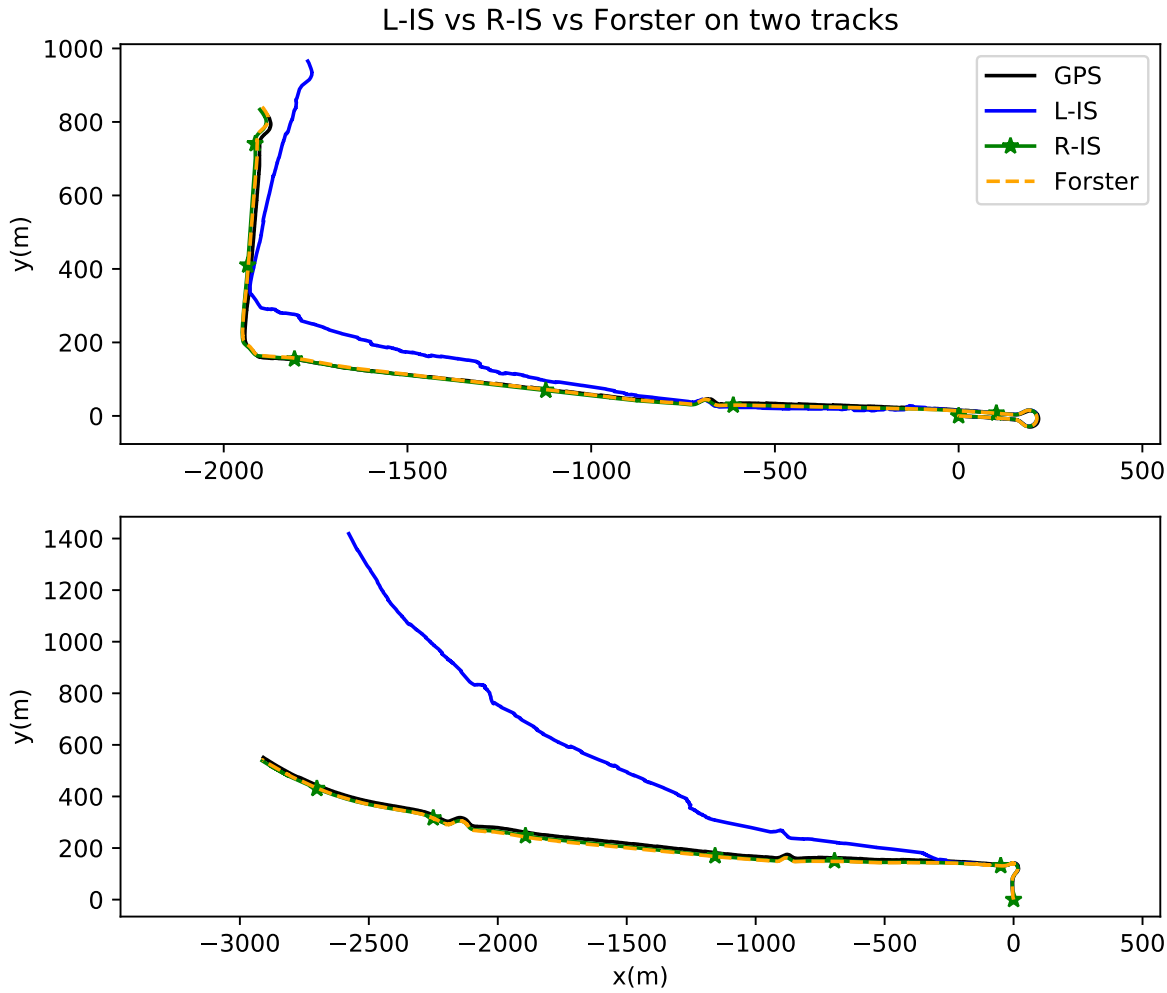


Figure 6.8: Comparison of the consistency of LIO methods on two different experimental runs. The left-invariant parametrisation used by L-IS leads to a strongly degraded behavior, illustrating its false observability issues. On the contrary, R-IS and [53] seem to have highly similar behaviors at this point.

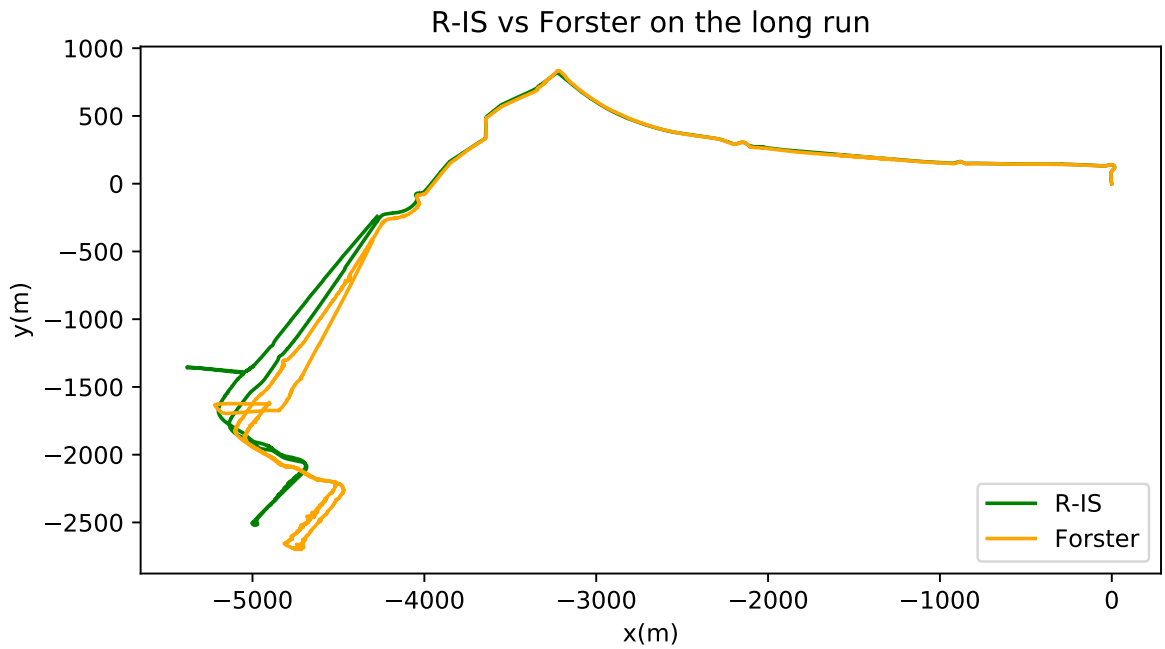


Figure 6.9: Comparing the behaviors of R-IS and [53] parametrisations on the long-term consistency of LIO. The effects appear only after several minutes of drive. There are two potential reasons for this: consistency of the method and the Earth's rotation, or even an interplay of both.

- These experiments also raised the question of the interest of expanding the sliding window in smoothing. Indeed, it sometimes had an impact, e.g., in global navigation with strong biases, however in many cases its effect turned out negligible, meaning that it was mainly a waste of computational power.

Part II

Navigation with highly precise sensors

Chapter 7

The shortcomings of standard smoothing with highly precise sensors

Highlights

- Smoothing methods are based on linear solvers which take advantage of the problem's sparsity to be efficient.
- They are all based on the precision matrices of the sensors. With the improvements of the IMU, this can lead to ill-conditioned information matrix, and in turn degraded estimation.

Introduction (En/Fr)

In this chapter, a shortcoming of the smoothing methods that has gone unnoticed yet to our best knowledge is raised. Indeed, all the mentioned formulations are based explicitly on the information matrix of each factor - the inverse of the covariance matrix - whose eigenvalues tend to increase as the sensors improve. But a large discrepancy between the most accurate and least accurate sensors leads to ill-conditioned information matrices. Numerical issues may thus degrade the solvers' computed solution, and in turn the state estimate accuracy and consistency.

This realisation came from a simple observation: it is, with the current methods, impossible to extend to the smoothing case the results of Section 3.3, on filtering with constraints. Indeed, while this surely was a limit case, it brought out the fact that the KF never requires to inverse process noise covariance matrices, allowing simplified implementations on hardware with precision as low as 8-bits processors [84]. However, the cost of high performance inertial measurement units (IMU) keeps decreasing, and one may anticipate that high-grade IMUs will be used in an increasing number of autonomous systems. This prompts the need for solvers being robust to ill-conditioning of the information matrix.

Ce chapitre soulève une limite des méthodes de lissage qui, à notre connaissance, est passée inaperçue jusqu'ici. En effet, toutes les formulations sont basées explicitement sur la matrice d'information de chaque facteur - l'inverse de la covariance - dont les valeurs propres ont tendance à croître avec l'amélioration des capteurs. Mais un grand écart entre les capteurs les plus et les moins précis peuvent créer des matrices d'information du système mal conditionnées. Les erreurs numériques peuvent dès lors dégrader la solution calculée par les solveurs, puis la précision et la cohérence de l'état estimé.

Ce constat est venu d'une simple observation : il était, en l'état, impossible d'étendre au lissage les résultats de la Section 3.3, sur le filtrage avec contraintes. En effet, bien que ce soit évidemment un cas limite, il souligne le fait que le KF ne nécessite jamais l'inversion de la covariance du processus, permettant son implémentation sur du matériel avec des processeurs de précision descendant jusqu'à 8-bits [84]. Cependant, le coût des IMU de haute performance continue de décroître, et on peut anticiper qu'elles pourront être utilisées dans un nombre croissant de systèmes autonomes. Des solveurs robustes au mauvais conditionnement de la matrice d'information sont donc nécessaires.

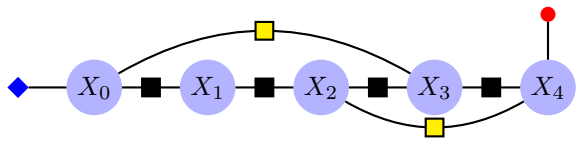


Figure 7.1: MAP estimation as a factor graph: Blue circles denote the successive states of the vehicle $X_0, X_1, \cdots, X_N$ through times $0, 1 \cdots, N$. Factor nodes shown in black involve successive states and correspond to dynamical model relations (7.3). Factors nodes shown in orange correspond to measurements involving a pair of state variables. The factor nodes shown by a red circle and a blue diamond respectively represent unary factors (the latter being a prior on the state)

7.1 Standard factor graph based smoothing through MAP

7.1.1 Main considered problem

Assume we want to track the state of a system (say, a vehicle) equipped with an IMU and/or wheel odometry, as well as a set of sensors such as cameras, LiDARs, GNSS, acoustic positioning system for UAVs. The current formulation of SLAM uses the formalism of factor graphs to cast the maximum likelihood estimation problem as a nonlinear optimisation problem. Up until Section 8.3, we will not consider landmark-based SLAM, but rather "pose SLAM", where vision sensors provide relative poses between key frames. Moreover, and contrary to SLAM, we perfectly allow absolute sensors such as GNSS to provide information about the vehicle, and refer to SLAM as the obtained extended multi-sensor fusion problem, mainly for localisation and navigation purposes. See Figure 7.1 for a visualisation of such a system.

Following Section 4.1, the estimate is obtained by minimising the non-linear quadractic function associated to the measurements (4.9). Adopting a GN optimisation scheme, one must compute at each step the minimiser of a single quadratic linearised cost function

$$\delta\chi^* = \underset{\delta\chi}{\operatorname{argmin}}\, \bar{L}(\delta\chi), \quad \text{where } \bar{L}(\delta\chi) = ||A\delta\chi - b||^2_\Sigma, \tag{7.1}$$

whose solution satisfies the normal equation

$$(A^T\Sigma^{-1}A)\delta\chi^* = A^T\Sigma^{-1}b. \tag{7.2}$$

This is classically done through Cholesky or QR factorisation.

Here, the state is a set of all variables that characterise the vehicle and its future evolution, typically its orientation, position, velocity, and all parameters of interest such as IMU biases, camera to IMU transformation parameters. The state at time k is denoted X_k. In particular, we split the measurements in two groups. First those coming from a dynamical model giving the state's evolution

$$X_{k+1} = f(X_k, u_k, w_k) \tag{7.3}$$

with w_k a random variable (noise) encoding model uncertainties, and u_k a measurement typically output by wheel speed sensors or the IMU. Then the observations $Z_1, Z_2, \cdots, Z_K$ of the form $Z_k = h_k(\chi_k, V_k)$ where $\chi_k \subset \chi$ is a subset of the states $X_1, X_2, \cdots, X_N$. As they are the ones which appear most in inertial navigation, visual inertial odometry, and inertial pose SLAM, we focus on unary and binary observations respectively given by:

- Unary observation at time k
$$Z_k = h_k(X_k, V_k) \tag{7.4}$$
- Binary observation at time k
$$Z_{ik} = h_k(X_i, X_k, V_{ik}), \quad i < k \tag{7.5}$$

The binary measurements appearing in pose graph SLAM are related to the computation of relative transformations between poses at different times, typically using stereo cameras or LiDARs. When they are not taken as propagation (for instance in visual odometry), they are usually called "loop closures" [49].

To simplify the notation, let $Z_k = h_k(\chi, V_k)$ denote both types of observations (7.4) and (7.5) at time k. Finally $Z = (Z_1, \cdots, Z_K)$ denotes all available measurements.

We also separate the prior from the other measurements. Generally, the initial uncertainty about the state X_0 is assumed Gaussian, i.e., $X_0 \sim \mathcal{N}(\hat{X}_0, P_0)$.

Therefore, in this context the likelihood of the posterior from (4.9) becomes

$$\begin{aligned} L(\chi) :&= -\log p(\chi \mid Z) = ||X_0 - \hat{X}_0||^2_{P_0} \\ &+ \sum_{n=1}^{N-1} ||\phi_n(X_{n+1}, X_n, u_n)||^2_{Q_n} + \sum_{k=1}^{K} ||\psi_k(\chi, Z_k)||^2_{R_k}. \end{aligned} \tag{7.6}$$

7.1.2 Resolution of the linearised optimisation problem

At each step the algorithm is faced with the resolution of the linearised optimisation problem (7.1). This is a standard least squares problem, and the solution comes in closed form, given by the normal equations (7.2). Solving them is the main task to be addressed when implementing factor graph optimisation. Indeed, one is faced with a system of linear equations. Most, if not all, of the smoothing solvers rely on factorisations of the information matrix defined by $\mathcal{I} := A^T \Sigma^{-1} A$ in the form of $A^T \Sigma^{-1} A = LL^T$ with L lower triangular (Cholesky) or through a QR factorisation. Matrix L is referred as the "square root" of the information matrix, and allows to solve the problem by first solving $L\eta = A^T \Sigma^{-1} b$ and then $\eta = L^T \delta\chi^*$ by back-substitution, see [45]. Using this square-root creates algorithms which are numerically more stable, and even enabled implementations with single precision on mobile devices [103]. However, even square-root formulations are inherently limited by the inverses of the covariances appearing in the formulas, and there is no *continuum* between actually zero covariances, which represent hard constraints, and numerically invertible covariances.

7.2 The limits of factor graphs and MAP

7.2.1 Potential issue n°1: computational complexity

A is a matrix having size $((N+1)d_x + Kd_z) \times ((N+1)d_x)$ where d_x, d_z are the dimension of the state and observations, K is the number of observations (we assume for simplicity all observations have identical dimension). In principle, the problem would not be tractable for localisation applications. However, the sparsity of A stemming from the particularity of the problem estimation structure allows smoothing solvers encode various ways of inverting this equation. For instance, [73] embeds the problem in a particularly well adapted Bayes tree structure which allows for local constant-time updates at each time step by reusing previous factorisations. In a more recent work, [78] uses preconditioned conjugate gradient to invert the problem, with a high efficiency gain coming from a number of computational tricks specific to the SLAM problem. These methods usually prefer working with an equivalent formulation of (7.1):

$$\bar{L}(\delta X) = ||\tilde{A}\delta\chi - \tilde{b}||^2$$

with $\tilde{A} = \Sigma^{-1/2} A$ and $\tilde{b} = \Sigma^{-1/2} b$, where $\Sigma^{-1/2}$ is the square-root of Σ^{-1}. However, both formulations explicitly rely on the inverse of the covariance matrices at play.

7.2.2 Potential issue n°2: ill-conditioned information matrix

The complexity issue is obvious, and much efforts in the SLAM back-end community have been devoted to it. Another potential issue that has scarcely been addressed so far, and that we bring forward in the present chapter, is as follows. If the normalisation matrix $(A^T \Sigma^{-1} A)$ is ill conditioned because Σ has very small eigenvalues (or even null eigenvalues), then measurement noise is amplified and the solution may become grossly inaccurate. In the present work, focus is put on the case where dynamics (7.3) rely on high precision inertial sensors, which are becoming increasingly common, and lead to very small covariance matrices.

Assume the covariance matrix Q_i has very small eigenvalues. As $Q \to 0$, we see that $\Sigma^{-1} \to \infty$ and thus

$$\begin{aligned} (A^T \Sigma^{-1} A) &\longrightarrow \infty, \quad (A^T \Sigma^{-1} b) \longrightarrow \infty, \\ \Sigma^{-1/2} A &\longrightarrow \infty, \quad \Sigma^{-1/2} b \longrightarrow \infty. \end{aligned}$$

But this does not mean that the quantity of interest $\delta\chi^*$ degenerates in the same way, as we have an "$\frac{\infty}{\infty}$" undetermined from (7.2). And in fact, it does converge to a finite value $\delta\chi_\infty$ which is simply the result of a lower-dimensional problem: we will prove in the next section that when $Q_i \to 0$ we have

$$(A^T\Sigma^{-1}A)^{-1}A^T\Sigma^{-1}b \longrightarrow \delta\chi_\infty, \tag{7.7}$$

Ideally, we would like the solver to be such that $\delta\chi^* \to \delta\chi_\infty$ when $Q_i \to 0$, and also to be able to find $\delta\chi_\infty$ even in the degenerate case where $Q_i = 0$. However, we cannot expect such a desirable behavior from a solver directly based on the normal equations (7.2). This is why we advocate in the present paper a different approach based on the Kalman smoother to solve the linearised optimisation problem (7.1).

The limits of the square-root approach

Using the square-root approach of the information matrix reduces the impact of conditioning. Numerical issues related to it depend on the precision of the hardware used, though. Indeed, single precision implementations will much more suffer. This is noticeable as there is a need to speed up computations for real-time applications, for which single precision is generally considered [40,103], and most industrial aerospace computers still use single precision (32 bits or less). Besides, navigation methods relying on machine learning to improve some of their bricks [31,32], mimicking Kalman filters [60] or even smoothing [42], are on the rise, and some recent methods even rely on Gauss-Newton as one of their bricks [82]. They could benefit from single precision algorithms to speed up both their training phase, and their inference.

Most, if not all, of the smoothing solvers use the information form of the MAP problem, in order to take advantage of its sparsity [2,45,78]. It has been long recognised that using the square-root, for instance the Cholesky factorisation of the information matrix, leads to algorithms which are numerically more stable, and even enabled implementations with single precision on mobile devices [103]. This lead to highly efficient incremental implementations [73,78]. However, even square-root formulations are inherently limited by the inverses of the covariances appearing in the formulas, and there is no *continuum* between actual zero covariances, which represent hard constraints, and numerically invertible covariances.

7.3 A solution without matrix inversion: the robust batch solver

Assume the noise matrix of the model Q should not be inverted. Then, Lemma 1 ensures that, given a split into two parts of (7.1) with $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$, $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, and $\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}$, where A_1 is square and invertible (it may be completed with additional factors to make it square if need be), the minimum is found as:

$$\begin{aligned} \delta\chi^* &= A_1^{-1}((I - KJ)b_1 + Kb_2), \\ K = \Sigma_1 J^T(J\Sigma_1 J^T + \Sigma_2)^{-1}, &\quad J = A_2 A_1^{-1}. \end{aligned} \tag{7.8}$$

Here, Σ_1 contains the covariance which should not be inverted. Therefore we are now in a position to prove the result announced in equation (7.7).

Lemma 2. *As $Q \to 0$, the solution $\delta\chi^*$ to (7.2) tends to a finite value.*

Proof. Replacing in (7.8) K and J by their values we obtain:

$$\delta\chi = A_1^{-1}\left[b_1 + \Sigma_1 A_1^{-T} A_2^T (A_2 A_1^{-1}\Sigma_1 A_1^{-T} A_2^T + \Sigma_2)^{-1}(b_2 - A_2 A_1^{-1} b_1)\right].$$

We see $(A_2 A_1^{-1}\Sigma_1 A_1^{-T} A_2^T + \Sigma_2) \to (A_2 A_1^{-1}\Sigma_\infty A_1^{-T} A_2^T + \Sigma_2)$ when $\Sigma_1 \to 0$, and this latter term is lower-bounded by Σ_2, so there is no indetermination in (7.7): δX_∞ exists. □

As it considers all measurements at once, this solver is called herein the batch solver. This solution may become intractable as it requires the full inverse of A_1, and the size of the matrix A_1 is quadratic in the size of the trajectory. According to (8.1), A_1 has a lower block-triangular structure, and its inverse may be obtained analytically, but the number of linear systems to solve because of the term $J = A_2 A_1^{-1}$ still is a major caveat of this method. It is however used in Chapter 9 as a reference for comparison

purposes. In the next sections we investigate an approach based on Kalman smoothers, which reaches identical result, but with a complexity being linear in the length of the trajectory.

This batch solver still proves that it is possible to solve (7.6) without inverting the Q_i matrices, and therefore become robust to this form of ill-conditioning. There still remains to find a scalable solver, which will exhibit the same property, but will also take advantage of the sparsity of the problem. This is the goal of Chapter 8

Chapter 8

SC-BIFM : A new solver based on the Kalman smoother and stochastic cloning

Highlights

- A new linear solver is proposed to deal with some ill-conditioned information matrices. It is based on the link between the linear Kalman smoother and least-squares.
- A recent formulation of the smoother, called Backward Information Forward Marginal (BIFM), is used, which avoids inverting the dynamics' covariance. A more intuitive derivation of this smoother is given.
- BIFM is extended to n-ary measurements thanks to Stochastic Cloning (SC), leading to the proposed solver, SC-BIFM.
- Numerical simulations validate the proposed methods, and outline the limits of the standard ones.

Points marquants

- Un nouveau solveur linéaire est proposé, afin de traiter certaines matrices d'information mal conditionnées. Il est basé sur le lien entre le lisseur de Kalman linéaire et les moindres carrés.
- Une nouvelle formulation du lisseur, dite Information à rebours Marginale en avant (BIFM), est utilisée, évitant d'inverser la covariance de la dynamique. Une dérivation plus intuitive de ce lisseur est donnée.
- Le BIFM est étendu aux mesures n-aires grâce au clonage stochastique (SC), menant au solveur proposé, SC-BIFM.
- Des simulations numériques valident la méthode proposée, et soulignent les limites des solveurs standards.

Introduction (En/Fr)

The issue of ill-conditioned systems has been addressed in the signal processing community by devising new formulations of the Kalman smoother. Additionally to the well-known Modified-Bryson-Frazier Smoother [21], a new alternative has been derived, the so-called Backward Information Forward Marginals (BIFM) Smoother, which also avoids covariance inverting [79,101]. Nevertheless, they were only expressed for acyclic graphs, which only cover a small part of the existing factor graphs. A more general smoother, able to solve a broader class of linear least-squares, is introduced here by applying the stochastic cloning method of [85] to the BIFM. Furthermore, BIFM is presented in an original way, which can seem more intuitive, as it is explicitly based on the Kalman equations, for the forward sweep, the backward one, and the final fusion. Thanks to the correspondence between Bayesian inference and least-squares in the linear

case, this allowed deriving, after the robust batch approach of 7.3, a second new exact way of solving well-posed linear least-squares problems while avoiding issues related to ill-conditioning. In addition to its capacity to handle low propagation noise, it also has a computational complexity equivalent to the current solvers, as it respects the intrinsic sparsity of navigation problems.

The majority of this chapter focuses on pose-graph like problems. A generalisation of this new solver is given at the end, exhibiting the link with the batch version.

The interest of the method is illustrated using a simple toy example. Experiments on real data, illustrating the interest of the approach in the context of highly precise motion sensors, are presented in Chapter 9.

Les problèmes liés au mauvais conditionnement de systèmes ont été résolus dans la communauté du traitement du signal par la conception de nouvelles formulations du lisseur de Kalman. En plus du lisseur Bryson-Frazier-Modifié [21], une nouvelle alternative, appelée lissuer Information à rebours Marginale en avant (BIFM), évitant l'inversion des covariances [79,101]. Néanmoins, ces formulations n'ont été mises en place que pour des graphes acycliques, qui ne couvrent qu'une faible part des graphes factoriels existant. Un lisseur plus général, capable de résoudre une classe de systèmes linéaires plus large, est introduit ici via l'application de la méthode du clonage stochastique de [85] au BIFM. De plus, le BIFM est présenté d'une manière originale, pouvant apparaître plus intuitive, car explicitement basée sur les équations de Kalman pour le passage avant, celui à rebours, et la fusion finale. Grâce à la correspondance entre l'inférence bayésienne et les moindres-carrés dans le cas linéaire, cela a permis, après l'approche groupée robuste de la Section 7.3, de dériver une seconde manière exacte de résoudre le problèmes de moindres-carrés linéaires bien posés évitant ces problèmes liés au mauvais conditionnement. En mplus de sa capacité à gérer un faible bruit de propagation, sa complexité algorithmique est équivalente à celle des solveurs actuels, car il respecte la parcimonie intrinsèque aux problèmes de navigation.

La majorité du chapitre se concentre sur les problèmes de type graphe de poses. Une généralisation du nouveau solveur est donnée à la fin, exhibant le lien avec la version groupée.

L'intérêt de la méthode est illustrée sur un exemple jouet. Des expériences sur données réelles, illutrant l'intérêt de l'approche dans le contexte de capteurs de mouvement précis, sont présentées dans le Chapitre 9.

8.1 The linear Kalman smoother as a least-squares solver

Consider the problem of finding the minimum of cost function $\bar{L}$ given by (7.1). In particular, in the case of a pose-graph like system, the measurements can be ordered such that A has the following structure:

$$A = \begin{pmatrix} I & 0 & \dots & \dots & 0 \\ -F_1 & I & \dots & \dots & 0 \\ \dots & \dots & \dots & -F_n & I \\ & & H_1 & & \\ & & \vdots & & \\ & & H_K & & \end{pmatrix}, \tag{8.1}$$

i.e., a prior on a first state and binary factors which serve as propagation. In this section we will recall the Kalman smoother may serve as a solver to this optimisation problem, albeit (for now) in the particular case of **unary** observations.

Remark 19. *The linearisation* (8.1) *still holds when the Jacobian of the propagation $i \to i+1$ with respect to $\delta\chi_{i+1}$ is not identity, as it should be invertible and could therefore be simply put on the other side of the equation.*

8.1.1 The standard Kalman smoother

Consider the following linear system with unary observations:

$$\begin{aligned} X_{k+1} &= F_k X_k + u_k + w_k, \\ Z_k &= H_k X_k + V_k \quad \text{(unary observations)} \end{aligned} \tag{8.2}$$

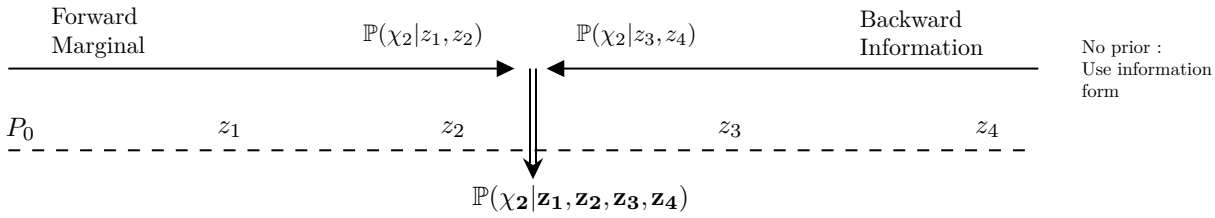


Figure 8.1: Schematised illustration of the idea behind BIFM.

It is easily proved that $\bar{L}$ given by (7.1) is related to this linear system as

$$\bar{L} = -\log p(X_0, \ldots, X_N \mid Z_1, \ldots Z_N) + cste.$$

A direct corollary of Theorem 4 is that the linear Kalman smoother computes the MAP estimate which minimizes $\bar{L}$, and thus may output the quantity of interest $\delta\chi^*$ that is required at each linearisation step of the more general Gauss-Newton solver. Its implementation is generally based on a forward and backwards recursion.

8.1.2 The Backward Information Forward Marginal (BIFM) variant of the Kalman smoother

In this work, we propose to use an alternative formulation of the Kalman smoother equations, the Backward Information Filter Forward Marginal (BIFM), since we want to avoid inverting the states' covariances [79]. This approach has already been used for message passing in Gaussian factor graphs in signal processing, but only for acyclic graphs. Another version of the Kalman smoother, namely the Modified-Bryson Frasier smoother, also exists [21]. Square-root forms of both algorithms have been produced and were compared on various signal processing problems in [101]. BIFM appeared slightly superior in these cases. Moreover, it can be derived in a very Kalman-filter-like manner which does not require writing the inverse of the covariance matrices Q_n nor the inverse of the forward covariance matrix P_n, as we are about to prove.

Consider the system (8.2). BIFM is based on the following approach. Suppose $P(X_k|Z_0, \cdots, Z_k) \sim \mathcal{N}(x_k^f, P_k^f)$; and likewise $P(X_k|Z_{k+1}, \cdots, Z_N) \sim \mathcal{N}(x_k^b, P_k^b)$. Here the superscripts stand for forward and backward. Then x_k^b may be considered as a measurement of X_k with noise covariance P_k^b, and treated as such in a Kalman update to merge it with (x_k^f, P_k^f) and obtain a final estimate $\hat{X}_k$. This is simply schematised on Figure 8.1. As there is no prior for the backward phase, that is, the prior is "flat", it must be done in information form with null information prior. The detailed equations are given in the following. For the sake of readability, the forward and backward distributions are denoted by, for all k:

$$P(X_k|Z_0, \ldots, Z_k) \sim \mathcal{N}(x_k, P_k) \tag{8.3}$$

$$P(X_k|Z_{k+1}, \ldots, Z_N) \sim \mathcal{N}(J_k^{-1} y_k, J_k^{-1}), \tag{8.4}$$

where y_k and J_k are the information vector and matrix respectively. The forward recursion may be performed based on the standard Kalman filtering belief. The parameters of the Gaussian are returned by the standard linear Kalman filter by alternating between the propagation step:

$$\begin{aligned} x_{k+1|k} &= F_k x_k + u_k \\ P_{k+1|k} &= F_k P_k F_k^T + Q_k \end{aligned} \tag{8.5}$$

and letting $K_{k+1} := P_{k+1|k} H_k^T (H_k P_{k+1|k} H_k^T + R_k)^{-1}$, the update step

$$\begin{aligned} x_{k+1} &= x_{k|k+1} + K_{k+1}(Z_k - H_k x_{k|k+1}) \\ P_{k+1} &= P_{k+1|k} - K_{k+1} H_k P_{k+1|k}. \end{aligned} \tag{8.6}$$

We see that even if some or even all the eigenvalues of P_0 and Q are null, the computation may be performed as long as R is non-singular. As a consequence, Q can be as small as desired without leading

to numerical issues. The equations followed by J_k and y_k in the backward recursion are obtained by setting J_N to be null, and then considering (8.2) in reverse time, and combining it with (8.5), using the identities $x_k = J_k^{-1} y_k$, $P_k = J_k^{-1}$. They read:

$$\begin{aligned} J_k &= F_k^T (I + J_{k+1} Q_k)^{-1} J_{k+1} F_k, \\ y_k &= F_k^T (I + J_{k+1} Q_k)^{-1} (y_{k+1} - J_{k+1} u_k) \end{aligned} \tag{8.7}$$

And the final solution $X_k^* = \text{argmax}_{X_k} P(X_k|Z)$ is given by merging the estimates obtained at the forward and backward pass as follows for each k, see [79]:

$$X_k^* = x_k + P_k (P_k + J_k^{-1})^{-1} (J_k^{-1} y_k - x_k). \tag{8.8}$$

However, the given formula implies inverting J_k, which we want to avoid. Therefore, we propose to modify (8.8) as follows:

$$X_k^* = (I + P_k J_k)^{-1} (x_k + P_k y_k), \tag{8.9}$$

Moreover, the final covariance of X_k^* given Z, $P_{k|N}$ is given by

$$P_{k|N} = (I + P_k J_k)^{-1} P_k \tag{8.10}$$

We see the equations above allow performing optimal linear smoothing without involving at any time the matrix inverses Q^{-1} or P_k^{-1} where P_k denotes the forward covariance matrices, see (8.3). Indeed, in (8.7) and (8.8) each time those matrices are involved in an inversion operation there is a natural regularisation term $(I + \cdot)$ involved as well. These terms are much better behaved for two reasons. First, they have eigenvalues bigger than one, since the product of two symmetric positive semi-definite matrices is also positive semi-definite. Second, conditionning problems would arise if they had both eigenvalues close to one, which is likely, and eigenvalues much bigger. The latter should not happen. Indeed, considering Q to be small, the eigenvalues of P_k are driven by the initial covariance P_0 and the inverse of the measurement covariances R_k, while those of J_k come from the R_k's only. In the present context, the inversion of P_0 is avoided, meaning that the initial state is assumed to be known with at least reasonable precision. Similarly, R_k is supposed to be large enough to allow the computation of the Kalman gain, thus the eigenvalues of its inverse will not explode.

8.1.3 Summary of the approach

The duality between the probabilistic and the optimisation approaches, as well as the interplay between linear and non-linear systems may create confusion. We thus propose to summarise the approach and the results obtained so far.

- The solution to the smoothing problem corresponds to optimisation problem (4.9) with cost function (7.6). To attack this problem one usually linearises the cost at current estimate and solves a simplified optimisation problem (7.1) with quadratic cost function, i.e., a least squares problem. This provides a correction to the current estimate of χ, and then the cost (7.6) is relinearised at corrected estimate, yielding another least squares problem which in turn provides a novel correction. This is repeated until convergence to the optimum.

- At each optimisation step, the least squares solution (7.1) involves solving the normal equations (7.2), which may be ill-conditioned when process noise is too low. In the present chapter, we address the problem of smoothing in the presence of process noise covariance matrices that may be ill-conditioned or even singular, owing to the use of highly accurate motion sensors.

- In the case of *unary* observations (7.4) only, one may associate a linear dynamical system (8.2) with the linearised cost (7.1), (8.1) at each step. The Kalman smoother then provides a solution to the ensuing optimisation problem. By using a slightly modified implementation related to the BIFM of [79], we obtain a solution to (7.1) without inverting matrices that may be ill-conditioned or singular.

- In the sequel, we seek to adapt the proposed latter solution to the case where measurements of the form (7.5) are also involved, as typically arises when using vision sensors.

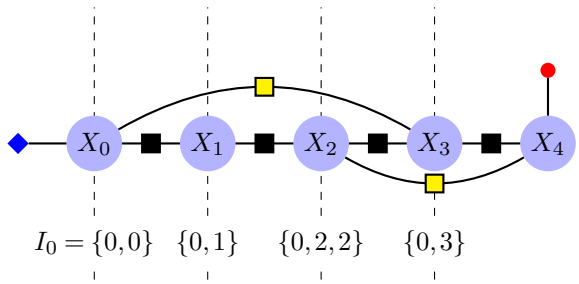


Figure 8.2: Stochastic cloning methodology for the forward pass in the factor graph framework. At each time k, state variable X_k is augmented to form the variable $\tilde{X}_k := X_{I_k}$ where I_k consists of current state index k and all the indices of the current and past variables that are to be used in a relative measurement. In a factor graph representation, variables involved in stochastic cloning are easily visualised: "clones" at time k are the variables related by an orange factor that spans from, to, or "over" the current state.

8.2 Proposed method: the BIFM Kalman smoother with stochastic cloning

Throughout this section, we still consider the linearised optimisation problem (7.1), but where observations may involve pairs of states, to account for measurements of the form (7.5). As previously done in (8.2), one may then associate a linear system

$$X_{k+1} = F_k X_k + u_k + w_k, \tag{8.11}$$

$$Z_k = H_k X_k + V_k \quad \text{(unary observations)} \tag{8.12}$$

$$Z_k = H_k X_k + H_l X_l + V_k,\ l < k \text{ (pair of states observations)} \tag{8.13}$$

to the optimisation problem, where (8.13) stems from the linearisation of observations of the form (7.5). Building upon the BIFM of the previous section, we will attempt to solve the corresponding optimisation problem (i.e., smoothing) without ever inverting matrices that may be ill-conditioned or singular in the presence of low or even null process noise w_k.

8.2.1 Stochastic cloning for filtering

Although we are concerned with smoothing, we have seen the Kalman filter is the main component of the forward pass of the BIFM smoother. Standard Kalman filtering for linear systems of the form (8.11)-(8.13) has been rendered possible through the stochastic cloning method, introduced in [85] and now one of the key components of state-of-the-art Multi-State Constrained Kalman Filter (MSCKF) for visual-inertial odometry [86]. The state is cloned at some point in time l and kept in memory to be able to compute an update that will involve it in the future at time k, see (8.13). The clones are discarded once they are not useful anymore. Note that the variables designated as clones are easily visualised in the factor graph framework, see Figure 8.2.

Remark 20. *Since the focus is put on the linearised problem, the factor graph is fixed, therefore it is known in advance which state is related to which one, and thus when cloning or discarding. Removing a clone does not delete it from the higher-level estimation process, as does marginalisation for instance.*

To incorporate a measurement (8.13) in the Kalman filter update, one needs the mean and covariance matrix of the vector (X_l, X_k). Thus, denoting by I_k the set containing k and all indices of past states involved in a future measurement, we may define the current state at time step k as $\tilde{X}_k = (X_i)_{i \in I_k}$, and we see its size changes over time, see Figure 8.2.

The stochastic cloning pipeline

The goal of Kalman filtering with SC is to compute at each time step k the parameters of the Gaussian density $P((X_i)_{i \in I_k} | Z_0, \dots Z_{k-1})$, and then to update them accounting for observations available at k, i.e., compute the parameters of distribution $P((X_i)_{i \in I_k} | Z_0, \dots Z_k)$. This can be done along the lines of the

standard Kalman filter, even though the dimension of the state keeps changing over time. The cloning pipeline may be intuitively described as follows:

$$\text{Cloning} \rightarrow \text{Propagation} \rightarrow \text{Update} \rightarrow \text{Clone discarding}$$

Let $\tilde{P}_k$ denote the covariance matrix of the augmented state $\tilde{X}_k = (X_i)_{i \in I_k}$. The SC steps are as follows.

Clone creation (duplication step) The cloning step corresponds to the fact that if the current state X_k is to be used in a future relative measurement, then it must be cloned. This simply consists in duplicating the X_k and concatenating it with the full state $\tilde{X}_k$, which thus now contains twice the state X_k. Indeed, the current state is of the form $\tilde{X}_k$ where I_k contains current index k and index of clones already created and not yet discarded. Duplication of current state then writes:

$$\tilde{X}_k \leftarrow (X_k, \tilde{X}_k) \tag{8.14}$$

This may be rewritten in terms of matrix computation as follows:

$$\tilde{X}_k \leftarrow F_k^c \tilde{X}_k, \qquad F_k^c = \begin{pmatrix} I & & \\ I & & \\ & \ddots & \\ & & I \end{pmatrix} \tag{8.15}$$

Since X_k and its copy are fully correlated, the covariance of the extended vector must be replaced with

$$\tilde{P}_k \leftarrow F_k^c \tilde{P}_k (F_k^c)^T = \begin{pmatrix} P_k & P_{I_k,k} \\ P_{k,I_k} & \tilde{P}_k \end{pmatrix}, \tag{8.16}$$

where P_k is the marginal covariance of the current state variable X_k, $\tilde{P}_k$ the marginal covariance of the augmented state $\tilde{X}_k$ before duplication, $P_{I_k,k}$ and P_{k,I_k} the cross-correlations. The computation of P_{k,I_k} is made by copying blocks from $\tilde{P}_k$ and the correlation between X_k and X_k is the identity matrix.

Propagation Clones remain static during propagation, since the rationale is merely to keep past states in the augmented state. Hence we use a propagation step analogous to (8.5) but with augmented dynamics, input, and process noise as follows:

$$F_k^p = \begin{pmatrix} F_k & 0 \\ 0 & I \end{pmatrix}, \quad \tilde{u}_k = \begin{pmatrix} u_k \\ 0 \end{pmatrix}, \quad \tilde{w}_k = \begin{pmatrix} G_k w_k \\ 0 \end{pmatrix}. \tag{8.17}$$

Update The main purpose of SC is to allow carrying out the updates as in the standard Kalman filter. Indeed, as each relative observation at time k involves a past state variable with index $l < k$ that has been cloned, and thus $l, k \in I_k$, observations (8.13) may be written using a novel observation matrix $\tilde{H}_k = (H_i)_{i \in I_k}$, defined such that

$$\tilde{H}_k X_{I_k} = \sum_{i \in I_k} H_i X_i. \tag{8.18}$$

Clone discarding Once a clone is no longer useful, i.e., it will be involved in no later measurement, it can be discarded. This is done by marginalizing out the considered clone, which reads in terms of matrices

$$\tilde{X}_k \leftarrow F_k^d \tilde{X}_k, \quad \tilde{P}_k \leftarrow F_k^d \tilde{P}_k (F_k^d)^T, \tag{8.19}$$

where F_k^d is the identity matrix from which we removed the rows corresponding to states we discard, i.e., those in $I_k \setminus I_{k+1}$.

SC implementation

Gathering all the steps above, we see the obtained Kalman filter maintains the augmented state $\tilde{X}_k = (X_i)_{i \in I_k}$ and its covariance matrix $\tilde{P}_k$. Formally, $\tilde{X}_k$ follows the state equations

$$\tilde{X}_{k+1} = \tilde{F}_k \tilde{X}_k + \tilde{u}_k + \tilde{w}_k, \tag{8.20}$$

$$Z_k = \tilde{H}_k \tilde{X}_k + V_k, \tag{8.21}$$

using the augmented quantities defined in (8.17), and where the matrix $\tilde{F}_k$ is defined with the help of matrices introduced in (8.15), (8.17), and (8.19), by

$$\tilde{F}_k = F_k^p F_k^c F_k^d, \tag{8.22}$$

(note it is suggested to perform the discarding step before duplication to spare the algorithm undesirable computations), and with observations defined by (8.18) corresponding to (8.12)-(8.13). Formally the obtained system (8.20)-(8.21) fits into the standard Kalman form, and the fact the dimension of the state changes does not result in any modification of Kalman's equations.

8.2.2 Proposed "SC-BIFM" (stochastic cloning for smoothing)

The proposed solver consists in applying the BIFM smoother of Section 8.1.2 dedicated to system (8.2) to the augmented system created by stochastic cloning to address systems of the form (8.11)-(8.12)-(8.13). This way, we will achieve robustness to singular prior and process noise covariance matrices.

Forward pass with stochastic cloning

The forward pass is akin to Kalman filtering, and the method of Kalman filtering with stochastic filtering was recalled in the previous paragraphs, and put in perspective in the proposed factor graph context. Forward pass consists of the BIFM forward recursion (8.5)-(8.6), applied to the augmented system (8.20)-(8.21) having augmented state variables $\tilde{X}_k$.

Backward stochastic cloning with an information filter

This step consists in applying the backward equations in information form (8.7) to the augmented system (8.20)-(8.21) instead of the standard filtering equations with unary observations (8.2). As a noticeable difference, we observe that contrary to F_k in (8.2), matrix $\tilde{F}_k$ is not square. However, only its transpose is involved in the backwards equations and this poses no problem. This is an advantage of having written the backwards recursion in information form: in standard form writing state at time k from state at time $k+1$ would require matrix inversion.

Let $\tilde{y}_k$ be the augmented backward information vector, and $\tilde{J}_k$ the associated information matrix. Let us comment on the steps involved in the backwards pass in the presence of clones.

Clone creation In the forward pass, the vector state is augmented each time a pose has to be kept for later use, while the covariance matrix is augmented with full correlation as in Eq. (8.16). In the backward case, the order of the steps is reversed, so the information vector and information matrix are augmented each time a clone was discarded in the forward pass. We see on Figure 8.2 that in our factor graph context, the process of clone creation and destruction in backwards time strictly mirrors the forward time. However when augmentation occurs in the backward sense new entries of both $\tilde{y}_k$ and $\tilde{J}_k$ are padded with zeros: we create a state with no correlation with the rest of the system. Note indeed that the clone refers to another state variable X_k appearing in the "past", i.e., X_l with $l < k$. Thus immediately after creating it the observation linking the current state k to the added state having index l is performed, which will create correlation. Note this interesting fact: in the backward sense the clone does not "know" it is a clone (i.e., is by no means *fully* correlated with any state but only loosely correlated to it via the noisy observation involving the pair of states). Full correlation only appears when it gets discarded, mirroring the forward case where the clone is "informed" it is a clone, through Eq. (8.16), at its creation.

Clone discarding In the backward propagation, the counterpart of clone creation is surprisingly simple, since it boils down to applying $(F_k^d)^T$, see (8.22) and recall from (8.19) that F_k^d is the identity matrix from which we removed the rows corresponding to states we can discard. Assuming the clone lies at position l in the information vector and we are dealing with state at position k, the corresponding operation reads:

$$\begin{aligned} \tilde{y}_k &\leftarrow \tilde{y}_k + \tilde{y}_l \\ \tilde{J}_{k,k} &\leftarrow \tilde{J}_{k,k} + \tilde{J}_{k,l} + \tilde{J}_{l,k} + \tilde{J}_{l,l} \end{aligned} \tag{8.23}$$

These formulas are directly derived from (8.7) and (8.15) reading what happens to the entries of the matrix J and the vector $\tilde{y}$. Indeed, the transpose of F_k^c is applied, which sums the last two blocks of a vector. Combined with the transpose of the clone discarding matrix F_k^d, we get the above formulas.

Interpretation

Although the backward pass was obtained through "blind" matrix manipulations of the system in reverse time, the following interesting interpretation of what cloning means in reverse time provides insight. Indeed, discarding the clone in backward time is actually a sequence of two actions: informing the clone it is a clone, then killing it (mirroring clone creation and full correlation information (8.16) in the forward pass). The first action is equivalent to a noise-free observation $Z_k = X_k - X_l$, taking value $Z_k = 0$. To study this operation mathematically, let us associate a non-zero covariance matrix R to observation Z_k. We will then study what happens when $R \to 0$. Since we are in information form, only the components $\tilde{y}_k, \tilde{y}_l$ and the blocks $\tilde{J}_{k,k}, \tilde{J}_{k,l}, \tilde{J}_{l,k}, \tilde{J}_{l,l}$ are affected by this operation. The Kalman update equations then yield:

$$\begin{aligned} \begin{pmatrix} \tilde{J}^+_{k,k} & \tilde{J}^+_{k,l} \\ \tilde{J}^+_{l,k} & \tilde{J}^+_{l,l} \end{pmatrix} &= \begin{pmatrix} \tilde{J}_{k,k} & \tilde{J}_{k,l} \\ \tilde{J}_{l,k} & \tilde{J}_{l,l} \end{pmatrix} + \begin{pmatrix} I & -I \end{pmatrix}^T R^{-1} \begin{pmatrix} I & -I \end{pmatrix} \\ &= \begin{pmatrix} \tilde{J}_{k,k} + R^{-1} & \tilde{J}_{k,l} - R^{-1} \\ \tilde{J}_{l,k} - R^{-1} & \tilde{J}_{l,l} + R^{-1} \end{pmatrix} \end{aligned} \tag{8.24}$$

After this observation, we marginalize out the clone. In information form, the remaining block $\tilde{J}^{++}_{k,k}$ is given by the classical Schur complement formula:

$$\tilde{J}^{++}_{k,k} \leftarrow \tilde{J}^+_{k,k} - \tilde{J}^+_{k,l} \left[\tilde{J}^+_{l,l}\right]^{-1} \tilde{J}^+_{l,k}$$

Picking in Eq. (8.24) the values of the starred blocks, the new block $\tilde{J}^+_{k,k}$ after the sequence " observation + marginalisation" reads:

$$\tilde{J}^{++}_{k,k} = \tilde{J}_{k,k} + R^{-1} - \left(\tilde{J}_{k,l} - R^{-1}\right)\left(\tilde{J}_{l,l} + R^{-1}\right)^{-1}\left(\tilde{J}_{l,k} - R^{-1}\right) \tag{8.25}$$

$$= \tilde{J}_{k,k} + R^{-1} - \left(\tilde{J}_{k,l}R - I\right)\left(\tilde{J}_{l,l}R + I\right)^{-1}\left(\tilde{J}_{l,k} - R^{-1}\right) \tag{8.26}$$

$$= \tilde{J}_{k,k} + R^{-1} - \left(\tilde{J}_{k,l}R - I\right)\left(I - \tilde{J}_{l,l}R + \circ(R)\right)\left(\tilde{J}_{l,k} - R^{-1}\right), \tag{8.27}$$

where, after factorising R^{-1} in $\left(\tilde{J}_{ll} + R^{-1}\right)^{-1}$, the first-order expansion $(I+\epsilon)^{-1} = I - \epsilon + \circ(\epsilon)$ was carried out. Developing the parentheses, we see a $-R^{-1}$ term appears, canceling the R^{-1} term (second term in the right hand side), and we end up with:

$$\tilde{J}^{++}_{k,k} = \tilde{J}_{k,k} + \tilde{J}_{k,l} + \tilde{J}_{l,k} + \tilde{J}_{l,l} + \circ(R)$$

And finally we can make R tend to zero (i.e., precision of the virtual measurement to infinity) to recover indeed (8.23). Interestingly, the action "informing two states they are clones", easily encoded in the covariance form, cannot be made in the information form, due to the infinite R^{-1} terms. On the other hand the sequence ""informing two states they are clones, and then keeping only one" works out beautifully in the information matrix form.

Algorithm 3: SC-BIFM

Input: $a_0, P_0, (f_i)_i, (Q_i)_i, (Z_k)_k, (R_k)_k$;

Forward pass

1 Set $\tilde{X}_0 = a_0$, $\tilde{P}_0 = P_0$;

For $k < N$ **do**

2 Compute $\tilde{X}_{k+1}$ and $\tilde{P}_{k+1}$, based on (8.20), (8.21), using the Kalman equations (8.5), (8.6);

Backward pass

3 Set $\tilde{y}_N = 0$, $\tilde{J}_N = 0$, $k = N$;

For $k > 0$ **do**

4 Compute $\tilde{y}_{k-1}$ and $\tilde{J}_{k-1}$, based on (8.20), (8.21) using the information form equations (8.7) ;

For $k \leq N$**, do**

5 Compute the augmented solution $\tilde{X}^*_{I_k}$ based on (8.9) and extract only the state variables X^*_k ;

Output: $\chi = (\tilde{X}^*_k)_{k \leq N}$ which is the exact solution to the linearised optimisation problem (7.1);

Final fusion

The final fusion is carried out for each step as in standard BIFM, according to (8.9). The full pipeline is summarised in Algorithm 3.

Remark 21. *Alternatively, at step 5 of Algorithm 3 one could first extract simple state variables X_k from augmented ones $\tilde{X}_k = X_{I_k}$, and then perform fusion* (8.9) *with smaller matrices to save computation time. However the extraction is a marginalisation, which has to be performed on the information variables $\tilde{y}, \tilde{J}$, which is computationally costly.*

Note that from the final update we may also get the posterior covariances of the related variables. However, this only yields covariances of variables which are neighbors in the measurement graph, contrary to more general methods [72]. To this respect, note that an alternative to SC-BIFM, which does not try to maintain sparsity, is to simply not discard the created clones. Then, at the end of the forward pass, the filter outputs $P(X_0, \cdots, X_n | Z_0, \cdots, Z_n)$, i.e., the solution we pursue.

8.2.3 Current limits of the proposed SC-BIFM

Although SC-BIFM takes advantage of the sparsity of the smoothing problem, there are still limitations in its application. We state two out of the major ones. The first one is the fact that the forward sweep needs to start from a state on which a prior is available, and to follow a propagation route. This highly limits the orders in which the variables are visited, which plays a crucial role in the filling of the Cholesky factor, and thus on the number of clones we have to consider [45]. The second one is the absence of an incremental formulation. Indeed, the forward part is easily continued when a new measurement is available, but the backward information filter seems to have to be recomputed from scratch each time. We believe that there might be links with the Bayes tree used in iSAM2 [73], for instance, which could help resolving this issue.

8.3 Generalisation to all linear least-squares

The solver proposed in the previous section only applies to specific linear least-squares problems, which are based on factor graphs having a particular structure, such as that in Figure 7.1. Indeed, it relies on a subgraph encoding a smaller but full-rank least-squares, which has to be a simple path. Although it covers most of pose-graph problems encountered in practice, it can not be directly applied to feature-based SLAM for instance, where such a subgraph would be at best a spanning-tree. However, it can be easily generalised, based on one of the key considerations used: the Jacobian F used for propagation does not need to be square. In the following, this generalisation is detailed in two steps. First, how to take into account a factor graph built around a spanning tree of full-rank factors, as can appear in

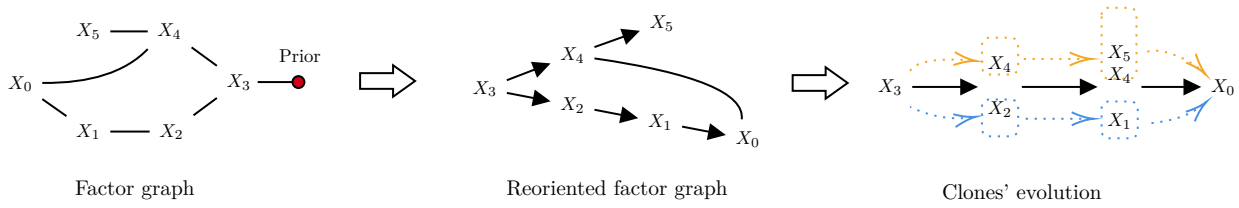


Figure 8.3: Example of SC-BIFM applied to a problem where the prior is not on the "first" state. Assuming all the propagation factors are invertible, the factor graph can be ordered following a spanning tree rooted at the prior. Then cloning is used to treat both branches in parallel, as shown by the colored boxes and arrows, allowing to treat the factors between each branch seamlessly.

feature-based SLAM, and not a simple path as in Figure 7.1. This also allows considering a system with a prior on a state which would not be X_0. Second, how to handle a variable which is only connected by lower-dimensional factors, for instance bearing-only measurements of a landmark.

8.3.1 Dealing with spanning-trees of invertible factors

The factor graphs and least-squares problems considered in this work are connected and full-rank respectively. Therefore, there is a minimal subgraph encoding a full-rank least-squares. Suppose that this subgraph is a tree (it will necessarily be a spanning-tree), composed of full-rank Jacobians with respect to the variable not yet seen. This is for instance the case with inertial propagation factors, or bearing-distance observations for feature-based SLAM. It will serve as a propagation route for SC-BIFM. The assumptions made mean that the factors can be inverted to give a value for the newly seen variable. Then, SC is simply applied to keep trace of the states of both (or more) branches. In feature-based SLAM, for instance, there are factors of the form

$$\|y - H_X X - H_p p\|_R^2, \tag{8.28}$$

where X represents the vehicle state, p a feature, y the observation, H_X and H_p the related Jacobians and R a covariance. If the feature had not been seen before, the procedure is the same as for the EKF-SLAM (and any Kalman-based SLAM):

$$X \leftarrow \begin{pmatrix} X \\ p \end{pmatrix} = \begin{pmatrix} Id \\ -H_p^{-1} H_X \end{pmatrix} X + \begin{pmatrix} 0 \\ H_p^{-1} y \end{pmatrix}. \tag{8.29}$$

The same applies for instance to pose-graph instances with branching, as illustrated in Figure 8.3. In this case, the branches are simply treated in parallel through the clones. In the pictured example, a prior on X_3 can be treated.

However, note that this generalised SC-BIFM *does not get the complexity of EKF-SLAM*. Indeed, the features are dropped from the states once they no longer take part in factors. Given an order in which the spanning-tree is visited, the size of the state varies as would the filling of the Cholesky factor or the size of marginals in a Bayes Tree [45,73]. The choice of that order is more restricted in SC-BIFM though, as explained in Section 8.2.3

8.3.2 Regrouping factors and variables to propagate

It can happen that the minimal full-rank subgraph is not a spanning-tree. This is for instance the case in feature-based SLAM, where several observations of a single feature are needed to initialise it, or in 2D localisation where two position measurements between two different locations are needed to initialise the orientation. Both cases and examples of strategies to be used are shown in Figure 8.4. In both cases, the idea is to regroup the states which are needed to initialise the new variable in a single extended state, just as SC does. Once this is done, the parts that are not needed anymore can be discarded, like clones which would not be involved in further measurements. In a way, this is similar to the treatments used in MSCKF [85]. The generalised version of SC-BIFM is summarised in Algorithm 4.

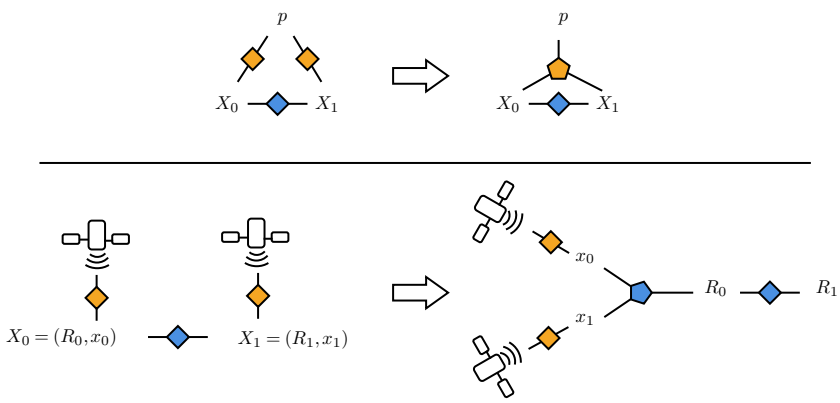


Figure 8.4: Two examples of SC-BIFM applied to systems where propagation is not direct. Top: a feature needs to be seen twice before being initialised. Therefore, the two binary factors allowing its initialisation are turned into a ternary one. In turn X_0 is cloned, and the extended state allows the initialisation. Bottom: two consecutive 2D poses are linked to position measurements, but no prior exists to initialise the heading of X_0. Both positions x_0, x_1 are taken first. The odometry allows initialising R_0, which is then propagated to R_1.

Algorithm 4: Generalised SC-BIFM

Input: Full-rank linear least-squares;

1 Find a minimal subgraph encoding a full-rank least-squares ;

2 Turn it into a spanning-tree as in Section 8.3.2 ;

3 Apply SC-BIFM using the spanning-tree as propagation as in Section 8.3.1 ;

Output: $(\hat{X}_k)_{k \leq N}$;

Remark 22. *The regrouping of factors actually links SC-BIFM and the Batch method presented in Section 7.3. Indeed, splitting the factors in two groups with one having a square Jacobian, A_1 is equivalent to regrouping all these factors to turn them into a prior on the whole variables.*

8.4 Numerical illustration of the methodology

We illustrate numerically the behavior of each method using a linear simplified navigation example. Consider a body equipped with a biased accelerometer moving along a horizontal line. By letting $p(t)$ denote its position, $v(t)$ its velocity, and b a static accelerometer bias, the noiseless dynamical motion equations read

$$\dot{p} = v, \ \dot{v} = (u + b), \dot{b} = 0$$

with u a known input delivered by the acceledometer. We suppose moreover that at discrete time instants, using e.g., vision, the system is able to measure relative displacement with respect to a past position. In discrete time, and adding some process noise, the system $X = (b, v, p)$ may be approximated by

$$X_{k+1} = \begin{pmatrix} 1 & & \\ -dt & 1 & \\ & dt & 1 \end{pmatrix} X_k + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} u_k + w_k \tag{8.30}$$

$$Z_k = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} (X_{i_k} - X_{j_k}) + V_k \tag{8.31}$$

where we assume a relative displacement is measured between X_3 and X_0 and X_4 and X_2, that is, same relative measurements as displayed in the graph of Figure 7.1. This model is a simplified version of the standard mechanisation used in inertial navigation. Usually, position integration with respect to the speed is considered exact on a single time step, however IMU outputs are given at a much higher rate than the observations, and are therefore preintegrated between two states [53]. This leads to a propagation factor with full-rank (although low eigenvalues) covariance. We take the length of the

trajectory to be $N = 4$. Observation noise V_k has variance σ_z. The process noise w_k has diagonal covariance $\text{diag}(\sqrt{dt}\sigma_b, \sqrt{dt}\sigma_{acc}, \sqrt{dt}\sigma_{int})$, which represent the bias random walk, the accelerometer and the integration uncertainty respectively. Their magnitude reflects typical characteristics of high-grade inertial sensors:

$$\sigma_b = 1e-3,\ \sigma_{acc} = 1e-2,\ \sigma_{int} = 1e-3, \tag{8.32}$$

with initial prior parameters and observation noise

$$\sigma_b^0 = 1e-2,\ \sigma_v^0 = 1,\ \sigma_x^0 = 1, \sigma_z = 0.1. \tag{8.33}$$

We compare the following solvers:

- The standard square-root information solvers based on Cholesky (or QR) factorisation of Section 7.1.2, with precision of 64-(Sq-Root-64) and 32-bits (Sq-Root);
- The robust batch solver ("Batch solver") of Section 7.3, with precision of 64-(Batch-64) and 32-bits (Batch 32);
- The proposed SC-BIFM solver in 32-bits (only).

8.4.1 Ideal case numerical experiment

First, we study the impact of conditioning on the solvers in the noise free case, that is where the ground truth actually follows (8.30), (8.31) with $w_k = 0$ and $V_k = 0$. The state is initialised at the ground-truth, so that we can provide the true minimum as ground truth, by simply integrating (8.30). The estimation here is made relative to the initial position x_0, which is naturally considered fixed and is removed from the state. The effect of the time step dt on the numerical behavior of the various solvers is displayed on Figure 8.5, top plot. Indeed, dt is tightly linked with the system's conditioning, since the propagation covariance varies with its square-root.

We can see that, as expected, the solvers based on information matrix suffer from degeneracy when the time step becomes too small, while the others maintain a stable precision, in both 32-and 64-bits. SC-BIFM is not displayed here, as it sticks to the true state owing to perfect initialisation in the forward pass, and thus returns the true solution.

8.4.2 Numerical experiment with low noise

Monte-Carlo simulations were run with noise turned on. Since the 64-bits batch robust solver actually managed to find the true solution of the linear least-squares problem in all cases, it was taken as the reference for the comparison of the remaning solvers. 200 sets of measurements were randomly generated with noises (8.32)-(8.33). The distance between the solution of each solver and the ground-truth are computed at each run, and the average distances are shown on Figure 8.5, bottom plot. In the noisy case, we also see that the standard solvers perform equivalently or slightly better than the robust ones at large time steps, but degrade rapidly. These results clearly indicate that one should be careful when designing least-squares-based estimators for accurate sensors.

Remark 23. *The degeneracy of the square-root filters comes from the fact that they are based on QR factorisation. Using an SVD to invert the system proves to be much more robust. However, it does not maintain sparsity and is therefore not desirable in the context of factor graph navigation and SLAM.*

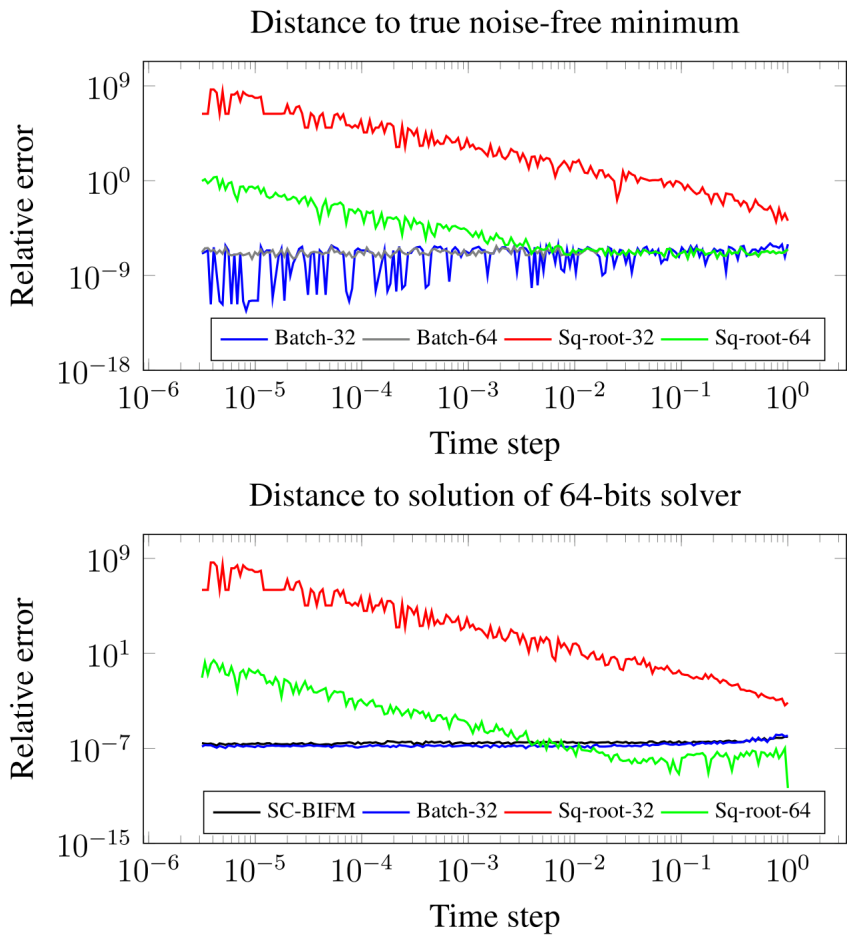


Figure 8.5: Top plot: distance between the solutions of various solvers and the true minimum of the simple numerical example of Section 8.4 in the process noise free case. Bottom plot: average Monte-Carlo distance to the solution of the 64-bits robust batch solver (which systematically finds the optimum) for various solvers in the presence of process noise. In both cases performances are plotted against the discretisation time step dt in (8.31). Standard solvers based on the information form (Sq-root-32 and Sq-root-64) degrade rapidly when dt is taken small, whereas the ones we propose (Batch and SC-BIFM) do not degrade, even with single precision (32-bits).

Chapter 9

Real-world experiments

Highlights

- The standard linear solvers used in smoothing fail when used in single precision (32-bits) on experimental data with a high-grade IMU.
- Smoothing relying on SC-BIFM proves to be robust and carries out the estimation seamlessly.

Points marquants

- Les solveurs linéaires utilisés classiquement pour le lissage défaillent lorsqu'ils sont appliqués sur des données expérimentales comprenant une centrale inertielle de haute qualité, avec une précision simple (32-bits).
- Le lissage reposant sur le SC-BIFM s'avère robuste et réalise l'estimation sans problèmes.

Introduction (En/Fr)

The consistency and robustness of SC-BIFM have been assessed on a simulated toy example. However, its usefulness must still be evaluated in practice. To this end, experiments using the same setting as in Section 6.3 were conducted to compare different smoothing algorithms sharing the same parametrisation, R-IS, but relying on different linear solvers. In particular, all solvers were tested with both single and double precision settings. While in the latter case, the estimation is carried out without trouble every time, the standard solvers turn out to diverge in the middle of the run in single precision. In the meantime, methods relying on both the robust batch and SC-BIFM solvers seamlessly completed the estimation.

La cohérence et la robustesse du SC-BIFM ont été établies sur un exemple jouet simulé. Cependant, son utilité doit encore être évaluée en pratique. Pour ce faire, des expériences se basant sur le même dispositif que pour la Section 6.3 ont été conduites pour comparer les différents algorithmes de lissage partageant la même paramétrisation, R-IS, mais s'appuyant sur différents solveurs linéaires. En particulier, tous les solveurs ont été testés en précisions simple et double. Tandis que dans ce dernier cas, l'estimation est toujours menée sans problème, les solveurs standards divergent au milieu de l'estimation en précision simple. Dans le même temps, les méthodes basées sur les méthodes groupée robuste et le SC-BIFM mènent l'estimation à son terme sans encombre.

9.1 Experimental setup

9.1.1 Considered models and parameters

The experimental setup is close to that of Section 6.3, the only considered sensors are the IMU and relative translations obtained from laser scans. The underlying smoothing method is R-IS as detailed in Section 4.5.1. The relative translations between states i and j follow the standard model

$$t_{ij} = R_i^T (t_j - t_i) + v_{ij}, \tag{9.1}$$

with v_{ij} a white noise.

The laser scans were transformed into relative translations obtained at 4 Hz.

The state's position and heading is initialised thanks to a global reference, the biases and initial velocity are set to zero, and the uncertainty is modeled the following way,

$$
\begin{aligned}
&P_0 = \mathrm{diag}(\sigma_R^0 I_3, \sigma_v^0\ I_3, 0, \sigma_{b^\omega}^0\ I_3, \sigma_{b^a}^0\ I_3),\\
&\sigma_R^0 = 1\ ^\circ,\ \sigma_v^0 = 100\ m/s,\ \sigma_x^0 = 0.1\ m,\ \sigma_{b^\omega}^0 = 7e-4\ ^\circ/s, \sigma_{b^a}^0 = 7e-2\ m/s^2 \sigma_\omega = 0.1\ ^\circ/h\\
&\sigma_a = 0.0015\ m/s^2\ (0.15mg),\ \sigma_{b^\omega} = 2.5e-5\ ^\circ/s/\sqrt{s},\\
&\sigma_{b^a} = 8e-4\ m/s^{-2}/\sqrt{s}.
\end{aligned}
$$

The initial heading uncertainty was not set exactly to 0, as the quality of the gyroscope makes it potentially sensible to the Earth's rotation, which gives hints about the heading.

9.1.2 Implementation

A sliding window approach spanning 5 seconds was adopted, in which states are added each time a relative translation measurement is received. Different solvers were implemented for comparison purposes. To focus on them, the rest of the pipeline was fully treated in double precision. All algorithms were implemented in Python, based on "Numpy" and "Scipy" built-in functions. In the following we detail the specifics of the implementation of the solvers.

Computing the uncertainties of the prior and the IMU preintegration

The uncertainty of the preintegrated IMU factors, was computed by propagating the square-root of the associated covariance. Likewise, the uncertainty of the prior was computed in square-root form, and kept in this form through the marginalisation process. This was needed to avoid some numerical issues, and was useful for feeding the Square-Root solvers with the most accurate uncertainty estimation.

SC-BIFM

It was implemented according to Algorithm 3. The covariance matrices P_0 and Q were retrieved by simply squaring the triangular matrices stored for the Prior and the IMU.

Square-Root Smoothing

Square-Root smoothing was achieved using the QR decomposition of $\tilde{A}$, based on its "Numpy" implementation. Then the resulting linear systems were solved thanks to the "Scipy" package.

Robust Batch Approach

As for SC-BIFM, the Batch solver was fed with full covariance matrices (i.e., not square roots). The solution was computed using the "Scipy.sparse" package, and especially its "spsolve" method to invert the ensuing square linear systems.

9.2 Experimental comparison of linear solvers illustrating the decay of standard methods

Three linear least squares solvers are compared, in their 32- and 64-bits floating point formats:

- The standard square-root information solvers based on Cholesky (or QR) factorisation of Section 7.1.2, with precision of 64 (Sq-Root-64) and 32 bits (Sq-Root-32);
- The robust batch solver (Batch solver) based on reformulation of Section 7.3, with precision of 64 (Batch-64) and 32 bits (Batch 32);
- The proposed SC-BIFM solver in 64 and 32 bits.

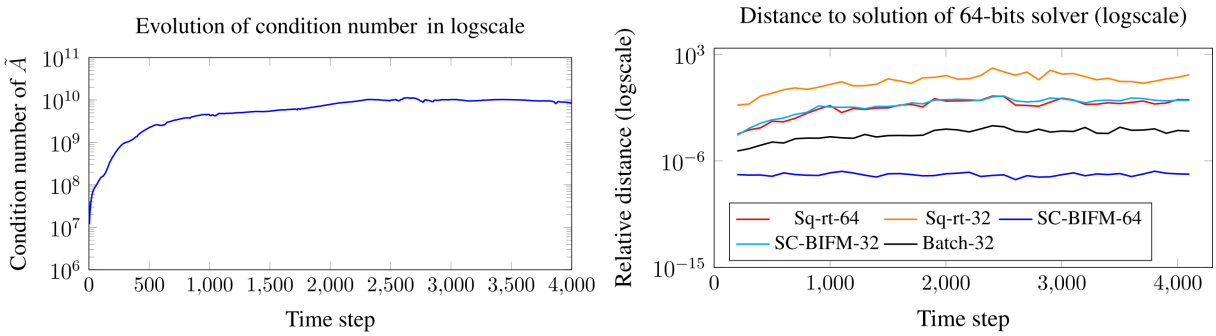


Figure 9.1: Left: Evolution of the condition number of $\tilde{A}$ during the trajectory, in log scale; Right: Distance of the solvers' solution of the linear least-squares $\text{argmin}_{\delta\chi} \|A\delta\chi - b\|_P^2$ to that given by Batch 64 considered herein as ground truth, in log scale. A moving average on 100 steps was performed to improve readability. There is a clear link between the increase of ill-conditioning and the accuracy degradation of the single precision solvers. However, the proposed SC-BIFM systematically beats its square-root information counterpart described in Section 7.1.2. Notably it is remarkable SC-BIFM achieves comparable results with 32 bits single-precision as square root with 64 bits double-precision implementation.

Here, the performance is not measured in terms of quality of the navigation estimate, but in terms of *distance to the solution* of the involved linear least-squares. That is, when numerically solving $\text{argmin}_{\delta\chi} \|A\delta\chi - b\|_P^2$, how far from $\delta\chi^*$ is the result ? As noticed in the simple numerical example of Section 8.4, the best method in terms of accuracy (but not in terms of complexity) is the Batch formulation of Section 7.3 which solves the linear least squares while avoiding ill-conditioned related issues. As a result, it was chosen as the reference to be compared to. Moreover, to provide all solvers meaningful linearisation points, we performed fusion betwen IMU and relative poses observations based on scan matching using the Batch solver with 64 bits based double precision.

Each time step where a GN iteration was carried out, the condition number of the weighted Jacobian $\tilde{A} = \Sigma^{-1/2} A$ associated to the full normal equations (7.2), all the solvers were applied to the corresponding linear least squares system, and compared to the solution of the 64-bits robust batch solver Batch-64. Figure 9.1, shows the evolution of the condition number in log-scale, and the distance between the solution of each of the single precision solvers with respect to that of the Batch-64, which is considered herein as ground truth.

Accuracy

The methods rank as follows. The Batch approach we proposed in Section 7.3 runs first, then the proposed SC-BIFM, and finally the square root resolution of full normal equations. The fact that Batch beats SC-BIFM may be due to our taking advantage of the existing libraries to solve the linear systems that involve A_1^{-1}, whereas SC-BIFM uses no such library owing to its different nature and implementation based on Kalman smoothing, and thus had to be coded entirely from scratch.

Execution time

Average computation times of the various solvers are displayed in Table 9.1. compete with the Sq-Root, in both precision formats. Batch solvers are slightly slower, which was expected because of the A_1^{-1} term to be computed. This is on the other hand encouraging for SC-BIFM, which is based on an academic code developed from scratch and thus does not benefit yet from the same level of code optimisation as the other methods that use bricks such as Numpy built-in QR decomposition. Theoretically, our code runs in $O(m^3 n)$, where n is the number of variables and m the dimension of the largest X_{I_k} related to the number of clones. It is difficult to go further into complexity analysis owing to the clones, but we believe its complexity is closely related to the complexity of sparse Cholesky or QR decompositions, which varies with the filling of the R factor.

Solver	Average computation time (s)
Sq-Root 64	0.021
Sq-Root 32	0.010
Batch 64	0.029
Batch 32	0.024
SC-BIFM 64	0.020
SC-BIFM 32	0.016

Table 9.1: Average computation times for the inversion of the linear systems over trajectory. Computations were made on a standard laptop with Intel i5-5300 2.3 GHz CPU

Solver	Maximum distance to Batch-64 (m)
Sq-Root 64	0.09
Sq-Root 32	∞
Batch 64	0
Batch 32	0.01
SC-BIFM 64	$7e{-}4$
SC-BIFM 32	0.1

Table 9.2: Maximum distance to the Batch-64 trajectory for the various solvers. Note that this gives no information on the actual error of the navigation estimate.

9.3 Experimental comparison of corresponding localisation algorithms

The full pipeline of iterative linearisation procedure of pose-graph based SLAM was implemented using the various linear solvers, and their results have been compared. Table 9.2 displays the maximum distance to Batch 64 estimate (considered as optimal) in terms of position discrepancy of the car. The results confirm what could be anticipated from the linear solvers comparison above. Moreover, using single-precision Sq-root-32, the estimation could not even be carried out until the end, as it diverged after about an eighth of the trajectory, see Figure 9.2. SC-BIFM-64 is as expected the closest to Batch-64, followed by Batch-32, while Sq-root-64 and SC-BIFM-32 show similar behaviors. The latter feature is remarkable, though, as it shows he proposed algorithm SC-BIFM may achieve good results in single precision on a real application.

9.4 Conclusions regarding SC-BIFM

The first merit of the conducted experiments is to prove that the problem of ill conditioned information matrix for pose graph SLAM or Kalman smoothing for navigation may arise when using highly precise inertial sensors (namely the Epsilon 10, which is a cost effective high precision IMU). Given the progresses made in the field of inertial sensors of the past decades, one can anticipate performances will keep increasing, and costs decreasing. As evidenced, the resulting problems even lead to divergence of the localization estimate based on the standard Cholesky resolution of linear least squares (Sq-Root 32) when implemented with 32 bits single-precision. This is an important point, as most industrial grade inertial navigation embarked systems use programming languages based on 32 bits. To this respect, SC-BIFM and the robust batch solver are satisfactory solutions and achieve good performance in single precision. From a computational point of view, SC-BIFM is promising, as its complexity is reasonably low since only matrices of limited size need to be inverted, as opposed to the robust batch solver.

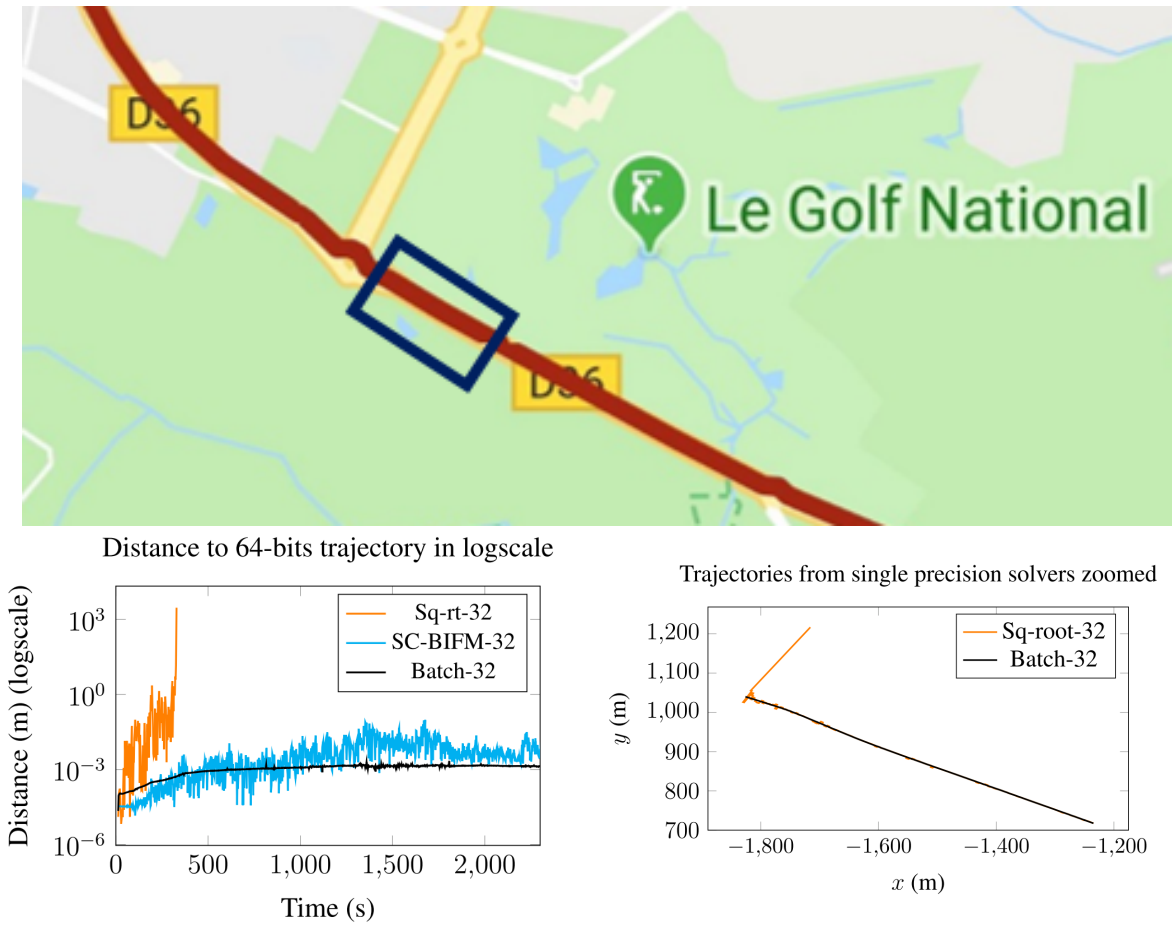


Figure 9.2: Top: Zoom on the 11 km long trajectory followed by the car (see Figure 6.1), where the standard methods fail. The bottom right image displays a zoom on the zone inside the black box, and shows the trajectories estimated by Sq-root-32 and Batch-32. We see Sq-root-32 diverges whereas Batch-32 properly follows the true trajectory. Both proposed single precision solvers, SC-BIFM-32 and Batch-32, managed to stay stable during the whole experiment, as shown in the bottom left picture. More generally, the deviations observed are consistent with the results reported in Table 9.2.

Chapter 10

Invariant SC-BIFM : a non-linear Kalman smoother for precise navigation

Highlights

- A non-linear counterpart of SC-BIFM is derived, creating a novel estimation algorithm. In particular, it appears that the estimation must focus on updates of the states, not directly the state itself.
- As L-IS, Invariant SC-BIFM solves the issues that the IEKF may encounter when biases are large.

Points marquants

- Un équivalent non-linéaire du SC-BIFM est dérivé, donnant un nouvel algorithme d'estimation. En particulier, il apparaît que l'estimation doit se concentrer sur les mises à jour de l'état, et non directement sur l'état lui-même.
- Comme le L-IS, le SC-BIFM invariant résout les problèmes que peut rencontrer l'IEKF lorsque les biais sont importants.

Introduction (En/Fr)

SC-BIFM was developed to solve linear systems stemming from non-linear estimation. However, just as the Invariant EKF is a non-linear counterpart of the linear KF, a non-linear version of SC-BIFM can be developed as an alternative smoother. Cloning is a linear action, and is as such straightforward to extend. Therefore, the focus here is on the translation of BIFM into the invariant framework.

In particular, it appears a counterpart of the BIFM can only be defined when working on *updates* of the current guess. Indeed, only through this parametrisation can one apply the log-linearity of group-affine dynamics for instance. Reparametrisation, for instance used in the Information filter on Lie groups of [50], must also be avoided in the backward sweep. Using first-order approximations for the observations and cases such as biased navigation, three possible alternatives of the invariant BIFM are proposed. Its interest is illustrated on the same simulated alignment experiment as in Section 4.4.6, where it also corrects the IEKF shortcomings.

Le SC-BIFM a été développé afin de résoudre les systèmes linéaires produits par l'estimation non-linéaire. Cependant, tout comme l'IEKF est un équivalent non-linéaire du KF linéaire, une version non-linéaire du SC-BIFM peut être développée comme lisseur alternatif. Le clonage étant une action linéaire, il peut être étendu facilement. L'accent est donc mis sur la transposition du BIFM dans le cadre invariant.

En particulier, il apparaît qu'un équivalent du BIFM ne peut être défini qu'en travaillant sur les *mises à jour* de l'estimation actuelle. En effet, ce n'est qu'à travers ce prisme que la log-linéarité des dynamiques groupe-affines peuvent être appliquées, par exemple. La reparamétrisation, utilisée notamment dans le filtre d'information sur groupe de Lie de [50], doit aussi être évitée dans la partie à rebours. En utilisant des approximations du premier ordre pour les observations et dans les cas tels que la navigation inertielle biaisée, trois variantes possibles du BIFM invariant sont proposées. Son intérêt est illustré sur le même problème dd'alignement simulé que dans la Section 4.4.6, pour lequel lui aussi corrige les limitations de l'IEKF.

10.1 Information filter on Lie groups: exactly transposing the IEKF

The extension of the information form of the Kalman filter to Lie groups is not straightforward, and was only recently introduced in [50]. It was then successfully applied to SLAM [77]. In this work, they transpose the LG-EKF of [26] into an information version, i.e., on the same data, up to numerical rounding, both filters should output the same theoretical result, but with different numerical load. Note that starting from an IEKF would give the same algorithm. In this section we recall their method.

Concentrated gaussian distribution

The developments of [50] are based on the same distribution model as the one the invariant framework is based on. Since the choice between left- and right-invariant error does not impact the following reasoning, we consider a left-concentrated gaussian distribution without loss of generality. The information form of a standard Gaussian of mean m and variance Σ is a simple reparametrisation by the information matrix $J = \Sigma^{-1}$ and the information vector $y = Jm$. It can be extended to the concentrated Gaussian. Let $\chi \sim \mathcal{N}_L(\hat{\chi}, \Sigma)$, ie $\chi = \hat{\chi}\exp(\xi)$, where $\xi \sim \mathcal{N}(0, \Sigma)$. Then its information form is obtained by defining

$$J = \Sigma^{-1}, \qquad y = J\log(\hat{\chi}). \tag{10.1}$$

That is, the true state χ rewrites $\chi = \exp(J^{-1}y)\exp(\xi)$. Therefore, it appears that there is a fundamental difference with the linear case, due to the fact that ξ is forced to have zero mean. J^{-1} denotes a covariance in the tangent space at $\exp(J^{-1}y)$, and in particular, y has no reason to be considered small enough to carry out fist-order linearisations. This is why, as for the IEKF and the LG-EKF, a reparametrisation step is needed at the update. This very step will appear to be the troublesome part in order to apply a non-linear version of BIFM.

In the following, Kalman-like notations are employed to represent the propagated and updated information vector and matrix, e.g., $y_{k+1|k}, J_{k|k}$.

Propagation

In general, since propagation directly impacts the mean $\hat{\chi}$, the new information vector is obtained by going back to the Lie group, while the information matrix is propagated with the system's Jacobians as usual:

$$J_{k+1|k} = \tilde{Q}^{-1} - \tilde{Q}^{-1}\tilde{F}(J_{k|k} + \tilde{F}^T\tilde{Q}^{-1}\tilde{F})^{-1}\tilde{F}^T\tilde{Q}^{-1} \tag{10.2}$$

$$y_{k+1|k} = J_{k+1|k}\log(f(\exp(J_{k|k}^{-1}y_{k|k}))), \tag{10.3}$$

where $\tilde{F}$ and $\tilde{Q}$ are the propagation Jacobian and noise covariance respectively, whose definitions depend on the chosen distribution (left or right).

Update

As the arrival of a new observation changes the mean of the posterior distribution, the update step of the information filter on Lie groups is not defined as simply as in the linear case. Indeed, a reparametrisation of the state is needed to keep the random variable $\xi_{k+1|k+1}$ with zero-mean.

Consider the supplementary information term

$$i_{k+1} = H^T R^{-1} b_{k+1}, \tag{10.4}$$

where b_{k+1} is the innovation term. In the linear case, the information matrix is updated in the standard way, i.e., $J_{k+1|k+1} = J_{k+1|k} + H^T R^{-1} H$, and the impact on the mean is given by

$$m_{k+1|k+1} = J_{k+1|k+1}^{-1} y_{k+1|k+1} \tag{10.5}$$

$$= J_{k+1|k+1}^{-1} y_{k+1|k} + J_{k+1|k+1}^{-1} i_{k+1} \tag{10.6}$$

Therefore, in the Lie group setting, this mean needs to be subtracted from the Lie algebra part, which leads to the following reparametrisation

$$\hat{\chi}_{k+1|k+1} = \hat{\chi}_{k+1|k} \exp(J^{-1}_{k+1|k+1} i_{k+1}) \tag{10.7}$$

Note that this corresponds *exactly* to the update carried out by the LG-EKF or the IEKF in a similar setting.

The information form of Kalman filtering on Lie groups described herein is not suited for BIFM, as it makes heavy use of the inverse of the information matrix, J^{-1}, precisely the one we want to avoid. Therefore, we need to rewrite the propagation and update steps without explicitly relying on the mean of the distribution. That means, we need to stay on the Lie algebra. For this, we will make full use of the invariant framework machinery, and especially the properties of the group-affine dynamics. This relies on a different setting of the problem, which we first illustrate on the most standard form of Kalman smoother, the Rauch-Tung-Striebel (RTS) one.

10.2 Rauch-Tung-Striebel Kalman smoother on Lie groups

The RTS formulation of the Kalman smoother, in the linear case, relies on a standard Kalman filter for the forward sweep, while its backwards update reads in linear form

$$\hat{\chi}_{k|n} = \hat{\chi}_{k|k} + L_k(\hat{\chi}_{k+1|n} - \chi_{k+1|k}) \tag{10.8}$$

$$P_{k|n} = P_{k|k} + L_k(P_{k+1|n} - P_{k+1|k})L_k^T \tag{10.9}$$

$$L_k = P_{k|k} F_k^T P^{-1}_{k+1|k}. \tag{10.10}$$

full state vs. perturbation: two views of non-linear Kalman smoothing Extending the linear case, the backward sweep in non-linear Kalman smoothers should at first sight seek to obtain another estimation of the full state. The classic RTS formulation of the Kalman smoother can be easily adapted to Lie groups in this sense, for instance. Indeed, in the spirit of the IEKF, one would get the following update equation

$$\hat{\chi}_{k|n} = \hat{\chi}_{k|k} \exp(L_k \log(\hat{\chi}^{-1}_{k+1|k} \hat{\chi}_{k+1|n})). \tag{10.11}$$

However, it is far from clear how the backward information sweep of BIFM should be adapted in this manner, and even initialised. It starts with only the last innovation as information, which does not in general define a full state. Think for instance of a GPS measurement: it only constraints a part of the state, its position, but says nothing on its attitude, which would thus have infinite covariance. It rather gives a hint on how the state should be **updated**. Indeed, let us rewrite the innovation term of (10.11), given that

$$\hat{\chi}_{k+1|k} = f(\hat{\chi}_{k|k} \exp(-\xi_{k|k})) \tag{10.12}$$

$$\hat{\chi}_{k+1|n} = \hat{\chi}_{k+1|n} \exp(\xi_{k+1|n}) = f(\hat{\chi}_{k|k}) \exp(\xi_{k+1|n}) \tag{10.13}$$

Injecting in the innovation, this leads to

$$\log(\hat{\chi}^{-1}_{k+1|k} \hat{\chi}_{k+1|n}) = BCH(-F\xi_{k|k},\ \xi_{k+1|n}) \tag{10.14}$$

$$= BCH(-\xi_{k+1|k},\ \xi_{k+1|n}) \tag{10.15}$$

Based on this remark, it appears that non-linear Kalman smoothers could be seen as a way to refine the update to be applied, i.e., perturbations. Building up on the RTS Kalman smoother, we have, after the forward sweep:

$$\hat{\chi}_{k|k} = \hat{\chi}_{k|k-1} \exp(\xi_{k|k}) \tag{10.16}$$

We can consider that, during the backward sweep, we look for a refined estimate written as

$$\hat{\chi}_{k|n} = \hat{\chi}_{k|k-1} \exp(\xi_{k|n}) \tag{10.17}$$

This way, two other non-linear versions of the RTS Kalman smoother could be derived, as follows

A fully linearised approach

$$\hat{\chi}_{k|n} = \hat{\chi}_{k|k-1} \exp(\xi_{k|n}) \tag{10.18}$$
$$\xi_{k|n} = \xi_{k|k} + L_k(\xi_{k+1|n} - \xi_{k+1|k}), \tag{10.19}$$

where L_k is a gain which could take into account the Jacobian of the BCH equations.

A hybrid approach

$$\hat{\chi}_{k|n} = \hat{\chi}_{k|k-1} \exp(\xi_{k|k}) \exp(L_k(\xi_{k+1|n} - \xi_{k+1|k})), \tag{10.20}$$

where L_k could be a slightly different Jacobian.

The raises a few questions however, notably on the exact definition of $\xi_{k+1|k}$. We will investigate this in the case of BIFM.

10.3 Adapting the Information form of Kalman filtering on Lie groups for BIFM

The forward sweep of BIFM on Lie groups is not a problem, as it simply corresponds to an IEKF. We therefore study how to transpose the backward sweep, based on the conclusions obtained for the RTS.

BIFM's backward sweep: Backpropagation In BIFM, the backwards sweep starts with only the information coming from the last innovation. In general, its initial information matrix is therefore not invertible. Thus we cannot retrieve the mean associated to the distribution and apply backpropagation to it. However, in the group-affine framework it is not needed, if we look for updates. Indeed, the dynamics tells us that

$$\chi_{k+1} = f_k(\chi_k) \exp(w_k), \qquad \forall \chi,\ f_k(\chi) = \phi_k(\chi) \cdot u_k, \tag{10.21}$$

with ϕ_k a gorup automorphism, and u_k a group element, using Corollary 1. For the sake of readability, we drop the subscript k on the dynamics f, ϕ and u. Rewriting this with perturbations, as is done for smoothing for instance, gives

$$\hat{\chi}_{k+1} \exp(\xi_{k+1}) = f(\hat{\chi}_k \exp(\xi_k)) \exp(w_k) \tag{10.22}$$

In this case, χ_k needs to be isolated, and the updates ξ_i are parametrised by their information form $J_i^{-1} y_i$. This becomes, thanks to f^{-1} being group-affine,

$$\hat{\chi}_k \exp(J_k^{-1} y_k) = f^{-1}(\hat{\chi}_{k+1} \exp(J_{k+1}^{-1} y_{k+1}) \exp(-w_k)) \tag{10.23}$$
$$= f^{-1}(\hat{\chi}_{k+1})(f^{-1}(Id))^{-1} f(\exp(J_{k+1}^{-1} y_{k+1}) \exp(-w_k)) \tag{10.24}$$
$$= f^{-1}(\hat{\chi}_{k+1}) \exp(F^{-1} J_{k+1}^{-1} y_{k+1}) \exp(-F^{-1} w_k) \tag{10.25}$$
$$\exp(J_k^{-1} y_k) = \hat{\chi}_k^{-1} f^{-1}(\hat{\chi}_{k+1}) \exp(F^{-1} J_{k+1}^{-1} y_{k+1}) \exp(-F^{-1} w_k) \tag{10.26}$$

A first order expansion classically gives, ignoring the noise

$$J_k^{-1} y_k \approx a_k + F^{-1} J_{k+1}^{-1} y_{k+1}, \tag{10.27}$$

where $a_k = \log(\hat{\chi}_k^{-1} f^{-1}(\hat{\chi}_{k+1}))$. This is not in an acceptable form, since it involves inverted information matrices. We can rewrite it thanks to the linear case, as we know how to obtain J_k from J_{k+1}:

$$J_k = F^T (I + J_{k+1} Q)^{-1} J_{k+1} F \tag{10.28}$$

We can isolate y_k to obtain

$$y_k \approx F^T (I + J_{k+1} Q)^{-1} (J_{k+1} F a_k + y_{k+1}), \tag{10.29}$$

i.e., a form quite close to that of the linear case. The last question remaining is what Fa_k represents. Going back to the group through the exponential we get

$$\exp(Fa_k) = \phi(\hat{\chi}_k^{-1} f^{-1}(\hat{\chi}_{k+1})) \tag{10.30}$$
$$= \phi(\hat{\chi}_k)^{-1} \phi \circ f^{-1}(\hat{\chi}_{k+1}) \tag{10.31}$$
$$= f(\hat{\chi}_k)^{-1} \hat{\chi}_{k+1} \tag{10.32}$$

Thus, we finally obtain

$$y_k \approx F^T (I + J_{k+1} Q)^{-1} (J_{k+1} \alpha_k + y_{k+1}), \qquad \alpha_k = \log(f(\hat{\chi}_k)^{-1} \hat{\chi}_{k+1}) \tag{10.33}$$

The case of imperfect group-affine dynamics

This derivation used the properties of group-affine dynamics to obtain (10.33). However, this equation only uses Jacobians, group multiplications and logarithm, and can therefore be extended to imperfect gorup-affine dynamics, as was done for Invariant Smoothing. This simply means that further approximations are made.

BIFM's backward sweep: Update

The EIF defined in [50] aims at directly estimating a state in concentrated gaussian distribution form, and therefore uses the reparametrisation trick during the update. However, here we look for an update, defined on the Lie algebra. We thus do not use this trick, and instead maintain a non-zero mean distribution through

$$y_k^+ = y_k + H^T R^{-1} b_k \tag{10.34}$$

Remark 24. *The innovation z_k to use here is not clear, and might depend on how exactly the final estimate is formulated. Indeed, there are at least three ways to compute it:*

- *Use the innovation term of the propagated state in the forward sweep, i.e., $i_k - h(\chi_{k|k-1})$*
- *Use the innovation term of the updated state in the forward sweep, i.e., $i_k - h(\chi_{k|k})$*
- *Use the innovation term of the backpropagated fused term of BIFM, i.e., $i_k - h(f^{-1}(\chi_{k+1|n}))$*

BIFM: Final fusion

Once the forward and backward sweep of BIFM have been computed, there lies the final fusion to be carried out. The final estimate is looked for in the form

$$\chi_{k|n} = \hat{\chi}_{k|k-1} \exp(\xi_{k|n})$$

As was pointed out for the RTS smoother, there are at least two approaches that one could chose to compute the final estimate:

The fully linearised approach

$$\chi_{k|n} = \hat{\chi}_{k|k-1} \exp((I + P_{k|k} J_k)^{-1} (\xi_{k|k} + P_{k|k} y_k)) \tag{10.35}$$

The hybrid approach

$$\chi_{k|n} = \hat{\chi}_{k|k-1} \exp(\xi_{k|k}) \exp((I + P_{k|k} J_k)^{-1} (\xi_{k|k} + P_{k|k} y_k) - \xi_{k|k}) \tag{10.36}$$

In fact, there even is a third possible way of carrying out the update: simply considering the backward sweep as a kind of second iteration for the update

The sequential updates approach

$$\chi_{k|n} = \hat{\chi}_{k|k-1} \exp(\xi_{k|k}) \exp((I + P_{k|k} J_k)^{-1} P_{k|k} y_k) \tag{10.37}$$

10.4 Numerical validation of non-linear BIFM

Non-linear BIFM was validated through a first numerical simulation, using the same setup as in Section 4.4.6, i.e., in an imperfect group-affine case. In this case, the forward sweep of BIFM corresponds to the IEKF. The backward sweep was carried out using the sequential update approach of (10.37). No noticeable differences were observed when experimenting the other two approaches, which are therefore not reported herein. Then, using the updated value of the initial state output by the BIFM, a new forward sweep was conducted. Recall that a single BIFM (non-linear or not) iteration does not change its last estimated state, and that the IEKF estimation fails in this case. Therefore, a new forward sweep is necessary if one hopes to improve the output. The estimated heading and biases of the first and second forward sweeps are reported in Figure 10.1.

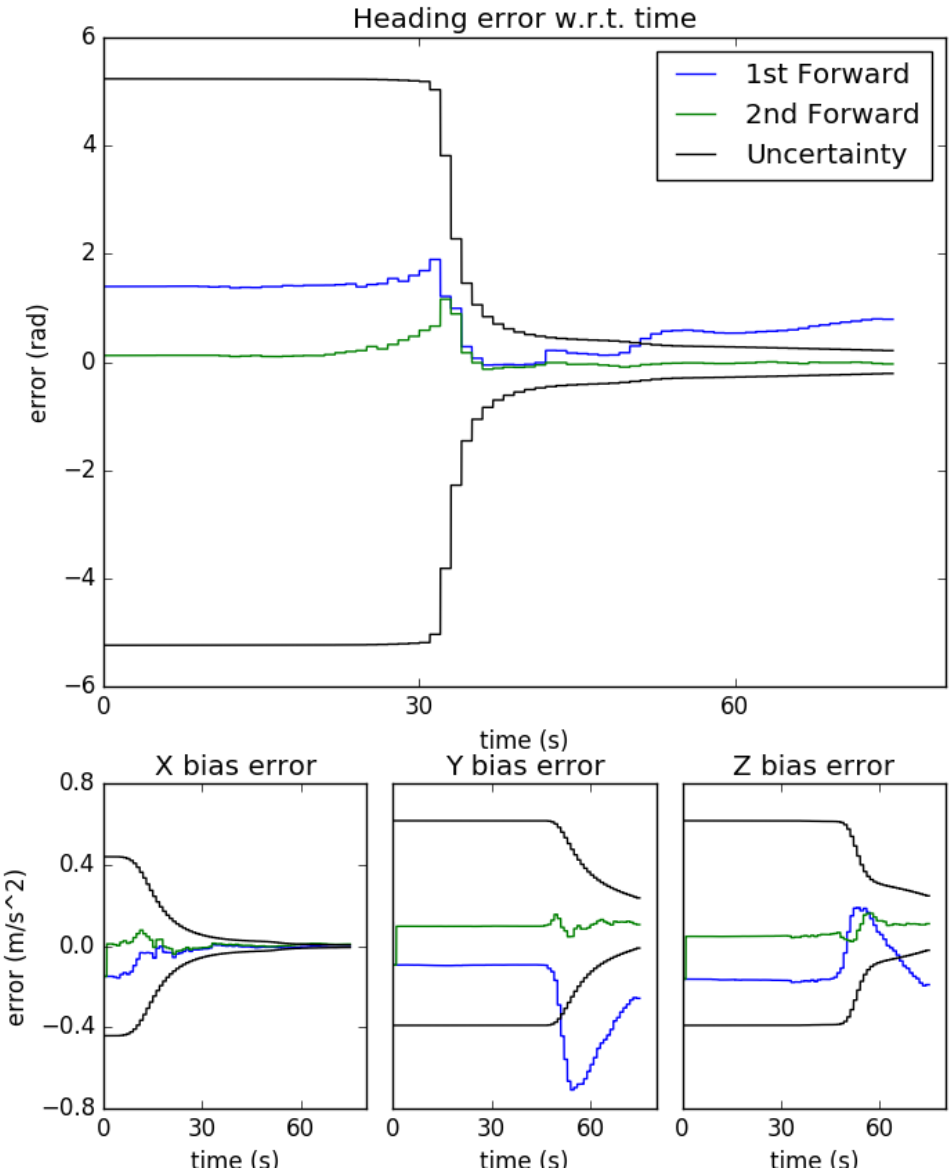


Figure 10.1: Comparison of the 1st (i.e., IEKF) and 2nd sweeps of the non-linear BIFM applied to inertia-GPS fusion, using the same data as in Section 4.4.6. It appears that the IEKF converges to a wrong solution, where the yaw and biases have been wrongly corrected, which leads to subsequent estimation problems. However, even with such a wrong estimate, the backward sweep of BIFM carries enough information back to the initial state, in order for the second forward estimation to provide consistent estimation.

10.5 Conclusion

This chapter explores how the Invariant framework could be combined with the linear (SC-)BIFM solver proposed in Chapter 8. It appears that this is far from obvious, as there are various ways of defining how the backward update and the final fusion could be carried out. In any case, this shed light on the difference between refining a state estimate, and an update of this state, a difference which is inherent to the non-linear case. While these questions remain open, non-linear BIFM proved its interest in a biased inertia-GPS alignment case, where it overcomes the issues the IEKF faced, as did L-IS in Section 4.4.6. These first results call for further experiments, in particular a detailed comparison with Invariant Smoothing.

Chapter 11

Conclusion of the thesis

This work investigated different aspects of smoothing as a state estimation paradigm in inertial navigation with precise sensors. It brings three main contributions.

The first contribution is in fact on invariant filtering, and served as a starting point for the remainder of the thesis. A novel property of the IEKF was proved, in the ideal context of noise-free dynamics. Indeed, in this case, if additional geometric information is available on the initial state, then the actual space the true state can be on is constrained. This information is seamlessly respected and leveraged along the estimation by the IEKF, while the EKF breaks this constraint if it is not artificially enforced. While that was an ideal limit case, it gave some explanations to the success of Invariant filtering in the presence of low process noise.

The second contribution was regarding the non-linear part of smoothing. Here, the extension of the invariant filtering framework to the smoothing paradigm is presented, leading to Invariant Smoothing. Just as the IEKF had proven superior to other filters, invariant smoothing was shown to behave better than other non-linear smoothing methods. This came from three main differences: the use of the exponential update, the way the Jacobians are computed, and, local navigation, the choice of the right-invariant parametrisation of the error. This once again confirmed that the invariant framework is particularly well-suited for inertial navigation. Moreover, Invariant Smoothing solved some of the problems that still existed with the IEKF, notably in the presence of strong biases. In order to benefit from these results in an industrial context, a versatile version of Invariant Smoothing was implemented in the navigation toolbox of Safrantech, which can be easily adapted to a number of different sensor fusion scenarios.

Finally, a new linear solver, SC-BIFM, has been proposed to solve the linear least-squares ensuing from non-linear smoothing. It is a sparsity-preserving exact solver, robust to a particular type of ill-conditioning of the smoothing information matrix, owing to a strong discrepancy between the various sensors' precisions. Its development was motivated by the fragility of the existing solvers against precise navigation (and in particular inertial) sensors, a shortcoming which was also brought to light during this work and could become more and more apparent in the futur years due to the continuous improvement of IMUs. It was experimentally validated and is now part of the navigation toolbox of the autonomous vehicle prototype of Safrantech.

In the meantime, several questions about smoothing in general were raised. First, while it improved IEKF results in some cases, expanding the sliding window size in smoothing sometimes showed little to no impact on the estimation results, which goes against the common view of smoothing. This was observed in several different settings, and both in global and local navigation. These first experimental observations should make the practitioner cautious when implementing such methods, and leave room for future research directions. Second, the impact of the prior covariance on the observability properties of a smoothing system were outlined on a toy system, using an approach based on smoothing updates, rather than the standard observability matrix. Finally, a novel non-linear estimation scheme, obtained as an extension SC-BIFM through the invariant framework, was proposed, and was shown to also solve some issues of the IEKF. However, its performances still need to be thoroughly evaluated, in particular compared to Invariant Smoothing.

Therefore, this thesis opened about as many doors as it closed. The invariant framework brought a deeper understanding of the existing non-linear smoothing methods, in particular in inertial navigation, with the preintegrated IMU factors. But once the parametrisation is set, there are still many aspects of smoothing which are not well understood, some of which appeared in this work. While SC-BIFM brings a first solution to the disregarded problem of ill-conditioned information matrices, it exhibits strong limits regarding incremental settings. On the other hand, the question of the sliding window width is a somewhat original approach of a research problem which gained a lot of attention: maintaining tractable systems in long-term estimation. Indeed, knowing when to expand the window is a first step to a more

fundamental question: when and which states can I marginalise? These questions, along with the crucial issue of dealing with outliers and sensor failure, are fundamental if one wants to provide autonomous systems with reliable information in a long-term setting.

Appendix A

Right-Invariant Smoothing factors

This appendix details the derivation of the factors used in R-IS in Section 4.5.1.

A.1 Prior factor

The prior factor must be expressed in terms of a difference consistent with the chosen parametrisation, i.e., the opposite as the one used in left-invariant smoothing (4.30). For some initial guess $\bar{\chi}$, we have:

$$\exp(x_0) = \bar{\chi}^{-1} \exp(\xi_0^R) \hat{\chi}_0 \tag{A.1}$$

$$x_0 \approx Ad_{\hat{\chi}_0}^{-1} (J_0^R \xi_0^R + \log(\hat{\chi}_0 \bar{\chi}^{-1})) \tag{A.2}$$

where J_0 is the Jacobian of the BCH formula, defined by $BCH(p_0, \xi) = p_0 + J_0 \xi + o(\|\xi\|^2)$ [6]. We also have $J_0 p_0 = p_0$. As in the lest-invariant case, it can not always be approximated as the identity. Moreover, we see that $Ad_{\hat{\chi}_0}$ must be applied. On $SE(3)$ for instance, this accounts for the variations of position induced by a correction of the rotation.

A.2 Group-affine factors

For group-affine dynamics, a first-order perturbation of the propagation writes:

$$\exp(w_i) = f(\hat{\chi}_i)^{-1} \exp(-F_i^R \xi_i^R) \exp(\xi_{i+1}^R) \hat{\chi}_{i+1} \tag{A.3}$$

$$\exp(Ad_{\hat{\chi}_{i+1}} w_i) = \hat{\chi}_{i+1} f(\hat{\chi}_i)^{-1} \exp(-F_i^R \xi_i^R) \exp(\xi_{i+1}^R) \tag{A.4}$$

$$Ad_{\hat{\chi}_{i+1}} w_i \approx a_i^R - F_i^R \xi_i^R + \xi_{i+1}^R, \tag{A.5}$$

where $a_i^R = \log(\hat{\chi}_{i+1} f(\hat{\chi}_i)^{-1})$. This means that linearisation will lead to a factor with a covariance corrected by $Ad_{\hat{\chi}_{i+1}}$. Thus, approximative independence of the information matrix to the estimate which had been obtained with the left-invariant error is lost here. However, we argue that this is balanced by the consistency induced by this parametrisation, as shown in Section 5.2.4.

A.3 Measurement factors

Similarly to the derivation of the measurement factors of L-IS in Section 4.2.2, the measurement factors are obtained through the following quantities

$$z_k \star \hat{\chi}^{-1} = b_k \star (\chi \hat{\chi}^{-1}) \ \exp_Y(\hat{v}_k), \tag{A.6}$$

where $\hat{v}_k$ is the noise associated to this innovation, and depends on the group action, see Section 2.1.3. Isolating this noise, injecting (4.18), and linearising lead to

$$\exp_Y(\hat{v}_k) = (b_k \star \exp(\Xi))^{-1} z_k \star \hat{\chi}^{-1} \tag{A.7}$$

$$\hat{v}_k \approx \log_Y((b_k \star \exp(\Xi))^{-1} (z_k \star \hat{\chi}^{-1})) \tag{A.8}$$

$$\approx c_k - H\Xi \tag{A.9}$$

where $c_k = \log_Y(z_k \star \hat{\chi}^{-1})$, H is the matrix given by a Taylor expansion of the matrix exponential in (2.2) of $\log_Y(b_k \star \exp(\Xi))$.

Bibliography

[1] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[2] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[3] Tim Bailey and Hugh F. Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[4] Tim Bailey, Juan Nieto, and Eduardo Nebot. Consistency of the fastslam algorithm. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 424–429, May 2006.

[5] Timothy Barfoot, James R Forbes, and Paul T Furgale. Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica*, 68(1-2):101–112, 2011.

[6] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[7] Timothy D. Barfoot and Paul. T. Furgale. Associating Uncertainty with Three-Dimensional Poses for Use in Estimation Problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

[8] A. Barrau and S. Bonnabel. Extended kalman filtering with nonlinear equality constraints: a geometric approach. *IEEE Transactions on Automatic Control*, pages 1–1, 2019.

[9] Axel Barrau. *Non-linear state error based extended Kalman filters with applications to navigation*. Theses, Ecole Nationale Supérieure des Mines de Paris, September 2015.

[10] Axel Barrau and Silvère Bonnabel. Intrinsic Filtering on SO(3) with Discrete-time Observations. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3255–3260. IEEE, 2013.

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

[12] Axel Barrau and Silvère Bonnabel. Intrinsic filtering on lie groups with applications to attitude estimation. *IEEE Transactions on Automatic Control*, 60(2):436 – 449, 2015.

[13] Axel Barrau and Silvère Bonnabel. Navigating with highly precise odometry and noisy GPS: a case study. *IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2016.

[14] Axel Barrau and Silvère Bonnabel. The invariant extended kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, 62(4):1797–1812, April 2017.

[15] Axel Barrau and Silvere Bonnabel. Three examples of the stability properties of the invariant extended kalman filter. *IFAC-PapersOnLine*, 50(1):431–437, 2017.

[16] Axel Barrau and Silvère Bonnabel. Invariant kalman filtering. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):null, 2018.

[17] Axel Barrau and Silvere Bonnabel. A Mathematical Framework for IMU Error Propagation with Applications to Preintegration. working paper or preprint, December 2019.

[18] Axel Barrau and Silvère Bonnabel. Linear observed systems on groups. *Systems Control Letters*, 129:36 – 42, 2019.

[19] B. M. Bell and F. W. Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, Feb 1993.

[20] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992.

[21] Gerald J Bierman. *Factorization Methods for Discrete Sequential Estimation*, volume 128. Academic Press, 1977.

[22] Silvère Bonnabel, Philippe Martin, and Pierre Rouchon. Symmetry-Preserving Observers. *IEEE Transactions on Automatic Control*, 53(11):2514–2526, December 2008.

[23] Silvère Bonnabel, Philippe Martin, and Pierre Rouchon. Non-linear symmetry-preserving observers on Lie groups. *IEEE Transactions on Automatic Control*, 54(7):1709–1713, 2009.

[24] Silvère Bonnabel, Philippe Martin, and Erwan Salaün. Invariant extended Kalman filter: theory and application to a velocity-aided attitude estimation problem. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 1297–1304. IEEE, 2009.

[25] Guillaume Bourmaud, Rémi Mégret, Marc Arnaudon, and Audrey Giremus. Continuous-Discrete Extended Kalman Filter on Matrix Lie Groups Using Concentrated Gaussian Distributions. *Journal of Mathematical Imaging and Vision*, 51(1):209–228, 2015.

[26] Guillaume Bourmaud, Rémi Mégret, Audrey Giremus, and Yannick Berthoumieu. Discrete extended Kalman filter on Lie groups. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, pages 1–5. IEEE, 2013.

[27] Guillaume Bourmaud, Rémi Mégret, Audrey Giremus, and Yannick Berthoumieu. From intrinsic optimization to iterated extended kalman filtering on lie groups. *J. Math. Imaging Vis.*, 55(3):284–303, July 2016.

[28] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, Sept 2017.

[29] M. Brossard, A. Barrau, and S. Bonnabel. Exploiting symmetries to design ekfs with consistency properties for navigation and slam. *IEEE Sensors Journal*, 19(4):1572–1579, Feb 2019.

[30] M. Brossard, S. Bonnabel, and A. Barrau. Unscented kalman filter on lie groups for visual inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 649–655, Oct 2018.

[31] Martin Brossard, Axel Barrau, and Silvere Bonnabel. RINS-W: Robust Inertial Navigation System on Wheels. Accepted at IROS 2019, March 2019.

[32] Martin Brossard and Silvere Bonnabel. Learning Wheel Odometry and IMU Errors for Localization. In *International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019.

[33] Martin Brossard, Silvère Bonnabel, and J.P. Condomines. Unscented kalman filtering on lie groups. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2485–2491, Sept 2017.

[34] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian D. Reid, and John J. Leonard. Simultaneous localization and mapping: Present, future, and the robust-perception age. *CoRR*, abs/1606.05830, 2016.

[35] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice. Generic node removal for factor-graph slam. *IEEE Transactions on Robotics*, 30(6):1371–1385, Dec 2014.

[36] P. Chauchat, A. Barrau, and S. Bonnabel. Kalman filtering with a class of geometric state equality constraints. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2581–2586, Dec 2017.

[37] Paul Chauchat, Axel Barrau, and Silvere Bonnabel. Invariant Smoothing on Lie Groups. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018*, Madrid, Spain, October 2018.

[38] Gregory S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 1: Classical Results and Geometric Methods*. Applied and numerical harmonic analysis. Birkhäuser, 2009.

[39] Gregory S Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011.

[40] Han-Pang Chiu, Stephen Williams, Frank Dellaert, Supun Samarasekera, and Rakesh Kumar. Robust vision-aided navigation using sliding-window factor graphs. In *2013 IEEE International Conference on Robotics and Automation*, pages 46–53. IEEE, 2013.

[41] S. Choudhary, V. Indelman, H. I. Christensen, and F. Dellaert. Information-based reduced landmark slam. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4627, May 2015.

[42] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. Learning to solve nonlinear least squares for monocular stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018.

[43] John L. Crassidis, F. Landis Markley, and Yang Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.

[44] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, March 2002.

[45] Frank Dellaert and Michael Kaess. Square root sam: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, 2006.

[46] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.

[47] 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, December 1968.

[48] Gamini Dissanayake, Paul Newman, Hugh F. Durrant-Whyte, Steven Clark, and M. Csobra. A solution to the simultaneous localisation and mapping (slam) problem. *IEEE Trans. Robot. Automat.*, 17:229–241, 2001.

[49] Hugh F. Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[50] Josip Ćesić, Ivan Marković, Mario Bukal, and Ivan Petrović. Extended information filter on matrix lie groups. *Automatica*, 82:226 – 234, 2017.

[51] N. Filipe, M. Kontitsis, and P. Tsiotras. Extended kalman filter for spacecraft pose estimation using dual quaternions. In *2015 American Control Conference (ACC)*, pages 3187–3192, July 2015.

[52] James Richard Forbes and David Evan Zlotnik. Sigma point kalman filtering on matrix lie groups applied to the slam problem. In *Geometric Science of Information*, pages 318–328. Springer International Publishing, 2017.

[53] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, Feb 2017.

[54] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, April 2017.

[55] J. Gallier. Notes on differential geometry and lie groups, 2016. University of Pennsylvannia.

[56] C. L. Gentil, T. Vidal-Calleja, and S. Huang. In2lama: Inertial lidar localisation and mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6388–6394, May 2019.

[57] J. S. Goddard and Mongi A. Abidi. Pose and motion estimation using dual quaternion-based extended Kalman filtering. In Richard N. Ellson and Joseph H. Nurre, editors, *Three-Dimensional Image Capture and Applications*, volume 3313, pages 189 – 200. International Society for Optics and Photonics, SPIE, 1998.

[58] Giorgio Grisetti, Ramier Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, winter 2010.

[59] Jens-Steffen Gutmann and Kurt Konolige. Incremental mapping of large cyclic environments. In *Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings. 1999 IEEE International Symposium on*, pages 318–325, 1999.

[60] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop kf: Learning discriminative deterministic state estimators. In *Advances in Neural Information Processing Systems*, pages 4376–4384, 2016.

[61] Ross Hartley, Maani Ghaffari Jadidi, Ryan M. Eustice, and Jessy W. Grizzle. Contact-aided invariant extended kalman filtering for robot state estimation. *CoRR*, abs/1904.09251, 2019.

[62] Ross Hartley, Maani Ghaffari Jadidi, Jessy Grizzle, and Ryan M Eustice. Contact-aided invariant extended kalman filtering for legged robot state estimation. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

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

[64] Jerry Hsiung, Ming Hsiao, Eric Westman, Rafael Valencia, and Michael Kaess. Information sparsification in visual-inertial odometry. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS*, October 2018.

[65] Baichuan Huang, Jun Zhao, and Jingbin Liu. A survey of simultaneous localization and mapping, 2019.

[66] Guoquan Huang. Visual-inertial navigation: A concise review. *2019 International Conference on Robotics and Automation (ICRA)*, pages 9572–9582, 2019.

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

[68] K. Huang, J. Xiao, and C. Stachniss. Accurate direct visual-laser odometry with explicit occlusion handling and plane detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1295–1301, May 2019.

[69] Shoudong Huang. A review of optimisation strategies used in simultaneous localisation and mapping. *Journal of Control and Decision*, 6(1):61–74, 2019.

[70] Shoudong Huang and Gamini Dissanayake. A critique of current developments in simultaneous localization and mapping. *International Journal of Advanced Robotic Systems*, 13(5):1729881416669482, 2016.

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

[72] Michael Kaess and Frank Dellaert. Covariance recovery from a square root information matrix for data association. *Robotics and autonomous systems*, 57(12):1198–1210, 2009.

[73] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J. Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31:217–236, 2012.

[74] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[75] Martin Kobilarov, Duy-Nguyen Ta, and Franck Dellaert. Differential dynamic programming for optimal estimation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 863–869, May 2015.

[76] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, May 2011.

[77] Kruno Lenac, Josip Ćesić, Ivan Marković, and Ivan Petrović. Exactly sparse delayed state filter on lie groups for long-term pose graph slam. *The International Journal of Robotics Research*, 37(6):585–610, 2018.

[78] Haomin Liu, Mingyu Chen, Guofeng Zhang, Hujun Bao, and Yingze Bao. Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[79] H. Loeliger, L. Bruderer, H. Malmberg, F. Wadehn, and N. Zalmai. On sparsity by nuv-em, gaussian message passing, and kalman smoothing. In *2016 Information Theory and Applications Workshop (ITA)*, pages 1–10, Jan 2016.

[80] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.

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

[82] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *CVPR*, 2019.

[83] J. Maley and G. Huang. Unit quaternion-based parameterization for point features in visual navigation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6880–6886, Oct 2018.

[84] Philippe Martin and Erwan Salaün. Design and implementation of a low-cost observer-based attitude and heading reference system. *Control Engineering Practice*, 18(7):712–722, February 2010.

[85] A. I. Mourikis, S. I. Roumeliotis, and J. W. Burdick. Sc-kf mobile robot localization: A stochastic cloning kalman filter for processing relative-state measurements. *IEEE Transactions on Robotics*, 23(4):717–730, Aug 2007.

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

[87] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, Oct 2017.

[88] Peter J. Olver. *Classical Invariant Theory.* London Mathematical Society Student Texts. Cambridge University Press, 1999.

[89] Gianfranco Parlangeli, Paola Pedone, and Giovanni Indiveri. Relative pose observability analysis for 3d nonholonomic vehicles based on range measurements only. *IFAC Proceedings Volumes*, 45(27):182 – 187, 2012. 9th IFAC Conference on Manoeuvring and Control of Marine Craft.

[90] M. Pilté, S. Bonnabel, and F. Barbaresco. Tracking the frenet-serret frame associated to a highly maneuvering target in 3d. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1969–1974, Dec 2017.

[91] Tong Qin, Peilian Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, Aug 2018.

[92] Herbert E Rauch, CT Striebel, and F Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[93] T. Sandy, L. Stadelmann, S. Kerscher, and J. Buchli. Confusion: Sensor fusion for complex robotic systems using nonlinear optimization. *IEEE Robotics and Automation Letters*, 4(2):1093–1100, April 2019.

[94] Weizhao Shao, Srinivasan Vijayarangan, Cong Li, and George Kantor. Stereo visual inertial lidar simultaneous localization and mapping. *CoRR*, abs/1902.10741, 2019.

[95] Dan Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory Applications*, 4(8):1303–1318, August 2010.

[96] Ashutosh Singandhupe and Hung La. A review of slam techniques and security in autonomous driving. pages 602–607, 02 2019.

[97] Cyrill Stachniss, John J Leonard, and Sebastian Thrun. Simultaneous localization and mapping. In *Springer Handbook of Robotics*, pages 1153–1176. Springer, 2016.

[98] Jeffrey Uhlmann. Unit consistency, generalized inverses, and effective system design methods. *CoRR*, abs/1604.08476, 2016.

[99] Jeffrey. Uhlmann. A generalized matrix inverse that is consistent with respect to diagonal transformations. *SIAM Journal on Matrix Analysis and Applications*, 39(2):781–800, 2018.

[100] J. Vallvé, J. Solà, and J. Andrade-Cetto. Graph slam sparsification with populated topologies using factor descent optimization. *IEEE Robotics and Automation Letters*, 3(2):1322–1329, April 2018.

[101] F. Wadehn, L. Bruderer, V. Sahdeva, and H. Loeliger. New square-root and diagonalized kalman smoothers. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1282–1290, Sep. 2016.

[102] A. Walsh, J. Arsenault, and J. R. Forbes. Invariant sliding window filtering for attitude and bias estimation. In *2019 American Control Conference (ACC)*, pages 3161–3166, July 2019.

[103] Kejian Wu, Ahmed Ahmed, Georgios A Georgiou, and Stergios I Roumeliotis. A square root inverse filter for efficient vision-aided inertial navigation on mobile devices. In *Robotics: Science and Systems*, volume 2, 2015.

[104] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Proceedings of Robotics: Science and Systems Conference*, July 2014.

[105] L. Zhao, S. Huang, and G. Dissanayake. Linear slam: A linear solution to the feature-based and pose graph slam based on submap joining. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 24–30, Nov 2013.

[106] Sheng Zhao, Yiming Chen, Haiyu Zhang, and Jay A. Farrell. Differential gps aided inertial navigation: a contemplative realtime approach. *IFAC Proceedings Volumes*, 47(3):8959 – 8964, 2014. 19th IFAC World Congress.

RÉSUMÉ

Les systèmes mobiles ont besoin de se localiser toujours plus précisément, dans des situations toujours plus complexes. C'est particulièrement vrai pour les systèmes autonomes, dont le contrôle de l'erreur de positionnement est une question de sécurité critique. Pour cela, ils sont équipés de différents capteurs, dont les données sont fusionnées pour obtenir une estimation de la localisation du véhicule, soit globalement (par GPS par exemple), soit localement, par rapport à son environnement direct (grâce à des caméras par exemple).

Cette thèse porte sur les algorithmes de localisation basée sur de la fusion de capteurs, le filtrage et le lissage, dans le cas où le mobile est équipé de capteurs inertiels de haute qualité. La première partie se concentre sur les non-linéarités inhérentes à la navigation inertielle utilisant des capteurs de haute qualité, et démontre comment, en s'appuyant sur la théorie du filtrage invariant, elles peuvent être mieux intégrées à la structure des algorithmes de filtrage et de lissage, grâce à la théorie du filtrage invariant. La seconde partie s'intéresse aux problèmes encourus, du fait de la présence de capteurs très précis, par les solveurs linéaires utilisés à chaque itération des algorithmes de lissage non-linéaires. Un nouveau solveur de problèmes de moindres carrés linéaires est introduit, qui s'affranchit de ces problèmes.

MOTS CLÉS

Navigation - SLAM - Inertiel - Filtrage - Lissage - Invariant - Estimation

ABSTRACT

Mobile systems need to locate themselves ever more accurately, and in ever more complex situations. This is in particular true for autonomous systems, for which controlling the position error is a critical safety issue. To this end, they are endowed with various sensors, the data of which are fused to obtain an estimate of the vehicle's location, either globally (with the GPS for instance), or locally, with respect to its surroundings (with cameras for instance).

This thesis investigates algorithms for localisation by sensor fusion, namely filtering and especially smoothing, when the mobile is equipped with high-grade inertial sensors. The first part deals with the nonlinear consequences of the use of high-grade inertial sensors, and demonstrates how the nonlinear structure of both filtering and smoothing algorithms may be improved by leveraging the invariant filtering framework. The second part deals with the problems incurred by the linear solvers that are used at each step of nonlinear smoothing algorithms as a result of having highly precise sensors. It introduces a novel least-squares linear solver that solves the issues.

KEYWORDS

Navigation - SLAM - Inertial - Filtering - Smoothing - Invariant - Estimation