PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Mines Paris - PSL

Covariance-Regularized Optimal Control through Extended Kalman Filter Backpropagation

Commande optimale régularisée par la covariance au moyen de la rétropropagation du filtre de Kalman étendu

Soutenue par
Jonas Benhamou
Le 11 mai 2026

École doctorale n°621
ISMME

Spécialité
Informatique temps réel, robotique et automatique - Paris

Préparée au
Centre de Robotique

Composition du jury :

Paolo ROBUFFO GIORDANO Directeur de recherche, INRIA	*Président du jury* *Rapporteur*
Frédéric JEAN Professeur, ENSTA	*Rapporteur*
Stéphane CARON Chargé de recherche, CNRS	*Examinateur*
Camille CHAPDELAINE Ingénieur, Safran	*Examinateur*
Lucia PALLOTINO Professeur, Université de Pise	*Examinatrice*
Silvère BONNABEL Professeur, Mines Paris-PSL	*Directeur de thèse*

MINES PARIS | PSL

Remerciements

Se lancer dans une thèse est une véritable aventure, faite de sprints, d'orages, de découvertes, et bien d'autres choses encore. Bien que ce soit une aventure en apparence solitaire, je l'ai vécue bien accompagné et entouré, et je remercie chaque personne que j'ai croisée sur ce chemin.

À commencer par mon directeur de thèse, Silvère, qui a été la boussole de cette aventure. Merci pour tes conseils, ton encouragement, et pour avoir partagé sans compter tes connaissances et ta passion. Un grand merci également à Camille pour ton implication, nos discussions et ton accompagnement tout au long du chemin.

Merci au jury d'avoir accepté de m'évaluer, et notamment aux deux rapporteurs, Frédéric Jean et Paolo Robuffo Giordano, pour leurs lectures minutieuses du manuscrit ainsi que pour leurs remarques.

Étant en thèse CIFRE, mon aventure s'est déroulée sur deux navires que j'ai tout autant appréciés. Un grand merci à l'ensemble de mes collègues de Safran, et en particulier à l'équipe du « petit fantôme ». Les moments passés avec vous ont été conviviaux, joyeux, mais aussi enrichissants. Merci également à l'ensemble du centre de robotique pour l'atmosphère qui facilite la recherche et favorise les rencontres. En particulier, un grand merci à tous les doctorants (et post-docs) qui ont accepté de ramer à mes côtés : Colin, Jules, Louis $\times 2$, Hugo, Fabio, Agapius, Clément, Nathan, Julien, Souheib, Mohamed, Ugo et Wassil. Merci pour nos échanges, nos rigolades, et nos quiz de culture générale. Ce fut à chaque fois un grand plaisir.

Merci à tous mes amis qui me soutiennent et m'entourent depuis mon plus jeune âge, mais aussi au quotidien. J'apprécie chacun de nos moments partagés : sorties, repas, activités, randonnées et bavardages.

Merci à mes parents et grands-parents qui m'accompagnent depuis le premier jour, ainsi qu'à mes sœurs, mon frère et mes beaux-frères. Vous êtes bien trop présents dans ma vie pour que je puisse énumérer tout ce que vous m'apportez. Merci à vous, ma famille et ma belle-famille. Merci à mon petit bonhomme d'avoir rejoint le navire après seulement deux jours de thèse.

Enfin, merci à toi, ma Alisa. Tu m'apportes énormément de joie, et l'aventure à tes côtés est toujours un immense bonheur.

Merci à vous tous d'avoir transformé cette aventure scientifique en une aventure humaine.

Résumé

La localisation et la perception sont des fonctions fondamentales d'un système autonome. Elles reposent sur des données issues de capteurs, généralement partielles et incertaines. Dans certains cas, la trajectoire suivie par le système influence fortement la qualité de ces données : certaines trajectoires sont peu informatives et conduisent à une estimation imprécise, tandis que d'autres améliorent sensiblement la perception/localisation. Exploiter ce phénomène pour planifier des trajectoires qui optimisent la qualité de l'information constitue le principe de la perception active, au cœur de cette thèse.

La première contribution consiste à formuler ce problème dans un cadre de contrôle optimal déterministe, en intégrant explicitement l'incertitude issue d'un filtre de Kalman étendu (EKF) le long de la trajectoire. La covariance du filtre est utilisée pour modéliser la qualité de la perception, et un terme lié à cette covariance est ajouté à la fonction de coût afin de favoriser les trajectoires les plus informatives. Cette intégration permet non seulement d'améliorer la perception et la localisation, mais aussi de renforcer la validité du cadre déterministe utilisé pour la planification.

La seconde contribution porte sur la résolution efficace de ce problème. Nous enrichissons les équations de rétropropagation d'un filtre de Kalman étendu afin de calculer analytiquement le gradient de la covariance par rapport aux variables influençant la trajectoire. Ces gradients se révèlent particulièrement adaptés à la perception active et rendent les problèmes d'optimisation numériquement gérables. Nous montrons enfin que cette approche est générale et applicable à un large éventail de situations, notamment la localisation, la calibration en ligne de capteurs et l'optimisation de séquences d'acquisition de mesures.

Mots clés : Perception Active, Filtre de Kalman Etendu, Rétropropagation, Planification de Trajectoires

Abstract

Localization and perception are fundamental functions of an autonomous system. They rely on sensor data, which are generally partial and uncertain. In some cases, the trajectory followed by the system strongly affects the quality of these data: certain trajectories are weakly informative and lead to poor estimation accuracy, while others significantly improve perception and localization. Exploiting this phenomenon to plan trajectories that optimize information quality is the core principle of active sensing, which lies at the heart of this thesis.

The first contribution is to formulate this problem within a deterministic optimal control framework, explicitly incorporating the uncertainty provided by an Extended Kalman Filter (EKF) along the trajectory. The filter covariance is used to model perception quality, and a term related to this covariance is added to the cost function in order to promote more informative trajectories. This integration not only improves perception and localization but also strengthens the validity of the deterministic framework used for planning.

The second contribution concerns the efficient solution of this problem. We extend the backpropagation equations of an Extended Kalman Filter to analytically compute the gradient of the covariance with respect to the variables that influence the trajectory. These gradients are particularly well suited for active perception and make the optimization problems tractable. Finally, we show that this approach is general and applicable to a wide range of situations, including localization, online sensor calibration, and optimal measurement selection.

Keywords : Active Sensing, Extended Kalman Filter, Backpropagation, Trajectory Generation

Contents

Remerciements i

Résumé ii

Abstract iii

Table of Contents iii

List of figures vii

List of Tables x

1 Introduction 1
- 1 Context . . . 2
- 2 Motivation for the Work . . . 4
- 3 Publications . . . 4
- 4 Structure of the Manuscript . . . 5

2 Planning in Robotics Under Uncertainty: A Brief Historical Perspective 7
- 1 POMDPs: A General Framework for Decision-Making under Uncertainty . . . 8
  - 1.1 Brief Introduction to Markov Decision Process . . . 8
  - 1.2 Some Remarks on POMDPs . . . 9
  - 1.3 Low-Complexity Alternatives to POMDPs . . . 11
- 2 Active Sensing Background . . . 13
  - 2.1 Dual Control Theory . . . 13
  - 2.2 Theory of Optimal Experimental Design . . . 15
- 3 Uncertainty-Aware Sensing . . . 18
  - 3.1 Active Sensing . . . 18
    - 3.1.1 Problem Formulation . . . 19
    - 3.1.2 Computational Simplifications . . . 20
    - 3.1.3 Active Sensing Applications . . . 22
  - 3.2 Perception-aware Planning . . . 22
  - 3.3 Measurements Selection . . . 25
  - 3.4 Optimal Sensor Placement . . . 25
- 4 Uncertainty-Aware Control . . . 27
- 5 Conclusion . . . 29

3 Overall Contributions of the Present Thesis 31
- 1 Positioning Our Contributions Within the Literature . . . 31

2 Our Approach to the Field of Active Sensing . . . 34
2.1 Contributions of the Present Thesis . . . 35

4 General Backpropagation Equations through Extended Kalman Filter 37
1 The Extended Kalman Filter . . . 38
2 Analytical Result on Active Sensing . . . 40
3 Gradient Computation Through the EKF . . . 42
3.1 Considered Problems . . . 42
3.2 Overview of Gradient Computation in EKF . . . 43
3.3 Fundamental Principles of Backpropagation . . . 44
3.4 Matrix Derivatives . . . 45
4 Backpropagation Through the Extended Kalman Filter . . . 46
4.1 Backpropagation Through the Propagation Step . . . 46
4.2 Backpropagation Through the Update Step . . . 48
5 Conclusion . . . 49

5 Minimum-Covariance Active Sensing with the Bicycle Model 50
1 Considered Problems . . . 52
1.1 Bicycle Model . . . 52
1.2 Motivating Example for Active Sensing . . . 53
1.3 Optimal Motion Planning Problem . . . 54
1.4 Motion Planning Algorithm . . . 56
1.5 Gradient Calculation via Analytical Backpropagation . . . 57
2 Simulation Experiments . . . 58
2.1 Simulation Results . . . 58
2.2 Computation Time . . . 58
2.3 Discussion . . . 61
3 Real-World Experiments . . . 62
3.1 Experimental Setting . . . 62
3.2 Results . . . 63
3.3 Analysis . . . 64
4 Conclusion . . . 65

6 Minimum-Covariance Active Sensing with Quadrotor Models 66
1 Minimum Covariance Active Sensing Method . . . 68
1.1 Overall Architecture . . . 68
1.2 Quadrotor Control Subsystem . . . 68
1.2.1 Control System Equations . . . 68
1.2.2 Differential Flatness for Quadrotors . . . 70
1.3 Details About Planning Ahead of Time . . . 71
1.4 Final Optimal Control Problem . . . 71
1.5 Backprop-based Optimization Scheme . . . 72
2 Application of Active Sensing to Collision Avoidance . . . 72
2.1 Dynamic Collision Avoidance with Bearing Measurements . . . 73
2.2 Results . . . 74
2.3 Comparisons with Prior Metrics . . . 75
3 Application to IMU-GNSS fusion . . . 77
3.1 Enhancing IMU-GNSS Localization via Active Sensing . . . 77
3.2 Results . . . 79
3.3 Gradient Computation Time . . . 80

4 Discussion . . . 82
4.1 Global Optimality Analysis . . . 82
4.2 Trade-Off Between Smoothness and Observability . . . 83
5 Conclusion . . . 84

7 Optimal Sensor Selection via Backpropagation Through the EKF 85
1 Measurement Selection in EKF . . . 87
1.1 Optimization-Based Measurement Selection . . . 87
1.2 Measurement Selection in EKF . . . 88
1.3 Application . . . 89
1.4 Simulation Results . . . 90
2 Mobile Sensor Trajectory Optimization . . . 93
2.1 Simulation Setting . . . 93
2.2 Considered Problem . . . 94
2.3 Simulation Results . . . 95
3 Conclusion . . . 98

Conclusion and Perspectives 99

Bibliographie 101

List of figures

1.1 Functional block diagram required for the autonomous vehicle from [Van Brummelen, 2018]. . . . 2
1.2 Where to take the second elevation angle measurement to estimate the height of the Eiffel Tower with minimal uncertainty. . . . 4

2.1 This diagram shows how different approaches to handling uncertainty in robotics are organized. It progresses from the most general theory to more specific ones. . . 13
2.2 Illustrations of four active sensing applications.Figure 1, adapted from [Napolitano, 2021], shows localization improvement using landmarks with limited range. Figure 2, from [Preiss, 2018], addresses the calibration of an IMU-GNSS system. Figure 3, taken from [Salaris, 2019], focuses on landmark-based localization. Figure 4, from [Flayac, 2019], illustrates terrain correlation-based map localization. 23
2.3 This figure, from [van den Berg, 2011], illustrates the result of their chance-constrained algorithm. We observe that the uncertainty ellipses do not intersect with the obstacles. . . . 29
2.4 This diagram shows how different approaches to handling uncertainty in robotics are organized. It progresses from the most general theory to more specific ones. For each approach, we present a representative optimization problem to be solved. 30

3.1 Illustration of the control problem aimed at reducing the risk of collision when the relative distance is unknown. One relevant question is how to adjust the altitude (up or down?) of our drone to decrease the risk. . . . 32

4.1 A basic example of a computational graph. . . . 44
4.2 Computational graph of propagation step. . . . 46
4.3 Computational graph of update step. . . . 48

5.1 Two vehicles with different initial states (position and lever arm). When moving in a straight line, the GNSS measurements are identical, and the system is not observable. During a turn, however, the GNSS measurements differ, enabling us to distinguish between the orange and green configurations. . . . 54
5.2 Monte Carlo simulation of process noise propagation for two trajectories: one straight (right) and one turning (left). The shape of the particle distribution differs significantly, illustrating the influence of trajectory choice on uncertainty propagation. . . . 55

5.3 On the left, an example of an initial random guess in blue. The other two trajectories are solutions to the active sensing problem where the loss is the trace (in orange) and the Schatten norm (in green). The right plot shows the expected trace of the covariance evolution for each trajectory. 59
5.4 Absolute estimation error of the lever arm during the trajectory. On the left the error for the lever arm in x and on the right in y. One-σ envelope illustrates the dispersion of errors over trials. Although the error is positive, the envelope may take negative values since it represents the mean error $\pm$ one standard deviation. 60
5.5 Final MAE for 10 different random trajectories with 20 trials (in blue) and optimal trajectories where the loss is either the trace (in orange) or the Schatten norm (in green). 60
5.6 Evolution of the smallest eigenvalue of the local observability Gramian for active sensing trajectories (green and orange) and the initial random trajectory (blue). The result shows that active sensing improves the system's local observability. . . 62
5.7 Off-road trajectories of the RTK-GNSS in a local tangent plane coordinates oriented East-North-Up of the scenario at 5 km/h. 64
5.8 On the left, trajectories of the RTK-GNSS in local tangent plane coordinates at 10 km/h. On the right, the evolution of the covariance trace along each trajectory in the scenario at a speed of 10 km/h. 64

6.1 Framework for optimal active sensing. The estimation block and the control block operate in parallel and are tightly coupled. The objective is to improve estimation by acting on the control block. The estimated system may be the quadrotor itself, or another system, observed through the quadrotor. Backpropagation efficiently relates the cost to the flat output. 69
6.2 Projected trajectories of the observer quadrotor robot with (blue) and without (green) active sensing for intruder (black) avoidance. The reference trajectory is shown as waypoints (red crosses). The top plot shows the xy-plane projection and the bottom plot shows the xz-plane projection. 75
6.3 Relative distance estimation error averaged over 100 Monte Carlo simulations. Results are shown without active sensing (green) and with active sensing using E^{2}LOG [Preiss, 2018] (purple), FIM-based sensing [Li, 2023] (burgundy), and our covariance-based method (blue). 77
6.4 The top figure shows the drone's trajectory from above. The figure below shows the evolution of the altitude during the trajectory. The reference trajectory must pass through the red waypoints. The optimal trajectory minimizes a combination of energy and state uncertainty, while being constrained to stay within a 3 m radius about the reference trajectory. 80
6.5 The top figure shows the evolution of the (x-axis) acceleration over time for the reference trajectory (green) and the active sensing trajectory (blue). The bottom plot displays the 1σ uncertainty envelope on the yaw angle for both the reference and the active sensing trajectory. 81
6.6 The attitude estimation RMSE for the reference and active sensing trajectories was computed using 10^3 Monte Carlo simulations. 82
6.7 Optimized trajectories from 100 randomized initializations, showing similar deformation patterns. In the bottom plot, almost all of the accelerations along the x-axis coincide. 83

7.1 Measurement selection sequence obtained using three different methods, including our PGD approach. The greedy and PGD methods both exhibit a clear transition between a transient phase with dense measurements and a steady-state phase with periodic updates. 91
7.2 Final loss values for 10^4 random measurement selection sequences, as well as the greedy, local search, and PGD methods. PGD was run 100 times with random initialization. The red dashed line indicates the loss when all measurements are used. 92
7.3 Computation time of the optimization process as a function of trajectory length, for both PGD and the greedy method. 93
7.4 The four optimal trajectories start from the same point and move between the points of interest (red crosses). The direction of the ocean current is indicated by a blue arrow. The black color gradient indicates the spatial density followed by the ergodic controller. The more informative the zone, the higher the probability that the robot will end up in it. 97
7.5 Comparison of the three trajectories and the resulting final temperature uncertainty. The ergodic trajectory yields significantly less accurate estimates than those obtained from our method. For readability, the clarity-based trajectory is omitted due to its large covariances, which would distort the plot. 98

List of Tables

5.1 Average and standard deviation of gradient calculation time (over 100 calculations) using different methods. 59
5.2 Estimation error according to (5.8) for all trajectories 63

6.1 Average and standard deviation of gradient computation times (over 100 computations) using different methods. 81
6.2 Comparison of smoothness and uncertainty across three different scenarios. . . . 84

7.1 Monte Carlo results over 100 trials. Our method achieves a lower RMSE. 96

Chapter 1

Introduction

Contents

1 Context 2
2 Motivation for the Work 4
3 Publications 4
4 Structure of the Manuscript 5

Résumé

Ce chapitre présente le contexte général des systèmes autonomes, composés de quatre blocs fonctionnels : perception, localisation, planification et contrôle. Lorsque le système évolue en boucle fermée, la perception et la localisation sont influencées par la planification et le contrôle. Exploiter cette rétroaction pour améliorer la qualité des données acquises constitue le principe de la perception active.

Les systèmes robotiques doivent également composer avec des incertitudes issues du modèle dynamique, des capteurs et de l'environnement, rendant la localisation et la navigation particulièrement difficiles, notamment dans des environnements encombrés ou dynamiques. Pour y remédier, des approches probabilistes ont été développées, parmi lesquelles la perception active qui permet de limiter l'impact de ces incertitudes. Une illustration du concept de perception active est que la localisation est d'autant plus précise que l'on se place à proximité de repères visuels connus. Par exemple, se tenir sous la tour Eiffel plutôt que dans une rue voisine réduit l'incertitude d'estimation.

Dans cette optique, la thèse propose un cadre général de mise en œuvre de la perception active, fondé sur la minimisation de la covariance estimée par un filtre de Kalman étendu (EKF). Une contribution majeure réside dans la résolution du problème par une méthode de calcul de gradient dans un EKF. Enfin, une part importante du manuscrit est consacrée aux applications de cette formulation à divers cas d'étude : calibration extrinsèque, alignement d'une centrale inertielle, suivi de cibles et estimation de champs scalaires.

1 Context

Autonomous systems are increasingly playing a key role across a wide range of domains and environments whether on land, in the air, or at sea. These systems are inherently complex, making it challenging to fully understand all the components required for their operation. From a systems architecture perspective, several definitions and terminologies exist to describe the functional blocks [Pendleton, 2017; Velasco-Hernandez, 2020]. Despite these variations, four blocks can be identified as essential: perception, localization, planning, and control. **Perception** enables the system to sense its surrounding environment, typically using cameras, LiDAR, or radar. Sensor measurements are then processed by dedicated signal processing algorithms. **Localization** estimates the intrinsic state of the vehicle, such as its position and orientation, relying on sensors like IMUs, odometers, GNSS, or magnetometers, and is typically fused via a Kalman filter. **Planning** is responsible for computing trajectories and making decisions, while **control** translates these decisions into physical actions, such as sending commands to the motors.

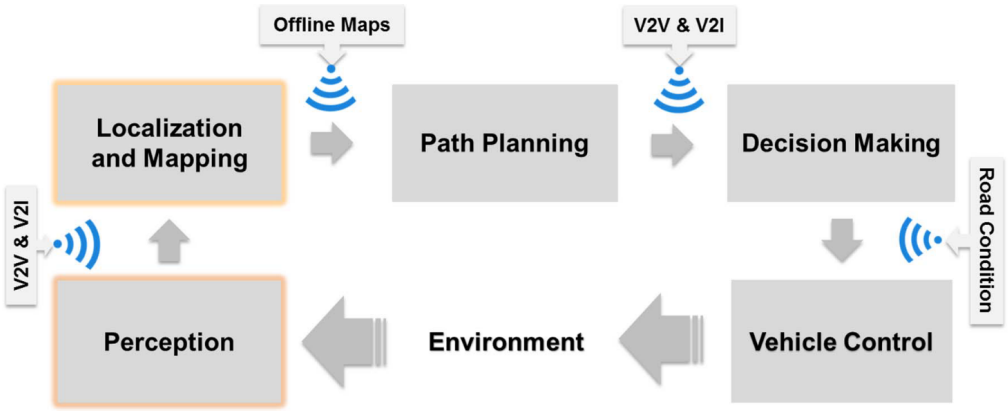

Figure 1.1: Functional block diagram required for the autonomous vehicle from [Van Brummelen, 2018].

These four blocks do not operate independently; they are deeply interconnected. Some links are straightforward, while others are more subtle. Especially planning directly impacts localization. For example, a trajectory that induces strong vibrations may degrade the quality of camera images. This arises because actuators interact with the physical world, thereby affecting what sensors perceive. Such feedback mechanisms make the blocks interdependent. The key idea is therefore to carefully design trajectories or control inputs in order to improve estimation performance. This is the essence of active sensing.

In addition to this interdependence, robotic systems must cope with pervasive uncertainties. These arise primarily from the dynamics model, the sensors, and the environment. The robot's model is rarely perfectly known: several parameters, such as the inertia matrix or friction coefficients, are uncertain or difficult to identify. Similarly, sensors used for perception and localization provide noisy and partial information, leading to an imperfect knowledge of both the robot's state and its environment. Finally, the environment itself is partially unknown and non-

stationary: it may evolve over time, other agents may appear and cause collisions, and some areas may remain occluded (e.g., behind walls). Typical examples [Lauri, 2023] include warehouse robots that must determine their location while navigating to collect goods, mining machines operating underground without GNSS, or outdoor robots that must adapt to changing weather, lighting conditions, and environmental occlusions. The combination of these factors makes localization and navigation particularly challenging in cluttered and dynamic environments. To address these difficulties, numerous probabilistic methods have been developed to provide robust estimation [Thrun, 2005]. Beyond the performance of estimation algorithms, one may ask whether there exist situations in which uncertainties are amplified and others in which they are mitigated.

To illustrate the concept of active sensing, let us consider two simple examples. In the first one, imagine that you attempt to localize yourself in Paris using only observations of well-known landmarks. Naturally, your position estimate will be far more accurate if you stand directly under the Eiffel Tower than if you are located in a nearby street since the position in Paris of Eiffel tower is perfectly known. The second example concerns the estimation of the Eiffel Tower's height.The goal is to measure the height of the Eiffel Tower using a sextant (a navigational tool used to measure relative angles) and a meter. From two measurements of the elevation angles and the relative position at which they are taken, one can estimate the tower's height. The result is, however, only an estimate, as the measurement tools are subject to various uncertainties. After taking the first measurement, the question arises: where should one move to take the second measurement in order to obtain the most accurate estimate of the height? Is it better to take the second measurement by moving closer to the Eiffel Tower, or by moving orthogonally to the axis of the first measurement? Which second measurement minimizes the uncertainty in the height estimate? Such a problem is typical of active sensing and is illustrated in Figure 1.2.

This perspective resonates with insights from human sensing. Gibson [Gibson, 1962] distinguished between situations in which sensors are used intentionally and those in which they act passively. For instance, there is a fundamental difference between touching an object and being touched by it. Although the same physical sensors are involved, the information obtained is richer when the action is deliberate. In robotics, this translates into the ability to make sensors active by orienting or positioning them to gather information where it matters most. The goal is to improve the quality and robustness of estimation, not by changing the estimation algorithm itself, but by optimizing how information is gathered. This is precisely the objective of active sensing.

Thus, active sensing seeks to address a set of fundamental questions: How can uncertainties be accounted for to improve overall system performance? How can the data acquisition process itself be optimized? How can the system be made more robust in the presence of uncertainty? How should uncertainties be incorporated into sensor placement strategies? And finally, how can one formulate an optimal control problem that explicitly integrates localization uncertainty?



Figure 1.2: Where to take the second elevation angle measurement to estimate the height of the Eiffel Tower with minimal uncertainty.

2 Motivation for the Work

Although the earliest studies on active sensing date back several decades, these works were typically tailored to specific applications of relatively simple applications. More recently, active sensing has experienced renewed interest, particularly through the development of perception-aware methods. However, these methods are either specific to a given application or require complex analytical derivations and are difficult to understand. One of the objectives of this thesis is to propose a general and accessible framework (both conceptually and practically) for active sensing. In particular, we exploit the covariance estimated by the Extended Kalman Filter to quantify sensor fusion performance. A second objective is to enrich gradient computation methods within the EKF, thereby improving the data acquisition process. This has direct applications not only in active sensing but also in measurement selection and sensor placement. Finally, we aim to demonstrate the versatility and efficiency of the proposed method across a broad range of applications, including sensor extrinsic calibration, IMU alignment for quadrotors, bearing-only target tracking, scalar field estimation, and measurement selection.

3 Publications

The work conducted in this thesis has led to several publications, each reflecting different methodological developments as well as the evolution of our active sensing framework and its

applications.

International journals:

- Jonas Benhamou et al. "Optimal Sensor Selection via Backpropagation Through the EKF". ***Major revision requested*** *from IEEE Transactions on Signal Processing* (2026)

International conferences:

- Jonas Benhamou et al. "Minimum-Covariance Active Sensing for Quadrotors". ***In preparation***
- Jonas Benhamou et al. "Backpropagation-Based Analytical Derivatives of EKF Covariance for Active Sensing". *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2024, pp. 12240–12247
- Jonas Benhamou et al. "Optimal Active Sensing Control for Two-Frame Systems". *2023 62nd IEEE Conference on Decision and Control (CDC)*. 2023, pp. 5967–5974

4 Structure of the Manuscript

In this thesis, we present the context, methodology, and contributions of our work on Extended Kalman Filter covariance minimization. The chapters are organized as follows:

- **Chapter 2** provides a brief historical perspective, situating active sensing within the broader theoretical landscape of planning under uncertainty. After explaining the challenges of solving the general POMDP framework, we show that active sensing is a tractable and relevant approach to reduce sensing uncertainties. Finally, we highlight its applicability across a wide range of problems.
- **Chapter 3** presents and organizes the contributions of the thesis, while placing them in the broader context of existing literature.
- **Chapter 4** extends the backpropagation equations within an Extended Kalman Filter (EKF). This method is essential for active sensing applications but also opens the door to other uses. The usefulness of backpropagation for active sensing is also described.
- **Chapter 5** shows how to implement active sensing for a robot with bicycle kinematics. After describing our planning problem based on analytical gradient backpropagation through an EKF, our method is applied to extrinsic sensor calibration. The goal is to overcome observability limitations and control noise propagation for the lever arm estimation. We demonstrate through simulations and real-world experiments that our active sensing trajectories reduce localization errors.
- **Chapter 6** introduces a generic active sensing framework for quadrotors. The efficiency of our method relies on flatness-based system theory, B-spline output parameterization, and gradient computation via backpropagation. The generic and and computational efficiency

of our approach is demonstrated through two applications: first, improving attitude estimation in an IMU-GNSS fusion; second, reducing collision risk between two drones using bearing-only measurements. The second application differs from the first because the controlled system and the estimated system are not the same. The relevance of the approach is validated in simulations, showing performance comparable to existing methods.

- **Chapter 7** extends the application of backpropagation equations in an EKF beyond active sensing. We formulate a general optimization problem for the optimal selection of measurements to minimize estimation uncertainty and for the positioning of a mobile sensor. This illustrates the potential of backpropagation equations and allows for a systematic and efficient tuning of EKF parameters (in a broad sense).

Chapter 2

Planning in Robotics Under Uncertainty: A Brief Historical Perspective

Contents

1 POMDPs: A General Framework for Decision-Making under Uncertainty . . . **8**
1.1 Brief Introduction to Markov Decision Process . . . 8
1.2 Some Remarks on POMDPs . . . 9
1.3 Low-Complexity Alternatives to POMDPs . . . 11
2 Active Sensing Background . . . **13**
2.1 Dual Control Theory . . . 13
2.2 Theory of Optimal Experimental Design . . . 15
3 Uncertainty-Aware Sensing . . . **18**
3.1 Active Sensing . . . 18
3.2 Perception-aware Planning . . . 22
3.3 Measurements Selection . . . 25
3.4 Optimal Sensor Placement . . . 25
4 Uncertainty-Aware Control . . . **27**
5 Conclusion . . . **29**

Résumé

Ce chapitre présente un panorama des principales approches pour la planification et le contrôle des systèmes en présence d'incertitudes. Il débute par la description du cadre général des POMDPs (Partially Observable Markov Decision Processes), qui permet de modéliser rigoureusement les incertitudes et d'aborder de manière systématique le compromis exploration-exploitation. Toutefois, la complexité calculatoire élevée des POMDPs limite leur applicabilité.

Pour pallier ces limites, plusieurs approximations peuvent être envisagées : le principe d'équivalence des incertitudes, qui ne considère que la moyenne des états, l'utilisation

de mesures d'information (telles que la matrice d'information de Fisher) plutôt que de densités de probabilité complètes, ou encore l'hypothèse gaussienne pour simplifier le traitement des incertitudes.

Deux stratégies moins calculatoire permettent d'en tenir compte : la réduction de l'incertitude par la perception active et la conception de contrôleurs robustes. La perception active planifie les actions afin d'améliorer l'estimation des états ou des paramètres. Elle s'appuie sur des mesures d'information héritées de la théorie du design expérimental optimal et sur l'idée d'exciter le système pour mieux l'observer issue du dual control. Plusieurs variantes sont abordées, notamment les approches perception-aware (spécifiques aux caméras) ainsi que la sélection et le placement optimal des capteurs. Le contrôle robuste, quant à lui, constitue un domaine vaste dont l'étude dépasse le cadre de ce manuscrit.

This chapter provides an overview of the main approaches used for planning and control of systems under uncertainty. Section 1 begins by presenting the general framework of Partially Observable Markov Decision Processes (POMDPs), which offer a rigorous way to model uncertainty. However, the high computational complexity of POMDPs often limits their practical applicability. The following sections introduce alternative approaches that reduce computational demands.

1 POMDPs: A General Framework for Decision-Making under Uncertainty

1.1 Brief Introduction to Markov Decision Process

A Markov Decision Process (MDP) provides a mathematical framework for modeling decision making in stochastic environments. Formally, a MDP is defined as a tuple $\langle \mathcal{S}, \mathcal{U}, f, R \rangle$, where $\mathcal{S}$ denotes the set of possible states, $\mathcal{U}$ the set of actions, $f(\boldsymbol{s'}|\boldsymbol{s}, \boldsymbol{u})$ the state transition probability function, $\gamma R(\boldsymbol{s}, \boldsymbol{u})$ the expected immediate reward received after executing action $\boldsymbol{u}$ in state $\boldsymbol{s}$ discounted by a factor $\gamma \in [0, 1]$ that encodes the preference for immediate versus future rewards. The Markov property assumes that the probability distribution over next states depends only on the current state and action, i.e., $p(\boldsymbol{s'}|\boldsymbol{s}_t, \boldsymbol{u}_t, \boldsymbol{s}_{t-1}, \boldsymbol{u}_{t-1}, \dots) = p(\boldsymbol{s'}|\boldsymbol{s}_t, \boldsymbol{u}_t)$.

At each discrete time step t, the current state $\boldsymbol{s}_t$ is observed, an action $\boldsymbol{u}_t$ is selected, a reward $R(\boldsymbol{s}_t, \boldsymbol{u}_t)$ is received, and the system transitions to a new state $\boldsymbol{s}_{t+1}$ according to f. A policy $\pi(\boldsymbol{u}|\boldsymbol{s})$ defines the system's behavior by specifying the probability of taking action $\boldsymbol{u}$ in state $\boldsymbol{s}$. The objective is to find an optimal policy π^* that maximizes the expected cumulative discounted return $G_t = \mathbb{E}_\pi \left[\sum_{k=t}^{\infty} \gamma^{k-t} R(\boldsymbol{s}_{t+1}, \boldsymbol{u}_{t+1}) \right]$.

The optimal value function $v^*(\boldsymbol{s})$ represents the maximal expected return achievable from state $\boldsymbol{s}$, and satisfies the Bellman optimality equation:

$$v^*(s) = \max_{\boldsymbol{u} \in \mathcal{U}} \left[R(\boldsymbol{s}, \boldsymbol{u}) + \gamma \sum_{\boldsymbol{s'} \in \mathcal{S}} f(\boldsymbol{s'}|\boldsymbol{s}, \boldsymbol{u}) v^*(\boldsymbol{s'}) \right]. \tag{2.1}$$

Similarly, the optimal action-value function $Q^*(\boldsymbol{s}, \boldsymbol{u})$ encodes the expected return of executing $\boldsymbol{u}$ in $\boldsymbol{s}$ and then following π^*. Solving a MDP thus involves estimating these value functions, either exactly via dynamic programming or approximately via reinforcement learning methods when $\mathcal{S}$ or $\mathcal{U}$ are large or continuous.

In conclusion, MDPs constitute the discrete-time counterpart of the general optimal stochastic control problem.

1.2 Some Remarks on POMDPs

One of the key assumptions of MDPs is that, at each time step, the system state is perfectly observed. This means it is known with accuracy and fully measured. When this assumption no longer holds, the problem is referred to as a Partially Observable Markov Decision Process (POMDP). The earliest work on this topic, by Karl Johan Åström, dates back to 1965. This section shows that partial observability adds an additional layer of complexity.

Similar to MDP, a Partially Observable Markov Decision Process is defined as a tuple $\langle \mathcal{S}, \mathcal{U}, \mathcal{Y}, f, h, R \rangle$. Compared to an MDP, the definition of a POMDP additionally includes the set of observations $\mathcal{Y}$ and an observation model h. h is defined as $h(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{u})$, which denotes the probability of receiving observation $\boldsymbol{y}$ after taking action $\boldsymbol{u}$ in state $\boldsymbol{s}$. Another key difference is the belief state at time t, $b_t \in \mathcal{B}$, which represents the posterior distribution over states conditioned on the initial belief b_0 and the sequence of past actions and observations, i.e., $b_t = p\left(\boldsymbol{s} \mid b_0, \boldsymbol{u}_{0:t-1}, \boldsymbol{y}_{1:t}\right)$.

As in MDPs, the goal is to find the optimal policy of a POMDP over a finite (or potentially infinite) horizon T, defined as a mapping from the belief space to the action space at each time step $t \in [0, T]$: $\pi^* : \mathcal{B} \to \mathcal{U}$. This optimal policy solves the following optimization problem:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r\left(b_t, \pi(b_t)\right)\right], \tag{2.2}$$

where $\gamma \in [0, 1]$ is the discount factor for future rewards, and $r(b, \boldsymbol{u}) = \sum_{\boldsymbol{s} \in \mathcal{S}} R(\boldsymbol{s}, \boldsymbol{u}) b(s)\, ds$ is the expected belief reward, representing the expected reward of taking action $\boldsymbol{u}$ at belief state b.

One important aspect of this formulation is the belief update after observing $\boldsymbol{y}_{t+1}$. The update is performed using a Bayesian filter of the form:

$$b_{t+1}(\boldsymbol{s'}) = \frac{p(\boldsymbol{y}_{t+1} \mid \boldsymbol{s'}, \boldsymbol{u}) \sum_{\boldsymbol{s} \in \mathcal{S}} p(\boldsymbol{s'} \mid \boldsymbol{u}, \boldsymbol{s}) b_t(\boldsymbol{s})}{\eta}, \tag{2.3}$$

where η is the normalizing factor ensuring that the belief remains a probability distribution.

Now that the notation has been introduced, several remarks can be made about this formulation. The first remark is that for exact resolution of POMDP the state, observation, and action spaces have to be discrete. Classical methods must therefore be adapted for continuous problems which significantly increases computational cost.

The second remark concerns the belief itself, which is a finite-dimensional vector without particular assumptions on its form. This makes the POMDP framework applicable to a wide

range of problems. However, belief propagation over time is computationally expensive, as it requires performing Bayesian filtering at every time step.

In addition to handling sensing uncertainties, one of the key advantages of POMDPs is their ability to systematically address the exploration-exploitation trade-off. The exploration-exploitation trade-off in reinforcement learning (RL) refers to the balance between exploiting known information to maximize immediate rewards and exploring to gain knowledge for future decisions. It is fundamental in sequential decision-making under uncertainty, since excessive exploitation risks missing better strategies, while excessive exploration delays convergence to optimal behavior. A central challenge lies in deciding when to sacrifice short-term rewards for long-term information gain, especially when the value of information is uncertain. Uncertainty in the environment's state and dynamics complicates action selection, requiring reasoning beyond immediate reward expectations. POMDPs provide a principled solution by explicitly modeling uncertainty and partial observability. In a POMDP, the agent maintains a belief state that captures both knowledge and uncertainty. This enables policies that jointly account for immediate rewards and the value of reducing uncertainty over time. Through belief-space planning, POMDPs support non-myopic strategies that adapt exploration levels dynamically.

In a survey by Lauri *et al.* [Lauri, 2023] on POMDPs and their applications in robotics, solution methods are divided into three categories. The first includes offline algorithms, which solve the problem exactly either by using dynamic programming or by exploiting specific forms of the value function, as in point-based algorithms. Although exact solutions guarantee optimality, they suffer from exponential growth in computational complexity with respect to the planning horizon and the size of the state, action, and observation spaces.

The second category includes online algorithms. Online algorithms iteratively compute and execute actions by selecting an optimal or near-optimal action for the current belief b_t, updating the belief with new observations, and repeating the cycle. By focusing only on the current belief, they can handle larger problems than offline methods. They typically represent all or part of the possible future scenarios as a search tree, where forward search and sampling-based methods are used to improve scalability in tree construction and exploration. However, it should be noted that the size of the tree grows rapidly with the dimensionality of the problem.

The third category consists of heuristic methods, which drastically reduce the computational cost at the expense of having no guarantees on policy optimality. In general, solving POMDPs requires a trade-off between computational resources and policy optimality.

In their survey, the authors emphasize the necessity of having a good transition and observation model from the outset to ensure that the resulting policy is meaningful. In practice, domain knowledge is often used to model the problem, but if this is not possible, an initial model-learning step is required.

Finally, they conclude the review with three main challenges for POMDPs:

- Model uncertainty: obtaining an accurate uncertainty model is essential; otherwise, the resulting policies will not be relevant.

- Continuous state, action, observation, and time: adapting methods to operate in continu-

ous spaces.

- Safe POMDPs in robotics: incorporating hard constraints to guarantee that certain states are never reached.

In addition to these challenges, in many cases the "right" reward to use is not known, and assigning a value to a given situation is difficult. Although this issue is not unique to POMDPs, it becomes more complex in this context, since in some cases it is necessary to assign a reward to information gain itself.

Despite these difficulties, the POMDP framework is used in many applications. A natural application is active Simultaneous Localization and Mapping (SLAM). A recent review dedicated to this topic [Placed, 2023] shows how the problem naturally fits into this framework and explains why a direct or naive solution is impossible. They present numerous works that use approximations to address the problem, as well as alternative approaches to active SLAM. Another application of POMDPs was proposed in [Bai, 2012] for an efficient collision avoidance system for drones. The approach models the problem as a continuous-state POMDP and applies Monte Carlo Value Iteration to avoid discretization. In simulation, they demonstrate a low collision risk with their method. Another example of navigation in an unknown environment, proposed in [Kim, 2021], uses a hierarchical POMDP that decomposes the problem into smaller subproblems (with reduced dimensionality). A first policy selects waypoints, and a second connects them. This approach greatly simplifies the problem, but no global guarantees are available for such an algorithm. We can also mention [Eidenberger, 2010], which presents a POMDP-based method for estimating 6D object poses in cluttered scenes. In particular, they plan sequential sensing actions that reduce camera occlusions.

Another idea to directly account for belief variability is to add a term that minimizes belief dispersion, thereby reducing overall uncertainty. For example, [Kim, 2021] addresses exploration and coverage of an unknown environment. In the reward used for the waypoint selection policy, they include a term that reduces the entropy of the belief. This encourages exploration of unknown areas and improves coverage.

In summary, POMDPs are an important tool for planning and control under uncertainty. Despite numerous applications, they remain computationally expensive. Compared to MDPs, the use of the belief is both a strength and a weakness. It is a strength because it enables intelligent handling of uncertainties. It is also a weakness because it increases algorithmic complexity due to the dimensionality of the belief and the cost of its propagation. A possible solution is to avoid relying on the full belief and instead use a sufficient information measure that can be propagated more efficiently.

1.3 Low-Complexity Alternatives to POMDPs

Once the POMDP framework has been established and its difficulties highlighted, we can examine alternatives for handling uncertainty. Many approaches in the field consist of cleverly formulating the problem or parameterizing certain quantities to simplify it. A dedicated chapter

on POMDP simplifications can be found in [Thrun, 2005]. Here, we focus on systematic simplifications that allow us to remain aware of uncertainties without relying on the full POMDP framework. In particular, we ask: what approximations or assumptions can be made to reduce the complexity of the problem?

The first simplification comes from dual control theory (Section 2.1), which introduces the certainty equivalence principle. Instead of using a cost function that considers the entire distribution of the state, this principle simplifies the problem by focusing only on the mean (of the transition dynamics). In other words, the control objective is to regulate the mean of the distribution. This is similar to the principle of separation of estimation and control [Tse, 1973]. This principle states that, for a linear system, it is not necessary to account for the impact of estimation errors when designing the control law. One can design an observer to estimate the system state and a controller to compute the control input from the estimated state separately. If both the observer and controller are stable, then the overall system is also stable. Although no such guarantees exist for nonlinear systems, separating the two problems still provides significant simplification, albeit at the cost of losing stability guarantees.

A second simplification suggests not considering the full belief but only an information measure. In particular, the theory of optimal experimental design focuses on defining relevant information metrics. These are often matrices, and optimal experimental design studies how these measures can be reduced to scalar quantities.

Taken together, these two simplifications suggest considering only the mean of the belief along with an associated measure of information or uncertainty. In addition to these simplifications, more general approaches can be applied, such as linearizing the transition and observation functions when they are nonlinear, which brings the problem back to the linear case, as well as assuming Gaussian white noise for disturbances.

Assuming that only the mean and an information measure are considered, two possibilities arise. Either one tries to reduce uncertainties, or one designs the controller to be robust to them. It is also possible to use both approaches simultaneously. One effective way to reduce uncertainty is to employ active sensing methods, which broadly encompass subfields such as perception-aware, optimal sensor placement, and measurement selection. All these approaches, which rely on reducing uncertainty during the sensing process, are grouped under the concept of uncertainty-aware sensing. On the other hand, robust control focuses on incorporating uncertainty directly into the controller, allowing it to compensate for imprecise localization. These approaches, which explicitly account for uncertainty in the controller design process, are grouped under the term uncertainty-aware control.Figure 2.1 illustrates how these domains are organized relative to each other. An arrow from one domain to another indicates that the theoretical scope of the child domain is included within the parent domain. However, this does not imply any chronological relationship between the domains.

In the following sections, we will first review the theoretical background used for active sensing in Section 2. Then, we will examine in detail the field of active sensing and subfields that emerge from it in Section 3. Finally, we provide a brief overview of robust control in Section 4.

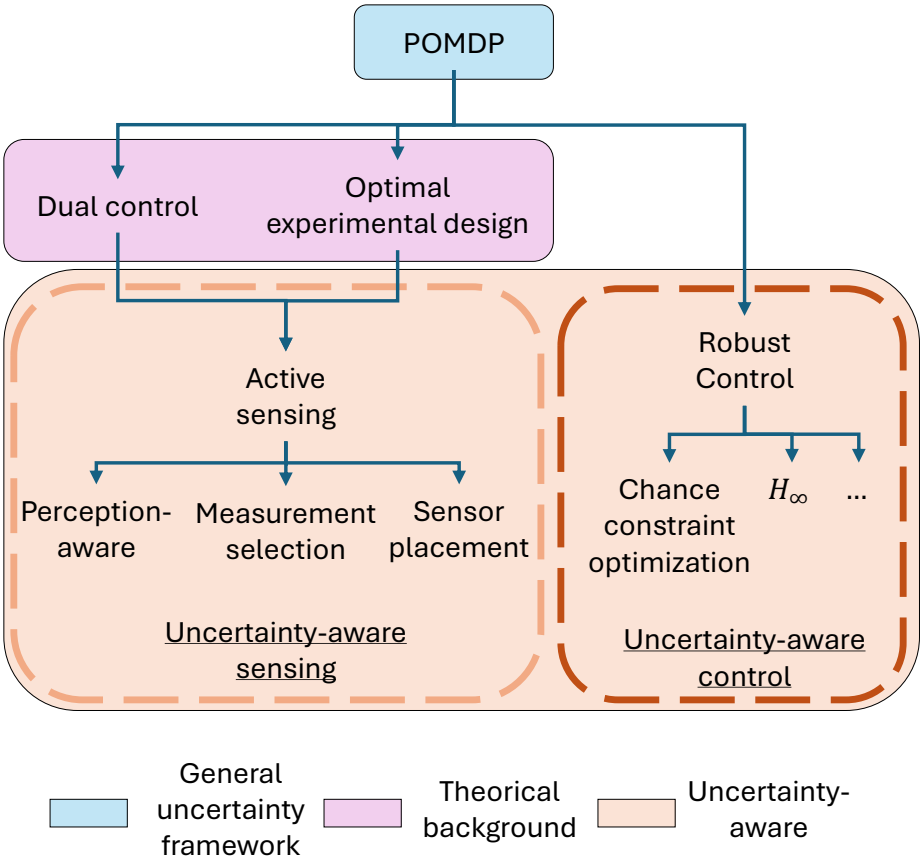

Figure 2.1: This diagram shows how different approaches to handling uncertainty in robotics are organized. It progresses from the most general theory to more specific ones.

2 Active Sensing Background

2.1 Dual Control Theory

Dual control theory is a branch of control theory that deals with the control of systems whose characteristics are initially unknown. For example, the transition function could be unknown. It is called dual because in controlling such a system the controller's objectives are twofold: first to control the system as well as possible based on current system knowledge. Secondly to experiment with the system so as to learn about its behavior and control it better in the future.

The term dual control was introduced in the 1960s by the article of Feldbâum [Feldbâum, 1963] to describe the trade-off between short term control objectives and actions to promote learning. A closely related term is the exploration-exploitation trade-off in reinforcement learning. In the article of Feldbâum, the dual control problem is expressed using an augmented state with parameters to estimate and using dynamic programming. He showed analytical solutions in very simple cases.

However, as shown in the review of dual control methods [Wittenmark, 1995], solving such a problem in the general case is intractable. Two main reasons explain this difficulty. The first is the curse of dimensionality, which arises from including the parameters to be estimated

in the system state. The second, deeper reason is that the inference is inherently nonlinear [Klenske, 2016]. For example, if we assume that the dynamical system is linear with unknown matrices, the dynamics of the augmented state will be nonlinear. Therefore, it is necessary to use approximations. A common simplification is the certainty equivalence principle, where the controller relies only on the mean of the estimated matrices and does not actively reduce uncertainty. Such approaches are not considered dual control, since no compromise is made between estimation and control. Moreover, the estimates produced by these controllers do not always converge. To efficiently approximate dual control Wittenmark *et al.* summarized the following methods:

- Add a random perturbation to the control signal. The challenge is to determine which perturbation to apply and when to stop it.
- Constraining the variance of parameter estimates one step ahead.
- Add covariance into the loss function and use serial expansion of the loss.
- Restricting to finite parameter sets, which greatly simplifies computation.

Regarding the series expansion method of the loss proposed by Tse *et al.* [Tse, 1973], the optimized loss function is defined as a sum of a control cost and an uncertainty cost. To simplify the computation of the uncertainty term, the authors assume Gaussian noise and employ a Kalman filter. Moreover, this method can be interpreted as an approximation of a POMDP, as argued in [Klenske, 2016].

As discussed in [Hewing, 2020], these approaches can be divided into two main categories. The first includes methods that do not explicitly solve the exploration-exploitation trade-off but instead introduce exploration through heuristic mechanisms, such as random perturbations. The second category approximates the loss function and explicitly accounts for the exploration-exploitation balance, as in the series expansion method. Despite these differences, both categories share a common objective: introducing excitation into the control strategy to improve information gathering.

Finally, Wittenmark *et al.* emphasize an important point concerning the incompatibility between robust control and dual control. Indeed, a cautious controller tends to reduce control effort when its effect is uncertain, which can ultimately lead to the system turning off or stagnating.

Among recent applications of dual control, an important line of work is dual Model Predictive Control (MPC), which explicitly accounts for information gain in the control strategy. A comprehensive review is presented in [Mesbah, 2018], including several industrial applications such as wood cutting with time-varying material properties. The authors conclude that implicit methods are generally more practical, although they provide weaker guarantees compared to explicit methods, which are often too complex to implement in practice. Other works exploit dual control to enhance the integration of MPC with learning approaches, as reviewed in [Hewing, 2020]. Dual control has also been applied to the estimation of other agents intentions in the environment, when these cannot be predicted in advance [Hu, 2024]. New works have

extended dual control to nonlinear systems parameterized by neural networks or Gaussian processes [Klenske, 2016]. Finally, recent research in this field have focused on deriving an upper bound on the regret. The regret is defined as the difference between the achieved cost function and the optimal cost that would have been obtained if the system dynamics were perfectly known (i.e., with an oracle). In the case of an LQR, such a bound has been established in [Mania, 2019]. Other studies aim to refine this regret bound or to demonstrate that certain dual control algorithms can asymptotically achieve it.

In summary, improved sensing comes at the cost of higher energy consumption and more aggressive maneuvers. For most systems it is obvious that more excitation leads to better estimation, but also to worse control performance. However, in situations with a short time horizon and poor initial estimates, dual control becomes a particularly important tool.

2.2 Theory of Optimal Experimental Design

The theory of Optimal Experimental Design (OED) is a well-established research field. Its exact origins are debated in the literature. For example, one of the earliest surveys on the subject by Mehra [Mehra, 1974] cites the work of Levin [Levin, 1960] as the first systematic contribution to the field. In another review by Karlin *et al.* [Karlin, 1966], the pioneering work is instead attributed to Kiefer [Kiefer, 1959]. In short, the theory appears to have emerged in the 1960s from several independent contributions.

As its name suggests, the objective is to plan a sequence of controls (i.e., an experiment) that will be used to gather measurements and then estimate (i.e., identify) a set of system parameters. The central question is how to design these experiments to obtain the most accurate possible estimates? The field focuses in particular on parameters that do not evolve over time.

The motivations for such an approach are numerous. Conducting experiments requires significant financial and time resources, so the goal is to minimize this process without degrading the quality of the estimates. Moreover, parameter identification may require a certain level of excitation. OED ensures a sufficient level of excitation to make parameter identification possible. Finally, OED is also used to design experiments that enable statistical tests with high sensitivity.

Let us consider an example of OED implementation presented in Mehra [Mehra, 1974]. The objective is to estimate a parameter vector $\boldsymbol{\theta}$ that appears in the system dynamics. The first step is to define a measure of estimation quality for $\boldsymbol{\theta}$, namely the Fisher Information Matrix (FIM). A D-optimal criterion (log-det) is then used to transform the matrix into a scalar measure, as this design is invariant under scaling. Once the optimization problem is formulated, it is shown that a random control sequence provides a good estimation, or alternatively, an analytical solution can be derived. Finally, the study demonstrates that the problem can be treated in both the time and frequency domains.

In general, the procedure can be summarized in three steps: the first step defines a quality measure, the second selects an optimal design criterion, and the third solves the optimization problem. This yields the control sequence to be applied $\boldsymbol{u}$. From the resulting observations $\boldsymbol{y}$, the parameters θ can be identified. Research in this field focuses either on the formulation of

the problem or on the methods for solving it. To briefly present the different elements of OED theory, we rely on the survey by Huan *et al.* [Huan, 2024]

Let us begin by examining optimal design criteria for OED mentioned in the article:

- The first criterion is the Fisher Information Matrix (FIM). The goal of OED is to find the experiment that maximizes the FIM, defined as:

$$\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}, \boldsymbol{u}) \coloneqq \mathbb{E}_{\boldsymbol{y}} \left[\nabla_{\boldsymbol{\theta}} \log \left(p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{u}) \right)^{\top} \nabla_{\boldsymbol{\theta}} \log(p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{u})) \right]. \tag{2.4}$$

  The use of this criterion is widespread. However, a difficulty lies in the choice of $\boldsymbol{\theta}$ around which it is evaluated. A common simplification is to compute it around the current best estimate.

- The second category of methods consists of maximizing the expected utility. In general form, the criterion can be written as:

$$\boldsymbol{\mathcal{U}}(\boldsymbol{u}) = \mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{y}|\boldsymbol{u}} \left[u(\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{\theta}) \right], \tag{2.5}$$

  with $u(\cdot)$ a utility function. This generic formulation allows many metrics to be expressed. For example, the expected information gain in the parameters, which quantifies the difference between the entropy of the prior and the entropy of the posterior. This loss is equivalent to mutual information:

$$\boldsymbol{\mathcal{U}}_{\mathbf{KL}}(\boldsymbol{u}) \coloneqq \mathbb{E}_{\boldsymbol{y}|\boldsymbol{u}} \left[\mathrm{KL} \left(p(\boldsymbol{\theta}|\boldsymbol{u}, \boldsymbol{y}) \,\|\, p(\boldsymbol{\theta}) \right) \right], \tag{2.6}$$

  where KL is the Kullback-Leibler divergence, defined as: $\mathrm{KL}(p, q) \coloneqq \int_{x \in \chi} p(x) \log \left(\frac{p(x)}{q(x)} \right)$. Another criterion is the trace of the covariance of the estimate:

$$\boldsymbol{\mathcal{U}}_{\boldsymbol{Cov}}(\boldsymbol{u}) \coloneqq - \int \int \left(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{u}) \right)^{\top} \left(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{u}) \right) p(\boldsymbol{y}, \boldsymbol{\theta}|\boldsymbol{u}) \, d\boldsymbol{\theta} \, d\boldsymbol{y}, \tag{2.7}$$

  with $\bar{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{u}) = \mathbb{E}[\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{u}]$ the optimal estimate of $\boldsymbol{\theta}$. More generally, any function of the covariance can be used.

However, these measures present several challenges. They often require the computation of expensive expectations (a common workaround is to use simple distributions, but even though the problem remains complex). Another difficulty lies in constructing the prior distribution on $\boldsymbol{\theta}$.

Another difficulty arises when the parameter to be estimated is multi-dimensional. In this case, information criteria take the form of matrices, whereas optimization requires a scalar criterion. To overcome this, OED theory has introduced a broad family of alphabetic optimality criteria. The most common include:

- A-optimal: corresponds to maximizing the trace of the information measure (L-optimal when weights are added to the matrix). This criterion is analytically simple, allowing

closed-form derivations. It can also be interpreted as an average criterion since it equals the sum of the eigenvalues.

- D-optimal: corresponds to maximizing the log-determinant of the information matrix. This measure is invariant under rescaling ($\log|\boldsymbol{S}\boldsymbol{\mathcal{I}}| = \log|\boldsymbol{\mathcal{I}}| + cte$). However, deriving analytical expressions under this design is more challenging.

- E-optimal: corresponds to maximizing the smallest eigenvalue of the information measure. By optimizing this metric, we aim to ensure that the information matrix has full rank. Nevertheless, this measure is difficult to handle analytically and is not differentiable numerically (except in dimensions ≤ 2).

Beyond these, many other decision criteria exist. The choice generally depends on the overall objective of the problem. For instance, if one is particularly interested in accurately identifying a specific variable, the optimization can directly target the information related to it. In [Mehra, 1974], a more general design is proposed of the form $[\mathrm{Tr}(\boldsymbol{\mathcal{I}}^s)]^{1/s}$, with s a parameter that tunes the behavior of the criterion. For example, as $s \to +\infty$, this expression converges to the largest eigenvalue of the information matrix $\boldsymbol{\mathcal{I}}$. Moreover, this formulation is differentiable and can be used as an approximation of the smallest or largest eigenvalue [Salaris, 2019].

The multi-dimensional case introduces another difficulty. The information matrix contains variables with different units and magnitudes, which can cause issues when combined with the optimal design criteria described above. A simple approach is to use only a subset of the information matrix corresponding to homogeneous variables. Alternatively, renormalization techniques for the information matrix, such as the one proposed in [Preiss, 2018], can be applied.

After defining the optimization problem, the solution relies on numerical optimization methods. For OED, the main approaches include:

- Local search and exchange algorithms: starting from an initial experimental design, permutations are tested to find those that most improve the criterion. These methods are simple to implement but converge to local extremum.

- Continuous convex relaxation approaches: particularly useful for measurement selection (see Section 3.3).

- Sequential algorithms: candidates are added successively, which is especially effective for submodular losses.

- Combinatorial approaches for discrete design variables.

- Optimization with derivative-free methods: such as Bayesian optimization or stochastic approximation methods based on finite differences.

- Gradient-based methods: applicable when a model is available that links the experiment to its impact on the information measure.

Huan *et al.* conclude with a discussion on using POMDPs for sequential OED, i.e., updating the experimental plan as new information is acquired. This is equivalent to working with POMDPs, along with the computational challenges they entail. They suggest that a more suitable approach in this context is to represent the belief through an information matrix rather than maintaining the full belief. This requires an update equation for the information measure. A key drawback is that its size grows quadratically with the dimension of $\boldsymbol{\theta}$.

As mentioned, POMDPs form a more general framework than OED. For instance, [Bai, 2013] applies POMDPs to an OED problem. They present two case studies illustrating how planning improves parameter identification in dynamic models. One example is the Acrobot swing-up, where the goal is to swing a two-link pendulum from the downward to the upright position using only elbow torque, with the mass of the second link initially unknown. The resulting policy achieves performance close to that of an oracle with exact knowledge of the parameter. Among recent works applying OED, one can cite [Daş, 2025], which focuses on planning calibration motions for a manipulator. The information gain of each motion is modeled by a Gaussian process that is updated during calibration.

Beyond robotics, OED is widely used in dynamical systems theory whenever one seeks to identify model parameters with a limited number of experiments. Its range of applications spans diverse domains, including biochemical networks, chemical kinetics, and heat transfer. A comprehensive overview of such applications can be found in [Franceschini, 2008].

In summary, the theory of optimal experimental design defines metrics that quantify the quality of parameter estimation resulting from an experiment. Since these metrics are typically matrices, OED pays special attention to scalarizing them for optimization. These two aspects will appear repeatedly throughout this manuscript.

3 Uncertainty-Aware Sensing

In this section, we present the various domains grouped under the term uncertainty-aware sensing in Figure 2.1. We begin with a detailed analysis of active sensing methods in Subsection 3.1, followed by its subfields: perception-aware planning in Subsection 3.2, measurement selection in Subsection 3.3, and finally optimal sensor placement in Subsection 3.4.

So far, we have introduced two domains, dual control and optimal experimental design, which primarily focus on designing optimal control problems that improve the identification or calibration of system parameters. In contrast, uncertainty-aware sensing methods aim to enhance the estimation of system states that evolve over time. The dynamic and online nature of this estimation problem adds an additional layer of complexity to its resolution.

3.1 Active Sensing

Before delving into the core of active sensing, it is common to distinguish between active and passive sensors. Active sensors require emitting a signal in order to make a measurement. For example, a LiDAR emits an electromagnetic wave to measure distance. In contrast, passive sensors perform measurements without emitting any signal, such as an IMU. In some contexts,

the emissive nature of active sensors can be problematic, either due to energy consumption or the need for stealth. However, this distinction is not relevant for active sensing: one can perform active sensing with both active and passive sensors, as the key idea lies not in signal emission but in selecting actions or configurations to improve state estimation.

In the literature, several definitions of active sensing can be found. For instance, [Bajcsy, 1988] defines it as: "Active Sensing can be stated as a problem of controlling strategies applied to the data acquisition process which will depend on the current state of the data interpretation and the goal or the task of the process." Similarly, [Mihaylova, 2003] states: "Active sensing in robotics incorporates the following aspects: (i) where to position sensors, and (ii) how to make decisions for next actions, in order to maximize information gain and minimize costs." These two definitions clearly highlight that active sensing lies at the intersection of dual control and optimal experimental design: it involves both accomplishing a task and optimizing an information measure.

To discuss the methodology of active sensing, we will first present the insights of problem formulations in Section 3.1.1. Then, we will discuss the simplifications that make the problem computationally tractable in Section 3.1.2, and finally, we will present some applications in Section 3.1.3.

3.1.1 Problem Formulation

As previously discussed in Section 1 on POMDPs, [Bajcsy, 1988] emphasizes that effective active sensing requires accurate models of both the sensor and system dynamics. For each, it is necessary to specify not only the physical model but also the noise model. In discrete time, the dynamics and observation models with Gaussian white noise can be written as:

$$\begin{cases} \boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n, \boldsymbol{w}_n)\,, \text{ where } \boldsymbol{w}_n \sim \mathcal{N}(\mathbf{0}, Q_n)\,, \\ \boldsymbol{y}_n = h(\boldsymbol{x}_n) + \boldsymbol{\nu}_n\,, \text{ where } \boldsymbol{\nu}_n \sim \mathcal{N}(\mathbf{0}, R_n)\,. \end{cases} \tag{2.8}$$

After specifying the models, as in OED, a criterion must be defined to quantify the effectiveness of sensing. These criteria can be grouped into three main categories:

- OED-based measures (see Section 2.2). These include the Fisher Information Matrix (FIM) [Zhao, 2013; Xu, 2023; Li, 2023] as well as the estimation covariance obtained from an Extended Kalman Filter [Leung, 2006; He, 2008; Patil, 2015; Hausman, 2015; Cognetti, 2018].

- Observability-based measures. In systems theory, a system is said to be observable if its initial state can be estimated from successive partial measurements. While observability is a binary property, several metrics have been proposed to quantify how easily a system is observable. The most widely used is the Observability Gramian (OG) [Hinson, 2013; Pagano, 2025]. For a fixed horizon N, it is defined as:

$$W_N \coloneqq \sum_{n=1}^{N} \Phi_n^\top H_n^\top H_n \Phi_n, \tag{2.9}$$

where $\boldsymbol{H_n} = \frac{\partial h(\boldsymbol{x}_n)}{\partial \boldsymbol{x}}$, $\boldsymbol{\Phi}_n = \boldsymbol{F}_n \boldsymbol{\Phi}_{n-1}$, $\boldsymbol{\Phi}_{-1} = \boldsymbol{I}_d$, and $\boldsymbol{F}_{n+1} = \frac{\partial f(\boldsymbol{x}_n, \boldsymbol{u}_{n+1}, \boldsymbol{0})}{\partial \boldsymbol{x}}$. Other observability-based measures include the Empirical Local Observability Gramian (ELOG) introduced by Krener and Ide in [Krener, 2009] and its extensions [Preiss, 2018]. Finally, constructibility-based measures [Napolitano, 2024; Napolitano, 2021; Salaris, 2019] focus on reconstructing the final state from partial measurements.

- Task-specific measures. This category includes criteria that may not be theoretically grounded but are designed to intuitively improve estimation performance. Examples include trajectory-sensitivity measures computed as a gradient [Brault, 2021; Wasiela, 2023], losses that penalize occlusions [Murali, 2019; Xing, 2023], and strategies favoring flight over textured areas to enhance visual-inertial odometry [Bartolomei, 2020].

Although seemingly different, OED-based measures and observability-based measures are closely related. For example, the FIM of the observations evaluated at the initial state coincides with the observability Gramian when the observation noise covariance is the identity. Similarly, by turning off process noise, the inverse of the EKF covariance nearly matches the constructibility Gramian of [Salaris, 2019]. However, information-based criteria neglect the effect of process noise on estimation, which in some cases can reduce their effectiveness, as demonstrated in the simulations of [Rafieisakhaei, 2017].

Once the information criterion has been defined and the problem constraints specified (e.g., energy limitations or actuator bounds, etc.), the optimization problem can then be solved.

3.1.2 Computational Simplifications

To address the computational challenges associated with POMDPs (Section 2.2), active sensing approaches often introduce several simplifications. One such simplification is the use of a multi-objective loss function of the form:

$$\mathcal{L}(\boldsymbol{u}) = \mathcal{L}_c(\boldsymbol{u}) + \lambda \mathcal{L}_a(\boldsymbol{u})\,. \tag{2.10}$$

The loss function $\mathcal{L}_c(\boldsymbol{u})$ captures the control objective for task execution in deterministic (non-stochastic) manner, whereas $\mathcal{L}_a(\boldsymbol{u})$ focuses on increasing the gained information (i.e., reducing uncertainty). The parameter λ is used to regulate the trade-off between these two objectives. This multi-objective formulation for active sensing is clearly discussed in [Bajcsy, 1988; Mihaylova, 2003]. In practice, many methods adopt this approach [Patil, 2015; Falanga, 2018; Murali, 2019; Bartolomei, 2020] (and so does our approach in the present thesis).

By taking a step back, the decomposition of the problem into two loss terms can be justified as follows. The control cost acts only on the mean and ignores the impact of uncertainty. This rational omission of uncertainty appears across many domains, often under different names. In dual control theory, it is known as a certainty equivalence principle; in active sensing methods, it is often referred to as the maximum likelihood hypothesis; and in classical control theory, it corresponds to the separation principle. However, for nonlinear systems, this assumption does not guarantee that the system remains stable or successfully completes the task. The idea is

therefore to add a second term that give a bonus to reduce uncertainty. When uncertainties are small, the certainty equivalence principle becomes "more valid". Ideally, the system optimizing both losses should succeed at the task more often than a system that considers only the control loss.

It is important to note that this approach is inherently suboptimal. For instance, in situations where the system is highly variable but still achieves good performance, there may be no need to take actions specifically to gain information. Yet an active sensing strategy will still attempt to reduce uncertainty, potentially resulting in unnecessary exploration.

Careful management of the trade-off between control and sensing objectives through λ tuning is essential, as these objectives are inherently contradictory. The active sensing loss seeks to excite the system [Willems, 2004] to make it more observable, while the control loss aims to minimize energy consumption and produce smooth trajectories as explained for dual control (see Section 2.1). An alternative to handle the trade-off is to minimize an information-based criterion under task completion constraints [Preiss, 2018; Salaris, 2019; Moon, 2022; Napolitano, 2024]. From a Lagrangian perspective, this corresponds to the previous method, except that the balance between control and sensing (i.e., the exploration-exploitation trade-off) is determined by the optimization itself. However, in terms of computational complexity, these approaches require more sophisticated solvers.

Another common simplification in active sensing concerns the degree of feedback in the sensing loop. [Yang, 2018] identifies three levels of active sensing. The first is a task-dependent open-loop plan, which relies solely on the task without considering observations. The second, a single-shot open-loop approach, uses only the initial observation to compute the full action sequence, without any subsequent replanning. The third, a closed-loop approach, replans the trajectory at each new observation. These three approaches are commonly found in the literature. The first approach is used, when optimal trajectories for a given task are known. For example, dynamic alignment [Porat, 1981] or drone pursuit with bearing-only measurements [Li, 2023]. Then, the required maneuvers can be directly implemented. Other methods compute an active sensing trajectory at the beginning of a mission and then update it at a low frequency [Bartolomei, 2020]. Finally, active sensing can be integrated directly at the controller level. For instance, [Falanga, 2018] proposes a Model Predictive Control (MPC) that incorporates an active sensing loss. Depending on the difficulty of the problem, the optimization may be solved more or fewer times per iteration in order to adapt to the computational resources available.

Once the underlying optimization problem is formulated, it can be solved either analytically in simple cases or numerically. Among the numerical solution methods, many are the same techniques used in planning and control algorithms. The key difference is that it is also necessary to account for the dynamics of the information measure, which can significantly increase the computational complexity. Commonly used approaches include discretizing the system dynamics and parameterizing the decision variables. For the actual optimization, one can distinguish between zero-order methods [He, 2008; Costante, 2017; Moon, 2022], which rely on sampling, and first-order methods, which exploit gradient information using solvers such as Sequential Quadratic Programming (SQP) or BFGS. These methods require computing the gradient of the

information-based loss, which is generally challenging and represents one of the contributions of the approach in [Salaris, 2019].

3.1.3 Active Sensing Applications

Active sensing has been applied to a wide range of robotic estimation and calibration problems. Here, we review a few representative examples. Many studies [Cognetti, 2018; Salaris, 2019; Napolitano, 2024] focus on localizing a unicycle using two known position landmarks, with the vehicle obtaining either range or bearing measurements relative to these landmarks. Extensions consider multiple landmarks and measurements that are only available within a limited range around each landmark [Patil, 2015; Napolitano, 2021]. This problem is a typical active sensing scenario, as observations are partial and measurement quality strongly depends on the vehicle's position. In another study, [Leung, 2006] addresses the problem of identifying landmarks in a scene using a distance sensor with a limited field of view and a range of a few meters. Active sensing has also been applied to localize the end-effector of a manipulator using stereo cameras [Patil, 2015] and to guide ground vehicles through information-rich regions [Platt, 2010]. Additionally, the active SLAM problem can be addressed using active sensing approaches, as proposed in [Kontitsis, 2013]. Another application is related to terrain correlation-based map localization. [Flayac, 2019] presented a method that optimizes the FIM, significantly improving the estimates obtained from a particle filter. Active sensing also applies to multi-robot systems. For example, [Hausman, 2015] optimizes the geometric configuration of a swarm in order to track a target. Beyond localization, active sensing has proven effective for calibration tasks, including the simultaneous calibration and localization of an inertial center and GNSS system (estimating biases and lever arms) and a camera-IMU system [Preiss, 2018], as well as the calibration of a 6-DOF manipulator [Sun, 2008]. A subset of these applications is illustrated in Figure 2.2. These examples illustrate the versatility of active sensing in improving estimation accuracy and overall system performance through motion planning.

In this section, we focus on general active sensing without specifying a particular goal or sensor. In the following sections, we will investigate more specific cases.

3.2 Perception-aware Planning

One of the subfields of active sensing is perception-aware planning, as illustrated in Figure 2.1. Before delving into the details of how these methods operate, we first explain why more specialized approaches than active sensing are required when the sensors involved are cameras.

Among the various sensors considered in active sensing, cameras stand out as particularly challenging and interesting. They are unidirectional sensors with a limited field of view, which implies that in addition to choosing the sensor's position, one must also determine its orientation. This additional consideration significantly complicates trajectory planning. Therefore, the field of view must be explicitly taken into account in the optimization. However, this limitation is not differentiable because an object is either in the field of view or it is not. Therefore, several works have proposed smoothed approximations to take into account the limited field of view

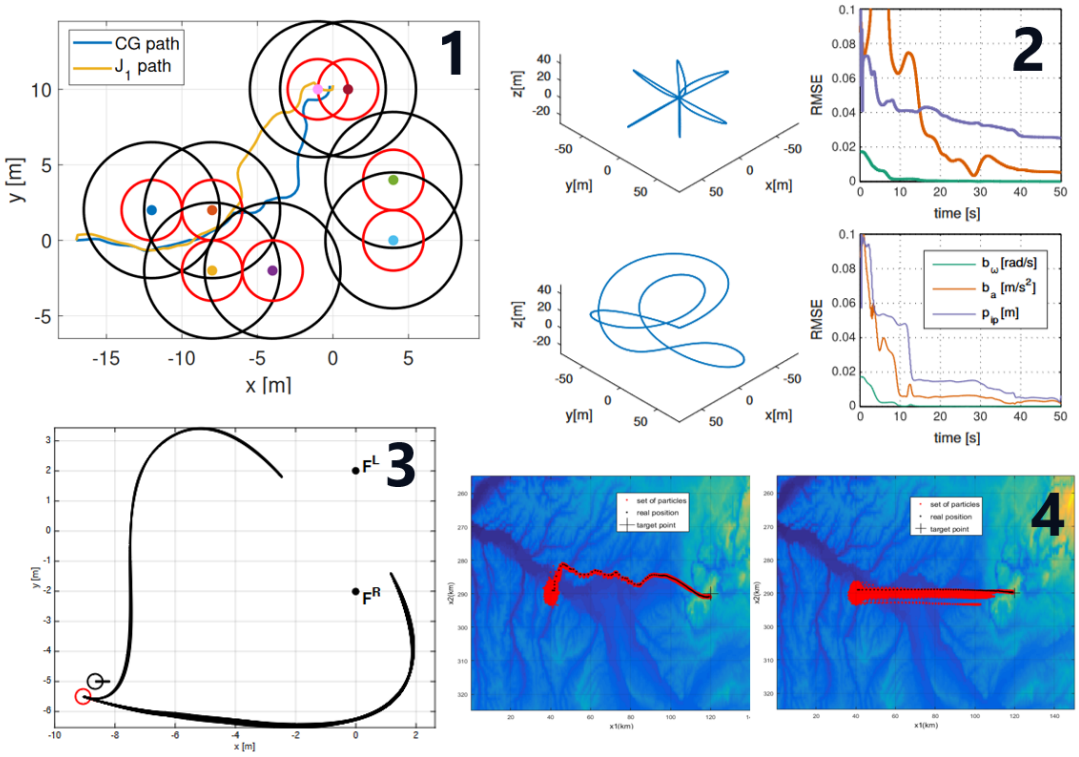

Figure 2.2: Illustrations of four active sensing applications.Figure 1, adapted from [Napolitano, 2021], shows localization improvement using landmarks with limited range. Figure 2, from [Preiss, 2018], addresses the calibration of an IMU-GNSS system. Figure 3, taken from [Salaris, 2019], focuses on landmark-based localization. Figure 4, from [Flayac, 2019], illustrates terrain correlation-based map localization.

[Murali, 2019; Bartolomei, 2020]. Similarly, occlusions or obstacles can render parts of the scene temporarily unobservable [Penin, 2018].

A second particularity of vision-based sensing lies in the nature of the measurements. Cameras provide bearing-only observations, especially when the scale factor is unknown. Such measurements are only partial observations of the system's state. Full state estimation relies on sensor fusion with a dynamical model. In this setting, the geometry of the planned trajectory critically affects localization performance, as certain trajectories can greatly improve or degrade the quality of estimation.

Finally, vision measurements are inherently coupled with the vehicle and environment interaction. In static scenes, if the vehicle is motionless, no optical flow is generated (this idea was presented by J. Gibson). Moreover, the quality of acquired images depends on the environment's texture. For example, flying over untextured surfaces leads to poor feature extraction, whereas textured areas provide rich information for visual odometry [Bartolomei, 2020]. These characteristics make camera-based systems archetypal cases for active sensing, leading to perception-aware fields.

Early applications of perception-aware were proposed in [Bajcsy, 1988]. Since then, much work has been done on the subject. For example, [Burgard, 1997] proposed an application in which the environment was assumed known and represented as an occupancy map. Localization

was achieved through occupancy measurements derived from a camera. To improve accuracy, the authors proposed a multi-objective trajectory planning scheme that balanced task execution with an entropy-based measure of localization uncertainty. This formulation also illustrates the dual nature of perception-aware: simultaneously achieving control objectives and reducing estimation uncertainty. Later works have refined this principle to incorporate more complex vision models. For instance, [Costante, 2017] considered a dense visual odometry in a known environment. Then, they quantified information gain through the FIM of the reprojection error, with planning performed using RRT* augmented with this objective. The method was subsequently adapted to unknown environments and validated experimentally on a micro aerial vehicle.

A complementary line of work has integrated active sensing directly within control formulations. [Falanga, 2018] introduced a perception-aware Model Predictive Control framework that solves in real time a receding-horizon problem. The approach balances trajectory tracking with the requirement of keeping predefined points of interest centered in the camera field of view while minimizing their projected velocity. This yields trajectories that are simultaneously task-feasible and perception-aware.

The limitations of feature-based localization and the restricted field of view in visual systems have also inspired research in this area. [Murali, 2019] introduced the notion of co-visibility, measuring how many visual features are jointly visible across successive keyframes. By leveraging differential flatness, they optimized the loss by only changing the yaw. Experiments demonstrated the impact of yaw design on the reference trajectory tracking task.Similarly, [Bartolomei, 2020] incorporated semantic loss to assess whether areas of the environment were textured. Their method encouraged drones to fly over textured zones, thereby improving the robustness of visual-inertial odometry. The perception-aware cost was combined with task objectives in a closed-loop optimization framework, yielding higher mission success rates.

Perception-aware has also been explored in pursuit and tracking problems. [Li, 2023] optimized the trajectory of a drone pursuing another drone using bearing-only measurements. Their method leveraged trajectory geometry to ensure that angular observations provided sufficient information for target state estimation.

Beyond localization, perception-aware planning has been extended to safety and obstacle avoidance. [Zhou, 2021] assumed that unexplored regions might conceal obstacles and proposed slowing down near such areas to ensure sufficient reaction time. In the same spirit, [Penin, 2018] explicitly accounted for occlusion risk during visual tracking, allowing more reliable execution. Other works have used perception costs to guide navigation, such as [Xing, 2023], where an additional MPC cost encouraged power line inspection while avoiding obstacles.

Altogether, these works demonstrate the rich set of challenges cameras are used for active sensing (limited field of view, bearing-only observations, and dependence on scene texture). At the same time, these works highlight the benefit of active sensing for vision (i.e., perception-aware) in the context of localization, calibration, and safety. More broadly, camera-based systems embody the essence of active sensing, where sensing and control are deeply coupled and must be optimized jointly.

3.3 Measurements Selection

As presented in Figure 2.1, another related but seemingly distinct subfield of active sensing is the measurement selection problem. In this type of problem, it is generally assumed that a finite number of sensors is available. The central question becomes: When to use the sensors or which measurements in a sequence are most informative?

Measurements selection frequently arises in contexts of highly constrained systems. For example, energy limitations motivate minimizing the sensing cost [Bian, 2006; Liu, 2016]; limited availability of sensors requires using them optimally [Moreno-Salinas, 2013]; while stealth constraints or limited communication bandwidth in sensor networks [Shamaiah, 2010]. Beyond these, sensor selection has also been studied in the context of sparse signal recovery, where the goal is to compress signals into sparse representations [Chepuri, 2015].

Conceptually, measurements selection problem is closely related to active sensing and optimal experimental design. The methodology is similar: one defines an information criterion to optimize with respect to a set of decision variables. The difference lies in the nature of these variables: while active sensing typically optimizes over control inputs, sensor selection optimizes temporal arrangements of sensors. This static nature often simplifies the problem, since constraints are easier to express and handle. A simplification often encountered is the absence of an explicit mission objective: the only goal is to gather information, subject to physical constraints. Thus, the exploration-exploitation trade-off does not arise. Despite these simplifications, the problem remains computationally hard. In fact, the optimal measurements selection problem has been shown to be NP-hard in many cases [Ko, 1995; Bian, 2006; Krause, 2008].

In the problem formulation, the information metrics used are largely the same as those in active sensing (see Section 3.1.1). However, the optimization methods differ due to the discrete and combinatorial nature of the problem. The main approaches include greedy algorithms [Yu, 1973], local search [Takahashi, 2023], mixed-integer programming [Krause, 2008], and convex relaxation [Joshi, 2009].

In addition to direct applications of measurement selection for signal compression [Chepuri, 2015] or energy consumption reduction [Bian, 2006], the concept has been applied across a wide range of domains. From theoretical studies of observability to practical implementations in robotics. Early work, such as [Yu, 1973], investigated linear distributed parameter systems, addressing whether a finite set of measurements could ensure observability and how to minimize the number of sensors required. This study introduced one of the first greedy algorithms for optimal measurement scheduling, based on covariance minimization and gradient descent. More recently, in the context of machine learning, [Krause, 2008] linked sensor selection to Gaussian process regression, exploiting the submodularity of mutual information to provide performance guarantees for greedy selection methods.

3.4 Optimal Sensor Placement

Another subfield closely related to active sensing and measurement selection is optimal sensor placement (see Figure 2.1). This field can be viewed as the spatial counterpart of measurement

selection, where the problem concerns the spatial arrangement of sensors. The central question becomes: where should sensors be placed to maximize the information acquired? Therefore, the decision variables are the sensor location parameters. Nonetheless, sensor placement can also involve mobile sensors that actively move to gather information, in which case the distinction with active sensing becomes more blurred.

The methods used to solve these problems are conceptually similar to those for measurement selection (Section 3.3). When the information criterion is differentiable with respect to the decision variables, gradient-based optimization can be applied [Vitus, 2011]. Extensions have even considered distributed implementations for mobile sensor control laws [Pagano, 2025]. This approach is particularly appealing since the decision variables are often continuous, unlike in discrete measurement selection problems. Moreover, analytical solutions for sensor placement exist in specific settings—typically when the geometry of the problem is simple or exhibits a particular structure [Zhang, 1992; Moreno-Salinas, 2013].

A separate class of methods focuses explicitly on the geometry of sensor configurations, such as maximizing spatial coverage of a region—often studied under the framework of coverage control. For instance, [Cortes, 2004] optimized the deployment of a swarm of agents to monitor a spatial density distribution, while [Duan, 2021] formulated surveillance as a graph-based Markov decision process. These approaches naturally link sensor placement to the broader field of coverage control.

Applications of optimal sensor placement are numerous. For example, [Zhang, 1992] derived optimal sensor orientation strategies to enhance position estimation with bearing only measurements, while [Moreno-Salinas, 2013] characterized the optimal geometric configuration of a 2D sensor network to maximize the information available for tracking multiple known targets. Later, [Vitus, 2011] explored a similar problem with mobile integrator dynamics, proposing a gradient-based method to optimize the placement of range sensors using smooth sigmoidal activation functions. More recently, [Kaveti, 2024] addressed camera placement for SLAM, leveraging the Fisher Information Matrix and its Schur complement to isolate vehicle position information. Their greedy algorithm iteratively added the most informative landmarks, and a convex relaxation was used to certify near-optimality. While promising, these results rely on prior assumptions about landmark availability, which limit their realism. In distributed robotics, [Pagano, 2025] proposed active placement strategies for vehicles tasked with locating a diffusive source, maximizing the observability Gramian while maintaining safe distances. In localization contexts, [Bo, 2019] examined surface vehicles equipped with GNSS, whose relative orientation measurements were exploited by underwater vehicles for self-localization.

Collectively, these studies demonstrate the versatility of sensor placement methods, spanning from geometric optimization to modern applications in robotics, perception, and large-scale sensor networks.

4 Uncertainty-Aware Control

In this section, we briefly discuss the methods grouped under the category uncertainty-aware control in Figure 2.1. This category includes approaches that aim to mitigate the effects of uncertainty, particularly through robust and stochastic control techniques. A detailed study of robust control methods is beyond the scope of this manuscript, as the literature is both vast and highly specialized. Nevertheless, we highlight three main reasons for addressing this topic here: first, robust control provides an alternative formulation to account for uncertainties, assuming they cannot be directly influenced and asking instead how to minimize their impact on the mission; second, some robust control approaches are fundamentally incompatible with active sensing; and third, certain robust control techniques share strong similarities with active sensing strategies. For the general introduction, we rely on Chapter 13 of [Tedrake, 2023].

There are two ways to address uncertainties in control. Robust control aims to provide worst-case performance guarantees, allowing one to work with bounded models of uncertainty. Conversely, stochastic control optimizes performance in the presence of uncertainty without providing worst-case outcome guarantees. Stochastic approaches are typically used when uncertainties are unbounded.

As in the previous sections, robust control begins by defining a performance criterion. Beyond the mission objective itself, the effect of uncertainty on this objective must also be considered. The main performance metrics presented in Chapter 13 of [Tedrake, 2023] are:

- Desired state distribution: the objective is expressed as the discrepancy between a reference distribution and the achieved distribution, aiming for full control of the state distribution.
- Average cost: the objective is to minimize the expected cost, which corresponds to applying the certainty equivalence principle.
- Worst-case cost: a min-max formulation that minimizes the maximum possible cost under uncertainty. This is especially useful for bounded uncertainty, since otherwise the control law becomes overly conservative.
- Relative worst-case: the objective is to minimize the relative error between the system output (e.g., state) and the uncertainty level, a criterion particularly suited to unbounded uncertainties.
- Value at risk: minimize the cost function under the assumption that uncertainties remain below a given risk threshold α.
- Regret: minimize the discrepancy between the cost achieved by the chosen controller and the cost that would have been achieved by an oracle controller with full knowledge of the uncertainties, a formulation particularly relevant for online learning.

Once the cost function is defined, many formulations and solution strategies are possible. One of the main challenges in robust and stochastic control is to avoid excessive conservatism, which would decrease performance.

In order to illustrate how robust and stochastic control works, we present three representative approaches: H_∞ control, chance-constrained optimization, and sensitivity-based methods.

The H_∞ approach is a cornerstone of robust control, designed to guarantee stability and performance under uncertainty. Unlike stochastic methods, which optimize average performance, H_∞ control adopts a min-max perspective: it minimizes the worst possible impact of disturbances on the system. The problem can be written as:

$$\begin{cases} \min\limits_{\boldsymbol{u}} \max\limits_{\boldsymbol{w}} \sum\limits_{n=0}^{\infty} \boldsymbol{x}_n^\top \boldsymbol{Q} \boldsymbol{x}_n + \boldsymbol{u}_n^\top \boldsymbol{R} \boldsymbol{u}_n - \gamma^2 \boldsymbol{w}_n^\top \boldsymbol{w}_n \\ \text{s.t.} \\ \boldsymbol{x}_{n+1} = \boldsymbol{A}\boldsymbol{x}_n + \boldsymbol{B}\boldsymbol{u}_n + \boldsymbol{G}\boldsymbol{w}_n \,. \end{cases} \tag{2.11}$$

The term $-\gamma^2 \boldsymbol{w}_n^\top \boldsymbol{w}_n$ penalizes disturbances, and the parameter γ represents the $\mathcal{L}_2$ gain of the system, i.e., the maximum ratio between the output energy (LQR cost) and the disturbance energy. In other words, a smaller γ indicates greater robustness. A modified Riccati equation provides the analytical solution, which exists only if γ exceeds a certain threshold. This formulation thus limits the effect of disturbances. However, it also tends to reduce system excitation, which can harm parameter estimation accuracy. This issue, known as the "turn-off phenomenon" is illustrated in [Klenske, 2016] on a simple example of cautious control.

The second approach is chance-constrained optimization, which ensures that system constraints are satisfied with a specified minimum probability despite uncertainties. Typically, disturbances and measurement errors are modeled probabilistically, often as Gaussian distributions. The controller is then designed to maximize performance while enforcing probabilistic safety constraints, such as ensuring that the probability of collision remains below a fixed threshold α. This strikes a balance between robustness and performance, avoiding overly conservative solutions that would degrade system efficiency. A notable implementation is LQG-MP (Linear-Quadratic Gaussian Motion Planning) introduced in [van den Berg, 2011]. This method employs a Kalman filter to propagate uncertainties and an LQG controller to evaluate, before execution, the state distribution along candidate trajectories. From these, both the collision risk and LQR cost can be computed, enabling the selection of trajectories with acceptable risk and low control cost. The result of the chance-constrained algorithm applied to obstacle avoidance is shown in Figure 2.3.The approach is validated on a variety of systems, including mobile robots, multi-robot scenarios, and articulated manipulators. A related formulation is given in [Okamoto, 2018], which focuses solely on the propagation of uncertainty due to dynamics. In the linear case, the chance-constrained control problem can be cast as a semidefinite program.

The final approach, found in [Brault, 2021; Wasiela, 2023], is not standard in robust control but is of particular interest for our context due to its similarity with active sensing. The idea is to define a sensitivity measure with respect to uncertain parameters, given by the gradient of the system state with respect to those parameters. A trajectory is then sought that minimizes this sensitivity, thereby reducing the effect of parameter uncertainty. For example, the method has been applied to a quadrotor moving in the xz-plane with uncertain mass and inertia. The

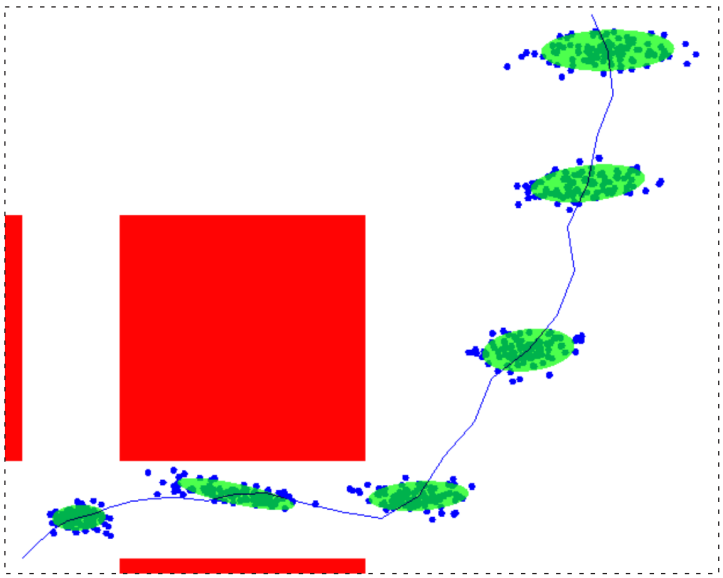

Figure 2.3: This figure, from [van den Berg, 2011], illustrates the result of their chance-constrained algorithm. We observe that the uncertainty ellipses do not intersect with the obstacles.

authors show that trajectories with low sensitivity lead to significantly improved performance.

To conclude, we have presented three representative approaches to robust and stochastic control. It should be emphasized, however, that many other methods exist, including stochastic LQR, MPC, tube MPC, polytopic containment, $\mathcal{L}_2$ control, and adaptive control. In general, robust and stochastic control aim to control systems in the presence of uncertainties without attempting to reduce those uncertainties directly. The goal is instead to limit their impact. This philosophy differs markedly from active sensing, which focuses on reducing uncertainties as much as possible before addressing average performance. In some cases, robust control is antagonistic to active sensing [Klenske, 2016], while in others it employs very similar tools [Wasiela, 2023].

5 Conclusion

This chapter has shown that although POMDPs constitute a rigorous but computationally intensive framework for planning under uncertainty, active sensing provides a practical, scalable alternative. By dynamically optimizing trajectories and sensor configurations, it reduces uncertainty while simultaneously fulfilling task objectives. We have also emphasized that active sensing is a well-established field, built upon a strong theoretical foundation from optimal experimental design and dual control theory. Importantly, it naturally addresses the exploration-exploitation trade-off, enabling robust performance without resorting to excessive conservatism. In this sense, active sensing is a versatile and powerful tool for solving a wide range of applications.

To summarize the different fields, Figure 2.4 presents, for each domain, an optimization problem. A shared aspect of these approaches is the formulation of an optimization problem. However, they differ significantly in the choice of decision variables, cost functions, and solution

methods.

The following chapters build upon these ideas to develop a unified framework for active sensing that leverages both theoretical insights and practical methods.

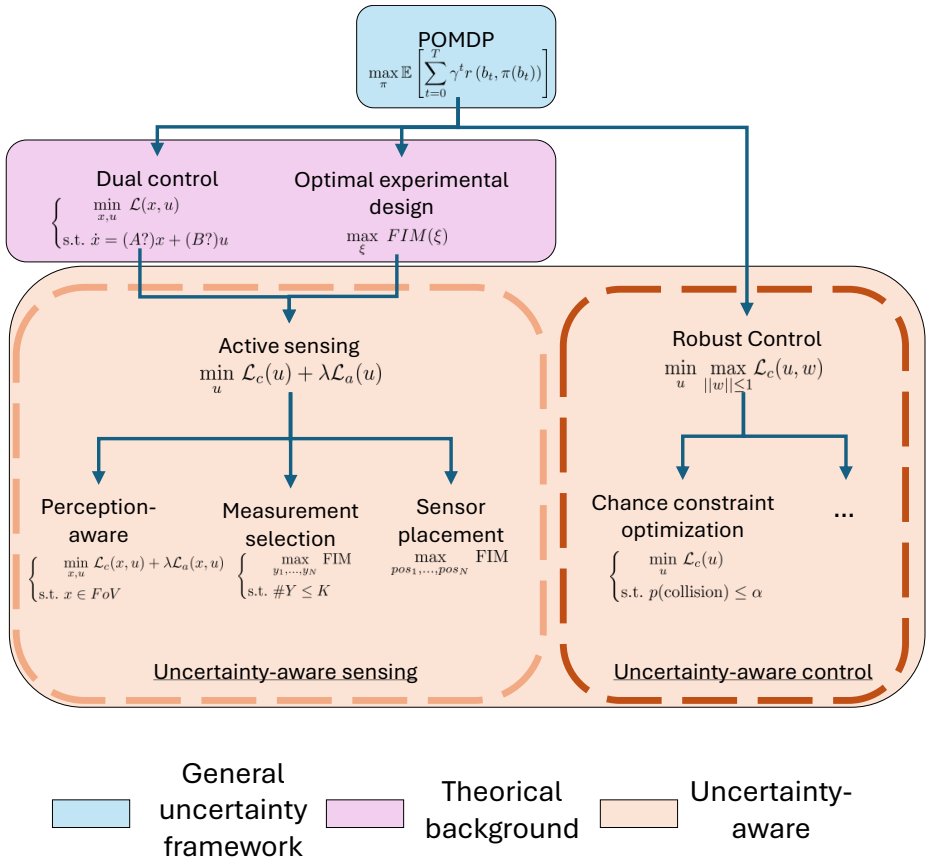

Figure 2.4: This diagram shows how different approaches to handling uncertainty in robotics are organized. It progresses from the most general theory to more specific ones. For each approach, we present a representative optimization problem to be solved.

Chapter 3

Overall Contributions of the Present Thesis

Contents

1 Positioning Our Contributions Within the Literature **31**
2 Our Approach to the Field of Active Sensing **34**
2.1 Contributions of the Present Thesis 35

Résumé

Ce chapitre a pour objectif de situer la perception active, et plus particulièrement notre approche, au sein de la littérature existante. Dans une première partie (Section 1), nous présentons le problème d'évitement de collision entre deux drones. Cet exemple illustre comment la perception active, telle que nous la considérons ici, s'inscrit dans la littérature du Chapitre 2. Dans une seconde partie (Section 2), nous rappelons l'organisation des contributions de la thèse et expliquons comment elles s'articulent.

This chapter aims to position active sensing, and more specifically our approach, within the existing literature. In the first section (Section 1), we use the example of drones collision avoidance to illustrate how active sensing fits into the literature presented in Chapter 2. In the second section (Section 2), we explain how the contributions of this thesis relate to one another.

1 Positioning Our Contributions Within the Literature

Autonomous aerial vehicles are becoming increasingly widespread across many applications, such as inspection, surveillance, and search and rescue. In some cases, several vehicles collaborate on a shared task, while in others, they operate in potentially hostile environments with non-cooperative vehicles. Moreover, they share the same airspace, which introduces collision risks. The objective is therefore to design a trajectory or control law that ensures collision avoidance and improves safety.

This problem is a typical example of planning under uncertainty. Indeed, collision avoidance

algorithms require knowledge of the position and possibly the velocity of other vehicles. However, these quantities are often noisy and only partially observable. For example, when a camera is used to estimate the position of another vehicle, only bearing measurements are available, which must be accumulated over time to estimate the relative position. To reduce the risk of collision, the objective is to carry out maneuvers that enable a good estimation of the other vehicle's state while simultaneously avoiding collision. The problem of choosing which action to take to avoid the intruder when the relative distance is unknown is illustrated in Figure 3.1.

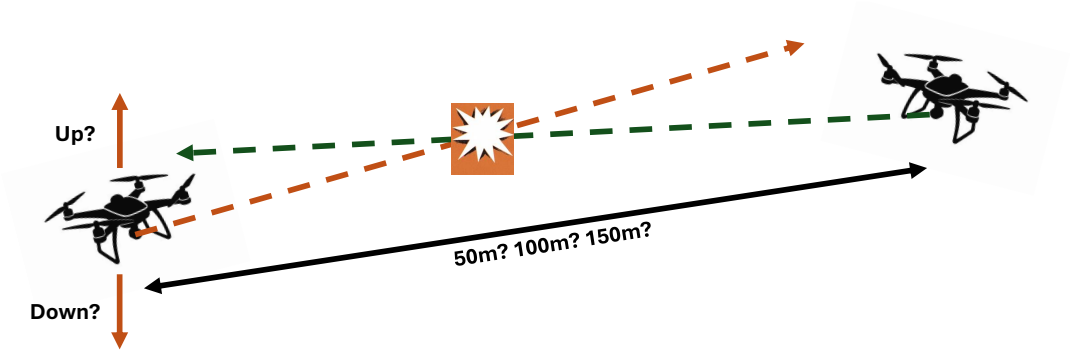

Figure 3.1: Illustration of the control problem aimed at reducing the risk of collision when the relative distance is unknown. One relevant question is how to adjust the altitude (up or down?) of our drone to decrease the risk.

Among the methods that address this problem, we find the main approaches presented in this chapter. POMDPs can be used to solve it, as demonstrated in [Bai, 2012]. The problem consists of determining the optimal policy that maximizes the discounted reward given the initial belief (b_0). The policy typically specifies whether the altitude should change or whether a turn should be made. As for the reward, a large penalty is assigned in the event of a collision, and a small penalty is applied to the controls to discourage unnecessary maneuvers. The problem is summarized in equation (3.1).

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n R(\boldsymbol{x}_n, \boldsymbol{u}_n) | b_0\right]. \tag{3.1}$$

The reported results are promising, as the approach significantly reduces the risk of collision. However, to keep the computational cost of POMDPs manageable, several simplifications are introduced:

- Simplified vehicle dynamics: altitude is constant, and a unicycle model is used for turns.
- Coarse discretization of the action space: only nine actions are available, three for turns and three for forward acceleration.
- Although POMDP methods are theoretically exact, their results are highly sensitive to the design of the reward function. For example, modifying the collision penalty can drastically change the resulting policy.

These limitations motivate the use of alternative approaches that relax or replace these simplifications.

Another approach to dealing with uncertainty is robust control. For the application, an example of such approach is discussed in [Kamel, 2017]. The authors formulate the problem as a Model Predictive Control (MPC) scheme that encourages each vehicle to maintain a safe distance, thus reducing collision risk. This is achieved through a soft constraint in the objective function and a hard safety constraint. Both are parameterized by a safety radius computed from the covariance of an EKF that estimates the other vehicle's position and velocity. For example, a simplified form of the hard constraint is:

$$\forall\, t \;||\boldsymbol{p}(t) - \boldsymbol{p}^{\text{intruder}}(t)|| \geq r_{\min}(t), \tag{3.2}$$

with $r_{\min}(t) = r_{\min} + 3\sigma$. σ is the square root of the maximum eigenvalue of the position covariance matrix.

This method makes the avoidance algorithm robust to measurement uncertainty. However, as with POMDPs, the problem formulation strongly influences the resulting behavior. Another drawback is that the controller is overly conservative. This causes the trajectories to deviate excessively from the reference or consume unnecessary energy. In extreme cases, large uncertainties may even render the MPC optimization problem infeasible.

Finally, the last approach involves active sensing, or perception-aware control since cameras are involved. This is the approach we adopt to develop our method, detailed in Chapter 6, Section 2. Our method builds upon prior work such as that of Li *et al.* [Li, 2023]. The idea is to control the system to avoid collisions based on deterministic controller using the estimates provided by an EKF. However, nothing guarantees that the other vehicle will strictly follow this mean behavior. To address this, we introduce a term in the cost function that encourages our vehicle to move in ways that reduce the uncertainty in the estimated position and velocity. Thereby lowering the overall collision risk. The assumptions underlying this approach can be summarized as follows:

- The control objective only accounts for the mean behavior of the other vehicle. For the collision avoidance task, the controller is deterministic.
- Uncertainties are only taken into account through a measure of uncertainty.
- The EKF covariance is assumed to correctly represent uncertainties.

These central assumptions enable a significant reduction in computational cost, as explained in Section 1.3. It should be noted, however, that like other approaches, our method has the drawback of being highly dependent on the formulation of the optimization problem.

In the rest of the manuscript, we present various applications of active sensing, each introducing contributions in both problem formulation and solution methodology.

2 Our Approach to the Field of Active Sensing

The objective of this section is to explain in more detail how our contributions are positioned within the field of active sensing.

We begin by examining a stochastic optimal control problem involving a nonlinear system. The goal is to design a controller that minimizes a cost function. The controller relies on state feedback, typically computed from partial and imperfect observations, which are obtained through a nonlinear measurement model. The dynamic and observation models are generally written as:

$$\begin{cases} \boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n, \boldsymbol{w}_n), \\ \boldsymbol{y}_n = h(\boldsymbol{x}_n) + \boldsymbol{v}_n, \end{cases} \tag{3.3}$$

where $\boldsymbol{x}_n$ is the system state, $\boldsymbol{u}_n$ is the control input, and $\boldsymbol{w}_n$ and $\boldsymbol{v}_n$ represent process and observation noise. The objective is to compute an optimal control sequence or policy that selects controls to minimize an expected cost function, such as

$$\mathcal{L} \coloneqq \mathbb{E}_{\boldsymbol{w}_{1:N}, \boldsymbol{v}_{0:N-1}} \left[\sum_{n=1}^{N} l(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n) \right]. \tag{3.4}$$

This formulation constitutes the core of POMDPs (Chapter 2, Section 1), as it captures the fundamental challenge of acting under uncertainty. However, solving a nonlinear stochastic optimal control problem remains particularly demanding from a computational standpoint, especially given that the control inputs shall only depend on partial past observations, as the state is unknown.

To simplify the problem, the effects of partial observability can be ignored and the stochastic nature of the system can be eliminated by replacing the noise terms with their most probable value. Since the process and observation noises are typically modeled as zero-mean Gaussian variables, their most likely value is $\mathbf{0}$. The system to be controlled then becomes:

$$\begin{cases} \min\limits_{\boldsymbol{u}_{1:N}} \mathcal{L} \coloneqq \sum\limits_{n=1}^{N} l(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n)\,, \\ \text{s.t.} \quad \boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n)\,. \end{cases} \tag{3.5}$$

Solving a deterministic optimal control problem then becomes significantly simpler. However, the resulting controller may perform poorly because the partial nature of the feedback is ignored. The typical approach is to "assume" an approximate separation principle, and to address the following problem instead:

$$\begin{cases} \min\limits_{\boldsymbol{u}_{1:N}} \mathcal{L} \coloneqq \sum\limits_{n=1}^{N} l(\hat{\boldsymbol{x}}_{n-1}, \boldsymbol{u}_n)\,, \\ \text{s.t.} \quad \hat{\boldsymbol{x}}_n = f(\hat{\boldsymbol{x}}_{n-1}, \boldsymbol{u}_n)\,. \end{cases} \tag{3.6}$$

where the notation $\hat{\boldsymbol{x}}$ underlines that an observer is used to estimate the state from the measurements, and as the most likely (future) values for the noise in (3.3) are null, the observer is considered to consist of a mere copy of the dynamics. However, after solving the optimal

planning problem, when implementing this strategy to actually perform the trajectory, the observations differ from their most likely values, and the observer does correct the estimated states (beyond propagating them through the dynamics as is done for planning). A common idea is therefore to retain the observation model and to augment the cost function with a criterion that promotes observability, so as to obtain trajectories where the observer will efficiently play its role, enabling accurate path following. For instance see [Salaris, 2019], where the objective is

$$\mathcal{L} \coloneqq -\sum_{n=0}^{N} \mathcal{O}(\boldsymbol{x}_n), \tag{3.7}$$

where $\mathcal{O}$ denotes an observability measure, such as the local observability Gramian. The idea of selecting trajectories that maximize observability forms the basis of active sensing approaches relying on observability metrics.

However, these approaches suffer from the limitation that they do not directly account for the uncertainty associated with either the process model or the observation model. A natural solution is to use as a criterion the a posteriori covariance computed by the EKF. More precisely, one may consider a scalar measure of the covariance. For example, the objective can be defined as

$$\mathcal{L} \coloneqq \sum_{n=0}^{N} \operatorname{Tr}(\boldsymbol{P}_{n|n}), \tag{3.8}$$

where $\boldsymbol{P}_{n|n}$ denotes the a posteriori covariance produced by the EKF. The relevance of this information measure has already been demonstrated, for instance, in [Patil, 2015]. However, the numerical cost associated with the optimization of such a criterion has long been prohibitive, see, e.g., [Preiss, 2018]. This is owed to the recursive nature of the Extended Kalman filter, that makes the last covariance's dependency on an early action complicated. The active sensing approaches presented in Chapters 5 and 6 are based on the covariance computed by the EKF, where efficient optimization is enabled by novel analytical formulas derived in Chapter 4.

2.1 Contributions of the Present Thesis

Optimizing a scalar measure of the covariance is challenging because the influence of a control input or a state on the covariance is not straightforward. To address this difficulty, Chapter 4 presents a method that analytically computes the gradient of the cost function with respect to the control inputs. This approach is essential for reducing the computational cost associated with other techniques, such as finite difference methods or sensitivity equations. The method, based on the principle of backpropagation, proves particularly well suited for active sensing. Indeed, these equations make it possible to efficiently compute

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}}^{\top} = \left[\frac{\partial \mathcal{L}}{\partial u_1}^{\top}, \frac{\partial \mathcal{L}}{\partial u_2}^{\top}, \ldots, \frac{\partial \mathcal{L}}{\partial u_N}^{\top} \right]. \tag{3.9}$$

With this gradient, the optimal control problem involving a covariance-based objective can be solved using a first-order numerical optimizer (gradient descent, quasi-Newton methods, etc.).

All subsequent chapters rely on efficient backpropagation-based computation of gradients of the EKF's uncertainty, whether based on the analytical formulas of Chapter 4, or on modern numerical methods.

Chapter 5 then introduces a formulation of the active sensing problem for a bicycle model, with the goal of minimizing the covariance while ensuring admissibility with respect to the kinematic model. The method produces trajectories that lead to improved state estimation. However, it remains limited to small-dimensional examples.

Chapter 6 aims to overcome these limitations, in particular for active sensing involving a quadrotor. The first lever is the use of differential flatness theory, which enables the generation of admissible trajectories. The second lever consists in introducing a mixed objective function of the form

$$\mathcal{L} = \mathcal{L}_c(\boldsymbol{u}, \boldsymbol{x}) + \lambda \mathcal{L}_a(\boldsymbol{P}_{1|1}, \dots, \boldsymbol{P}_{N|N}). \tag{3.10}$$

The first term corresponds to a standard control objective such as trajectory tracking, energy minimization, or similar goals. The second term corresponds to an active sensing objective based on the EKF covariance. This yields a deterministic optimal control problem in which uncertainty is nonetheless incorporated through the covariance dynamics.

Furthermore, writing the problem in this way allows the control and estimation objectives to pertain to two different dynamical systems. For instance, in the collision avoidance application (see Chapter 6, Section 2), the control objective acts on the controlled drone, which must keep a safe distance from a dynamic obstacle, whereas the active sensing term improve the quality of the estimate of the intruder drone's position and velocity.

Once an optimization problem has been formulated and can be solved, it can be implemented in various ways. This may involve fully offline planning, as in the dynamic alignment example of Chapter 6; stopping the system to update the trajectory, as in the lever-arm calibration problem of Chapter 5; or using a receding horizon approach, as in Model Predictive Control (MPC) for the collision avoidance application in Chapter 6. The degree of feedback can serve as a lever to reduce computational constraints, as discussed in Chapter 2, Section 3.1.2.

When optimizing the trajectory of a drone to improve the estimation of an intruder's position, this is equivalent to optimizing external parameters that influence the estimation process. Accordingly, Chapter 7 formulates an optimization problem for selecting external parameters to minimize the covariance. This makes it possible to address the measurement selection problem. In addition, the problem of positioning a mobile sensor can also be handled within this framework. These applications leverage the backpropagation equations to efficiently compute the gradient of the covariance with respect to the chosen parameters.

Chapter 4

General Backpropagation Equations through Extended Kalman Filter

Contents

1 The Extended Kalman Filter **38**
2 Analytical Result on Active Sensing **40**
3 Gradient Computation Through the EKF **42**
3.1 Considered Problems 42
3.2 Overview of Gradient Computation in EKF 43
3.3 Fundamental Principles of Backpropagation 44
3.4 Matrix Derivatives 45
4 Backpropagation Through the Extended Kalman Filter **46**
4.1 Backpropagation Through the Propagation Step 46
4.2 Backpropagation Through the Update Step 48
5 Conclusion **49**

Résumé

Ce chapitre traite du calcul efficace du gradient d'une sortie d'un filtre de Kalman étendu (EKF) par rapport à un grand nombre de ses paramètres. Par "sortie", on entend des fonctions de coût telles que la log-vraisemblance ou la somme des traces de covariance ; par "paramètres", on inclut les contrôles, les matrices de covariance ou encore les jacobiens. Chaque gradient possède ses propres applications, par exemple le gradient de la covariance par rapport aux contrôles, essentiel en perception active.

Après avoir rappelé le fonctionnement de l'EKF et présenté les principales méthodes de calcul de gradient (approches analytiques, différences finies, différentiation automatique ou calcul symbolique), nous expliquons pourquoi la différentiation automatique en mode rétropropagation constitue une solution particulièrement adaptée à la perception active. Nous introduisons ensuite un ensemble d'équations permettant de calculer itérativement les gradients par rétropropagation. Cette approche offre des avantages

notables pour l'embarqué, notamment un gain de performance et une meilleure traçabilité des calculs. Ces équations constituent ainsi un outil fondamental pour la perception active.

Par ailleurs, avant d'aborder les équations de rétropropagation, nous présentons brièvement un résultat analytique : la détermination de trajectoires maximisant une mesure d'observabilité pour une certaine classe de systèmes, montrant ainsi la difficulté à résoudre analytiquement un problème de perception active.

Once the active sensing problem has been formulated as an optimization problem, the main challenge lies in how to find an optimal solution. First-order optimization algorithms (i.e., those that use gradient information) are often used, as they offer a good trade-off between convergence speed and computational cost [Nocedal, 1999]. However, computing the gradient of the cost function can be challenging in certain active sensing problem formulations. In this chapter, we introduce a gradient computation method based on backpropagation through the Extended Kalman Filter (EKF). Such equations allow us to perform first-order optimization of active sensing problems formulated using the covariance provided by an EKF. Before delving into the details of the method, we briefly present an analytical result on active sensing obtained during our work.

This chapter is organized as follows. First, we introduce the EKF notations in Section 1. Then, Section 2 begins by presenting an analytical result on active sensing. This result highlights both the inherent difficulty of deriving analytical solutions for active sensing problems, and the effectiveness of invariant filtering [Barrau, 2017]. Next, Section 3 focuses on gradient computation through the EKF, reviewing existing methods, explaining why the backpropagation approach is particularly advantageous, and introducing elements of matrix derivatives. Finally, Section 4 presents the analytical backpropagation equations within the EKF.

1 The Extended Kalman Filter

To ensure a smoother reading of this chapter, we begin by introducing the filter used in our approach. These notations will be used throughout the manuscript. For generality, we consider the Error-State Extended Kalman Filter (ES-EKF) [Grewal, 2015]. This filter focuses on estimating the error on the system state and the covariance of the error. It encompasses methods defining a representation of the state based on Lie groups [Brossard, 2022], such as the Invariant Extended Kalman Filter (IEKF) [Barrau, 2017; Barrau, 2023]. We consider a nonlinear discrete-time system:

$$\begin{cases} \boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n, \boldsymbol{w}_n), \quad \boldsymbol{x}_0 = \boldsymbol{x}^{\mathbf{0}}, \\ \boldsymbol{y}_n = h(\boldsymbol{x}_n) + \boldsymbol{v}_n. \end{cases} \tag{4.1}$$

where $\boldsymbol{x}_n \in S$ is the system state, $\boldsymbol{u}_n \in \mathbb{R}^q$ is the control input, and $\boldsymbol{w}_n \in \mathbb{R}^r$ is the process noise, which follows a Gaussian distribution with zero mean and covariance matrix $\boldsymbol{Q}_n$. The measured output is denoted by $\boldsymbol{y}_n \in \mathbb{R}^l$, and is corrupted by Gaussian white noise $\boldsymbol{v}_n$ with zero

mean and covariance matrix $\boldsymbol{R}_n$. The state space S is a smooth manifold of dimension p. For instance, S may be the special orthogonal group $SO(3)$ crossed with $\mathbb{R}^6$, in order to represent the orientation, position, and velocity of the system.

The error function is defined as:

$$\begin{aligned} e : S \times S &\to \mathbb{R}^p \\ (\boldsymbol{x}, \hat{\boldsymbol{x}}) &\mapsto e(\boldsymbol{x}, \hat{\boldsymbol{x}}) \end{aligned} \tag{4.2}$$

By taking a linear error, $e(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \boldsymbol{x} - \hat{\boldsymbol{x}}$, we recover the classical EKF equations, with Jacobians as the transition matrices. We denote the conditional error as $\boldsymbol{e}_{n|i} = e(\boldsymbol{x}_n, \hat{\boldsymbol{x}}_{n|i})$.

The filter consists of two steps: propagation and update. In the propagation step, the state estimate is propagated using the system dynamics, along with the covariance of the error:

$$\begin{aligned} \hat{\boldsymbol{x}}_{n|n-1} &= f(\hat{\boldsymbol{x}}_{n-1|n-1}, \boldsymbol{u}_n, \boldsymbol{0}) \,, \\ \boldsymbol{P}_{n|n-1} &= \boldsymbol{F}_n \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_n^\top + \boldsymbol{G}_n \boldsymbol{Q}_n \boldsymbol{G}_n^\top \,. \end{aligned} \tag{4.3}$$

with $\boldsymbol{F}_n$ and $\boldsymbol{G}_n$ resulting from the linearization of the error around zero, since we assume that the estimated value is close to the true value:

$$\boldsymbol{e}_{n|n-1} \simeq \boldsymbol{F}_n \boldsymbol{e}_{n-1|n-1} + \boldsymbol{G}_n \boldsymbol{w}_n \,. \tag{4.4}$$

The second step is the update, where the state estimate is corrected using a noisy and often partial measurement of the system state, denoted by y_n:

$$\begin{aligned} \boldsymbol{S}_n &= \boldsymbol{H}_n \boldsymbol{P}_{n|n-1} \boldsymbol{H}_n^\top + \boldsymbol{R}_n \\ \boldsymbol{K}_n &= \boldsymbol{P}_{n|n-1} \boldsymbol{H}_n^\top \boldsymbol{S}_n^{-1} \\ \boldsymbol{z}_n &= \boldsymbol{y}_n - h(\hat{\boldsymbol{x}}_{n|n-1}) \\ \hat{\boldsymbol{x}}_{n|n} &= \varphi(\hat{\boldsymbol{x}}_{n|n-1}, \boldsymbol{K}_n \boldsymbol{z}_n) \\ \boldsymbol{P}_{n|n} &= (\boldsymbol{I} - \boldsymbol{K}_n \boldsymbol{H}_n) \boldsymbol{P}_{n|n-1} \,. \end{aligned} \tag{4.5}$$

with φ the retraction from: $S \times \mathbb{R}^p \to S$ and $\boldsymbol{H}_n$ resulting from the linearization of the innovation around zero:

$$\boldsymbol{z}_n = \boldsymbol{y}_n - h(\hat{\boldsymbol{x}}_{n|n-1}) \simeq \boldsymbol{H}_n \boldsymbol{e}_{n|n-1} + \boldsymbol{\nu}_n \,. \tag{4.6}$$

The retraction φ is a mapping from the tangent space (Locally linearized space) of a manifold to the manifold. For example, on $SO(3)$, the matrix exponential maps a skew-symmetric matrix to a rotation matrix, providing a natural retraction. This retraction is widely used in robotics applications.

Having introduced the notations, we can return to the heart of the matter.

2 Analytical Result on Active Sensing

As a first step, one may consider simple systems, and try to derive analytical results for the active sensing problem, that is, trajectories that do optimize a sensor information criterion. By considering as a criterion a measure of the observability (observability or constructibility Gramian), and by considering simple systems highlighted by the theory of invariant Kalman filtering, we could obtain some results in this direction. They were published in [Benhamou, 2023]. In this section, we summarize only the key ideas that led to these results, as we would like to turn to more complicated systems and to covariance-based metrics in the present thesis. Further details and derivations can be found in the paper. Interestingly, we needed to prove a formula that provides a stochastic interpretation of the observability Gramian, that we could not find in the literature, and that we will shortly provide.

The problem we consider consists of characterizing the trajectories that maximize the observability Gramian of the dynamical system. A dynamical system is said to be observable if its initial state can be estimated from successive partial measurements of its state. In particular, the properties of the observability Gramian provide valuable insights into the system's observability. Considering a dynamic model and an observation model of the form (4.1), the observability Gramian over a fixed horizon N is defined as:

$$\boldsymbol{W_N} \coloneqq \sum_{n=1}^{N} \boldsymbol{\Phi_n^\top H_n^\top H_n \Phi_n}, \tag{4.7}$$

where $\boldsymbol{H_n} = \frac{\partial h(\boldsymbol{x}_n)}{\partial \boldsymbol{x}}$, $\boldsymbol{\Phi}_n = \boldsymbol{F}_n \boldsymbol{\Phi}_{n-1}$, $\boldsymbol{\Phi}_{-1} = \boldsymbol{I}_d$, and $\boldsymbol{F}_{n+1} = \frac{\partial f(\boldsymbol{x}_n, \boldsymbol{u}_{n+1}, \boldsymbol{0})}{\partial \boldsymbol{x}}$.

When the information measure used in the active sensing problem is the trace of the observability Gramian, the objective is to find trajectories that maximize this criterion. Solving this problem directly is challenging because the Gramian depends not only on the sequence of control inputs but also on the system states along the trajectory. Furthermore, the resulting trajectory must satisfy both the system dynamics and actuator constraints. To reformulate the problem as an unconstrained optimization, we rely on two key elements.

The first element is a reinterpretation of the observability Gramian. When the control sequence is fixed, let $\boldsymbol{y}_n(\boldsymbol{x}_0)$ denote the measurement at time n obtained from the initial state $\boldsymbol{x}_0$, and let $\boldsymbol{\delta x}_n = \boldsymbol{x}_n(\boldsymbol{x}_0 + \boldsymbol{\delta x}_0) - \boldsymbol{x}_n(\boldsymbol{x}_0)$ be the difference between two trajectories whose initial states differ by $\boldsymbol{\delta x}_0$. If we further assume that $\boldsymbol{\delta x}_0$ follows a normal distribution, $\boldsymbol{\delta x}_0 \sim \mathcal{N}(0, \boldsymbol{I}_p)$, then the following relation holds:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\delta x}_0}\left[\sum_{n=1}^{K}||\boldsymbol{y}_n(\boldsymbol{x}_0+\boldsymbol{\delta x}_0)-\boldsymbol{y}_n(\boldsymbol{x}_0)||_2^2\right] &\approx \mathbb{E}_{\boldsymbol{\delta x}_0}\left[\sum_{n=1}^{K}||\boldsymbol{H}_n\boldsymbol{\delta x}_n||_2^2\right] \\
&\approx \mathbb{E}_{\boldsymbol{\delta x}_0}\left[\sum_{n=1}^{K}\boldsymbol{\delta x}_0^\top\boldsymbol{\Phi}_n^\top\boldsymbol{H}_n^\top\boldsymbol{H}_n\boldsymbol{\Phi}_n\boldsymbol{\delta x}_0\right] \\
&\approx \sum_{n=1}^{K}\mathbb{E}_{\boldsymbol{\delta x}_0}\left[\mathrm{Tr}\left(\boldsymbol{\delta x}_0^\top\boldsymbol{\Phi}_n^\top\boldsymbol{H}_n^\top\boldsymbol{H}_n\boldsymbol{\Phi}_n\boldsymbol{\delta x}_0\right)\right] \\
&\approx \mathrm{Tr}\left(\sum_{n=1}^{K}\boldsymbol{\Phi}_n^\top\boldsymbol{H}_n^\top\boldsymbol{H}_n\boldsymbol{\Phi}_n\mathbb{E}_{\boldsymbol{\delta x}_0}\left[\boldsymbol{\delta x}_0\boldsymbol{\delta x}_0^\top\right]\right) \\
&\approx \mathrm{Tr}\left(\boldsymbol{W}_{\boldsymbol{K}}\right).
\end{aligned}
\tag{4.8}
$$

In the case of a linear system, the approximations can be replaced by equalities.

In the same spirit, we can derive the following relation for the smallest eigenvalue of the observability Gramian:

$$
\min_{||\boldsymbol{\delta x}_0||=1}\sum_{n=1}^{K}||\boldsymbol{y}_n(\boldsymbol{x}_0+\boldsymbol{\delta x}_0)-\boldsymbol{y}_n(\boldsymbol{x}_0)||_2^2 \approx \lambda_{min}(\boldsymbol{W}_{\boldsymbol{K}}) \tag{4.9}
$$

These two relations provide, on one hand, a clearer interpretation of the observability Gramian, and on the other hand, allow reformulating the active perception problem as follows: characterize the trajectories that, on average, induce the largest variation in the observation sequence.

The second key element is to restrict our analysis to systems that can be described within the framework of the *two-frames theory* [Barrau, 2023]. This theory defines a class of systems for which invariant filtering can be applied. In particular, we exploit the fact that, in this framework, the Jacobians are independent of the system state which is an important property of invariant filtering. As a result, the optimization problem no longer explicitly depends on the state, and the dynamic feasibility constraint is implicitly satisfied, leading to an unconstrained optimization problem.

Using these two key elements, we were able to establish the following results:

- Measurements in the fixed frame: for example, a unicycle equipped with a GNSS sensor with a lever arm. The optimal trajectory is obtained by applying the maximum admissible acceleration while maintaining zero angular velocity.

- Measurements in the vehicle frame: in this case, all trajectories are equivalent. This corresponds, for instance, to the SLAM problem when the full position of landmarks is measured in the robot's frame, as opposed to visual SLAM where only their direction is observed.

This analytical study leads to several conclusions. First, the trace of the observability Gramian does not appear to be a relevant criterion, since the trajectories that maximize it are not always locally observable. Second, it is extremely challenging to derive analytical results

for active sensing problems, even in relatively simple settings. Numerical methods therefore appear to be more suitable for solving active sensing problems. This motivates the development for numerical approaches, particularly those that allow the computation of the gradient of an information measure to be used within an optimization solver. In this context, we have extended the backpropagation equations for Kalman Filter [Parellier, 2023b] to EKF framework. For applications related to active sensing, this extension is of primary importance. Indeed, if the linearization point of the system is not taken into account, the Kalman filter covariances can be computed off-line, and any modification of the control sequence will have no impact on the covariance.

3 Gradient Computation Through the EKF

3.1 Considered Problems

To enhance EKF performance, it is often valuable to evaluate how variations in its parameters affect the filter's behavior. Knowledge of these variations (i.e., gradients) has numerous applications. For example, it can be used to tune the process and observation noise covariance matrices. Other applications include active sensing, learning system dynamics or observation models, measurement selection, and sensor placement.

To address all these problems, the goal is to compute the gradient of the EKF output with respect to as many parameters as possible. We adopt the following general EKF output to encompass a wide range of applications:

$$\mathcal{L} = \sum_{n=1}^{N} l_n \left(\hat{\boldsymbol{x}}_{n|n-1}, \boldsymbol{P}_{n|n-1}, \boldsymbol{Q}_n, \hat{\boldsymbol{x}}_{n|n}, \boldsymbol{P}_{n|n}, \boldsymbol{y}_n, \boldsymbol{R}_n, \boldsymbol{S}_n \right) . \tag{4.10}$$

This generic loss formulation is versatile and can be used to express a wide range of quantities of interest. In this general framework, we can write the negative log-likelihood: $\mathcal{L}_{NLL} = \sum_{n=1}^{N} \log\det \boldsymbol{S}_n + \boldsymbol{z}_n^\top \boldsymbol{S}_n^{-1} \boldsymbol{z}_n$,. Then, one can automatically tune covariance matrices. Active sensing loss functions fall into this formulation: $\mathcal{L}_{PA} = \sum_{n=1}^{N} l\left(\boldsymbol{P}_{n|n}\right)$ where l is a scalar function, such as the trace, the log-determinant, or the largest eigenvalue [Cognetti, 2018]. Moreover, it can represent the root mean square error (RMSE) when the ground truth is available. Another important loss that fits this form is the normalized estimation error squared (NEES). In this case, the goal is to find parameters that ensure the NEES equals the state dimension, thereby making the filter consistent. Furthermore, the generic loss formulation can address other challenges, such as detecting spurious measurements in an EKF using the gradient of the NLL with respect to the observations [Parellier, 2023a].

Then, the goal is to compute the gradient of the generic loss (4.10) with respect to various EKF parameters. This gradient information is essential for optimizing the performance of the EKF in different scenarios. Additionally, achieving fast computation of these gradients is vital, especially for real-time applications and onboard systems where computational resources are limited. Ultimately, we seek a method that can seamlessly operate on onboard computers.

3.2 Overview of Gradient Computation in EKF

Given the importance of computing gradients in an EKF, many approaches have been developed and applied in the past. As discussed in [Baydin, 2018], there are four principal methods for computing gradients:

- Analytical gradient computation: This approach is often too cumbersome.
- Finite differences method: This method numerically approximates the gradient by applying a small perturbation. Although it is quick to implement, it is numerically unstable and computationally expensive.
- Symbolic differentiation [Meurer, 2017]: This is an exact method; however, when dealing with complex functions, it encounters the problem of "expression swell".
- Automatic differentiation [Paszke, 2017; Bradbury, 2018]: This is an exact method that decomposes a complex function into smaller operations for which the derivative is known. Automatic differentiation exists in two modes: forward and backward.

In the context of tuning the process and observation noise covariances in the EKF, sensitivity equations [Gupta, 1974; Särkkä, 2023] have been developed to compute gradients, which can be interpreted as analytical formulations for forward differentiation. For covariance tuning, the goal is to determine the covariance matrices that minimize the negative log-likelihood (NLL) of the innovation sequence. This requires evaluating the gradient of the NLL with respect to the covariance matrices. Once these gradients are available, a first-order optimization algorithm can be employed to find the optimal tuning of the covariances [Hassani, 2013]. Sensitivity equations have also been extended to the square-root Kalman filter, improving numerical stability [Kulikova, 2015]. However, when the objective differs from the NLL; such as when using the residual prediction error, sensitivity equations become invalid. Additionally, Abbeel *et al.* [Abbeel, 2005] demonstrate that the residual prediction error criterion is particularly relevant for adaptive filtering.

With the recent rise of deep learning and the need for gradient computation tools, many automatic differentiation (AD) methods have been developed [Paszke, 2017; Bradbury, 2018]. These methods make it possible to compute the gradient of very complex functions by decomposing the computations into smaller ones. Even though the method is numerical (as opposed to analytical methods), AD methods perform exact computations, without the need for numerical approximations. Advantages and further explanations of AD methods can be found in [Baydin, 2018] and some of them are discussed in the next section. Such tools have also been used to compute the gradient within an EKF. For example, Haarnoja *et al.* [Haarnoja, 2016] use automatic differentiation to learn a compact representation of the observation model from high-dimensional data (images) and the associated observation covariance matrix for data fusion. Similarly, Brossard *et al.* [Brossard, 2020] learn an observation covariance matrix generator to improve inertial-odometer fusion with pseudo-measurements. A final example is the use of AD tools to solve an active sensing problem, formulated as an optimization problem, where the

gradient of the cost function is expensive to compute numerically [Patil, 2015]. Although AD tools are powerful and widely used in deep learning, they can make gradient computation less transparent compared to manually computed gradients. Additionally, they may not be available for specific hardware, which can be a limitation for onboard systems.

To overcome this difficulty, but also to speed up the computation time, Parellier *et al.* [Parellier, 2023b] developed an analytical method based on the backpropagation of the gradients through a Kalman filter (KF). They showed that this analytical backpropagation leads to time savings compared to backpropagation performed by automatic differentiation with PyTorch or compared to sensitivity equations. More recently, this work has been extended to EKF in [Xu, 2024] with application to covariance tuning. In the same spirit, herein we derive general EKF backpropagation equations.

3.3 Fundamental Principles of Backpropagation

As discussed in [Baydin, 2018], automatic differentiation tools provide a powerful approach for a broad spectrum of computational tasks. In this section, we outline the specific advantages of using AD in our applications, with a particular emphasis on the benefits of employing backward mode differentiation.

To begin, we will explain how an automatic differentiation tool works. The first step is to construct a computational graph, breaking down a complex function into simpler operations for which derivatives are readily available.

Consider a simple example: let $\alpha \in \mathbb{R}$ and $\mathcal{L}$ be a loss function, where the computation of $\mathcal{L}$ proceeds through intermediate variables x, y and z. Specifically, x is used to compute y and z, which in turn are used to compute the final quantity $\mathcal{L}$. This entire computation process is represented in the Figure 4.1.

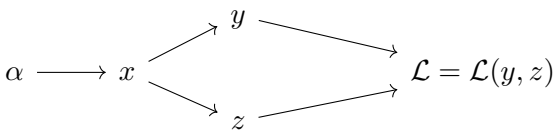

Figure 4.1: A basic example of a computational graph.

From the computational graph, there are two approaches to calculate the gradient of $\mathcal{L}$ with respect to α. The first approach is called the forward mode, and the second is the backward mode. Both approaches apply the chain rule but in different ways, ultimately producing the same result.

In forward mode, we propagate a perturbation of the input parameter, here α, through to the final output $\mathcal{L}$. Thus, in our example, we successively compute the following quantities:

1. $\frac{\partial x}{\partial \alpha}$
2. $\frac{\partial y}{\partial \alpha} = \frac{\partial y}{\partial x}\frac{\partial x}{\partial \alpha}$ and $\frac{\partial z}{\partial \alpha} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial \alpha}$
3. $\frac{\partial \mathcal{L}}{\partial \alpha} = \frac{\partial \mathcal{L}}{\partial y}\frac{\partial y}{\partial \alpha} + \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial \alpha}$

The second approach is to use the backward mode. In this mode, we start from the final gradient and work backward through the graph to trace the origin of the perturbation. In our example, the following computations are performed:

1. $\frac{\partial \mathcal{L}}{\partial y}$ and $\frac{\partial \mathcal{L}}{\partial z}$
2. $\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial \mathcal{L}}{\partial y}\frac{\partial y}{\partial x} + \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial x}$
3. $\frac{\partial \mathcal{L}}{\partial \alpha} = \frac{\partial \mathcal{L}}{\partial x}\frac{\partial x}{\partial \alpha}$

At first glance, the two methods may not seem different, aside from the order in which gradients are accumulated. However, when $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}^m$, the order of gradient/Jacobian accumulation has a significant impact. This is particularly relevant when m is much smaller than n. The key point is that matrix multiplication, specifically $(AB)C$ versus $A(BC)$, has a different computational cost. When the matrix C is "small", the second option is more efficient, and this is precisely what backpropagation exploits.

Another advantage of the backward mode is that it provides access to the gradient of the final output with respect to intermediate variables without additional computational cost. In our example, we obtain $\frac{\partial \mathcal{L}}{\partial x}$ for free in the backward mode, but not in the forward mode.

3.4 Matrix Derivatives

To derive the formulas of loss (4.10) with respect to the various EKF parameters, we first introduce some matrix derivatives. The following formulas are derived via the chain rule.

Consider $\mathcal{L}(\boldsymbol{M})$, a scalar function of a matrix defined as $\boldsymbol{M} = f(\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are also matrices. $\frac{\partial \mathcal{L}}{\partial \boldsymbol{M}}$ denotes the matrix defined by $\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{M}}\right)_{ij} = \frac{\partial \mathcal{L}}{\partial M_{ij}}$. We write $\frac{\partial \mathcal{L}}{\partial \boldsymbol{X}}$ to denote $\frac{\partial \mathcal{L} \circ f}{\partial \boldsymbol{X}}$, to simplify the notation.

Using the chain rule, we have the following formulas [Petersen, 2012]:

$$\begin{aligned}
\boldsymbol{M} = \boldsymbol{X}\boldsymbol{Y}\boldsymbol{X}^\top \quad &\Rightarrow \frac{\partial \mathcal{L}}{\partial \boldsymbol{X}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{M}}\boldsymbol{X}\boldsymbol{Y}^\top + \frac{\partial \mathcal{L}}{\partial \boldsymbol{M}}^\top \boldsymbol{X}\boldsymbol{Y}\,, \\
\boldsymbol{M} = \boldsymbol{Y}\boldsymbol{X}\boldsymbol{Y}^\top \quad &\Rightarrow \frac{\partial \mathcal{L}}{\partial \boldsymbol{X}} = \boldsymbol{Y}^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{M}}\boldsymbol{Y}\,, \\
\boldsymbol{M} = \boldsymbol{X}^{-1} \quad &\Rightarrow \frac{\partial \mathcal{L}}{\partial \boldsymbol{X}} = -\boldsymbol{M}^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{M}}\boldsymbol{M}^\top\,,
\end{aligned} \tag{4.11}$$

In the particular case where $\boldsymbol{x}$ and $\boldsymbol{m}$ are vectors, we have:

$$\boldsymbol{m} = f(\boldsymbol{x}) \Rightarrow \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}} = \boldsymbol{J}^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{m}}, \tag{4.12}$$

with $\boldsymbol{J}$ the Jacobian matrix of f w.r.t. $\boldsymbol{x}$.

When the matrix $\boldsymbol{X}$ depends on a scalar variable s, we have the following formula:

$$\boldsymbol{M} = X(s) \Rightarrow \frac{d\mathcal{L}}{ds} = \mathrm{Tr}\left(\frac{d\boldsymbol{X}^\top}{ds}\frac{\partial \mathcal{L}}{\partial \boldsymbol{M}}\right). \tag{4.13}$$

At first glance, computing such quantities requires approximately $O(n^3)$ operations. However, in practice, we use a different formula: $\frac{d\mathcal{L}}{ds} = \sum_{i,j} \left(\frac{d\boldsymbol{X}^\top}{ds} \odot \frac{\partial \mathcal{L}}{\partial \boldsymbol{M}} \right)_{i,j}$, where $\odot$ denotes the pointwise product. This reduces the computational complexity to $O(n^2)$.

4 Backpropagation Through the Extended Kalman Filter

To derive the backpropagation equations, we adopt a similar approach to that used in automatic differentiation. However, in the case of the EKF, some parts of the computational graph remain constant. Therefore, it is advantageous to compute all analytical equations once and for all, eliminating the need to repeatedly construct the computational graph. This manual derivation offers two key benefits. First, it accelerates computations, which is crucial for embedded applications. Second, in applications requiring certification, manually derived analytical equations are more easily verifiable than relying entirely on an automatic differentiation tool.

Similar to the EKF, which consists of a propagation step and an update step, the backpropagation of gradients follows the same two stages. These steps are independent, and we treat them as such when deriving the backpropagation equations. Hence, as in the EKF, these two decoupled stages can be applied at different frequencies: propagation is often performed at a high frequency, while update is typically executed at a lower frequency.

4.1 Backpropagation Through the Propagation Step

The propagation step begins with the current state estimate and its associated uncertainty, which are then propagated through the system dynamics. The computational graph of this propagation step is shown in Figure 4.2. During forward propagation, we begin with $\hat{\boldsymbol{x}}_{n-1|n-1}$ and $\boldsymbol{P}_{n-1|n-1}$ and proceed to compute $\hat{\boldsymbol{x}}_{n|n-1}$ and $\boldsymbol{P}_{n|n-1}$. In contrast, the backpropagation step reverses the computational flow, starting from $\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}_{n|n-1}}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n-1}}$ and ultimately computing $\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}_{n-1|n-1}}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n-1|n-1}}$.

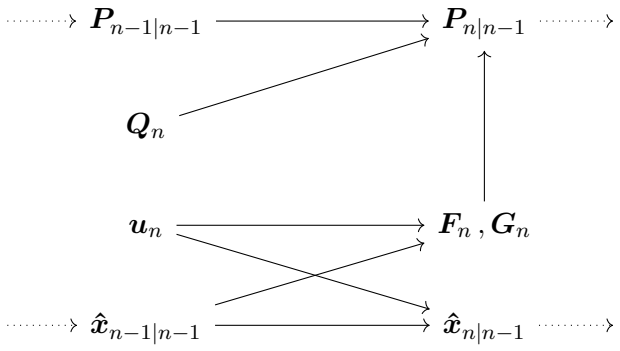

Figure 4.2: Computational graph of propagation step.

We begin by backpropagating the gradient of the covariance through the Riccati equation. This equation helps clarifying the concept of propagating gradients in reverse (i.e., backpropagation). Specifically, we start with the gradient of the covariance at time step $n|n-1$ and compute it for the previous step at $n-1|n-1$. For this equation, we detail the steps of the

calculation. First, we examine the effect of a change in $\boldsymbol{P}_{n-1|n-1}$ on the loss. A change to this covariance affects both the loss term l_{n-1} and the updated covariance $\boldsymbol{P}_{n|n-1}$. The contribution of each part is denoted with a superscript. Then using the chain rule, we sum the contributions of each term:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n-1|n-1}} = \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n-1|n-1}}\right)^{(l_{n-1})} + \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n-1|n-1}}\right)^{(\boldsymbol{P}_{n|n-1})} . \tag{4.14}$$

The gradient of each term is then computed using the matrix derivative equations given in (4.11). Finally, we obtain the following equation:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n-1|n-1}} = \frac{\partial l_{n-1}}{\partial \boldsymbol{P}_{n-1|n-1}} + \boldsymbol{F}_n^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n-1}} \boldsymbol{F}_n \,. \tag{4.15}$$

We also compute the gradient of the loss with respect to the process noise covariance matrix. In applications such as adaptive filtering, this gradient represents the quantity of interest:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{Q}_n} = \frac{\partial l_n}{\partial \boldsymbol{Q}_n} + \boldsymbol{G}_n^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n-1}} \boldsymbol{G}_n \,. \tag{4.16}$$

Next, the gradients are backpropagated through the error Jacobians:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{F}_n} &= \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n-1}} + \frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n-1}}^\top\right) \boldsymbol{F}_n \boldsymbol{P}_{n-1|n-1} \,, \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{G}_n} &= \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n-1}} + \frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n-1}}^\top\right) \boldsymbol{G}_n \boldsymbol{Q}_n \,. \end{aligned} \tag{4.17}$$

At this point, the gradient of the loss with respect to the control inputs $\boldsymbol{u}_n$ is computed. The gradient is computed component-wise for each element of the control inputs $\boldsymbol{u}_n$, with the superscript k denoting the specific component being computed. For some active sensing applications, this gradient is used for solving the optimization problem [Benhamou, 2024]:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial u_n^k} = (\boldsymbol{J}_n^u \boldsymbol{e_k})^\top \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{x}}_{n|n-1}} + \operatorname{Tr}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{F}_n}^\top \frac{\partial \boldsymbol{F}_n}{\partial u_n^k}\right) \\ + \operatorname{Tr}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{G}_n}^\top \frac{\partial \boldsymbol{G}_n}{\partial u_n^k}\right) . \end{aligned} \tag{4.18}$$

Finally, the gradient of the loss w.r.t the state at time $n-1|n-1$ is computed. Through this equation and equation (4.15), the propagation and update steps are interconnected in the backpropagation process:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{x}}_{n-1|n-1}^k} = \operatorname{Tr}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{F}_n}^\top \frac{\partial \boldsymbol{F}_n}{\partial \hat{x}_{n-1|n-1}^k}\right) + \frac{\partial l_{n-1}}{\partial \hat{x}_{n-1|n-1}^k} \\ + \operatorname{Tr}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{G}_n}^\top \frac{\partial \boldsymbol{G}_n}{\partial \hat{x}_{n-1|n-1}^k}\right) + (\boldsymbol{J}_n^x \boldsymbol{e_k})^\top \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{n|n-1}}} . \end{aligned} \tag{4.19}$$

4.2 Backpropagation Through the Update Step

The update step involves more computations than the propagation step. To simplify the derivation of equations, calculations are broken down into smaller steps, resulting in the computational graph shown in Figure 4.3.

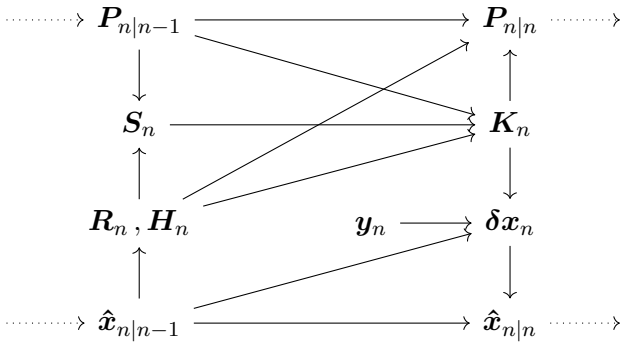

Figure 4.3: Computational graph of update step.

We first compute the gradient w.r.t the correction term $\boldsymbol{\delta x}_n = e\left(\hat{\boldsymbol{x}}_{n|n}, \hat{\boldsymbol{x}}_{n|n-1}\right)$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta x}_n} = \left(\boldsymbol{J}^{\varphi}_{\boldsymbol{\delta x}}\right)^{\top} \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{x}}_{n|n}}\,, \tag{4.20}$$

where $\boldsymbol{J}^{\varphi}_{\boldsymbol{\delta x}}$ is the retraction Jacobian w.r.t the correction (i.e., $\boldsymbol{J}^{\varphi}_{\boldsymbol{\delta x}} = \frac{\partial \varphi(\boldsymbol{x}, \boldsymbol{\delta x})}{\partial \boldsymbol{\delta x}}$). For a linear error, it reduces to the identity matrix. Using this, we backpropagate the gradient to obtain the gradient w.r.t the Kalman gain:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{K_n}} = -\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n}} \boldsymbol{P}_{n|n-1} \boldsymbol{H}_n^{\top} + \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta x}_n} \boldsymbol{z}_n^{\top}\,. \tag{4.21}$$

Next, we backpropagate through the computational graph to derive the gradient of the loss w.r.t the innovation covariance and w.r.t measurements.

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{S}_n} = \frac{\partial l_n}{\partial \boldsymbol{S}_n} - \boldsymbol{S}_n^{-1} \boldsymbol{H}_n \boldsymbol{P}_{n|n-1} \frac{\partial \mathcal{L}}{\partial \boldsymbol{K}_n} \boldsymbol{S}_n^{-1}\,, \tag{4.22}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_n} = \frac{\partial l_n}{\partial \boldsymbol{y}_n} + \boldsymbol{K}_n^{\top} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta x}_n}\,. \tag{4.23}$$

With these elements, we can compute the gradient w.r.t the observation covariance, a key factor in covariance tuning:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{R}_n} = \frac{\partial l_n}{\partial \boldsymbol{R}_n} + \frac{\partial \mathcal{L}}{\partial \boldsymbol{S}_n}\,. \tag{4.24}$$

Furthermore, the gradient of the loss w.r.t the Jacobian of the observation can be determined. This allows observation model learning and assessing the impact of an observation on the loss,

which is valuable for active sensing:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{H}_n} = \boldsymbol{S}_n^{-1} \frac{\partial \mathcal{L}}{\partial \boldsymbol{K}_n^\top} \boldsymbol{P}_{n|n-1} - \boldsymbol{K}_n^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n}} \boldsymbol{P}_{n|n-1} \\ + \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{S}_n} + \frac{\partial \mathcal{L}}{\partial \boldsymbol{S}_n}^\top \right) \boldsymbol{H}_n \boldsymbol{P}_{n|n-1} \,.\end{aligned} \tag{4.25}$$

Finally, we compute the gradient of the loss w.r.t the covariance before update and w.r.t the state before update.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n-1}} = \frac{\partial l_n}{\partial \boldsymbol{P}_{n|n-1}} + (\boldsymbol{I} - \boldsymbol{K}_n \boldsymbol{H}_n)^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{n|n}} \\ + \boldsymbol{H}_n^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{S}_n} \boldsymbol{H}_n + \frac{\partial \mathcal{L}}{\partial \boldsymbol{K}_n} \boldsymbol{S}_n^{-1} \boldsymbol{H}_n \,.\end{aligned} \tag{4.26}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{x}_{n|n-1}^k} = \operatorname{Tr}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{H}_n}^\top \frac{\partial \boldsymbol{H}_n}{\partial \hat{x}_{n|n-1}^k} \right) + \operatorname{Tr}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{R}_n}^\top \frac{\partial \boldsymbol{R}_n}{\partial \hat{x}_{n|n-1}^k} \right) \\ + \frac{\partial l_n}{\partial \hat{x}_{n|n-1}^k} - (\boldsymbol{K}_n \boldsymbol{H}_n \boldsymbol{e_k})^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta x_n}} \\ + (\boldsymbol{J}_{\boldsymbol{x}}^{\varphi} \boldsymbol{e_k})^\top \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{n}|\boldsymbol{n}}} \,.\end{aligned} \tag{4.27}$$

With this last equation, all the necessary elements are gathered to perform the recursive gradient computation. However, the first backpropagation step is different from the others since the gradients have to be first initialized. In our case, this means first computing $\frac{\partial l_N}{\partial \boldsymbol{P}_{N|N}}$ and $\frac{\partial l_N}{\partial \hat{\boldsymbol{x}}_{N|N}}$ for the loss and initializing $\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{\boldsymbol{N}|\boldsymbol{N}}} = \frac{\partial l_N}{\partial \boldsymbol{P}_{N|N}}$ and $\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{x}}_{N|N}} = \frac{\partial l_N}{\partial \hat{\boldsymbol{x}}_{N|N}}$. Once this has been done, we unroll the equations presented iteratively to compute the gradients from time n to time $n-1$.

5 Conclusion

This chapter has highlighted the advantages of using backpropagation to compute gradients in an EKF. It also presents the equations that enable its implementation. The main benefit of these equations is the acceleration of computations, which is crucial for embedded applications. Moreover, in contexts requiring certification, analytical equations are more easily verifiable than solely relying on automatic differentiation tools.

The derived equations extend the work of Parellier *et al.* [Parellier, 2023b] from a linear Kalman filter to an Extended Kalman Filter. Compared to the work of Wu *et al.* [Xu, 2024], we provide gradients of the loss with respect to a broader set of quantities of interest and for a more general loss function. For instance, the gradient of the loss with respect to the controls $\boldsymbol{u}_n$ is particularly important for active sensing, as it quantifies how a modification of a control input impacts uncertainties and is directly used in the optimization.

These backpropagation equations are a central tool in this work. We show that the loss gradient with respect to EKF parameters extends beyond covariance tuning [Parellier, 2023b; Xu, 2024], and that intermediate gradients in EKFNet are also useful in several applications.

Chapter 5

Minimum-Covariance Active Sensing with the Bicycle Model

Contents

1 Considered Problems **52**
1.1 Bicycle Model 52
1.2 Motivating Example for Active Sensing 53
1.3 Optimal Motion Planning Problem 54
1.4 Motion Planning Algorithm 56
1.5 Gradient Calculation via Analytical Backpropagation 57
2 Simulation Experiments **58**
2.1 Simulation Results 58
2.2 Computation Time 58
2.3 Discussion 61
3 Real-World Experiments **62**
3.1 Experimental Setting 62
3.2 Results 63
3.3 Analysis 64
4 Conclusion **65**

Résumé

Ce chapitre traite de la calibration extrinsèque des capteurs, en particulier de l'estimation du bras de levier entre capteurs. Une calibration imprécise peut en effet dégrader la fusion de mesures et donc les performances d'estimation. La méthode proposée s'intéresse à la manière dont les données de calibration sont acquises, car deux difficultés majeures peuvent apparaître : un manque d'observabilité et une propagation défavorable du bruit.

La méthode développée repose sur la perception active, dont l'objectif est de planifier des trajectoires d'un modèle de bicyclette maximisant l'information acquise. Cette information

est mesurée à travers la trace de la covariance d'un filtre de Kalman étendu (EKF), et les trajectoires optimales sont obtenues en résolvant un problème d'optimisation. Celui-ci exploite la rétropropagation analytique des gradients à travers l'EKF, ce qui accélère significativement le calcul des gradients nécessaires par rapport aux méthodes numériques classiques ou à la différentiation automatique (Autograd de PyTorch).

La méthode est enfin validée pour la calibration du bras de levier entre une antenne GNSS et le centre d'un modèle de véhicule. Une première validation est menée en simulation, suivie d'expériences réelles sur un véhicule tout-terrain équipé de capteurs GNSS et d'odomètres. Les résultats montrent que les trajectoires issues de la perception active permettent de réduire à la fois la trace de la covariance d'estimation et l'erreur réelle de localisation.

In robotics, many sensors enable the location and perception of the surrounding environment. When measurements are fused together, it is necessary to express all measurements within a common reference frame. To perform this coordinate transformation, one must know the sensor's extrinsic calibration parameters. These parameters typically include the lever arm (i.e., the relative position) and the relative orientation between the two sensor frames. If sensor calibration is inaccurate, fusion may degrade rather than improve estimation.

In general, sensor calibration follows a common procedure. It begins with a data acquisition campaign conducted under specific experimental conditions. Then, the calibration parameters are estimated by solving an optimization problem using the acquired data. A major challenge that can arise during calibration is insufficient excitation of the sensors (i.e., a lack of observability): if the acquired data do not sufficiently stimulate the system, the calibration problem remains underdetermined. It is also worth noting that, in some applications, calibration must be performed online, meaning that it takes place while the system is operating.

One way to address this issue is through the use of active sensing to improve the quality of the acquired data. Indeed, the goal of active sensing approaches is to determine trajectories that maximize the information gathered from sensors. In the context of calibration, active sensing approaches have been developed. For instance, [Preiss, 2018] proposed generating trajectories for optimal online calibration between a Global Navigation Satellite System (GNSS) and an inertial measurement unit (IMU), as well as for visual-inertial systems. Their approach relies on the maximization of an observability criterion. Similarly, [Xu, 2023] introduced a Fisher Information Matrix (FIM)-based optimization method to enhance extrinsic calibration. In the same spirit, this chapter presents an active sensing strategy for online sensor calibration.

Our active sensing approach adopts a tractable formulation based on the EKF. The EKF estimates in real time the state $\boldsymbol{x}_n$ from various sensor measurements and (approximately) conveys the associated level of uncertainty through the state error covariance matrix $\boldsymbol{P}_n$. The magnitude of $\boldsymbol{P}_n$ can then be used as an objective to minimize. Although the covariance matrix and the observability Gramian are related (see [Salaris, 2019] and the probabilistic interpretation of the observability Gramian in Chapter 4 Section 2), in this work, we focus on the covariance matrix, as it more realistically captures the noise characteristics [Rafieisakhaei, 2017]. Moreover, the covariance matrix is widely recognized in the literature as an effective criterion for active

sensing [Leung, 2006; He, 2008; Kontitsis, 2013; Patil, 2015; Hausman, 2015; Rafieisakhaei, 2017; Cognetti, 2018].

To derive trajectories that minimize a scalar function of $\boldsymbol{P}_n$, a first step is to compute its gradients with respect to the control inputs. Several approaches can be considered. One may use a brute-force approach (in the vein of [Abbeel, 2005]) or numerical differentiation. However, these techniques are often ill-conditioned and computationally expensive due to the complexity of the EKF equations. To overcome these limitations, [Patil, 2015] proposed using backpropagation through automatic differentiation and argued that deriving analytical expressions would be difficult.

In this chapter, we propose leveraging the backpropagation equations within the EKF, as introduced in Chapter 4, to solve the active sensing problem encountered during calibration. The main advantage of this approach is its ability to handle large-scale problems while significantly reducing computation time.

The remainder of this chapter is organized as follows. In Section 1, we introduce our motion planning formulation. The benefits of the proposed approach are then demonstrated through simulations in Section 2 and validated with real-robot experiments in Section 3.

1 Considered Problems

1.1 Bicycle Model

In this chapter, we consider the kinematic bicycle model. The system state consists of the orientation (heading) $\theta_n \in \mathbb{R}$ in 2D and, the position of the vehicle $\boldsymbol{p}_n \in \mathbb{R}^2$. The control inputs are the steering angle ν_n and, the forward velocity μ_n. The kinematic equations based on a roll-without-slip assumption are as follows:

$$\begin{cases} \theta_n = \theta_{n-1} + \dfrac{dt}{L}(\mu_n + w_n^{\mu}) \tan(\nu_n + w_n^{\nu}), \\ \boldsymbol{p}_n = \boldsymbol{p}_{n-1} + dt\Omega\left(\theta_{n-1}\right)(\mu_n + w_n^{\mu})\boldsymbol{e_1}, \end{cases} \tag{5.1}$$

where $\Omega(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$, L is the distance between the front and the rear wheels, $\boldsymbol{e_1} = (1, 0)^\top$ indicates that the velocity is aligned with the robot's heading, and dt is the sampling time. A Gaussian white noise $\boldsymbol{w}_n = (w_n^{\mu}, w_n^{\nu})^\top$ with covariance matrix $\boldsymbol{Q}$ corrupts the forward velocity μ_n and steering angle ν_n to account for actuators' imperfections, and the mismatch with idealized kinematic model, i.e., slip.

In this chapter, we consider that the dynamics of our system follow the bicycle model, and that the observation model measures the position using a GNSS. The estimation problem consists of estimating the state of the vehicle. There are two difficulties associated with this problem. First, the heading is not directly measured. Then, and more importantly, we assume that the position of the GNSS antenna in the robot's frame (we call lever arm) is unknown, or inaccurately known, or may slightly vary over time. The resulting problem pertains to simultaneous self-calibration and navigation, in the vein of [Preiss, 2018] but in a simpler context.

Letting $\boldsymbol{l}_n \in \mathbb{R}^2$ be the position of the GNSS antenna in the vehicle's frame w.r.t. its center (the midpoint of the rear axle), the position measured by the GNSS is

$$\boldsymbol{y}_n = \boldsymbol{p}_n + \Omega(\theta_n)\boldsymbol{l}_n + \boldsymbol{v}_n \tag{5.2}$$

where $\boldsymbol{v}_n$ is a 2D white noise with covariance matrix $\boldsymbol{R}_n$.

Although the bicycle example may appear relatively simple, it is nonetheless a nonlinear system that effectively illustrates the benefits of active sensing for calibration. Moreover, because the problem is low-dimensional, it is possible to directly optimize the control inputs to improve localization. Finally, the bicycle model remains a relevant representation of ground vehicles when their dynamics are not aggressive.

1.2 Motivating Example for Active Sensing

The nonlinear nature of the bicycle model directly affects its observability. Indeed, for nonlinear systems, observability is not necessarily a global property, as it is for linear systems. Instead, different "levels" of observability exist, including the notion of local observability [Hermann, 1977]. Local observability refers to the ability to estimate small perturbations of the initial state around a reference trajectory.

For the bicycle with a lever arm, this phenomenon is clearly illustrated in Figure 5.1 with vehicle length $L = 2\,\text{m}$. In this figure, two bicycle (one in green and one in orange) have different centers of mass and lever arms but share the same initial orientation. Both vehicles move in a straight line, which results in identical GNSS measurements. Consequently, from GNSS data alone, we cannot distinguish between the two setups (i.e., the system is not observable). However, once the vehicles start turning, the GNSS measurements differ, allowing us to identify which configuration we are in. From this illustration, we conclude that, in the case of lever arm calibration, the trajectories must sufficiently excite the system. Therefore, active sensing trajectories should lead to more accurate robot state estimation.

The second point of interest in the bicycle model lies in how process noise propagates along the trajectory. Indeed, the propagation of noise depends directly on the vehicle's path. To illustrate this phenomenon, we conducted two Monte Carlo simulations, each with 10^3 particles. The first simulation corresponds to a constant forward velocity with zero angular velocity, while the second involves the same constant forward velocity with a nonzero angular velocity. The particles were initialized according to $\mathcal{N}\left(\mathbf{0_3}, \text{diag}(10\text{ deg}, 0\text{ m}, 0\text{ m})^2\right)$, and the process noise was drawn from normal distribution with covariance matrix $\boldsymbol{Q} = \text{diag}(5\text{ deg}, 0.1\text{ ms}^{-1})^2$. The final positions of the particles are shown in Figure 5.2. In the straight-line case, we observe the well-known "banana-shaped" distribution frequently encountered in robotics [Long, 2013]. Furthermore, the shape of the resulting particle cloud depends on the nominal trajectory. This observation highlights the importance of using active sensing to select trajectories that shape the uncertainty distribution in a desirable way.

Considering both process and measurement noise in the active sensing planning is therefore crucial. This motivates the use of the covariance matrix as an information measure rather than

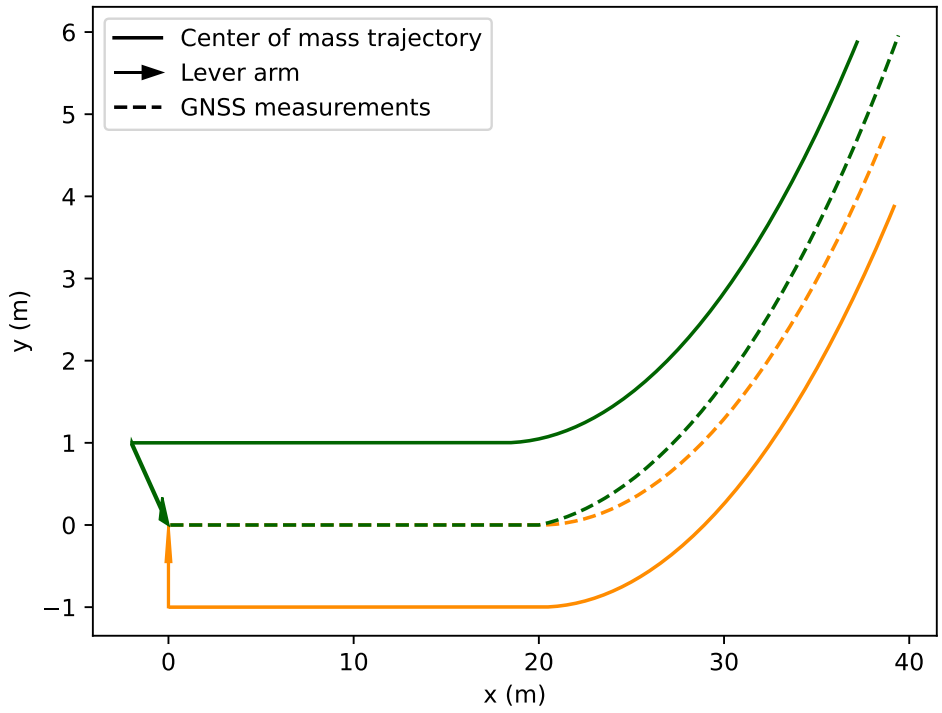

Figure 5.1: Two vehicles with different initial states (position and lever arm). When moving in a straight line, the GNSS measurements are identical, and the system is not observable. During a turn, however, the GNSS measurements differ, enabling us to distinguish between the orange and green configurations.

classical observability-based criteria, as already emphasized in [Rafieisakhaei, 2017].

1.3 Optimal Motion Planning Problem

Building upon the notations and equations introduced in Chapter 4, this section presents the optimization problem for calibration. Let $\mathcal{L}$ denote a scalar function that quantifies the magnitude of its argument and N the time horizon of the planning. The system state $\boldsymbol{x} = (\theta, \boldsymbol{p}, \boldsymbol{l})$ consists of its orientation and position in the fixed frame, as well as the lever arm expressed in the body frame. The resulting minimum-uncertainty (or active sensing) motion planning problem can then be expressed as

$$\begin{cases} \min\limits_{\{\boldsymbol{u}_i\}_{i=1}^{N}, \{\boldsymbol{x}_i\}_{i=0}^{N}} \mathcal{L}\left(\boldsymbol{P}_{N|N}\right) \\ \text{subject to} \\ \quad \forall n \leq N \ \boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n, \boldsymbol{0}), \quad \boldsymbol{x}_0 = \boldsymbol{x}^{\boldsymbol{init}}, \\ \quad \forall n \leq N \ \boldsymbol{P}_{n|n-1} = \boldsymbol{F}_n \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_n^\top + \boldsymbol{G}_n \boldsymbol{Q}_n \boldsymbol{G}_n^\top, \\ \quad \forall n \leq N \ \boldsymbol{P}_{n|n} = (\boldsymbol{I} - \boldsymbol{K}_n \boldsymbol{H}_n) \, \boldsymbol{P}_{n|n-1}, \\ \quad \forall n \leq N \ \boldsymbol{u}_{\mathbf{min}} \leq \boldsymbol{u}_n \leq \boldsymbol{u}_{\mathbf{max}}. \end{cases} \tag{5.3}$$

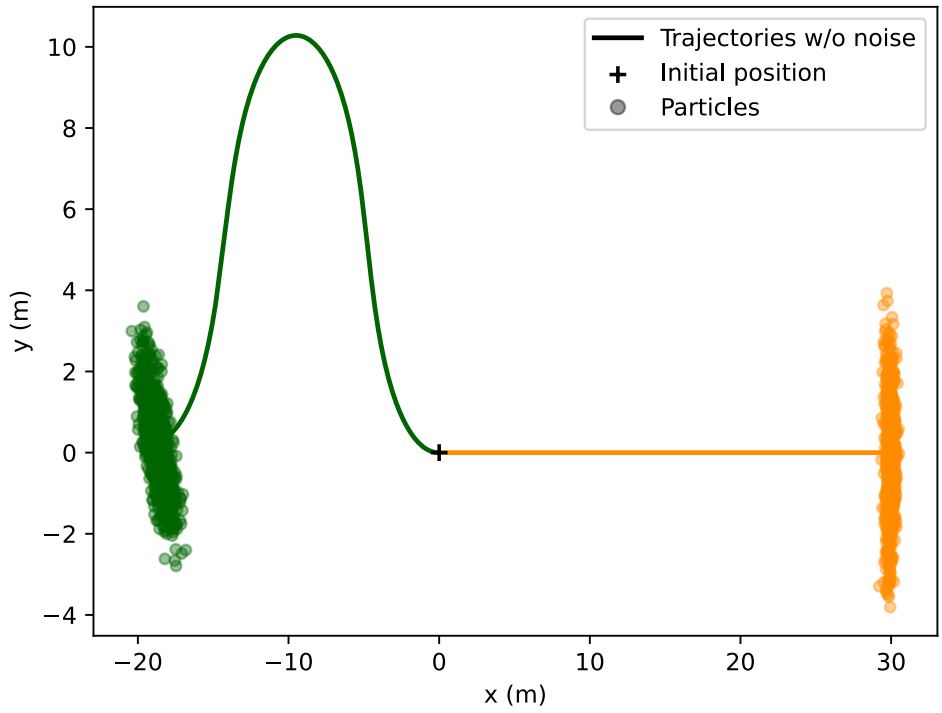

Figure 5.2: Monte Carlo simulation of process noise propagation for two trajectories: one straight (right) and one turning (left). The shape of the particle distribution differs significantly, illustrating the influence of trajectory choice on uncertainty propagation.

The most common choices for $\mathcal{L}$ to reflect the final uncertainty are the trace $\mathcal{L} = \mathrm{Tr}(\boldsymbol{P}_{N|N})$, used in [Patil, 2015; Rafieisakhaei, 2017; Preiss, 2018], or the maximum eigenvalue, which has to be regularized using Schatten's norm, see [Salaris, 2019]. As the trace of $\boldsymbol{P}_{N|N}$ sums the diagonal terms, which are variances expressed in possibly different units and choice of scales, we renormalize the matrix, as suggested in [Preiss, 2018], leading to the objective $\mathcal{L} = \mathrm{Tr}\left(\boldsymbol{P}_0^{-1}\boldsymbol{P}_{N|N}\right)$.

The difficulty of the optimal control problem above stems from the complicated relationship between the control inputs $\{\boldsymbol{u}_i\}_{i=1}^{N}$ and the final covariance matrix $\boldsymbol{P}_{N|N}$. In these equations, matrices $\boldsymbol{F}_n$, $\boldsymbol{G}_n$ and $\boldsymbol{H}_n$ are all Jacobians that depend on state estimation $\hat{\boldsymbol{x}}_n$ and control inputs $\boldsymbol{u}_n$:

$$\boldsymbol{F}_n = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{n-1|n-1},\boldsymbol{u}_n,\boldsymbol{0}}, \quad \boldsymbol{G}_n = \left.\frac{\partial f}{\partial \boldsymbol{w}}\right|_{\hat{\boldsymbol{x}}_{n-1|n-1},\boldsymbol{u}_n,\boldsymbol{0}}, \quad \boldsymbol{H}_n = \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{n|n-1}}. \tag{5.4}$$

Through these Jacobians, the control inputs affect the covariance matrices, which is in contrast with the linear case where all trajectories are equivalent. It thus makes sense to compute the sensitivity of covariance matrices with respect to control inputs.

Our optimization problem follows a *partial collocation* formulation, as in [Patil, 2015]. In this approach, the system states are explicitly included as optimization variables, while the covariance is computed implicitly. Consequently, the constraints arising from covariance propagation

and the Kalman update are handled implicitly. This significantly reduces the computational complexity compared to a *full collocation* formulation [Patil, 2015], where the decision variables include states, control inputs, and covariance matrices. Moreover, this formulation achieves faster convergence than a *shooting* formulation [Patil, 2015], in which only the control inputs are treated as decision variables.

1.4 Motion Planning Algorithm

The optimization problem formulated in (5.3) has to be solved while accounting for both computational time constraints and the quality of the solution. Leveraging backpropagation equations of Chapter 4, we employ a first-order optimization algorithm, summarized by Algorithm 1. From a computational perspective, the gradient computation stands out as the most computationally demanding step, hence the interest for the method developed above. To solve the nonlinear constrained optimization problem, methods such as Sequential Quadratic Programming (SQP) or interior-point algorithms can be employed [Nocedal, 1999].

Algorithm 1 Motion planning algorithm

Require: $\boldsymbol{x}_0,\ \boldsymbol{P}_0,\ N, \mathcal{L}$
  $\boldsymbol{u} \leftarrow \{\boldsymbol{u}_i\}_{i=1}^{N}$
  while $\mathcal{L}(\boldsymbol{P}_{N|N})$ not converge **do**
    $\boldsymbol{g} \leftarrow$ gradient_computation$(\boldsymbol{u})$
    $\alpha \leftarrow$ line_search$(\boldsymbol{u}, \boldsymbol{g})$
    $\boldsymbol{u} \leftarrow$ update$(\boldsymbol{u}, \boldsymbol{g}, \alpha)$
  end while
  Return: $\{\boldsymbol{u}_i\}_{i=1}^{N}$

The algorithm involves computing the gradient $\boldsymbol{g}$ of the cost function w.r.t. the control variables using forward and backward passes through the EKF of Chapter 4. At each iteration, the filter is executed in forward mode, after which the gradients are computed in backward mode. The control inputs are then updated by the optimizer, and the forward-backward computation is repeated. The following section provides further details on the computation method. Additionally, we compute the gradient of the constraints with respect to decision variables. Subsequently, line search is conducted to determine an appropriate step size α for efficient convergence. However, line search requires evaluating the cost function, which corresponds to running a full EKF in our case.

While our formulation aims to find trajectories that minimize the final uncertainty on the robot state over a given horizon, it can easily be extended to include additional constraints or task-specific objectives. For instance, adding the constraint $\boldsymbol{x}_N = \boldsymbol{x}^{\text{end}}$ enforces reaching a desired final state while still enabling active sensing. The gradient of such constraints with respect to the decision variables is usually straightforward to compute. Similarly, it is simple to make $\mathcal{L}$ depend on intermediate covariances $\boldsymbol{P}_{n|n}$, for $n < N$, as in [Preiss, 2018]. In our calibration problem, however, the loss only depends on the final covariance since we are primarily concerned with the uncertainty at the end of the trajectory.

The challenging aspect of the optimization lies in computing the gradient of the loss with respect to the decision variables. To reduce computational complexity, we implicitly account for the dependence of the state in the loss and use analytical backpropagation equations to compute the gradient of the loss w.r.t. input controls.

1.5 Gradient Calculation via Analytical Backpropagation

The goal of our problem is to derive optimal controls $\boldsymbol{u}_1, \dots, \boldsymbol{u}_N$ that minimize loss $\mathcal{L}(\boldsymbol{P}_{N|N})$. Since this motion planning is performed offline, the (noisy) observations $\boldsymbol{y}_n$ in (4.5) are not available. In order to perform forward execution of the EKF, the most reasonable choice is to plan using the *a priori* value of the $\boldsymbol{y}_n$, i.e., $\boldsymbol{y}_n = h(\hat{\boldsymbol{x}}_{n|n-1})$. Therefore, we simplify the notation by writing $\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} := \hat{\boldsymbol{x}}_n$ and $\hat{\boldsymbol{x}}_{n-1|n-1} := \hat{\boldsymbol{x}}_{n-1}$. In dual control theory, this assumption is known as the certainty equivalence principle, which considers only the mean of random variables which correspond to the same assumption made here. In the context of active sensing, this assumption is sometimes referred to as the maximum likelihood hypothesis.

Once the forward pass of the EKF has been executed, we can start backpropagating the gradients. As explained at the end of Section 4.2, the first step is to initialize the gradients. In our case, to initialize the backward process, one needs to compute $\frac{\partial \mathcal{L}(\boldsymbol{P}_{N|N})}{\partial \boldsymbol{P}_{N|N}}$. This quantity depends on the chosen loss function.

In the case of the normalized trace, we have [Petersen, 2012]:

$$\frac{\partial \mathcal{L}(\boldsymbol{P}_{N|N})}{\partial \boldsymbol{P}_{N|N}} = \frac{\partial \mathrm{Tr}\left(\boldsymbol{P}_0^{-1}\boldsymbol{P}_{N|N}\right)}{\partial \boldsymbol{P}_{N|N}} = \boldsymbol{P}_0^{-1}. \tag{5.5}$$

When the objective is to minimize the largest eigenvalue of $\boldsymbol{P}_{N|N}$, the loss function must be defined accordingly. Since this objective is non-differentiable, we use its regularized version based on the Schatten norm, as proposed in [Salaris, 2019]. For a smoothness parameter s, the Schatten norm is defined as:

$$||\boldsymbol{P}||_s = \sum_i \lambda_i(\boldsymbol{P})^s = \sum_i \left(\boldsymbol{v_i}^\top \lambda_i \boldsymbol{v_i}\right)^s, \tag{5.6}$$

where λ_i are the eigenvalues of the covariance matrix and $\boldsymbol{v_i}$ the associated normalized eigenvectors. When $s \to +\infty$, the loss converges to the largest eigenvalue of the covariance matrix. Such a loss was in fact already used in [Mehra, 1974]. The gradient of the Schatten norm with respect to $\boldsymbol{P}$ is given by:

$$\frac{\partial ||\boldsymbol{P}||_s}{\partial \boldsymbol{P}} = \left(\sum_i \lambda_i^s\right)^{\frac{1-s}{s}} \left(\sum_i \lambda_i^{s-1} \boldsymbol{v_i}\boldsymbol{v_i}^\top\right). \tag{5.7}$$

Starting from the initialization at time step $N|N$, we compute the gradient at time $N|N-1$ using the backpropagation equations. Continuing this process backward yields the derivatives with respect to all control inputs. This approach leads to a significant computational speed-up, as the backward equations provide the derivatives with respect to *all* control inputs in a

single pass. In contrast, forward propagating perturbations (i.e., finite differences) would require running an entire EKF-like process from scratch for each control input $\boldsymbol{u}_n$. Similar to dynamic programming, backpropagation allows reusing previous computations at each step, resulting in substantial efficiency gains.

2 Simulation Experiments

2.1 Simulation Results

For bicycle model simulation, we take $L = 4$ m and, $dt = 1$ s. In terms of noise parameters, we take $\boldsymbol{Q} = \text{diag}(0.1, \pi/180)^2$, and $\boldsymbol{R}_n$ be the identity matrix, i.e., a standard deviation of 1 m for the GNSS position measurements. To account for actuator physical limits, we assume $|\nu_n| \leq 30\pi/180$ rad and $0 \leq \mu_n \leq 5$ m.s^{-1}. To account for acceleration limits, we add the constraints $|\Delta\nu_n| \leq 15\pi/180$ rad.s^{-1} and $|\Delta\mu_n| \leq 1$ m.s^{-2}.

We start by sampling admissible control inputs over a horizon $N = 150$s. Then, the corresponding trajectory is obtained by integration. We then use the sequential least-squares programming (SLSQP) algorithm from Scipy [Virtanen, 2020] to optimize the sequence of controls to apply. The calculation of the gradient of the loss with respect to the control inputs is performed using analytical backpropagation equations. An example of the initial random trajectory and the solution of the active sensing problem can be found in Figure 5.3. The optimal trajectory for the Schatten norm and for the trace are quite similar in this case, and both oscillate around the initial random trajectory. These oscillations are maneuvers that increase the observability of the lever arm as we previously explained (see Section 1.2). They lead to an actual reduction of the expected error covariance.

To demonstrate that the final covariance minimization translates into an actual reduction of the average state estimation error, we simulate 200 trials of each optimal trajectory by adding process noise and observation noise. During the simulation, the state is estimated with an EKF based on (5.1), (5.2). The evolution of the absolute estimation error of the lever arm is displayed on Figure 5.4. For active sensing trajectories, the error decreases and converges more rapidly, illustrating the benefit of active sensing optimal trajectory generation.

We then simulate each trajectory 20 times. For each state variable, the final Mean Absolute Error (MAE) is calculated. We repeat this process for 10 initial random trajectories. The results are presented in Figure 5.5. The active sensing trajectories yield the lowest MAE for almost all state variables. However, the heading MAE appears largely insensitive to the chosen method. This behavior, previously noted by [Salaris, 2019], stems from the different magnitudes of the state variables, which our normalization does not fully compensate for.

2.2 Computation Time

We compared the computation times for the gradient of the loss w.r.t. control inputs with three different methods. The first method uses (forward) finite differences to compute the gradient, as in [Indelman, 2016; Preiss, 2018]. The second method is an automatic differentiation method,

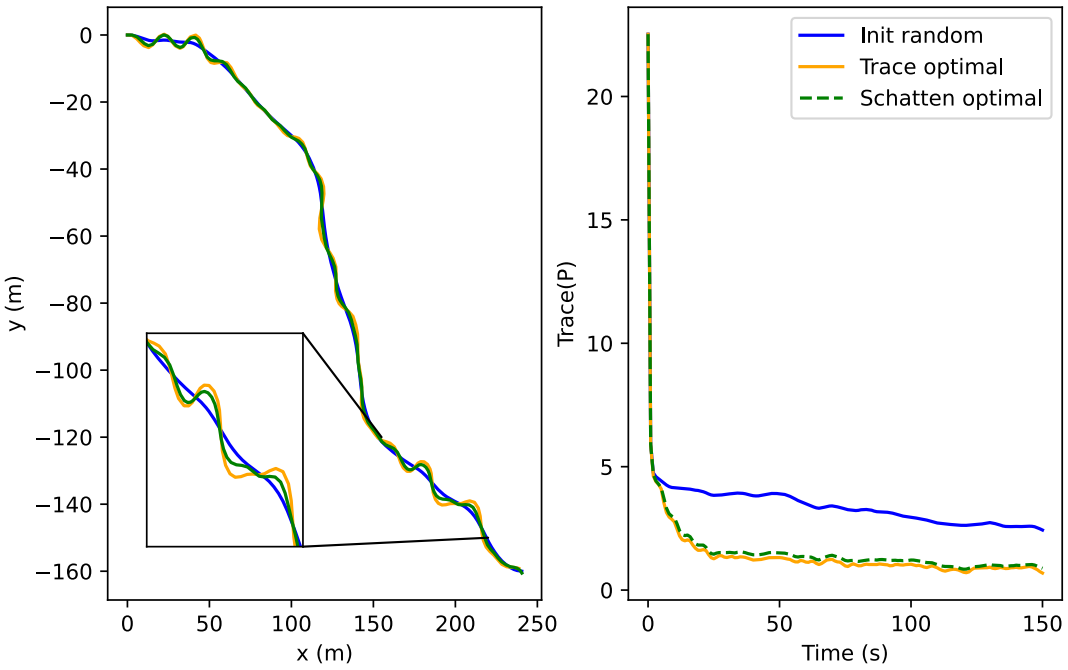

Figure 5.3: On the left, an example of an initial random guess in blue. The other two trajectories are solutions to the active sensing problem where the loss is the trace (in orange) and the Schatten norm (in green). The right plot shows the expected trace of the covariance evolution for each trajectory.

which adopts the backpropagation paradigm, but through a numerical tool, akin to [Patil, 2015]. Namely, we used the state-of-the-art PyTorch automatic differentiation (AD) [Paszke, 2019]. Finally, the last method uses our backpropagation analytical formulas of Chapter 4. The code was executed on computer with an Intel Core i5-1145G7 at 2.60 GHz.

Table 5.1: Average and standard deviation of gradient calculation time (over 100 calculations) using different methods.

Method	Execution time
Finite differences	26.92 ± 8.45s
PyTorch Autograd	0.55 ± 0.19s
Analytical backpropagation	**0.19** ± 0.07s

Table 5.1 shows that gradient calculation using backpropagation (Autograd and ours) leads to large speedups. Indeed, when using a finite difference based method, optimization is computationally expensive, as mentioned in [Preiss, 2018], where optimizing the trace of the sum of all covariances is reported to take 13 hours for a 3D inertial navigation system. Even when compared to state-of-the-art PyTorch Autograd, our closed-form formulas are much faster, and more stable in terms of variability of computation time.

The total computation time to solve the optimization problem (5.3) using our gradient calculation method and Algorithm 1 is 351s. Although this perfectly suits off-line trajectory generation, it means the robot should stop for a few minutes to plan in a real-time context.

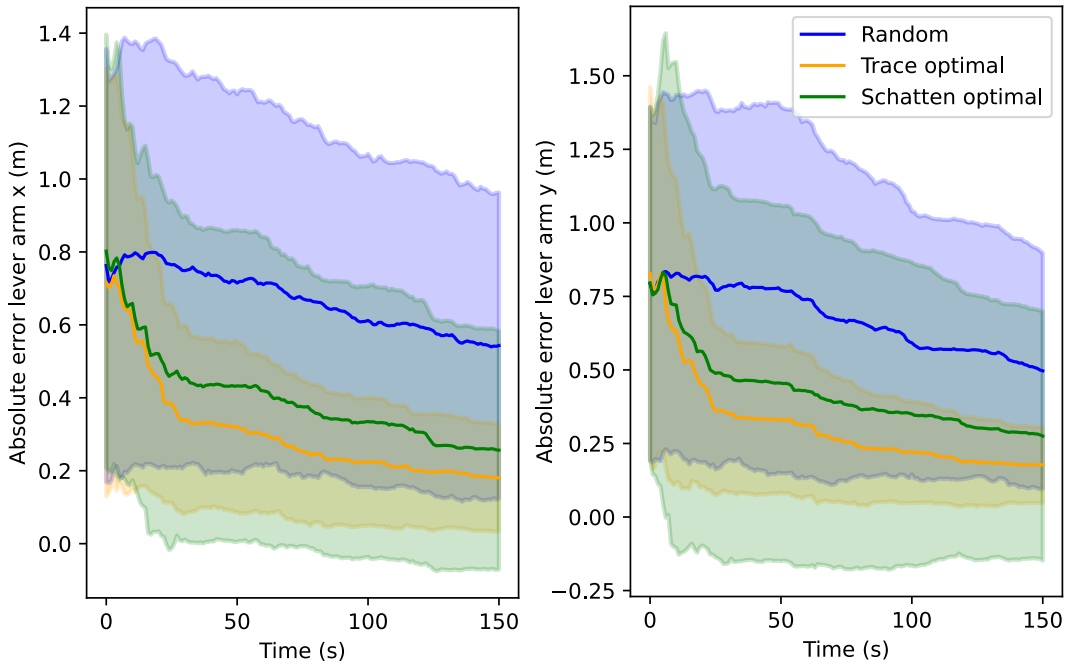

Figure 5.4: Absolute estimation error of the lever arm during the trajectory. On the left the error for the lever arm in x and on the right in y. One-σ envelope illustrates the dispersion of errors over trials. Although the error is positive, the envelope may take negative values since it represents the mean error $\pm$ one standard deviation.

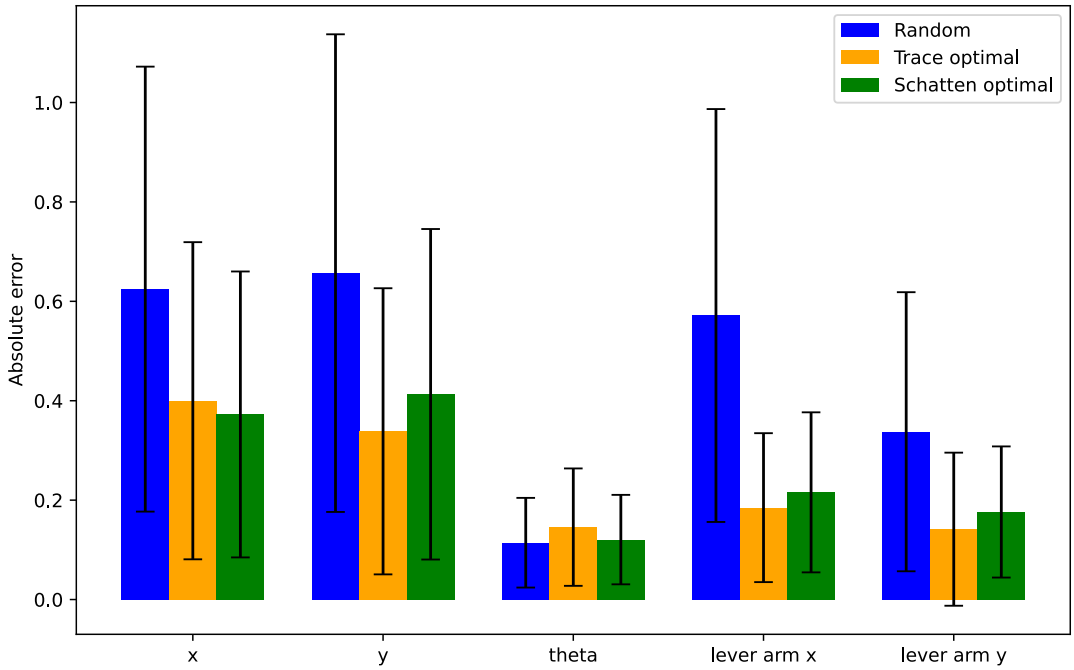

Figure 5.5: Final MAE for 10 different random trajectories with 20 trials (in blue) and optimal trajectories where the loss is either the trace (in orange) or the Schatten norm (in green).

2.3 Discussion

A few remarks are in order. First, we see that, by replacing state-of-the-art *Autograd* differentiation with our formulas, one may (roughly speaking) double the planning horizon for an identical computation budget. Besides, analytical formulas better suit onboard implementation. It is interesting to note that they are totally akin to the EKF equations, which must be implemented on the robot anyway. Finally, analytical formulas (when available) may be preferable to numerical methods, as one keeps a better control over what is being implemented, possibly opening up for some guarantees (the behavior of *Autograd* may be harder to anticipate, and leads to higher computation time variability). Moreover, we anticipate that coding them in C++ may lead to further speedups.

Note that the overall computation time of the motion planning algorithm could be significantly reduced. First, we may obviously optimize over a shorter horizon. Then, the optimizer currently uses a line search algorithm to determine the descent step size. In our case, evaluating the objective function is computationally expensive because it requires running the EKF. Therefore, finding a method that reduces the number of calls to the objective function should prove efficient. Finally, another option to reduce computation time is to decrease the number of decision variables, by for instance parameterizing trajectories using B-splines, as in [Salaris, 2019].

We may also comment on the obtained trajectories. As the optimization problem is highly nonlinear, non-convex, and constrained, one should expect an optimization method to fall into a close-by local minimum. In Figure 5.3, the local nature of the optimum proves visible, as the obtained trajectory oscillates around the initial trajectory. Methods to step out of local minima go beyond the scope of this paper. However, it is worth noting that albeit a (close-by) local minimum, the obtained trajectory succeeds in much reducing state uncertainty. It reduces the final average error on the lever arm, which is the most difficult variable to estimate, by a factor 3, see Figure 5.4.

In the context of real-time online planning, this suggests a sensible way to use the formulas of the present paper would be to compute a real-time trajectory that optimizes a control objective, and then to refine it in real time, by performing a few gradient descent steps. This shall increase the information gathered by the sensors.

As discussed in Section 1.2, one of the main difficulties in lever arm calibration lies in the issue of observability. To verify that our strategy effectively improves the local observability of the system, we compute, for each trajectory, the evolution of the smallest eigenvalue of the local observability Gramian. When the local observability Gramian is rank-deficient, the system is not locally observable.

As shown in Figure 5.6, the results show that active sensing enhances observability. In particular, for the initial random trajectory, the smallest eigenvalue remains close to zero for nearly 90 s, indicating poor local observability. In contrast, the trajectories optimized through active sensing maintain significantly higher minimum eigenvalues. This demonstrates experimentally that reducing the state covariance leads to an improvement in the system's local observability.

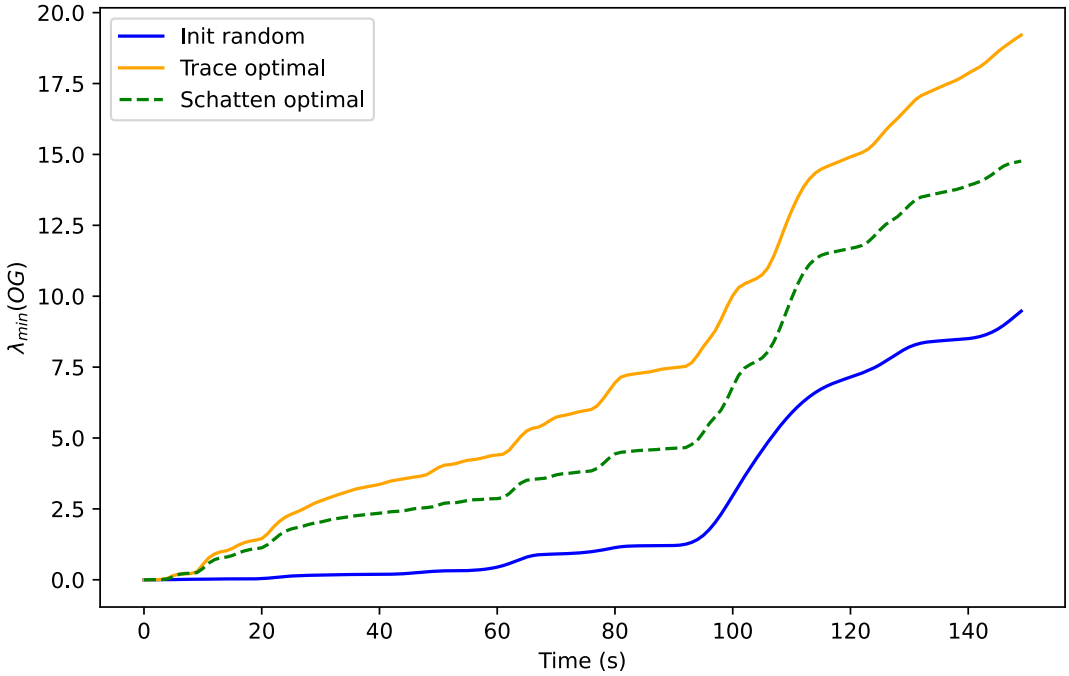

Figure 5.6: Evolution of the smallest eigenvalue of the local observability Gramian for active sensing trajectories (green and orange) and the initial random trajectory (blue). The result shows that active sensing improves the system's local observability.

3 Real-World Experiments

Real experiments were conducted jointly with the company Safran. With the help from Safran engineers, we used an experimental off-road car owned by the company, which is approximately 4m long and 2.1m wide[1].

3.1 Experimental Setting

The vehicle is equipped with a standard GNSS, odometers, and a RTK (Real Time Kinematic) GNSS, which is not used by the EKF, but serves as ground-truth for position owing to its high accuracy. The lever arm between the GNSS and the RTK is denoted by $l_{GNSS/RTK}$ has been pre-calibrated and is only used as a comparison to the ground-truth.

To further test our method, we conducted localization experiments on both ordinary and active sensing trajectories and compared their performance. We used the model described in Section 1.1, and devised an EKF that fuses odometer data (kinematic model) with GNSS position measurements. The vehicle state is represented by a 5-dimensional vector: two dimensions for position, one for the orientation, and two for the lever arm between the GNSS and the center of the vehicle frame (midpoint of the rear axle).

Initially, we generated ordinary trajectories at two different speeds (5 km/h, 10 km/h). Subsequently, we employed our active sensing motion planning Algorithm 1 initialized with

[1]Because of confidentiality requirements, the company has not wished to publish a picture of its experimental vehicle.

those trajectories, to address optimization problem (5.3), using the final covariance trace as the loss. The constraints used were those relative to the actuators. To demonstrate the ability of optimization-based planning methods to handle further constraints, we also constrained the distance between the initial trajectory and the active sensing trajectory to be less than 1.5m. Then, we used local tangent plane coordinates computed near the experiment locations to convert 2D planning to world frame coordinates. Then, we followed each trajectory and computed the localization error committed by the EKF.

To measure localization accuracy, we utilized the RTK-GNSS system as a ground-truth, since its uncertainties are of the order of a few centimeters. The position error of the vehicle was computed as follows:

$$e_n = \left\| \boldsymbol{p}_n^{\boldsymbol{RTK}} - (\hat{\boldsymbol{p}}_n + \Omega(\hat{\theta}_n)(\hat{\boldsymbol{l}}_n + \boldsymbol{l}_{\boldsymbol{GNSS/RTK}})) \right\| \tag{5.8}$$

3.2 Results

The off-road experiment was performed in a field. It consists of 4 runs: 2 reference runs and 2 active sensing runs (at 5km/h and 10 km/h). Trajectories returned by RTK-GNSS over the first run are displayed in Figure 5.7. We note the oscillations of the active sensing trajectory around the reference trajectory, enhancing state observability.

Figure 5.8 shows the trajectories for the second run, along with the evolution of the trace of $\boldsymbol{P}_{n|n}$ output by the EKF (on the first run, the latter is wholly similar). We see the improvement in terms of trace through the optimized trajectories. The visible wiggling of the covariance corresponds to the correction steps of the EKF. Indeed, the dynamical model is run at 100Hz, corresponding to the odometer frequency, while corrections are made when GNSS data are available, at approximately 1Hz.

Table 5.2: Estimation error according to (5.8) for all trajectories

Speed (km/h)	Type	Mean error (m)
5	Reference	3.32
	Active sensing	**2.88**
10	Reference	4.73
	Active sensing	**4.34**

Although the trace of the covariance is smaller along the active sensing trajectory, it doesn't necessarily guarantee that the error with respect to ground truth is reduced. The uncertainty calculated by the filter assumes all random variables are Gaussian, and the EKF is based on approximations. For each trial, we calculated the average error obtained for each type of trajectory. The results, presented in Table 5.2, point out that active sensing trajectories enhance localization accuracy. Indeed, in the scenario at 5 km/h, the error decreases by 13.24% between the reference and the active sensing trajectory, and by 8.25% in the scenario at 10 km/h.

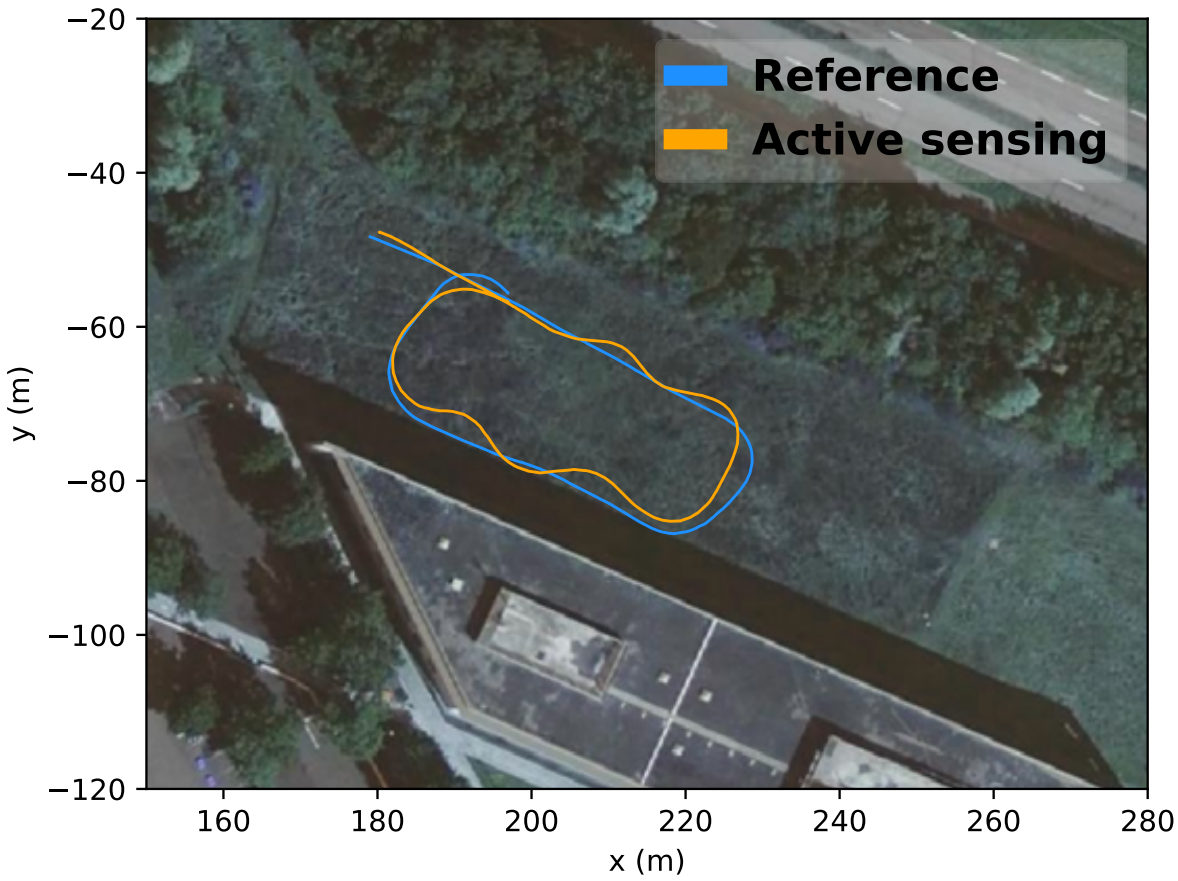

Figure 5.7: Off-road trajectories of the RTK-GNSS in a local tangent plane coordinates oriented East-North-Up of the scenario at 5 km/h.

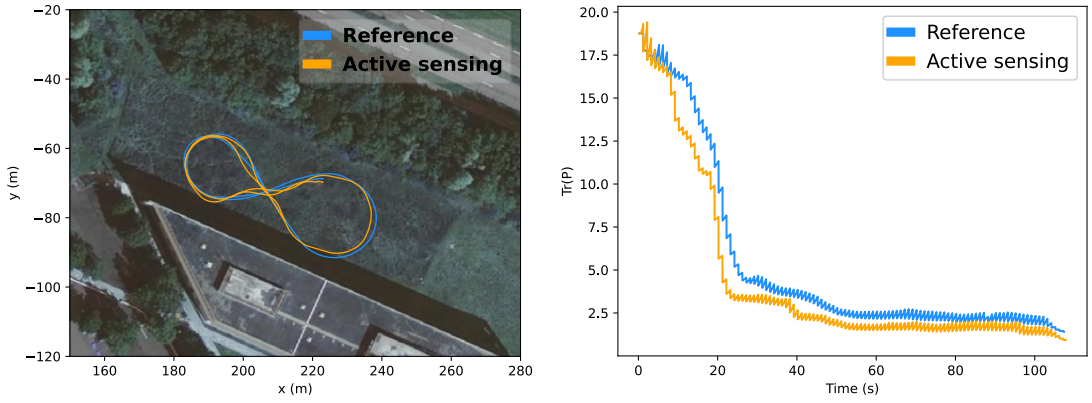

Figure 5.8: On the left, trajectories of the RTK-GNSS in local tangent plane coordinates at 10 km/h. On the right, the evolution of the covariance trace along each trajectory in the scenario at a speed of 10 km/h.

3.3 Analysis

When compared with results from simulations, the improvements are smaller. Several factors may explain this difference. Firstly, in simulations, we calculated an average result using the

Monte Carlo method, whereas here we have a single trial. Additionally, in simulations, model and noises are perfectly known, whereas in reality, they are not. The roll-without-slip assumption is especially challenged when using an off-road vehicle. This also makes the trajectory tracking quite imperfect. Finally, due to the limited area, we explicitly constrained the optimized trajectory to stay around the reference one, producing a trajectory that optimally "refines" the reference trajectory, limiting the accuracy increase.

4 Conclusion

Through the example of extrinsic lever-arm calibration between the GNSS antenna and the center of the kinematic model, we addressed an active sensing problem. Specifically, we leveraged the backpropagation equations introduced in Chapter 4 to perform active sensing motion planning, and validated the proposed approach both in simulation and in real-world experiments over large off-road trajectories exceeding 50 meters. Furthermore, we demonstrated that the analytical gradient computation method provides a significant computational advantage, being approximately twice as fast as PyTorch's *Autograd*.

While the current optimization problem formulation explicitly accounts for the vehicle's dynamic constraints, this approach may become computationally inefficient as the problem dimension increases. This observation motivates the development of a more suitable formulation capable of handling higher-dimensional active sensing problems more efficiently and extending to a broader range of applications. It is primarily this observation that motivates the approach developed in the next chapter.

Chapter 6

Minimum-Covariance Active Sensing with Quadrotor Models

Contents

1 Minimum Covariance Active Sensing Method **68**
- 1.1 Overall Architecture 68
- 1.2 Quadrotor Control Subsystem 68
- 1.3 Details About Planning Ahead of Time 71
- 1.4 Final Optimal Control Problem 71
- 1.5 Backprop-based Optimization Scheme 72

2 Application of Active Sensing to Collision Avoidance **72**
- 2.1 Dynamic Collision Avoidance with Bearing Measurements 73
- 2.2 Results 74
- 2.3 Comparisons with Prior Metrics 75

3 Application to IMU-GNSS fusion **77**
- 3.1 Enhancing IMU-GNSS Localization via Active Sensing 77
- 3.2 Results 79
- 3.3 Gradient Computation Time 80

4 Discussion **82**
- 4.1 Global Optimality Analysis 82
- 4.2 Trade-Off Between Smoothness and Observability 83

5 Conclusion **84**

Résumé

Ce chapitre propose une formulation générique de perception active pour les quadrotors. La méthode est fondée sur la théorie des systèmes plats, qui garantit la faisabilité dynamique des trajectoires, ainsi que sur la paramétrisation des sorties plates par B-splines. Le problème de perception active consiste à minimiser une fonction scalaire des covariances calculées par un filtre de Kalman étendu (EKF), et sa résolution s'appuie sur le calcul

des gradients par rétropropagation à travers l'EKF. Une analyse des temps de calcul montre par ailleurs que la bibliothèque Python JAX est particulièrement performante dans certaines configurations.

Deux applications illustrent la polyvalence du cadre proposé. La première concerne l'évitement d'un drone intrus à partir de mesures angulaires uniquement : la méthode retrouve des trajectoires hélicoïdales connues pour améliorer l'observabilité et donc la précision de la localisation de l'intrus. La seconde application porte sur l'amélioration de la fusion inertie-GNSS pour l'estimation de l'attitude. Pour cette application, des manœuvres spécifiques comme *la baïonnette* ou l'accélération axiale sont nécessaires pour mieux observer les biais des capteurs. La méthode proposée permet de retrouver ces mouvements. Ces deux exemples mettent en évidence des situations fondamentalement différentes : dans le premier, le système contrôlé diffère du système estimé, tandis que dans le second il s'agit du même système.

Les résultats obtenus en simulation montrent des performances au moins équivalentes, et souvent supérieures, aux approches existantes, tout en offrant un temps de calcul réduit grâce à la rétropropagation analytique ou automatique des gradients dans l'EKF.

In this chapter, we propose a unifying, versatile, and simple computational framework for optimal active sensing about a planned trajectory for differentially flat systems [Fliess, 1995], using an EKF's covariance matrix as metric [Patil, 2015]. In spite of the previously acknowledged computational complexity associated with this metric [Preiss, 2018], we make the optimization scheme tractable by leveraging backpropagation of gradients through the estimation and control pipeline enabling gradient-based optimization. Complexity from the dynamical model is also reduced by using B-splines to parametrize the trajectory of the flat outputs as in [Salaris, 2019].

As an instance of flat systems, we focus on quadrotors. For quadrotors and other classes of mobile robots modeled by differential equations, differential flatness theory [Fliess, 1995] simplifies trajectory planning. Many modern quadrotors trajectory generation algorithms leverage differential flatness to enable real-time applications such as drone racing [Qin, 2024], occlusion-aware tracking [Penin, 2018], and power line inspection [Xing, 2023].

After introducing the method, we specifically focus on two problems. The first, concerns the collision avoidance for a quadrotor confronted with an unknown dynamically moving object, observed through bearing-only measurements. The proposed approach automatically recovers trajectories that were previously derived in similar settings, see [Ning, 2024; He, 2019; Li, 2023].

The second case study illustrates the generality of the method by applying it to a different context: the fusion of Inertial Measurement Unit (IMU) and position measurements (typically GNSS outdoors) for drift-compensated pose estimation for quadrotors. Since IMU dynamics are nonlinear, localization accuracy depends on the system's motion [Hermann, 1977]. This has long been recognized in aeronautics: aircraft pilots perform acceleration maneuvers such as "S-turns" or "axial acceleration" to initialize yaw estimation and IMU calibration [Porat, 1981]. However, such manoeuvers, called "in-flight alignment", have hardly been investigated for quadrotors.

Even though these two applications differ significantly at first glance, as in the first case the

control system (quadrotor) and the estimated system (moving object) are different, they can both be addressed within the same unified framework, thus illustrating its versatility. Moreover, covariance-based optimization is both intuitive, theoretically grounded, and practical, since the EKF is already required for estimation in most applications.

This chapter is organized as follows. Section 1 presents the proposed active sensing optimization method. Section 2 applies it to dynamic collision avoidance, and compares the results with prior work. Finally, Section 3 demonstrates the method's generality through IMU-GNSS fusion on quadrotors, and discusses execution time.

1 Minimum Covariance Active Sensing Method

1.1 Overall Architecture

Consider a quadrotor robot equipped proprioceptive sensors (e.g., IMU) or exteroceptive sensors (e.g., camera) to perform estimation. The robot must follow a predefined reference trajectory, and relies on an EKF to estimate its state or its environment. To reduce the uncertainty of the estimation, our goal is to optimally deform the reference trajectory, ahead of time. To enforce physical feasibility of the trajectory and limit extra energy, we formulate the optimal control problem of minimizing a hybrid objective as in Chapter 2 Section 3.1.2: $\mathcal{L}_a + \lambda\mathcal{L}_c$, where $\mathcal{L}_a$ is a penalty on the EKF's uncertainty, $\mathcal{L}_c$ represents a penalization on the energy associated with the deviation to the reference trajectory, and λ is a tuning parameter that controls the energy penalty.

An important consideration is that the EKF's dynamical model need not match the quadrotor's own dynamics. This flexibility is useful when the sensing system has its own dynamics (e.g., IMUs) or when the quadrotor observes an autonomous system of interest. In spite of this, thanks to flatness properties [Faessler, 2018], the physical control inputs (thrust and torques), as well as the EKF's inputs, may generally be expressed as (distinct) functions of the flat outputs and their derivatives, allowing for a common optimization framework.

The flat output's trajectory may be expressed in turn as a function of a number of control points, using B-Splines. Starting from the final cost $\mathcal{L}_a + \lambda\mathcal{L}_c$, the gradients may then be backpropagated all the way up to B-Splines control points, allowing for optimization with respect to these control points, as illustrated on Figure 6.1. In the remainder of this section, we detail the subsystems at play, and the optimization scheme.

1.2 Quadrotor Control Subsystem

1.2.1 Control System Equations

To model the dynamics of a quadrotor, two coordinate frames are defined: the world frame W and the body frame B. The origin O_B of the body frame is attached to the quadrotor's center of mass (COM), with the $\mathbf{z}_B$-axis aligned with the propeller rotational axes. The COM is assumed to coincide with the barycenter of the propellers. The state of the quadrotor is given by: $\boldsymbol{\chi} = [\mathbf{p}, \mathbf{v}, \mathbf{R}, \boldsymbol{\omega}]$, where $\mathbf{p} \in \mathbb{R}^3$ is the position of the COM in W, $\mathbf{v}$ is the linear velocity

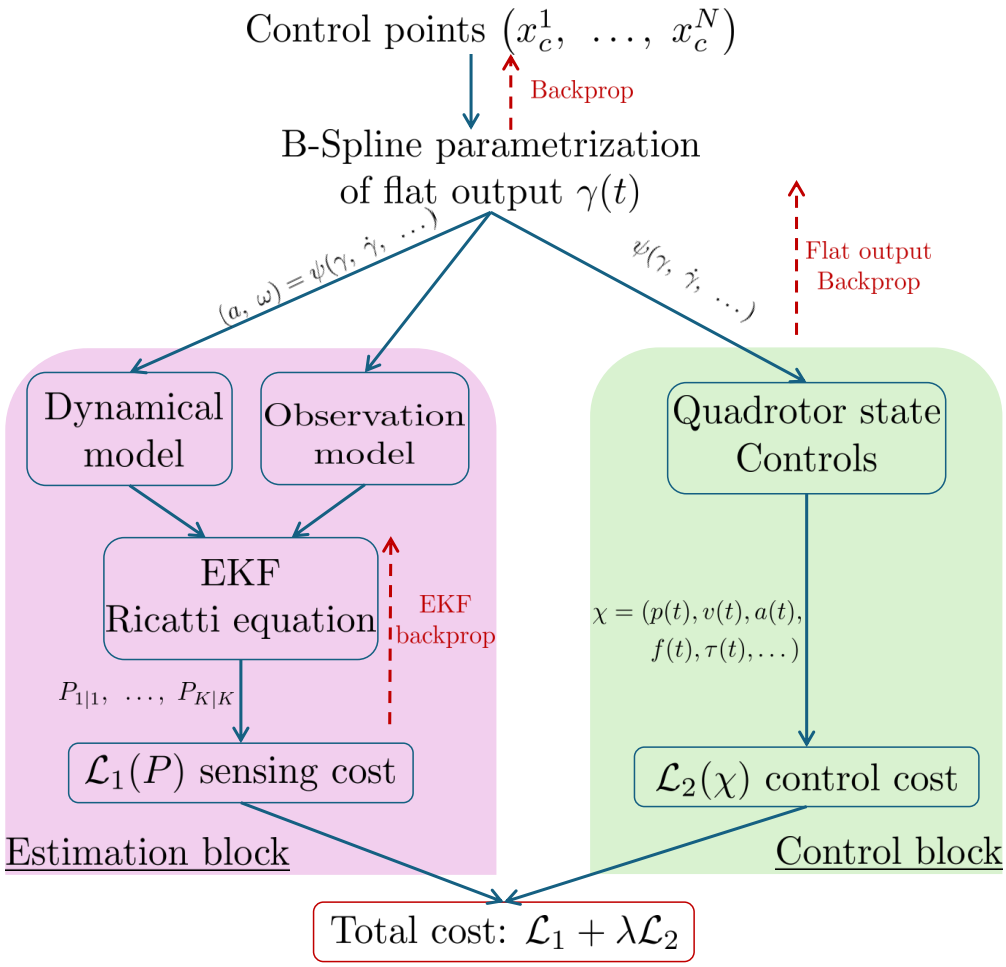

Figure 6.1: Framework for optimal active sensing. The estimation block and the control block operate in parallel and are tightly coupled. The objective is to improve estimation by acting on the control block. The estimated system may be the quadrotor itself, or another system, observed through the quadrotor. Backpropagation efficiently relates the cost to the flat output.

of the COM in W, $\mathbf{R} \in SO(3)$ is the rotation matrix from W to B, and $\boldsymbol{\omega}$ is the angular velocity in B. The rotation matrix is defined with the ZXY Euler angle convention (ϕ, θ, ψ) as $\mathbf{R} = R_z(\psi)R_x(\phi)R_y(\theta)$. The quadrotor dynamics is described by the following equations:

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v}\,, \\ \dot{\mathbf{v}} = -g\mathbf{z}_W - \dfrac{f}{m}\mathbf{z}_B\,, \\ \dot{\boldsymbol{R}} = \boldsymbol{R}(\boldsymbol{\omega})_\times\,, \\ \dot{\boldsymbol{\omega}} = J^{-1}(-(\boldsymbol{\omega})_\times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{\tau})\,, \end{cases} \tag{6.1}$$

with m the mass of the quadrotor, $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is the inertia matrix, f and $\boldsymbol{\tau}$ are the total thrust and torques applied by the propellers. $(.)_\times$ denotes the 3×3 skew operator.

1.2.2 Differential Flatness for Quadrotors

Flat systems, are a class of nonlinear control systems formalized by Fliess *et al.* in the 1990s [Fliess, 1995] and has since become a cornerstone in nonlinear control theory. Differential flatness enables the transformation of certain trajectory optimization problems, typically nonconvex in their standard state-input formulation, into convex problems via an appropriate change of variables. Introduced as an extension of the ideas underlying partial feedback linearization, differential flatness has proven particularly impactful in many applications.

Formally, a dynamical system is said to be differentially flat if there exists a set of outputs, referred to as flat outputs, such that all states and inputs can be explicitly written as functions of the flat outputs and a finite number of their time derivatives. This property has implications for trajectory planning. By parameterizing the desired system behavior in terms of flat outputs, one can guarantee that any trajectory in the flat output space is feasible. Moreover, the flatness property often allows for the decoupling of the planning problem into lower-dimensional subspaces, significantly reducing computational complexity. Another useful property is that parameterization with flat outputs eliminates the need for numerical integration of the system dynamics, which can be prone to numerical instabilities.

Importantly, the identification of flat outputs is highly system-dependent and generally requires substantial insight into the system's structure, often involving algebraic manipulation or exploiting the system's mechanical properties. There is, as of yet, no general numerical procedure for determining the existence of flat outputs or for constructing them for a given system.

Fortunately, quadrotors are known to be differentially flat, with the position and the yaw angle $\boldsymbol{\gamma}^\top = [\mathbf{p}^\top, \psi]$ as flat outputs [Mellinger, 2011]. Then, an invertible algebraic mapping can be established between the flat outputs and the full state and control inputs of the system. This eliminates the need for numerical integration of the system dynamics, reducing computational complexity and allowing for real-time implementation of optimization-based control strategies. In the case of quadrotor, there exists a function to compute the full state and input control from flat outputs and their derivatives to the 4-*th* order:

$$\chi = \Psi_\chi \left(\boldsymbol{\gamma},\ \dot{\boldsymbol{\gamma}},\ \ldots,\ \boldsymbol{\gamma}^{(4)} \right) , \tag{6.2}$$

$$\begin{bmatrix} f & \boldsymbol{\tau}^\top \end{bmatrix}^\top = \Psi_u \left(\boldsymbol{\gamma},\ \dot{\boldsymbol{\gamma}},\ \ldots,\ \boldsymbol{\gamma}^{(4)} \right) . \tag{6.3}$$

These mappings are detailed in [Mellinger, 2011], with corrections provided in [Faessler, 2018]. More refined quadrotor models that incorporate drag effects have been proposed in the literature [Kai, 2017]. Nonetheless, our approach remains applicable, as these extended models continue to exhibit differential flatness [Faessler, 2018].

The application of flatness theory to active sensing has been investigated in the literature; notable examples include [Preiss, 2018; Salaris, 2019; Murali, 2019; Napolitano, 2024].

To further simplify trajectory optimization, flat outputs are parameterized using B-Splines [Biagiotti, 2009]. This reduces the problem from infinite to finite dimensional by encoding the trajectory with a set of control points. A B-Spline of order k with N_c control points $x_c^i i = 0^{N_c}$

is defined as:

$$\gamma(t) = \sum_{i=1}^{N_c} x_c^i B_{i,k}(t), \tag{6.4}$$

where $B_{i,k}(t)$ are the basis functions, defined recursively by the Cox-de Boor formula. We use $N_k = N_c + 2k$ uniformly spaced knots over $[0, T]$, repeating the initial and final knots $k + 1$ times. The number of control points manage both trajectory flexibility and computational cost.

B-Spline parametrization offers two main advantages. First, a B-Spline of order k ensures C^{k-2} continuity. Second, the B-Spline curve is contained in the convex hull of its control points so that convex constraints are seamlessly handled. B-Spline parameterization makes trajectory optimization more tractable, as states, inputs, and cost function terms can be expressed in terms of the B-Spline parameters. This enables efficient and feasible motion generation.

1.3 Details About Planning Ahead of Time

For the system state ($\boldsymbol{\chi}$) estimation, an Error-State Extended Kalman Filter is employed, as described in Chapter 4, Section 1. This filter provides an estimate, denoted by $\hat{\boldsymbol{\chi}}$, and also quantifies the uncertainty associated with this estimate through the error covariance matrix, $\boldsymbol{P}_{k|k}$. The EKF equations governing the evolution of the covariance matrix are recalled below:

$$\begin{aligned}
\boldsymbol{P}_{n+1|n} &= \boldsymbol{F}_n \boldsymbol{P}_{n|n} \boldsymbol{F}_n^\top + \boldsymbol{G}_n \boldsymbol{Q} \boldsymbol{G}_n^\top \,, \\
\boldsymbol{K}_{n+1} &= \boldsymbol{P}_{n+1|n} \boldsymbol{H}_n^\top \left(\boldsymbol{H}_n \boldsymbol{P}_{n+1|n} \boldsymbol{H}_n^\top + \boldsymbol{R} \right)^{-1} \,, \\
\boldsymbol{P}_{n+1|n+1} &= (\boldsymbol{I} - \boldsymbol{K}_{n+1} \boldsymbol{H}_n) \boldsymbol{P}_{n+1|n} \,,
\end{aligned} \tag{6.5}$$

where $\boldsymbol{F}_n$ and $\boldsymbol{G}_n$ are the Jacobian of the dynamics with respect to the state and noise, and $\boldsymbol{H}_n$ is the Jacobian of the observation. These Jacobian matrices may depend on the estimated state and inputs, which highlights the intrinsic coupling between the quadrotor's control and estimation processes.

Since our goal is to deform a reference trajectory to make it more informative over a future time horizon, we need to simulate the equations (6.5) that govern the EKF's uncertainty ahead of time, and to relate them to the flat outputs. This may be done as follows.

As random noises in (4.1) are unknown, they are turned off (which amounts to a maximum likelihood assumption as they are Gaussian). The dynamics are integrated in open loop, starting from the current estimates, and fed with the (future) inputs.

Similarly, as one cannot predict what the measurements $\boldsymbol{y}$'s in (4.1) will be, we assume $\boldsymbol{\nu} = \boldsymbol{0}$ (a maximum likelihood assumption). Thus, the EKF estimates coincide with open-loop noise-free integration of equations (4.1).

Finally, we obtain the desired Jacobians to simulate the Riccati equation (6.5) ahead of time, and extract the future covariance matrices we are to optimize.

1.4 Final Optimal Control Problem

The trajectory optimization problem for a finite time horizon T is formulated as follows:

$$\begin{cases} \min\limits_{\mathbf{x_c}} & \mathcal{L}_a(\boldsymbol{P}_{1|1},\ \ldots,\ \boldsymbol{P}_{N|N}) + \lambda \mathcal{L}_c(\mathbf{x_c}) \\ \text{s.t.} & e(\mathbf{x_c}) = 0\,, \quad i(\mathbf{x_c}) \leq 0. \end{cases} \tag{6.6}$$

As discussed in Chapter 2 Section 3.1.2, the cost function consists of two key terms. The first term, $\mathcal{L}_a$, accounts for active sensing by either minimizing final estimation uncertainty or improving estimation quality along the trajectory. Since $\boldsymbol{P}_{n|n}$ is matrix-valued, a scalar metric must be used for optimization. Common choices include the trace (A-optimality) [Preiss, 2018] or the maximal eigenvalue (E-optimality) [Salaris, 2019]. In practice, optimization is typically restricted to specific submatrices, focusing on the variables for which improved information is desired. The second term, $\mathcal{L}_c$, may encode objectives such as reference trajectory tracking, energy efficiency, or the penalization of high-order derivatives to promote smooth trajectories. While actuator limits cannot be explicitly enforced within the flatness framework, this approach yields feasible trajectories in practice, e.g., [Qin, 2024]. Finally, equality constraints can enforce final states or waypoints, while inequality constraints may ensure obstacle avoidance and impose spatial restrictions.

1.5 Backprop-based Optimization Scheme

Gradient-based numerical solvers for the path planning problem require computing the gradient of the cost function with respect to control points. However, using finite difference methods for this gradient is computationally expensive, limiting applicability to small-scale problems [Rafieisakhaei, 2017] or leading to prohibitively large computation times (hours in [Preiss, 2018]). For our problem, the bottleneck lies in computing the gradient of the active sensing cost, owing to the Riccati matrix differential equations (6.5). Since the loss function is scalar-valued while the input is vector-valued, backpropagation is particularly well-suited for this setting, as argued in Chapter 4.

Recalling Figure 6.1, backpropagation is divided into three steps. First, we perform backpropagation through the EKF to compute $\frac{\partial \mathcal{L}_a}{\partial \boldsymbol{\omega}_n}$ and $\frac{\partial \mathcal{L}_a}{\partial \mathbf{f}_n}$, which is the most computationally intensive part. Second, backpropagation through the flat output mapping is performed using JAX [Bradbury, 2018], yielding $\frac{\partial \mathbf{f}_n}{\partial \boldsymbol{\gamma}}$ and $\frac{\partial \boldsymbol{\omega}_n}{\partial \boldsymbol{\gamma}}$. Third, the gradient from flat outputs to control points is obtained through a direct matrix multiplication, giving $\frac{\partial \boldsymbol{\gamma}}{\partial \mathbf{x_c}}$. Finally, the full gradient of the active sensing cost is computed using the chain rule. Section 3.3 proposes and compares four methods to compute $\frac{\partial \mathcal{L}_a}{\partial \boldsymbol{\omega}_n}$ and $\frac{\partial \mathcal{L}_a}{\partial \mathbf{f}_n}$, which is the bottleneck.

2 Application of Active Sensing to Collision Avoidance

In this section we present the application of the method to collision avoidance in the face of an unknown dynamically moving object, observed through bearing-only measurements.

2.1 Dynamic Collision Avoidance with Bearing Measurements

We consider a quadrotor equipped with a monocular camera that enables environment perception. When an "intruder" is detected, the camera provides the relative bearing between the two objects. The goal is to avoid the intruder while following a planned trajectory. However, efficient avoidance requires estimating the intruder's state (e.g., position, velocity), which is particularly challenging with bearing-only measurements due to limited observability. Indeed, many configurations are unobservable and the observing robot shall exhibit higher excitation than the intruder to ensure observability [Ning, 2024]. To solve the issue, [He, 2019] proposed optimizing the heading to maximize an observability criterion for a 2D vehicle based on a bicycle model. More recently, [Li, 2023] optimized the Fisher Information Matrix (FIM) and showed optimal maneuvers in 3D take the form of helical trajectories.

The continuous dynamic model we use for tracking the intruded is the celebrated Singer model [Singer, 1970] in the radar tracking literature. The intruder state, denoted $\boldsymbol{\chi}_k^I$, includes position, velocity, and acceleration, and follows the model (in continuous time):

$$\begin{cases} \dot{\boldsymbol{p}}^I(t) = \boldsymbol{v}^I(t)\,, \\ \dot{\boldsymbol{v}}^I(t) = \boldsymbol{a}^I(t)\,, \end{cases} \tag{6.7}$$

where the acceleration is a Gaussian process such that $\mathbb{E}[\boldsymbol{a}^I(t)] = 0$, with correlation function $\mathbb{E}[\boldsymbol{a}^I(t)\boldsymbol{a}^I(t+\tau)] = \sigma_m^2 \exp(-|\tau|/\gamma)$. σ_m^2 is the variance of the intruder acceleration and γ is the time constant of its manoeuvers, see [Singer, 1970].

The observation model consists of the noisy relative bearing between the two vehicles, in the observer's frame:

$$\boldsymbol{y}_k = \frac{\boldsymbol{R}_k^\top \left(\boldsymbol{p}_k^O - \boldsymbol{p}_k^I\right)}{||\boldsymbol{p}_k^O - \boldsymbol{p}_k^I||} + \boldsymbol{\nu}_k = \frac{\boldsymbol{K}_{cam}^{-1}\boldsymbol{q}_{pix}}{||\boldsymbol{K}_{cam}^{-1}\boldsymbol{q}_{pix}||} + \boldsymbol{\nu}_k\,, \tag{6.8}$$

where $\boldsymbol{R}_k$ is the rotation matrix from the observer drone's body frame to the world frame. $\boldsymbol{p}_k^O$ and $\boldsymbol{p}_k^I$ denote the positions of the observer and intruder drones, respectively. $\boldsymbol{K}_{\text{cam}}$ is the intrinsic calibration matrix of the camera, and $\boldsymbol{q}_{\text{pix}}$ is the pixel measurement vector in homogeneous coordinates. An EKF is used to estimate the intruder state $\hat{\boldsymbol{\chi}}_k$, based on the Singer model and the observation. The quadrotor dynamical model is the same as described in Section 1.2, and its state is assumed perfectly known. At a given time t, the trajectory planning problem for the observing quadrotor over a time horizon T is formulated as:

$$\begin{cases} \min\limits_{\mathbf{x_c}^t} & \lambda \sum\limits_{n=1}^{\lfloor T f_{\text{cam}} \rfloor} \text{Tr}(\boldsymbol{P}_{n|n}^{p^I}) - \widetilde{\min\limits_{t' \in [0,T]}} \left\{ \|\boldsymbol{p}^O(t') - \hat{\boldsymbol{p}}^I(t')\| \right\} \\ \text{s.t.} & \mathbf{x_c}^{t,0} = \mathbf{x_c}^{init}\,, \quad \mathbf{x_c}^{t,f} = \mathbf{x_c}^{final\ ref}\,, \\ & 0 \leq \mathbf{x_c}^{t,z}\,, \quad \mathbf{x_c}^{t\,\top} S \mathbf{x_c}^t \leq \bar{E}\,, \\ & \widetilde{\min\limits_{t' \in [0,T]}} \left\{ \|\boldsymbol{p}^O(t') - \hat{\boldsymbol{p}}^I(t')\| \right\} \geq d_{min}\,. \end{cases} \tag{6.9}$$

The cost function consists of two terms. The first term improves the estimation of the intruder's position, for safety, while the second encourages collision avoidance by maximizing

the minimum distance between the two drones. Since the minimum function is non-differentiable, we use a softmin regularization:

$$\widetilde{\min}(\{v_k\}_{1\leq k\leq N}) = -\frac{1}{p}\log\left(\sum_{k=1}^{N}\exp\left(-pv_k\right)\right), \tag{6.10}$$

where p is a regularization parameter. The future positions of the observer robot are known, as they are decision variables. The future intruder positions are predicted using the Singer model and the current estimated state $\hat{\boldsymbol{\chi}}(t)$. Equality constraints ensure that our quadrotor starts from its current state and returns to the reference trajectory at the end of the optimization horizon. The second-to-last constraint limits energy consumption, while the last imposes a minimum safety distance $d_{\min}$ between the two vehicles.

The problem is solved numerically using a constrained trust-region algorithm [Nocedal, 1999]. The result of the planning is highly dependent on the estimation of the intruder $\hat{\boldsymbol{\chi}}(t)$, which evolves over time in an unpredictable way. Therefore, the trajectory is replanned every τ seconds. This approach is closely analogous to the receding horizon framework employed in Model Predictive Control (MPC).

This application wholly falls into the proposed framework of Figure 6.1, where the estimation block tracks the intruder using the Singer dynamic model, with observations provided by bearing measurements, although control and estimation blocks correspond to different systems. Note that a more realistic vision model could be obtained, by merely disabling EKF updates outside a regularized camera FoV [Murali, 2019].

2.2 Results

The proposed planning method is tested in a scenario where the reference trajectory of the observer quadrotor is a straight line at constant speed. The intruder drone intersects this reference path at time $t = 18\,\mathrm{s}$, following a trajectory with non-constant acceleration (generated using the Singer model with a maneuver time constant $1/\gamma = 3\,\mathrm{s}$). In this setup, both the propagation and observation models run at 10 Hz. The planning problem is solved every $\tau = 2\,\mathrm{s}$ with a time horizon of $T = 20\,\mathrm{s}$.

To highlight the benefit of active sensing, the planning is solved for two values of the weight parameter: $\lambda = 0$, which focuses solely on avoidance, and $\lambda = 10^{-3}$, which balances estimation accuracy and collision avoidance. The resulting trajectories are shown in the xy- and xz-planes in Figure 6.2. The trajectory with active sensing exhibits a helical shape around the reference path, thus recovering automatically the result of [Li, 2023]. However, in our case, the parameters of the helical maneuver are optimized and not manually tuned. Such trajectories result in parallax, enhancing depth estimation. By contrast, the trajectory with $\lambda = 0$ shows only minor deviations from the reference.

On a machine equipped with an AMD EPYC 9534 64-Core CPU at 2.4 GHz, the planning problem is solved online in an average of $1.12 \pm 0.09\,\mathrm{s}$, enabling replanning at $1/\tau = 0.5\,\mathrm{Hz}$. The number of optimization iterations is set to 100.

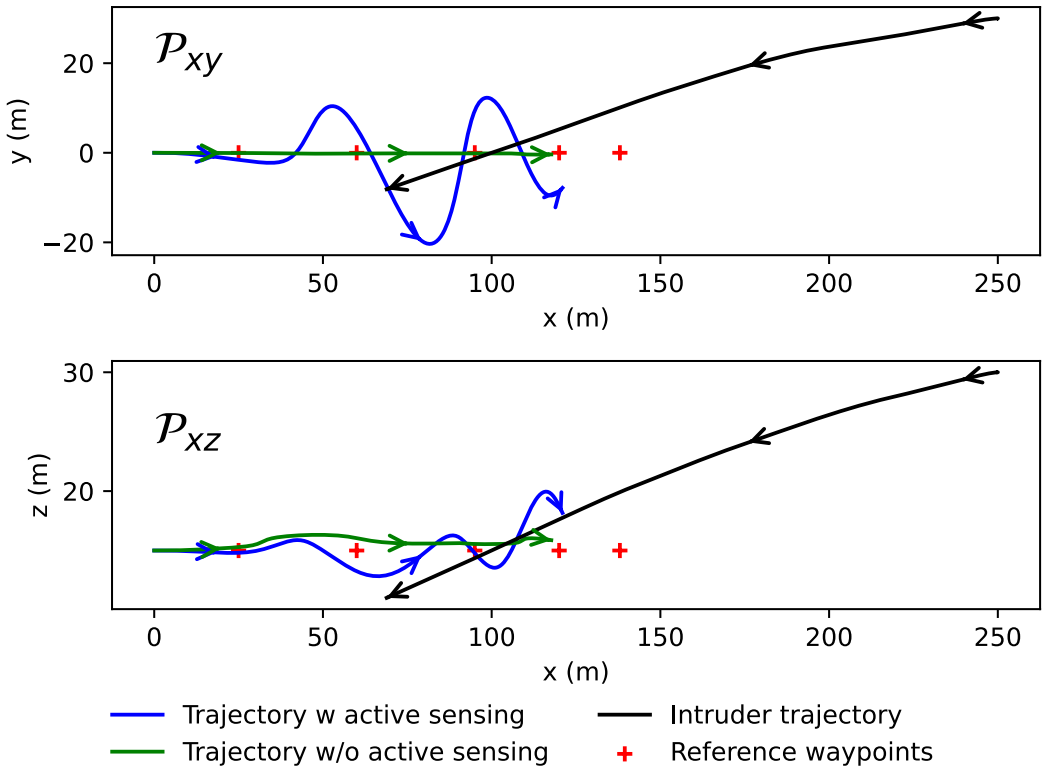

Figure 6.2: Projected trajectories of the observer quadrotor robot with (blue) and without (green) active sensing for intruder (black) avoidance. The reference trajectory is shown as waypoints (red crosses). The top plot shows the xy-plane projection and the bottom plot shows the xz-plane projection.

Regarding differences with [He, 2019] and [Li, 2023], note that our method, albeit numerical, addresses the full 3D case, explicitly accounts for the dynamics of the observer (i.e., our quadrotor), whereas in those previous works a simple bicycle (in 2D) or double integrator (in 3D) serve as the observer's model, and uses a more sophisticated Singer model for tracking, that is known to better track manoeuvers as opposed to a simple constant-speed models.

2.3 Comparisons with Prior Metrics

Our unifying computational framework seamlessly allows for comparisons with alternative approaches from previous literature, based on sensing metrics being simpler to compute numerically. The first considered metric, proposed by Preiss *et al.* [Preiss, 2018], introduces a numerically stable observability measure called the Empirical Local Observability Gramian (E^2LOG). The second is the FIM, used in [Li, 2023] to design helical trajectories for quadrotor tracking. In both cases, the sensing loss seeks to maximize the smallest eigenvalue. For optimization purposes, this eigenvalue is regularized using the Schatten norm: $\mathrm{Tr}(M^p)^{1/p}$ with $p \ll 0$ a large regularization parameter [Salaris, 2019].

To evaluate estimation performance, we conducted 100 Monte Carlo simulations. Figure 6.3 shows the mean estimation error of the relative distance between the two vehicles for the various

metrics. We focus on the relative distance estimation for two reasons. First, it is a critical quantity for safe intruder avoidance. Second, it is known to be poorly observable. Without active sensing, the estimation error increases significantly until the two vehicles come very close. At that point, a sudden drop in the estimation error is observed just before the potential collision, due to the strong parallax effect at close range that makes the problem highly observable. By contrast, active sensing results in a consistently lower estimation error, enabling safer collision avoidance and anticipation. Indeed, the average minimum distance between the drones is $6.01 \pm 3.09\,\mathrm{m}$ with active sensing, compared to $2.25 \pm 0.10\,\mathrm{m}$ without. Note, though, that we intentionally selected a generic representative scenario. Degenerate situations, where both trajectories are initially aligned, would make the depth unobservable, and even more dramatically highlight the role of active perception.

Figure 6.3 allows for further comparisons. We see that although all active sensing losses improve estimation accuracy, the $\mathrm{E}^2\mathrm{LOG}$ measure yields only limited improvement in the present context, whereas both the FIM and the covariance achieve significantly better performance, with comparable results between them. These conclusions should be interpreted with caution, as fair comparisons are difficult to establish. Indeed, the different measures have varying magnitudes, which affect the trade-off between the sensing and control objectives (even though the parameter λ is tuned to ensure comparable active sensing loss scales).

The main result from these simulations is that our covariance-based method matches state-of-the-art methods, while remaining more generic and intuitive as, typically, an EKF is implemented onboard. Moreover, from a theoretical standpoint, the covariance has the additional advantage of accounting for the actual process noise levels in the dynamics, whereas the others not [Rafieisakhaei, 2017]. By turning process noise off, the inverse of the EKF's covariance matches the constructibility Gramian of [Salaris, 2019], as noticed therein, so that our method encompasses such techniques. Thus, the method could lead to larger improvements, when additionally accounting in the EKF for the drone's imperfect onboard sensors (namely IMU), which is more realistic for collision avoidance, whereas for simplicity and comparison purposes we assumed its state perfectly known.

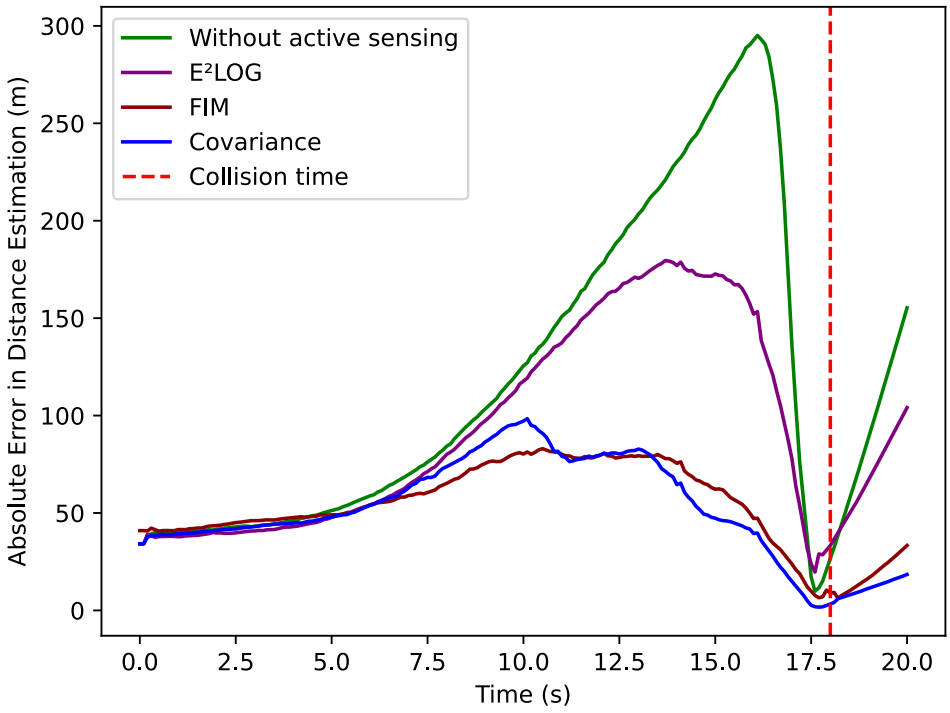

Figure 6.3: Relative distance estimation error averaged over 100 Monte Carlo simulations. Results are shown without active sensing (green) and with active sensing using E^{2}LOG [Preiss, 2018] (purple), FIM-based sensing [Li, 2023] (burgundy), and our covariance-based method (blue).

3 Application to IMU-GNSS fusion

This section illustrates the generality of the proposed method, by applying it to a quite different application: fusion between IMU and position measurements for quadrotors. The application is non-trivial, and we perform a detailed study and discuss execution times.

3.1 Enhancing IMU-GNSS Localization via Active Sensing

The localization of the quadrotor relies on an Error-State EKF that fuses IMU dynamics with GNSS measurements. To simplify notation, we assume the IMU frame is aligned with the quadrotor's body frame, so the variables from Section 1.2 are identical to those used in this section.

The system's state consists of the quadrotor state augmented with gyroscope drift, accelerometer bias, and the IMU-GNSS lever arm as the GNSS antenna and the IMU are at different places: $\boldsymbol{\chi}^{\text{IMU}} = [\boldsymbol{\chi}, \mathbf{d}^{\omega}, \mathbf{b}^{a}, \mathbf{l}]$. The error is defined on the manifold $SO(3) \times \mathbb{R}^{15}$. For attitude, a rotation error $\boldsymbol{\xi}^{R} = \hat{\mathbf{R}}^{-1}\mathbf{R}$ is used; for the remaining state, a linear error $\boldsymbol{\xi}^{\backslash R} = \boldsymbol{\chi}^{\text{IMU}}_{\backslash R} - \hat{\boldsymbol{\chi}}^{\text{IMU}}_{\backslash R}$ is employed. In continuous time, the IMU state evolves according to the following dynamics:

$$\begin{cases} \dot{\mathbf{R}}(t) = \mathbf{R}(t)\left(\boldsymbol{\omega}^g(t) - \mathbf{d}^\omega(t) + \mathbf{w}^\omega(t)\right)_\times, \\ \dot{\mathbf{v}}(t) = \mathbf{R}(t)\left(\mathbf{f}^a(t) - \mathbf{b}^a(t) + \mathbf{w}^a(t)\right) - g\mathbf{z}_w, \\ \dot{\mathbf{p}}(t) = \mathbf{v}(t), \\ \dot{\mathbf{d}}^\omega(t) = \mathbf{w}^{d^\omega}(t), \quad \dot{\mathbf{b}}(t) = \mathbf{w}^{b^a}(t), \quad \dot{\mathbf{l}}(t) = 0, \end{cases} \tag{6.11}$$

where $\boldsymbol{\omega}^g(t)$ and $\mathbf{f}^a(t)$ are the angular velocity (measured by the gyroscope) and specific force (measured by the accelerometers) at time t. All process noise terms $\mathbf{w}$ follow a zero-mean white noise with (overall) covariance $Q(t)$. With such dynamics, the Jacobians in the EKF's Riccati equation (6.5) depend on $\hat{\boldsymbol{\chi}}^{\text{IMU}}(t), \boldsymbol{\omega}^g(t), \mathbf{f}^a(t)$.

As $\mathbf{R}(t)$ and $\mathbf{v}(t)$ in (6.11) coincide with those in (6.1), those variables and their derivatives can be expressed in terms of the flat outputs. We thus easily see that $\boldsymbol{\omega}^g(t) - \mathbf{d}^\omega(t)$ and $\mathbf{f}^a(t) - \mathbf{b}^a(t)$ can be expressed in terms of the flat outputs too. Thus, given flat outputs, as well as a known constant value for the biases $\mathbf{d}^\omega(t)$ and $\mathbf{b}^a(t)$, one can express the inputs of the EKF's equations (6.5) in terms of the flat outputs, and therefore in terms of B-Splines control points. At each GNSS update step k, a measurement is received that depends on the IMU state:

$$\mathbf{y}_k = h(\boldsymbol{\chi}^{\text{IMU}}(t_k)) = \mathbf{p}(t_k) + \mathbf{R}(t_k)\mathbf{l}(t_k) + \boldsymbol{\nu}_k, \tag{6.12}$$

where the measurement $\mathbf{y}_k$ is corrupted by white noise $\boldsymbol{\nu}_k$ with covariance N_k.

The considered application aims to enhance the localization of a quadrotor over a $T = 30$m time window. We simulate an Epsilon™ 10 IMU delivering high-frequency signals with $f_{\text{IMU}} = 100\,\text{Hz}$, and a low-frequency GNSS at $f_{\text{GNSS}} = 5\,\text{Hz}$ with standard deviation of $3\,\text{m}$ in each direction. We used B-Spline of degree 5 with 50 knots distributed uniformly between $[0, T]$. The total number of decision variables is therefore 200. In the experiment, the initial attitude is assumed unknown. The standard deviations for the initial attitude uncertainty are set to 30° for roll and pitch, and 60° for yaw, resulting in significant estimation errors. To improve attitude estimation, we define a specific instance of problem (6.6) as follows:

$$\begin{cases} \min\limits_{\mathbf{x_c}} & \sum\limits_{n=1}^{K} \text{Tr}(\boldsymbol{P}^R_{n|n}) + \lambda {\mathbf{x_c}}^\top \boldsymbol{S} \mathbf{x_c} \\ \text{s.t.} & {\mathbf{x_c}}^0 = {\mathbf{x_c}}^{0\ ref}, \quad 0 \leq {\mathbf{x_c}}^z, \\ & -d_{\max} \leq \mathbf{x_c} - {\mathbf{x_c}}^{ref} \leq d_{\max}, \end{cases} \tag{6.13}$$

with $K = \lfloor T f_{\text{GNSS}} \rfloor$ the number of GNSS measurements and $\boldsymbol{P}^R_{n|n}$ the covariance sub-matrix corresponding to the attitude.

The active sensing part of the cost function concerns only the attitude. Indeed, through position and IMU measurements, estimating attitude is the difficult part (as opposed to the more directly measured position and velocity). Moreover, this avoids mixing variables with different units, which implies nontrivial tuning of weights or normalizations (see e.g., [Preiss, 2018]). The second part of the cost function ensures that the resulting trajectories are smooth and do not excessively violate actuator limits. This term represents the snap (the derivative

of jerk), as advised by [Mellinger, 2011], which is weighted by a parameter $\lambda = 10^{-3}$. Indeed, flatness mapping need derivatives until degree 4. $\boldsymbol{S}$ is a symmetric positive-definite matrix constructed from the integrals of the B-spline basis functions, and it satisfies the relation:

$$\mathbf{x_c}^T \boldsymbol{S} \mathbf{x_c} = \int_0^\top ||\psi^{(2)}(t)||^2 + \sum_{w \in \{x,y,z\}} ||w^{(4)}(t)||^2 \, dt. \tag{6.14}$$

Furthermore, to account for sensing information, while remaining close enough to a reference trajectory generated by an initial planner designed to achieve specific objectives, we impose the constraint that the trajectory must remain inside of a corridor of $d_{\max} = 3\,\mathrm{m}$ around the reference trajectory.

The problem is solved using limited-memory BFGS for bound-constrained (L-BFGS-B) algorithm [Nocedal, 1999], a powerful Quasi-Newton based optimizer. The reference control points are used to "warm start" the optimization solver.

3.2 Results

The presented method is tested to study how a reference trajectory at a constant speed of $3.6\,\mathrm{m.s}^{-1}$ is deformed to improve attitude estimation. The initial trajectory is a straight line in the horizontal plane with a smooth elevation profile, generated by minimizing snap through uniformly spaced waypoints [Mellinger, 2011]. Figure 6.4 shows the top-down view of the reference and active sensing trajectories, as well as the corresponding altitude profiles over time. In the xy-plane, the path remains unchanged. On the attitude plot, the active sensing trajectory shows altitude oscillations, obtained through pitch oscillations.

Even though the path in the xy-plane remains unchanged, the trajectories differ. As shown in Figure 6.5, the acceleration along the x-axis exhibits non-negligible oscillations in the optimized trajectory, reflecting the influence of active sensing optimization. Those oscillations induce a drastic decrease in the yaw uncertainty, see Figure 6.5 bottom plot. This phenomenon is known as the "axial acceleration" in inertial navigation in aerospace engineering, see [Porat, 1981].

The uncertainty on Euler angles is then computed from uncertainty on $\mathfrak{so}(3)$, using an unscented transform. We observe roll and yaw angles remain constant and null throughout the motion, while the pitch angle oscillates, as it is directly related to accelerations along the x-axis through the physics of the quadrotor motion.

To summarize, our planning formulation allows for a deformation of the reference trajectory to enhance yaw estimation accuracy. Yaw is the most difficult orientation component to estimate, as it is orthogonal to the gravity vector and cannot be directly inferred from accelerometer measurements alone, nor from position measurements only. To further evaluate the benefits of our path planning strategy, we performed 10^3 Monte Carlo simulations and report the root mean square error (RMSE) of the attitude estimation in Figure 6.6. The results show improved yaw estimation, with only a slight increase in the RMSE for roll and pitch.

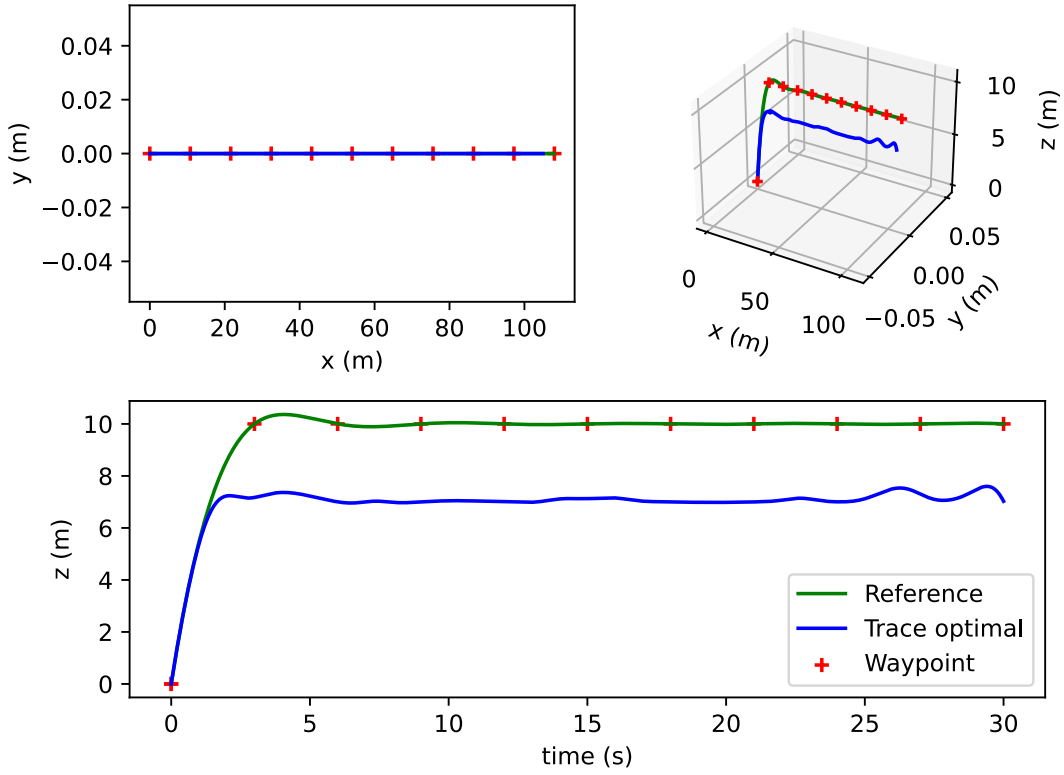

Figure 6.4: The top figure shows the drone's trajectory from above. The figure below shows the evolution of the altitude during the trajectory. The reference trajectory must pass through the red waypoints. The optimal trajectory minimizes a combination of energy and state uncertainty, while being constrained to stay within a 3 m radius about the reference trajectory.

3.3 Gradient Computation Time

Computing the gradient through the EKF equations over the entire time horizon is particularly computationally demanding. Therefore, four different computation methods are compared. These comparisons were performed on a machine equipped with an AMD EPYC 9534 64-Core CPU at 2.4 GHz. The first one is the forward mode of the automatic differentiation (AD) tool JAX [Bradbury, 2018]. The remaining are all based on applying the chain rule of calculus backwards, a.k.a. backpropagation. The second is the AD tool Autograd from PyTorch [Paszke, 2017]. The third is the AD tool JAX in backward mode. The last one is analytical backpropagation of the EKF equations, proposed in Chapter 4. The computation times for the different methods are given in Table 6.1.

The main conclusion, is that using JAX in backward mode yields the best computation time. An important lesson from our studies is that using JAX instead of PyTorch to compute gradients through an EKF yields a very large speedup (×200). However, this result should be interpreted with caution, as the comparison between JAX and other methods is not entirely fair. JAX leverages just-in-time (JIT) compilation, meaning the code is compiled on-the-fly, unlike other approaches. When JIT is disabled, JAX's computation time increases to 6.71 s, making it comparable to PyTorch, and outperformed by analytical backpropagation formulas. Moreover,

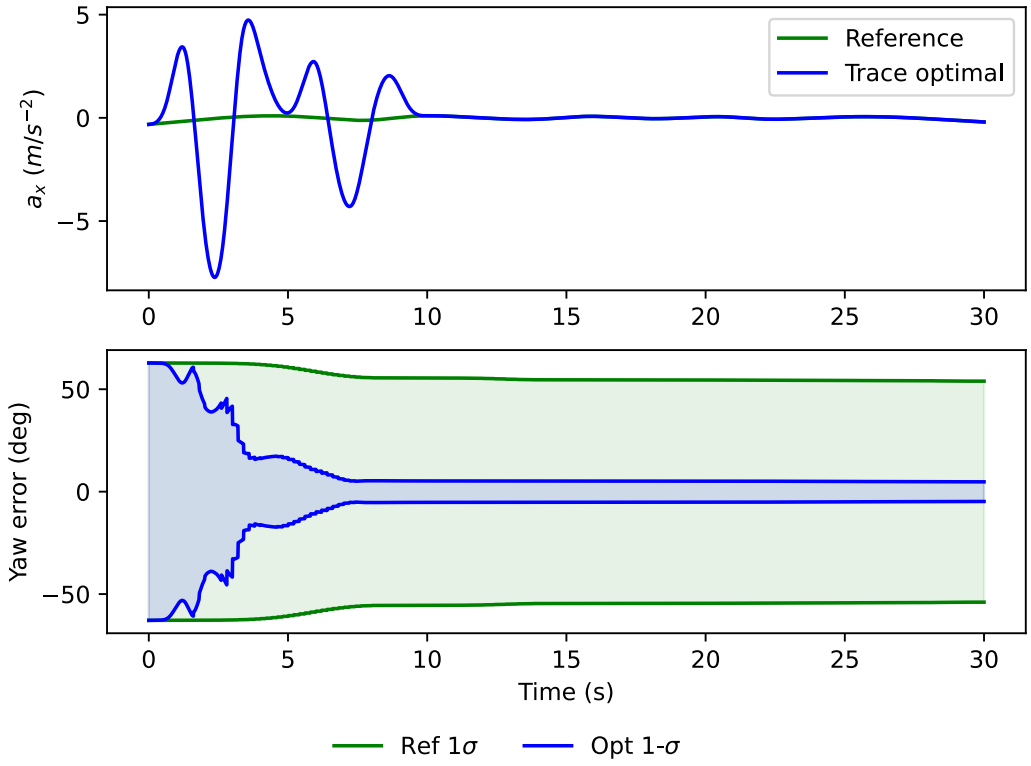

Figure 6.5: The top figure shows the evolution of the (x-axis) acceleration over time for the reference trajectory (green) and the active sensing trajectory (blue). The bottom plot displays the 1σ uncertainty envelope on the yaw angle for both the reference and the active sensing trajectory.

Method	Computation time
JAX forward [Bradbury, 2018]	227.12 s
PyTorch Autograd [Paszke, 2017]	8.47 s ± 0.64 s
Analytical backpropagation	0.96 s ± 58 ms
JAX backward [Bradbury, 2018]	**43 ms** ± 141 μs

Table 6.1: Average and standard deviation of gradient computation times (over 100 computations) using different methods.

analytical formulas have their own merit over automatic methods as they allow one to keep control over the computations, for repeatability, safety, and validation.

The overall optimization of the 30-second straight-line trajectory using JAX-backward with line search took 14.01 ± 0.71 s (over 100 runs), requiring 44 iterations on average. However, this does not permit online replanning. Regarding the in-flight alignment application, which is an initialization procedure used to quickly determine the yaw when starting the quadrotor after it has been transported, finding an optimal trajectory within a few seconds is acceptable.

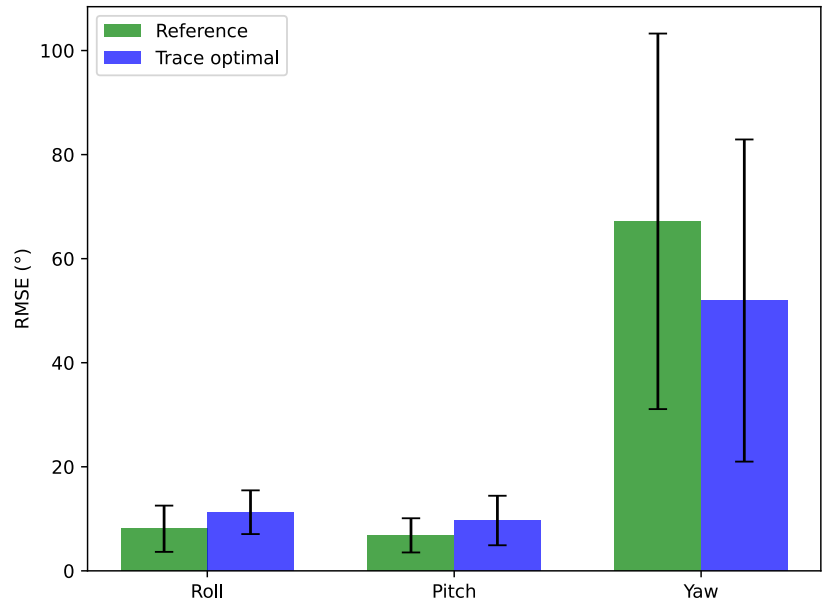

Figure 6.6: The attitude estimation RMSE for the reference and active sensing trajectories was computed using 10^3 Monte Carlo simulations.

4 Discussion

4.1 Global Optimality Analysis

This section discusses the extent to which the method is prone to local minima. In the previous section, the initial guess was set to the reference control points. This choice is motivated by the goal of deforming the reference trajectory to enhance estimation performance. Moreover, the reference trajectory inherently satisfies the constraints. However, one could end up in a local minimum around this particular trajectory. Thus, we performed a sensitivity analysis of the optimal trajectory with respect to the initial B-Spline control points. This helps to determine whether the optimization converges to similar solutions or is prone to local minima. We ran 100 optimization schemes with initial guesses obtained by perturbing the reference: $\mathbf{x_c}^{\text{init}} = \mathbf{x_c}^{ref} + \mathcal{N}(\mathbf{0}_N, 0.5^2\, \boldsymbol{I}_N)$.

During the optimization process, the evolution of the cost function provides insight into convergence behavior. We note that the final objective values after 200 iterations are similar for all initializations, except few instances where optimization converges in a few iterations to a poor local minimum, owing to the initialization (these cases are excluded from the remainder of the study). Despite similar final costs, the corresponding optimal solutions differ significantly, clearly indicating the local nature of the optimization process.

The trajectories in the xy-plane are shown in Figure 6.7. The optimized trajectories exhibit similar patterns, notably alternating turns—referred to as "S-turns" in aeronautics, as well as "axial acceleration" maneuvers. These two motion strategies were previously studied in, e.g., [Porat, 1981], and we see that the optimization algorithm combines them to further improve the localization. The "S-turns" reach the maximum allowable lateral deviation from the reference

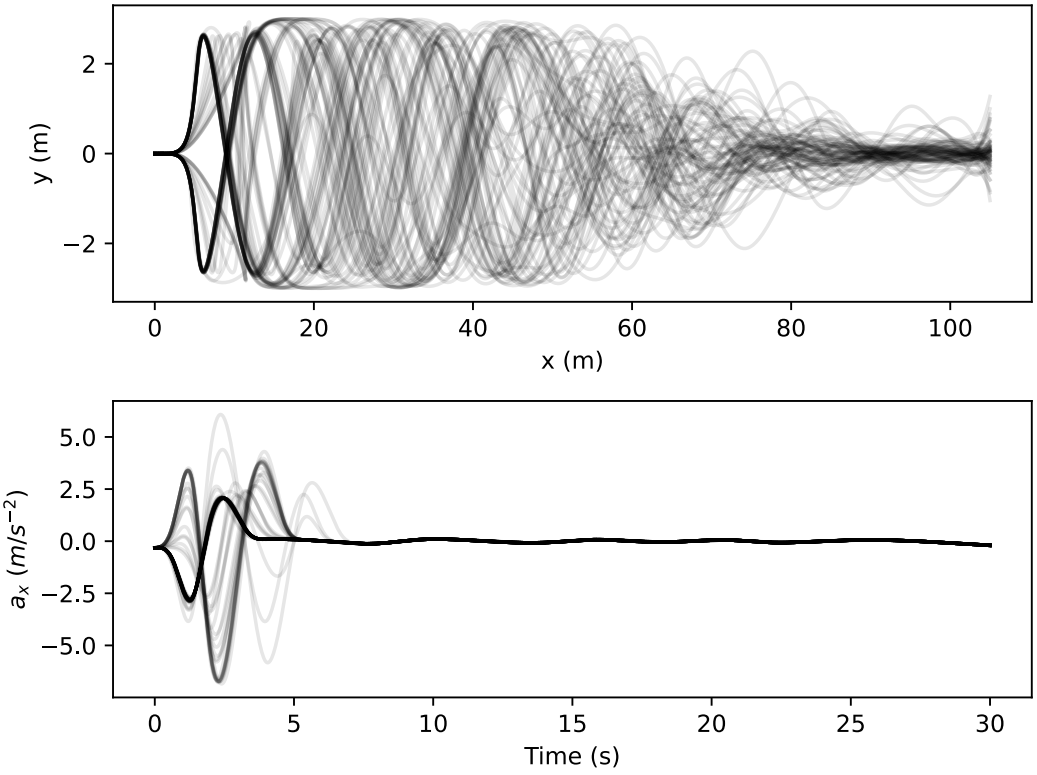

Figure 6.7: Optimized trajectories from 100 randomized initializations, showing similar deformation patterns. In the bottom plot, almost all of the accelerations along the x-axis coincide.

trajectory (3 m). We note that the oscillations eventually fade, as when enough information has been gathered, it is optimal to spend less energy. Along the x-axis, most trajectories exhibit a sharp acceleration variation at the start. By contrast, altitude remains nearly constant across all trajectories, and the yaw does not show any significant variation. Finally, although the optimization may converge to local minima, the presence of recurring patterns across trajectories demonstrates the robustness of the approach.

Returning to the trajectory obtained in Section 3.2, we see the absence of oscillations in the xy-plane ("S-turns") is singular. This is due to a saddle point: the optimization fails to resolve whether to turn left or right. However, for robustness reasons, random initialization of the optimization seems undesirable, as unfortunate initialization may lead to convergence issues.

4.2 Trade-Off Between Smoothness and Observability

A second point we discuss is the relationship between estimation quality and smoothness. Therefore, we designed two additional reference trajectories with different levels of aggressiveness, yielding three different scenarios. We then applied our active sensing planning formulation to each of them. Maximum Euler angle and maximum thrust are used as indicators of trajectory aggressiveness. Thrust is computed for a mass $m = 1\,\text{kg}$.

The results in Table 6.2 show that active sensing reduces attitude covariance. The improve-

	Straight line	Smooth turn	Sharp turn
Loss ref	151.56	25.76	20.72
Max angle ref (°)	2.68	45.4	74.59
Max thrust ref (N)	9.97	16.61	33.93
Loss opt	35.5	25.3	19.68
Max angle opt (°)	39.57	50.49	72.33
Max thrust opt (N)	12.63	18.11	28.7

Table 6.2: Comparison of smoothness and uncertainty across three different scenarios.

ment is most notable for smoother reference trajectories. This is because a smoother trajectory (typically generated by minimizing snap) tends to consume less energy, which leads to poorer observability. Incorporating a active sensing component compensates for this lack of excitation. This result supports the well-known link between system excitation (energy input) and state observability [Willems, 2004]. However, the gain in localization comes at the cost of more aggressive motion as reflected by higher Euler angles and increased thrust, which may be undesirable in contexts involving tight control constraints, limited battery capacity, or fragile payloads.

5 Conclusion

Througth this chapter we proposed a versatile optimization-based trajectory generation method for active sensing, which leverages quadrotor flatness to guarantee feasibility without explicit dynamic integration, while directly minimizing an uncertainty measure provided by the EKF. The method achieves performance comparable to state-of-the-art approaches [Preiss, 2018; Li, 2023]. Efficient gradient computation through the discrete EKF based on backpropagation enables low computation time. The method was shown in simulation to improve attitude estimation for fusion of IMU with position sensors, and to enable dynamic obstacle avoidance, automatically recovering known optimal patterns from [Porat, 1981] and [Li, 2023], using more complex models. This highlights its potential for discovering optimal trajectories without manual tuning.

A direct extension of this work is to apply the proposed approach to other flat systems, such as the bicycle model, as well as to other applications like landmark-based localization. Future work could also exploit and adapt the method for multi-robot systems [Pagano, 2025].

However, two limitations of this work should be noted. The first is that the camera's FoV is not taken into account. A possible solution to this issue is to disable the EKF updates when the target is outside the regularized camera FoV [Murali, 2019]. The second limitation concerns the tuning of the controller gains that tracks the trajectories computed by our method. This tuning is delicate: if the gains are too low, the active sensing effect will be reduced, whereas if they are too high, the system may become overly excited and unstable.

Chapter 7

Optimal Sensor Selection via Backpropagation Through the EKF

Contents

1 Measurement Selection in EKF **87**
- 1.1 Optimization-Based Measurement Selection 87
- 1.2 Measurement Selection in EKF 88
- 1.3 Application 89
- 1.4 Simulation Results 90

2 Mobile Sensor Trajectory Optimization **93**
- 2.1 Simulation Setting 93
- 2.2 Considered Problem 94
- 2.3 Simulation Results 95

3 Conclusion **98**

Résumé

Ce dernier chapitre montre que les équations de rétropropagation dans un filtre de Kalman étendu (EKF) possèdent des applications plus larges que la seule perception active. En particulier, elles offrent un moyen de mieux maîtriser la façon dont les données sont acquises. À cette fin, un problème d'optimisation général pour la sélection optimale de mesures est formulé : il s'agit d'optimiser des paramètres de mesure, tels que leur positionnement spatial ou temporel, afin de minimiser l'incertitude d'estimation.

Deux études de cas illustrent cette démarche. La première concerne la fusion de mesures inertielles avec un recalage cartographique LiDAR, où l'objectif est de sélectionner, sous contrainte de budget, les instants de mesure LiDAR les plus informatifs. Les résultats montrent que l'approche proposée atteint des performances comparables aux méthodes de référence, tout en présentant une complexité plus faible.

La seconde étude de cas traite de la planification de trajectoires d'un capteur mobile, appliquée à un navire mesurant la température d'un environnement marin. La trajectoire est optimisée pour minimiser l'incertitude d'estimation de la température en des points d'intérêt, tout en intégrant les contraintes énergétiques ainsi que l'influence des courants marins. En tenant compte de l'évolution des incertitudes, la méthode obtient une précision supérieure aux approches classiques de surveillance dynamique.

Filtering addresses the statistical estimation of partially observed dynamical systems. Among various approaches, the Extended Kalman Filter (EKF) remains one of the most widely used methods in industry and applications. Constraints such as limited bandwidth, data compression requirements, or restricted energy resources often motivate measurement selection strategies, which aim to determine when or where to acquire measurements in order to maximize the information gained. In the case of linear dynamical systems, early work [Yu, 1973] proposes a greedy algorithm for selecting measurement times, a concept revisited in [Takahashi, 2023; Liu, 2025]. Similarly, [Joshi, 2009] formulates a convex optimization problem to select a fixed number of measurements that maximize the Fisher Information Metric (FIM). This was later extended in [Liu, 2016] to account for measurement noise correlations.

In this chapter, we turn to the nonlinear case, and we focus on measurement selection in a broad sense. We consider a noisy nonlinear dynamical system in the state-space standard form as equation (4.1) in Chapter 4:

$$\begin{cases} \boldsymbol{x_n} = f(\boldsymbol{x_{n-1}}, \boldsymbol{u_n}, \boldsymbol{w_n})\,, \quad \boldsymbol{x_0} = \bar{\boldsymbol{x}}^{\boldsymbol{0}}\,, \\ \boldsymbol{y_n} = h_{\boldsymbol{\gamma_n}}(\boldsymbol{x_n}) + \boldsymbol{v_n}\,, \end{cases} \tag{7.1}$$

The particularity of our framework is that the observation map $h_{\boldsymbol{\gamma_n}}$ depends on a parameter $\boldsymbol{\gamma_n}$ on which we can act, and which is subject to some constraints. We assume the state of the system $\boldsymbol{x_n}$ is estimated using an EKF. We then develop a general method to minimize estimation uncertainty with respect to the parameters $\boldsymbol{\gamma_n}$ as follows

$$\begin{cases} \min\limits_{\boldsymbol{\gamma_0},\dots,\boldsymbol{\gamma_N}} \sum\limits_{n=1}^{N} l_n(\boldsymbol{P_{n|n}}) \\ \text{s.t.} \quad c(\boldsymbol{\gamma_0}, \dots, \boldsymbol{\gamma_N}) \leq 0\,. \end{cases} \tag{7.2}$$

where $\boldsymbol{P_{n|n}}$ is the EKF's error covariance matrix at time n, l_n is a generic loss measuring its magnitude, and c represents constraints.

This encompasses the measurement selection problem, e.g., [Joshi, 2009] (see Section V.G therein) , which focuses on choosing Γ from N sensors in a linear system estimated by a Kalman filter, with the objective of minimizing the final error $\mathrm{Tr}(\boldsymbol{P_N})$. In this setting, $\boldsymbol{\gamma_n}$ can be modeled as a binary decision variable indicating whether a measurement is selected. Building upon the convex-relaxation heuristic of [Joshi, 2009], our approach applies to nonlinear systems and more general objectives, where convexity (and thus some optimality guarantees) no longer holds.

Conceptually, measurement selection can be viewed as the temporal analogue of the spatial

sensor placement problem, which enhances data acquisition by strategically positioning sensors in a static setting. Related works can be found in Chapter 2 Section 3.4. Extending these ideas to dynamic settings leads to the domains of active sensing and coverage control [Pagano, 2025], where, for example, a swarm of mobile robots navigates to locate a diffusive source and estimate its parameters, see also [Gupta, 2006]. The proposed framework naturally accommodates this problem by letting $\boldsymbol{\gamma_n}$ denote the robots' states and control actions, while $\boldsymbol{x_n}$ represents the source parameters the swarm seeks to infer.

In optimization problem (7.2), the loss function l_n is supposed to reflect an uncertainty metric, and as such can take various forms, such as the trace, log-determinant, or maximum eigenvalue of the covariance matrix depending on the chosen decision criterion (for more details see Chapter 2 Section 2.2).

To effectively solve the optimization problem, a first-order solver must be used. The solver requires the gradients of both the loss function and the constraints with respect to $\boldsymbol{\gamma_0}, \dots, \boldsymbol{\gamma_N}$. Because $\boldsymbol{P_{n|n}}$ depends indirectly on $\boldsymbol{\gamma_0}, \dots, \boldsymbol{\gamma_n}$, computing the loss function's gradients remains challenging. As previously explained in Chapter 4, the backpropagation equations within an EKF are particularly efficient for computing the gradient of a loss function with this structure.

In the following sections of this chapter, we address the optimization problem for measurement selection in Section 1. Then, in Section 2, we apply the method to dynamic sensor placement.

1 Measurement Selection in EKF

This section builds upon our problem formulation and gradient computation strategy to tackle the temporal measurement selection problem of [Joshi, 2009]. By extending the framework to nonlinear models and non-convex cost functions, we broaden the range of applicable scenarios, albeit at the expense of losing optimality guarantees.

Measurement selection in an EKF essentially serves two purposes. The first is *a priori*, to plan when and which measurements should be selected to achieve the best estimation performance under sensor budget constraints. The second is *a posteriori*, to select the most informative observations and effectively "compress" the data relevant to the filter.

1.1 Optimization-Based Measurement Selection

We consider a nonlinear dynamical system of the form (7.1), with binary decision variables $\{\gamma_n, 1 \leq n \leq N\}$, such that $h_{\boldsymbol{\gamma_n}}(\boldsymbol{x_n}) = \boldsymbol{\gamma_n} h(\boldsymbol{x_n})$, i.e., a value $\gamma_n = 1$ denotes that measurement n is taken, while $\gamma_n = 0$ indicates otherwise.

Due to resource limitations (e.g., communication, storage, or energy), Γ measurements may be selected, i.e., $\sum_{n=1}^{N} \gamma_n = \Gamma$. The goal is then to choose the most informative subset by optimizing a metric $\mathcal{L}$ that reflects estimation uncertainty or information. This yields a constrained version of the general problem (7.2), where $\gamma_n \in \{0, 1\}$ and $\sum_{n=1}^{N} \gamma_n = \Gamma$.

This is a combinatorial optimization problem that can be addressed using various methods. The most prominent ones are summarized below:

- Greedy methods: these methods iteratively add the sensor or measurement that provides the largest marginal information gain [Yu, 1973; Bian, 2006; Takahashi, 2023; Kaveti, 2024]. Conversely, one may start with all sensors and remove the least informative ones. This approach is justified by the submodularity of many information criteria [Shamaiah, 2010]. Formally, for an information measure $\boldsymbol{\mathcal{I}}$ and a selected sensor set S, the marginal gain of adding a sensor y satisfies

$$\boldsymbol{\mathcal{I}}(S) + \boldsymbol{\mathcal{I}}(y) \geq \boldsymbol{\mathcal{I}}(S \cup y) + \boldsymbol{\mathcal{I}}(S \cap y). \tag{7.3}$$

  Submodularity ensures that greedy methods can achieve provable approximation guarantees [Krause, 2008]. Building on this, [Kaveti, 2024] recently introduced quality certificates for greedy solutions.

- Mixed-Integer Programming (MIP): sensor selection can be naturally formulated as an integer optimization problem, where decision variables take the value 1 if a sensor is selected and 0 otherwise. In similar contexts, this problem has been proven NP-hard [Bian, 2006]. However, methods such as branch-and-bound could exploit upper bounds derived from submodularity [Ko, 1995; Krause, 2008].

- Continuous relaxation: the constraint $\gamma_n \in \{0, 1\}$ is relaxed to $\gamma_n \in [0, 1]$. This relaxation reduces the problem to a convex program in certain cases [Joshi, 2009; Chepuri, 2015; Liu, 2016]. Integer solutions are then obtained through rounding or heuristic procedures. In [Joshi, 2009], this relaxation was applied in a linear setting to minimize the covariance of the MAP estimate of the final state, a convex objective, but it essentially led to selecting only the most recent measurements.

- Local search: iterative schemes that swap sensors in and out of the set to improve the objective function can provide effective heuristics [Takahashi, 2023].

Compared to these methods, we build upon the idea of convex relaxation but use the covariance produced by an EKF as the information measure. Our method could minimizes the average covariance over time, which is more meaningful than the last covariance, and yields more informative results than [Joshi, 2009] at the cost of losing convexity. It also applies to general nonlinear systems (7.1), beyond the linear case.

1.2 Measurement Selection in EKF

In our formulation, the decision variables are scaling factors γ_n applied to the Kalman gain. When $\gamma_n = 0$, the update is skipped (i.e., measurement n is not selected); when $\gamma_n = 1$, the standard EKF update is performed. Using the notation introduced in Chapter 4 Section 1, the problem can be formulated as follows:

$$\begin{cases} \min\limits_{\{\gamma_n\}_{1\leq n\leq N}} \sum\limits_{n=1}^{N} \mathcal{L}(\boldsymbol{P_{n|n}}) \\ \text{s.t.} \quad \boldsymbol{P_{n|n-1}} = \boldsymbol{F_n P_{n-1|n-1} F_n^\top} + \boldsymbol{G_n Q_n G_n^\top}\,, \\ \quad \tilde{\boldsymbol{K}}_{\boldsymbol{n}} = \gamma_n \boldsymbol{K_n} = \gamma_n \boldsymbol{P_{n|n-1} H_n^\top S_n^{-1}}\,, \\ \quad \boldsymbol{P_{n|n}} = (\boldsymbol{I} - \boldsymbol{K_n H_n}) \boldsymbol{P_{n|n-1}}\,, \\ \quad \sum\limits_{n=1}^{N} \gamma_n = \Gamma \text{ and } \forall n,\ \gamma_n \in [0,1]. \end{cases} \tag{7.4}$$

EKF constraints are handled implicitly, and gradients are computed using the backpropagation equations from Chapter 4 Section 4, where the Kalman gain is replaced by a scaled version $\tilde{\boldsymbol{K}}_{\boldsymbol{n}} = \gamma_n \boldsymbol{K_n}$. In the binary case, this parameterization is equivalent to writing $\tilde{\boldsymbol{H}}_{\boldsymbol{n}} = \gamma_n \boldsymbol{H_n}$. Furthermore, this novel parameterization offers greater numerical stability during optimization. Indeed, the direct parameterization with $\boldsymbol{H_n}$ requires inversions and squared values of γ, which can lead to instability. The gradient with respect to the decision variables is then obtained using the chain rule in the context of matrix parameters:

$$\frac{\partial \mathcal{L}}{\partial \gamma_n} = \mathrm{Tr}\left(\frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{K}}_{\boldsymbol{n}}}^{\top} \boldsymbol{K_n} \right). \tag{7.5}$$

To enforce constraints on the decision variables, we apply a projected gradient descent algorithm. The projection onto the convex set $\mathcal{C} = \left\{ \gamma \in \mathbb{R}^N \mid \sum \gamma_n = \Gamma \text{ and } \gamma_n \in [0,1] \right\}$ is performed using the algorithm in [Wang, 2013]. A discrete vector of selected measurements is then obtained by taking the Γ highest values of $\boldsymbol{\gamma}$, akin to [Joshi, 2009]. The obtained cost provides an upper bound of the optimal cost.

1.3 Application

We focus on the *a priori* measurement selection in an inertial system that uses map-based matching using LiDAR to localize and navigate in a known environment, a problem of engineering interest, see e.g., [Xia, 2023]. Owing to its active nature that illuminates the environment, it may be desirable in certain contexts to limit its usage by allowing a maximum number of measurements. Another motivation for selection could be storage or computation limitations. The system state corresponds to the IMU attitude $\boldsymbol{R} \in SO(3)$, velocity $\boldsymbol{v}$, position $\boldsymbol{p}$ expressed in the fixed frame. Additionally the system state is composed of accelerometer biases $\boldsymbol{b}$ and gyroscope drifts $\boldsymbol{d}$ expressed in the body frame. The state is denoted by $\boldsymbol{\chi_n} \in SO(3) \times \mathbb{R}^{12}$ and

evolves as (see, e.g., [Barrau, 2023]):

$$\begin{cases} \boldsymbol{R_{n+1}} = \boldsymbol{R_n} \exp_{SO(3)} \left(dt(\boldsymbol{\omega_n} - \boldsymbol{d_n} - \boldsymbol{w_n^\omega}) \right) , \\ \boldsymbol{v_{n+1}} = \boldsymbol{v_n} + dt\, \boldsymbol{R_n} \left(\boldsymbol{a_n} - \boldsymbol{b_n} - \boldsymbol{w_n^a} \right) - dt\, \boldsymbol{g} , \\ \boldsymbol{p_{n+1}} = \boldsymbol{p_n} + dt\, \boldsymbol{v_n} , \\ \boldsymbol{d_{n+1}} = \boldsymbol{d_n} + \boldsymbol{w_n^d} , \\ \boldsymbol{b_{n+1}} = \boldsymbol{b_n} + \boldsymbol{w_n^b} . \end{cases} \tag{7.6}$$

The noise terms $\boldsymbol{w_n} = \left[\boldsymbol{w_n^\omega}, \boldsymbol{w_n^a}, \boldsymbol{w_n^d}, \boldsymbol{w_n^b} \right]^\top \sim \mathcal{N}(\boldsymbol{0_{12}}, \boldsymbol{Q})$ model the IMU uncertainty. The observation model corresponds to position-based map matching through LiDAR, and is defined as:

$$\boldsymbol{y_n} = h(\boldsymbol{\chi_n}) + \boldsymbol{\nu_n} := \boldsymbol{p_n} + \boldsymbol{\nu_n} , \tag{7.7}$$

where $\boldsymbol{\nu_n} \sim \mathcal{N}(\boldsymbol{0_3}, \boldsymbol{R})$ is the observation noise. In our experiments, we generate inertial and observation data based on the above models, assuming known noise levels. For state estimation, we use an error-state EKF where the attitude error is represented on $SO(3)$ as $\boldsymbol{\xi_n^R} = \log_{SO(3)}(\hat{\boldsymbol{R}}^\top \boldsymbol{R})$ and a linear error is used for the remaining variables.

The loss function is the sum of the attitude covariance traces: $\mathcal{L} = \sum_{n=1}^{N} \mathrm{Tr}(\boldsymbol{P_{n|n}^R})$. Note that in the present *a priori* context, this makes sense, as one may want to find a trajectory that minimizes uncertainty at all times along the planned trajectory, to avoid collisions or unsafe trajectories, and not at the final moment only, as in [Joshi, 2009].

The problem in (7.4) is solved numerically using the Projected Gradient Descent algorithm. In the following, this approach is referred to as PGD.

In the following, we compare our measurement selection method to the greedy approach proposed in [Yu, 1973; Takahashi, 2023]. The greedy approach starts with an empty sequence of measurements. At each iteration, all possible candidates are evaluated to determine which measurement provides the greatest information gain. The most informative one is then added to the sequence. This process is repeated until the maximum allowed number of measurements is reached. The second method is adapted from [Joshi, 2009] and consists of a local search optimization: starting from a valid measurement selection (i.e., satisfying the constraints) , the algorithm searches for the swap (activation/deactivation) that improves the loss the most, repeating until no local improvement is possible.

1.4 Simulation Results

The proposed approach was validated in simulation and compared against alternative optimization strategies. For the numerical simulation, we use the IMU parameters from the Epsilon™ 10, with a process noise covariance sampled at 100 Hz ($dt = 10^{-2}$ s). LiDAR measurements for map-based localization are available at 1 Hz with a measurement noise covariance set to $\boldsymbol{R} = \boldsymbol{I}_3$. The initial covariance is defined with a standard deviation of 5° for each rotation axis, 1 m.s^{-1} for velocity, and 1 m for position. In all scenarios, only 30% of the available LiDAR measurements are selected. We applied our method to a trajectory of duration $T = 150$ s. In Figure 7.1,

the measurement selection vector $\boldsymbol{\gamma}$ is displayed for our method (PGD) as well as for the greedy and local search baselines. Selected LiDAR measurements are shown in black, while unselected time steps are in gray.

Both our method and the greedy approach exhibit similar selection patterns: a dense measurements sequence during the transient phase, followed by regularly spaced observations during steady-state. While this structure is intuitive, designing such a schedule manually would be challenging, as the optimal number and spacing of measurements depend on the system's covariance tuning. In contrast, the local search strategy results in a nearly uniform distribution of selected measurements, lacking adaptation to system dynamics.

To confirm that PGD's solution is locally optimal, we verified that no pairwise swap of selected and unselected measurements further reduces the loss. If this condition were violated, a few additional local search steps as in [Joshi, 2009] could be applied to refine the solution.

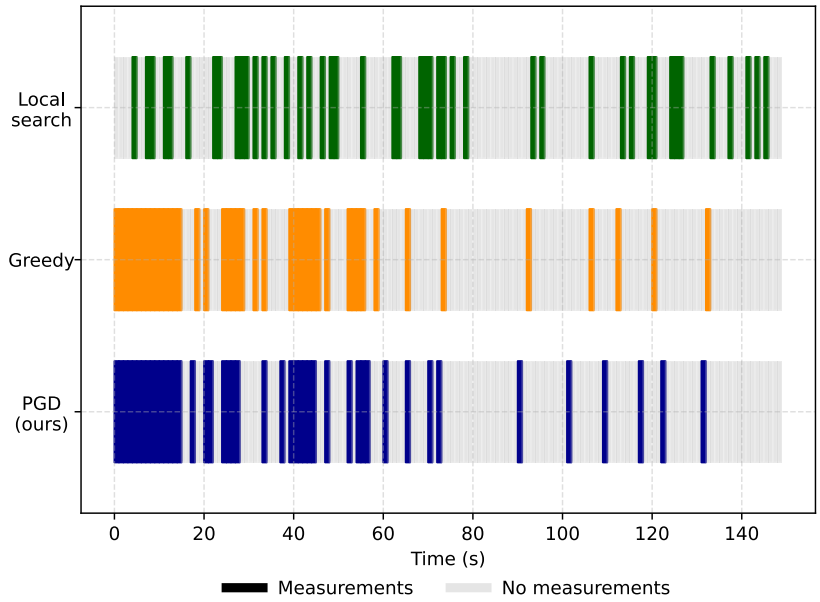

Figure 7.1: Measurement selection sequence obtained using three different methods, including our PGD approach. The greedy and PGD methods both exhibit a clear transition between a transient phase with dense measurements and a steady-state phase with periodic updates.

To assess the quality of the selected measurement sequences, Figure 7.2 reports the final loss values obtained using different approaches. We first generated 10^4 random selection sequences, which result in widely dispersed and generally suboptimal performance. Similarly, the local search method yielded relatively high loss values.

For PGD, we ran 100 optimizations with different random initializations to evaluate robustness with respect to the starting point. The resulting distribution of loss values is tight, indicating that PGD consistently avoids poor local minima. Interestingly, the greedy method achieves a final loss very close to that of PGD. Notably, both PGD and greedy methods, while using only 30% of the available measurements, achieve performance close to that obtained when all measurements are used, in an application of actual engineering interest.

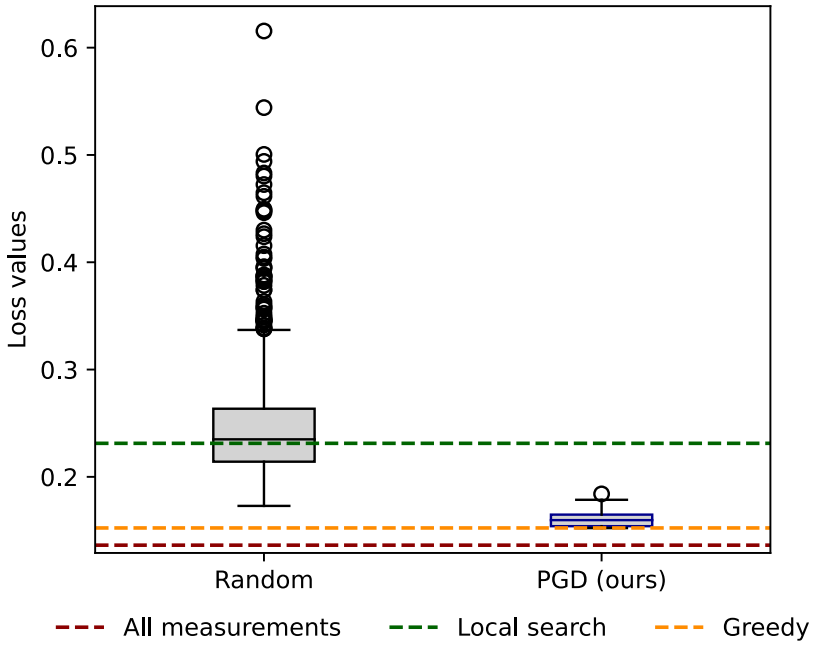

Figure 7.2: Final loss values for 10^4 random measurement selection sequences, as well as the greedy, local search, and PGD methods. PGD was run 100 times with random initialization. The red dashed line indicates the loss when all measurements are used.

Although PGD and the greedy method achieve comparable performance in terms of loss minimization, their computational complexity differs significantly. The greedy strategy has a computational cost of $O(\Gamma N^2)$, while PGD operates at $O(N)$ complexity, assuming a fixed number of iterations.

Figure 7.3 shows the computation time as a function of trajectory length. This highlights the scalability of PGD and the advantage it offers for longer trajectories, where greedy becomes increasingly expensive.

In summary, the proposed method based on PGD with backpropagation through the EKF yields strong performance in terms of estimation quality while maintaining excellent computational efficiency.

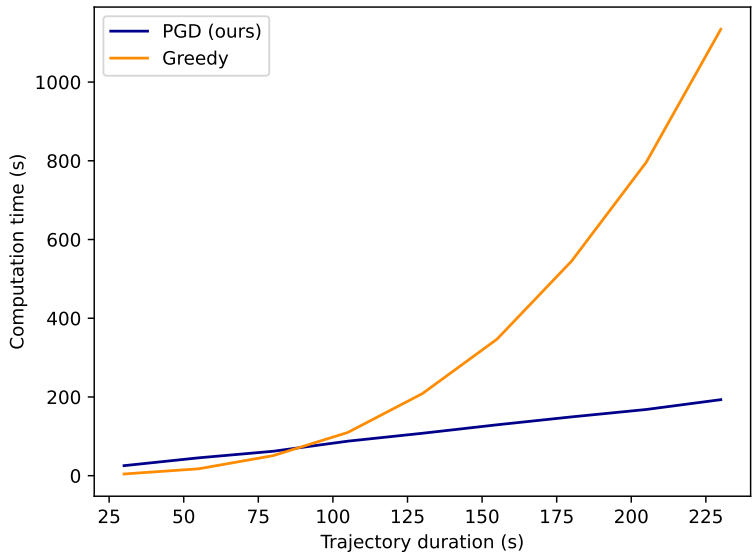

Figure 7.3: Computation time of the optimization process as a function of trajectory length, for both PGD and the greedy method.

2 Mobile Sensor Trajectory Optimization

In this section, we address the problem of optimal trajectory planning for a mobile temperature sensor. The goal is to determine a trajectory for the vessel that maximizes the accuracy of temperature estimation at several points of interest, while accounting for the influence of ocean stream on the vessel's motion [Agrawal, 2023].

2.1 Simulation Setting

We consider a vessel equipped with a temperature sensor that has a measurement range of r and is subject to white Gaussian noise with variance σ_{obs}. The mission aims to plan a temperature acquisition campaign for N_{POI} points of interest, maximizing estimation accuracy at the mission's end. The mission duration is fixed and denoted as T. The state vector representing the temperatures at the points of interest is denoted as $\boldsymbol{m}$, with its estimation covariance $\boldsymbol{P}$. Since ocean temperatures evolve over time, the state $\boldsymbol{m}$ follows a dynamic process. Dynamical and observation models are summarized by the following equations:

$$\begin{cases} \boldsymbol{m_{n+1}} = \boldsymbol{m_n} + \boldsymbol{w_n} \text{ with } \boldsymbol{w_n} \sim \mathcal{N}(\boldsymbol{0}_{N_{\text{POI}}}, \sigma_{dyn}\boldsymbol{I}_{N_{\text{POI}}}) \,, \\ \quad \boldsymbol{y_n} = \boldsymbol{H_n}\boldsymbol{m_n} + \boldsymbol{\nu_n} \text{ with } \boldsymbol{\nu_n} \sim \mathcal{N}(\boldsymbol{0}_{N_{\text{POI}}}, \sigma_{obs}\boldsymbol{I}_{N_{\text{POI}}}) \,. \end{cases} \tag{7.8}$$

In the observation model, $\boldsymbol{H_n}$ is a square matrix of size $N_{\text{POI}} \times N_{\text{POI}}$ which depends on the position of the sensor. The matrix is zero everywhere except for the element (i, i) which is 1 if the point of interest i is in the sensor range at time n and 0 otherwise. The matrix is written

as:

$$(\boldsymbol{H_n})_{i,i} = \boldsymbol{H}(\mathbf{x_n})_{i,i} = \mathbb{1}_{\left\{||\mathbf{x}_n^{pos} - \mathbf{p}_i|| \leq r\right\}}\,, \tag{7.9}$$

where $\boldsymbol{p_i}$ corresponds to the position of the i-th point of interest and $\boldsymbol{x_n^{pos}}$ is the sensor position at time n. The sensor dynamics correspond to the dynamics of the vessel. In our case, we use the kinematic unicycle model, with constant speeds c_x and c_y in each direction modeling ocean stream:

$$\begin{cases} x_{n+1} = x_n + dt\,(v_n \cos(\theta_n) + c_x)\,, \\ y_{n+1} = y_n + dt\,(v_n \sin(\theta_n) + c_y)\,, \\ \theta_{n+1} = \theta_n + dt\omega_n\,. \end{cases} \tag{7.10}$$

At each time step, the water temperature is estimated using a Kalman Filter (KF). Although the vessel dynamics are not directly included in the KF, they influence the observations, as shown in Equation (7.9).

This planning problem can be approached using various methods already established in the literature. For instance, a similar problem arises in robotic surveillance. A common strategy is to discretize the environment into a graph and design patrol routes using Markov chains [Duan, 2021]. While these methods are computationally efficient, they do not easily account for vessel dynamics constraints. Another approach is ergodic control, which aims to ensure that the time-averaged trajectory of the vehicle matches a specified probability distribution over space [Miller, 2013]. Ergodic control can enable closed-loop coverage, dynamically adapting to the evolving expected information density to enhance sensing [Miller, 2016]. A more recent approach introduces an information measure called clarity and formulates the optimal control problem using Hamilton-Jacobi-Bellman (HJB) equations [Agrawal, 2023].

2.2 Considered Problem

We leverage gradient backpropagation within the EKF framework to solve the sensor trajectory planning. To ensure trajectory feasibility, we exploit the flatness property of the unicycle model [Fliess, 1995], using its flat outputs (x and y) parametrized by B-spline control points $\boldsymbol{x_c}$. An energy constraint is also introduced to limit consumption, expressed in quadratic form as: $\int_0^T (\dot{x}(t) - c_x)^2 + (\dot{y}(t) - c_y)^2\,dt = \bar{E}$. This leads to the following constrained optimization problem:

$$\begin{cases} \min\limits_{\boldsymbol{x_c^x}, \boldsymbol{x_c^y}} & \mathcal{L}(\boldsymbol{P_{N|N}}) \\ \text{s.t.} & \boldsymbol{x_c}^\top \boldsymbol{A} \boldsymbol{x_c} + \boldsymbol{B} \boldsymbol{x_c} + c = \bar{E}\,. \end{cases} \tag{7.11}$$

To enable gradient-based optimization, we need the gradient of the loss with respect to the flat output parameters. The cost function, however, is not directly differentiable with respect to the sensor position because the observation Jacobian (7.9) contains indicator functions. To address this, we replace each indicator function with a smooth sigmoid approximation, controlled

by a sharpness parameter $\alpha \gg 1$:

$$\tilde{\boldsymbol{H}}(\boldsymbol{x_n})_{i,i} = \sigma\left(\alpha(r - \|\boldsymbol{x_n^{pos}} - \boldsymbol{p_i}\|)\right) . \tag{7.12}$$

The gradient of the loss with respect to the decision variables is then computed using the backpropagation equations. Specifically, equation (4.25) provides the sensitivity of the cost to the observation Jacobians. Then, using the chain rule (with matrix parameters), we compute at each instant:

$$\begin{aligned}\frac{\partial \mathcal{L}(\boldsymbol{P_{N|N}})}{\partial \boldsymbol{x_n^{pos}}} = \Bigg[&\mathrm{Tr}\left(\frac{\partial \mathcal{L}(\boldsymbol{P_{N|N}})}{\partial \boldsymbol{H_n}}^\top \frac{\partial \boldsymbol{H_n}}{\partial x_n^x}\right), \\ &\mathrm{Tr}\left(\frac{\partial \mathcal{L}(\boldsymbol{P_{N|N}})}{\partial \boldsymbol{H_n}}^\top \frac{\partial \boldsymbol{H_n}}{\partial x_n^y}\right)\Bigg] .\end{aligned} \tag{7.13}$$

In comparison with previous works, this gradient has not been derived. Finally, the gradient of the cost function with respect to the spline control points is computed using a matrix product. Once the gradient has been computed, the problem is solved using the SLSQP optimization algorithm [Nocedal, 1999].

We compare our method against two baselines: the first is an ergodic iLQR controller [Miller, 2016], where the spatial distribution is constructed by averaging uniform densities of radius r centered at each point of interest. To ensure fair comparison, the weight matrix $\boldsymbol{R_u}$ in the control cost is tuned using Brent's algorithm so that the total energy consumed matches the budget $\bar{E}$. The second method, proposed by Agrawal *et al.* [Agrawal, 2023], introduces a clarity-based information metric defined as:

$$q[\boldsymbol{X}] \coloneqq \left(1 + \frac{\exp(2h[\boldsymbol{X}])}{(2\pi e)^n}\right)^{-1}, \tag{7.14}$$

where n is the dimension of the random variable $\boldsymbol{X}$ and $h[\boldsymbol{X}]$ its entropy. The associated optimization problem is then solved via Hamilton-Jacobi equations. However, for each point of interest, a new differential equation governing clarity evolution must be added to the system state. Solving these equations on a grid leads to an exponential growth in computational cost. For example, with more than three points of interest, even coarse discretizations require over 32 GB of RAM. For comparison, we adopt their clarity metric as the cost function within our optimization framework (7.11), thereby retaining the original optimization objective while avoiding scalability issues. In this formulation, $q[\boldsymbol{m_N}] = 1/\left(1 + \det(\boldsymbol{P_{N|N}})\right)$.

2.3 Simulation Results

For the numerical experiments, we set the time horizon to 6 min, with ocean stream defined as $c_x = 1\,\mathrm{km/h}$ and $c_y = -0.2\,\mathrm{km/h}$. The EKF operates with a propagation and update every $dt = 10^{-4}\,\mathrm{h}$). The covariance matrices are set as $\boldsymbol{R} = 0.01\,\boldsymbol{I}_{N_{\mathrm{POI}}}\,(^\circ\mathrm{C})^2$, $\boldsymbol{Q} = 0.01\,dt\,\boldsymbol{I}_{N_{\mathrm{POI}}}\,(^\circ\mathrm{C})^2$, and $\boldsymbol{P_0} = 25\,\boldsymbol{I}_{N_{\mathrm{POI}}}\,(^\circ\mathrm{C})^2$. For simplicity, all matrices are diagonal, although cross-covariances could be introduced to model temperature correlations between nearby points of interest. The energy constraint is fixed at $\bar{E} = 50\,\mathrm{km}^2/\mathrm{h}^2$, and the sensing radius is set to $r = 0.25\,\mathrm{km}$. We

consider $N_{\text{POI}} = 10$ points of interest, uniformly sampled over the $[0\text{km}, 1\text{km}]^2$ domain. The optimizer is initialized with random control points for the B-spline trajectory.

The choice of the loss function $\mathcal{L}$ directly influences the shape of the resulting covariance. In this work, we consider two commonly used criteria [Mihaylova, 2003]. The first is the trace of the covariance matrix, which corresponds to the sum of its eigenvalues. The second is the log-determinant, which is proportional to the entropy of the estimate and represents the volume of the uncertainty ellipsoid.

The trajectories obtained for the different optimization objectives are shown in Figure 7.4. We also display the spatial density used by the ergodic controller: dark regions indicate areas where the sensor can simultaneously observe multiple points of interest, while light gray regions represent areas with no measurement.

The ergodic trajectory tends to pass near many points of interest and remains mostly within the dark regions, effectively covering the high-density areas. The trajectories optimized for minimizing the trace and log-determinant of the covariance are quite similar in shape. Both traverse high-information regions and terminate in areas where more than four points can be sensed simultaneously.

In contrast, the trajectory optimized using the clarity metric appears less informative. It fails to visit several regions, resulting in high uncertainty in those areas. A closer look at the loss evolution during clarity optimization reveals that the metric saturates near its maximum value (i.e., 1) within just 2-3 iterations. This behavior is due to numerical issues: the optimization becomes ill-conditioned, as the clarity loss involves the product of eigenvalues.

We proceed to assess the impact of the various trajectories on the accuracy of temperature field estimation. To this end, Figure 7.5 shows the standard deviation of the estimated temperature computed by the EKF at each point of interest. The results show that the ergodic trajectory yields higher final uncertainty compared to the other methods. To further demonstrate the benefits of our method, we conducted 100 Monte Carlo simulations. The final root mean squared error (RMSE) obtained for each trajectory is reported in Table 7.1.

Table 7.1: Monte Carlo results over 100 trials. Our method achieves a lower RMSE.

Trajectory	Error (°C)
Ergodic	0.071 ± 0.02
Trace (ours)	0.048 ± 0.01
Log-det (ours)	$\mathbf{0.047} \pm 0.01$
Clarity	8.524 ± 3.60

This occurs because the ergodic controller ignores the temporal evolution of the temperature field. Although it initially measures several points of interest, it does not revisit them later. Consequently, by the end of the mission, the temperature at these points may have changed, but no additional measurements are taken to update the estimate.

In contrast, our proposed methods explicitly account for how information degrades over time. Analyzing the velocity profiles further supports this: the ergodic trajectory maintains a roughly constant speed, whereas the optimized trajectories start slowly and accelerate at the end. Since

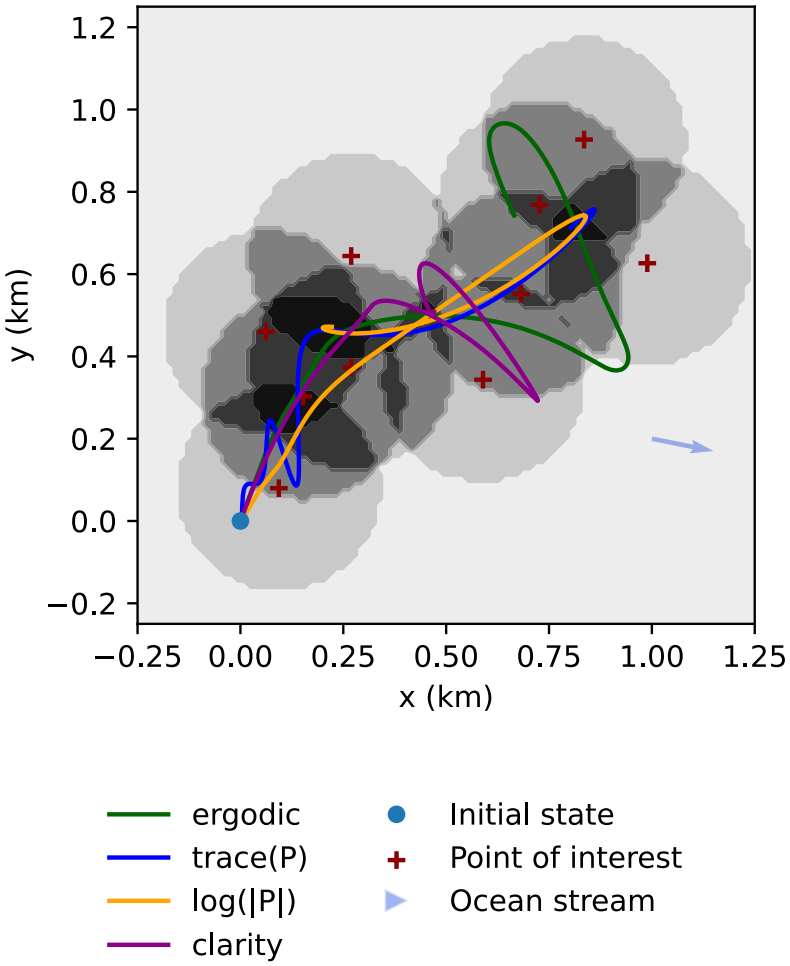

Figure 7.4: The four optimal trajectories start from the same point and move between the points of interest (red crosses). The direction of the ocean current is indicated by a blue arrow. The black color gradient indicates the spatial density followed by the ergodic controller. The more informative the zone, the higher the probability that the robot will end up in it.

our goal is to minimize the final uncertainty, it is more effective to conserve energy early in the mission and invest it later, when observations are more critical for improving estimation quality.

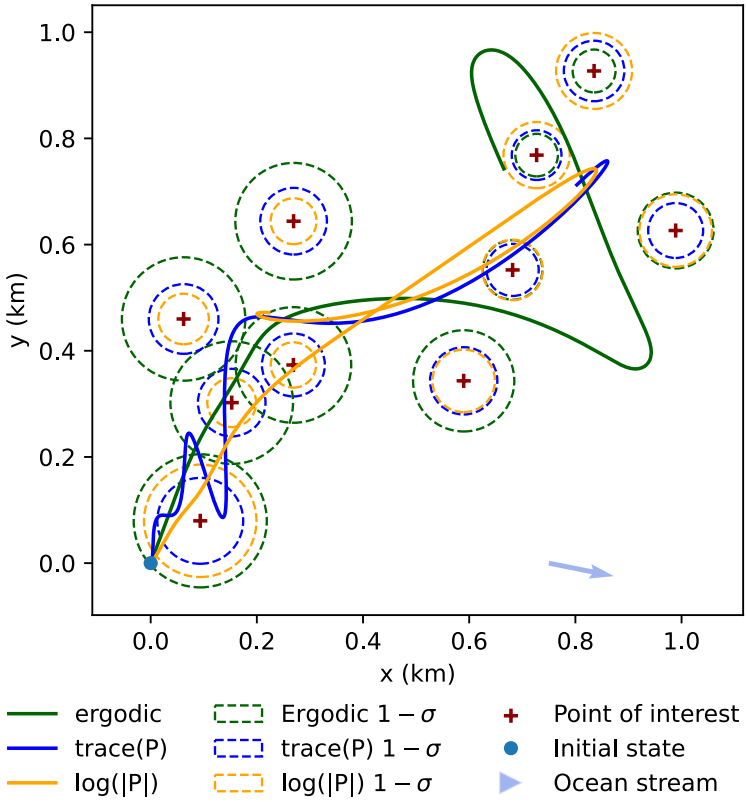

Figure 7.5: Comparison of the three trajectories and the resulting final temperature uncertainty. The ergodic trajectory yields significantly less accurate estimates than those obtained from our method. For readability, the clarity-based trajectory is omitted due to its large covariances, which would distort the plot.

3 Conclusion

This chapter presented a unified and versatile methodology for addressing both temporal and dynamic spatial sensor placement problems by leveraging the backpropagation equations within the Extended Kalman Filter. The methodology was applied to two distinct engineering case studies, demonstrating its effectiveness, although its potential applications extend far beyond these examples.

This chapter illustrates how the backpropagation equations within an EKF make it possible to control the entire data acquisition process. After introducing active sensing, it shows how measurement selection and sensor placement can be implemented to enhance the overall data acquisition process.

Conclusion and Perspectives

In the introduction, we highlighted the importance of accounting for the many sources of uncertainty inherent to robotics (and more generally automatic control and signal processing) in order to improve system robustness and performance. Solving a stochastic optimal control problem with partial information (i.e., POMDPs) remains computationally challenging. We therefore chose to focus on the uncertainties that directly affect the data acquisition process, and to assume the models are well-known and the statistics of the noise known. Yet even under this restriction, the problem remains computationally expensive.

The proposed formulation, based on minimizing a scalar criterion of the covariance produced by an Extended Kalman Filter (EKF), appears as an appealing compromise. It allows us to address a deterministic control problem while explicitly capturing the evolution of uncertainties through the EKF covariance. Compared with methods in the literature, relying on the EKF to model uncertainty results in a simple and practical formalism. Moreover, many applications already use an EKF for state estimation.

Our first conclusion is that the active sensing strategy we propose effectively improves estimation accuracy across a wide range of applications. We demonstrated this through diverse Monte Carlo simulations as well as a real-world experiment. Our approach is particularly beneficial in two situations: firstly, when the system state is only partially observable, as in the collision-avoidance example, and secondly, when the trajectory-generation algorithm produces very smooth trajectories. In such cases, the system is insufficiently excited and our method becomes especially valuable.

The second conclusion relates to the backpropagation equations. Compared with many existing techniques, these equations substantially reduce computation time. Another strategy is the use of automatic differentiation tools, such as JAX, which can offer even faster computations. However, automatic differentiation relies on computational mechanisms that are difficult to inspect and not always compatible with embedded platforms. In contrast, the proposed backpropagation equations mirror Kalman filtering equations for gradient computation and are more easily embedded. Depending on the target application, one or the other approach may therefore be preferable.

One last remark concerns the selection and tuning of EKF parameters. We showed that optimization can determine not only when but also where measurements should be acquired. This extends optimization-based covariance tuning approaches such as [Parellier, 2023b; Xu, 2024],

suggesting that EKF parameters in a broader sense may be selected through optimization.

During this work, the question arose regarding the impact of using the EKF on estimation performance. A key limitation of the EKF is the dependence of the Kalman gain on the linearization point. However, this issue largely disappears when the filter is used only to propagate uncertainty forward. It is therefore worth investigating whether our EKF-based active sensing approach also improves estimation when the actual filtering is performed by another method, such as a particle filter or an Unscented Kalman Filter (UKF). If the answer is negative, it may be interesting to compute covariance in the active sensing problem using another differentiable filtering technique. For example, one could use an UKF or adapt our method to Multiple Model Adaptive Estimation (MMAE and IMM) approach, provided that the filtering steps remain differentiable with respect to the controls.

A second research direction is invariant filtering [Barrau, 2017]. As mentioned in Chapter 4, Section 2 and more thoroughly in our paper [Benhamou, 2023], reducing the dependence of Jacobians on linearization points allows the active sensing problem to be solved analytically in simple settings. Using an Invariant Extended Kalman Filter (IEKF) to model uncertainty may therefore change the conditioning of the optimization problem in a way that helps convergence toward global extrema.

A third perspective is to study how an active sensing method can be coupled with a trajectory-tracking controller. It is important to avoid using a controller whose objective is to minimize energy consumption or to smooth the reference trajectory, as this would counteract the excitation introduced by active sensing. Similarly, one should avoid a controller that is overly aggressive and further increases the excitation already present in the reference trajectory. Consequently, it is necessary to examine how to ensure effective coexistence between active sensing and the system's control architecture.

Finally, the method is general and applies to a large class of systems of engineering interest, provided they possess the flatness property. Beyond investigating other examples, one could study how to use the techniques for systems that are not flat, for instance the IMU-GNSS fusion example of Chapter 6, where the quadrotor equations, known to be flat, are replaced with those of an aircraft. A natural first step could be to use a simplified surrogate model (e.g., a bicycle-type or other reduced-order representation) to compute informative trajectories, before examining how to reconcile this approximation with the aircraft's full dynamics.

Bibliographie

[Abbeel, 2005] Pieter Abbeel, Adam Coates, Michael Montemerlo, Andrew Ng, and Sebastian Thrun. "Discriminative Training of Kalman Filters". *Robotics: Science and Systems*. 2005, pp. 289–296 (cit. on pp. 43, 52).

[Agrawal, 2023] Devansh R. Agrawal and Dimitra Panagou. "Sensor-Based Planning and Control for Robotic Systems: Introducing Clarity and Perceivability". *IEEE Control Systems Letters* 7 (2023), pp. 2623–2628 (cit. on pp. 93–95).

[Bai, 2012] Haoyu Bai, David Hsu, Mykel J. Kochenderfer, and Wee Sun Lee. "Unmanned Aircraft Collision Avoidance Using Continuous-State POMDPs". *Robotics: Science and Systems VII*. The MIT Press, 2012 (cit. on pp. 11, 32).

[Bai, 2013] Haoyu Bai, David Hsu, and Wee Sun Lee. "Planning How to Learn". *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 2853–2859 (cit. on p. 18).

[Bajcsy, 1988] R. Bajcsy. "Active Perception". *Proceedings of the IEEE* 76.8 (1988), pp. 966–1005 (cit. on pp. 19, 20, 23).

[Barrau, 2023] Axel Barrau and Silvere Bonnabel. "The Geometry of Navigation Problems". *IEEE Transactions on Automatic Control* 68.2 (2023), pp. 689–704. arXiv: 2201.04426 [cs, eess] (cit. on pp. 38, 41, 90).

[Barrau, 2017] Axel Barrau and Silvère Bonnabel. "The Invariant Extended Kalman Filter as a Stable Observer". *IEEE Transactions on Automatic Control* 62.4 (2017), pp. 1797–1812 (cit. on pp. 38, 100).

[Bartolomei, 2020] Luca Bartolomei, Lucas Teixeira, and Margarita Chli. "Perception-Aware Path Planning for UAVs Using Semantic Segmentation". *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 5808–5815 (cit. on pp. 20, 21, 23, 24).

[Baydin, 2018] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. "Automatic Differentiation in Machine Learning: A Survey". *Journal of Machine Learning Research* 18.153 (2018), pp. 1–43 (cit. on pp. 43, 44).

[Benhamou, 2023] Jonas Benhamou, Silvere Bonnabel, and Camille Chapdelaine. "Optimal Active Sensing Control for Two-Frame Systems". *2023 62nd IEEE Conference on Decision and Control (CDC)*. 2023, pp. 5967–5974 (cit. on pp. 5, 40, 100).

[Benhamou, 2024] Jonas Benhamou, Silvère Bonnabel, and Camille Chapdelaine. "Backpropagation-Based Analytical Derivatives of EKF Covariance for Active Sensing". *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2024, pp. 12240–12247 (cit. on pp. 5, 47).

[Benhamou, 2026] Jonas Benhamou, Silvère Bonnabel, and Camille Chapdelaine. "Optimal Sensor Selection via Backpropagation Through the EKF". ***Major revision requested*** *from IEEE Transactions on Signal Processing* (2026) (cit. on p. 5).

[Benhamou,] Jonas Benhamou, Silvère Bonnabel, and Camille Chapdelaine. "Minimum-Covariance Active Sensing for Quadrotors". ***In preparation*** (cit. on p. 5).

[Biagiotti, 2009] Luigi Biagiotti and Claudio Melchiorri. *Trajectory Planning for Automatic Machines and Robots*. Berlin, Heidelberg: Springer, 2009 (cit. on p. 70).

[Bian, 2006] F. Bian, D. Kempe, and R. Govindan. "Utility-Based Sensor Selection". *2006 5th International Conference on Information Processing in Sensor Networks*. 2006, pp. 11–18 (cit. on pp. 25, 88).

[Bo, 2019] Xu Bo, Asghar A. Razzaqi, and Ghulam Farid. "A Review on Optimal Placement of Sensors for Cooperative Localization of AUVs". *Journal of Sensors* 2019.1 (2019), p. 4276987 (cit. on p. 26).

[Bradbury, 2018] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, et al. *JAX: Composable Transformations of Python+NumPy Programs*. 2018 (cit. on pp. 43, 72, 80, 81).

[Brault, 2021] Pascal Brault, Quentin Delamare, and Paolo Robuffo Giordano. "Robust Trajectory Planning with Parametric Uncertainties". *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 11095–11101 (cit. on pp. 20, 28).

[Brossard, 2020] Martin Brossard, Axel Barrau, and Silvère Bonnabel. "AI-IMU Dead-Reckoning". *IEEE Transactions on Intelligent Vehicles* 5.4 (2020), pp. 585–595 (cit. on p. 43).

[Brossard, 2022] Martin Brossard, Axel Barrau, Paul Chauchat, and Silvére Bonnabel. "Associating Uncertainty to Extended Poses for on Lie Group IMU Preintegration With Rotating Earth". *IEEE TRANSACTIONS ON ROBOTICS* 38.2 (2022) (cit. on p. 38).

[Burgard, 1997] W. Burgard, D. Fox, and S. Thrun. "Active Mobile Robot Localization by Entropy Minimization". *Proceedings Second EUROMICRO Workshop on Advanced Mobile Robots*. 1997, pp. 155–162 (cit. on p. 23).

[Chepuri, 2015] Sundeep Prabhakar Chepuri and Geert Leus. "Sparsity-Promoting Sensor Selection for Non-Linear Measurement Models". *IEEE Transactions on Signal Processing* 63.3 (2015), pp. 684–698 (cit. on pp. 25, 88).

[Cognetti, 2018] Marco Cognetti, Paolo Salaris, and Paolo Robuffo Giordano. "Optimal Active Sensing with Process and Measurement Noise". *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 2118–2125 (cit. on pp. 19, 22, 42, 52).

[Cortes, 2004] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. "Coverage Control for Mobile Sensing Networks". *IEEE Transactions on Robotics and Automation* 20.2 (2004), pp. 243–255 (cit. on p. 26).

[Costante, 2017] Gabriele Costante, Christian Forster, Jeffrey Delmerico, Paolo Valigi, and Davide Scaramuzza. *Perception-Aware Path Planning*. 2017. arXiv: 1605.04151 [cs] (cit. on pp. 21, 24).

[Daş, 2025] Ersin Daş, Thomas Touma, and Joel W. Burdick. "Bayesian Optimal Experimental Design for Robot Kinematic Calibration". *2025 IEEE International Conference on Robotics and Automation (ICRA)*. 2025, pp. 1–7 (cit. on p. 18).

[Duan, 2021] Xiaoming Duan and Francesco Bullo. "Markov Chain–Based Stochastic Strategies for Robotic Surveillance". *Annual Review of Control, Robotics, and Autonomous Systems* 4.Volume 4, 2021 (2021), pp. 243–264 (cit. on pp. 26, 94).

[Eidenberger, 2010] Robert Eidenberger and Josef Scharinger. "Active Perception and Scene Modeling by Planning with Probabilistic 6D Object Poses". *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 1036–1043 (cit. on p. 11).

[Faessler, 2018] Matthias Faessler, Antonio Franchi, and Davide Scaramuzza. "Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories". *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 620–626. arXiv: 1712.02402 [cs] (cit. on pp. 68, 70).

[Falanga, 2018] Davide Falanga, Philipp Foehn, Peng Lu, and Davide Scaramuzza. "PAMPC: Perception-Aware Model Predictive Control for Quadrotors". *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 1–8 (cit. on pp. 20, 21, 24).

[Feldbâum, 1963] úndefined. Á. Feldbâum. "Dual Control Theory Problems". *IFAC Proceedings Volumes*. 2nd International IFAC Congress on Automatic and Remote Control: Theory, Basle, Switzerland, 1963 1.2 (1963), pp. 541–550 (cit. on p. 13).

[Flayac, 2019] Emilien Flayac. "Coupled Methods of Nonlinear Estimation and Control Applicable to Terrain-Aided Navigation". Theses. Université Paris Saclay (COmUE), 2019 (cit. on pp. 22, 23).

[Fliess, 1995] Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon. "Flatness and Defect of Non-Linear Systems: Introductory Theory and Examples". *International Journal of Control* 61.6 (1995), pp. 1327–1361 (cit. on pp. 67, 70, 94).

[Franceschini, 2008] Gaia Franceschini and Sandro Macchietto. "Model-Based Design of Experiments for Parameter Precision: State of the Art". *Chemical Engineering Science*. Model-Based Experimental Analysis 63.19 (2008), pp. 4846–4872 (cit. on p. 18).

[Gibson, 1962] James J. Gibson. "Observations on Active Touch". *Psychological Review* 69.6 (1962), pp. 477–491 (cit. on p. 3).

[Grewal, 2015] Mohinder S. Grewal and Angus P. Andrews. "Kalman Filtering: Theory and Practice with MATLAB". *Kalman Filtering: Theory and Practice with MATLAB*. IEEE, 2015, pp. 1–36 (cit. on p. 38).

[Gupta, 1974] N. Gupta and R. Mehra. "Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations". *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 774–783 (cit. on p. 43).

[Gupta, 2006] Vijay Gupta, Timothy H. Chung, Babak Hassibi, and Richard M. Murray. "On a Stochastic Sensor Selection Algorithm with Applications in Sensor Scheduling and Sensor Coverage". *Automatica* 42.2 (2006), pp. 251–260 (cit. on p. 87).

[Haarnoja, 2016] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. "Backprop KF: Learning Discriminative Deterministic State Estimators". *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016 (cit. on p. 43).

[Hassani, 2013] Vahid Hassani, António M. Pascoal, and Asgeir J. Sørensen. "A Novel Methodology for Adaptive Wave Filtering of Marine Vessels: Theory and Experiments?" *52nd IEEE Conference on Decision and Control*. 2013, pp. 6162–6167 (cit. on p. 43).

[Hausman, 2015] Karol Hausman, Jörg Müller, Abishek Hariharan, Nora Ayanian, and Gaurav S Sukhatme. "Cooperative Multi-Robot Control for Target Tracking with Onboard Sensing". *The International Journal of Robotics Research* 34.13 (2015), pp. 1660–1677 (cit. on pp. 19, 22, 52).

[He, 2008] Ruijie He, Sam Prentice, and Nicholas Roy. "Planning in Information Space for a Quadrotor Helicopter in a GPS-denied Environment". *2008 IEEE International Conference on Robotics and Automation (ICRA)*. 2008, pp. 1814–1820 (cit. on pp. 19, 21, 52).

[He, 2019] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. "Trajectory Optimization for Target Localization With Bearing-Only Measurement". *IEEE Transactions on Robotics* 35.3 (2019), pp. 653–668 (cit. on pp. 67, 73, 75).

[Hermann, 1977] R. Hermann and A. Krener. "Nonlinear Controllability and Observability". *IEEE Transactions on Automatic Control* 22.5 (1977), pp. 728–740 (cit. on pp. 53, 67).

[Hewing, 2020] Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. "Learning-Based Model Predictive Control: Toward Safe Learning in Control". *Annual Review of Control, Robotics, and Autonomous Systems* 3.Volume 3, 2020 (2020), pp. 269–296 (cit. on p. 14).

[Hinson, 2013] Brian Hinson, Michael Binder, and Kristi A Morgansen. "Path Planning to Optimize Observability in a Planar Uniform Flow Field". *American Control Conference.* (2013) (cit. on p. 19).

[Hu, 2024] Haimin Hu, David Isele, Sangjae Bae, and Jaime F Fisac. "Active Uncertainty Reduction for Safe and Efficient Interaction Planning: A Shielding-Aware Dual Control Approach". *The International Journal of Robotics Research* 43.9 (2024), pp. 1382–1408 (cit. on p. 14).

[Huan, 2024] Xun Huan, Jayanth Jagalur, and Youssef Marzouk. "Optimal Experimental Design: Formulations and Computations". *Acta Numerica* 33 (2024), pp. 715–840 (cit. on p. 16).

[Indelman, 2016] Vadim Indelman, Luca Carlone, and Frank Dellaert. "Towards Planning in Generalized Belief Space". *Robotics Research: The 16th International Symposium ISRR*. Ed. by Masayuki Inaba and Peter Corke. Springer Tracts in Advanced Robotics. Cham: Springer International Publishing, 2016, pp. 593–609 (cit. on p. 58).

[Joshi, 2009] Siddharth Joshi and Stephen Boyd. "Sensor Selection via Convex Optimization". *IEEE Transactions on Signal Processing* 57.2 (2009), pp. 451–462 (cit. on pp. 25, 86–91).

[Kai, 2017] Jean-Marie Kai, Guillaume Allibert, Minh-Duc Hua, and Tarek Hamel. "Nonlinear Feedback Control of Quadrotors Exploiting First-Order Drag Effects". *IFAC-PapersOnLine*. 20th IFAC World Congress 50.1 (2017), pp. 8189–8195 (cit. on p. 70).

[Kamel, 2017] Mina Kamel, Javier Alonso-Mora, Roland Siegwart, and Juan Nieto. "Robust Collision Avoidance for Multiple Micro Aerial Vehicles Using Nonlinear Model Predictive Control". *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 236–243 (cit. on p. 33).

[Karlin, 1966] Samuel Karlin and William J. Studden. "Optimal Experimental Designs". *The Annals of Mathematical Statistics* 37.4 (1966), pp. 783–815. JSTOR: 2238570 (cit. on p. 15).

[Kaveti, 2024] Pushyami Kaveti, Matthew Giamou, Hanumant Singh, and David M. Rosen. "OASIS: Optimal Arrangements for Sensing in SLAM". *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 2024, pp. 13818–13824 (cit. on pp. 26, 88).

[Kiefer, 1959] J. Kiefer. "Optimum Experimental Designs". *Journal of the Royal Statistical Society. Series B (Methodological)* 21.2 (1959), pp. 272–319. JSTOR: 2983802 (cit. on p. 15).

[Kim, 2021] Sung-Kyun Kim, Amanda Bouman, Gautam Salhotra, David D. Fan, Kyohei Otsu, Joel Burdick, et al. “PLGRIM: Hierarchical Value Learning for Large-scale Exploration in Unknown Environments”. *Proceedings of the International Conference on Automated Planning and Scheduling* 31 (2021), pp. 652–662 (cit. on p. 11).

[Klenske, 2016] Edgar D. Klenske and Philipp Hennig. “Dual Control for Approximate Bayesian Reinforcement Learning”. *Journal of Machine Learning Research* 17.127 (2016), pp. 1–30 (cit. on pp. 14, 15, 28, 29).

[Ko, 1995] Chun-Wa Ko, Jon Lee, and Maurice Queyranne. “An Exact Algorithm for Maximum Entropy Sampling”. *Operations Research* 43.4 (1995), pp. 684–691. JSTOR: 171694 (cit. on pp. 25, 88).

[Kontitsis, 2013] Michail Kontitsis, Evangelos A. Theodorou, and Emanuel Todorov. “Multi-Robot Active SLAM with Relative Entropy Optimization”. *2013 American Control Conference*. 2013, pp. 2757–2764 (cit. on pp. 22, 52).

[Krause, 2008] Andreas Krause, Ajit Singh, and Carlos Guestrin. “Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies.” *Journal of Machine Learning Research* 9.2 (2008) (cit. on pp. 25, 88).

[Krener, 2009] Arthur J. Krener and Kayo Ide. “Measures of Unobservability”. *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*. 2009, pp. 6401–6406 (cit. on p. 20).

[Kulikova, 2015] M. V. Kulikova and J. V. Tsyganova. “Constructing Numerically Stable Kalman Filter-Based Algorithms for Gradient-Based Adaptive Filtering”. *International Journal of Adaptive Control and Signal Processing* 29.11 (2015), pp. 1411–1426 (cit. on p. 43).

[Lauri, 2023] Mikko Lauri, David Hsu, and Joni Pajarinen. “Partially Observable Markov Decision Processes in Robotics: A Survey”. *IEEE Transactions on Robotics* 39.1 (2023), pp. 21–40 (cit. on pp. 3, 10).

[Leung, 2006] Cindy Leung, Shoudong Huang, Ngai Kwok, and Gamini Dissanayake. “Planning under Uncertainty Using Model Predictive Control for Information Gathering”. *Robotics and Autonomous Systems* (2006) (cit. on pp. 19, 22, 52).

[Levin, 1960] M. Levin. “Optimum Estimation of Impulse Response in the Presence of Noise”. *IRE Transactions on Circuit Theory* 7.1 (1960), pp. 50–56 (cit. on p. 15).

[Li, 2023] Jianan Li, Zian Ning, Shaoming He, Chang-Hun Lee, and Shiyu Zhao. “Three-Dimensional Bearing-Only Target Following via Observability-Enhanced Helical Guidance”. *IEEE Transactions on Robotics* 39.2 (2023), pp. 1509–1526 (cit. on pp. 19, 21, 24, 33, 67, 73–75, 77, 84).

[Liu, 2025] Lingya Liu, Yiyin Wang, and Cunqing Hua. “Global and Fast Refinement of Greedy Sensor Selection Algorithms for Linear Models”. *IEEE Signal Processing Letters* 32 (2025), pp. 2574–2578 (cit. on p. 86).

[Liu, 2016] Sijia Liu, Sundeep Prabhakar Chepuri, Makan Fardad, Engin Maşazade, Geert Leus, and Pramod K. Varshney. “Sensor Selection for Estimation with Correlated Measurement Noise”. *IEEE Transactions on Signal Processing* 64.13 (2016), pp. 3509–3522 (cit. on pp. 25, 86, 88).

[Long, 2013] Andrew W. Long, Kevin C. Wolfe, Michael J. Mashner, and Gregory S. Chirikjian. “The Banana Distribution Is Gaussian: A Localization Study with Exponential Coordinates” (2013) (cit. on p. 53).

[Mania, 2019] Horia Mania, Stephen Tu, and Benjamin Recht. “Certainty Equivalence Is Efficient for Linear Quadratic Control”. *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019 (cit. on p. 15).

[Mehra, 1974] R. Mehra. “Optimal Input Signals for Parameter Estimation in Dynamic Systems–Survey and New Results”. *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 753–768 (cit. on pp. 15, 17, 57).

[Mellinger, 2011] Daniel Mellinger and Vijay Kumar. “Minimum Snap Trajectory Generation and Control for Quadrotors”. *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 2520–2525 (cit. on pp. 70, 79).

[Mesbah, 2018] Ali Mesbah. “Stochastic Model Predictive Control with Active Uncertainty Learning: A Survey on Dual Control”. *Annual Reviews in Control* 45 (2018), pp. 107–117 (cit. on p. 14).

[Meurer, 2017] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, et al. “SymPy: Symbolic Computing in Python”. *PeerJ Computer Science* 3 (2017), e103 (cit. on p. 43).

[Mihaylova, 2003] Lyudmila Mihaylova, Tine Lefebvre, Herman Bruyninckx, Klaas Gadeyne, and Joris De Schutter. “A Comparison of Decision Making Criteria and Optimization Methods for Active Robotic Sensing”. *Numerical Methods and Applications*. Ed. by Ivan Dimov, Ivan Lirkov, Svetozar Margenov, and Zahari Zlatev. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2003, pp. 316–324 (cit. on pp. 19, 20, 96).

[Miller, 2013] Lauren M. Miller and Todd D. Murphey. "Trajectory Optimization for Continuous Ergodic Exploration". *2013 American Control Conference*. 2013, pp. 4196–4201 (cit. on p. 94).

[Miller, 2016] Lauren M. Miller, Yonatan Silverman, Malcolm A. MacIver, and Todd D. Murphey. "Ergodic Exploration of Distributed Information". *IEEE Transactions on Robotics* 32.1 (2016), pp. 36–52 (cit. on pp. 94, 95).

[Moon, 2022] Brady Moon, Satrajit Chatterjee, and Sebastian Scherer. "TIGRIS: An Informed Sampling-based Algorithm for Informative Path Planning". *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 5760–5766 (cit. on p. 21).

[Moreno-Salinas, 2013] David Moreno-Salinas, Antonio M. Pascoal, and Joaquin Aranda. "Optimal Sensor Placement for Multiple Target Positioning with Range-Only Measurements in Two-Dimensional Scenarios". *Sensors (Basel, Switzerland)* 13.8 (2013), pp. 10674–10710 (cit. on pp. 25, 26).

[Murali, 2019] Varun Murali, Igor Spasojevic, Winter Guerra, and Sertac Karaman. "Perception-Aware Trajectory Generation for Aggressive Quadrotor Flight Using Differential Flatness". *2019 American Control Conference (ACC)*. 2019, pp. 3936–3943 (cit. on pp. 20, 23, 24, 70, 74, 84).

[Napolitano, 2021] Olga Napolitano, Daniele Fontanelli, Lucia Pallottino, and Paolo Salaris. "Gramian-Based Optimal Active Sensing Control under Intermittent Measurements". *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 9680–9686 (cit. on pp. 20, 22, 23).

[Napolitano, 2024] Olga Napolitano, Annamaria Pinizzotto, Matteo Verdecchia, Alessio Pettinari, Daniela Selvi, Lucia Pallottino, et al. "Active Sensing Control for Differentially Flat Systems". *IEEE Control Systems Letters* 8 (2024), pp. 1835–1840 (cit. on pp. 20–22, 70).

[Ning, 2024] Zian Ning, Yin Zhang, Jianan Li, Zhang Chen, and Shiyu Zhao. "A Bearing-Angle Approach for Unknown Target Motion Analysis Based on Visual Measurements". *The International Journal of Robotics Research* 43.8 (2024), pp. 1228–1249 (cit. on pp. 67, 73).

[Nocedal, 1999] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 1999 (cit. on pp. 38, 56, 74, 79, 95).

[Okamoto, 2018] Kazuhide Okamoto, Maxim Goldshtein, and Panagiotis Tsiotras. "Optimal Covariance Control for Stochastic Systems Under Chance Constraints". *IEEE Control Systems Letters* 2.2 (2018), pp. 266–271 (cit. on p. 28).

[Pagano, 2025] Francesca Pagano, Nicola De Carli, Esteban Restrepo, Antonio Marino, and Paolo Robuffo Giordano. "Distributed Multi-Robot Active-Sensing of a Diffusive Source". *IEEE Robotics and Automation Letters* 10.10 (2025), pp. 10807–10814 (cit. on pp. 19, 26, 84, 87).

[Parellier, 2023a] C. Parellier, A. Barrau, and S. Bonnabel. "A Computationally Efficient Global Indicator to Detect Spurious Measurement Drifts in Kalman Filtering". *2023 62nd IEEE Conference on Decision and Control (CDC)*. 2023, pp. 8641–8646 (cit. on p. 42).

[Parellier, 2023b] Colin Parellier, Axel Barrau, and Silvere Bonnabel. "Speeding-Up Backpropagation of Gradients Through the Kalman Filter via Closed-Form Expressions". *IEEE Transactions on Automatic Control* 68.12 (2023), pp. 8171–8177 (cit. on pp. 42, 44, 49, 99).

[Paszke, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, et al. "Automatic Differentiation in PyTorch" (2017) (cit. on pp. 43, 80, 81).

[Paszke, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. arXiv: 1912.01703 [cs, stat] (cit. on p. 59).

[Patil, 2015] Sachin Patil, Gregory Kahn, Michael Laskey, John Schulman, Ken Goldberg, and Pieter Abbeel. "Scaling up Gaussian Belief Space Planning through Covariance-Free Trajectory Optimization and Automatic Differentiation". *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*. Ed. by H. Levent Akin, Nancy M. Amato, Volkan Isler, and A. Frank van der Stappen. Cham: Springer International Publishing, 2015, pp. 515–533 (cit. on pp. 19, 20, 22, 35, 44, 52, 55, 56, 59, 67).

[Pendleton, 2017] Scott Drew Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Hong Eng, et al. "Perception, Planning, Control, and Coordination for Autonomous Vehicles". *Machines* 5.1 (2017), p. 6 (cit. on p. 2).

[Penin, 2018] Bryan Penin, Paolo Robuffo Giordano, and François Chaumette. "Vision-Based Reactive Planning for Aggressive Target Tracking While Avoiding Collisions and Occlusions". *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3725–3732 (cit. on pp. 23, 24, 67).

[Petersen, 2012] Kaare Brandt Petersen and Michael Syskind Pedersen. “The Matrix Cookbook”. *[Online]* (2012) (cit. on pp. 45, 57).

[Placed, 2023] Julio A. Placed, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, et al. “A Survey on Active Simultaneous Localization and Mapping: State of the Art and New Frontiers”. *IEEE Transactions on Robotics* 39.3 (2023), pp. 1686–1705 (cit. on p. 11).

[Platt, 2010] Robert Platt, Russ Tedrake, Leslie Kaelbling, and Tomás Lozano-Pérez. “Belief Space Planning Assuming Maximum Likelihood Observations”. *Proceedings of the Robotics: Science and Systems Conference, 6th* (2010) (cit. on p. 22).

[Porat, 1981] B. Porat and I. Y. Bar-Izack. “Effect of Acceleration Switching during INS In-Flight Alignment”. *Journal of Guidance and Control* 4.4 (1981), pp. 385–389 (cit. on pp. 21, 67, 79, 82, 84).

[Preiss, 2018] James Preiss, Karol Hausman, Gaurav Sukhatme, and Stephan Weiss. “Simultaneous Self-Calibration and Navigation Using Trajectory Optimization”. *The International Journal of Robotics Research* 37 (2018) (cit. on pp. 17, 20–23, 35, 51, 52, 55, 56, 58, 59, 67, 70, 72, 75, 77, 78, 84).

[Qin, 2024] Chao Qin, Maxime S. J. Michet, Jingxiang Chen, and Hugh H.-T. Liu. *Time-Optimal Gate-Traversing Planner for Autonomous Drone Racing*. 2024. arXiv: 2309.06837 (cit. on pp. 67, 72).

[Rafieisakhaei, 2017] Mohammadhussein Rafieisakhaei, Suman Chakravorty, and P. R. Kumar. “On the Use of the Observability Gramian for Partially Observed Robotic Path Planning Problems”. *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2017, pp. 1523–1528. arXiv: 1801.09877 [cs] (cit. on pp. 20, 51, 52, 54, 55, 72, 76).

[Salaris, 2019] Paolo Salaris, Marco Cognetti, Riccardo Spica, and Paolo Robuffo Giordano. “Online Optimal Perception-Aware Trajectory Generation”. *IEEE Transactions on Robotics* 35.6 (2019), pp. 1307–1322 (cit. on pp. 17, 20–23, 35, 51, 55, 57, 58, 61, 67, 70, 72, 75, 76).

[Särkkä, 2023] Simo Särkkä and Lennart Svensson. *Bayesian Filtering and Smoothing*. 2nd ed. Institute of Mathematical Statistics Textbooks. Cambridge: Cambridge University Press, 2023 (cit. on p. 43).

[Shamaiah, 2010] Manohar Shamaiah, Siddhartha Banerjee, and Haris Vikalo. “Greedy Sensor Selection: Leveraging Submodularity”. *49th IEEE Conference on Decision and Control (CDC)*. 2010, pp. 2572–2577 (cit. on pp. 25, 88).

[Singer, 1970] Robert A. Singer. “Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets”. *IEEE Transactions on Aerospace and Electronic Systems* AES-6.4 (1970), pp. 473–483 (cit. on p. 73).

[Sun, 2008] Yu Sun and John M. Hollerbach. “Active Robot Calibration Algorithm”. *2008 IEEE International Conference on Robotics and Automation*. 2008, pp. 1276–1281 (cit. on p. 22).

[Takahashi, 2023] Shun Takahashi, Yasuo Sasaki, Takayuki Nagata, Keigo Yamada, Kumi Nakai, Yuji Saito, et al. “Sensor Selection by Greedy Method for Linear Dynamical Systems: Comparative Study on Fisher-Information-Matrix, Observability-Gramian and Kalman-Filter-Based Indices”. *IEEE Access* 11 (2023), pp. 67850–67864 (cit. on pp. 25, 86, 88, 90).

[Tedrake, 2023] Russ Tedrake. *Underactuated Robotics. Algorithms for Walking, Running, Swimming, Flying, and Manipulation*. 2023 (cit. on p. 27).

[Thrun, 2005] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Ed. by Ronald C. Arkin, Associate Editors: Henrik Christensen, Edmund Durfee, Cecilia Laschi, Robin R. Murphy, and Michael Wooldridge. Intelligent Robotics and Autonomous Agents Series. Cambridge, MA, USA: MIT Press, 2005 (cit. on pp. 3, 12).

[Tse, 1973] E. Tse, Y. Bar-Shalom, and L. Meier. “Wide-Sense Adaptive Dual Control for Nonlinear Stochastic Systems”. *IEEE Transactions on Automatic Control* 18.2 (1973), pp. 98–108 (cit. on pp. 12, 14).

[Van Brummelen, 2018] Jessica Van Brummelen, Marie O’Brien, Dominique Gruyer, and Homayoun Najjaran. “Autonomous Vehicle Perception: The Technology of Today and Tomorrow”. *Transportation Research Part C: Emerging Technologies* 89 (2018), pp. 384–406 (cit. on p. 2).

[van den Berg, 2011] Jur van den Berg, Pieter Abbeel, and Ken Goldberg. “LQG-MP: Optimized Path Planning for Robots with Motion Uncertainty and Imperfect State Information”. *The International Journal of Robotics Research* 30.7 (2011), pp. 895–913 (cit. on pp. 28, 29).

[Velasco-Hernandez, 2020] Gustavo Velasco-Hernandez, De Jong Yeong, John Barry, and Joseph Walsh. “Autonomous Driving Architectures, Perception and Data Fusion: A Review”. *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*. 2020, pp. 315–321 (cit. on p. 2).

[Virtanen, 2020] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. *Nature Methods* 17.3 (2020), pp. 261–272 (cit. on p. 58).

[Vitus, 2011] Michael P. Vitus and Claire J. Tomlin. “Sensor Placement for Improved Robotic Navigation”. *Robotics: science and Systems VI* 217 (2011) (cit. on p. 26).

[Wang, 2013] Weiran Wang and Miguel Á Carreira-Perpiñán. *Projection onto the Probability Simplex: An Efficient Algorithm with a Simple Proof, and an Application*. 2013. arXiv: 1309.1541 [cs] (cit. on p. 89).

[Wasiela, 2023] Simon Wasiela, Paolo Robuffo Giordano, Juan Cortés, and Thierry Siméon. “A Sensitivity-Aware Motion Planner (SAMP) to Generate Intrinsically-Robust Trajectories”. *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 12707–12713 (cit. on pp. 20, 28, 29).

[Willems, 2004] J.C. Willems, I. Markovsky, P. Rapisarda, and B.L.M. De Moor. “A Note on Persistency of Excitation”. *2004 43rd IEEE Conference on Decision and Control (CDC)*. Vol. 3. 2004, 2630–2631 Vol.3 (cit. on pp. 21, 84).

[Wittenmark, 1995] Björn Wittenmark. “Adaptive Dual Control Methods: An Overview”. *IFAC Proceedings Volumes*. 5th IFAC Symposium on Adaptive Systems in Control and Signal Processing 1995, Budapest, Hungary, 14-16 June, 1995 28.13 (1995), pp. 67–72 (cit. on p. 13).

[Xia, 2023] Xin Xia, Neel P. Bhatt, Amir Khajepour, and Ehsan Hashemi. “Integrated Inertial-LiDAR-Based Map Matching Localization for Varying Environments”. *IEEE Transactions on Intelligent Vehicles* 8.10 (2023), pp. 4307–4318 (cit. on p. 89).

[Xing, 2023] Jiaxu Xing, Giovanni Cioffi, Javier Hidalgo-Carrió, and Davide Scaramuzza. “Autonomous Power Line Inspection with Drones via Perception-Aware MPC”. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023, pp. 1086–1093 (cit. on pp. 20, 24, 67).

[Xu, 2024] Liang Xu and Ruixin Niu. “EKFNet: Learning System Noise Covariance Parameters for Nonlinear Tracking”. *IEEE Transactions on Signal Processing* 72 (2024), pp. 3139–3152 (cit. on pp. 44, 49, 99).

[Xu, 2023] Shida Xu, Jonatan Scharff Willners, Ziyang Hong, Kaicheng Zhang, Yvan R. Petillot, and Sen Wang. “Observability-Aware Active Extrinsic Calibration of Multiple Sensors”. *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 2091–2097 (cit. on pp. 19, 51).

[Yang, 2018] Scott Cheng-Hsin Yang, Daniel M. Wolpert, and Máté Lengyel. “Theoretical Perspectives on Active Sensing”. *Current opinion in behavioral sciences* 11 (2018), pp. 100–108 (cit. on p. 21).

[Yu, 1973] T. K. Yu and J. H. Seinfeld. “Observability and Optimal Measurement Location in Linear Distributed Parameter Systems”. *International Journal of Control* 18.4 (1973), pp. 785–799 (cit. on pp. 25, 86, 88, 90).

[Zhang, 1992] H. Zhang. “Optimal Sensor Placement”. *Proceedings 1992 IEEE International Conference on Robotics and Automation*. 1992, 1825–1830 vol.2 (cit. on p. 26).

[Zhao, 2013] Shiyu Zhao, Ben M. Chen, and Tong H. Lee. “Optimal Sensor Placement for Target Localisation and Tracking in 2D and 3D”. *International Journal of Control* 86.10 (2013), pp. 1687–1704 (cit. on p. 19).

[Zhou, 2021] Boyu Zhou, Jie Pan, Fei Gao, and Shaojie Shen. “RAPTOR: Robust and Perception-Aware Trajectory Replanning for Quadrotor Fast Flight”. *IEEE Transactions on Robotics* 37.6 (2021), pp. 1992–2009 (cit. on p. 24).

RÉSUMÉ

La localisation et la perception sont des fonctions fondamentales d'un système autonome. Elles reposent sur des données issues de capteurs, généralement partielles et incertaines. Dans certains cas, la trajectoire suivie par le système influence fortement la qualité de ces données : certaines trajectoires sont peu informatives et conduisent à une estimation imprécise, tandis que d'autres améliorent sensiblement la perception/localisation. Exploiter ce phénomène pour planifier des trajectoires qui optimisent la qualité de l'information constitue le principe de la perception active, au cœur de cette thèse.

La première contribution consiste à formuler ce problème dans un cadre de contrôle optimal déterministe, en intégrant explicitement l'incertitude issue d'un filtre de Kalman étendu (EKF) le long de la trajectoire. La covariance du filtre est utilisée pour modéliser la qualité de la perception, et un terme lié à cette covariance est ajouté à la fonction de coût afin de favoriser les trajectoires les plus informatives. Cette intégration permet non seulement d'améliorer la perception et la localisation, mais aussi de renforcer la validité du cadre déterministe utilisé pour la planification.

La seconde contribution porte sur la résolution efficace de ce problème. Nous enrichissons les équations de rétropropagation d'un filtre de Kalman étendu afin de calculer analytiquement le gradient de la covariance par rapport aux variables influençant la trajectoire. Ces gradients se révèlent particulièrement adaptés à la perception active et rendent les problèmes d'optimisation numériquement gérables. Nous montrons enfin que cette approche est générale et applicable à un large éventail de situations, notamment la localisation, la calibration en ligne de capteurs et l'optimisation de séquences d'acquisition de mesures.

MOTS CLÉS

Perception Active, Filtre de Kalman Etendu, Rétropropagation, Planification de Trajectoires

ABSTRACT

Localization and perception are fundamental functions of an autonomous system. They rely on sensor data, which are generally partial and uncertain. In some cases, the trajectory followed by the system strongly affects the quality of these data: certain trajectories are weakly informative and lead to poor estimation accuracy, while others significantly improve perception and localization. Exploiting this phenomenon to plan trajectories that optimize information quality is the core principle of active sensing, which lies at the heart of this thesis.

The first contribution is to formulate this problem within a deterministic optimal control framework, explicitly incorporating the uncertainty provided by an Extended Kalman Filter (EKF) along the trajectory. The filter covariance is used to model perception quality, and a term related to this covariance is added to the cost function in order to promote more informative trajectories. This integration not only improves perception and localization but also strengthens the validity of the deterministic framework used for planning.

The second contribution concerns the efficient solution of this problem. We extend the backpropagation equations of an Extended Kalman Filter to analytically compute the gradient of the covariance with respect to the variables that influence the trajectory. These gradients are particularly well suited for active perception and make the optimization problems tractable. Finally, we show that this approach is general and applicable to a wide range of situations, including localization, online sensor calibration, and optimal measurement selection.

KEYWORDS

Active Sensing, Extended Kalman Filter, Backpropagation, Trajectory Generation